

# PROGRAMAR

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

EDIÇÃO #40 - ABRIL 2013

ISSN 1647-0710



## OBJECTOS VS ESTRUTURAS



SAIBA MAIS AQUI DENTRO

### A PROGRAMAR

**LUA** LINGUAGEM DE PROGRAMAÇÃO  
PARTE 11

**WINDOWS** WORKFLOW FOUNDATION  
INTRODUÇÃO

**P VS NP**

### ANÁLISES

**MATLAB**

**JQUERY**

**DESENVOLVIMENTO** DE APLICAÇÕES  
EM PHP

### COLUNAS

TRAZENDO ASYNC E AWAIT AO SERVIÇO DE CONTACTOS DO WINDOWS PHONE **C#**

A DOCUMENTAÇÃO É ALGO IMPRESCINDÍVEL **CORE DUMP**

OPERADOR LIKE **VISUAL(NOT)BASIC**

### COMUNIDADES

PÁGINA AVANÇADA "SOBRE" PARA APLICAÇÕES DE WINDOWS PHONE **NETPONTO**

### NO CODE

INTRODUÇÃO AO **RASPBERRY**

KEYWORDS PARTE 1 **SEO**

**CRIANÇAS DIGITAIS**

## EQUIPA PROGRAMAR

**Coordenador**  
António Santos

**Editor**  
António Santos

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

**Redacção**  
António Santos  
Augusto Manzano  
Bruno Almeida  
Fábio Domingos  
Fernando Martins  
Jorge Paulino  
Miguel Araújo  
Miguel Lobato  
Nuno Santos  
Paulo Morgado  
Rita Peres  
Sara Santos  
Sara Silva

**Staff**  
Ana Barbosa  
António Pedro Cunha  
António Santos  
António Silva  
Fábio Domingos  
Jorge Paulino  
Sara Santos

**Contacto**  
[revistaprogramar@portugal-a-programar.org](mailto:revistaprogramar@portugal-a-programar.org)

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

**ISSN**  
1 647-071 0

## Erro 103 (net::ERR\_CONNECTION\_ABORTED)

Sendo edição de Abril e de dia 1 de Abril ainda pensei em escrever alguma brincadeira no editorial, mas tal seria de muito mau tom, pelo que decidi em contrário!

Como tem vindo a ser sequência ao longo das últimas edições, tenho escolhido os mais variados códigos de erro para título e nesta edição para não me afastar daquilo que tem sido o editorial, escolhi um novo mas diferente. Desde a última edição muita coisa mudou no mundo da tecnologia. O vosso feedback tem sido uma grande mais-valia para nós que vos trazemos a revista edição após edição.

Nesta edição, um pouco mais “atribulada” com muitas coisas a serem feitas em cima da hora, mas sempre com o profissionalismo de uma equipe que acredita no seu trabalho, chegamos um pouco mais tarde, mas chegamos! Não foi desta, nem se perspectiva um “ERR\_CONNECTION\_ABORTED” para a revista.

Estamos a pouco tempo do evento da Comunidade Portugal-a-Programar, pelo que não posso deixar de salientar que a revista é feita por nós, para vós, da comunidade, para a comunidade e todos aqueles que lêem esta nossa e vossa edição!

O mundo anda a tecnologia evolui, nós crescemos convosco, mas sempre num formato “aberto”. Aberto à vossa participação, aberto à vossa opinião, aberto à opinião de todos quantos decidirem participar, para que nunca se veja na revista um “ERR\_CONNECTION\_ABORTED” e a revista possa continuar livre e imparcial como a editamos.

Este mês comemora-se mais um aniversário da “liberdade” e não querendo fugir do tema da revista, mas sem deixar passar uma parte da nossa história comum, gostaria de lembrar que graças à liberdade “aquela que acaba quando a dos outros começa”, podemos hoje editar esta publicação sem constrangimentos nem “vitorias”, de um qualquer lápis colorido, a aprovar aquilo que os autores que conosco colaboram e nós enquanto revista, escrevemos.

Num editorial que se faz extenso mas sem grande razão para o ser resta-me agradecer a todos a vossa participação com um muito obrigado e um “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](#) Estruturas vs Objetos (**Rita Peres**)

## A PROGRAMAR

- [22](#) Lua - Linguagem de Programação (**Augusto Manzano**)
- [24](#) Introdução à Windows Workflow Foundation (**Fábio Domingos**)
- [34](#) P vs NP (**Miguel Araújo**)

## COLUNAS

- [39](#) **C#** - Trazendo Async E Await ao Serviço de Contactos do Windows Phone (**Paulo Morgado**)
- [42](#) **Visual(not)Basic** - Operador Like (**Jorge Paulino**)
- [44](#) **Core Dump** - A documentação é algo imprescindível (**Fernando Martins**)

## EVENTOS

- [46](#) II Mega Mostra IPVC

## ANÁLISES

- [48](#) jQuery : A sua biblioteca de Javascript (**Bruno Almeida**)
- [49](#) Matlab Curso Completo (**Nuno Santos**)
- [50](#) Desenvolvimento de Aplicações em PHP (**António Santos**)

## COMUNIDADES

- [52](#) NetPonto - Página avançada "sobre" para aplicações Windows Phone (**Sara Silva**)

## NO CODE

- [65](#) Raspberry Pi - Pequeno mas grande (**Rita Peres**)
- [67](#) SEO - Keywords Parte I (**Miguel Lobato**)
- [70](#) Crianças Digitais (**Sara Santos**)
- [72](#) Entrevista: **Rui Guimarães**
- [74](#) Projecto em Destaque na Comunidade P@P: Sueca Online

## EVENTOS

6 de Abril	Edubits 2013 (Aveiro - Universidade de Aveiro)
28 de Abril	38ª Reunião Presencial da Comunidade NetPonto em Lisboa
15 a 17 Maio	UX-Lx - User Experience Lisbon
<b>25 de Maio</b>	<b>1º Evento Presencial da Comunidade Portugal-a-Programar</b>
28 a 30 de Maio	ISEL Tech '13 (Lisboa - ISEL)

Para mais informações/eventos: [http://bit.ly/PAP\\_Eventos](http://bit.ly/PAP_Eventos). Divulga os teus eventos para o email [eventos@portugal-a-programar.pt](mailto:eventos@portugal-a-programar.pt)

## Maior ataque de sempre afeta desempenho da Internet

Serviços como o Netflix estão entre os mais prejudicados por um ataque massivo que teve como alvo a infraestrutura da Internet. Tudo começou com uma disputa entre um grupo de combate ao spam e uma empresa holandesa.

Os especialistas foram rápidos a classificar este ataque como o maior de sempre à infraestrutura da Internet e os efeitos fazem-se sentir no acesso a diversos serviços e sites web, escreve o jornal The New York Times.

"Este é o maior ataque DDoS anunciado publicamente na história da Internet", afirmou ao jornal Patrick Gilmore, chief architect da Akamai Networks, uma empresa que faz o acompanhamento do tráfego na rede e que hoje regista um crescimento de 142% no número de ataques detetados.

Para além da lentidão no acesso a páginas web, os técnicos responsáveis pela gestão da rede a nível global mostram preocupação com a escalada dos ataques, que estão a subir de intensidade e que podem vir a tornar inacessíveis serviços considerados básicos, como email ou homebanking.

Os ataques são dirigidos ao sistema de resolução de nomes na Internet (DNS - Domain Name System) e foram já relatados na semana passada por uma empresa de segurança norte-americana. O método usado pelos atacantes é o denial of service, DDoS, e segundo a mesma fonte estão a atingir picos de magnitude, crescendo para 300 mil milhões por segundo.

Os ataques terão tido origem numa disputa entre a Spamhaus - uma organização que regista entidades envolvidas no envio de spam - e uma empresa holandesa de alojamento, a Cyberbunker. A Spamhaus adicionou a Cyberbunker à sua lista negra e esta terá retaliado com ataques DDoS, que têm um efeito mais lato do que o objetivo inicial de evitar que a Spamhaus distribuisse a sua lista.

A Cyberbunker conta a sua versão da história no seu site, mas nunca admite ser a origem dos ataques.

A BBC escreve que há cinco grupos de combate ao cibercrime de diferentes países a investigar os ataques e cita os responsáveis da Spamhaus, que afirmam estar a combater o ataque há mais de uma semana, com sucesso, mantendo os seus mais de 80 servidores a funcionar.

Mas este foi classificado pelos responsáveis da Spamhaus como um ataque de "uma dimensão sem precedentes", embora estejam habituados a retaliações em modelos semelhantes de serviços que são colocados na lista negra.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo

## IBM desmantela primeiro supercomputador a processar um Petaflop por segundo

O avanço tecnológico é de tal forma veloz que o supercomputador mais rápido de 2008 vai ser descontinuado passados cinco anos por ser "obsoleto" em eficiência.

O Roadrunner da IBM foi o primeiro computador no mundo a atingir a marca de um Petaflop de cálculos por segundo de forma sustentada e chegou mesmo a liderar a lista dos supercomputadores mais potentes em atividade. Mas a máquina de 125 milhões de dólares vai ser substituído pelo Cielo, também ele um supercomputador mas que tem uma relação "processamento de dados-consumo energético" mais reduzido.

O complexo de processamento Roadrunner que vai ser gradualmente desmantelado ainda figura no top 25 dos supercomputadores mais potentes do mundo, o que até podia fazer com que parecesse uma mentira do primeiro de abril não tivesse sido anunciado ainda durante o dia 29 de março pelo Laboratório Nacional de Los Alamos.

Durante cinco anos estiveram em funcionamento 12,960 processadores gráficos IBM PowerXCell 8i, 6,480 processadores AMD Opteron de dois núcleos, 114TB de memória RAM e 1,09 milhões de terabytes de espaço de armazenamento. Todos estes componentes foram usados para suportar investigações na área dos lasers, nanotecnologia, investigação nuclear e estudo do universo.

Agora o Roadrunner vai continuar em funcionamento mas para analisar outros tipos de dados como compressão de memória em sistemas operativos ou otimização da distribuição de dados. O supercomputador ocupa uma área de 557,4 metros quadrados e tem os vários processadores ligados entre si por mais de 88 quilómetros de fibra ótica.

"O Roadrunner deixou toda a gente a pensar em novas formas de usar e construir os supercomputadores, com a utilização de processador especializados. As nossas demonstrações com o Roadrunner chamaram a atenção", refere em comunicado o investigador do laboratório de computação de alta performance, Gary Grider.

O Laboratório Nacional de Los Alamos acredita ainda que entre 2020 e 2030 os supercomputadores vão atingir a escala dos exabytes, isto é, vão ser mil vezes mais potentes que o IBM Roadrunner.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo

## PC compacto com Android para a educação é o novo sucesso do Kickstarter

Mais um dispositivo que promete brilhar nos próximos meses: o MiiPC é um projeto educativo e ao mesmo tempo um centro de produtividade e multimídia para a família.

O MiiPC, um mini computador baseado em Android, quer dar um novo significado à ligação entre os mais novos e o mundo da tecnologia. Com acesso a aplicações, sites e conteúdos multimídia, o equipamento além de ser portátil tem a vantagem de ter uma forte integração com os dispositivos móveis que permitem o controlo das funcionalidades do computador. Além desta componente, que torna o PC num dispositivo recomendado para crianças e escolas sem que haja o risco de acesso a conteúdos indevidos, o MiiPC tem um preço de 99 dólares que tornam esta máquina uma alternativa a outros dispositivos do mercado.



Ao contrário do Raspberry Pi, o MiiPC vem pronto a funcionar de origem e não precisa de grandes configurações, bastando apenas ser ligado a um monitor ou a uma televisão. Apesar de também ter objetivos educativos, o Raspberry Pi é mais virado para a comunidade de jovens que pretende desenvolver competências na área da informática e da robótica.

O MiiPC é descrito pelos fundadores como uma forma simples de as famílias ganharem de novo o controlo das atividades online, em alusão às aplicações para Android e iOS que vão acompanhar o lançamento do mini computador. Definir quais as apps são permitidas, para quem e durante quanto tempo, são alguns dos exemplos da "liberdade limitada" que o equipamento garante.

Mas mais do que um computador para os mais pequenos, o MiiPC quer ser um equipamento de toda a família e quer roubar o papel de centro multimídia a outros dispositivos. O Android 4.2 vem otimizado de maneira a que possa ser controlado com rato e teclado e a saída HDMI permite a ligação a

quase todo o tipo de ecrãs e televisões recentes. Além disso, o computador permite a criação de contas particulares para que ninguém invada o espaço alheio.

A ZeroDesktop, empresa responsável pelo projeto, está a pedir 50 mil dólares em donativos para que a produção do computador comece e o valor está perto de ser atingido logo no primeiro dia de vida do projeto. Submetido a 25 de março, poucas horas depois já contava com 30 mil dólares de suporte. O valor foi atingido graças a dois investidores de dez mil dólares que compraram o pacote mais caro do projeto. Mas as primeiras duas mil unidades do MiiPC também já estavam esgotadas, restando apenas as versões EarlyBird. Por 99 dólares, os compradores levam um computador com processador Marvel de dois núcleos a 1,2Ghz, 1GB de memória RAM, 4GB de armazenamento interno e entrada para cartões SD, suporte a discos externos por USB, ligação Wi-Fi e Bluetooth 4.0.

Dado o sucesso inicial do projeto, e olhando para outros exemplos, é provável que a MiiPC acrescente novos objetivos de crowdfunding já que a angariação de investimento apenas termina a 9 de maio.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo

## Maioria das empresas usa versões desatualizadas do Java

Três quintos das empresas usam versões do Java desatualizadas há pelo menos seis meses, conclui um estudo levado a cabo pela Websense. A empresa de soluções de segurança analisou a realidade junto dos seus utilizadores e apurou números surpreendentes, tendo em conta toda a divulgação e avisos das empresas de segurança em torno dos problemas com o Java.

No universo de produtos analisados pela Websense com o Runtime do Java desatualizado foi ainda detetado que em 25% dos produtos o software não era atualizado há mais de quatro anos.

Só um em cada 20 produtos analisados executava a última versão do programa, sendo que a larga maioria dos produtos analisados estão a utilizar versões tão antigas do Java que já não são suportadas, corrigidas ou atualizadas. Face aos dados apurados, a Websense concluiu que 94% dos end points analisados estavam vulneráveis à falha mais recentemente detetada no Java. À falha detetada em novembro do ano passado, 75% das empresas mantinham-se vulneráveis.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo

# TEMA DE CAPA

Estruturas vs. Objetos

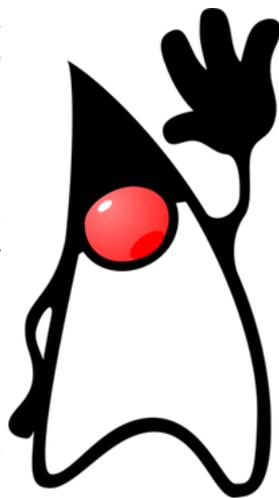
## Estruturas vs. Objetos

No artigo desta edição vamos recordar uma discussão que, possivelmente, nunca terá fim. Estruturas Vs. Objetos, que é como quem diz Linguagem C vs. Linguagem Java. E esta é uma discussão sem prazo para terminar.

Mas, antes de começarmos, vamos recuar no tempo e conhecer um pouco da história destas linguagens.

A linguagem C foi criada, no início da década de 70, por Dennis Ritchie com o objectivo de desenvolver o sistema operacional Unix que inicialmente era apenas escrito em Assembly. Esta linguagem evoluiu rapidamente e, nos finais da década de 70, começou até a substituir a linguagem Basic que anteriormente era a linguagem de microcomputadores mais usada. No mundo Unix, ainda hoje, a Linguagem C é uma das mais populares. A primeira versão padronizada, pela Organização Internacional para a Padronização, do C surgiu no início da década de 90, ficando conhecida como ANSI C.

Por sua vez, a Linguagem Java surgiu na década de 90 pela mão de uma equipa de programadores liderada por James Gosling no Green Project, na empresa Sun Microsystems. Foi criado um protótipo que foi chamado \*7(StarSeven) e foi nesta altura que surgiu o Duke, que ainda hoje é conhecido por ser a mascote do Java. Mais tarde, James Gosling quis renomear, como Oak, o protótipo mas tal não aconteceu por já existir uma outra linguagem com esse mesmo nome. E foi assim que surgiu o nome Java, por ser a cidade de origem do café que a equipa habitualmente bebia (daí o símbolo da Linguagem Java ser uma chávena de café). O objetivo da equipa que criou a Linguagem Java era criar uma linguagem que permitisse uma comunicação fácil entre vários dispositivos electrónicos.



**Imagem 1 - Duke, Mascote da Linguagem Java**



**Imagem 2 - Símbolo Linguagem Java**

Felizmente, no fórum do Portugal-a-Programar há sempre as mais variadas trocas de ideias e entre os nossos leitores há certamente quem prefira a Linguagem C e quem prefira a Linguagem Java. Sei que há até fervorosos adeptos quer de uma quer de outra linguagem. Eu prefiro não me incluir em nenhum dos grupos, porque além de considerar que a minha experiência ainda está a largos passos de ser vasta, se considerar o meu percurso académico, tive formação base em ambas as linguagens, contudo pela força das circunstâncias, a Linguagem C acabou por ser a que mais absorvi.

Ambas as linguagens têm os seus pontos fortes e fracos. Se o Java é uma linguagem orientada a objetos, o C é uma linguagem procedural (linguagem que se baseia no conceito de chamadas de funções que contêm um conjunto de passos computacionais para serem executados). Ambas são linguagens estruturais e imperativas, ou seja, cada programa é baseado em ações e comandos que mudam o estado das variáveis presentes no programa. Também é do conhecimento geral que o C influenciou o Java, e podemos considerar o C++ como uma “mistura” destas duas linguagens.

Sendo uma linguagem procedural, o C tem o seu principal foco não nos dados em si mas nas ações que são feitas sobre esses dados. E, como já referimos há algumas linhas atrás, C tornou-se popular na programação de sistemas operativos em que é necessário escalonar processos ou alocar memória rapidamente.

Uma das características mais conhecidas do Java é a sua portabilidade. Os programas criados em Java são compilados em códigos chamados bytecodes que, por sua vez, são executados recorrendo à máquina virtual do Java. Os bytecodes são traduzidos em tempo real na máquina que os executa. Por este motivo, os programas em Java são teoricamente um pouco mais lentos que os executáveis em C, visto que a sua tradução é feita em tempo real.

Nos softwares que surgiram até ao início da década de 90, a Linguagem C era realmente suficiente pois era rápida e simples. Mas como felizmente o mundo não pára, houve uma nova evolução tecnológica e os programas começaram a ser cada vez maiores. E todos sabemos como é difícil encontrar um erro pequeno na execução do nosso programa quando temos um ficheiro com alguns milhares de linhas de código.

E creio que foi aqui que o Java ganhou a sua primeira vantagem - “Dividir para conquistar” – porque, recorrendo às classes e objectos, foi possível organizar de forma mais simples toda uma estrutura do programa. Tinha “nascido” um dos

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

triângulos mais famosos do meio da informática, “Encapsulamento, Herança e Polimorfismo”.

Se olharmos para as características de uma e outra linguagem podemos verificar que o Java tem o *Garbage Collector* que, por exemplo, avisa o programador de que se uma variável está ou não a ser usada no programa e que, caso a variável não seja utilizada, então é descartada automaticamente de modo a que a memória alocada por essa variável não seja desperdiçada. Infelizmente, a Linguagem C não tem este recurso.

Contudo se usarmos os acessos de baixo níveis do C, podemos incluir código Assembly no meio do nosso programa em C.

E se em C os apontadores trazem mais flexibilidade à linguagem, em Java os mesmos não são permitidos. O Java não permite manipulações de dados usando apontadores.

Mas o Java tem vetores que “crescem” de forma automática, enquanto que a Linguagem C não, isso tem que ser especificado pelo programador usando *realloc*.

Como o caro leitor pode verificar, isto seria uma discussão sem fim. Então vamos voltar ao verdadeiro propósito do nosso artigo... “Estruturas vs. Objetos”. Ora o mais próximo que a linguagem C tem de um objeto é uma estrutura. E se devido à herança da Linguagem Java, os objetos de determinada classe podem herdar todas as características de uma outra classe, na linguagem C se recorrermos aos apontadores podemos fazer o mesmo, colocando uma estrutura “dentro” de outra (Devo recordar o leitor que, também na linguagem C, uma estrutura pode ser chamada dentro de outra estrutura sem ser preciso usar apontadores, contudo na minha opinião acho mais simples recorrer aos apontadores). E é este o principal ponto que este artigo pretende ilustrar. Que Java pode ser mais simples, mas que o C também é flexível.

Quando pensei no mote que ilustra este artigo para a edição de Abril, lembrei-me que faz precisamente um ano que foi publicado o meu primeiro artigo na nossa revista. Então numa espécie de homenagem simples e improvisada foi precisamente a nossa revista que escolhi como exemplo.

Como é do conhecimento de todos os leitores, esta é uma revista em que qualquer um de nós pode aventurar-se a ser escritor caso se proponha a isso. Há uma data para a entrega de artigos e posteriormente há uma escolha para decidir quais os artigos que farão parte da próxima edição.

Então vamos assumir que, quando enviamos os artigos para a revista estes são registados (assim como o autor que os escreveu) num pequeno programa e ali ficam “à espera” que termine o prazo de entrega para fazer a nova edição, pode-

mos também assumir que para cada edição temos um super utilizador que além de introduzir esses artigos também introduz os coordenadores da mesma (que por sua vez são responsáveis por colunas específicas da revista). Quando temos estes dados introduzidos podemos então fazer a nova edição, escolhendo quais os artigos que queremos que façam parte da revista. Após essa escolha podemos então ver a próxima edição.

Após esta breve explicação, passemos então aos respetivos programas quer em Linguagem Java quer em Linguagem C. (Para ser mais simples a comparação pelo leitor, em ambos os programas foram mantidos os mesmos nomes das variáveis assim como os mesmos nomes dos métodos e funções utilizadas. Recordo ainda o leitor que devido à simplicidade dos programas não foram feitos testes de verificação dos valores introduzidos pelo utilizador, partindo do princípio que os valores introduzidos são corretos.)

### Programa em Linguagem JAVA

#### Classe PortugalAProgramar

```
import java.util.*;

public class PortugalAProgramar {

    public static void main (String str[]){
        Vector<Artigo> artigos = new Vector<Artigo>();
        Vector<Coordenador> coordenadores = new
            Vector<Coordenador>();

        int flagCoordenadores=0;
        int numeroArtigos=0;
        int numeroPublicados=0;
        int opcao=0;

        do{

            System.out.println("          --- Revista
                                PROGRAMAR --- ");
            System.out.println(" 1 --> Registos);
            System.out.println(" 2 --> Fazer Edicao");
            System.out.println(" 3 --> Ver Edicao ");
            System.out.println("Escolha uma opcao ( 0 para
                                Terminar )");

            opcao=Leitura.umInteiro();

            int opcaoregisto=-1;
            int opcaofuncionamento=-1;
            int op=-1;
            int opretira=-1;

            switch(opcao){

                case 1: //REGISTOS

                    do{
                        System.out.println(" ----- MENU
                                            REGISTOS -----");
                        System.out.println(" 1 --> Registrar
                                            Artigo e Autor");
                        System.out.println(" 2 --> Registrar
                                            Coordenador ");
                        System.out.println("Escolha uma opcao
                                            ( 0 para voltar ao menu anterior ) ");

                        opcaoregisto=Leitura.umInteiro();
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
switch(opcaoregisto){
    case 1: //Artigo-Autor
        do{
            System.out.println(" ---- MENU REGIS
TOS Artigos Autores -----");
            System.out.println("1 --> Inserir Arti
go e Autor ");
            System.out.println("2 --> Listar Arti
gos/Autores");
            System.out.println("3 --> Consultar
dados Artigo/Autor (por numero de Artigo) ");
            System.out.println("4 --> Modificar
dados Artigo/Autor (por numero de Artigo) ");
            System.out.println("Escolha uma opcao
( 0 para voltar ao menu anterior ");
            op=Leitura.umInteiro();

            switch(op){
                case 1:
                    numeroArtigos= Metodos.insereArtigo
                    (artigos, numeroArtigos);
                    System.out.println(numeroArtigos);
                    break;

                case 2:
                    if(numeroArtigos==0){
                        System.out.println("Nenhum Artigo na
Base de Dados\n");
                    }
                    else
                        Metodos.listarArtigos(artigos);

                    break;

                case 3:
                    if(numeroArtigos==0){
                        System.out.println("Nenhum Artigo
na Base de Dados\n");
                    }
                    else
                        Metodos.consultaNumeroArtigo(artigos);
                    break;

                case 4:
                    if(numeroArtigos==0){
                        System.out.println
                        ("Nenhum Artigo na Base de Dados\n");
                    }
                    else
                        Metodos.modificarArtigo(artigos);

                    break;
            }
        }
        while(op!=0);

        break;

    case 2: //coordenadores
        do{
            System.out.println("-----
MENU REGISTOS Coordenadores -----");
```

```
System.out.println(" 1 --> Inserir
coordenador");
System.out.println(" 2 --> Listar
Coordenadores");
System.out.println(" 3 --> Consultar dados
Coordenador (por numero) ");
System.out.println(" 4 --> Modificar dados
Coordenador ");
System.out.println("Escolha uma opcao ( 0
para voltar ao menu anterior ) ");
op=Leitura.umInteiro();

switch(op){
    case 1: Metodos.insereCoordenador
            (coordenadores);

        flagCoordenadores++;

        break;

    case 2:
        if(flagCoordenadores==0){
            System.out.println("Nenhum Coordenador
na Base de Dados\n");
        }
        else
            Metodos.listarCoordenadores
            (coordenadores);

        break;

    case 3: if(flagCoordenadores==0){
            System.out.println("Nenhum Coordenador
na Base de Dados\n");
        }
        else
            Metodos.consultaNumeroCoordenador
            (coordenadores);

        break;

    case 4: if(flagCoordenadores==0){
            System.out.println("Nenhum Coordenador
na Base de Dados\n");
        }
        else
            Metodos.modificarCoordenador
            (coordenadores);

        break;
    }
}while(op!=0);
break;
}

}while(opcaoregisto!=0);
break;

case 2: //FAZER EDICAO
do{
    System.out.println("----- FAZER
EDICAO ----- ");
    System.out.println(" 1 --> Listar
Artigos Disponiveis ");
    System.out.println(" 2 --> Adicionar
Artigos a edicao ");
    System.out.println(" 3 --> Retirar
Artigos da edicao ");
    System.out.println("Escolha uma opcao
( 0 para voltar ao menu anterior ) ");

    opcaofuncionamento=Leitura.umInteiro();
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
        switch(opcaofuncionamento){
        case 1:
            if((numeroArtigos-numeroPublicados)==0){
                System.out.println("Nenhum Artigo
                    disponivel para mostrar.\n");
            }
            else
                Metodos.listarArtigosDisponiveis(artigos);
            break;
        case 2: //adicionar artigos a edicao

            if(numeroArtigos-numeroPublicados>0)
                numeroPublicados= Metodos.criaEdicao
                    (artigos, numeroPublicados);
            else
                System.out.println("\nNao ha novos artigos
                    disponiveis para publicar.\n");
            break;

            case 3: //retirar artigos da edicao
                if(numeroPublicados>0){
                    do{
                        System.out.println("Ha " +
                            numeroPublicados + " artigos nesta edicao.\n");
                        numeroPublicados=Metodos.retirarArtigoEdicao
                            (artigos, numeroPublicados);
                    }if(numeroPublicados==0)
                        break;
                    System.out.println("\nDeseja retirar outro
                        artigo? ( Prima qualquer numero pra continuar
                            0 para Sair)");
                    opretira=Leitura.umInteiro();
                }while(opretira!=0);
            }
            else
                System.out.println("\nNao ha artigos para
                    remover.\n");
            break;
        }
        }while(opcaofuncionamento!=0);

        break;

        case 3: //ver edicao

            if(numeroPublicados>0){
                System.out.println("Ha " + numeroPublicados +
                    " artigo nesta edicao.\n");
                Metodos.mostraEdicao(artigos);
            }
            else
                System.out.println("\nNao ha ainda artigos
                    adicionados na proxima edicao.\n");
            break;
        }
        }while(opcao!=0);
    }
}
```

### Classe Pessoa

```
public class Pessoa {

    private String nome;
    private String email;
    private String localidade;
    private int telefone;

    public Pessoa(){
```

```
        this.nome="";
        this.email="";
        this.localidade="";
        this.telefone=0;
    }

    public Pessoa(String nome, String email,
        String localidade, int telefone){
        this.nome=nome;
        this.email=email;
        this.localidade=localidade;
        this.telefone=telefone;
    }

    public String getNome(){
        return nome;
    }

    public String getEmail(){
        return email;
    }

    public String getLocalidade(){
        return localidade;
    }

    public int getTelefone(){
        return telefone;
    }

    public void setNome(String nome){
        this.nome=nome;
    }

    public void setEmail(String email){
        this.email=email;
    }

    public void setLocalidade(String localidade){
        this.localidade=localidade;
    }

    public void setTelefone(int telefone){
        this.telefone=telefone;
    }

    public String toString(){
        String p;
        p= " Nome:" + nome + " Email:" + email
            + " Localidade:" + localidade + "
                Telefone:" + telefone ;

        return p;
    }
}
```

### Classe Coordenador

```
public class Coordenador extends Pessoa{

    private static int nCoordenadores=0;
    private int numCoordenador;
    private String coluna;

    public Coordenador(String nome, String email,
        String localidade, int telefone, String coluna){
        super(nome, email, localidade, telefone);
        nCoordenadores++;
        this.numCoordenador=nCoordenadores;
        this.coluna=coluna;
    }

    public int getNumCoordenador(){
        return (this.numCoordenador);
    }

    public String getColuna(){
        return this.coluna;
    }

    public void setColuna(String coluna){
        this.coluna=coluna;
    }

    public void setNumCoordenador(int n){
        this.numCoordenador=n;
    }
}
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
}
public String toString(){
    String a="Numero do Coordenador: " +
        numCoordenador + " Responsavel Coluna: " +
        coluna + super.toString();
    return a;
}
}
```

### Classe Artigo

```
public class Artigo extends Pessoa {

    private static int nArtigo=0;
    private int numArtigo;
    private String nomeArtigo;
    private String coluna;
    private String areaCientifica;
    private int flagPublicado;
    private String publicar;

    public Artigo(String nomeArtigo, String
        coluna, String areaCientifica, String
        nomeAutor, String email, String
        localidade, int telefone){

        super(nomeAutor, email, localidade,
            telefone);

        nArtigo++;
        this.numArtigo=nArtigo;
        this.nomeArtigo=nomeArtigo;
        this.coluna=coluna;
        this.areaCientifica=areaCientifica;
        this.flagPublicado=0;
        this.publicar="Nao Publicado";
    }

    public int getFlagPublicado(){
        return this.flagPublicado;
    }
    public String getPublicar(){
        return this.publicar;
    }
    public int getNumArtigo(){
        return (this.numArtigo);
    }
    public String getNomeArtigo(){
        return(this.nomeArtigo);
    }
    public String getcoluna(){
        return this.coluna;
    }
    public String getAreaCientifica(){
        return this.areaCientifica;
    }
    public void setNumArtigo(int n){
        this.numArtigo=n;
    }
    public void setNomeArtigo(String nomeArtigo){
        this.nomeArtigo=nomeArtigo;
    }
    public void setColuna(String coluna){
        this.coluna=coluna;
    }
    public void setAreaCientifica(String
        areaCientifica){
        this.areaCientifica=areaCientifica;
    }
    public void setPublicar(String publicar){
        this.publicar=publicar;
    }
    public void setFlagPublicado(int
        flagPublicado){
```

```
        this.flagPublicado=flagPublicado;
    }
    public String toString(){
        String a="Nº Artigo: " + numArtigo +
            " Titulo: " + nomeArtigo + " Estado: " +
            publicar + " Coluna: " + coluna +
            " Area Cientifica: " + areaCientifica +
            super.toString();
        return a;
    }
}
```

### Classe Métodos

```
import java.util.*;

public class Metodos {
    /*-----Metodos Artigos - Autores-----*/
    public static int insereArtigo(Vector<Artigo> v,
        int numero)
    {

        int telefone;
        String nome, email, localidade;
        String nomeArtigo, coluna, areaCientifica;

        System.out.println("Insira o nome do novo
            Artigo:");
        nomeArtigo=Leitura.umaString();
        System.out.println("Insira a coluna do novo
            artigo: ");
        coluna = Leitura.umaString();
        System.out.println("Insira a area
            cientifica do artigo: ");
        areaCientifica=Leitura.umaString();
        System.out.println("Insira o nome do
            autor:");
        nome=Leitura.umaString();
        System.out.println("Insira o email do
            autor: ");
        email = Leitura.umaString();
        System.out.println("Insira a localidade do
            autor: ");
        localidade=Leitura.umaString();
        System.out.println("Insira o numero de
            telefone do autor: ");
        telefone=Leitura.umInteiro();

        Artigo novoArtigo= new Artigo(nomeArtigo,
            coluna, areaCientifica, nome, email,
            localidade, telefone);
        v.addElement(novoArtigo);
        System.out.println("Artigo inserido com
            sucesso");

        numero=numero+1;
        return(numero);
    }

    public static void listarArtigos(Vector<Artigo>
        v){

        for(int i=0;i<v.size();i++)
            System.out.println(v.elementAt(i));
    }
    public static void consultaNumeroArtigo
        (Vector<Artigo> v){

        int num;
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
System.out.println("Indique o numero do
                    Artigo: ");
num=Leitura.umInteiro();

for (int i=0; i<v.size();i++){
    Artigo aux = (Artigo)v.elementAt(i);
    int numero=aux.getNumArtigo();
    if (numero==num){
        System.out.println(aux.toString());
        return;
    }
}
System.out.println("Artigo nao
                    encontrado\n");
}

public static void modificarArtigo(Vector<Artigo>
                                   v){

    int telefone;
    String nome, email, localidade, nomeArtigo,
          coluna, areaCientifica;
    char leitura;
    int num;
    System.out.println("Indique o numero do
                       Artigo a Alterar: ");
    num=Leitura.umInteiro();

    for (int i=0; i<v.size();i++){
        Artigo aux = (Artigo)v.elementAt(i);
        int numero=aux.getNumArtigo();

        if (numero==num){
            System.out.println("Deseja
                                modificar o nome do artigo? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||
              (leitura=='S')){
                System.out.println("Insira novo
                                    nome do artigo: ");
                nomeArtigo=Leitura.umaString();
                aux.setNomeArtigo(nomeArtigo);
            }

            System.out.println("Deseja
                                modificar o nome da coluna? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||((leitura=='S'))){
                System.out.println("Insira novo
                                    nome da coluna: ");
                coluna=Leitura.umaString();
                aux.setColuna(coluna);
            }

            System.out.println("Deseja modificar
                                a area cientifica? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||
              (leitura=='S')){
                System.out.println("Insira nova area
                                    cientifica: ");
                areaCientifica=Leitura.umaString();
                aux.setAreaCientifica
                    (areaCientifica);
            }

            System.out.println("Deseja modificar
                                o nome do autor? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||((leitura=='S'))){
                System.out.println("Insira novo
                                    nome do Autor: ");
                nome=Leitura.umaString();
```

```
aux.setNome(nome);
        }

        System.out.println("Deseja
                            modificar o email do autor? ");
        leitura=Leitura.umChar();
        if((leitura=='s')||
          (leitura=='S')){
            System.out.println("Insira novo
                                email do autor: ");
            email=Leitura.umaString();
            aux.setEmail(email);
        }

        System.out.println("Deseja
                            modificar a localidade do autor? ");
        leitura=Leitura.umChar();
        if((leitura=='s')||((leitura=='S'))){
            System.out.println("Insira a nova
                                localidade do autor: ");
            localidade=Leitura.umaString();
            aux.setLocalidade(localidade);
        }

        System.out.println("Deseja
                            modificar o telefone do autor? ");
        leitura=Leitura.umChar();
        if((leitura=='s')||((leitura=='S'))
        {
            System.out.println("Insira novo
                                telefone do autor: ");
            telefone = Leitura.umInteiro();
            aux.setTelefone(telefone);
        }
    }
}

/*-----Metodos Coordenadores -----
-----*/
public static void insereCoordenador
    (Vector<Coordenador> v){// throws Expecao{

    int telefone;

    String nome, email,localidade, coluna;

    System.out.println("Insira o nome do
                        Coordenador: ");
    nome=Leitura.umaString();
    System.out.println("Insira o email do
                        Coordenador: ");
    email = Leitura.umaString();
    System.out.println("Insira a localidade do
                        Coordenador: ");
    localidade=Leitura.umaString();
    System.out.println("Insira o numero de
                        telefone do Coordenador: ");
    telefone=Leitura.umInteiro();

    System.out.println("Insira a coluna do
                        Coordenador: ");
    coluna = Leitura.umaString();

    Coordenador novoCoordenador = new Coordenador
        (nome, email, localidade, telefone, coluna);
    v.addElement(novoCoordenador);

    System.out.println("Coordenador inserido com
                        sucesso");
}

public static void listarCoordenadores
    (Vector<Coordenador> v){
    for(int i=0;i<v.size();i++)
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
System.out.println(v.elementAt(i));
}

public static boolean consultaNumeroCoordenador
(Vector<Coordenador> v){

    int num;
    System.out.println("Indique o numero do
                        Coordenador: ");
    num=Leitura.umInteiro();

    for (int i=0; i<v.size();i++){
        Coordenador aux = (Coordenador)
                            v.elementAt(i);
        int numero=aux.getNumCoordenador();
        if (numero==num){
            System.out.println(aux.toString());
            return true;
        }
    }
    System.out.println("Artigo nao
                        encontrado\n");

    return false;
}

public static void modificarCoordenador
(Vector<Coordenador> v){

    int telefone;
    String nome, email, localidade, coluna;
    char leitura;
    int num;

    System.out.println("Indique o numero do
                        Coordenador a Alterar: ");
    num=Leitura.umInteiro();

    for (int i=0; i<v.size();i++){
        Coordenador aux = (Coordenador)
                            v.elementAt(i);
        int numero=aux.getNumCoordenador();
        if (numero==num){
            System.out.println("Deseja modificar
                                o nome do Coordenador? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||(leitura=='S')){
                System.out.println("Insira novo nome do
                                    Coordenador: ");
                nome=Leitura.umaString();
                aux.setNome(nome);
            }

            System.out.println("Deseja modificar o email do
                                Coordenador? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||(leitura=='S')){
                System.out.println("Insira novo email
                                    do Coordenador: ");
                email=Leitura.umaString();
                aux.setEmail(email);
            }

            System.out.println("Deseja modificar
                                a localidade do Coordenador? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||(leitura=='S')){
                System.out.println("Insira a nova
                                    localidade do Coordenador: ");
                localidade=Leitura.umaString();
                aux.setLocalidade(localidade);
            }

            System.out.println("Deseja modificar
```

```
                o telefone do Coordenador? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||(leitura=='S')){
                System.out.println("Insira novo
                                    telefone do Coordenador: ");
                telefone = Leitura.umInteiro();
                aux.setTelefone(telefone);
            }

            System.out.println("Deseja modificar
                                o nome da coluna? ");
            leitura=Leitura.umChar();
            if((leitura=='s')||(leitura=='S')){
                System.out.println("Insira novo nome
                                    da coluna: ");
                coluna=Leitura.umaString();
                aux.setColuna(coluna);
            }
        }
    }

}

/*----- Menu Fazer Edicao -----*/
-----*/

public static int criaEdicao(Vector <Artigo> v, int
numero){

    char publicar;

    for(int i=0;i<v.size();i++){
        if(v.elementAt(i).getFlagPublicado()==0){
            System.out.println(v.elementAt(i));
            System.out.println("\nDeseja inserir
                                este artigo na proxima edicao?\n");
            publicar=Leitura.umChar();
            if((publicar=='s')||(publicar=='S')){
                v.elementAt(i).setFlagPublicado(1);
                v.elementAt(i).setPublicar
                    ("Publicado");
                numero++;
            }
        }
    }
    return(numero);
}

public static int retirarArtigoEdicao(Vector
<Artigo> v, int numero){

    System.out.println("\nArtigos na Edicao\n");
    for(int i=0;i<v.size();i++){
        if(v.elementAt(i).getFlagPublicado()==1)
            System.out.println(v.elementAt(i));
    }
    System.out.println("\nInsira o numero do
                        artigo a retirar:");
    int retirar=Leitura.umInteiro();

    for(int i=0;i<v.size();i++){
        if (v.elementAt(i).getNumArtigo()
            ==retirar){
            // v.removeElementAt(i);
            v.elementAt(i).setFlagPublicado(0);
            v.elementAt(i).setPublicar("Nao
                                        publicado");

            numero=numero-1;
            System.out.println("\n\nArtigo
                                Removido com Sucesso da edicao.\n");
        }
    }
    return(numero);
}
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
}  
  
public static void mostraEdicao(Vector<Artigo> v){  
    for(int i=0;i<v.size();i++){  
        if(v.elementAt(i).getFlagPublicado()  
           ==1){  
            System.out.println(v.elementAt(i));  
        }  
    }  
}  
  
public static void listarArtigosDisponiveis  
    (Vector<Artigo> v){  
  
    for(int i=0;i<v.size();i++){  
        if(v.elementAt(i).getFlagPublicado()==0){  
            System.out.println(v.elementAt(i));  
        }  
    }  
}  
}
```

### Classe Leitura

```
public class Leitura{  
    public static String umaString() {  
        String s = "";  
        try {  
            BufferedReader in = new BufferedReader(new  
                InputStreamReader(System.in), 1);  
            s = in.readLine();  
        }  
        catch (IOException e) {  
            System.out.println("Erro ao ler fluxo de  
                entrada.");  
        }  
        return s;  
    }  
    public static int umInteiro() {  
        while (true) {  
            try {  
                return Integer.valueOf(umaString().trim  
                    ()).intValue();  
            }  
            catch (Exception e) {  
                System.out.println("Não é um inteiro  
                    válido!!!");  
            }  
        }  
    }  
    public static char umChar() {  
        while (true) {  
            try {  
                return umaString().charAt(0);  
            }  
            catch (Exception e) {  
                System.out.println("Não é um char válido");  
            }  
        }  
    }  
}
```

Como se pode verificar, o programa em Java é relativamente simples. Há uma classe Pessoa e a Classe Artigo e a Classe Coordenador facilmente herdam os atributos da sua super classe.

Um coordenador é uma pessoa, logo os dados do coordenador são guardados fazendo o uso dos atributos da classe Pessoa, adicionando-lhe ainda um numero de coordenador e a indica-

ção de pela qual coluna da revista é esse coordenador responsável. Por sua vez, um artigo tem sempre um autor que por sua vez é uma pessoa, logo Pessoa pode também ser usada como superclasse da Classe Artigo para poder guardar os dados do autor desse artigo. Um objeto do tipo Artigo tem também a indicação do nome do artigo, da coluna e da área científica à qual pertence. Foi criado nesta classe um atributo *flagPublicado* para facilmente sabermos se um artigo foi ou não publicado. Caso seja esteja publicado (já escolhido) esse campo tem o valor 1, caso o artigo não tenha sido escolhido (logo não publicado) então esse campo tem o valor 0.

Como os vetores em Java são acrescentados automaticamente à medida que o utilizador vai introduzindo valores é simples criar um novo objeto de classe quando isso é necessário. Para apresentar os artigos disponíveis para publicar (os que têm a *flagPublicado* a zero) e os que estão já publicados, basta percorrer o vetor de forma a encontrar quais os artigos estão ou não publicados recorrendo ao atributo *flagPublicado*.

No menu Fazer Edição, o facto de retirar ou não um artigo da publicação da revista apenas implica percorrer o vetor de artigos e alterar o valor *flagPublicado* conforme nos dá mais jeito.

### Passemos agora ao programa em Linguagem C:

#### Programa em Linguagem C

```
#include<stdio.h>  
#include<stdlib.h>  
#include<string.h>  
  
//estruturas  
typedef struct PESSOA{  
    char nome[50];  
    char email[50];  
    char localidade[50];  
    int telefone;  
    struct PESSOA *nseg;  
    struct PESSOA *nant;  
}  
}Pessoa;  
  
typedef struct COORDENADOR{  
  
    int numCoordenador;  
    char coluna[50];  
    Pessoa *p;  
    struct COORDENADOR *nseg;  
  
}  
}Coordenador;  
  
typedef struct ARTIGO{  
  
    int numArtigo;  
    char nomeArtigo[50];  
    char coluna[50];  
    char areaCientifica[50];  
    int flagPublicado;  
    char nome[50];  
    char email[50];  
    char localidade[50];  
}
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
int telefone;
    struct ARTIGO *nseg;
}Artigo;

Pessoa *makeNode();
Artigo *makeNodeArtigo();
Artigo *insereArtigoAutor();
Artigo* removeFirstArtigos(Artigo *A);
Artigo* clearListArtigos(Artigo *A);
Artigo* insertLast(Artigo *A, Artigo *nv);
Coordenador *makeNodeCoordenador();
Coordenador* removeFirstCoordenador(Coordenador
                                     *C);
Coordenador* clearListCoordenadores(Coordenador
                                     *C);
Coordenador* insertLastCoordenadores(Coordenador
                                     *C, Coordenador *nv);
Coordenador *insereCoordenador();
int mostraEdicao(Artigo *A);
int listarArtigosDisponiveis(Artigo *A);
int criaEdicao(Artigo *A, int numPublicados);
void mostraArtigoAutor(Artigo *a);
void listarArtigos(Artigo *A);
void consultaNumeroArtigo(Artigo *A);
void modificarArtigo(Artigo *A);
void consultaNumeroCoordenador(Coordenador *C);
void modificarCoordenador(Coordenador *C);
void listarCoordenadores(Coordenador *c);
void retirarArtigoEdicao(Artigo *A);
void mostraCoordenador(Coordenador *c);

Coordenador* removeFirstCoordenador(Coordenador *C)
{
    Coordenador *aux=C;
    if(C==NULL)
        return (NULL);
    C=C->nseg;
    free(aux);
    return (C);
}

Coordenador* clearListCoordenadores(Coordenador *C)
{
    while(C!=NULL){
        C=removeFirstCoordenador(C);
    }
    return (NULL);
}

Artigo* removeFirstArtigos(Artigo *A){
    Artigo *aux=A;
    if(A==NULL)
        return (NULL);
    A=A->nseg;
    free(aux);
    return (A);
}

Artigo* clearListArtigos(Artigo *A){
    while(A!=NULL){
        A=removeFirstArtigos(A);
    }
    return (NULL);
}

void listarCoordenadores(Coordenador *C){
    while(C!=NULL){
        mostraCoordenador(C);
        C=C->nseg;
    }
}
```

```
int criaEdicao(Artigo *A, int numPublicados){
    Artigo *aux=A;
    char lixo, leitura;
    while(A!=NULL){
        if(A->flagPublicado==0){
            mostraArtigoAutor(A);
            getchar();
            printf("\nDeseja inserir o artigo na
                    proxima edicao?");
            scanf("%c", &leitura);
            if(leitura=='s' || leitura=='S'){
                A->flagPublicado=1;
                numPublicados++;
            }
            A=A->nseg;
        }
    }
    return(numPublicados);
}

void retirarArtigoEdicao(Artigo *A){
    int retirar;
    printf("\nQual o numero do artigo a retirar?");
    scanf("%d", &retirar);
    while(A!=NULL){
        if(A->numArtigo==retirar){
            A->flagPublicado==0;
            break;
        }
        A=A->nseg;
    }
}

int listarArtigosDisponiveis(Artigo *A){
    int artigosDisponiveis=0;
    while(A!=NULL){
        if(A->flagPublicado==0){
            mostraArtigoAutor(A);
            artigosDisponiveis++;
        }
        A=A->nseg;
    }
    return(artigosDisponiveis);
}

int mostraEdicao(Artigo *A){
    int controle=0;
    while(A!=NULL){
        if(A->flagPublicado==1){
            mostraArtigoAutor(A);
            controle++;
        }
        A=A->nseg;
    }
    return(controle);
}

void modificarArtigo(Artigo *A){
    char lixo[50];
    int numArtigo;
    char nomeArtigo[50];
    char coluna[50];
    char areaCientifica[50];
    int i;
    char nome[50];
    char email[50];
}
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
char localidade[50];
int telefone;
int num;
char leitura;
printf("\nIntroduza o numero do Artigo a
      Modificar: ");
scanf("%d", &num);

while(A!=NULL){
    if(A->numArtigo==num){
        gets(lixo);
        printf("\nDeseja modificar o nome
              do Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\n Introduza o novo
                  nome do Artigo: ");
            gets(nomeArtigo);
            strcpy(A->nomeArtigo, nomeArtigo);

        }
        gets(lixo);
        printf("\nDeseja modificar a
              coluna Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\nIntroduza a nova
                  coluna do Artigo: ");
            gets(coluna);
            strcpy(A->coluna, coluna);
        }
        gets(lixo);
        printf("\nDeseja modificar a area
              cientifica do Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\nIntroduza a nova
                  area cientifica do Artigo: ");
            gets(areaCientifica);
            strcpy(A->areaCientifica,
                  areaCientifica);
        }
        getchar();
        printf("\nDeseja modificar o nome
              do autor do Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\nIntroduza o novo
                  nome do Autor do Artigo: ");
            gets(nome);
            strcpy(A->nome, nome);
        }
        getchar();
        printf("\nDeseja modificar o
              email do autor do Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\nIntroduza o novo
                  email do autor do Artigo: ");
            gets(email);
            strcpy(A->email, email);
        }
        getchar();
        printf("\nDeseja modificar a
              localidade do autor do Artigo: ");
        scanf("%c", &leitura);
        if(leitura=='s' || leitura=='S'){
            gets(lixo);
            printf("\nIntroduza a nova
```

```
localidade do autor do Artigo: ");
        gets(localidade);
        strcpy(A->localidade,
              localidade);

    }
    getchar();
    printf("\nDeseja modificar o
          telefone do autor do Artigo: ");
    scanf("%c", &leitura);
    if(leitura=='s' || leitura=='S'){
        gets(lixo);
        printf("\nIntroduza o novo
              telefone do autor do Artigo: ");
        scanf("%d", &telefone);
        A->telefone=telefone;
    }
    return;
}
A=A->nseg;
}
printf("\n Artigo nao encontrado.\n");
}

void consultaNumeroArtigo(Artigo *A){
    int num;
    printf("\nInsira o numero do Artigo: ");
    scanf("%d", &num);
    while(A!=NULL){
        if(A->numArtigo==num){
            mostraArtigoAutor(A);
            return;
        }
        A=A->nseg;
    }
    printf("\nArtigo nao encontrado.\n");
}

void modificarCoordenador(Coordenador *C){
    char lixo[50];
    char coluna[50];
    int i;
    char nome[50];
    char email[50];
    char localidade[50];
    int telefone;
    int num;
    char leitura;
    printf("\nIntroduza o numero do Coordenador
          a Modificar: ");
    scanf("%d", &num);
    while(C!=NULL){
        if(C->numCoordenador==num){
            gets(lixo);
            printf("\nDeseja modificar a
                  coluna do Coordenador? ");
            scanf("%c", &leitura);
            if(leitura=='s' || leitura=='S'){
                gets(lixo);
                printf("\nIntroduza a nova
                      coluna do Coordenador: ");
                gets(coluna);
                strcpy(C->coluna, coluna);
            }
            gets(lixo);
            printf("\nDeseja modificar o
                  nome do Coordenador? ");
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
scanf("%c", &leitura);
    if(leitura=='s' || leitura=='S'){
        gets(lixo);
        printf("\nIntroduza o novo
            nome do Coordenador: ");
        gets(nome);
        strcpy(C->p->nome, nome);
    }
    getchar();
    printf("\nDeseja modificar o
        email do Coordenador? ");
    scanf("%c", &leitura);
    if(leitura=='s' || leitura=='S'){
        gets(lixo);
        printf("\nIntroduza o novo
            email do Coordenador?");
        gets(email);
        strcpy(C->p->email, email);
    }
    getchar();
    printf("\nDeseja modificar a
        localidade do Coordenador? ");
    scanf("%c", &leitura);
    if(leitura=='s' || leitura=='S'){
        gets(lixo);
        printf("\nIntroduza a nova
            localidade do Coordenador: ");
        gets(localidade);
        strcpy(C->p->localidade,
            localidade);
    }
    getchar();
    printf("\nDeseja modificar o
        telefone do Coordenador? ");
    scanf("%c", &leitura);
    if(leitura=='s' || leitura=='S'){
        gets(lixo);
        printf("\nIntroduza o novo
            telefone do Coordenador: ");
        scanf("%d", &telefone);
        C->p->telefone=telefone;
    }

    return;
}
C=C->nseg;
}
printf("\nCoordenador nao encontrado.\n");
}

void consultaNumeroCoordenador(Coordenador *C){
    int num;
    printf("\nInsira o numero do Coordenador: ");
    scanf("%d", &num);
    while(C!=NULL){
        if(C->numCoordenador==num){
            mostraCoordenador(C);
            return;
        }
        C=C->nseg;
    }
    printf("\nCoordenador nao encontrado.\n");
}

void listarArtigos(Artigo *A){
    if(A==NULL)
        printf("\nNenhum Artigo para Listar\n");
    while(A!=NULL){
        mostraArtigoAutor(A);
```

```
        A=A->nseg;
    }
}

Artigo* insertLast(Artigo *A, Artigo *nv){
    Artigo *aux=A;
    if(A==NULL) return (nv);
    while(A!=NULL){
        if(A->nseg==NULL){
            A->nseg=nv;
            break;
        }
        A=A->nseg;
    }
    return(aux);
}

Coordenador* insertLastCoordenadores(Coordenador
*C, Coordenador *nv){
    Coordenador *aux=C;
    if(C==NULL) return (nv);
    while(C!=NULL){
        if(C->nseg==NULL){
            C->nseg=nv;
            break;
        }
        C=C->nseg;
    }
    return(aux);
}

Pessoa *makeNode(){
    Pessoa *p;
    p=(Pessoa *) malloc(sizeof(Pessoa));
    strcpy(p->nome, " ");
    strcpy(p->localidade, " ");
    strcpy(p->email, " ");
    p->telefone=0;
    p->nseg=NULL;

    return(p);
}

Coordenador *makeNodeCoordenador(){
    Coordenador *c;
    c=(Coordenador *) malloc(sizeof
(Coordenador));
    c->p=makeNode();
    c->numCoordenador=-1;
    strcpy(c->coluna, " ");
    return(c);
}

Artigo *makeNodeArtigo(){
    Artigo *a;
    a=(Artigo *) malloc(sizeof(Artigo));
    a->numArtigo=-1;
    strcpy(a->nomeArtigo, " ");
    strcpy(a->coluna, " ");
    strcpy(a->areaCientifica, " ");
    a->flagPublicado=0;
    strcpy(a->nome, " ");
    strcpy(a->localidade, " ");
    strcpy(a->email, " ");
    a->telefone=0;
    a->nseg=NULL;
    return(a);
}
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
Artigo *insereArtigoAutor(){
    char lixo[50];
    int numArtigo;
    char nomeArtigo[50];
    char coluna[50];
    char areaCientifica[50];
    int i;
    char nome[50];
    char email[50];
    char localidade[50];
    int telefone;
    Artigo *novo;
    gets(lixo);
    printf("\nInsira o nome do Artigo: ");
    gets(nomeArtigo);
    printf("\nInsira o nome da Coluna: ");
    gets(coluna);
    printf("\nInsira a area cientifica: ");
    gets(areaCientifica);
    printf("\nInsira o nome do Autor: ");
    gets(nome);
    printf("\nInsira o email do Autor: ");
    gets(email);
    printf("\nInsira a localidade do Autor: ");
    gets(localidade);
    printf("\nInsira o telefone do Autor: ");
    scanf("%d", &telefone);
    novo=makeNodeArtigo();
    strcpy(novo->nomeArtigo, nomeArtigo);
    strcpy(novo->coluna, coluna);
    strcpy(novo->areaCientifica, areaCientifica);
    strcpy(novo->nome, nome);
    strcpy(novo->email, email);
    strcpy(novo->localidade, localidade);
    novo->telefone=telefone;
    return(novo);
}
void mostraArtigoAutor(Artigo *a)
{
    printf("\nNumero Artigo: %d Nome Artigo: %s
Coluna: %s Area Cientifica: %s ", a->numArtigo,
a->nomeArtigo, a->coluna, a->areaCientifica);
    if(a->flagPublicado==0)
        printf("Estado: Nao Publicado");
    if(a->flagPublicado==1)
        printf("Estado: Publicado");
    printf("\nNome Autor: %s Localidade: %s
Email:%s Telefone: %d\n", a->nome,
a->localidade, a->email, a->telefone );
}
void mostraCoordenador(Coordenador *c){
    printf("\nNumero Coordenador: %d Coluna: %s ",
c->numCoordenador, c->coluna);
    printf("\nNome Autor: %s Localidade: %s
Email:%s Telefone: %d\n", c->p->nome,
c->p->localidade, c->p->email, c->p->telefone );
}
Coordenador *insereCoordenador(){
    char lixo[50];
    char coluna[50];
    char nome[50];
    char email[50];
    char localidade[50];
    int telefone;
    Coordenador *novo;
    gets(lixo);
    printf("\nInsira o nome da Coluna do
Coordenador: ");
    gets(coluna);
    printf("\nInsira o nome do Coordenador: ");
```

```
gets(nome);
printf("\nInsira o email do Coordenador: ");
gets(email);
printf("\nInsira a localidade do Coordenador: ");
gets(localidade);
printf("\nInsira o telefone do Coordenador: ");
scanf("%d", &telefone);

    novo=makeNodeCoordenador();

    strcpy(novo->coluna, coluna);
    strcpy(novo->p->nome, nome);
    strcpy(novo->p->email, email);
    strcpy(novo->p->localidade, localidade);
    novo->p->telefone=telefone;
    return(novo);
}
//programa principal
int main(){
    int i, opcao=-1, opcaoregisto=-1, opcaofuncionamen-
to=-1, op=-1, opetira=-1;
    Coordenador *C=NULL, *novoCoordenador=NULL;
    int numeroArtigos=0, nartigo, nEdicao,
        numPublicados=0, numeroCoordenadores=0;
    int artigosDisponiveis=0;
    Artigo *A=NULL, *novoArtigo=NULL;

    while(opcao!=0){

        printf("\n\n--- Revista PROGRAMAR --- \n");
        printf("\n1- Registos");
        printf("\n2- Fazer Edicao");
        printf("\n3- Ver Edicao");
        printf("\n0- sair");
        printf("\n\nEscolha a opcao:
(0 para terminar)");
        scanf("%d",&opcao);

        switch(opcao){
            case 1: //Registos
                printf("\n REGISTOS\n");
                do{
                    printf("\n----- MENU REGISTOS
-----\n");
                    printf("\n1 - Registrar Artigo e
Autor");
                    printf("\n2 - Registrar
Coordenador");
                    printf("\nEscolha uma opcao (0
para voltar ao menu anterior)\n");
                    scanf("%d", &opcaoregisto);
                    switch(opcaoregisto){
                        case 1: //artigo-autor
                            do{
                                printf("----- MENU REGISTOS
ARTIGOS E AUTORES -----");
                                printf("\n1 - Inserir artigo e
autor");
                                printf("\n2 - Listar artigos/
autores");
                                printf("\n3 - Consultar dados
artigo/autor (por numero)");
                                printf("\n4 - Modificar dados
artigo/autor (por numero)");
                                printf("\nEscolha uma opcao (0
para voltar ao menu anterior)");
                                scanf("%d", &op);
                                switch(op)
```

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

```
case 1:
    novoArtigo=inserereArtigoAutor();
    if(novoArtigo!=NULL){
        numeroArtigos++;
        novoArtigo->numArtigo=numeroArtigos;
        A=insertLast(A, novoArtigo);
    }
    break;
case 2:
    listarArtigos(A);
    break;
case 3:
    consultaNumeroArtigo(A);
    break;
case 4:
    printf("\nmodificar dados");
    modificarArtigo(A);
    break;
} //fim do switch do op
}while(op!=0);
break;

case 2: //coordenadores
do{
    printf("----- MENU REGISTOS COORDENADORES -----
        -");
    printf("\n1 - Inserir coordenador");
    printf("\n2 - Listar coordenadores");
    printf("\n3 - Consultar dados coordenador (por
        numero)");
    printf("\n4 - Modificar dados coordenador (por
        numero)");

    printf("\nEscolha uma opcao (0 para voltar ao
        menu anterior)");
    scanf("%d", &op);
switch(op){
    case 1:
        novoCoordenador=inserereCoordenador();
        if(novoCoordenador!=NULL){
            numeroCoordenadores++;
            novoCoordenador->
            numCoordenador=numeroCoordenadores;
            mostraCoordenador(novoCoordenador);
            C=insertLastCoordenadores(C,
                novoCoordenador);
        }
        break;
    case 2:
        listarCoordenadores(C);
        break;
    case 3: consultaNumeroCoordenador(C);
        break;
    case 4: modificarCoordenador(C);
        break;
} //fim do switch do op
}while(op!=0);
break;
} //fim do switch opcaoregisto
}while(opcaoregisto!=0);
break; //break do case 1 do switch opcao

case 2: //Fazer Edicao
    printf("\nFazer edicao\n");
do{
```

```
printf("\n ----- MENU FAZER EDICAO ----- \n");
printf("\n1 - Listar artigos disponiveis");
printf("\n2 - Adicionar artigos a edicao");
printf("\n3 - Retirar artigos da edicao");
printf("\nEscolha uma opcao (0 para voltar ao
    menu anterior)");
    scanf("%d", &opcaofuncionamento);
switch(opcaofuncionamento){
    case 1:
        artigosDisponiveis=listarArtigosDisponiveis(A);
        if(artigosDisponiveis==0)
            printf("\nNao ha artigos disponiveis para
                publicar");
        break;
    case 2:
        numPublicados=criaEdicao(A, numPublicados);
        break;
    case 3:
        retirarArtigoEdicao(A);
        break;
} //fim do switch de opcaofuncionamento
}while(opcaofuncionamento!=0);

break; //break do case 2 do switch opcao

case 3: //ver edicao
    nEdicao=mostraEdicao(A);
    if(nEdicao==0)
        printf("\nAinda nao ha artigos adicionados
            a proxima edicao\n");
    break; //break do case 3 do switch opcao
} //fim switch opcao
} //fim while do menu
A=clearListArtigos(A);
C=clearListCoordenadores(C);
return 0;
}
```

Nesta versão do programa, como o leitor pode ver as funcionalidades foram todas mantidas em relação à versão Java.

Como os vetores em C não “crescem” tão simplesmente como em Java, nesta versão foram utilizadas listas.

Foram criadas três estruturas (Pessoa, Coordenador e Artigo). Para exemplificar mais facilmente as diferenças, a estrutura Artigo foi criada com os atributos todos de forma simples. Por outro lado, a estrutura Coordenador tem “dentro” de si uma estrutura do tipo Pessoa (simulando a herança da Linguagem Java). No caso do exemplo, isto foi possível recorrendo aos famosos apontadores da Linguagem C. T

al como uma classe não é a mesma coisa que um objeto, visto que uma classe define um tipo de objeto e um objeto é uma instancia da sua classe, aqui também a declaração de uma estrutura não cria ainda um novo tipo de dado. Para isso, temos que alocar a memória necessária sempre que precisamos.

Por outras palavras, quando o leitor escolhe inserir os dados de um novo artigo ou coordenador, são guardados os dados introduzidos, alocada a memória para um novo elemento do tipo da estrutura pretendida e copiados os dados introduzidos para a nova estrutura e essa estrutura depois é colocada no final da lista de artigos ou coordenadores já existentes.

# TEMA DA CAPA

## ESTRUTURAS VS. OBJETOS

Tudo isto é feito recorrendo aos apontadores. Em C é comum usar um apontador para guardar o endereço de um dado elemento. E este é um dos pontos fortes desta linguagem, a sua flexibilidade.

Pegando num caso concreto dos programas deste artigo, vamos supor que queremos alterar o email de um dos nossos coordenadores previamente introduzidos:

Em Java usamos a instrução:

```
System.out.println("Insira novo email do autor: ");
email=Leitura.umaString();
aux.setEmail(email);
```

e em C usamos:

```
printf("\nIntroduza o novo email do Coordenador?");
gets(email);
strcpy(C->p->email, email);
```

“ O objetivo da equipa que criou a Linguagem Java era criar uma linguagem que permitisse uma comunicação fácil entre vários dispositivos electrónicos. ”

Mesmo para os adeptos mais fervorosos do C, é fácil admitir que em java é muito mais fácil e intuitivo fazer isso, contudo para os defensores do Java também é simples admitir que para aceder ao mesmo campo em C, basta usar o apontador para a estrutura coordenador e encaminha-lo para o apontador do campo email da estrutura pessoa que está dentro da do Coordenador. Ou seja, usar o apontador do apontador. São apenas

formas diferentes de fazer a mesma coisa.

Ora então chegados à reta final do artigo desta edição o que podemos concluir? Que ambas as linguagens são ótimas linguagens e podem ser usadas nas mais diversas situações. Se o Java nos traz vantagem pela sua elevada portabilidade (o que em tempos de explosão das aplicações na internet é bastante favorável), o C oferece acesso a recursos, por exemplo, de hardware, que o Java não oferece.

“ Se o Java nos traz vantagem pela sua elevada portabilidade (...), o C oferece acesso a recursos, por exemplo, de hardware, que o Java não oferece. ”

Recordando ainda os programas deste artigo, o utilizador apenas estava interessado nas funcionalidades que estão no programa e se o mesmo funciona bem ou não. E a satisfação do nosso utilizador ainda é independente da linguagem usada. Neste caso ambas as linguagens fizeram correctamente o seu trabalho, com mais ou menos esforço pela parte do programador.

Este artigo é o segundo de uma série de artigos de programação tendo como base principal a Linguagem C e a Linguagem Java como base auxiliar, que esperamos que siga atentamente.

## AUTOR



Escrito por Rita Peres

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

# A PROGRAMAR

**Lua – Linguagem de Programação – Parte 11**

**Introdução à Windows Workflow Foundation**

**P vs NP Compreender o maior problema por resolver em computação**

## Lua – Linguagem de Programação – Parte 11

Como informado no artigo número 10 desta série publicado na 30ª edição, esporadicamente seriam apresentados outros artigos complementares com usos da linguagem Lua. Cumprindo o prometido neste artigo, apresenta-se o novo comando **goto** que fora acrescido a linguagem Lua a partir da versão **5.2.0** lançada em 16 de dezembro de 2011 e o uso da linguagem Lua integrada com o programa **gnuplot**.

### COMANDO GOTO

A versão **5.2.0** da linguagem Lua vem com novidades, destacando-se o comando **goto**, que será neste artigo abordado. Outros recursos serão tratados oportunamente.

O comando **goto** é fonte de muita discussão e controvérsias entre programadores, pois alguns (na sua maioria) são contra seu uso, mas a aqueles que defendem sua aplicação.

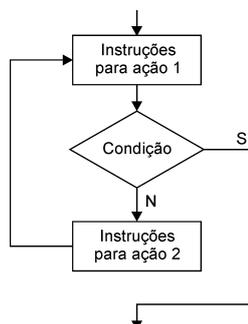
Não é objetivo polemizar tal discussão, mas de posicionar uma luz sobre este fato.

Aqueles que são contra o uso de **goto** em linguagens de programação alegam que seu uso pode desestruturar um programa. O que de fato é verdade. No entanto, o comando **goto** existente em diversas linguagens de programação estruturadas e em linguagens orientadas ao objeto não existe por acaso. Seu uso pode ser útil em algumas situações como, por exemplo, a possibilidade de se fazer o encerramento da execução de uma estrutura condicional ou de laço sem passar por todos os aninhamentos de finalização ou ainda para simular a execução da estrutura de laço de repetição **loop ... exit ... end**, que será aqui abordado em Lua.

O laço **loop ... exit ... end** é uma estrutura que permite a definição e colocação da condição de encerramento do laço em qualquer parte do bloco de execução. Este tipo de laço é encontrado nas linguagens de programação BASIC e ADA, mas inexistente em outras linguagens.

Uma forma de fazer uso do laço **loop ... exit ... end** nas linguagens de programação que não o possuem é por meio do comando **goto**.

O laço **loop ... exit ... end** (laço condicional seletivo) tem como estrutura gráfica, o formato:



O laço de repetição condicional seletivo, do ponto de vista sintático, pode ser escrito:

```
loop
[instruções para ação 1]
  exit (<condição>)
[instruções para ação 2]
end
```

A partir do exposto pode-se simular a ação de um laço condicional seletivo com **goto**. O exemplo seguinte mostra como efetuar o uso do comando **goto** definido na linguagem Lua a partir da versão **5.2.0**. Assim, escreva em um editor de texto o código a seguir, gravando-o com o nome **vaipara.lua**.

```
-- inicio do programa VAIPARA

I = 1
::PontoRetorno::
print(string.format("%2d", I))
if (I < 10) then
  I = I + 1
  goto PontoRetorno
end

-- fim do programa VAIPARA
```

Note ao executar o programa que enquanto o contador da variável **I** for menor que 10 será apresentado seu valor. No momento em que a condição se tornar verdadeira o laço será executado. Perceba que a condição pode ser posicionada em qualquer parte do bloco que se deseja executar simulando desta maneira um laço condicional seletivo. Eis aqui uma aplicação adequada para o comando **goto**.

### LUA COM GNUPLLOT

O programa **gnuplot** é uma ferramenta que possibilita o desenvolvimento de gráficos a partir do uso de equações matemáticas ou de arquivos de dados externos que estejam gravados em modo texto.

Para o teste deste artigo é fundamental que se possua instalado no computador a ser usado, além da linguagem Lua o programa **gnuplot** que pode ser obtido a partir da URL: <http://gnuplot.sourceforge.net/>.

Não é objetivo deste artigo “ensinar” o uso do programa **gnuplot**, mas sim demonstrar seu uso com a linguagem Lua.

O programa Lua a seguir tem por finalidade efetuar a criação de um arquivo de dados que será utilizado para a criação de gráficos no programa **gnuplot**.

Assim, escreva em um editor de texto o código a seguir, gravando-o com o nome **gerarq.lua**.

```
-- inicio do programa GERARQ

ARQ = io.open("dadoslua.dat", "w")
ARQ:write("#Tempo\tValor\n")
```

```
VP = 100
I = 1
ARQ:write("0\t" .. VP .. "\n")
for N = 1, 10, 1 do
  VF = VP * (1 + I/100) ^ N
  ARQ:write(N .. "\t" .. VF .. "\n")
End

ARQ:close()

-- fim do programa GERARQ
```

O programa anterior criar um arquivo do tipo texto chamado **dadoslua.dat** contendo os valores de uma aplicação financeira mensal com investimento inicial de **\$100** a uma taxa de juros mensal de **1%** durante **10** meses a partir da fórmula de matemática financeira para o cálculo do valor futuro.

O caractere de controle **\t** está sendo usado para separar os grupos de valores com um caractere **Tab** usado pelo programa **gnuplot** como separador de dados do arquivo.

A partir da criação do arquivo de dados o programa a seguir fará a criação de um arquivo de *script* para o programa **gnuplot** contendo as instruções para o programa e fará a chamada deste *script* apresentando o gráfico.

Assim, escreva em um editor de texto o código a seguir, gra-

```
-- inicio do programa MOSTRAGRAF

ARQ = io.open("script.plt", "w")
ARQ:write("reset\n")
ARQ:write("set grid\n")
ARQ:write("set logscale x\n")
ARQ:write("plot \"dadoslua.dat\"")
ARQ:write(" with lines")
ARQ:write("\n")
ARQ:write("pause -1\n")
ARQ:close()
os.execute("gnuplot script.plt")

-- fim do programa MOSTRAGRAF
```

vando-o com o nome **mostragraf.lua**.

O programa anterior cria o arquivo de *script* **script.plt** com as instruções para a definição de um gráfico via programa **gnuplot**.

O comando **reset** faz a limpeza do ambiente a ser usado para a apresentação do gráfico.

O comando **set grid** define a apresentação de linhas na grade do gráfico.

O comando **set logscale x** estabelece para o eixo das abscis-

as uma escala logarítmica de modo que o gráfico possa mostrar a evolução do investimento proporcionalmente ao seu crescimento.

O comando **plot** com seu complemento **with lines** faz a apresentação de um gráfico de linha a partir do arquivo de dados indicado.

O comando **pause -1** faz com que a tela gráfica seja apresentada e mantida até ser fechada.

A instrução Lua com a função **os.execute()** faz a chamada do programa **gnuplot** passando como parâmetro o nome do arquivo de *script* a ser executado.

“ O comando **goto** é fonte de muita discussão e controvérsias entre programadores, pois alguns (na sua maioria) são contra seu uso(...) ”

### Conclusão

Neste artigo foi apresentado o uso do comando **goto** inserido na linguagem Lua a partir de uma postura que justifica sua existência a despeito das críticas que este tipo de comando recebe de diversos profissionais da área de desenvolvimento de *software*.

Outro ponto apresentado foi a integração da linguagem Lua com o programa **gnuplot**, que pode ser facilmente utilizado em conjunto com outras linguagens de programação.

Em outras ocasiões serão apresentados outras possibilidades de uso da linguagem Lua. Até lá.

## AUTOR



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

# A PROGRAMAR

## Introdução à Windows Workflow Foundation

Quando o .NET 3.0 foi lançado, a Microsoft lançou uma *Application Programming Interface* (API) denominada por “Windows Workflow Foundation”, que permitia a modelação, configuração, monitorização e execução de *Workflows* através de um *Designer Incorporado*.

Nessa altura, o ambiente fornecido pelo Visual Studio para trabalhar com esta API era bastante complexo e rudimentar, no entanto com o lançamento da versão 4.0 do Framework .NET e mais recentemente a 4.5, foram introduzidas diversas melhorias, como o facto de este ter sido totalmente reescrito utilizando a *Windows Presentation Foundation* e a sua linguagem XAML, o que melhorou significativamente a experiência do programador.

Este artigo tem como objetivo fazer uma pequena introdução ao potencial da *Windows Workflow Framework*, mostrando como poderá utilizar o Designer para desenhar alguns Workflows.

### O que se pode fazer com Workflows?

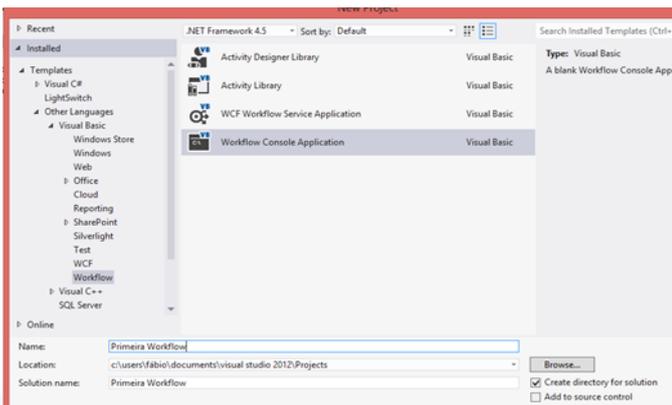
A principal vantagem que pode ser obtida através da utilização de Workflows é a simplificação da implementação de processos nas nossas aplicações, fornecendo para isso um Designer bastante intuitivo para o desenho dos mesmos.

### O Primeiro Workflow

Existem várias diferenças entre o desenvolvimento de um Workflow e o de uma aplicação normal .NET, sendo um dos principais o facto de ser utilizada uma linguagem declarativa no primeiro. Outra das diferenças é o ambiente baseado em gráficos utilizados no desenvolvimento de um Workflow. Para compreender melhor o seu funcionamento, vamos criar um Workflow inicial.

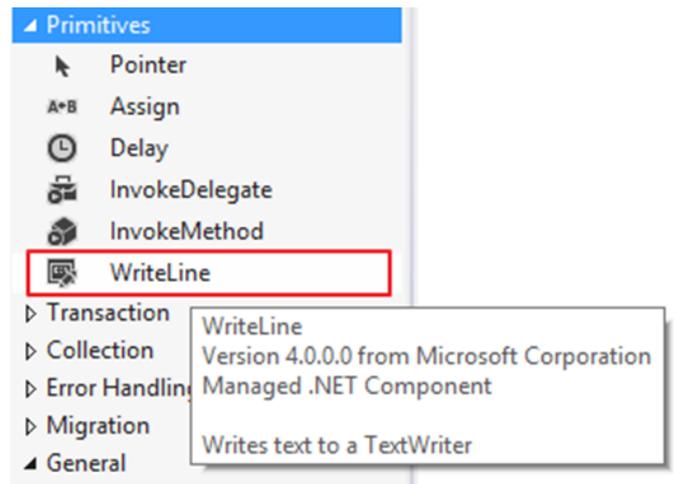
Neste primeiro Workflow vamos escrever na consola a já tradicional mensagem “Olá Mundo!”.

Para isso, vamos então criar um novo projeto, seleccionando-o nos Templates “Workflow” e no projeto “Workflow Console Application”.

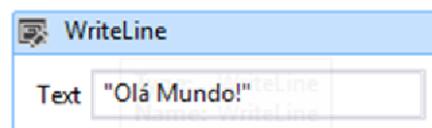


Será então apresentado o ambiente de criação de Workflows.

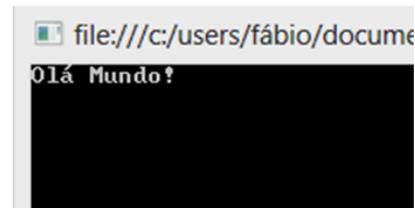
Como o que queremos é escrever uma mensagem na consola, devemos escolher uma instrução de escrita, neste caso a “WriteLine” que se encontra na secção “Primitives”:



Arrastamo-lo para a área “Drop activity here” do Designer e escrevemos a mensagem Olá Mundo!



Ao executar o Workflow, será então mostrada a seguinte mensagem:



No entanto, esta mensagem “aparece e desaparece”, não permitindo a leitura da mesma. Para resolvermos este problema, temos de adicionar um pouco de código, como iremos ver no ponto seguinte.

### Por detrás do Designer

Por forma a suportar as Workflows elaboradas através do Designer, como a que desenhámos anteriormente, é utilizada a linguagem “XAML”. O código gerado pode ser visto no

# A PROGRAMAR

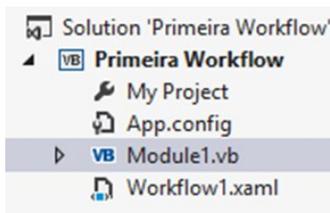
## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

código do ficheiro do Workflow, neste caso o “workflow1.xaml”, um excerto deste pode ser visto na figura seguinte:

```
</TextExpression.ReferencesForImplementation>
  <WriteLine Text="Olá Mundo!"
    sap2010:WorkflowViewState.IdRef="WriteLine_1"
    sads:DebugSymbol.Symbol="d3xj01x1c2Vyc1xmW6FiaW9cZG
9jdW11bnRzXHZpc3VhbCBzdHVkaW8gMjAxM1xQcm9qZWNoeXZp
3JrZmxvd0NvbnNvbGVbcHBsaWNhdG1vbG1vbnV29ya2Zsb3dDb25z
b2x1QXBwbGljYXRpb24zXFdvcmtmbG93MS54YW1sAhwDHIMBAGe
BHBMcHwIBAg==" />
  <sap2010:WorkflowViewState.IdRef>Workflow1_1
  </sap2010:WorkflowViewState.IdRef>
  <sap2010:WorkflowViewState.ViewStateManager>
```

Visualizando então este pedaço de código podemos então verificar na segunda linha o local onde é declarada a WriteLine cuja atividade colocámos anteriormente e o texto “Olá Mundo!”.

No entanto impõe-se mais uma questão: onde é que o Workflow é invocado, onde podemos controlar a sua execução? A resposta a esta questão reside no ficheiro “Module1.vb” que pode ser acessado através do “Solution Explorer”:



Abrindo-o, verificamos o seguinte código:

```
Imports System.Activities
Imports System.Activities.Statements
Imports System.Diagnostics
Imports System.Linq

Module Module1
  Dim s As Sequence
  Sub Main()
    Dim workflow1 = New Workflow1()
    WorkflowInvoker.Invoke(workflow1)
  End Sub
End Module
```

Ou seja, através deste módulo, é invocado o workflow através do WorkflowInvoker que faz parte do namespace “System.Activities” e que permite a execução do Workflow utilizando apenas uma linha de código.

É neste método que podemos também adicionar código que deve ser executado a seguir à execução do Workflow. Assim, se quisermos resolver o problema da janela “aparecer e desaparecer”, mostrado no exemplo anterior, basta a seguir à invocação adicionarmos um Console.ReadLine():

```
Sub Main()
  Dim workflow1 = New Workflow1()
  WorkflowInvoker.Invoke(workflow1)
  Console.ReadLine()
End Sub
```

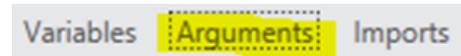
Deste modo, a consola ficará a aguardar que seja premeida uma tecla.

### Utilização de Argumentos

É possível enviar argumentos para o Workflow. Apesar de normalmente, quando trabalhamos com classes utilizarmos um construtor para passarmos esses argumentos, nos Workflows isso não é possível, pois estes são sempre criados utilizando o construtor por defeito.

Assim, para passar argumentos para o Workflow utiliza-se um Dictionary(Of String, Object), sendo para isso necessário definir primeiro os mesmos através do Designer. O procedimento é bastante simples e para o demonstrar iremos aproveitar o exemplo anterior, onde em vez de “Olá Mundo!”, vamos escrever o nome do utilizador:

Para definir os argumentos clicamos no botão “Arguments” que se encontra no rodapé do designer:

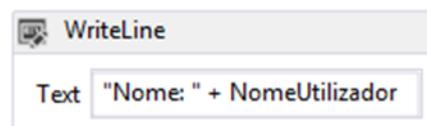


Surgirá então uma área onde podemos definir os mesmos, neste caso em **Name** atribuímos um nome ao mesmo, neste caso nomeUtilizador. Em **Direction** deixamos o **In**, pois este vai ser enviado para dentro do Workflow, em **Type** fica **String**, dado ser um nome e deixamos também em branco o Default Value, tal como podemos ver na figura seguinte:

Name	Direction	Argument type	Default value
NomeUtilizador	In	String	Enter a VB expression

Create Argument

Agora, na atividade WriteLine que criámos anteriormente e escrevemos “Olá Mundo!”, teremos de fazer referência ao argumento a escrever como se de um Console.WriteLine se tratasse. Assim devemos preencher a mesma da seguinte forma:



Desta forma, iremos escrever na consola, a frase Nome: seguida do nome do utilizador.

Agora vamos até ao **Module1.vb** e tratar de pedir o nome ao utilizador e enviá-lo para o Workflow:

```
Sub Main()
  'Pede e atribui a uma variável o nome do
  'utilizador
  Console.WriteLine("Qual o seu nome")
  Dim nome = Console.ReadLine

  'Colocamo-lo num dicionário por forma a
  'enviá-lo
  'para o Workflow
  Dim argumentoWF As New Dictionary(Of
  String, Object)()
  argumentoWF.Add("NomeUtilizador", nome)

  'Enviamo-lo para o Workflow
```

# A PROGRAMAR

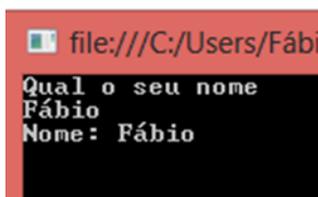
## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

```
Dim workflow1 = New Workflow1()  
WorkflowInvoker.Invoke(workflow1, argumentoWF)  
Console.ReadLine()  
End Sub
```

As primeiras duas linhas são bastante simples, no entanto chamo a atenção para a parte do dicionário. O seu tipo (Of String) deve sempre coincidir com o Argument Type definido no atributo criado. De seguida quando o adicionamos ao dicionário, a sintaxe a utilizar deve ser o nome do atributo tal como foi definido no Designer em Name dentro de aspas, pois o valor é Of String e a chave correspondente à variável a enviar.

Tal como podemos ver aquando a sua invocação, enviamos então o dicionário como argumento.

O resultado será o seguinte:

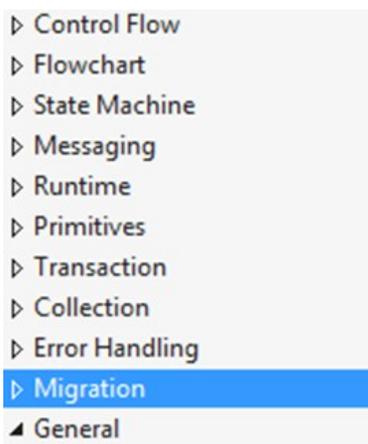


```
file:///C:/Users/Fábio/  
Qual o seu nome  
Fábio  
Nome: Fábio
```

O Workflow escreveu na consola o argumento enviado.

### As Atividades

Existem várias atividades enquadradas em várias categorias que podem ser utilizadas para construir os Workflows, tal como podemos verificar na Toolbox, sendo que até agora só usámos a WriteLine.



De uma forma breve, irei abordar algumas destas atividades e apresentarei também exemplos da utilização de algumas das suas atividades.

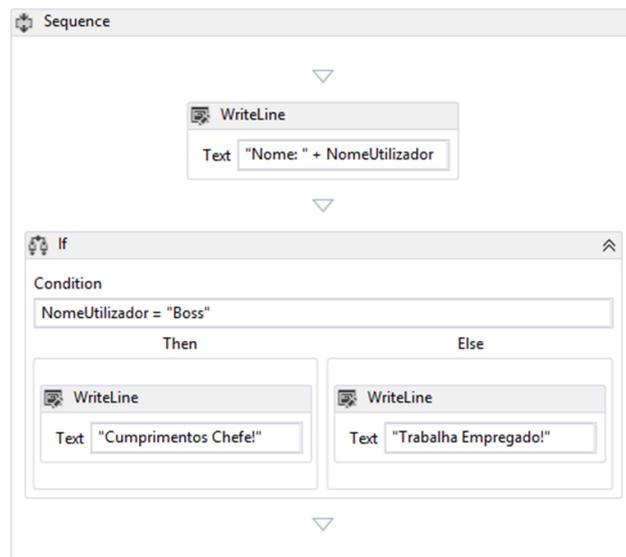
### Control Flow

Possui essencialmente atividades de decisão e repetição (loopings), como o If.

Existem também atividades especialmente criadas para definir o tipo de processamento desejado (Paralelo ou Sequencial).

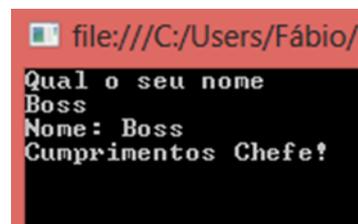
### Condição If

Exemplo de utilização de uma atividade de Control Flow (If):



Neste caso, será comparado a variável nome utilizador a "Boss", escrevendo mensagens diferentes dependendo se a condição for verdadeira ou falsa. Para isso serão escritos valores na consola através das atividades WriteLine usadas.

Um exemplo de resultado é o seguinte:



```
file:///C:/Users/Fábio/  
Qual o seu nome  
Boss  
Nome: Boss  
Cumprimentos Chefe!
```

### FlowChart

Nesta categoria encontram-se os elementos que são usados em FlowCharts, conhecidos em Portugal por fluxogramas.

Este conjunto de atividades permite utilizar este paradigma. Quando quisermos modelar com fluxogramas, temos de arrastar uma atividade Workflow para o Designer.

Exemplos de atividades desta categoria são a **FlowDecision** que fornece os tradicionais diamantes de decisão e **FlowSwitch** que permite modelar uma estrutura switch case.

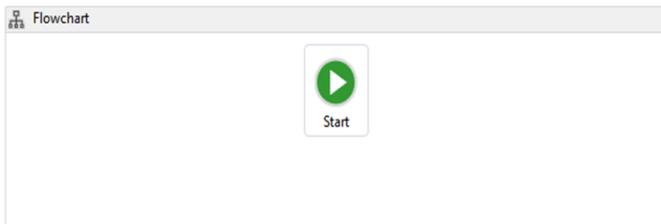
Vamos agora modelar o exemplo anterior utilizado para a "Condição If", elementos de FlowChart, para isso elimine os elementos do Designer ou crie um Workflow novo, certificando-se neste caso que declara os argumentos tal como vimos anteriormente e copia o código que criámos para o módulo para que este seja pedido e enviado.

Primeiro, adicionamos uma atividade "FlowChart" e note-se

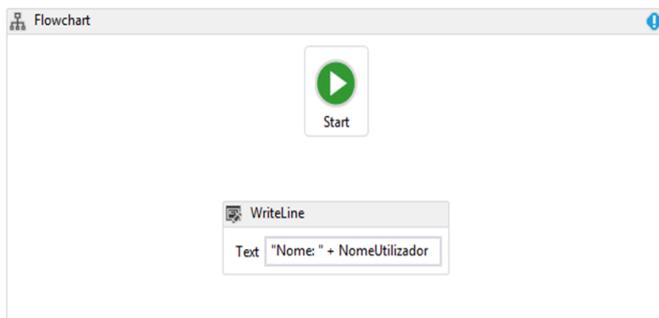
# A PROGRAMAR

## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

que temos agora uma atividade “Start” que é a atividade inicial dos fluxogramas:

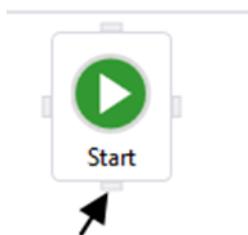


De seguida, arrastamos um WriteLine da categoria “Primitives” e escrevemos “Nome: ” + NomeUtilizador:

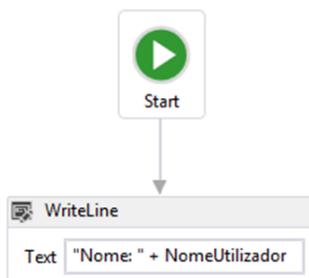


Como já deve ter reparado, desta vez e ao contrário dos exemplos anteriores que trabalhávamos apenas com sequências, não foi efetuada qualquer ligação entre as atividades.

Num Flowchart, a ligação tem de ser feita manualmente, dando um clique num dos pequenos retângulos que surgem por cima da atividade quando passamos com o rato sobre ela e arrastando para a atividade seguinte:

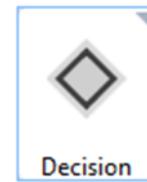


Desta forma, devemos ligar a atividade Start à WriteLine:

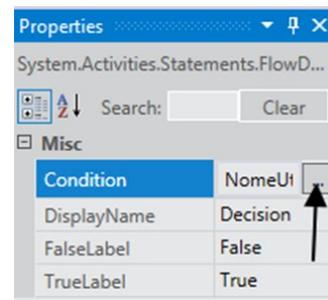


As atividades estão agora ligadas.

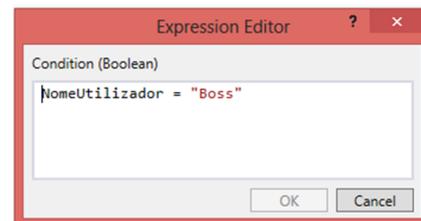
Arraste para o Designer uma FlowDecision (diamante de decisão):



Clique no diamante e de seguida nas Properties deste, defina a condição:



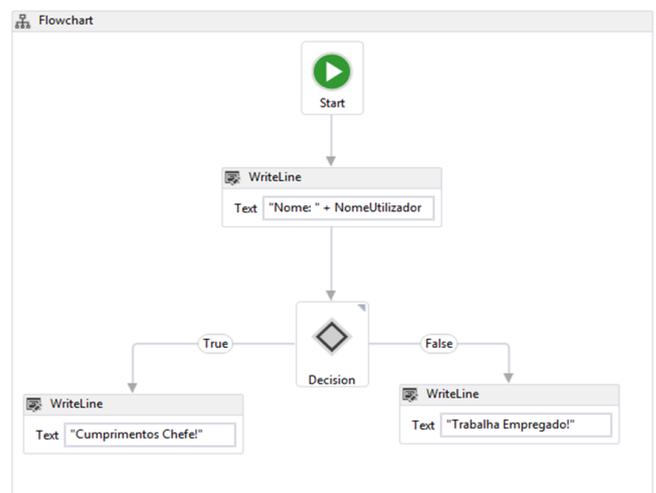
Neste caso deverá ser:



Ou seja, vamos comparar o valor da variável “NomeUtilizador” a Boss.

Crie mais dois WriteLine's um dizendo “Trabalha Empregado” e outro “Cumprimentos Chefe!”. Deverá colocar o primeiro do lado direito do diamante, pois se colocar o cursor em cima deste, verá que é aí onde está o lado do false, e o segundo do lado esquerdo que é o lado do true.

Ligue todas as atividades da seguinte forma:





portugal-a-programar  
org

# PROGRAMAR 2013 LISBOA

---

**O PRIMEIRO EVENTO PRESENCIAL DA COMUNIDADE  
PORTUGAL-A-PROGRAMAR**

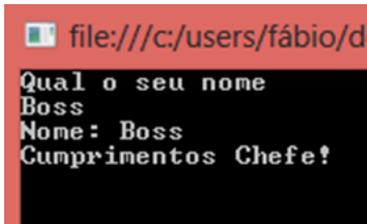
---

**25 DE MAIO DE 2013**  
**NO EDIFÍCIO DA MICROSOFT PORTUGAL**  
**PARQUE DA NAÇÕES, LISBOA - TODO O DIA**

# A PROGRAMAR

## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

O resultado será o mesmo:



```
file:///c:/users/fábio/d
Qual o seu nome
Boss
Nome: Boss
Cumprimentos Chefe!
```

Como estamos a ver, através de estereótipos bastante conhecidos dos fluxogramas, conseguimos fazer a modelação de Workflows.

### Messaging

Nesta categoria, existem atividades que permitem a invocar membros de XML Web Services ou serviços da Windows Communication Foundation.

### State Machine

Um sistema é composto por vários estados. Uma porta por exemplo, pode ter 2 estados, aberta ou fecha; uma máquina de venda de bebidas, pode ter como estados “Aguardando seleção do produto por parte do utilizador”, “Dispensando o produto” e “Devolvendo o troco”.

Esta categoria possui atividades para lidar com estes sistemas.

### Runtime e Primitives

Encontram-se nesta categoria as atividades que nos permitem efetuar tarefas básicas com o Workflow.

Uma das atividades desta categoria e que já utilizámos várias vezes é a WriteLine que nos permite escrever texto no ecrã, mas existem também outras que permitem por exemplo a invocação de métodos sendo por isso bastante importante e das mais usadas, bem como as de controlo do Runtime como a Persist e a TerminateWorkflow, sendo esta última utilizada para terminar o WorkFlow definindo uma exceção a ser lançada aquando o fim.

### Utilização de variáveis

Anteriormente vimos os argumentos que são enviados para dentro do WorkFlow, no entanto por vezes temos necessidade de utilizar variáveis, que são definidas e trabalhadas dentro do próprio WorkFlow.

Exemplo disso é quando ao invés de pedirmos os dados ao utilizador dentro do módulo, pedimos através do Workflow, utilizando para além de uma atividade WriteLine que formule a pergunta, uma atividade bastante importante que é o InvokeMethod, que nos permite invocar métodos a partir do Workflow.

De seguida apresentarei para exemplificar a utilização de variáveis, bem como outras atividades importantes, irei de seguida apresentar passo a passo o desenvolvimento de mais um Workflow, desta vez com o objetivo de mostrar ao utilizador os

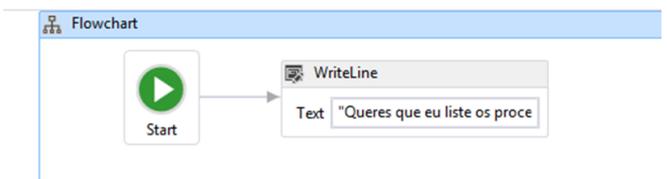
processos que estão a correr no sistema após perguntar ao utilizador.

### Ver a lista de processos que estão a correr

Neste Workflow, vamos perguntar ao utilizador se quer uma listagem de processos e de seguida, caso a resposta seja positiva vamos invocar os métodos necessários por forma a criar um array de processos e depois com um ciclo ForEach, percorrê-lo e mostrar o nome de cada um deles ao utilizador.

Primeiro, crie uma nova “workflow console application” e dê como nome “Listar Processos”.

Arraste para a área do designer um flowchart e uma writeline, sendo que esta última deverá escrever no ecrã a mensagem “Queres que eu liste os processos?” e ligue-os:



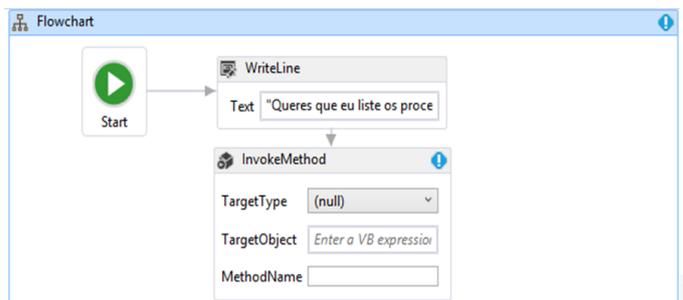
Agora, temos de recolher a resposta à pergunta por parte do utilizador e guardar a mesma numa variável.

Assim, devemos clicar na área do FlowChart para definir que esse deverá ser o âmbito da mesma, clicar em “Variables” na parte de baixo do Designer e criar uma nova variável do tipo String com o nome “SimOuNao”, que irá guardar a resposta do utilizador:

Name	Variable type	Scope	Default
SimOuNao	String	Flowchart	Enter a VB expression
Create Variable			

Agora, se fosse com código “Normal”, teríamos de fazer um Console.ReadLine, ou de uma forma mais completa, um System.Console.ReadLine() para que a resposta seja lida, nos workflows o método a invocar é o mesmo, mas como estamos a trabalhar graficamente vamos utilizar uma atividade especifica para esse efeito.

Arraste para o designer uma atividade “InvokeMethod” que se encontra na categoria “Primitives” e ligue à WriteLine, irá dar um aviso de erro, mas vamos tratar disso de seguida:

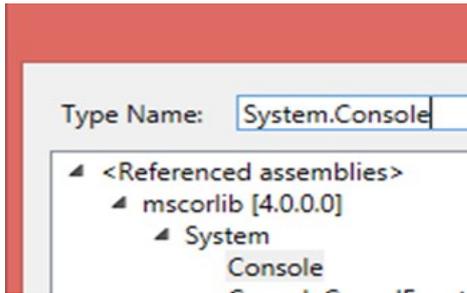


# A PROGRAMAR

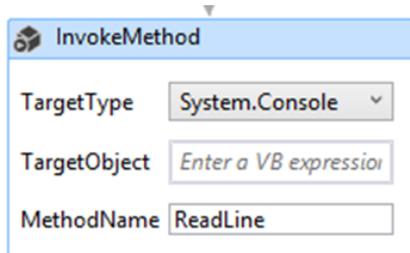
## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

Olhando para a atividade "InvokeMethod", é possível ver que esta possui três campos, o primeiro é o **TargetType**, onde é definido o tipo do método ou namespace do mesmo a invocar e em **MethodName** especifica-se qual o método que deverá ser invocado.

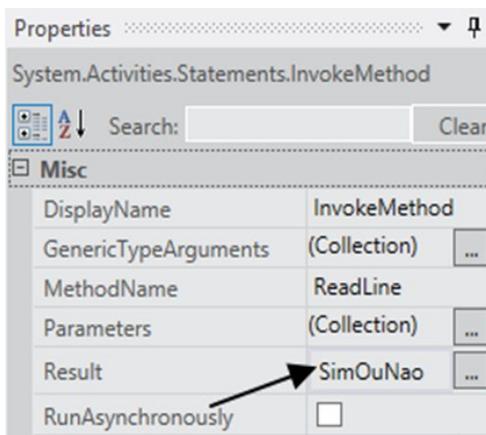
Clique na DropDown do **TargetType** e selecione a opção "Browser for Types", na caixa que surge, escreva "System.Console" e clique no botão "OK":



Em **MethodName**, escreva "ReadLine", pois o que pretendemos é ler o valor introduzido pelo utilizador, o aspeto final será o seguinte:



Agora não nos podemos esquecer de um pormenor, já invocamos o ReadLine, mas não especificámos qual o nome da variável onde queremos guardar o valor, assim deverá ir à caixa de propriedades da atividade "InvokeMethod" e em "Result" especificar o nome da variável, que no nosso caso é "SimOuNao":

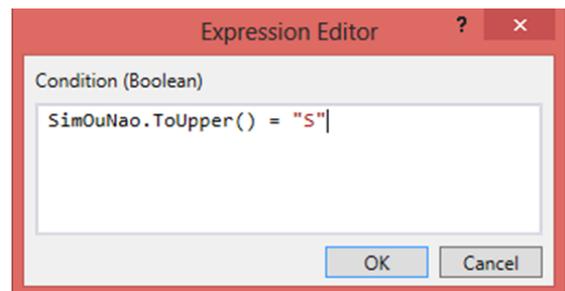


Agora que já temos a resposta guardada numa variável, temos de verificar logicamente se o utilizador quer ou não que os processos sejam listados, por isso deverá então arrastar para o

designer e ligar ao InvokeMethod um diamante de decisão ou seja uma atividade FlowDecision:

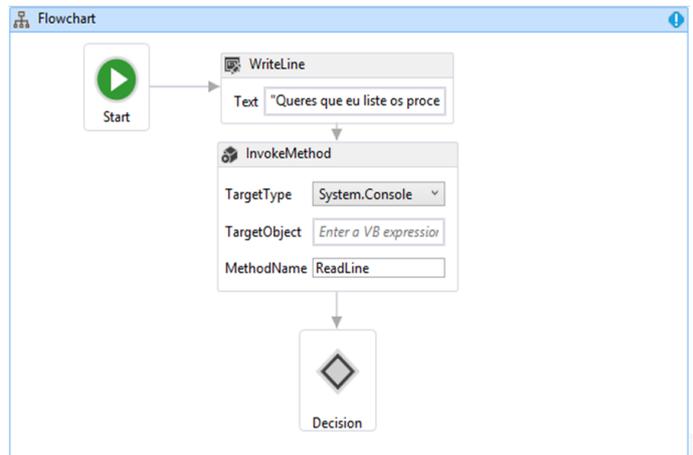


Assim, vamos definir que caso o utilizador introduza "S" como resposta, a listagem será feita, assim clique no diamante e de seguida nas propriedades vá a Condition tal como fez no exemplo anterior e escreva:



O motivo por ter inserido um ToUpper, é para que caso o utilizador insira um s minúsculo, a decisão tenha o resultado True que é o que queremos.

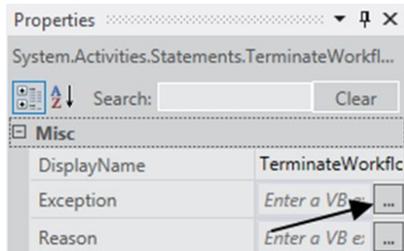
Continuando, arraste um WriteLine para a direita do diamante de decisão, ou seja para o lado do False e ligue-o a esse lado, escreva como texto "OK, fica para a próxima". Depois e para que o WorkFlow seja terminado, vá à categoria RunTime e arraste para o designer uma atividade "TerminateWorkflow", o aspeto será parecido com o seguinte:



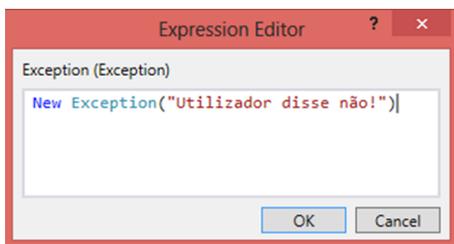
# A PROGRAMAR

## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

Se repararmos nos !, existe um problema com o nosso Workflow, e aconteceu porque tal como mencionei quando descrevi cada uma das categorias de atividades dos workflows, a atividade "TerminateWorkflow" tem de lançar uma exceção, para definir a exceção clique sobre esta última atividade e de seguida na área de propriedades no seguinte botão:

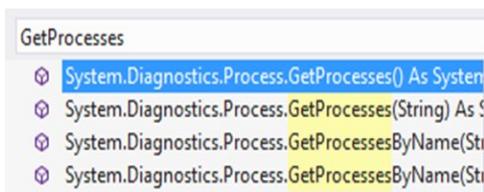


Defina a seguinte exceção:

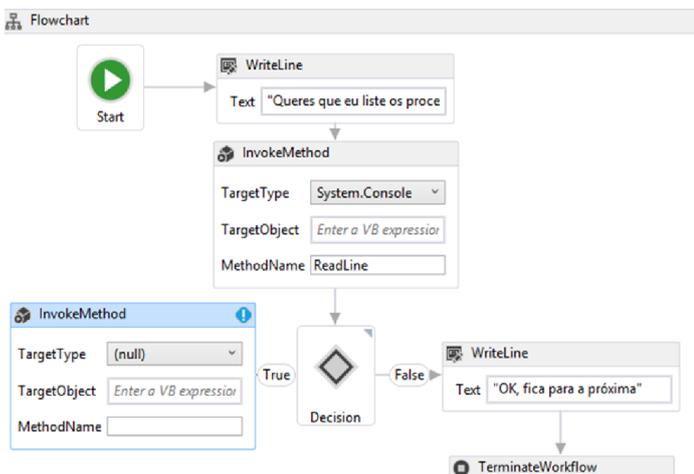


Prima OK, e repare que os avisos desapareceram.

Agora vamos tratar do lado do true, para que os processos sejam listados, vamos ter de invocar o método GetProcesses que se encontra em System.Diagnostics:



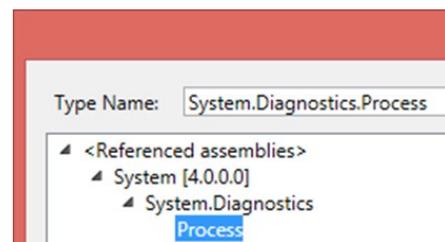
Assim, arraste para o designer um InvokeMethod e ligue-o ao lado "True" do diamante de decisão:



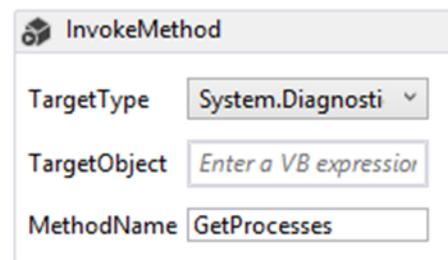
Mais uma vez, não se preocupe com os avisos, pois ainda não configurámos o InvokeMethod.

A configuração, deverá ser a seguinte, invocamos o System.Diagnostics.Process.GetProcesses, e vamos guardar a listagem num array de processos, para que depois através de um ciclo "For Each" possamos percorrer o mesmo e escrever no ecrã cada um dos processos.

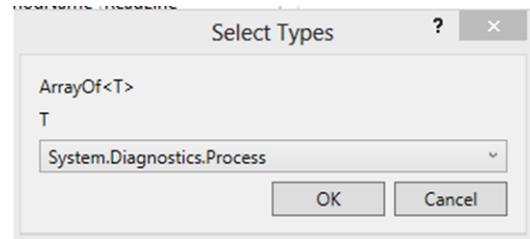
Assim, em TargetType, vá mais uma vez a "Browse for Types" e escreva "System.Diagnostics.Process":



De seguida em MethodName escreva "GetProcesses":



Agora temos então de declarar o Array de processos, para que o InvokeMethod guarde o resultado da invocação do GetProcesses, ou seja para que os nomes de todos os processos sejam lá guardados, assim tal como fez anteriormente, não esquecendo de selecionar primeiro o âmbito, ou seja o FlowChart, declaremos uma nova variável com o nome "ArrayProcessos", em Variable Type, seleccione "Array of [t]", o que fará aparecer uma janela para especificar o genérico t, deverá seleccionar "System.Diagnostics.Process" e clicar em OK:



Ficaremos então com as variáveis declaradas da seguinte forma:

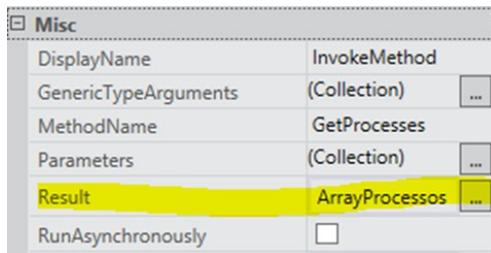
SimOuNao	String	Flowchart	Enter a VB expression
ArrayProcessos	Process[]	Flowchart	Enter a VB expression

Clique no InvokeMethod que criámos e nas propriedades em Result, escreva o nome da variável para a qual queremos

# A PROGRAMAR

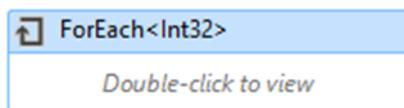
## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

que os processos sejam guardados, neste caso "ArrayProcessos":

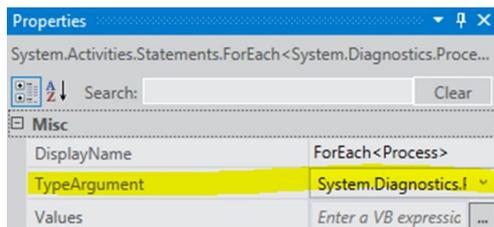


E podemos dar por concluído mais este InvokeMethod, agora só falta mostrar o conteúdo do Array ao utilizador, para isso iremos utilizar uma atividade ForEach que percorra o mesmo e depois escreva cada item deste através de um WriteLine.

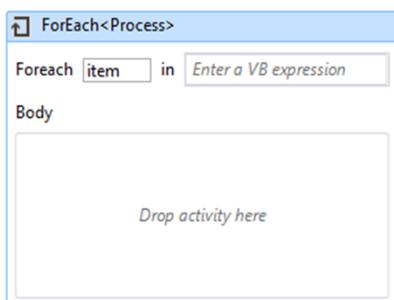
Passando mais uma vez à ação, vá à categoria "Control Flow" e arraste para a área do designer, mais especificamente para baixo do InvokeMethod, uma atividade "ForEach<t>", que ficará com o seguinte aspeto:



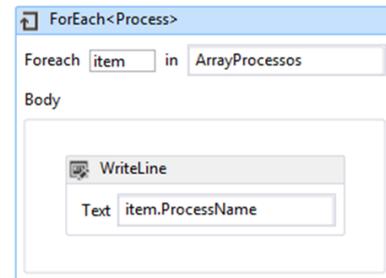
Primeiro temos de alterar o tipo do genérico <t> que por defeito é Int32, mas como não vamos trabalhar com inteiros, vamos ter processos, certificando-se que esta última atividade está selecionada, vá à área de propriedades e altere o valor do TypeArgument para "System.Diagnostics.Process", será este o tipo da variável que vai percorrer o array:



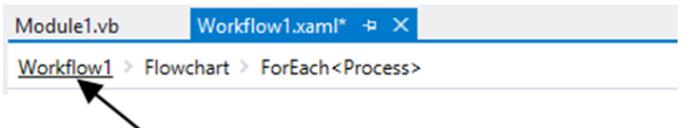
Certamente, terá reparado em algo que ainda não foi visto em outras atividades, neste caso a inscrição "Double-click to view" dentro da mesma, isso significa que iremos poder especificar mais propriedades fazendo esse evento, assim fazer esta ação, surgirá o seguinte:



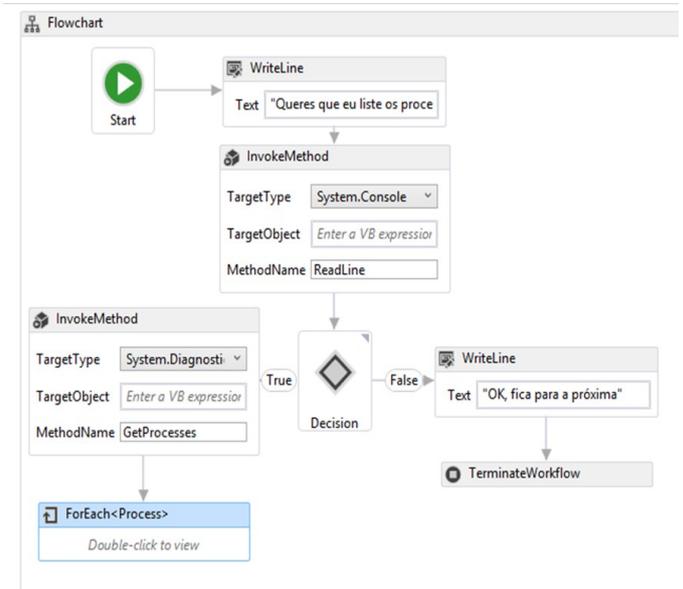
Como pode verificar, trata-se de uma representação gráfica do ciclo ForEach, assim em Enter a VB expression, escreva o nome do Array, neste caso ArrayProcessos e em Drop activity here, arraste um WriteLine e escreva dentro dele "item.ProcessName", o aspeto final será o seguinte:



Volte à vista de Workflow do Designer, clicando em Workflow1:



Ligue o InvokeMethod ao ForEach, e terá então a versão final do diagrama, que poderá conferir na figura seguinte:



Agora, para terminar e como temos uma atividade a lançar uma exceção, mais especificamente a "TerminateWorkflow", devemos capturá-la e mostrar ao utilizador, utilizando para isso um bloco "Try Catch", assim, vamos ao Module1, e escrevemos o seguinte código:

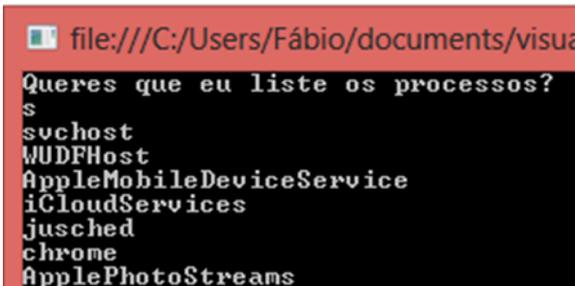
```
Module Module1
    Dim s As Sequence
    Sub Main()
        Try
```

# A PROGRAMAR

## INTRODUÇÃO À WINDOWS WORKFLOW FOUNDATION

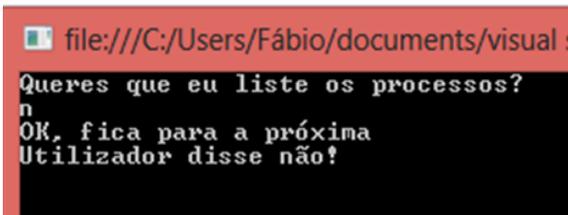
```
Dim workflow1 = New Workflow1()  
WorkflowInvoker.Invoke(workflow1)  
Catch ex As Exception  
Console.WriteLine(ex.Message)  
End Try  
Console.ReadLine()  
  
End Sub  
  
End Module
```

Agora, pode então fazer o debug e ver o resultado do Workflow, assim se premir a tecla s quando for colocada a questão, terá um resultado semelhante ao seguinte:



```
file:///C:/Users/Fábio/documents/visual...  
Queres que eu liste os processos?  
s  
svchost  
WUDFHost  
AppleMobileDeviceService  
iCloudServices  
jusched  
chrome  
ApplePhotoStreams
```

Caso contrário, se especificar outra letra, neste caso um n, o resultado irá ser este:

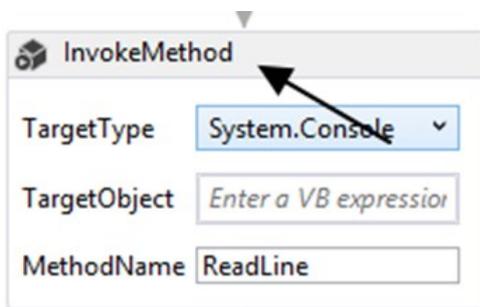


```
file:///C:/Users/Fábio/documents/visual...  
Queres que eu liste os processos?  
n  
OK, fica para a próxima  
Utilizador disse não!
```

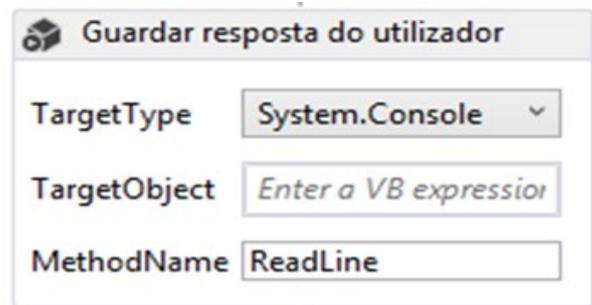
Foram então escritas, a mensagem especificada no WriteLine e a mensagem da exceção.

### Comentar atividades

Se quisermos comentar uma atividade, para isso basta dar um clique no nome da atividade:



O que irá permitir a edição do mesmo:



“ **A principal vantagem que pode ser obtida através da utilização de Workflows é a simplificação da implementação de processos nas nossas aplicações (...)** ”

### Conclusão

Neste artigo, foi apenas mostrada uma brevíssima introdução à Windows Workflow Foundation, mais propriamente sobre a utilização do Designer para elaborar os modelos. Muito mais há a explorar a partir daqui!

## AUTOR



Escrito por Fábio Domingos

licenciado em Gestão de Sistemas de Informação pela Escola Superior de Ciências Empresariais do Instituto Politécnico de Setúbal. Tem como principais hobbies a informática e a fotografia. Mantém um blog em [www.fabiodomingos.com](http://www.fabiodomingos.com).

# A PROGRAMAR

## P vs NP Compreender o maior problema por resolver em computação

Se estão a ler este artigo e estão interessados em tornarem-se milionários rapidamente, este é um bom problema para atacar. No ano 2000, o Clay Mathematics Institute definiu uma lista de 7 problemas aos quais chamou "Millennium Prize Problems" e a cada um destes foi atribuído um prémio de US \$1,000,000. Após a leitura deste artigo espero que consigam compreender melhor este que é, discutivelmente, o mais importante problema da ciência da computação.

Informalmente, o problema pede para demonstrarmos se qualquer problema cuja solução pode ser verificada rapidamente por um computador também pode ser rapidamente encontrada por um. Esquecendo por alguns momentos o que significa exactamente "rapidamente", que iremos detalhar de seguida, é-nos pedido para mostrarmos se todo um conjunto de problemas pode ou não ser resolvido de modo eficiente! Não desesperem já porque não terão de resolver todos os problemas já imaginados: alguns problemas cumprem um conjunto de critérios que faz com que só necessitemos de nos concentrar num deles, portanto só precisamos de encontrar uma solução eficiente (ou provar que não existe) para um desses problemas.

**“ ... o problema pede para demonstrarmos se qualquer problema cuja solução pode ser verificada rapidamente por um computador também pode ser rapidamente encontrada por um. ”**

### Um exemplo rápido

Dos problemas que eu conheço, aquele que é provavelmen-

te o mais simples de enunciar e que explica essa dificuldade é o problema da soma de subconjuntos. Dado um conjunto de números inteiros, existe algum subconjunto cuja soma seja 0? Por exemplo, dado o conjunto  $\{-2, -3, 15, 14, 7, -10\}$ , a resposta é "sim" porque a soma do subconjunto  $\{-2, -3, -10, 15\}$  é 0. Este é um problema aparentemente simples cuja resposta é "sim" ou "não", mas a solução mais eficiente que conhecemos envolve testar todas as combinações possíveis de números no conjunto e verificarmos se a sua soma é 0. Será que isto é o melhor que podemos fazer?

### Tempo polinomial

Se o nosso grande e famoso problema fosse descrito usando os termos "rapidamente" ou "mais eficiente", certamente que existiria uma grande controvérsia dada a sua subjectividade. O que nós procuramos na verdade é que a nossa solução possa ser executada em tempo polinomial. Isto significa que existe uma expressão polinomial no tamanho do input do nosso algoritmo que serve como limite superior ao tempo de computação.

Alguns exemplos de problemas que conseguimos resolver em tempo polinomial:

- A ordenação de  $N$  números pode ser efectuada efectuando um máximo  $O(N \cdot \log(N))$  comparações.
- O caminho mais curto entre dois nós num grafo pode ser encontrado em  $O(|E| + |V| \log(|V|))$ , sendo  $|E|$  o número de arestas e  $|V|$  o número de nós.
- Um sistema de  $N$  equações com  $N$  incógnitas pode ser resolvido efectuando um máximo de  $O(N^3)$  operações pelo método de eliminação Gaussiana.

Todos estes problemas se encontram no conjunto  $P$ . Por outro lado, alguns exemplos de problemas para os quais não conhecemos nenhuma solução polinomial:

- A solução mais eficiente que conhecemos para o problema do caixeiro-viajante, que procura o caminho mais curto que permita visitar um conjunto pré-determinado de  $N$  cidades, envolve testar todas as permutações das cidades e portanto tem uma complexidade de  $O(N!)$ . ( $N$  factorial)
- SAT, o problema de satisfatibilidade booleana, envolve dado um conjunto de expressões booleanas na forma conjunção de disjunções, como por exemplo  $(x \vee y) \wedge (\bar{x} \vee z)$  encontrar uma atribuição (verdadeiro ou falso) às variáveis que faça a expressão ser verdadeira. A solução mais eficiente que co-

# A PROGRAMAR

## P VS NP COMPREENDER O MAIOR PROBLEMA POR RESOLVER EM COMPUTAÇÃO

nhecemos envolve testar combinações, o que tendo  $N$  variáveis exige no máximo  $O(2^N)$  testes.

### Problemas NP

Mas afinal, o que é que significa **NP**? Não, ao contrário do que muita gente pensa não significa “não polinomial”. Na verdade, a sigla **NP** é uma abreviação de “nondeterministic polynomial time”. Este artigo é demasiado curto para uma explicação detalhada, mas o objectivo do conjunto NP é representar todos os problemas que podem ser resolvidos em tempo polinomial numa máquina de Turing não determinística.

Numa definição alternativa, podemos dizer que um problema (de resposta “sim/não”) se encontra no conjunto **NP** caso seja possível demonstrar que existe, para cada input do problema, uma prova pequena que podemos verificar em tempo polinomial que nos diz que para o input a resposta correcta é “sim”.

Por exemplo, no caso do problema da soma de subconjuntos, uma prova pequena poderia ser um subconjunto. Podemos facilmente, em tempo polinomial, verificar se este subconjunto pertence ao conjunto inicial e se a soma dos seus elementos é 0. Outro exemplo, no caso do problema de satisfatibilidade booleana, podemos rapidamente verificar dada uma atribuição às variáveis (a nossa prova) se a expressão é verdadeira. Podemos então mostrar que estes dois problemas se encontram no conjunto **NP**.

Adicionalmente, todos os problemas do conjunto **P** também se encontram necessariamente no conjunto **NP**. Neste caso, nem precisamos de qualquer “pequena prova”, uma vez que o nosso verificador pode de facto resolver o problema em tempo polinomial. Podemos então concluir que **P** e **NP** não são conjuntos diferentes, mas sim um conjunto que está dentro de outro.

O grande problema **P vs NP** pretende que nós demonstremos se estes dois conjuntos são ou não na realidade o mesmo conjunto.

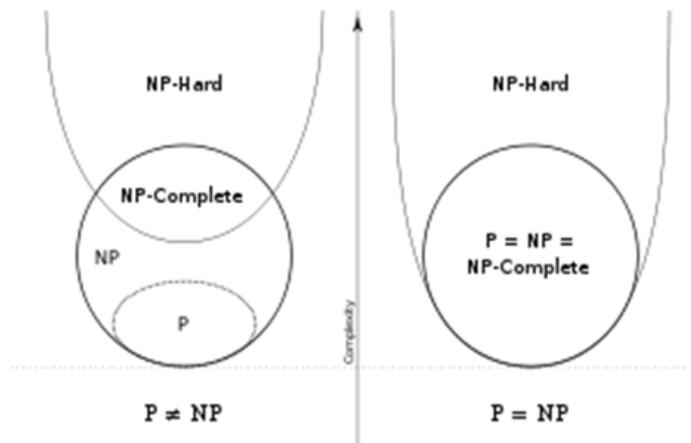
### NP-hard

Adicionalmente, podemos definir um outro conjunto de problemas que são “pelo menos tão difíceis como o problema mais difícil da classe **NP**”. São aqueles problemas que se assumirmos que temos uma solução eficiente para eles então podemos usá-los para construir uma solução polinomial para todos os problemas da classe **NP**.

O problema de satisfatibilidade booleana (SAT) foi o primeiro problema que se demonstrou ser **NP-hard**, através do teorema de Cook cuja prova não irei reproduzir.

A hierarquia seria bonita e simples até que se descobriu que alguns problemas que pertenciam ao conjunto **NP-hard**, na verdade, não pertenciam ao conjunto **NP**. Sim, é estranho mas na realidade nem todos os problemas que denominamos **NP-hard** são de facto **NP** (um problema exemplo pode ser encontrado numa secção posterior). Na figura abaixo encontram um

diagrama de Euler que mostra de que forma se relacionam afinal estes conjuntos.



### NP-complete

Olhando para a figura anterior rapidamente se consegue perceber que um problema **NP-complete** (em português, NP-completo) é um problema que é simultaneamente **NP** e **NP-hard**.

“ *...alguns problemas cumprem um conjunto de critérios que faz com que só necessitemos de nos concentrar num deles, portanto só precisamos de encontrar uma solução eficiente (...) para um desses problemas.* ”

Para simplificar, todos os exemplos anteriores de problemas **NP** que forneci são também problemas **NP-hard**, logo são todos **NP-complete**. Este conjunto de problemas ganhou grande notoriedade quando Richard Karp publicou a sua

# A PROGRAMAR

## P VS NP COMPREENDER O MAIOR PROBLEMA POR RESOLVER EM COMPUTAÇÃO

famosa lista de 21 problemas que demonstrou serem **NP-complete**, tudo utilizando somente o teorema de Cook e efectuando reduções ao problema de satisfatibilidade booleano (isto é, provando que os problemas são “equivalentes”).

Neste caso, o nosso trabalho está muito facilitado! Podemos concluir que só necessitamos de demonstrar que um destes problemas **NP-complete** possui (ou não) uma solução polinomial e automaticamente demonstramos para todos os problemas da classe **NP**! O milhão de dólares está bastante mais perto.

### Problemas indecidíveis e o Halting Problem

Decidi incluir esta secção somente para poder ilustrar um problema muito interessante e que já foi referido anteriormente, o “Halting Problem”. Antes disso, importa saber que um problema indecidível é aquele para o qual foi demonstrado ser impossível construir um algoritmo que produza uma resposta correcta.

É esse o caso do “problema da paragem”: dada uma descrição de um programa e um input, este programa vai terminar ou irá correr indefinidamente? Alan Turing demonstrou que tal algoritmo não pode existir, logo este problema é indecidível nas máquinas de Turing. Facilmente podemos concluir que se este algoritmo não existe, o Halting Problem não pode pertencer à classe **NP** pois é impossível construir o tal “verificador” em tempo polinomial que verifica a “prova pequena”.

No entanto, este problema é **NP-hard**, ou seja, uma solução para este problema permite resolver todos os problemas da classe **NP**. Deixo um desafio: conseguem provar de que forma possuindo um oráculo para este problema é possível resolver o problema de satisfatibilidade booleana?

### P vs NP e além

O nosso mundo gira assumindo que **P** e **NP** são dois conjuntos diferentes de problemas, mas uma prova de que são o mesmo conjunto não implica que seja possível construir um algoritmo que resolva em tempo polinomial todos os problemas **NP**. (isto é, a prova pode ser não construtiva).

Ainda assim, uma solução polinomial para SAT quebraria grande parte dos sistemas criptográficos actuais, como chaves públicas-privadas (a base de grande parte da segurança informática) e mesmo cifras simétricas como AES ou 3DES usadas em comunicações. Por outro lado, um enorme conjunto de problemas difíceis em investigação operacional são **NP-complete** e poderíamos agora resolvê-los eficientemente, como programa-

ção inteira, ou problemas importantes como o da previsão da estrutura de proteínas.

Contudo, o que a maioria dos investigadores espera é que **P** e **NP** sejam na verdade dois conjuntos diferentes. A crença neste facto é tal que parte da investigação actual parte deste princípio, pelo que não existiriam benefícios directos para o nosso dia-a-dia como numa prova em sentido contrário.

“ (...) procuramos na verdade é que a nossa solução possa ser executada em tempo polinomial. (...) existe uma expressão polinomial no tamanho do input (...) ”

Neste artigo foram deixadas de lado muitos outros conjuntos de problemas além de **P** e **NP** que possuem problemas igualmente desafiantes em aberto, nomeadamente PH (polynomial hierarchy) e PSPACE. Na realidade também não sabemos se são diferentes ou iguais entre si, sabemos somente que na cadeia  $NL \subseteq P \subseteq NP \subseteq PH \subseteq PSPACE$

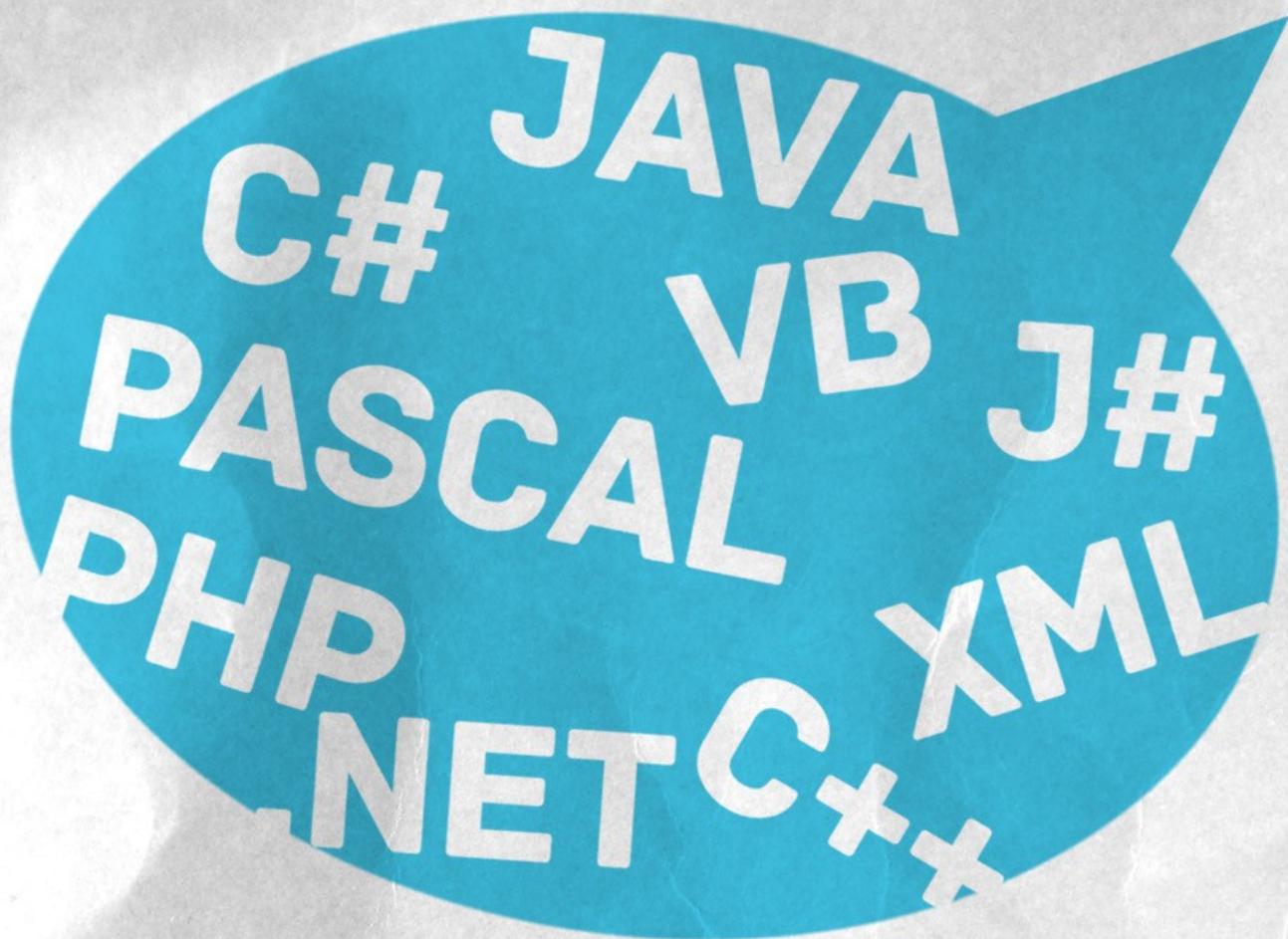
pelo menos uma das inclusões tem de ser estrita. Ou seja, pelo menos um daqueles  $\subseteq$  é na realidade  $\subset$ , mas não sabemos qual deles, desconfia-se que sejam todos. Portanto, deixando um conselho da boca do Professor Manuel Blum, se querem provar que **P** é diferente de **NP** podem começar por provar que **P** é diferente de **PSPACE**, é provavelmente mais fácil e será igualmente importante.

## AUTOR



Escrito por Miguel Araújo

Recebeu um Mestrado em Engenharia Informática e Computação pela Faculdade de Engenharia da Universidade do Porto e é actualmente aluno de doutoramento do Programa CMU|Portugal investigando data mining e detecção de anomalias em grafos.



ENTÃO, SÓ FALAS  
EM CÓDIGO?

TEMOS O REMÉDIO CERTO PARA TI!



[portugal-a-programar.pt](http://portugal-a-programar.pt)

A MAIOR COMUNIDADE PORTUGUESA DE  
PROGRAMAÇÃO, APARECE!

# COLUNAS

**C# - Trazendo Async e Await ao Serviço de Contactos do Windows Phone**

**Visual (Not) Basic - Operador Like**

**Core Dump - A documentação é algo imprescindível**

# C# - TRAZENDO ASYNC E AWAIT AO SERVIÇO DE CONTACTOS DO WINDOWS PHONE

Com a introdução de capacidades assíncronas na linguagem de programação C# torna-se mais fácil a programação assíncrona por se poder escrever de forma sequencial código que será executado de forma assíncrona.

No entanto, as novas funcionalidades assentam no [modelo de assincronismo baseado em tarefas](#) e nem todas as APIs o utilizam.

Para as APIs utilizam o [modelo de programação assíncrona \(APM\)](#) introduzido inicialmente para a plataforma .NET existe uma conversão disponibilizada pelo [método FromAsync da classe TaskFactory](#).

Mas para as APIs que usam [assincronismo baseado em eventos](#) (como é o caso do serviço de [Contacts](#) do Windows Phone) não foi disponibilizada nenhuma conversão.

A razão de não existir uma conversão do padrão de assincronismo baseado em eventos para o padrão de assincronismo baseado em tarefas é porque este não apresentam uma estrutura ou comportamento comum. Uns permitem cancelamento e erros (como é o caso da classe [WebClient](#)) e outros não (como é o caso da classe **Contacts**).

Neste artigo vamos ver várias formas de trazer **async/await** ao serviço de contactos do Windows Phone.

Trazendo Async E Await Ao Serviço De Contactos Do Windows Phone Usando Métodos de Extensão

A forma mais simples de criar um resultado futuro é usando a classe [TaskCompletionSource](#).

A forma de usar esta classe quando se lida com o padrão de assincronismo baseado em eventos é criar uma instância com um resultado do tipo pretendido e subscrever o evento pretendido para atribuir o resultado. Em seguida devolve-se ao chamador o tarefa (**Task**) criada para obtenção futura do resultado:

```
public static class ContactsAsyncExtensions
{
    public static Task<IEnumerable<Contact>>
        SearchAsync(
            this Contacts contacts,
            string filter,
            FilterKind filterKind)
    {
        var taskCompletionSource =
            new
            TaskCompletionSource<IEnumerable<Contact>>();

        EventHandler<ContactsSearchEventArgs>
            handler = null;
        handler = (s, e) =>
        {
            contacts.SearchCompleted -= handler;
```

```
            taskCompletionSource.TrySetResult
                (e.Results);
        };
        contacts.SearchCompleted += handler;

        contacts.SearchAsync(filter, filterKind,
            null);

        return taskCompletionSource.Task;
    }
}
```

A utilização de código seria algo deste género:

```
async Task<IEnumerable<Contact>>
    GetContactsAsync()
{
    var contactsService = new Contacts();
    var contacts = await contactsService
        .SearchAsync(
            filter:"filter",
            filterKind:FilterKind.None);
    return contacts;
}
```

Usando Async E Await Na Implementação de SearchAsync

Note-se que, uma vez que não temos controlo sobre a instância de **Contacts** sobre a qual estamos a operar, a primeira coisa que fazemos no tratamento do evento é cancelar a subscrição do evento.

De igual forma, caso ocorresse uma exceção na chamada ao método **SearchAsync** da classe **Contacts** deveríamos também cancelar a subscrição do evento.

Ao usarmos **async** e **await** também na implementação, o código torna-se mais simples e entendível.

```
public async static Task<IEnumerable<Contact>>
    SearchAsync(this Contacts contacts, string filter,
        FilterKind filterKind)
{
    var tcs = new
        TaskCompletionSource<IEnumerable<Contact>>();

    EventHandler<ContactsSearchEventArgs>
        handler = null;
    handler = (s, e) => { tcs.TrySetResult
        (e.Results); };

    try
    {
        contacts.SearchCompleted += handler;
        contacts.SearchAsync(filter, filterKind,
            null);

        return await tcs.Task;
    }
    finally
    {
        contacts.SearchCompleted -= handler;
    }
}
```

## C# - TRAZENDO ASYNC E AWAIT AO SERVIÇO DE CONTACTOS DO WINDOWS PHONE

Como se pode ver nesta simples implementação, a utilização de **async** e **await** torna o código mais legível.

### Trazendo Async E Await Ao Serviço De Contactos Do Windows Phone Usando Encapsulamento

Embora a utilização de métodos de extensão seja uma forma muito válida de estender tipos dos quais não podemos ou não queremos derivar, muitas vezes podem advir ganhos do encapsulamento da classe que pretendemos estender.

Ao estender a classe **Contacts** passamos a controlar o seu ciclo de vida e a forma como os seus métodos são chamados.

Na nossa implementação começamos por, ao invés de usar uma instância de **Contacts**, criar a nossa própria instância e subscrever de imediato o evento **SearchCompleted**.

Nas implementações anteriores da subscrição do evento fazia-se sempre uso da instância de **TaskCompletionSource** criada no momento o que agora não vai ser possível, uma vez que a subscrição já foi criada. Então como vamos fazer chegar a instância **TaskCompletionSource** que nos vai dar acesso ao resultado futuro à subscrição do evento?

Este problema de correlação entre a chamada e resposta assíncrona coloca-se com frequência e por essa razão quase todas as APIs disponibilizam a passagem de estado entre a chamada e a resposta.

No caso da classe **Contacts** essa passagem de estado é feita passando o estado no argumento **state** do método **SearchAsync** e obtendo-o na propriedade **State** dos argumentos do evento (**ContactsSearchEventArgs**).

A nossa implementação de **SearchAsync** passa então a ser:

```
public Task<IEnumerable<Contact>> SearchAsync(
    string filter,
    FilterKind filterKind)
{
    var tcs = new TaskCompletionSource<IEnumerable<Contact>>();

    this.contacts.SearchAsync(
        filter: filter,
        filterKind: filterKind,
        state: tcs);

    return tcs.Task;
}
```

E a subscrição do evento passa a ser:

```
this.contacts.SearchCompleted += (s, e) =>
{
    (e.State as TaskCompletionSource<IEnumerable<Contact>>)
        .TrySetResult(e.Results);
};
```

Para terminar a implementação basta expor a propriedade **Accounts** para manter toda a funcionalidade da classe original.

Aqui fica a implementação completa da nossa classe:

```
public class ContactsAsync
{
    private Contacts contacts = new Contacts();
    public ContactsAsync()
    {
        this.contacts.SearchCompleted += (s, e) =>
        {
            (e.State as TaskCompletionSource<IEnumerable<Contact>>)
                .TrySetResult(e.Results);
        };
    }
    public IEnumerable<Account> Accounts
    {
        get { return this.contacts.Accounts; }
    }
    public Task<IEnumerable<Contact>> SearchAsync(
        string filter,
        FilterKind filterKind)
    {
        var tcs = new TaskCompletionSource<IEnumerable<Contact>>();
        this.contacts.SearchAsync(
            filter: filter,
            filterKind: filterKind,
            state: tcs);
        return tcs.Task;
    }
}
```

### Recursos

- [Programação assíncrona com async e await \(C# e Visual Basic\)](#)
- [Resumo do padrão de assincronismo baseado em tarefas](#)
- [Modelo de programação assíncrona \(APM\)](#)
- [Método TaskFactory.Async](#)
- [Resumo do padrão de sincronismo baseado em eventos](#)

## AUTOR



### Escrito por Paulo Morgado

Bacharel em Engenharia Electrónica e Telecomunicações (Sistemas Digitais) pelo Instituto Superior de Engenharia de Lisboa e Licenciado em Engenharia Informática pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa exerce variadas funções relacionadas com o desenvolvimento, distribuição e manutenção de software há mais de 10 anos. Participa em diversas comunidades nacionais e internacionais (pontoNETpt, NetPonto, SharePointPT, SQLPort, Portugal-a-Programar, CodeProject, CodePlex, etc.). Pelo seu contributo para com estas comunidades, a Microsoft premeia-o com o prémio MVP (C#) desde 2003. É ainda co-autor do livro "LINQ Com C#" da FCA. <http://PauloMorgado.NET/> - @PauloMorgado

# ISEL Tech '13

28 de Maio a 30 de Maio de 2013

Lisboa, Portugal



O **ISEL Tech** é um evento anualmente realizado no campus do ISEL, de entrada **100% FREE** para todos os visitantes (sejam eles do ISEL ou não), onde damos voz ao que de melhor se faz em Portugal a nível tecnológico e de Engenharia. É também um lugar onde jovens estudantes podem entrar em contacto com as empresas da sua área, conhecer projetos interessantes, trocar impressões, CV's ou até mesmo arranjar emprego.

Ao longo das 3 edições anteriores, o ISEL Tech contou com talks, apresentação e demonstração de produtos, workshops, competições de hacking/coding, exposição de empresas e startups, brindes e muitos prémios.

Esta quarta edição será realizada de **28 a 30 de Maio** e tem como objetivo abordar temas como: Mobile, Web, Cloud, Segurança, Empreendedorismo, Startups, Projectos em parceria com o ISEL, Gaming, Multimédia entre outros...

Acima de tudo queremos despertar o espírito inovador que há dentro de ti!



# VISUAL (NOT) BASIC

## OPERADOR LIKE

Este poderá parecer um artigo bastante simples sobre Visual Basic e até é na verdade, mas a realidade é que muito programadores ou aspirantes a programadores de Visual Basic não utilizam determinadas funcionalidades, acredito que por desconhecimento, que podem simplificar bastante o código.

O operador Like não está disponível no C# e pode ser substituído pela utilização de Expressões Regulares (Regular Expressions). No entanto, embora as expressões regulares sejam muito mais poderosas e versáteis, pequenas validações podem ser feitas mais facilmente usando este operador.

O operado Like é um dos operadores de comparação disponíveis no Visual Basic assim como o = (igual), <= (menor ou igual), >= (maior ou igual), <> (diferente), Is e IsNot. Os operadores de comparação, como deve ser do vosso conhecimento, devolvem um resultado booleano (true ou false).

*resultado = texto like mascara*

O seguinte exemplo mostra a utilização do operador Like:

```
Dim local As String = "Lisboa"
Dim resultado As Boolean = local Like "Lisboa"
' Resultado: resultado = true
```

O resultado da expressão indicada (local Like "Lisboa") é True.

É um exemplo bastante simples e que até poderia ser utilizado o operador = (igual) pois está a validar se uma *string* é igual a outra *string*, mas a vantagem de utilizarmos o operador Like é que nos permite também utilizar *wildcards* para a mascara de validação.

Existem três *wildcards* que podemos utilizar:

"\*" – Substitui um grupo de caracteres

"?" – Substitui um caractere

"#" – Substitui um dígito

Alguns exemplos utilizando o *wildcard* asterisco (\*) e respectivos resultados:

```
Dim word As String = "Revista Programar"

Console.WriteLine(word Like
"*Programar")
' Resultado: True

Console.WriteLine(word Like "*ista*")
' Resultado: True

Console.WriteLine(word Like "*Revista")
' Resultado: False
```

Agora usando o *wildcard* ponto de interrogação (?)

```
Dim word As String = "VB.NET"

Console.WriteLine(word Like "VB.NE?")
' Resultado: True

Console.WriteLine(word Like "VB.?")
' Resultado: False

Console.WriteLine(word Like "VB.??")
' Resultado: True
```

E finalmente usando o *wildcard* cardinal (#):

```
Dim cod As String = "2900-110"

Console.WriteLine(cod Like "2900-11#")
' Resultado: True

Console.WriteLine(cod Like "2900-#")
' Resultado: False

Console.WriteLine(cod Like "2900-###")
' Resultado: True
```

Um exemplo conjugando vários *wildcards*, de modo a mostrar uma validação mais complexa poderá ser:

```
Dim artigo As String = "PT-AB1234"

If artigo Like "*-??####" Then
    Console.WriteLine("Artigo válido")
End If
```

Nesta validação, que nos dá um resultado verdadeiro, estamos a garantir que o artigo inicia por um texto qualquer (\*), seguido de um hífen (-), tendo de seguida dois caracteres alfanuméricos (??) e três números (###).

Mas existem ainda mais formas de usar o operador Like. Podemos definir, entre parênteses rectos, que letras são permitidas na substituição ou entre que intervalos são permitidos usar. Podemos também definir múltiplos intervalos entre os parênteses rectos.

É importante ter em atenção a distinção entre maiúsculas e minúsculas e, no caso de intervalos, definir da primeira letra para a última (de acordo com a ordem do abecedário).

Alguns exemplos para que entenda melhor:

```
Dim r = "Revista" Like "Revist[aeiou]"
' Resultado: True
```

Neste caso a última letra da palavra "Revista" está entre as permitidas nesta validação [aeiou], por isso o resultado é verdadeiro.

```
Dim r = "Revista" Like "[a-z]evista"
' Resultado: False
```

# VISUAL (NOT) BASIC

## OPERADOR LIKE

O resultado é falso pois a primeira letra da palavra “Revista” não está entre as letras a e z pois esta encontra-se em maiúsculas (“R”).

```
Dim r = "Revista" Like "[RV]evist[a-c]"  
' Resultado: True
```

Neste exemplo o resultado é verdadeiro pois a primeira letra é um “R” ou “V” e a última é um “a”, “b” ou “c”.

Podemos ainda usar da mesma forma, excluindo determinados caracteres, através da indicação de um ponto de exclamação, ou seja:

```
Dim r = "Revista" Like "Revist[!abc]"  
' Resultado: False
```

Ou seja, nesta validação estamos a verificar se a última letra não é um “a”, “b” ou “c” (através de [!abc]), o que torna a expressão falsa pois “Revista” termina em “a”.

Para terminar e para mostrar que podemos usar várias combinações, vamos validar:

1. Se temos no início da palavra uma letra em minúsculas ou a letra “R” - [a-zR]
2. Que a segunda letra não é um “a” – [!a]
3. Que a última letra é um “a”, “b” ou “c” – [a-c]

Usamos ainda *wildcards* para outras validações já referidas anteriormente.

```
Dim r = "Revista" Like "[a-zR][!a]vis?[a-c]*"  
' Resultado: True
```

Por predefinição o modelo de comparação de *strings* está definido como **binário** (Option Compare Binary). Desta forma o intervalo [A-E] valida as letra A, B, C, D e E. Se alterarmos o modelo de comparação para **texto** (Option Compare Text) então o mesmo intervalo valida maiúsculas e minúsculas.

```
Option Compare Text  
Module Module1
```

```
Sub Main()  
Dim r = "ÀeIÓu" Like "æiíóu"  
' Resultado: True  
  
End Sub
```

```
End Module
```

## AUTOR



### Escrito por Jorge Paulino

Exerce funções de analista-programador numa multinacional sediada em Portugal. É formador e ministra cursos de formação em tecnologias Microsoft .NET/VBA, é Microsoft Office Specialist (MOS) e Microsoft Most Valuable Professional (MVP) desde 2009, em Visual Basic, pela sua participação nas comunidades técnicas. É administrador da Comunidade Portugal-a-Programar e membro de várias comunidades (PontoNetPT, NetPonto, MSDN, Experts-Exchange, CodeProject, etc). É autor do blog <http://www.jorgepaulino.com> - twitter [@vbtuga](https://twitter.com/vbtuga)

“ **embora as expressões regulares sejam muito mais poderosas e versáteis, pequenas validações podem ser feitas mais facilmente usando este operador.** ”

O operador Like permite também o seu overloading, à semelhança de outros operadores, alterando desta forma o seu comportamento.

“ **a vantagem de utilizarmos o operador Like é que nos permite também utilizar wildcards para a mascara de validação.** ”

### Conclusão

O operador Like é um operador de comparação que permite validar expressões utilizando mascaras, um pouco à semelhança das Regular Expressions, embora muito mais limitado. No entanto, para muitas validações, pode ser uma opção bastante válida e simples de implementar.

## A documentação é algo imprescindível

A documentação é algo imprescindível. Até aqui todos estamos de acordo. No entanto, há uma relação de amor-ódio com a documentação que resulta num misto de algo que toda a gente quer ter, mas ninguém a quer fazer e muito menos pagar por isso.

Todas as pessoas que trabalham em TI compreendem bem o valor que a documentação representa, no entanto, e vá-se lá saber porquê, as organizações continuam a querer prescindir dela até ao momento em que se arrependem amargamente dessa decisão.

Há algumas semanas atrás, durante a implementação de um mecanismo de recuperação a falhas para sistemas em tempo real baseados em MySQL, queixava-me ao administrador de sistemas do cliente o quanto mais caro lhes estava a sair o facto de não terem o sistema documentado. Nesse momento, fomos como que interrompidos por um outro consultor externo com a seguinte afirmação “**Ninguém faz documentação**”, a qual obteve como resposta imediata “Queres vir ver o nosso código?”...

Com muita pena minha, esta mentalidade de desleixo proliferou nesta área, e a culpa, a responsabilidade, e o exemplo vêm de cima.

Ao longo dos anos fui-me cruzando com vários *managers* e clientes que sempre quiseram cortar na documentação e nos testes de forma a fazer entradas em produção mais rápidas e a baixar os custos do produto, ou serviço, final. Mas, na verdade, todos viviam numa mentira em que fingiam acreditar.

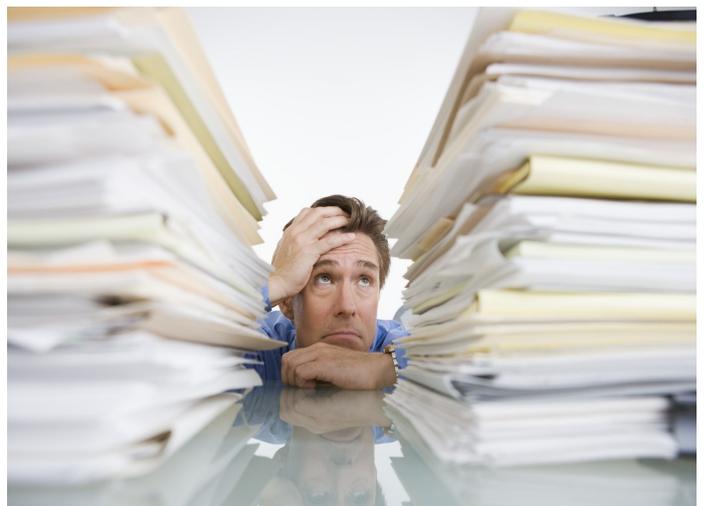
Cheguei a ter discussões, mais ou menos acesas, com *managers* sobre se eram eles quem se iriam responsabilizar pelo facto de me estarem a dizer para reduzir os testes e a documentação a um ponto que, na prática, era equivalente a não fazer testes nem ter documentação.

Os testes, de que não vou falar neste artigo, têm óbvias repercussões na qualidade final e, igualmente óbvio, no custo de manutenção e na imagem que se passa ao cliente.

A documentação, embora o efeito não seja tão imediato, possui um custo igualmente alto, basta pensar nas horas a mais que se consomem a tentar compreender algo que, havendo documentação, se compreenderia em 10 minutos.

O problema da documentação não é de agora, nem começa sequer no código. Em toda a minha carreira, foram poucas as vezes que vi documentos de requisitos, especificações e afins. Qualquer quadro médio, ligado às TI, de uma organização sabe que quanto mais cedo se detetar um problema, mais barata é a sua resolução, e no entanto continuo a ver as organizações a ignorarem a documentação relativa às especificações. Se as organizações adotassem um parâmetro para especificações de software, por exemplo o standard IEEE-830, poupariam uma quantidade de tempo e de dinheiro considerável, no entanto, continuam a preferir fazer “requisitos” em documentos de texto ou folhas de cálculo sem a devida validação e coerência necessária a um documento que é a génese de algo que suportará o negócio, ou parte dele, da organização.

**A documentação tem de deixar de ser vista como um drama** ou como um mal necessário e **tem de passar a ser vista como um ativo das organizações**, um bem imprescindível que torna os seus departamentos de TI mais céleres nas respostas às necessidades da organização. Basta pensar na redução de tempo no *time-to-market* para o lançamento de um novo produto ou no preço por hora de um consultor externo que consegue produzir resultados melhores e mais depressa.



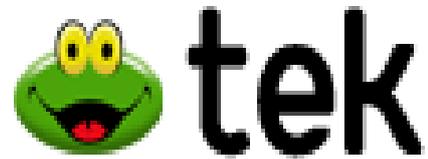
### AUTOR



#### Escrito por Fernando Martins

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

# Media Partners da Revista PROGRAMAR





À semelhança do ano lectivo 2011/2012, o Instituto Politécnico de Viana do Castelo levou a cabo a 2ª edição da Mostra. Durante os dias 7 e 8 de Março, alunos e professores das seis escolas do IPVC estiveram presentes nas instalações da Associação Empresarial de Viana do Castelo.

O objetivo deste evento foi o de dar a conhecer as ofertas formativas do Instituto aos futuros alunos do ensino superior. Florbela Correia, vice-presidente do IPVC apresenta este projecto nascido da vontade de, enquanto única instituição de ensino superior pública no Alto-Minho, reunir num único espaço todas as ofertas formativas da instituição, facilitando assim a visita de futuros alunos bem como respectivos pais.

O evento contou com exposições de cada escola, bem como demonstrações e conferências, nas quais os visitantes puderam ver mais de perto o trabalho desenvolvido nos diferentes cursos.



Este evento contou com a visita de cerca de mil jovens e com mais de quinze instituições da região, bem como algumas

exteriores a Viana do Castelo.

O Instituto Politécnico de Viana do Castelo tem uma vasta oferta formativa na área das informáticas com grande envolvimento de programação como é o caso do CET TPSI (Tecnologias e Programação de Sistemas de Informação); Licenciatura em Engenharia Informática; Licenciatura em Engenharia Electrónica e Licenciatura em Engenharia Computação Gráfica. Foram apresentados alguns projectos desenvolvidos pelo curso de Engenharia de Computação Gráfica e Multimédia, dos quais se destacaram, um jogo que utilizava o Kinect da Microsoft, com a SDK, foi desenvolvido em C#, utilizando Windows Presentation Foundation

Como não poderia deixar de ser, foi possível assistir à actualização da tuna, referência importante no mundo académico, bem como contactar com algumas experiências dos alunos do programa *Erasmus*, que frequentam actualmente este Instituto Politécnico.

Quando questionados sobre esta iniciativa as opiniões foram consensuais como referiu um dos alunos participantes, que esteve presente neste evento: “esta mostra foi importante para dar a conhecer ao exterior a variedade de cursos do politécnico, incentivando novos alunos a ingressarem nos mesmos”.

Os Cursos de Especialização Tecnológica, de nível superior, possibilitam por um lado uma carreira profissional e por outro o ingresso numa licenciatura, na área. O curso de TPSI (Técnico de Programação de Sistemas Informáticos), tem como objectivo formar profissionais com capacidades para revolucionar o mundo da programação. Fazendo-se representar por alunos e docentes, sempre prontos a esclarecer qualquer questão sobre o curso. Os trabalhos realizados pelos alunos, repletos de qualidade, e com enorme capacidade de inovação demonstram a evolução ao longo das várias edições do CET de TPSI.

Esta iniciativa mostrou-se uma mais-valia, permitindo um maior conhecimento do que é desenvolvido no interior do IPVC, aliciando ao ingresso de novos alunos no próximo ano lectivo.

A turma de TPSI 2011/2012 agradece ao Prf. Dr. Salvador Lima, Prof. Sérgio Laranjeira, aos ex-alunos do CET Ana Machado, David Machado, Diogo Soares, Hugo Viana, Miguel Oliveira, Nelson Lima, e demais docentes e alunos todo o seu apoio e dedicação.

## AUTOR



Escrito por António Santos

Entusiasta da tecnologia desde tenra idade, cresceu com o ZX Spectrum, autodidacta, com uma enorme paixão por tecnologia, tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM. Membro da Comunidade [Portugal-a-Programar](#) desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão. <http://linkd.in/Sq13Dc>

# Análises

**jQuery: A sua biblioteca javascript (primeira edição)**

**MATLAB – Curso Completo**

**Desenvolvimento de Aplicações em PHP**

## jQuery: A sua biblioteca JavaScript (primeira edição)

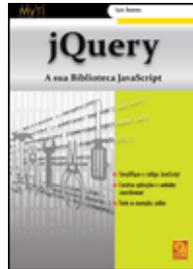
**Título:** jQuery: A sua biblioteca javascript

**Autor:** Luís Soares

**Editora:** FCA - Editora de Informática, Lda.

**Páginas:** 256

**ISBN:** 978-972-722-759-4



Logo quando comecei a ler o livro, ele é claro a indicar que o livro tem intenção de ser para quem está a iniciar a manipulação do DOM usando o jQuery e que o seu objectivo não é mostrar o API extenso do jQuery, mas sim mostrar partes-chave desta biblioteca para ele ser um bom ponto de partida. Ao afirmar isto e, depois, demonstrar-se bem fiel a esse parágrafo ao longo de todo o texto, é um dos elementos-chave que torna este livro realmente muito útil; não só para o seu público-alvo principal, mas também para quem quiser rever alguma parte específica que ele aborda num dos seus bem estruturados capítulos.

Composto por múltiplos capítulos que tornam a aprendizagem lógica e do mais simples ao mais complicado de se conseguir com o jQuery. Começa por introduzir para a funcionalidade mais usada e mais simples, o jQuery, passando pelo ajax e acabando “tocando” no assunto de como lidar com plugins (só a parte sobre os plugins dava direito a um livro!). Começa por pegar na credibilidade da biblioteca, mostrando porque é que é muito boa ideia usar o jQuery e qual é o papel da biblioteca numa aplicação web. Tudo com um texto bem acessível e direto ao assunto. Logo a seguir, em vez de se atirar imediatamente ao API, o livro continua, agora que já mostrou que o jQuery é a opção a seguir, mostrando como funciona a lógica da interface do jQuery. Isto é um ponto positivo muito forte do livro! Este capítulo dá aos leitores as ferramentas multiusos que irão necessitar para entenderem o resto do livro. Para mim, este capítulo foi algo que fez com que a leitura dos outros capítulos fosse muito mais fácil para pensar em como aplicar em código e o como é que funciona. E é aqui que se nota no quão direto e conciso este livro é, com uma escrita muito fácil de ler sem falhar nas precisões de que algo deste tipo realmente precisa. Mais um bom ponto é que este livro apela a que se use boas práticas de escrita de código. Um bom exemplo disto, que ainda me lembro muito bem, é o apelar para separar o CSS do javas-

cript, em que é melhor ter o javascript a trocar classes no HTML do que andar a alterar “manualmente” como cada elemento se deve apresentar na página. Toda a informação no livro está organizada em: Introdução teórica simples -> problema -> E com o jQuery? -> Como será que jQuery ajuda? -> O que é que o jQuery oferece para atacar este tipo de problema?-> Como atacar os problemas deste problema com o jQuery? (Mesmo que não use esta nomenclatura). Isso torna o livro muito mais direto para se conseguir aprender e torna-o também num manual de acesso semi-rápido à resolução de problemas usando o jQuery. É pena, mas este livro tem também umas partes bem negativas que espero que estejam resolvidas em edições futuras. O livro quase não faz referência ao DOM, que seria algo útil para quem já está habituado a lidar com ele e que passou agora usar o jQuery. Não existe exploração sobre como fazer código digamos... “equilibrado” entre quantidade ou complexidade de código e dificuldade no seu processamento por parte do browser, nem truques ou ideias de como melhorar se a velocidade do código estiver a tornar-se um problema por causa da complexidade. Não é dada devida atenção ao pormenor da complexidade dos seletores vs complexidade em seleccionar os elementos - só existe uma passagem rápida, demasiado leve, na minha opinião, e incompleta no assunto. Há também partes no livro em que a parte principal que está a ser aplicado é “faz o que eu digo, não faças o que eu faço”. Um exemplo disso é sobre o fazer cache de um objeto jQuery para uso posterior sem que se tenha que andar a “re-pesquisar” no DOM sempre que a função executa. Embora tenha exemplos que estão a suportar a ideia no sub-capítulo em que isto é referido, parece que é algo que é perdido no resto do tópico, o que acho ser bastante negativo. Embora fazer isto seja mau, não é algo recorrente e acontece em poucos aspectos do livro, mas sinceramente, acho que, no processo de ensino, a melhor opção para tudo é sempre “faz o que te digo, faz o que eu faço” contrariando o provérbio.

Resumindo, este livro sobre jQuery é um livro de não deve, de modo algum, ser ignorado e é uma das melhores leituras sobre uma biblioteca que já alguma vez li, não só comparando com o que já li para js, mas também o que já li para outras linguagens de programação.

### AUTOR



**Escrito por Bruno Almeida**

Actualmente estudante de mestrado em engenharia informática na FCUL (faculdade de ciências da universidade de Lisboa). Muito interessado em desafios de lógica, incluindo programação. Tem conhecimentos sólidos múltiplas linguagens de programação como php, javascript, C e java.

Actualmente dá suporte técnico em free-lancing e está a participar no desenvolvimento do phpBB.

## MATLAB – Curso Completo

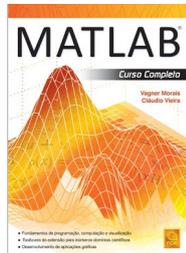
**Título:** MATLAB – Curso Completo

**Autor:** Vagner Morais e Cláudio Vieira

**Editora:** FCA

**Páginas:** 643

**ISBN:** 978 – 972 – 722 – 705 – 1



Sempre ouvi dizer que não devemos avaliar um livro pelo seu tamanho e pela sua capa, mas algo que me tenho apercebido ao longo do tempo e mais propriamente no que toca a livros técnicos é que um livro que tenha poucas páginas nunca poderá ser muito um livro muito aprofundado sobre o tema.

Ao nível físico da edição não tenho nada de errado a apontar, sendo que a tipografia é de qualidade permitindo uma fácil e rápida percepção visual dos conteúdos abordados.

O livro “MATLAB – Curso Completo” é uma obra que se destina a cobrir todos os conteúdos importantes ao conhecimento em profundidade da ferramenta que o MATLAB é para os estudantes, investigadores, engenheiros e profissionais da área Informática.

A organização dos conteúdos foi escolhida mediante a sua importância e dependência entre conteúdos, sendo que não são abordados conteúdos que necessitem de conhecimentos prévios sem que estes tenham sido abordados anteriormente. Na minha opinião é um ponto a favor da organização, pois ao ler sequencialmente a obra não iremos em momento algum sentir que estamos a abordar temas cujas bases ainda não foram abordadas.

Dividido em três grandes partes, as primeiras páginas da obra são orientadas ao esclarecimento e explicação do que é a ferramenta informática MATLAB e a quem se destina. Após a pequena introdução toma lugar o primeiro grande capítulo, e como não poderia deixar de ser aborda os Fundamentos Básicos do MATLAB. Os capítulos seguintes, respectivamente os capítulos dois e três abordam Toolboxes e por fim o desenvolvimento rápido de interfaces gráficas usando a ferramenta GUIDE.

No final de cada subcapítulo existe um resumo de todas as funções necessárias e úteis no respectivo tema o que nos

facilita de certa forma a acção de procurar uma determinada função quando já sabemos previamente em que tema se insere.

O conteúdo é de simples leitura e de fácil percepção, sendo que é usada uma linguagem muito “user-friendly” o que facilita muito a leitura da obra, não se tornando extremamente técnica do ponto de vista da linguagem utilizada, sendo que são dados exemplos variados para cada função e funcionalidade seguidos da explicação desses mesmos exemplos e das funções usadas.

“**conteúdo é de simples leitura e de fácil percepção, sendo que é usada uma linguagem muito “user-friendly” o que facilita muito a leitura da obra, não se tornando extremamente técnica**”

Como estudante de Engenharia Informática considero a obra “MATLAB – Curso Completo” uma obra de qualidade na apresentação dos conteúdos necessários para aprender e conhecer em profundidade a ferramenta informática MATLAB, tendo tudo o que é necessário para se tornar uma obra de referência sobre o tema e escrita em língua Portuguesa.

### 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/>

## Desenvolvimento de Aplicações em PHP

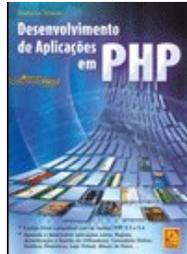
**Título:** Desenvolvimento de Aplicações em PHP

**Autor:** Frederico Tavares

**Editora:** FCA

**Páginas:** 464

**ISBN:** 978-972-722-748-8



Quando comecei a ler o livro pensei, “Bem mais um livro sobre PHP”, tendo em conta que já não é o primeiro que leio e dificilmente será o último, tentei ler como se fosse a primeira vez que lia um livro sobre a linguagem PHP.

Conforme fui lendo, página a página, capítulo a capítulo, saboreando as páginas sem pressas, colocando apontamentos, dei por mim a recordar livros de outrora em que apresentavam uma linguagem ou tecnologia, sempre acompanhada de exemplos práticos e interessantes, que levavam o leitor numa “viagem” de auto-aprendizagem, seguindo os exemplos passo a passo. Algo que não é muito comum nos dias de hoje e em livros de PHP, tão pouco.

Logo na capa o livro deixa clara a informação “Código-fonte compatível com as versões PHP 5.3 e 5.4”, o que por si só ajuda o leitor, pois a grande maioria das funções de acesso a dados que são utilizadas durante o livro foram “deprecated” na versão 5.5 do PHP e substituídas por outras, até serem removidas por completo numa futura versão.

Achei particularmente interessante a forma prática com que são abordados os temas e apresentados os exemplos, com um código “limpo e claro”, bem comentado, de fácil compreensão ao leitor menos experiente, se bem que algum conhecimento base de HTML e CSS se possa revelar uma mais-valia, aumentando a velocidade de aprendizagem.

Os exemplos apresentados são todos eles pertinentes, como o exemplo da “Loja Virtual”, ou do “Calendário Online” mesmo sabendo que já existem diversas ferramentas open-source para esse mesmo efeito.

Gostei particularmente de ver apresentados alguns aspectos como por exemplo Captcha, que nem sempre é usado nos websites, mas que se revela uma ferramenta necessária, o uso de hash's para passwords e outras informações sensí-

veis, a apresentação de bibliotecas como PHPlot, jpGraph, pChart, FPDF, entre outras que se revelam de grande utilidade e quando aplicadas num contexto prático, recheado de exemplos tornam mais fácil a compreensão da sua utilização.

Foi igualmente agradável e já perto do final do livro, ler um capítulo sobre integração com as redes sociais Facebook e Google+, demonstrando uma clara imparcialidade de preferências que achei extremamente bem vinda.

Para quem, como eu se envolveu no mundo da programação, no tempo em que os livros eram mais exemplos de código do que proza e toda a gente saltava grande parte da leitura para ir experimentar os exemplos, este livro tem o “melhor de dois mundos”, ao trazer à memória um passado não assim tão distante, como ao tornar novamente a leitura de um livro sobre uma linguagem de programação, numa descoberta e não, num “fardo” como muitas vezes os livros são vistos pelo público menos “técnico”, ou menos disposto a “esforçar-se” para obter resultados.

Em conclusão, o livro está bem estruturado, com pontos fortes bastante fundamentados, com uma linguagem “leve”, sem termos extremamente tecnicistas e pesados, uma leitura aprazível até para os leitores menos experientes em programação, bastante abrangente e no entanto detalhado na quantidade certa para ser um bom livro de iniciação à linguagem PHP. Os conteúdos abordados são muito semelhantes aos conteúdos abordados nas disciplinas de PHP de alguns cursos secundários e pós-secundários, podendo ser um bom guia no apoio a estudantes que estejam a aprender PHP, pois não é demasiado pesado e tem uma componente prática bastante pertinente e na porção adequada.

Na óptica de quem já leu mais alguns livros sobre PHP, penso que o livro perde um pouco ao não apresentar o paradigma da Programação Orientada a Objectos em PHP, apesar de reconhecer que não creio seja esse o objectivo do livro, creio que fosse uma adição bem vinda, para edições futuras, bem como o acréscimo de um capítulo sobre os modelos MVC e MVVM. De qualquer das formas é uma leitura recomendável e aconselhável para quem quer aprender PHP e não pretende despende muito tempo com conhecimentos demasiado teóricos.

### AUTOR



#### Escrito por António Santos

Entusiasta da tecnologia desde tenra idade, cresceu com o ZX Spectrum, autodidacta, com uma enorme paixão por tecnologia, tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM. Membro da Comunidade [Portugal-a-Programar](#) desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão. <http://linkd.in/Sql3Dc>

# COMUNIDADES

Comunidade NetPonto – Página avançada “sobre” para aplicações de  
Windows Phone

## PÁGINA AVANÇADA “SOBRE” PARA APLICAÇÕES DE WINDOWS PHONE

### Introdução

Cada aplicação que é desenvolvida para submeter na loja, deve conter uma página "Sobre" ou "Acerca de". Esta página tem como objetivo apresentar a informação mais relevante sobre a aplicação, que o utilizador deve saber.

De uma forma geral existe um padrão, para todas as aplicações, sobre isto. Desta forma, destaquemos os pontos principais:

- Nome da aplicação
- Versão da aplicação
- Direitos de autor
- Resumo dos detalhes da aplicação
- Referência web da aplicação, suporte e políticas de privacidade.

Outras funcionalidades que podem ser relevantes para esta página são:

- Avaliar a aplicação
- Enviar "Feedback" para o dono
- Partilhar a aplicação via correio eletrónico
- Partilhar a aplicação via redes sociais

Em alguns casos, algumas aplicações podem apresentar mais detalhes sobre a aplicação, é o caso do histórico da aplicação, regras externas, etc.

### Ecrãs

Os dois primeiros ecrãs mostram a página principal - "HomePage" (com ApplicationBar aberta e fechada, respetivamente) de onde podemos lançar a página "AboutPage". Os outros dois ecrãs mostram a página "AboutPage" (com ApplicationBar aberta e fechada, respetivamente).



**Nota:** HomePage.xaml tem dois botões "About" porque existem dois sítios onde podemos encontrar esta opção, no sentido de ajudar, implementamos em ambos os sítios.

### Código fonte

O código fonte pode ser encontrado no Nokia Projects: About Page Sample.

**Pré - Requisitos** O exemplo usa o padrão de desenvolvimento Model-View-ViewModel (MVVM), que é o padrão recomendado para o desenvolvimento de aplicações para Windows Phone.

**Os elementos do padrão MVVM são:**

- **Views** - representa as páginas (está relacionado com a UI).
- **ViewModel** - representa as classes que serão usadas para conectar a interface com o modelo de dados.
- **Model** - representa as classes que definem as entidades e a lógica de negócio.

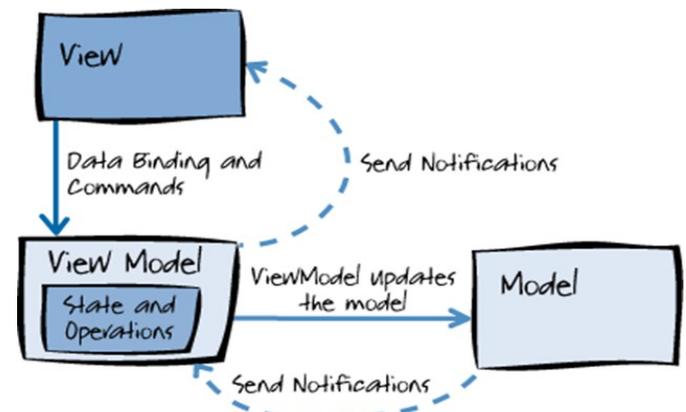


Diagrama com um sumário sobre o padrão MVVM.

# COMUNIDADE NETPONTO

<http://netponto.org>

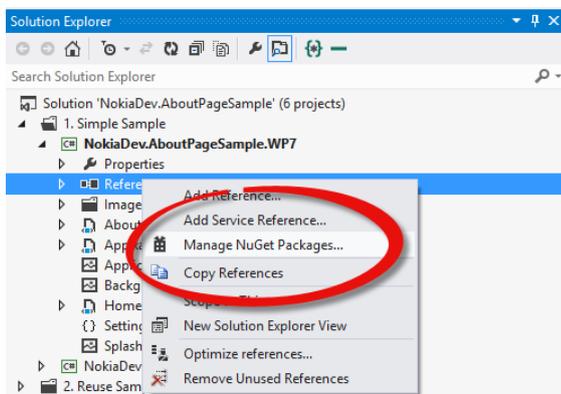
## PÁGINA AVANÇADA "SOBRE" PARA APLICAÇÕES DE WINDOWS PHONE

Para ajudar na implementação do padrão MVVM existem vários toolkits. Para este exemplo, iremos usar os seguintes toolkits:

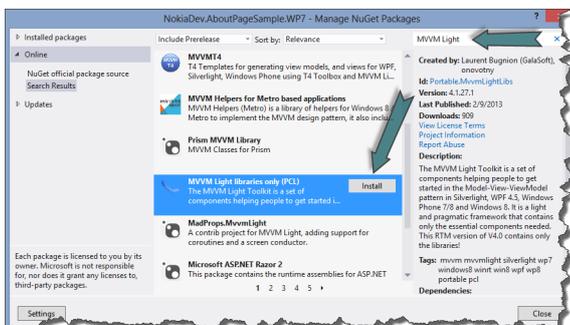
- Cimbalino Windows Phone Toolkit
- MVVM Light Toolkit (PCL version)

Os pacotes são disponíveis através do NuGet Package Manager (para ambos os targets) e podem ser instalados da seguinte forma:

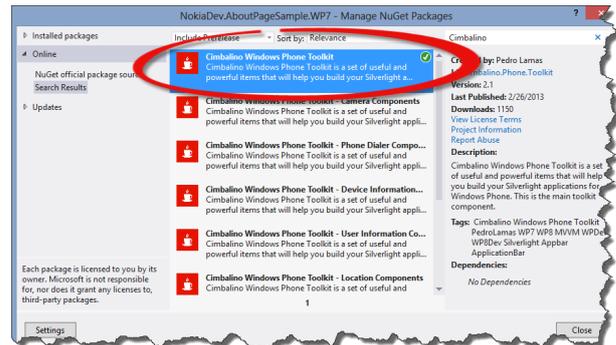
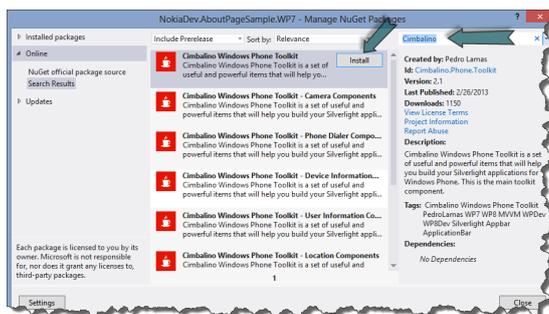
1- Abrir o projeto e selecionar Referencas, em seguida abrir o menu de contexto usando o rato (botão direito) e selecionar "Manage NuGet Packages ..."



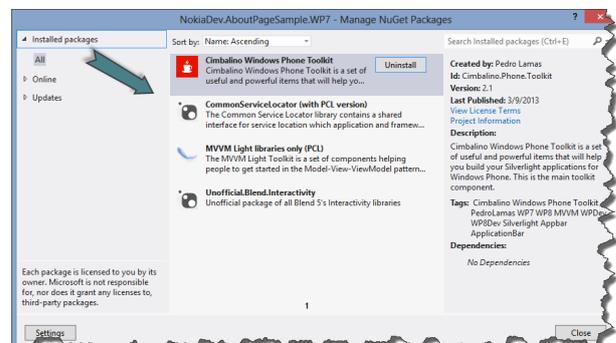
2- Instalação do MVVM Light toolkit



3- Instalação do Cimbalino toolkit



4- Os seguintes passos mostram como podemos ver os pacotes instalados.



**Nota:** Para mais informações sobre a implementação do padrão MVVM consultar as seguintes referências:

- (MSDN) Implementing the MVVM Pattern (\*)
- (Pattern & Praticce) Developing a Windows Phone Application using the MVVM Pattern (\*)
- (Pattern & Praticce) Developing an Advanced Windows Phone 7.5 App that Connects to the Cloud (\*)
- (MSDN) Model-View-ViewModel (MVVM) Applications: General Introduction (\*)

(\*) Artigos disponível, apenas, em Inglês.

### Criação da página "Sobre" ou "Acerca de"

Este exemplo é válido para as várias versões do Windows Phone SDK, não sendo relevante se estamos a desenvolver para Windows Phone OS 7.1 ou para Windows Phone OS 8.0, desta forma podemos seleccionar o que quisermos. No código fonte, podemos encontrar o exemplo completo para ambos os casos.

Começemos com a criação do projeto, o template Windows Phone App foi o selecionado no exemplo.

Em seguida, iremos criar:

- ViewModels - pasta que contém todas as view models;
- Views - pasta que contém todas as vistas;



```
</Grid>
</phone:PhoneApplicationPage>
```

No final do código do XAML, é possível ver a definição para a ApplicationBar que usa o ApplicationBarBehavior do Cimbalino Toolkit. A razão pela qual não estamos aplicar o controlo ApplicationBar do SDK está relacionado ao fato de este controlo não ser um FrameworkElement e não é possível aplicar bindings, o que é um problema para quem pretende implementar o padrão MVVM.

Para mais informações sobre o ApplicationBarBehavior do Cimbalino Toolkit, consultar:

- Cimbalino Windows Phone Toolkit: ApplicationBarBehavior (\*)
- Cimbalino Windows Phone Toolkit: MultiApplicationBarBehavior (\*)

(\*) Artigos disponível, apenas, em Inglês.

About Page.xaml

A página AboutPage será apresentada para mostrar a informação "Sobre" / "Acerca de" da aplicação. Esta página contém alguns controlos para mostrar os pontos descritos na Introdução e contém botões para as várias funcionalidades.

A interface com o utilizador da página AboutPage é a seguinte. Note-se, como estamos a usar o padrão MVVM, toda a lógica estará definida na view model AboutViewModel que será usada no binding com o DataContext.

```
<phone:PhoneApplicationPage
  x:Class="NokiaDev.AboutPageSample.Views.AboutPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"
  xmlns:cimbalino="clr-namespace:Cimbalino.Phone.Toolkit.Behaviors;assembly=Cimbalino.Phone.Toolkit"
  FontFamily="{StaticResource PhoneFontFamilyNormal}"
  FontSize="{StaticResource PhoneFontSizeNormal}"
  Foreground="{StaticResource PhoneForegroundBrush}"
  SupportedOrientations="Portrait" Orientation="Portrait"
  DataContext="{Binding AboutViewModel, Source={StaticResource Locator}}"
```

```
mc:Ignorable="d"
  shell:SystemTray.IsVisible="True">
<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>
  <!--TitlePanel contains the name of the application and page title-->
  <StackPanel Grid.Row="0" Margin="12,17,0,28">
    <TextBlock Text="Nokia Developer Wiki Samples" Style="{StaticResource PhoneTextNormalStyle}"/>
    <TextBlock Text="About" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/>
  </StackPanel>
  <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <StackPanel Margin="0,-12,0,24">
      <TextBlock Style="{StaticResource PhoneTextExtraLargeStyle}" Text="About Page Sample" Foreground="{StaticResource PhoneAccentBrush}"/>
      <TextBlock Style="{StaticResource PhoneTextLargeStyle}" Text="by <Run Text="{Binding Author}"/>"/>
    </TextBlock>
    <StackPanel Orientation="Horizontal" Margin="0,18,0,0">
      <TextBlock FontWeight="Bold" FontSize="{StaticResource PhoneFontSizeMedium}" Style="{StaticResource PhoneTextNormalStyle}" Text="Version:" />
      <TextBlock Margin="0" Text="{Binding Version}" Style="{StaticResource PhoneTextNormalStyle}" x:Name="AppVersion" />
    </StackPanel>
    <TextBlock Style="{StaticResource PhoneTextNormalStyle}" FontSize="{StaticResource PhoneFontSizeMedium}" Margin="12,12,0,0" FontWeight="Bold" Text="Copyright (c) 2013 Nokia Developer Wiki." />
    <TextBlock Style="{StaticResource PhoneTextNormalStyle}" Margin="12,24,0,0" FontSize="{StaticResource PhoneFontSizeSmall}" TextWrapping="Wrap">
      This application show an 'About page' sample for Windows Phone applications.
    </TextBlock>
    <TextBlock Text="See more about it, please see the following reference:" />
  </Grid>
</Grid>
```

```

        </TextBlock>
        <HyperlinkButton NavigateUri="http://
www.developer.nokia.com/Community/Wiki/
About_page_for_Windows_Phone_applications"
        Foreground="{StaticResource
        PhoneAccentBrush}"
        TargetName="_new"
        FontSize="{StaticResource PhoneFontSizeSmall}"
        HorizontalAlignment="Left"
        Content="About page for
        Windows Phone applications" />

        <TextBlock Style="{StaticResource
        PhoneTextNormalStyle}"
        Margin="12,24,0,12"
        FontWeight="Bold"
        Text="More informations:"
        FontSize="{
        StaticResource PhoneFontSizeMedium}" />

        <HyperlinkButton NavigateUri=
"http://www.developer.nokia.com/"
        Foreground="{
        StaticResource PhoneAccentBrush}"
        TargetName="_new"
        HorizontalAlignment="Left"
        Content="Website" />

        <HyperlinkButton NavigateUri=
"http://www.developer.nokia.com/"
        Foreground="{StaticResource
        PhoneAccentBrush}"
        TargetName="_new"
        HorizontalAlignment="Left"
        Content="Support" />

        <HyperlinkButton NavigateUri=
"http://www.developer.nokia.com/"
        Foreground="{StaticResource
        PhoneAccentBrush}"
        TargetName="_new"
        HorizontalAlignment="Left"
        Content="Privacy Policy" />
    </StackPanel>
</Grid>

<i:Interaction.Behaviors>
    <cimbalino:ApplicationBarBehavior>
        <cimbalino:
        ApplicationBarIconButton
        Command="{Binding RateCommand, Mode=OneTime}"

        IconUri="/Images/appbar.rate.png"

        Text="Rate it" />
        <cimbalino:ApplicationBarIconButton
        Command="{Binding
        SendFeedbackCommand, Mode=OneTime}"
        IconUri="/Images/
        appbar.reply.email.png"

        Text="Feedback" />
        <cimbalino:ApplicationBarIconButton
        Command="{Binding ShareToMailCommand,
        Mode=OneTime}"
        IconUri="/Images/appbar.email.png"
        Text="Email" />
        <cimbalino:ApplicationBarIconButton
        Command="{Binding
        ShareSocialNetworkCommand, Mode=OneTime}"

        IconUri="/Images/appbar.share.png"

        Text="Share it" />

```

```

    </cimbalino:ApplicationBarBehavior>
    </i:Interaction.Behaviors>
</Grid>
</phone:PhoneApplicationPage>

```

Assim como foi mostrado na página HomePage, nesta página usamos o ApplicationBarBehavior do Cimbalino Toolkit para definir a application bar com suporte a binding.

### HomeViewModel.cs

A Home Page view model é a seguinte:

```

/// <summary>
/// The home view model.
/// </summary>

public class HomeViewModel
{
    /// <summary>
    /// The navigation service.
    /// </summary>

    private readonly INavigationService
        _navigationService;

    /// <summary>
    /// Initializes a new instance of the
    <see cref="HomeViewModel"/> class.
    /// </summary>
    /// <param name="navigationService">
    /// The navigation service.
    /// </param>

    public HomeViewModel
        (INavigationService navigationService)
    {
        _navigationService = navigationService;
        AboutCommand = new RelayCommand
            (this.ShowAbout);
    }

    /// <summary>
    /// Gets the about command.
    /// </summary>
    /// <value>
    /// The about command.
    /// </value>
    public ICommand AboutCommand { get; private
        set; }

    /// <summary>
    /// The show about.
    /// </summary>
    private void ShowAbout()
    {
        _navigationService.NavigateTo(new Uri
            ("/Views/AboutPage.xaml", UriKind.Relative));
    }
}

```

Na implementação usamos

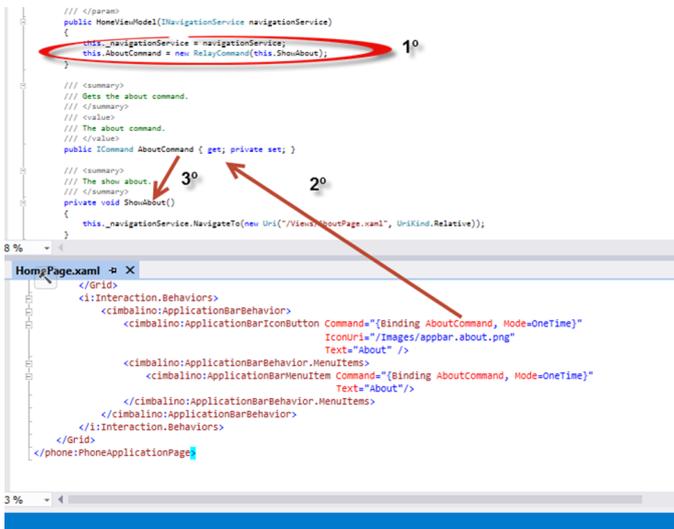
- INavigationService interface do Cimbalino toolkit, cujo código fonte é este. Representa a interface para o serviço com a capacidade de gerir a forma de navegação da aplicação. A sua implementação é NavigationService.
- RelayCommand classe da MVVM Light toolkit, cujo código fonte é este.

# COMUNIDADE NETPONTO

<http://netponto.org>

## PÁGINA AVANÇADA “SOBRE” PARA APLICAÇÕES DE WINDOWS PHONE

A seguinte imagem mostra o fluxo:



### AboutViewModel.cs

A About Page view model é a seguinte:

```
/// <summary>
/// The about view model.
/// </summary>
public class AboutViewModel
{
    /// <summary>
    /// The email compose service.
    /// </summary>
    private readonly IEmailComposeService
        _emailComposeService;

    /// <summary>
    /// The marketplace review service.
    /// </summary>
    private readonly IMarketplaceReviewService
        _marketplaceReviewService;

    /// <summary>
    /// The share link service.
    /// </summary>
    private readonly IShareLinkService
        _shareLinkService;

    /// <summary>
    /// The public application url.
    /// </summary>
    private readonly string _appUrl;

    /// <summary>
    /// The application manifest.
    /// </summary>
    private readonly ApplicationManifest
        _applicationManifest;

    /// <summary>
    /// Initializes a new instance of the
    /// <see cref="AboutViewModel"/> class.
    /// </summary>
    /// <param name="emailComposeService">
    /// The email Compose Service.
    /// </param>
    /// <param name="applicationManifestService">
    /// The application Manifest Service.
    /// </param>
    /// <param name="marketplaceReviewService">
    /// The marketplace review service
```

```
/// </param>
/// <param name="shareLinkService">
/// The share Link Service.
/// </param>
public AboutViewModel(
    IEmailComposeService emailComposeService,
    IApplicationManifestService
        applicationManifestService,
    IMarketplaceReviewService
        marketplaceReviewService,
    IShareLinkService shareLinkService)
{
    _emailComposeService =
        emailComposeService;
    _marketplaceReviewService =
        marketplaceReviewService;
    _shareLinkService = shareLinkService;
    RateCommand = new RelayCommand
        (this.Rate);
    SendFeedbackCommand = new RelayCommand
        (this.SendFeedback);
    ShareToMailCommand = new RelayCommand
        (this.ShareToMail);
    ShareSocialNetworkCommand = new
        RelayCommand(this.ShareSocialNetwork);
    _applicationManifest =
        applicationManifestService.
        GetApplicationManifest();
    _appUrl = string.Concat("http://
        windowsphone.com/s?
        appid=", _applicationManifest.App.ProductId);
}

/// <summary>
/// Gets the author.
/// </summary>
public string Author
{
    get
    {
        return _applicationManifest.App.Author;
    }
}

/// <summary>
/// Gets the version.
/// </summary>
public string Version
{
    get
    {
        return _applicationManifest.App.Version;
    }
}

/// <summary>
/// Gets the rate command.
/// </summary>
public ICommand RateCommand {
    get; private set; }

/// <summary>
/// Gets the send feedback command.
/// </summary>
public ICommand SendFeedbackCommand {
    get; private set; }

/// <summary>
/// Gets the share social network command.
/// </summary>
public ICommand ShareSocialNetworkCommand
{
    get; private set; }

/// <summary>
/// Gets the share to e-mail command.
```

```
/// </summary>
public ICommand ShareToMailCommand { get;
private set; }

/// <summary>
/// The rate.
/// </summary>
private void Rate()
{
    _marketplaceReviewService.Show();
}

/// <summary>
/// The send feedback.
/// </summary>
private void SendFeedback()
{
    const string To = "saramgsilva@gmail.com";
    const string Subject = "My Feedback";
    var body = string.Format(
        "Application {0}\n Version: {1}",
        _applicationManifest.App.Title,
        _applicationManifest.App.Version);
    _emailComposeService.Show
        (To, Subject, body);
}

/// <summary>
/// The share social network.
/// </summary>
private void ShareSocialNetwork()
{
    const string Message = "This
application is amazing, should try it! See in";
    _shareLinkService.Show
        (_applicationManifest.App.Title,
        Message, new Uri
            (_appUrl, UriKind.Absolute));
}

/// <summary>
/// The share to mail.
/// </summary>
private void ShareToMail()
{
    const string Subject =
        "Nokia Developer Wiki Samples - About page";
    var body = string.Concat
        ("This application is amazing, you should try it!
See in", _appUrl);
    _emailComposeService.Show
        (Subject, body);
}
}
```

Na implementação usamos

- Do MVVM Light toolkit

*RelayCommand* classe, cujo código fonte é este, representa a implementação da interface *ICommand*.

- Do Cimbalino toolkit

- *IEmailComposeService* interface, cujo código fonte é este, representa o serviço com a capacidade de enviar mensagens via e-mail. A implementação é EmailComposeService
- *IMarketplaceReviewService* interface, cujo

código fonte é este, representa o serviço com a capacidade de mostrar o ecrã do mercado, para avaliar a aplicação. A implementação é MarketplaceReviewService.

- *IShareLinkService* interface, cujo código fonte é este, representa o serviço com a capacidade para partilhar *links* na rede social, configuradas no dispositivo. A implementação é ShareLinkService.
- *IApplicationManifest* interface, cujo código fonte é este, representa o serviço com a capacidade para ler a informação do ficheiro Manifest. A implementação é ApplicationManifestService.

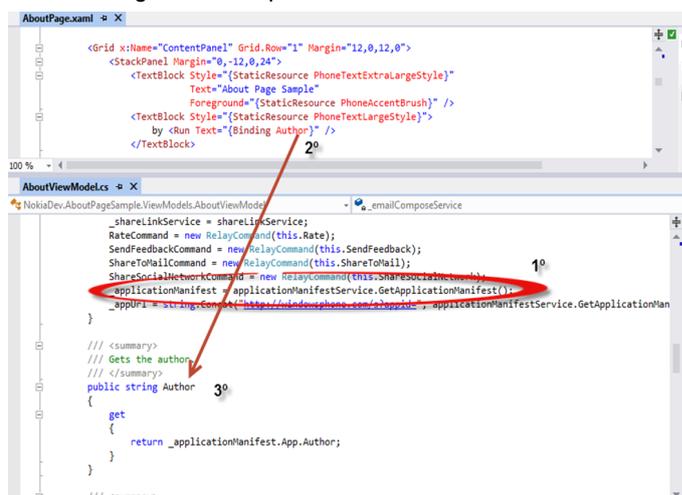
Cada interface é injetada na *View Model*, que são definidas no *ViewModelLocator*. Por exemplo, para o caso do *NavigationService* vamos precisar definir:

```
SimpleIoc.Default.Register
    <INavigationService, NavigationService>();
```

Isto significa que, quando o *loc* container precisa da implementação do *INavigationService*, irá criar um objeto do tipo *NavigationService*. (A ideia é igual para as outras dependências)

Assim como vimos no *HomeViewModel*, cada botão da *AppBar* irá ser binding a um comando que está associado a uma ação.

Veremos agora como aplicamos o nome do *Author*.



A propriedade *Text* da *textBlock* é binding à propriedade *Author* do *AboutViewModel* e esta propriedade irá obter o nome do *Author* definido no ficheiro *Manifest*. Não existe uma implementação direta para obter isto, mas com o uso da *ApplicationManifest* do *Cimbalino Windows Phone Toolkit* não precisamos de nos preocupar com isso, porque esta classe tem esta funcionalidade implementada.

O valor da *Version* é aplicado da mesma forma que o *Author*.

### ViewModelLocator.cs

Para ajudar na conexão da página com a sua view model, iremos usar o ViewModelLocator.

```
/// <summary>
/// The view model locator.
/// </summary>
public class ViewModelLocator
{
    /// <summary>
    /// Initializes a new instance of the
    /// <see cref="ViewModelLocator"/> class.
    /// </summary>
    public ViewModelLocator()
    {
        ServiceLocator.SetLocatorProvider
            (() => SimpleIoc.Default);

        if (!SimpleIoc.Default.IsRegistered<
            INavigationService>())
        {
            SimpleIoc.Default.Register
            <INavigationService, NavigationService>();
        }

        if (!SimpleIoc.Default.IsRegistered<
            IMarketplaceReviewService>())
        {
            SimpleIoc.Default.Register<
            IMarketplaceReviewService,
            MarketplaceReviewService>();
        }

        if (!SimpleIoc.Default.
            IsRegistered<IShareLinkService>())
        {
            SimpleIoc.Default.Register<
            IShareLinkService, ShareLinkService>();
        }

        if (!SimpleIoc.Default.IsRegistered<
            IApplicationManifestService>())
        {
            SimpleIoc.Default.Register<
            IApplicationManifestService,
            ApplicationManifestService>();
        }

        if (!SimpleIoc.Default.IsRegistered<
            IEmailComposeService>())
        {
            SimpleIoc.Default.Register<
            IEmailComposeService, EmailComposeService>();
        }

        SimpleIoc.Default.Register<
            HomeViewModel>();

        SimpleIoc.Default.Register<
            AboutViewModel>();
    }

    /// <summary>
    /// Gets the home view model.
    /// </summary>
    public HomeViewModel HomeViewModel
    {
        get
        {
            return ServiceLocator.
            Current.GetInstance<HomeViewModel>();
        }
    }
}
```

```
    }

    /// <summary>
    /// Gets the about view model.
    /// </summary>
    public AboutViewModel AboutViewModel
    {
        get
        {
            return ServiceLocator.Current.
            GetInstance<AboutViewModel>();
        }
    }

    /// <summary>
    /// The cleanup.
    /// </summary>
    public static void Cleanup()
    {
        // TODO Clear the ViewModels
    }
}
```

No construtor do ViewModelLocator iniciamos o loc container (SimpleIoc) e registamos todas as dependências e View Models. Este loc container gere todas as dependências e injeta cada uma nas view models. O ViewModelLocator contém uma propriedade para cada view model registrado, sendo esta propriedade usada no binding com o a propriedade DataContext da página.

**Nota:** A razão pela qual se verifica se cada dependência está registrada, tem a ver com erros em modo de desenho no Visual Studio:

```
if (!SimpleIoc.Default.IsRegistered<...>())
{
    ...
}
```

Sem isto, cada dependência será registrada duas vezes o que lança uma Exception.

O ViewModelLocator será adicionado ao App.xaml, que veremos de seguida.

### App.xaml

O App é uma classe criada quando a aplicação inicia.

Em XAML, é possível adicionar dicionários de resources ou mesmos simples objetos. Para o nosso exemplo, iremos adicionar o objeto ViewModelLocator que será estático.

```
<Application x:Class=
    "NokiaDev.AboutPageSample.App"
xmlns="http://schemas.microsoft.com/winfx/2006/
    xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/
        winfx/2006/xaml"
    xmlns:shell="clr-namespace:Microsoft.
        Phone.Shell;assembly=Microsoft.Phone"
    xmlns:viewModel="clr-namespace:NokiaDev.
        AboutPageSample.ViewModels">

<!--Application Resources-->
```

```
<Application.Resources>
  <viewModel:ViewModelLocator x:Key="Locator" />
</Application.Resources>

<Application.ApplicationLifetimeObjects>
  <!--Required object that handles
  lifetime events for the application-->
  <shell:PhoneApplicationService
    Launching="ApplicationLaunching" Closing=
    "ApplicationClosing"
    Activated="ApplicationActivated"
    Deactivated="ApplicationDeactivated"/>
</Application.ApplicationLifetimeObjects>
</Application>
```

### Benefícios do padrão MVVM

Este tipo de implementação é muito importante, destacando-se os seguintes pontos:

- A implementação do view model é independente da implementação de cada serviço (ApplicationManifestService, NavigationService, etc). Apenas temos de respeitar a implementação da interface, podendo o comportamento ser alterável, e sem afetar o resto do código.
- Desta forma podemos implementar testes unitários para cada view model, usando serviços falsos.
- O índice de manutenção é maior.
- Os conceitos, funcionalidades ficam separados da e obtemos um código mais limpo.
- Cada vez que a interface com o utilizador é alterado, apenas é preciso alterar o XAML, mas todo o comportamento se mantém (supondo claro que as funcionalidades se mantêm).

Cada toolkit em causa é utilizado por várias pessoas, permitindo que sejam corrigidos erros e permite um rápido desenvolvimento. Se por alguma razão quisermos remover estas dependências, apenas temos que implementar as nossas interfaces e implementações. Não implicando isto que a aplicação seja toda reescrita.

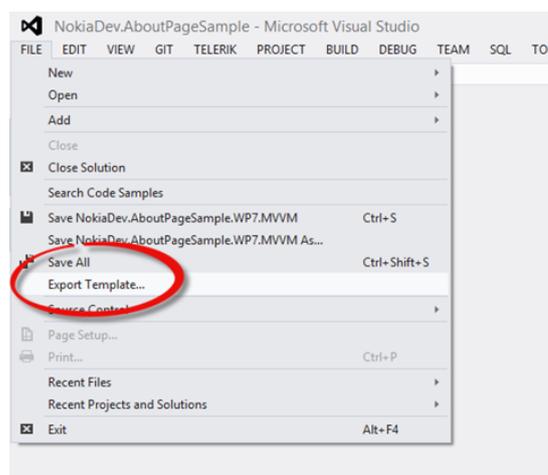
### Reutilizando a página Acerca de

Para cada aplicação que desenvolvemos, precisamos de ter um página "Acerca de", que de uma forma geral são iguais. Para simplificar o nosso desenvolvimento, devemos reutilizar este trabalho através da

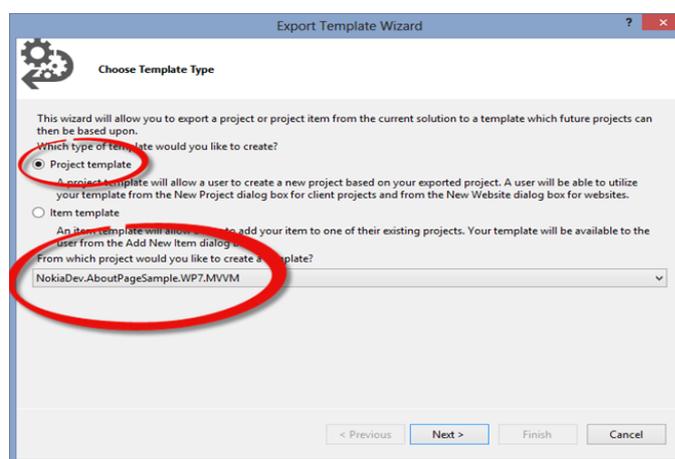
1. Criação de *item templates* para **AboutPage.xaml** e para **AboutViewModel.cs**
2. Criação de *project template* para o projeto, permitindo a escolha da *target* pretendida.

Por causa do nosso projeto ter dependências de referências externas (Cimbalino Windows Phone Toolkit, MVVM Light e de imagens contidas na pasta de imagens, a segunda opção (criar o *project template*) é a melhor opção. Esta solução será discutida de seguida.

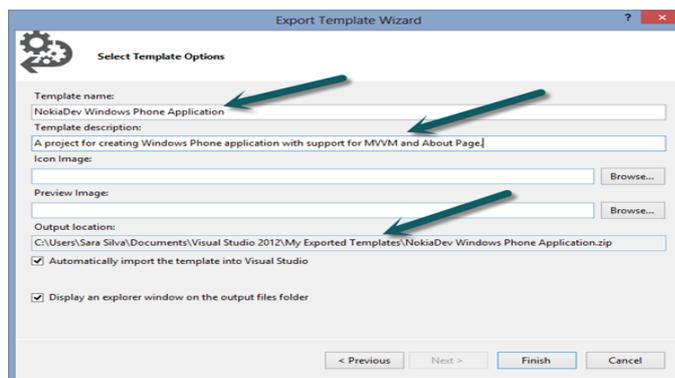
### Exportação do Template



Selecionando a opção "Export Template".



"Export Template Wizard" - Selecionando o Project Template para o projeto "NokiaDev.AboutPageSample.WP7.MVVM"

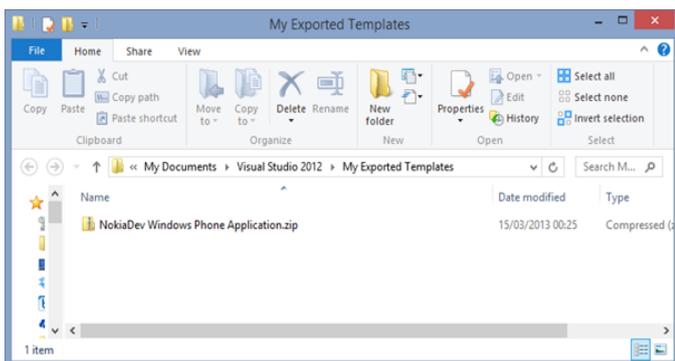


Definindo os detalhes do novo Template.

# COMUNIDADE NETPONTO

<http://netponto.org>

## PÁGINA AVANÇADA "SOBRE" PARA APLICAÇÕES DE WINDOWS PHONE



O \*.zip resultante da exportação

### Edição do template exportado

Depois de fazer "unzip" o ficheiro resultante, encontramos alguns ficheiros, \*.cs and \*.xaml comuns para nós, os quais reconhecemos. Mas existe um ficheiro importante com a extensão .vstemplate (normalmente renomeado para MyTemplate.vstemplate), o qual contém as definições para o projeto quando este é criado. De seguida podemos ver a versão inicial do ficheiro, com definições genéricas que serão alteradas na criação do projeto.

```
<VSTemplate Version="3.0.0" xmlns="http://schemas.microsoft.com/developer/vstemplate/2005" Type="Project">
  <TemplateData>
    <Name>NokiaDev Windows Phone Application </Name>
    <Description>A project for creating Windows Phone application with support for MVVM and About Page.</Description>
    <ProjectType>CSharp</ProjectType>
    <ProjectSubType></ProjectSubType>
    <SortOrder>1000</SortOrder>
    <CreateNewFolder>true</CreateNewFolder>
    <DefaultName>NokiaDev Windows Phone Application </DefaultName>
    <ProvideDefaultName>true</ProvideDefaultName>
    <LocationField>Enabled</LocationField>
    <EnableLocationBrowseButton>true </EnableLocationBrowseButton>
    <Icon>__TemplateIcon.ico</Icon>
  </TemplateData>
  <TemplateContent>
    <Project TargetFileName="NokiaDev.AboutPageSample.csproj"
      File="NokiaDev.AboutPageSample.csproj"
      ReplaceParameters="true">
      <ProjectItem ReplaceParameters="true" TargetFileName="App.xaml">App.xaml</ProjectItem>
      <ProjectItem ReplaceParameters="true" TargetFileName="App.xaml.cs">App.xaml.cs</ProjectItem>
      <ProjectItem ReplaceParameters="false" TargetFileName="ApplicationIcon.png">ApplicationIcon.png</ProjectItem>
      <ProjectItem ReplaceParameters="false" TargetFileName="Background.png">Background.png</ProjectItem>
    </Folder Name="Images" TargetFolderName="Images">
      <ProjectItem ReplaceParameters="false" TargetFileName="appbar.about.png">appbar.about.png</ProjectItem>
      <ProjectItem ReplaceParameters="false" TargetFileName="appbar.email.png">appbar.email.png</ProjectItem>
    </Folder>
  </TemplateContent>
</VSTemplate>
```

```
appbar.email.png</ProjectItem>
  <ProjectItem ReplaceParameters="false" TargetFileName="appbar.rate.png">appbar.rate.png</ProjectItem>
  <ProjectItem ReplaceParameters="false" TargetFileName="appbar.reply.email.png">appbar.reply.email.png</ProjectItem>
  <ProjectItem ReplaceParameters="false" TargetFileName="appbar.share.png">appbar.share.png</ProjectItem>
  <ProjectItem ReplaceParameters="false" TargetFileName="appbar.share.settings.png">appbar.share.settings.png</ProjectItem>
</Folder>
<ProjectItem ReplaceParameters="true" TargetFileName="packages.config">packages.config</ProjectItem>
<Folder Name="Properties" TargetFolderName="Properties">
  <ProjectItem ReplaceParameters="true" TargetFileName="AppManifest.xml">AppManifest.xml</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="AssemblyInfo.cs">AssemblyInfo.cs</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="WMAppManifest.xml">WMAppManifest.xml</ProjectItem>
</Folder>
<ProjectItem ReplaceParameters="true" TargetFileName="Read-Me.txt">Read-Me.txt</ProjectItem>
<ProjectItem ReplaceParameters="false" TargetFileName="Settings.StyleCop">Settings.StyleCop</ProjectItem>
<ProjectItem ReplaceParameters="false" TargetFileName="SplashScreenImage.jpg">SplashScreenImage.jpg</ProjectItem>
<Folder Name="ViewModels" TargetFolderName="ViewModels">
  <ProjectItem ReplaceParameters="true" TargetFileName="AboutViewModel.cs">AboutViewModel.cs</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="HomeViewModel.cs">HomeViewModel.cs</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="ViewModelLocator.cs">ViewModelLocator.cs</ProjectItem>
</Folder>
<Folder Name="Views" TargetFolderName="Views">
  <ProjectItem ReplaceParameters="true" TargetFileName="AboutPage.xaml">AboutPage.xaml</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="AboutPage.xaml.cs">AboutPage.xaml.cs</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="HomePage.xaml">HomePage.xaml</ProjectItem>
  <ProjectItem ReplaceParameters="true" TargetFileName="HomePage.xaml.cs">HomePage.xaml.cs</ProjectItem>
</Folder>
</Project>
</TemplateContent>
</VSTemplate>
```

**Nota:** **TemplateData** contém todas as definições do template e **TemplateContent** contém todas as pastas e ficheiros que serão adicionados ao projeto na sua criação.

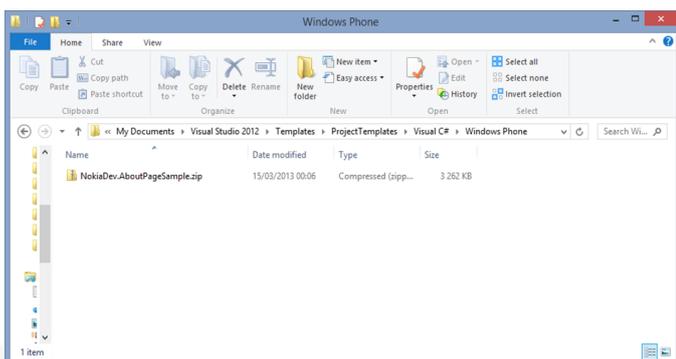
## PÁGINA AVANÇADA "SOBRE" PARA APLICAÇÕES DE WINDOWS PHONE

Name	Type	Compressed size	Password ...	Size	Ratio
Images	File folder				
Properties	File folder				
ViewModels	File folder				
Views	File folder				
_TemplateIcon.ico	ICO File	1 KB	No	10 KB	93%
App.xaml	Windows Markup File	1 KB	No	1 KB	60%
App.xaml.cs	CS File	2 KB	No	8 KB	79%
ApplicationIcon.png	PNG File	2 KB	No	2 KB	0%
Background.png	PNG File	4 KB	No	4 KB	5%
Cimbalino.Phone.Toolkit.2.1.nupkg	NUPKG File	301 KB	No	305 KB	2%
Microsoft.Bcl.1.0.16-rc.nupkg	NUPKG File	690 KB	No	755 KB	9%
MyTemplate.vstemplate	Visual Studio Project/Item...	2 KB	No	5 KB	78%
NokiaDev.AboutPageSample.csproj	Visual C# Project file	2 KB	No	8 KB	74%
packages.config	CONFIG File	1 KB	No	1 KB	49%
Portable.CommonServiceLocator.1.2.2.nupkg	NUPKG File	41 KB	No	45 KB	11%
Portable.MvvmLightLibs.4.1.27.1.nupkg	NUPKG File	48 KB	No	171 KB	73%
Read-Me.txt	Text Document	1 KB	No	1 KB	36%
Settings.StyleCop	STYLECOP File	1 KB	No	2 KB	78%
SplashScreenImage.jpg	JPG File	4 KB	No	10 KB	66%
Unofficial.Blend.Interactivity.1.0.0.nupkg	NUPKG File	2 154 KB	No	2 183 KB	2%

**Tip:** É preciso adicionar os ficheiros nupkg (Cimbalino.Phone.Toolkit.2.1.nupkg, Portable.MvvmLightLibs.4.1.27.1.nupkg, Unofficial.Blend.Interactivity.1.0.0.nupkg e Portable.CommonServiceLocator.1.2.2.nupkg) que serão usados na instalação dos NuGet packages usados no projeto. E é preciso adicionar referências a estes ficheiros no final do ficheiro \*.vstemplate.

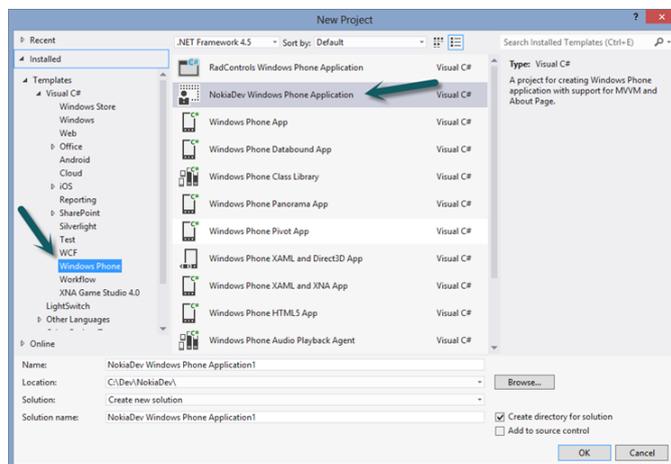
```
<VSTemplate Version="3.0.0" xmlns="http://schemas.microsoft.com/developer/vstemplate/2005" Type="Project">
  <WizardExtension>
    <Assembly>NuGet.VisualStudio.Interop, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</Assembly>
    <FullClassName>NuGet.VisualStudio.TemplateWizard</FullClassName>
  </WizardExtension>
  <WizardData>
    <packages repository="template">
      <package id="Cimbalino.Phone.Toolkit" version="2.1" targetFramework="wp71" />
      <package id="Portable.CommonServiceLocator" version="1.2.2" targetFramework="wp71" />
      <package id="Portable.MvvmLightLibs" version="4.1.27.1" targetFramework="wp71" />
      <package id="Unofficial.Blend.Interactivity" version="1.0.0" targetFramework="wp71" />
    </packages>
  </WizardData>
</VSTemplate>
```

Adição do novo template, para instalação. O template deve ficar junto do templates de Windows Phone, por esta razão foi adicionado a uma pasta com o nome Windows Phone.



**Tip:** O ficheiro \*.vstemplate deve estar contido na raiz da pasta \*zip, senão o template não irá aparecer na lista de templates.

### Instalação do novo template



O template resultante é: NokiaDev - Windows Phone Template.

“ Cada toolkit em causa é utilizado por várias pessoas, permitindo que sejam corridos erros e permite um rápido desenvolvimento. ”

O template foi criado para Windows Phone SO 7.1 e pode ser atualizado para Windows Phone SO 8.0. A boa prática sobre isto, mas fora do âmbito do artigo, será criar um IWizard que irá apresentar uma interface para selecionar a target do projeto aquando da sua criação.

**Tip:** Item e Project templates podem ser encontrados nas seguintes pastas. Note-se que alterações nestas pastas podem causar problemas ao executar o Visual Studio.

- Item Templates podem ser encontrados em "C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE\ItemTemplates"

# COMUNIDADE NETPONTO

<http://netponto.org>

## PÁGINA AVANÇADA “SOBRE” PARA APLICAÇÕES DE WINDOWS PHONE

- Project Template podem ser encontrados em "C:\Program Files (x86)\Microsoft Visual Studio 11.0\IDE\ProjectTemplates\"

Os nossos **Item Templates** ou **Project Template** devem ser adicionados em C:\Users\{USERNAME}\Documents\Visual Studio 2012\Templates

### Referências:

- [\(MSDN\) Creating Project and Item Templates \(\\*\)](#)
- [\(MSDN\) Templates for Creating Templates \(\\*\)](#)
- [\(MSDN\) Packages in Visual Studio Templates \(\\*\)](#)
- [\(MSDN\) How to: Use Wizards with Project Templates \(\\*\)](#)

**Nota:** (\*) Artigos disponível, apenas, em Inglêss.

Em conclusão, uma página Sobre mantém o utilizador informado sobre os detalhes mais importantes da aplicação. E permite adicionar funcionalidades como a avaliação, feedback e partilha, que aumentam a qualidade da mesma, podendo ser um bom meio de marketing.

“ **O XAML define toda a interface com o utilizador para a Home-Page.** ”

A opinião do utilizador é o melhor feedback que podemos ter para podermos ter uma aplicação de sucesso, por esta razão é recomendável que cada aplicação tenha uma.

Para garantir um desenvolvimento rápido em tarefas, que de uma forma geral são, repetidas devemos reutilizar o código criado.

“ **A opinião do utilizador é o melhor feedback que podemos ter para podermos ter uma aplicação de sucesso (...)** ”

Este artigo foi originalmente escrito para a comunidade [Nokia Developer](#), mais especificamente para a [Wiki](#):



[Página avançada Sobre para aplicações de Windows Phone](#)

## AUTOR



Escrito Por Sara Silva

é 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](http://www.saramgsilva.com) e o twitter é [@saramgsilva](https://twitter.com/saramgsilva)

# No Code

**Raspberry Pi - Pequeno mas grande**

**SEO - Keywords Parte 1**

**Crianças Digitais**

## Raspberry Pi - Pequeno mas Grande!

Neste artigo vamos, caro leitor, falar do pequeno computador que está a entrar na moda. O Raspberry Pi.

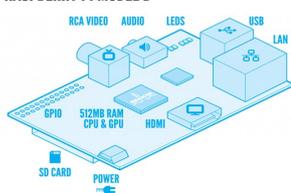
Desenvolvido pela Fundação Raspberry Pi e pela Universidade de Cambridge, no Reino Unido em que o principal objetivo é estimular o ensino da programação e da ciência nas escolas, este pequeno computador foi disponibilizado para venda ao público em 2012.

Disponível em dois modelos, a diferença principal é que o modelo B tem duas entradas USB e tem um controlador Ethernet (RJ45) enquanto que, o modelo A não tem nenhum controlador ethernet e apenas possui uma entrada USB.

No meu caso adquiri o Raspberry Pi – Modelo B Revisão 2.0 512 MB com as seguintes características:

- SoC Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
- CPU: 700 MHz ARM1176JZF-S core (ARM11 family)
- GPU: Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
- Memória (SDRAM): 512 Megabytes (MiB)
- Video outputs: Composite RCA, HDMI
- Audio outputs: 3.5 mm jack, HDMI
- Armazenamento: SD, MMC, SDIO card slot
- Controlador de Rede: 10/100 Ethernet RJ45
- Dimensões: 85,60 milímetros x 56 milímetros x 21 milímetros
- Peso: Cerca de 45 gramas

RASPBERRY PI MODEL B



É verdade que o Raspberry Pi custa pouco mais de 30€ mas depois de comprar o cabo HDMI, o cabo da fonte de alimentação, e caixa de proteção e mais um ou outro acessório como o adaptador de wireless e o cartão SD para poder instalar o Raspbian (sistema operativo), a conta final aumenta um bocadinho.

O segredo que faz com que este minicomputador seja bastante poderoso é o Broadcom BCM2835, cujo sistema System-on-chip que contém um ARM1176JZFS com 700Mhz, e uma Broadcom VideoCore IV, que faz com que o Raspberry Pi se desenrasque muito bem desempenhando o papel de central multimédia.

Se olharmos para o processador que vem na pequena placa, o ARM1176JZFS, podemos verificar que este processador faz parte da família ARM11, que utiliza um pipeline de 8 estágios.

Pipeline é uma técnica de hardware que permite que ao processador procurar uma ou mais instruções além da próxima a ser executada sendo que estas instruções são colocadas numa fila de espera dentro do processador (CPU) onde aguardam o momento de serem executadas.

Voltando atrás uma década, e recordando um dos computadores mais famosos do início dos anos noventa, o 486... para melhorar o desempenho do 486, na altura a Intel resolveu usar o pipeline, uma técnica inicialmente usada em processadores RISC, que consistia em dividir o processador em vários estágios distintos. O 486, possuía um pipeline de 5 níveis, ou seja, é dividido em 5 estágios. Ora o Raspberry Pi utiliza um pipeline de 8 estágios.

Além de ter um custo relativamente baixo e de ter um bom desempenho, outra das melhorias da microarquitetura ARM11 é a redução do consumo energético, tendo um menor risco de superaquecimento. Para os leitores mais curiosos é possível ter acesso à documentação do processador aqui: [goo.gl/k4htU](http://goo.gl/k4htU)

Um dos pontos considerados mais fracos deste hardware, é que o mesmo não vem de origem com um RTC, ou seja, com o real time clock (relógio de tempo real), o que pode ser introduzido manualmente no arranque do Raspberry Pi, contudo o modelo B, quando ligado à internet consegue ele próprio autoconfigurar-se.

Na página oficial do projeto <http://www.raspberrypi.org/>, podemos fazer o download do sistema operativo que preferirmos utilizar. É bastante simples conseguir formatar o cartão SD e o próprio site ensina passo a passo como fazer.

Apesar de haver vários sistemas operativos disponíveis (o RISC OS por exemplo, e está neste momento em desenvolvimento uma versão do Sistema Operativo Android especialmente para o Raspberry Pi), ainda só experimentei o Raspbian “wheezy” e sinceramente fiquei agradavelmente surpreendida.

Tendo em conta que um dos objetivos principais da

## RASPBERRY PI - PEQUENO MAS GRANDE!

Fundação é que as crianças aprendam desde pequenas a programar, na distribuição recomendada do Raspbian, vem já instalado o laboratório Stratch (uma linguagem de programação criada no MIT para crianças com uma interface gráfica que permite aos mais pequenos fazer os seus próprios programas e jogos através de instruções que funcionam como blocos de lego).

Este projeto tem já disponível a Pi Store, que é facilmente instalada no Raspberry Pi onde podemos fazer o download de diversos jogos e aplicações disponíveis para o mesmo. Entre outras coisas, algo já disponível gratuitamente, na Pi Store, é o CodeBlocks e o LibreOffice por exemplo. À parte disso, o Raspbian que é baseado no sistema operativo Debian/Linux, e consegue fazer praticamente tudo (senão conseguir fazer tudo mesmo) a que estamos habituados em Linux.

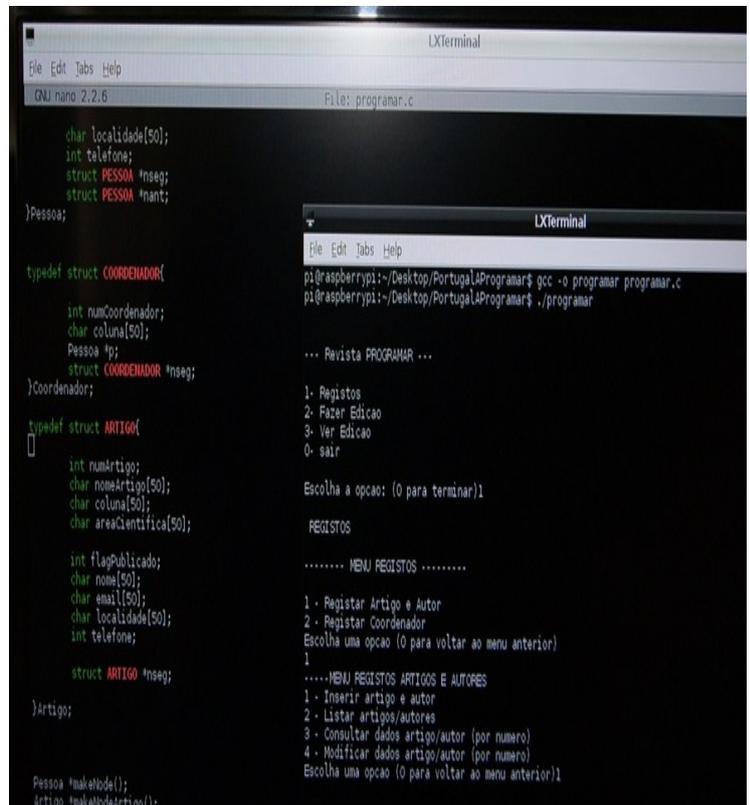
O código em Linguagem C do artigo “Estruturas vs Objetos” foi desenvolvido no Raspberry Pi e fiquei agradavelmente surpreendida com o seu desempenho. Para enquadrar bem o leitor, eu, como todas as mulheres, eu acho que devo ter meio mundo na carteira porque há sempre qualquer coisa que me pode fazer falta. E algo que raramente falta na minha carteira é sem dúvida o meu fiel amigo, o netbook.

E como gosto de ter sempre tudo disponível acho que um netbook com 1GB de RAM pode e deve comportar tudo e mais alguma coisa, claramente todos sabemos que isso às vezes não corre bem... Atualmente tenho instalado o Windows e o Ubuntu no netbook sendo que por norma em C, programo sempre no netbook em Ubuntu. Desta vez como referi há pouco, o código em C foi desenvolvido no Raspberry Pi e “ele não se queixou”, pelo contrário, este acima das minhas expectativas e neste caso concreto, igualou o desempenho do netbook.

Apesar de o python ser a linguagem que o Raspberry Pi promove, sendo que tem inclusive um IDLE para essa linguagem, contudo, todas as linguagens que o ARMv6 compila, podem também ser usadas neste minicomputador. A linguagem C é uma delas, como pode ser visto na imagem [img.1]

Ainda a nível de curiosidade, posso dizer ao leitor, que apesar de o Raspberry Pi trazer de origem as duas entradas USB, pessoalmente achei que ainda precisava de mais visto que as duas entradas seriam apenas para o rato e para o teclado. Como tinha adquirido também o adaptador wireless e queria também usar Pen's... Adquiri então adaptador de forma a ter mais portas USB disponíveis, e à boa moda do

Linux, o Raspbian autoinstalou o adaptador e não tive qualquer problema. Com os drivers certos, o wireless funcionou também às mil maravilhas e foi apenas preciso introduzir a password da rede.



```
GNU nano 2.2.6 File: programar.c
char localidade[50];
int telefone;
struct PESSOA *nseg;
struct PESSOA *nalt;
}Pessoa;

typedef struct COORDENADOR{
int numCoordenador;
char coluna[50];
Pessoa *p;
struct COORDENADOR *nseg;
}Coordenador;

typedef struct ARTIGO{
int numArtigo;
char nomeArtigo[50];
char coluna[50];
char areaCientifica[50];

int flagPublicado;
char nome[50];
char email[50];
char localidade[50];
int telefone;

struct ARTIGO *nseg;
}Artigo;

Pessoa *makeNode();
Artigo *makeNodeArtigo();

pi@raspberrypi:~/Desktop/Portugal/Programar$ gcc -o programar programar.c
pi@raspberrypi:~/Desktop/Portugal/Programar$ ./programar

... Revista PROGRAMAR ...

1- Registos
2- Fazer Edicao
3- Ver Edicao
0- sair

Escolha a opcao: (0 para terminar)1

REGISTOS

..... MENU REGISTOS .....

1 - Registrar Artigo e Autor
2 - Registrar Coordenador
Escolha uma opcao (0 para voltar ao menu anterior)
1

.....MENU REGISTOS ARTIGOS E AUTORES
1 - Inserir artigo e autor
2 - Listar artigos/autores
3 - Consultar dados artigo/autor (por numero)
4 - Modificar dados artigo/autor (por numero)
Escolha uma opcao (0 para voltar ao menu anterior)1
```

[img.1]

Devo ainda alertar/recordar o leitor que o Raspberry Pi não vem com o driver de som pré-configurado (o que, inicialmente, me fez pensar que não ia ter som no minicomputador), mas que após uma breve pesquisa, descobri no site oficial do Raspberry Pi qual a linha de comandos mágica que colocou o minicomputador a “falar”.

À falta de um monitor cá em casa, o monitor usado foi realmente a televisão da sala através do cabo HDMI.

Claro que o Raspberry Pi não substitui um computador, mas desenrasca-se muito bem e em termos de qualidade/preço, é uma máquina surpreendente.

Além da página oficial da Fundação (<http://www.raspberrypi.org/>), está disponível também a revista oficial, online e gratuita (<http://www.themagpi.com/>) onde o leitor pode adquirir um conhecimento mais profundo das potencialidades do Raspberry Pi.

## AUTOR



Escrito por Rita Peres

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

## SEO – Keywords Parte I

Bem vindos a mais um artigo sobre Search Engine Optimization, ou a nobre arte de atrair mais tráfego para um site, oriundo dos motores de busca. Uma vez mais não posso deixar de recomendar a leitura dos quatro primeiros artigos sobre este tema, disponíveis nas edições anteriores da Revista Programar. Existem vários termos técnicos (por exemplo os SERPs, Ranking, etc.) que já foram anteriormente explicados, e que irei abordando neste e em futuros artigos, não podendo ocupar estas preciosas linhas que me são concedidas, com explicações de certa forma repetitivas. Compreendam os leitores recém-chegados ao tema do SEO que esta série de artigos vai evoluindo ao longo das suas publicações, consolidando novos conhecimentos em cima dos que vão sendo divulgados anteriormente, sendo o processo de aprendizagem gradual. Neste quinto artigo irei centrar as atenções nas tão famosas Keywords, ou na nossa língua, as Palavras-Chave. O que são as Keywords? Para que servem e porque são tão importantes, ao ponto de levar muitos a tentar ludibriar os motores de busca, mediante práticas proibidas como o keyword stuffing, por exemplo. Este artigo sobre palavras-chave irá ser dividido em duas partes, face à sua importância e complexidade. Na primeira irei focar mais o aspecto teórico, explicando o conceito de keyword, bem como qual a sua importância enquanto factor ranking, deixando espaço para a vertente mais prática na segunda parte, onde irei tentar explicar algumas das técnicas e ferramentas de pesquisa das keywords mais adequadas e vantajosas para determinada página.

### Factores de Ranking Internos

Os factores internos que conferem directamente e indirectamente impacto sobre o *Ranking* de um portal, são aspectos que nós podemos controlar e manipular. Eles irão ser as nossas principais ferramentas no processo de optimização SEO de um site, uma vez que no que diz respeito aos externos, algo a que dedicarei outro artigo, nem sempre podemos influir sobre os mesmos. Basicamente existem dois tipos de factores de *ranking* internos. O primeiro diz respeito aos factores tidos como **On-site**, ou seja, aquilo que podemos fazer ao nosso site, como um todo, e os **On-page**, tudo o que podemos fazer ao nível da optimização de uma determinada página em particular, onde se enquadra também a atribuição de *keywords*.

### Factores de Ranking On-Page

As palavras-chave possuem um papel preponderante neste tipo de factores. A preocupação com a sua atribuição inicia-se dentro da própria TAG de título <TITLE></TITLE> do código HTML. E porquê esta importância? Irão perceber

melhor no seguinte exemplo. Imaginem que tinham ido ao motor de busca Google procurar pelos meus artigos sobre SEO na Revista Programar. No campo de pesquisa escreviam: “Artigo SEO Revista Programar”.



Analisando os resultados na página de SERPs, podemos ver por exemplo no segundo resultado, o texto que surge a azul e em destaque, é geralmente aquele que nos leva a clicar e abrir a página respectiva. Os motores de busca dão especial destaque a esta frase e a mesma é nada mais nada menos que o conteúdo do *title* da página indexada. Percebe-se então a importância do texto que utilizamos nos títulos das nossas páginas. Quanto mais “apelativo” ele for para quem pesquisa, ou seja, quanto mais semelhante for com as palavras empregues pelo utilizador na pesquisa, mais hipóteses terá a página de ser visitada. E como o fazer? Enriquecendo os títulos com palavras-chave! Quando clicamos no *link* e nos deslocamos para a respectiva página, podemos verificar que o seu título “Revista Programar – SEO” é gerado pelo código:

```
<script type="text/javascript"> document.title = "Revista PROGRAMAR – SEO ";</script>
```

Poderia ter sido simplesmente colocado dentro da TAG <TITLE> como no exemplo seguinte, contudo tudo isto depende da forma como as páginas foram desenvolvidas pelos seus programadores.

```
<title>Revista PROGRAMAR – SEO</title>
```

**Nota:** A TAG <TITLE> fica situada dentro da TAG <HEAD> do documento HTML.

Se ainda não conheciam a importância de atribuir títulos a páginas *web*, podem agora facilmente perceber que este é um dos factores *On-page* mais importantes, no que diz respeito ao processo de optimização SEO. Uma vez que conseguimos atrair visitantes oriundos dos motores de busca de uma forma mais directa, pois podemos editar, manipular, agir, sobre o texto dos títulos, e este é aquele para o qual o utilizador mais atenção dedica quando analisa os SERPs. Contudo o nosso trabalho não se fica por aqui. Neste exemplo a página clicada estava em segundo lugar no SERP,

mas para ela lá chegar a esse nível tão alto, são necessárias outras técnicas, não convém esquecer! Também convém não descurar o facto de que o *World Wide Consortium*(W3C) recomenda que os *webdesigners* utilizem a TAG TITLE e atribuam títulos às suas páginas, de forma a identificar o seu conteúdo da melhor forma aos cibernautas. De igual forma e como já demonstrei, os motores de busca utilizam também esta descrição nos títulos dos próprios resultados da pesquisa.

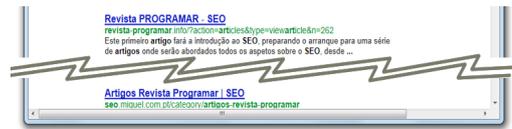
**E qual o número máximo de palavras que podemos escrever num título? Existe um limite?** Aqui os vários especialistas em SEO dividem-se. Uns defendem que não existe um limite teórico quanto ao número de caracteres a colocar num título, enquanto outros defendem títulos não superiores a 70 caracteres. Seguindo as *guidelines* do W3C (<http://www.w3.org/Provider/Style/TITLE.html>) os títulos não deverão exceder os 64 caracteres, sendo que um número situado entre os 60 e os 70 será o ideal. No caso da Google, embora seja permitido colocar mais do que 70 caracteres, a partir desse número (aproximadamente) os títulos começam a ser truncados, sendo que a Google ocultará determinadas palavras dos títulos das pesquisas, o que pode ser problemático para nós. Testes realizados por várias empresas dedicadas ao SEO, também comprovaram que não é apenas o número de caracteres que é tido em conta como limite, mas até os pixels que desenham os mesmos. Contudo e para não complicarmos mais uma questão que se quer simples, mantenham o número 70 como vossa baliza.

### Keywords no título e no conteúdo!

Outra regra básica na utilização de keywords nas páginas, é a que devemos utilizar as palavras-chave que enriquecem os nossos títulos, no próprio conteúdo das páginas (texto). Nada indicará melhor aos motores de busca que as nossas páginas contêm conteúdo relevante para determinada pesquisa, que ele detectar as palavras que utilizamos no TITLE, ao longo do texto da respectiva página. Não podemos cair no erro de tentar enganar os algoritmos dos indexadores, pois estes foram evoluindo no sentido de melhor servir os pesquisadores.

### Keywords nos URLs das páginas

Reparem no exemplo em cima, onde se encontra o segundo resultado da minha pesquisa (a ordem dos resultados nos SERPs pode variar de pesquisa para pesquisa e de utilizador para utilizador), em que as palavras a verde representam o URL para a página em questão.



O primeiro resultado desta imagem possui um URL não “amigável”, ou seja, desprovido de palavras-chave. Em contraposição, o segundo resultado desta imagem, que por acaso é um link para o meu bloque sobre o tema SEO (criado recentemente), está enriquecido com keywords, que surgem a bold. Esta poderá ser uma vantagem para subir uns degraus nos rankings. Contudo e por se tratar de um site muito recente, o mesmo só aparece no quinto resultado da minha pesquisa. Este texto a verde pode também ajudar o utilizador no momento de decisão pelo clique. Enquanto que no primeiro resultado o URL nada transmite ao pesquisador, já no segundo resultado o utilizador pode sentir-se mais confiante a clicar, uma vez que no próprio URL encontra palavras chave relativas aos termos que pesquisou (a negrito).

### Keywords na TAG META

As TAG <META> disponibilizam metadados sobre um determinado documento HTML. Estes não são “visíveis” aos visitantes comuns das páginas, mas são-no para os bots que as indexam. Os metadados podem descrever o conteúdo da página (description), reservam também um espaço para colocarmos as próprias keywords relativas à página, e até mesmo quem foi o autor do documento, a data da sua última modificação, etc. Tudo isto deve servir para descrever da melhor forma o conteúdo de uma página e não para enganar o motor de busca. Este utiliza a informação contida nos metadados para indexar da melhor forma as páginas, tentando catalogá-las em função desses dados.

```
<meta name="description" content="Artigo sobre
SEO.."/>
<meta name="keywords" content=
"PageRank,SERPs,robots,crawlers.. " />
```

**Nota:** As META TAGs devem de ser colocadas dentro da TAG <HEAD> de um documento HTML, geralmente antes da TAG de fecho </HEAD>

E por hoje ficamos por aqui no que diz respeito a keywords. Como já perceberam é um dos temas mais sensíveis do SEO e por isso iremos dedicar mais tempo a ele nos próximos artigos sobre Search Engine Optimization, na Revista Programar.

## AUTOR



### Escrito por Miguel Lobato

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

<https://www.facebook.com/MiguelLobatoSEO> - @MiguelLobatoSEO

**Elege o melhor artigo desta edição**

**Revista PROGRAMAR**

[http://tiny.cc/ProgramarED40\\_V](http://tiny.cc/ProgramarED40_V)

## Crianças Digitais

Quantos de nós já não nos cruzámos com imagens de brinquedos e desenhos animados da nossa infância, acompanhados da frase “Se conhecestes isto, estás velho” ou então “Se conhecestes isto tiveste uma infância feliz”?

Estamos na era do Homo-Tecnologicus e cada vez mais vivemos dependentes da tecnologia.

Proponho um exercício: Por um momento imaginemos o nosso dia-a-dia sem algo tão banal como o telemóvel, a internet, o computador, para não falar em tablet's e outros gadgets aos quais nos habituamos. Que tal? Muitos de nós veriam a vida virada do avesso. Outros pensariam, “Finalmente! Férias!!!”.

A verdade é que é inevitável o convívio com a tecnologia nos dias de hoje. Na minha profissão já me foi possível conhecer crianças que, mesmo em idade pré-escolar já dominam o manuseamento de algumas tecnologias melhor que alguns adultos. Uma mãe dizia-me, acerca do seu filho de apenas quatro anos: “Ele nem sabe ler, mas vai ao meu telemóvel e sabe mexer nos jogos todos. No computador conhece já como fazer para ir buscar o que quer. Outro dia disse-me: ó mamã procura aí no Google!”

**“ Por um momento imaginemos o nosso dia-a-dia sem algo tão banal como o telemóvel, a internet, o computador, para não falar em tablet's e outros gadgets aos quais nos habituamos ”**

A nova geração está cada vez mais tecnologicamente dependente. Mas será que está a ser guiada da melhor

forma?

Não estaremos nós na geração do “Copy & Paste”? Que caminho estamos nós a mostrar às gerações futuras?

Está provado cientificamente que a infância é a faixa etária por excelência para a aprendizagem. É quando o nosso cérebro absorve com mais facilidade o conhecimento e tudo é uma fantástica aventura de descoberta. Mas que uso damos aos cérebros das nossas crianças?

Nem todos os pais têm as mesmas possibilidades económicas e nos dias de hoje, cada vez menos. No entanto, os mais pequenos são invadidos por fatores que, ao invés de fomentarem o crescimento cognitivo e um maior desenvolvimento a nível de aprendizagem, geram comportamentos agressivos e de hiperactividade, sem qualquer tipo de valores morais. Onde estão as brincadeiras simples que tivemos na nossa infância? Onde está a sede de conhecimento que nos fazia devorar livros atrás de livros?

Como diz Stan Lee através do personagem Tio Bem em Homem Aranha, “Grandes poderes trazem grandes responsabilidades”. Estamos nós a ser responsáveis quando pomos a tecnologia na mão das nossas crianças? Esta é uma questão que deve ser alvo de reflexão.

Devemos ser nós a colocar as regras que guiarão devidamente as nossas crianças no caminho certo, de modo a conhecerem os limites e os perigos. Devemos ser presentes na educação e estar atentos ao que se passa e ao que fazem, com o que brincam e o que visualizam. Não devemos temer as birras e amuos, mas sim guiar cuidadosamente ao caminho certo.

Felizmente a educação já começa a dar os primeiros passos nesse sentido. Agora em alguns locais, os mais pequenos têm acesso ao ensino na área das tecnologias, feito de uma forma regrada e orientada. Mas porque não mais do que isso?

A programação é um mundo acessível, que pode tornar-se um exercício divertido para uma criança. Tornar-se criador de algo ao invés de ser um mero e apático utilizador, constitui uma aventura e um desafio que, ao mesmo tempo que diverte, enriquece cognitivamente e desenvolve outras capacidades como a linguagem, o raciocínio lógico, o pensamento matemático e a criatividade. Existem várias linguagens que não necessitam de compiladores complexos, para que se possa ver algo a funcionar. E nada mais fácil do que começar pelo simples “Olá mundo!” para apresentar um mundo novo e divertido, em oposição às ementas que são impingidas pela comunicação social e publicidade.

Será assim tão difícil ensinar uma criança de hoje como tantas outras que com eu, que um dia já fui criança, aprendi vendo, lendo, experimentando o que vinha nos livros?

Uma geração inteira de programadores, começou a programar num tempo em que os computadores nada mais traziam do que um interpretador de BASIC, sem rato nem ecrã táctil, sem facilitismos ou automatismos, onde até para se poder jogar era preciso saber uma elementar instrução de



programação, que a grande maioria dessas então crianças decorou em muito tenra idade (*Load ""*).

Essa geração, que aprendeu num tempo em que a tecnologia era "primitiva" comparada à actual, teve de "programar" para poder jogar e conseqüentemente no seu processo cognitivo, desenvolveu o interesse pela programação, agora em boa parte são adultos programadores. Teriam seguido essa carreira e desenvolvido essas aptidões, se na idade em que começaram a lidar com a tecnologia não lhes tivesse sido apresentado um interpretador de BASIC que lhes exigia que digitassem aquela "mágica" linha com uma instrução, para que o jogo que tanto queriam jogar fosse executado?

Nos dias de hoje existem linguagens que permitem ter o mesmo "benefício" com um esforço semelhante e por sinal estão disponíveis nas mais diversas plataformas. Basta ver o exemplo de Python, onde uma criança pode aprender o "Olá

Mundo", de forma tão simples como escrever (*Print "Olá Mundo"*) e passar a barreira do mero utilizador consumidor, para uma nova dimensão da experiência com a tecnologia. A do utilizador-criador.

“ Ensina à criança o caminho em que deve andar e ainda quando for velho não se desviará dele. (“Míshlê Shelomoh”) ”

Ao proporcionar esta aprendizagem a todas as gerações, não só se poderia estar a preencher o “gap de gerações” como estar a “ensinar a pescar” uma geração que já não sabe que os peixes não vêm em lata!

Investir no país? Claro... Mas estamos em crise... Não nos esqueçamos de investir nos mais pequenos, pois eles são o amanhã. Não é necessário investir avultadas somas, apenas investir tempo, atenção e conhecimento para a vida, pois este é o melhor investimento que alguém pode fazer.

Nas palavras de Lao-Tsé, “Se deres um peixe a um homem faminto, vais alimentá-lo por um dia. Se o ensinares a pescar, vais alimentá-lo toda a vida.”



## AUTOR



Escrito por Sara Santos

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

## ENTREVISTA A RUI GUIMARÃES

**Revista PROGRAMAR - (RP):** O que caracteriza a ANSOL ?

**Rui Guimarães - (RG):** A ANSOL é a voz de toda uma comunidade portuguesa que se preocupa com a partilha e reutilização de software e as ideias neste contidas. Eu não conseguiria descrever a finalidade da ANSOL de uma maneira melhor do que aquilo que está nos nossos estatutos: “tem como fim a divulgação, promoção, desenvolvimento, investigação e estudo da Informática Livre e das suas repercussões sociais, políticas, filosóficas, culturais, técnicas e científicas”. Assim, a ANSOL faz-se ouvir em vários domínios da nossa sociedade e tem vários projetos nestes domínios em que todos podem participar.

**RP:** A recente adopção de software livre por alguns grupos empresariais, terá um impacto significativo na adopção mais generalizada do uso de software livre ?

**RG:** Sim e Não. O sucesso de um pacote de software Livre mede-se pelo impacto que tem na comunidade a que se destina (não necessariamente por market share). Assim, o facto de alguns grupo empresariais, como o grupo Tranquilidade que, com a utilização de software Livre, viu uma redução global de 42% nos custos em IT, representa casos de estudo que podem levar outros grupos a adotar aqueles sistemas específicos baseados em software Livre. Sim, melhora a percepção de certo software Livre mas não significa necessariamente que o software Livre em geral terá mais aceitação. Não deixa de ser muito positivo, mas não é novidade que há software Livre com qualidade acima de software comercial. Basta olhar para o exemplo dos servidores web: cerca 60% destes são baseados em software Livre. A adopção mais generalizada do uso de software livre acontecerá através de mudanças “culturais” da respetiva classe profissional...

**RP:** Considera verdade ou mito, o facto de o software livre não ser economicamente proveitoso para quem o produz ?

**RG:** Mito! Ainda hoje assisti a uma palestra dos Manufatura Independente que mostravam projetos comissionados por clientes que acabariam no domínio publico. Há também vários modelos de negócio à volta do software Livre. Que o digam gigantes como a Novell, a Red Hat ou a Oracle. Suponho que este mito terá surgido porque o software Livre não permite o conhecido “vendor lock-in”. Uma das estratégias mais utilizadas pela indústria, e muito lucrativa,



mas que se baseia em obrigar um cliente a utilizar toda uma gama de produtos, sucessivas versões dos mesmos e serviços do mesmo fornecedor devido ao receio de incompatibilidades e altos custos de transição para outros fornecedores. Isto deturpa o espírito da livre competição e cria monopólios. Algo que se torna ainda mais alarmante quando o próprio Estado é um dos clientes e todo o negócio acaba por ficar muito caro aos contribuintes. Os modelos de negócio sem “vendor lock-in” são mais justos e transparentes e é, também por isso, importante que a classe profissional aja no

sentido da sua adopção no Estado para evitar graves desperdícios.

**RP:** Qual acha que é o maior entrave à adopção de software livre em Portugal ?

**RG:** Na utilização pública o maior entrave é, apesar de tudo, “cultural”. Terá havido muitos argumentos contra o uso de software Livre em ambientes mais exigentes mas também, ao mesmo tempo, vários exemplos tem demonstrado que essas criticas são infundadas. Torna-se então claro, uma vez que as criticas foram refutadas, que o maior entrave é a falta de preocupação pelo bem comum. O software Livre não tem departamento de marketing nem promove interesses comerciais junto das entidades governamentais. A sua utilização na Administração Pública tem que ser promovida e exigida pela respetiva classe profissional e por uma população bem informada. A razão pela qual isso não acontece é mesma pela qual o maior partido político português é a abstenção. Eu estou confiante que isso está mudar e que o debate em torno da utilização de software Livre, começa cada vez mais, a ser mais esclarecedor. Prova disso é a recente legislação sobre normas abertas que promove a utilização de formatos abertos na Administração Pública.

No ambiente empresarial, o maior obstáculo será, talvez, a falta de informação sobre a gama de serviços e produtos com base em software livre em oferta pelas empresas na área. Penso que isto mudará assim que as empresas perceberem os custos que acarretam os negócios com “vendor lock-in” e comecem a procurar alternativas, tal como aconteceu no caso do Grupo Tranquilidade que já mencionei.

**RP:** Na sua opinião que impacto teria o uso massificado de software livre nas escolas e universidades ?

# No Code

## ENTREVISTA A MIGUEL GONÇALVES

**RG:** Traria, acima de tudo, uma revolução cultural na área. Todos estariam a utilizar e a trabalhar para algo que a todos pertence o que é um conceito extremamente importante para uma cidadania ativa. Depois, o estudo livre do material em causa, a liberdade para todos o explorarem e com ele experimentarem tornaria os estudantes muito mais bem formados. Reparem que ensinar a utilizar certo software sem que se ensine, pelo menos aos mais interessados, como este é construído é quase equivalente a dar peixe sem que nunca se ensine a pescar.

Uma das estratégias definida pelas grandes multinacionais que estabelecem acordos com o governos a preços mais baixos para licenças educativas é exatamente não deixar espaço para que se ensinem alternativas cimentando assim o seu monopólio.

**RP:** Considera que seja viável para as instituições publicas substituírem o software proprietário que representa a maior fatia do software, por software livre ?

**RG:** Sim, para a esmagadora maioria do software em causa. Falta saber se a resposta é positiva para todo o software, mesmo para o qual ainda não existe alternativa já construída e para o qual seria necessário, portanto, um maior investimento inicial. A esta conclusão já chegou o Reino Unido cujo manual para os serviços digitais do governo, publicado recentemente, declara explicitamente que se deve dar “preferência a software open source, em particular para sistemas operativos, software de rede, servidores web, bases de dados e linguagens de programação”. Este manual diz ainda que software proprietário deve ser usado em raras circunstancias e que “cuidados devem ser tomados para minimizar o risco de lock-in”.

É imperativo, no mínimo, que em Portugal se façam projetos piloto neste sentido para se conhecer a real dimensão da diferença no peso das despesas à volta de software. Para o software mais comum, como ferramentas de edição de documentos e ferramentas de acesso à internet não há dúvidas que as alternativas Livres possuem as funcionalidades desejadas tal como é demonstrado em casos de estudos noutros países e em empresas. Resta conhecer o impacto a longo prazo destas mudanças para software Livre, uma vez que, certamente, haveria custos de transição (devido ao “vendor lock-in” já instalado) mas cujo investimento compensaria ao fim de algum tempo.

Pessoalmente, julgo que há também aqui uma questão de principio. Quando o Estado compra um carro esse veículo pertence ao Estado. A entidade compradora é livre de contratar um condutor de qualquer escola, livre de contratar qualquer serviço mecânico para a sua manutenção, livre até para retirar peças e utilizar noutros veículos. Quando o Estado compra software está adquirir uma licença que lhe concede um uso restrito. A manutenção e a formação estará sempre a cargo do fornecedor. Ao mesmo tempo, o Estado

não tem a possibilidade de inspecionar o software para verificar a sua segurança nem para o adaptar às suas necessidades. Se não deixamos que o dinheiro dos contribuintes seja gasto na aquisição de um veículo que imponha estas condições porque o deixamos que seja gasto em software vinculado a estas condições?

O software Livre permite não só a livre contratação de serviços como permite a qualquer um a sugestão de melhorias. Ou seja, permite uma cidadania ativa dos membros de uma classe profissional. Algo, que o próprio Estado, mais que pedir, deveria estimular.

Não tenho dúvidas, e deixo o desafio a qualquer entidade sub tutela governamental, que use software Livre quando necessitar de um pacote software para providenciar um serviço público. Tal é sempre possível fazê-lo sem colocar em risco dados sensíveis. Rapidamente verão as vantagens não só em termos de usabilidade mas também na participação voluntária de interessados em construir algo melhor com base nesse mesmo software.

Um bom exemplo é o site <http://demo.critica.org/> que construiu uma interface muito melhorada da informação livremente disponibilizada de modo arcaico no site do parlamento. O que não se conseguiria fazer se houvesse a colaboração das entidades responsáveis?

**RP:** É comum ler-se e ouvir-se em palestras, que o software livre é mais “perigoso”, pois toda a gente vê o código. Considera que estas afirmações têm algum fundamento sustentável ?

**RG:** Certamente que não e essas afirmações não podem ser de fontes com conhecimentos em segurança informática. Profissionais em segurança informática estão perfeitamente cientes que a segurança por obscuridade não é segurança de todo. Ou seja, não existe segurança só porque o código fonte não é do conhecimento comum. Isto porque um atacante tem possibilidades de descobrir vulnerabilidades quer obtenha ou não o código fonte. A não disponibilização do código fonte apenas torna mais difícil que terceiros testem e corrijam essas mesmas vulnerabilidades. Assim, o próprio Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (NIST) recomenda que a “segurança de um sistema não deverá depender do secretismo da sua implementação ou dos seus componentes”. Para dar exemplos práticos: a revista Wired publicou um artigo em 2004 intitulado “Linux: Fewer Bugs Than Rivals” com um estudo concluindo que o código fonte de Linux apresenta uma média de 0,17 “bugs” por 1000 linhas de código enquanto que software de código fechado apresentava, em geral, cerca de 20 a 30 “bugs” por 1000 linhas. Ou seja, tudo indica, que software com código aberto tem mais possibilidades de ser corrigido e mais rapidamente que software proprietário o que o torna inerentemente mais seguro.

## PROJECTO EM DESTAQUE NA COMUNIDADE P@P: SUECA ONLINE



Sueca Online, como o nome indica, permite jogar sueca, na internet, contra amigos ou desconhecidos.

Sem qualquer tipo de registo, basta colocar um "nickname" ao inicio entre 3 e 10 caracteres, que vai ser usado como o vosso nome durante o jogo, escolher a mesa de jogo (ou criar uma) e estão prontos a jogar!

Mais simples é complicado.

Este projecto desenvolvido por dois estudantes de Eng. Informática no Instituto Politécnico de Beja, muito dedicados à programação, especialmente jogos, que em virtude dessa dedicação decidiu desenvolver este jogo para Android, visto ser uma aplicação relativamente fácil de programar mas que permitiu uma aprendizagem maior da plataforma Android.

### Características:

- Jogue Sueca com os seus amigos!
- Crie a sua própria mesa com um simples toque.
- Jogue com bots enquanto está á espera de jogadores.
- Jogue no sentido do relógio (Brasil) ou contra o sentido do relógio (Portugal).
- Muito mais!

### Download (Google Play):

<https://play.google.com/store/apps/details?id=com.rtt3ch.suecaonline>



# No Code

A PROGRAMAR está disponível no Windows Phone 7.x e 8, com todas as edições e detalhes sobre cada edição.

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

Entre outras funcionalidades conta com possibilidade de pesquisa superficial de artigos e apontadores directos aos PDF.

A pesquisa e consulta de conteúdos suporta execução offline após carregamento inicial de dados base.

Actualização de conteúdos automática.

ATENÇÃO: A aplicação não gere PDFs offline. Apenas aponta. Deverá ser usado o leitor típico do telefone para o efeito.



# Veja também as edições anteriores da Revista PROGRAMAR

39ª Edição - Fevereiro 2013



38ª Edição - Dezembro 2012



37ª Edição - Outubro 2012



36ª Edição - Agosto 2012



35ª Edição - Junho 2012



34ª Edição - Abril 2012



e muito mais em ...  
[www.revista-programar.info](http://www.revista-programar.info)

**DUVIDAS?**

**IDEIAS?**

**AJUDAS?**

**PROJECTOS?**



**portugal-a-programar**  
•org

