

PROGRAMAR

A REVISTA PORTUGUESA DE PROGRAMAÇÃO

Revista nº5 - Novembro de 2006

www.portugal-a-programar.org

Gestão da Informação: Bases de Dados

Saiba tudo sobre a criação e administração de bases de dados

Programação em ASP

Páginas dinâmicas ASP com AJAX.

The next big thing

RFID Applications

Conheça a tecnologia que já revoluciona o mercado da logística e não só.



weSpendMoney

Uma aplicação web destinada a quem quer controlar as suas despesas



Mounting File Systems

O guia completo para montar qualquer tipo de sistema de ficheiros em Linux

índice

- 3 notícias
- 4 tema de capa
- 8 a programar
- 17 tecnologias
- 21 tutorial
- 23 gnu/linux
- 27 projecto
- 30 análises
- 32 internet
- 33 blue screen
- 34 comunidade

equipa PROGRAMAR

administração

Rui Maia
David Pintassilgo

coordenador

Sérgio Santos

coordenador adjunto

Miguel Pais

redacção

Fernando Martins
Patric Figueiredo
João Pereira
Fábio Correia
Daniel Correia
Tiago Salgado
Marcelo Martins
Pedro Sousa
Pedro Caldeira

colaboradores

José Oliveira

contacto

revistaprogramar
@portugal-a-programar.org

website

www.revista-programar.info



Apresentação e divulgação

A última edição foi um marco muito importante para o nosso projecto. Nunca a revista PROGRAMAR tinha sido tão falada, tão divulgada e, principalmente, nunca tinha sido tão lida. Foi também após o lançamento da quarta edição que fizemos a nossa primeira apresentação pública, que teve lugar no BarCampPT 2006. Este evento decorreu no Pólo 2 da Universidade de Coimbra e teve como público profissionais e amadores da nossa área.

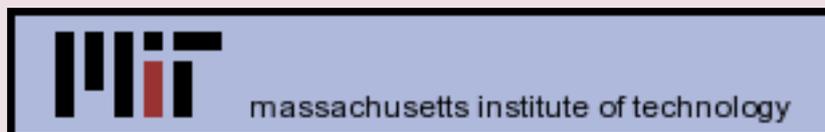
Estes resultados elevam a fasquia da qualidade que nos é exigida por parte dos nossos leitores, qualidade essa que queremos continuar a garantir. É essa, portanto, a razão do atraso de alguns dias desta edição. Pedimos desculpa desde já por este facto, mas consideramo-lo compreensível, visto que ainda somos uma equipa de amadores, que trabalha neste projecto por satisfação pessoal.

Nesta edição podem encontrar a nova secção *Tecnologias*, dedicada a artigos que, embora não estejam directamente relacionadas com programação, são importantes para o desenvolvimento de algumas aplicações. Esta começa com um artigo sobre RFID, um tema cada vez mais discutido actualmente. Iremos também apresentar em breve um novo portal para a revista, mais dinâmico e com mais informações sobre o projecto, desenvolvido por membros da nossa equipa.

Sérgio Santos

MIT financia institutos portugueses

O acordo do Governo português com o Massachusetts Institute of Technology (MIT) envolve um financiamento público global de 32 milhões de euros às instituições nacionais abrangidas pelo Programa MIT Portugal. O Programa envolve quatro áreas temáticas: Engenharia de Conceção e Sistemas Avançados, Sistemas de Energia, Sistemas de Transportes e Sistemas de Bioengenharia.



O acordo de cooperação com o instituto norte-americano MIT envolve a Escola de Engenharia da Universidade do Minho, a Faculdade de Ciências da universidade de Lisboa, as faculdades de Ciências e Tecnologia da Universidade de Coimbra e da Universidade Nova de Lisboa e a Faculdade de Engenharia da Universidade do Porto. Envolve ainda o Instituto Superior de Economia e Gestão da Universidade Técnica de Lisboa e o Instituto Superior Técnico da Universidade Técnica de Lisboa.

Segundo o texto do acordo, as instituições nacionais, em parceria com o MIT, poderão nos próximos cinco anos contratar professores convidados «de mérito internacional» e investigadores em pós-doutoramento. O programa prevê ainda que os estudantes, investigadores e docentes em instituições portuguesas colaborem com equipas de investigação no MIT.

AMD triplica lucro

A Advanced Micro Devices (AMD) superou as expectativas dos analistas financeiros, lucrando 134 milhões de dólares e facturando 1,33 bilhão de dólares no terceiro trimestre de 2006. A estimativa dos analistas era uma receita de 1,31 bilhões de dólares. As margens operacionais da empresa, contudo caíram, facto que a AMD atribui a vendas menos aceleradas de chips para desktops.

A queda de margem reflete também a disputa com a rival Intel – que reportou queda de 35% no lucro neste trimestre. Para o quarto trimestre, a AMD prevê um crescimento apoiado na fusão com a fabricante canadiano de chips gráficos ATI. O objectivo é o de oferecer produtos integrados ao mercado de computadores móveis e eletrónicos de consumo.

Firefox 2.0 atinge 2 milhões de downloads em 24 horas

O Firefox 2.0 atingiu os 2 milhões de downloads no primeiro dia, com picos de 30 downloads por segundo. Isto significa que o Firefox está em vias de superar os 3 milhões de downloads em 4 dias do IE7. Estes são resultados muito positivos para a equipa de desenvolvimento da Mozilla. No entanto, o número de utilizadores do Internet Explorer 7 deverá aumentar com o lançamento do Windows Vista.



Bases de Dados

As Bases de Dados (BD) são imprescindíveis a qualquer sistema e são por isso presença constante no dia-a-dia de todos quantos fazem desenvolvimento.

Esta realidade advém da necessidade de "memória persistente" que qualquer aplicação tem, por mais simples que seja. Isto é visível e óbvio num complexo sistema bancário, mas é também visível, por vezes menos óbvio, num jogo que guarda as pontuações dos melhores jogadores num simples ficheiro de texto.

Independentemente do suporte usado, a necessidade de guardar e recuperar dados existe e é uma constante em qualquer aplicação.

Âmbito

Neste âmbito, ao longo das próximas edições da Revista Programar vão ser apresentados artigos sobre bases de dados abrangendo um conjunto de temas que ajudam a compreender melhor este mundo, desde a idealização de um modelo de dados até à tarefa de tuning de uma base de dados.

Em paralelo com os artigos haverá uma secção prática que ilustrará e permitirá acompanhar os conhecimentos aqui apresentados. Esta secção prática seguirá a norma SQL-99 para que seja

possível a aplicação dos scripts divulgados independentemente da base de dados em causa. É de referir que é comum cada sistema de base de dados possuir as suas próprias extensões embora, as normas SQL-92 e SQL-99 sejam, grosso modo, implementadas por todos.

Objectivo

Uma BD tem como objectivo último guardar e recuperar dados, e quanto mais depressa melhor. Esta é a ideia que se tem em mente quando se fala em BD.

É também com este e outros objectivos em mente, que os artigos serão apresentados.

Os temas focados vão permitir a qualquer um iniciar-se no mundo das BDs e evoluir a ponto de criar conhecimento e sensibilidade suficientes para criticar e discutir sempre que o tema de conversa for o universo das BDs.

Base de Dados

De uma forma simples, uma BD pode ser definida como um simples banco de dados organizado segundo uma determinada estrutura.

Actualmente, os Sistemas de Gestão de Bases de Dados (SGBD) mais comuns e mais usados são os relacionais, que assentam sobre uma estrutura de relações entre entidades. No entanto, o mundo das BD não se esgota nos Sistemas de Gestão de Base de Dados Relacionais - Relational Database Management System na terminologia das BD - (RDBMS). Existem outras BD com estruturas hierárquicas ou por objectos, mas a sua utilização não é muito difundida e são normalmente usadas em casos particulares.

Exemplo

De forma a ilustrar e a permitir um melhor acompanhamento deste artigo, o mesmo vai ser complementado com um exemplo.

O desafio é construir uma BD para uma pequena biblioteca permitindo registar não só os livros, mas também os utilizadores e as requisições de livros.

Modelo Relacional

O modelo relacional define a estrutura da BD e é o primeiro passo no desenho de uma BD. Este modelo modela a realidade de forma lógica, ou, por outras palavras, representa a solução lógica para um dado problema.

Assim, o primeiro passo é identificar as entidades em causa. Dado o desafio de uma pequena biblioteca, a mesma terá:

- livro: que representa um livro;
- utilizador: que representa os utilizadores da biblioteca.

Uma das principais funções de uma biblioteca é emprestar livros, esta realidade não está ainda representada no modelo e é necessário criá-la.

Assim, numa segunda iteração, alteramos o modelo para permitir que os utilizadores se liguem aos livros através de requisições. Desta forma, as entidades identificadas são:

- livro: que representa um livro;
- utilizador: que representa os utilizadores da biblioteca;
- requisição: que representa a ligação entre um livro e um utilizador da biblioteca.

Assim, uma vez identificadas as entidades, é necessário identificar as características e propriedades de cada uma delas:

- livro: tem pelo menos um autor, tem um título, uma editora e o ISBN;
- utilizador: tem um nome, uma morada e o número do bilhete de identidade;
- requisição: tem o identificador do livro requisitado e tem o identificador do utilizador que efectuou a requisição.

Dado que é necessário identificar univocamente cada um dos intervenientes de cada entidade, torna-se necessário dotar cada entidade com um identificador único. Pensando no caso do utilizador, um possível identificador será o número do bilhete de identidade (BI). Pensando no caso do livro, um possível identificador será o ISBN. Neste caso, identificaram-se as chaves candidatas de cada uma das entidades, que são as propriedades que permitem identificar univocamente qualquer utilizador ou livro através de uma propriedade, ou de um conjunto de propriedades. Por exemplo, uma possível chave candidata poderia para um utilizador ser o nome e a morada, que juntos também identificam univocamente um utilizador.

Do conjunto das chaves candidatas selecciona-se a melhor que torna possível identificar univocamente qualquer utilizador, neste caso o utilizador através do BI, e o livro através do ISBN.

Tendo as chaves primárias seleccionadas, as entidades ficam assim representadas:

- livro: ISBN, tem pelo menos um autor, tem um título, tem uma editora;
- utilizador: BI, tem um nome, uma morada;
- requisição: ISBN, BI.

Agora a entidade requisição possui duas propriedades, ISBN e BI, que são as chaves primárias das entidades livro e autor, respectivamente. Ou seja, a entidade requisição possui duas chaves estrangeiras, uma vez que essas duas propriedades são chaves de outras entidades. Depara-se agora um desafio: qual a chave primária da requisição?

Meditemos um pouco: se uma requisição não é mais do que uma ligação entre um utilizador e um livro, então um utilizador e um livro definem univocamente uma requisição. Dado que um livro é identificado por ISBN e um utilizador por BI, então a chave primária desta entidade será o par ISBN, BI.

Dado que um livro pode possuir mais de um autor, e que um autor pode ter mais de uma obra, identificam-se duas novas entidades: autor e autoria; e reformulam-se as entidades.

Assim, numa terceira iteração sobre as entidades ficamos com:

- livro: que representa um livro;
- utilizador: que representa os utilizadores da biblioteca;
- requisição: que representa a ligação entre um livro e um utilizador da biblioteca;
- autor: que representa um autor;
- autoria: que representa um autor de um livro.

Numa segunda iteração sobre as características e propriedades de cada entidade conclui-se que:

- livro: ISBN, tem um título, tem uma editora, tem um número de cópias disponíveis;
- utilizador: BI, tem um nome, uma morada;
- requisição: ISBN, BI;
- autor: tem um nome;
- autoria: ISBN e o identificador do autor.

Uma chave candidata para a entidade autor será o nome, mas dado que pode haver mais de um autor com o mesmo nome, em particular porque os autores são, modo geral, conhecidos apenas pelo seu nome próprio e apelido, neste caso não existe nenhuma chave candidata suficientemente boa para ser eleita, razão pela qual se adiciona um campo extra para identificação e se efectua a ligação desse campo extra de identificação com a autoria.

Assim, ficamos com:

- livro: ISBN, tem um título, número de cópias disponíveis, idEditora;
- utilizador: BI, tem um nome, uma morada;
- requisição: ISBN, BI;
- autor: idAutor, tem um nome;
- autoria: ISBN, idAutor.

É de notar que a entidade de ligação autoria é análoga à de autor, ou seja, por fazer a relação de autores com livros, a sua chave primária é composta pelas duas chaves estrangeiras.

O modelo parece, à priori, finalizado, mas a verdade é que o mesmo não está normalizado. Olhando atentamente para a entidade livro, notamos que a editora ir-se-á repetir variadíssimas vezes em todas as entradas de livros que pertençam a uma mesma editora. Para evitar esta repetição do nome da editora, o modelo tem de ser normalizado. A normalização é um processo de reestruturação do modelo lógico de forma a eliminar redundância, organizar os dados de forma eficiente e reduzir potenciais anomalias. Olhando mais atrás no exemplo, também já efectuamos uma "pseudo-normalização", ao substituir no livro a regra "tem de ter pelo menos um autor" pelas entidades autor e autoria.

Normalizando o modelo, temos que a propriedade editora passa a ser uma entidade, e livro passa a fazer referência à editora em questão:

- editora: idEditora, nome;
- livro: ISBN, tem um título, cópias disponíveis, idEditora;
- utilizador: BI, tem um nome, uma morada;
- requisição: ISBN, BI;
- autor: idAutor, tem um nome;
- autoria: ISBN, idAutor.

Desta forma, todos os livros que pertencem a uma mesma editora têm a mesma referência, idEditora.

Note-se que também na entidade editora se aplicou a criação de uma propriedade própria para a identificação unívoca, embora aqui o nome pudesse ter sido seleccionado como chave primária, uma vez que a probabilidade de existirem duas editoras distintas com o nome exactamente igual é infínima.



Figura 1

As entidades principais definidas no exemplo podem ser vistas no diagrama da figura 1. É de referir que no diagrama não constam as entidades relacionais, as chaves primárias estão representadas a vermelho (PK) e o seu nome está normalizado para "id<Entidade>[Propriedade]".

Relações

As relações referidas atrás representam simplesmente a lógica de relação, ou ligação, entre as entidades. Essas relações podem ser de "1 para 1", "1 para N", ou de "M para N".

Uma relação diz-se de "1 para 1" quando um elemento de uma entidade se relaciona unicamente com outro elemento de outra entidade. Esta é a relação mais simples e menos interessante que existe, não necessitando sequer de entidades adicionais para a representar. No caso de um elemento de uma entidade se relacionar com mais de um elemento de outra entidade, estamos na presença de uma relação de "1 para N". No exemplo apresentado existe uma relação deste tipo: um livro possui um ou mais autores. Esta relação é representada pela entidade autoria: um ISBN relaciona-se com um ou mais idAutor. Uma relação "M para N" representa múltiplas relações entre os elementos de duas entidades. No exemplo apresentado existe uma relação deste tipo: um utilizador pode requisitar vários livros e um livro pode ser requisitado por vários utilizadores.

Só não é possível que um utilizador requisi-te o mesmo livro várias vezes devido à escolha do desenho do modelo. Esta relação é assim representada pela entidade requisição, que devido ao facto da chave primária ser constituída pelas chaves primárias das entidades relacionadas, não permite que um utilizador requisi-te várias vezes o mesmo livro. Esta restrição não é uma limitação, nem deve ser vista como tal, uma vez que é facilmente ultrapassável definindo uma outra chave primária para a entidade.

O diagrama da figura 2 apresenta o exemplo com as entidades relacionais materializadas.



Figura 2

Por fim, o diagrama da figura 3 (1), apresenta o modelo completo, com as relações e as entidades relacionais com os nomes normalizados para "PK_<idEntidade>". As chaves estrangeira estão representadas a verde (FK) e as chaves estrangeiras que são também chaves primárias estão representadas a azul (PFK).

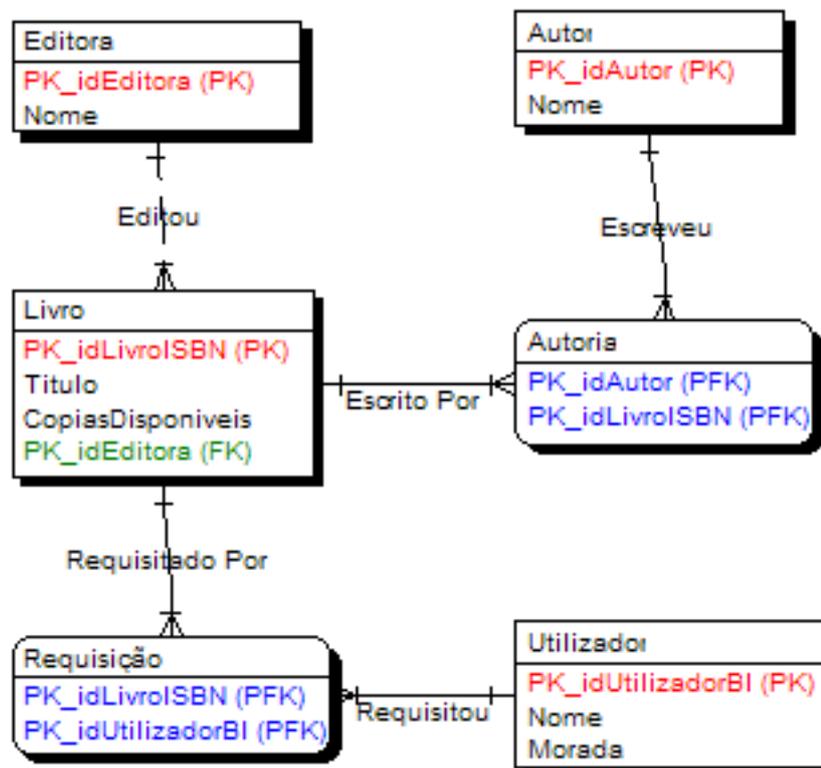


Figura 3

Cada uma destas entidades vai resultar num objecto na base de dados, uma tabela, e cada uma das propriedades de cada objecto vai resultar numa coluna. Mas este passo acontecerá apenas num próximo artigo, actualmente é importante reter os conceitos de Modelo Relacional, Modelo Lógico, Entidade, Propriedade, Chave Candidata, Chave Primária, Chave Estrangeira e Normalização.

1) A notação usada no diagrama é a notação Oracle™ (Oracle™ é uma marca registada)



Páginas dinâmicas com ASP e AJAX

Quem nunca pensou em fazer um Chat ou uma ShoutBox? Exacto, muitos de nós, no entanto, algo tão fulcral como a actualização da conversa tornava-se um quebra-cabeças difícil de solucionar. Ou se fazia um refresh à página (algo desagradável e a evitar), ou... Enfim, seria apenas mais um projecto não concretizado. Ora, foi precisamente para solucionar esses problemas que nasceu o AJAX (Asynchronous JavaScript and XML).

Quando falamos em AJAX convém ter a noção que não se trata propriamente de uma linguagem de programação nova, e muito menos se refere ao habitual detergente. Trata-se sim, de uma técnica para actualizar informação rapidamente. Imaginemos, por exemplo, um chat. Com o habitual HTML Refresh o browser iria carregar a página toda novamente. Já com o AJAX apenas será actualizado aquilo que quisermos, neste caso, o 'corpo' da conversa.

AJAX corre praticamente em todos os servidores mas, para que possamos usufruir deste, precisaremos também de uma linguagem Server-Side, neste caso o ASP. Além disso o AJAX tem a particularidade de ser apenas visível em browsers que tenham suporte completo XML. Actualmente existem apenas dois browsers a suportar totalmente esta tecnologia: o Internet Explorer e o Mozilla Firefox. Em browsers como o Safari ou o Opera é comum as aplicações com AJAX apresentarem alguns erros, ou deficiências.

Mas, deixemo-nos de teoria e passemos à prática. Iremos criar uma página de registo, com a particularidade de garantir que não hajam usernames repetidos.

```
<html>

<head>
<script src="codigoAJAX.js"></script>
</head>

<body>
<form action="" method="post">
    Utilizador Pretendido:
    <input type="text" name="username"
        id="username" onKeyUp=
            "verificaUtilizador(this.value)">
</form>
<div id="resposta"></div>

</body>
</html>
```

A parte do '`<script src="codigoAJAX.js"></script>`', inclui o código AJAX para que seja possível a verificação. Já o parágrafo abaixo, será a tal área em que nos é dito se o username já existe ou não. A acção `verificaUtilizador(this.value)`, requiere o `this.value` para que o browser saiba que deve actualizar a resposta sempre que o texto é alterado. A função sem o `this.value` teria de ser invocada por um botão através do `onClick` por exemplo, caso contrário não iria funcionar.

O código AJAX é o seguinte:

`codigoAJAX.js` (o código também poderia ser inserido na página do formulário mas, por motivos de organização, separamo-lo).

```
function handleHttpResponse() {
    if (http.readyState == 4) {
        if (http.status == 200) {
            if (http.responseText.indexOf(
                'invalid') == -1) {
                results = http.responseText.
                    split(",");
                document.getElementById(
                    "resposta").innerHTML
                    = results[0];
                emProceso = false;
            }
        }
    }
}
```

```
function verificaUtilizador() {
  if (!emProceso && http) {
    var valor = escape(document.
      etElementById("username").value);
    var url = "utilizadores.asp?
      txtUser="+ valor;
    http.open("GET", url, true);
    http.onreadystatechange =
      handleHttpResponse;
    emProceso = true;
    http.send(null);
  }
}
```

```
function getHTTPObject() {
  var xmlhttp;

  @cc_on
  @if (@_jscript_version >= 5)
    try {
      xmlhttp=new ActiveXObject(
        "Msxml2.XMLHTTP");
    } catch (e) {
      try {
        xmlhttp=new ActiveXObject(
          "Microsoft.XMLHTTP");
      } catch (E) {
        xmlhttp = false;
      }
    }
  @else
    xmlhttp = false;
  @end @

  if (!xmlhttp &&
    typeof XMLHttpRequest !=
    'undefined') {
    try {
      xmlhttp=new XMLHttpRequest();
    } catch (e) { xmlhttp=false; }
  }
  return xmlhttp;
}
```

```
var emProceso = false;
var http = getHTTPObject();
```

Este código, parece bem complicado mas não o é. Tem três etapas simples. Primeiro dizemos como e onde é que o browser nos vai responder, na área 'resposta'.

Em segundo, verificamos se o utilizador já existe, por uma QueryString com o valor inserido pelo utilizador, valor esse que será recebido no ficheiro ASP. Este dirá definitivamente se o utilizador está ocupado ou não. Por último, é executado o objecto HTTP, que só está disponível em alguns browsers.

Para terminar, só nos falta o código ASP, que não passa de meia dúzia de linhas. Primeiramente, o código irá recolher o valor da variável Utilizador, através do request.QueryString. De seguida, será feita a verificação e, finalmente, ser-nos-á dito se o utilizador está disponível. Para evitar 'bug's" só será apresentado resultado se o valor da variável não for nulo.

utilizadores.asp

```
<%
if request.querystring("Utilizador")
  <> "" then
  utilizador = request.querystring(
    "Utilizador")

  if utilizador = "p@p" then
    response.write "Ocupado,
      escolha outro."

  else
    response.write "Está
      Disponível!"

  end if
end if
%>
```

Agora é só testar! Este código está muito simples, mas é uma base para, por exemplo, incluir recordSets e interagir com uma base de dados. 





JavaScript (parte 2)

Nesta última parte do artigo sobre programação em JavaScript serão abordados temas um pouco mais técnicos, bem como mais avançados e evoluídos, do que na metade anterior. Abordaremos os operadores, capítulo onde serão tratados assuntos relacionados com a aritmética, as relações, a lógica entre outros, abordaremos as funções e as instruções que exprimem condições e repetições/ciclos, falaremos também um pouco sobre programação orientada a objectos, o básico a deste nível e uma pequena alusão aos vectores. Esperemos que goste da leitura.

Operadores

Os operadores actuam sobre uma ou mais variáveis de forma a produzir os resultados pretendidos. A linguagem de programação JavaScript possui muitos operadores de diversos tipos. Aqui iremos apenas abordar os aspectos mais básicos dos diversos tipos de operadores disponíveis.

Operadores Aritméticos

Os operadores aritméticos são utilizados para realizar operações matemáticas simples como a adição, a subtração, a multiplicação, a divisão e o resto da divisão inteira. Existe também o da negação unária que não vai ser abordado.

São também usados no incremento e decremento de uma unidade ou em operações relacionais como a igualdade, diferença, maior ou menor que, entre outros.

Operador Nome

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão

Exemplo de utilização:

Soma em JavaScript

```
<html><head>
<script language="Javascript">
  var x=1000;
  var y=18;
  soma=x+y;
</script>
</head><body>
<script language="Javascript">
  document.write(soma);
</script>
</body></html>
```

Como é possível verificar no código é muito simples e descreve uma pequena soma. O mesmo se verifica para as outras operações.

O JavaScript disponibiliza também operadores de incremento e decremento.

Operador Nome

++	Incremento
--	Decremento

As instruções de incremento como a de `a++` são de pós incremento e as instruções de incremento `++a` são instruções de pré incremento. Quando se tratam de instruções de decremento `a--` ou `--a`, a denominação será a mesma.

Operadores Relacionais

Os operadores relacionais como o próprio nome indica, especificam operações de relação entre valores.

Operador	Nome
==	igual
!=	diferente
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual

Exemplo de utilização:

```
<html><head>

<script type="text/javascript">
var x = 1;
var y = 2;
var z = y;
function relacionais(){
  document.write("<br/>Variável x="+x);
  document.write("<br/>Variável y="+y);
  document.write("<br/>Variável z="+z);
  document.write("<br/>");
  w = x == y;
  document.write ("<br/>y==z->" +w);
  w = x != y;
  document.write ("<br/>y!=z->" +w);
  w = x >= y;
  document.write ("<br/>x>=y->" +w);
  w = x < y;
  document.write ("<br/>x<y->" +w);
}
</script>

</head><body>

<script type="text/javascript">
relacionais();
</script>

</body></html>
```

Com estes operadores é possível fazer operações para determinar se os valores são iguais, diferentes, maiores ou menores, entre outros.

Operadores Lógicos

Os operadores lógicos existentes na linguagem JavaScript são:

Operador	Nome
!	Navegação
&&	Conjunção
	Disjunção

Estes operadores são os operadores que manipulam valores lógicos.

A disjunção inclusiva só resulta num valor falso quando os dois operadores são falsos (mesmo que OU). Caso contrário devolve verdadeiro.

A conjunção só pode resultar num valor verdadeiro quando ambos os valores o forem (mesmo que E). Já a negação resulta no valor contrário do operador.

Operadores bit a bit

Operadores bit a bit disponíveis em JavaScript.

Operador	Nome
x&y	Conjunção
x y	Disjunção inclusiva
x^y	Disjunção exclusiva
-x	Negação lógica
x<<y	Deslocamento à esquerda
x>>y	Deslocamento à direita
x>>>y	Deslocamento lógico à direita

Operadores de cadeias

Ao contrário dos outros operadores apenas existe um operador de cadeias que servem para concatenar cadeias (strings).

Operador	Nome
+	Concatenação

O operador + acrescenta uma cadeia no fim da outra.

Funções

Uma função é uma sequência de instruções que constitui uma entidade coerente e que pode ser invocada a partir de qualquer ponto do programa escrito em JavaScript. O programador pode criar as suas próprias funções que são na maioria das vezes escritas no cabeçalho do ficheiro e são invocadas posteriormente quando necessário.

Portanto, podemos ver uma função como uma série de instruções que englobamos dentro do mesmo processo. As funções utilizam-se constantemente, não só as que escrevem como também as que já estão definidas no sistema, como as `parseInt` (que é uma string), `parseFloat` (string) ou `NaN` (que é um valor), pois todas as linguagens de programação têm funções para realizar processos habituais. As funções servem para serem chamadas, sempre que existir uma acção que seja necessária em diversos locais da página/site/programa, só tendo definido o seu código uma única vez.

Para definir uma função é obrigatório utilizar certos padrões:

```
function nomedafuncao (){
  instruções da função
  ...
}
```

Depois é apenas necessário dar as instruções certas à função de forma a actuar da forma pretendida e invocá-la não esquecendo de adicionar `()` à frente do seu nome.

Instruções condicionais

Instrução if e else

If como o próprio termo em Inglês indica significa "se", assim sendo, esta instrução serve para realizar scripts que dependam de uma condição. Um If pressupõe que certa acção só será executada, caso se verificar uma certa condição.

Instrução switch

No JavaScript existe uma outra instrução de condição a instrução `switch`.

Formato `switch`:

```
switch (expressão){
  case valorx:
    instruçãox;
    ...
    break;
  case valory:
    instruçãoy;
    ...
    break;
  ...
}
```

Passando a explicar o comportamento do controlo `switch`: se o valor da expressão coincidir com o `valorx` executa-se a `instruçãox`, caso não seja, faz-se o mesmo mas para o `valory`, e assim sucessivamente, até ser encontrada uma igualdade.

Instruções Iterativas

Na linguagem de programação JavaScript apenas existem três instruções iterativas, o ciclo `while`, o `do-while`, e o ciclo `for`.

O ciclo while

Este ciclo utiliza-se quando é necessário repetir um número indefinido de vezes a execução de umas sentenças, sempre que se cumpra uma condição. É mais fácil de se entender que o ciclo `for` porque somente se indica a condição que tem que se cumprir para que se realize uma repetição.

```
while (condição){
  sentenças do ciclo;
  ...
}
```

O ciclo do-while

Utiliza-se geralmente quando não sabemos quantas vezes haverá de se executar o ciclo, assim como acontece com ciclo while, com a diferença de que o ciclo, independentemente da expressão teste, será sempre executado pelo menos uma vez.

```
do {
    sentenças do ciclo;
    ...
} while (condição)
```

O ciclo for

O ciclo for utiliza-se para repetir mais instruções num determinado número de vezes. Entre todos os ciclos o for costuma ser utilizado quando sabemos ao certo o número de vezes que queremos que seja executada a sentença, ou até quando.

```
for
(inicição;condição;actualização) {
    sentenças a executar;
    ...
}
```

O funcionamento do ciclo for é o seguinte: a expressão de iniciação é executada uma vez antes da primeira iteração; a condição é testada antes de cada iteração do ciclo e se for verdadeira executa a próxima instrução do ciclo, se for falsa termina o ciclo.

A instrução executada no ciclo for pode ser substituída por diversas instruções.

Instruções break e continue

A instrução break interrompe um ciclo. Interromper um ciclo significa sair dele e deixá-lo todo como está de forma a continuar com o fluxo do programa imediatamente. A instrução continue interrompe o ciclo passando assim para a iteração seguinte.

Objectos

O JavaScript dispõe de um conjunto de objectos predefinidos. Os objectos têm atributos (variáveis) e métodos (funções). A sintaxe para a utilização de objectos, métodos e atributos em JavaScript é muito simples:

```
objecto.atributo
objecto.metodo()
```

Alguns objectos são criados automaticamente pelo browser, outros têm de ser criados obrigatoriamente pelo programador.

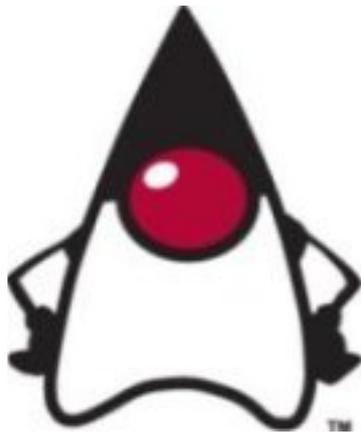
```
obj = new Object();
```

Vectores

Um vector é uma estrutura de dados com diversos elementos do mesmo tipo, em relação à qual, através de um índice, se pode designar cada um dos seus elementos. Os elementos de um vector podem também ser vectores, dando origem a uma estrutura muito complexa.

Conclusão

Nesta segunda e última parte do artigo de JavaScript foram abordados temas mais avançados, mas igualmente básicos, da programação em JavaScript. Nesta edição abordámos a programação com operadores, funções, instruções condicionais (if, switch), instruções iterativas (for, while, do-while, continue, brake), programação orientada a objectos (POO) e vectores. No fim deste artigo esperamos que o leitor tenha aprendido tudo o que necessita saber para se iniciar nesta linguagem. Agora é a sua vez. 



Input/Output em Java

Neste artigo vamos abordar uma matéria que, a nosso ver, é bastante interessante e importante no mundo da programação: o Input / Output neste caso aplicado a Java.

Iremos começar com Manipulação de Ficheiros e Directorias. Vamos ver a seguinte class Java onde são demonstradas as principais funções da Class File, que é a utilizada para realizar a manipulação. Em seguida vamos explicar o que cada linha faz.

```

1 import java.io.*;
2
3 public class JavaFile{
4     public static void main(String
                          args[]){
5
6         File file=new File("C:\\file.txt");
7
8         System.out.println(file.getName());
9
10        file.setReadOnly();
11
12        System.out.println(file.isHidden());
13        System.out.println(file.canRead());
14        System.out.println(file.canWrite());
15
16        file.renameTo(new File(
                          "C:\\Ficheiro.txt"));
17
18        try{
19            if( !file.exists() ){
20                file.createNewFile();
21            }
22        }

```

```

24     catch(Exception e){
25         System.out.println(e.getMessage());
26     }
27
28     if( file.isDirectory() ){
29         System.out.println("Directoria");
30     }
31
32     else if( file.isFile() ){
33         System.out.println("Ficheiro");
34     }
35
36     File [] roots = file.listRoots( );
37
38     for (int i=0; i<roots.length; i++){
39         System.out.println (roots[i] );
40     }
41
42     file.delete();
43 }
44 }

```

Como podemos ver na linha 6 é instanciado um objecto File que representa o caminho (path) para um possível local do sistema operativo. É bom lembrar que este apenas representa um ficheiro ou directoria, não pressupondo que o caminho exista realmente. Neste caso, o caminho é o C:\\file.txt, que aponta para o ficheiro file.txt na directoria C:. Também poderia apontar apenas para uma directoria e não para uma ficheiro, como é o caso.

Na linha 8 podemos ver o método getName(), que permite obter o nome do ficheiro ou directoria representada pelo File.

Esta class permite também dar atributos a ficheiros ou directorias, como é o caso do método setReadOnly() que dá ao ficheiro ou directoria o atributo de apenas poder ser lido e não escrito, tal como está representado na linha 10.

A class File permite também verificar atributos e, para isso, podemos usar os métodos isHidden() que verifica se o ficheiro ou directoria se encontra oculto(a) (linha 12), canRead() que verifica se é possível ler o ficheiro ou directoria (linha 13) e o método canWrite() que verifica se é possível escrever no ficheiro ou directoria (linha 14).

O método `renameTo()` na linha 16 permite renomear um ficheiro ou directoria, mas para além disso permite também mover ficheiros e directorias bastando para isso dar um caminho diferente no novo nome a dar, o que não é o caso neste exemplo.

Os métodos `exists()` e `createNewFile()` linha 19 e linha 20, são dois métodos muito importantes na manipulação de ficheiros e directorias, o método `exists()` permite verificar se o ficheiro ou directoria representados no File realmente existe. O método `createNewFile()` cria um novo ficheiro com o caminho representado, neste caso iria criar o ficheiro `file.txt` na directoria `C:`.

Embora aqui não esteja aqui presente, também temos os métodos `mkdir()` e `mkdirs()` que têm a mesma funcionalidade que o método `createNewFile()`, mas neste caso é criada uma directoria. A diferença do `mkdir()` para o `mkdirs()` é basicamente que o método `mkdir` apenas cria um directorio num caminho já existente, ou seja, por exemplo, no seguinte caminho `C:\Programas\` o `mkdir()` poderia criar directorias dentro da directoria `Programas`, apenas e só se a referida directoria já existisse previamente. Já o `mkdirs` permitiria criar toda a arvore de directorias, mesmo se esta não existisse.

Nas linhas 28 e 32 podemos ver os métodos `isDirectory()` e `isFile()` que verificam respectivamente se o caminho dado é um directório ou um ficheiro.

Na linha 36 temos o método `listRoots()`. Trata-se de um método bastante útil nomeadamente em sistemas Windows, visto que retorna um array com todas as drives ou raízes do sistema operativo, por exemplo `A:`, `C:`, `D:`, `E:`, etc... Já em sistemas GNU/Linux o conteúdo do array será apenas `/`, visto ser a raiz do sistema.

Para terminar esta parte do artigo, temos na linha 42 o método `delete()` que como o próprio nome indica permite eliminar o ficheiro ou directoria representado pelo File.

Depois de terminado o estudo sobre a manipulação de ficheiros vamos agora passar à escrita dos mesmos, observando o seguinte código:

```

1 import java.io.*;
2
3 public class FileWrite{
4
5     public static void main(String
6                                     args[]){
7
8         try{
9             PrintWriter out = new
10                PrintWriter(new BufferedWriter(new
11                    FileWriter("C:\\file.txt",true)));
12
13            out.println("Java Input/Output");
14            out.println(System.getProperty(
15                "os.name").toString());
16            out.println(System.getProperty(
17                "user.name").toString());
18            out.println("Fim");
19            out.println();
20            out.close();
21        }
22    }
23 }
24 }

```

Após executar esta aplicação será criado um ficheiro de texto em `C:\\` (nos sistemas Windows, para outros Sistemas Operativos bastará mudar o destino), com conteúdo semelhante a este:

```

Java Input/Output
Windows XP
Magician
Fim

```

Vamos agora analisar o código de forma a compreender o objectivo de cada linha.

Na linha 9 podemos ver o objecto "out" a ser instanciado. Esta é uma instanciação um pouco complexa mas tem a sua razão de ser. Segundo alguns testes já feitos esta instanciação é a forma mais simples e com

melhor rendimento de escrever num ficheiro. Vamos agora atentar em alguns detalhes. Na última parte da instância "new FileWriter("C:\\file.txt",true)" podemos ver que para além de termos como argumento o nome do ficheiro, temos também um argumento boolean. Este é sem dúvida um argumento muito importante pois quando colocado "true" diz ao programa que deve escrever no ficheiro, com a particularidade de não apagar nenhum do seu conteúdo anterior, ou seja, o programa não escreverá por cima do que já lá se encontra. No caso de se colocar como argumento um boolean "false", ou apenas o nome do ficheiro, o programa cada vez que for executado vai reescrever o ficheiro a partir do início apagando toda a informação já lá contida.

Num primeiro caso, irá aparecer duas vezes o conteúdo do ficheiro após executar o programa duas vezes com o argumento "true". O conteúdo do ficheiro só aparecerá uma vez, após executar o programa duas vezes, com o argumento "false" ou apenas sem argumento.

Na linha 11, bem como na 12,13,14,15, podemos ver a utilização do método println() sobre o objecto "out", que vai permitir escrever uma linha no ficheiro. Existem outros métodos como o print(), write() com funcionalidade idênticas embora menos utilizados a nível da escrita de ficheiros de texto.

Por fim, temos o método close que irá "fechar" o ficheiro, algo que deve ser sempre executado assim que se termina a escrita num ficheiro. É assim terminada a ligação entre o ficheiro e o programa. É, portanto, um passo fulcral, visto que caso não executado, poderá deitar a perder o conteúdo do ficheiro.

Agora que já sabemos como escrever em ficheiros apenas falta a leitura dos mesmos. Para isso vamos começar por observar o código seguinte.

```
1 import java.io.*;
2
3 public class FileRead{
4
```

```
5     public static void main(String
        args[]){
6
7     try{
8
9         BufferedReader leitor = new
            BufferedReader(new
                FileReader("C:\\file.txt"));
10    String linha = "";
11    linha = leitor.readLine();
12    while(linha != null){
13        System.out.println(linha);
14        linha = leitor.readLine();
15    }
16
17    leitor.close();
18
19    }
20
21    catch(IOException e){
22        System.out.println(e.getMessage());
23    }
24 }
25}
```

De uma forma resumida, o que este código irá fazer é simplesmente ir ao ficheiro file.txt, ler linha-a-linha e imprimir na consola. Como podemos ver na linha 9 é criado um objecto BufferedReader que, através de um objecto FileReader, irá aceder ao ficheiro file.txt. Na linha 10 iremos criar a variável 'linha', do tipo String, inicializando-a como String vazia (""). Em seguida, na linha 11, vamos ler a primeira linha do ficheiro e vamos colocar o conteúdo dessa linha na variável linha, sob forma de String.

Nas linhas 12,13,14 e 15, é feito um ciclo para ler todas as linhas do ficheiro, cada vez que o ciclo dá uma volta, é impresso o valor da linha anterior e lido a linha seguinte, até não existirem mais linhas no ficheiro. Finda a leitura, a ligação ao ficheiro é terminada, como podemos ver na linha 17. O resultado final vai ser exactamente o conteúdo do ficheiro.

Após estas três partes do artigo o leitor já conseguirá realizar as principais funções de IO em Java. Existem um sem número de outras formas de fazer o que foi aqui demonstrado, bem como um outro sem número de outras coisas de se pode fazer com Java IO. 



RFID Applications

Desenvolvimento de aplicações para dispositivos RFID

A tecnologia RFID já existe há alguns anos, mas apenas agora começa a emergir. Os dispositivos identificadores (chips) baixaram significativamente o preço/unidade, fazendo esta tecnologia mais propícia a investimentos.

RFID significa Radio Frequency Identification, ou seja, com pequenos chips que podem ser de vários formatos e tamanhos, com um determinado código, conseguimos identificar determinado produto. Por exemplo, quando este passa por uma antena RFID, que envia um sinal de rádio, este é refractado pelo chip, podendo ser este método feito de duas maneiras: por refracção simples, onde o chip recebe o sinal de rádio e com a força desse sinal é conseguida uma refractação, ou então, por transmissão independente, que quando o sinal é recebido o chip responde com o seu próprio sinal. Este último método permite uma maior distância entre o chip e a antena.

Para desenvolver aplicações para esta tecnologia existe um produto com grande experiência de mercado chamado RFID Anywhere, um software Sybase.

“Desenhado para automatizar o processo de desenvolvimento e deploying de aplicações RFID, o RFID Anywhere é uma plataforma de middleware que responde aos requisitos físicos da tecnologia, proporcionando também a capacidade de simulação e de resposta em ambientes RFID de produção.”

Quatro razões para a utilização deste software: simulação em tempo real de entrada/saída de dispositivos RFID numa rede totalmente personalizável, opções de desenvolvimento flexíveis, que permitem ao programador o total controlo sobre a plataforma que está a desenvolver, segurança de nível empresarial, onde todas as comunicações efectuadas são passadas por HTTPS e o suporte a vários Hardware Vendors.

Neste artigo irei apresentar ao leitor um tutorial step-by-step para executar simulações no RFID Anywhere, com o objectivo de observar a entrada e a saída das diversas tags (chips) no sistema.

Passos:

(no fim de cada passo gravar clicando na disquete verde à direita)

1º Configuração dos Simuladores:

Criar dois “RFID Multiprotocol Simulator connector” a partir da dropdown menu superior. Configurar os conectores com os seguintes dados:

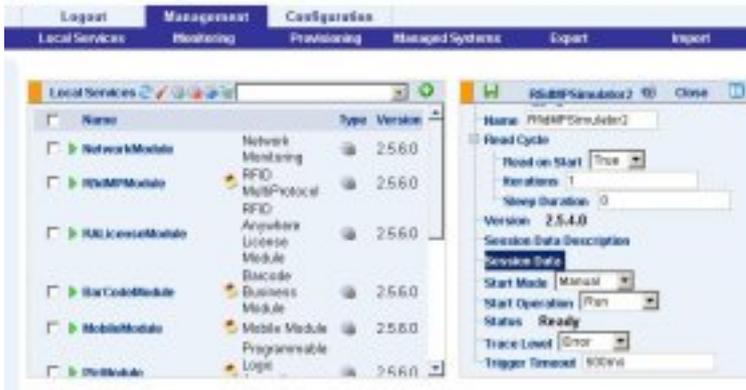
```

Antenna1
| Smoothing
| | Glimpsed Timeout: 2s
| | Lost Timeout: 5s
| | Observed Threshold: 2s
| | Oserved Timeout: 2s
Cycles Per Trigger: 1
Name: InvTracSim1 (ou 2 no 2º)
Session Data:
| simulation.smt (no 1º)
| simulation1.smt (no 2º)
  
```

2º Configurar o Protocolo ALE:

Seleccionar ALEServiceModule, expandir o grupo Sources e adicionar as seguintes fontes (inserir fonte e carregar no sinal + para cada um dos casos):

Source1: InvTracSim1
Source2: InvTracSim2



Name: InvTracSource1ALE
Send Reports as Object: True

- Subscribers
Escrever “InvTracAleTCP” e carregar no [+]

Criar um segundo objecto “ALE Event Cycle”, repetindo os passos anteriores com as seguintes propriedades:

Duration (ms): 5000
Logical Readers: Source2
Report Definitions: 2

1º Report:
Report Name: Source2Additions
EPC to Be Reported: ADDITIONS

2º Report:
Report Name: Source2Deletions
EPC to Be Reported: DELETIONS
Name: InvTracSource2ALE
Send Reports as Object: True
Subscribers: InvTracAleTCP

InvTracSource1ALE	ALE Event Cycle	2.5.5.0
InvTracSource2ALE	ALE Event Cycle	2.5.5.0
InvTracCountALE	ALE Event Cycle	2.5.5.0
InvTracAleTCP	Top Messaging	2.5.5.0
InvTracSim2	RFID Multiprotocol Simulator	2.5.4.0
InvTracSim1	RFID Multiprotocol Simulator	2.5.4.0

3º Configurar o TCP messaging:

Criar um novo objecto “TCPMessaging” a partir do dropdown menu superior e configurar com os seguintes dados:

IPAddress: localhost
Name: InvTracAleTCP
Port: 10002

4º Configurar os ALE Event Cycle connectors:

Criar um novo objecto “ALE Event Cycle” a partir do dropdown menu superior e configurar com os seguintes dados:

- Event Cycle
- - Start-Stop Conditions
- - - Duration (ms): 5000
- Logical Readers

Escrever “Source1” e carregar no [+]

Report(s) definition: 2

Gravar para um novo grupo de Reports aparecer:

- reportSpecs1
- Name: Source1Additions
- EPC to Be Reported: ADDITIONS
- reportSpecs2
- Name: Source1Deletions
- EPC to Be Reported: DELETIONS

Crie um último objecto “ALE Event Cycle”, repetindo os passos anteriormente dados com as seguintes propriedades:

Duration (ms): 5000
Logical Readers:
Source1
Source2
Report Definitions: 1
Grouping Patterns: urn:epc:pat:sgtin-96:X.X.X.*
Include Count: True
Include Tag Uri: False
Report If Empty: True
Report Name: CountReport
EPC to Be Reported: CURRENT
Name: InvTracCountALE
Send Reports as Object: True
Subscribers: InvTracAleTCP

Nota: Esta demonstração usa mais ALE reports e conectores do que seria normalmente requerido. Isto é feito para mostrar uma maior variedade de opções ALE.

Para ver o resultado:

- Iniciar/Reiniciar todos os componentes alterados/criados;
- Correr a seguinte aplicação:
 "...\Sybase\RFID
 Anywhere\Bin\iAnywhere.RfidNet.Ale.ReportViewer.exe";
- Clicar no separador TCP/IP
- Modificar a server port para 10002
- Clicar em "Start"

Acabou de configurar a sua primeira simulação no RFID-Anywhere!

CRIADO CENTRO DE INVESTIGAÇÃO RFID EM PORTUGAL

SYBASE RFID SOLUTIONS CENTER

"O Centro ir-se-á focar essencialmente no desenvolvimento de Frameworks opensource destinadas ao mercado empresarial português. Já no 2º trimestre de 2007 pretende-se disponibilizar uma primeira versão deste framework que irá contemplar soluções de tracking de pessoas, inventário de produtos, sistema de gestão de armazéns, entre outros"

Dia 19 de Setembro de 2006 a Sybase Portugal anunciou em comunicado a criação de um centro de investigação e desenvolvimento de soluções RFID, o primeiro da Sybase a nível mundial.

O Centro de Investigação RFID é coordenado por Paulo Lopes, Business Development Manager da Sybase, e é o primeiro centro de competências da Sybase exclusivamente dedicado a RFID. As soluções serão desenvolvidas com base em produtos middleware Sybase e de parceiros, nomeadamente a Symbol, Intermec/Cagicom e Alien/Creative Systems, entre outros.

O Centro está a funcionar nas instalações da Sybase Portugal em Carnaxide e conta

com todo o software (essencialmente RFID Anywhere e .NET) e hardware (vários tipos de tags, computadores, dispositivos móveis, readers, etc.) necessários para desenvolver e testar soluções RFID de qualidade empresarial.

Segundo a Sybase, os objectivos principais são o desenvolvimento da tecnologia, a disponibilização de Frameworks opensource para o mercado e o estabelecimento de parcerias com fornecedores de hardware, ISVs, Integradores, Associações e Institutos de Investigação.

A Sybase, um dos líderes de mercado na área em pleno desenvolvimento que é o RFID, aposta na investigação no campo e deixa assim um desafio às empresas mais viradas para a área da distribuição e da comercialização: o de poderem ter um significativo aumento de produtividade com a aposta nesta tecnologia emergente.

Casos de Sucesso

EM PORTUGAL SATU Oeiras

"Com a tecnologia RFID é acelerado o processo de validação de bilhetes, evitando assim a acumulação de pessoas nas entradas das estações do SATU."

O mais recente transporte criado pela Câmara Municipal de Oeiras também aposta na tecnologia RFID para a entrada de passageiros e carregamento dos bilhetes.

Um normal passageiro deste transporte apenas tem de adquirir um cartão (com o custo de 0.50€) e carregar o mesmo com as viagens que pretender efectuar, para qualquer das estações da linha do SATU (Sistema Automático de Transporte Urbano).

Na próxima imagem podemos ver a Antena de leitura das tags embutidas nos cartões dos utilizadores, que naturalmente estará ligada a um reader.



O veículo da próxima imagem utiliza para se mover um sistema de carris e é “puxado” por cabos de aço na parte inferior do mesmo, que por sua vez são movidos por motores presentes na primeira estação do percurso.



Em baixo podemos visualizar duas máquinas de carregamento de cartões, onde ao centro se coloca o cartão, sem estar em contacto directo com a própria máquina, permitindo assim aos utilizadores um carregamento mais rápido, evitando os normais atrasos de impressão de bilhetes.

Neste momento o percurso deste transporte ainda é algo reduzido, mas sem dúvida útil para a população da zona.



EM ESPANHA Correos de España



“Estamos actualmente a finalizar a instalação do hardware e a começar a instalação do middleware que foi fornecido pela Sybase”, em declarações dos Correos

A implementação deste sistema de controlo por radiofrequência nos seus 15 centros de transporte e armazenamento distribuídos por toda a Espanha teve um custo de mais de um milhão de euros, para aquisição da tecnologia necessária. Esta quantia inclui software específico e a sua integração nos sistemas informáticos dos Correos, a aquisição de 5.000 tags passivas, a instalação de mais de 1.900 antenas fixas e mais de 330 readers móveis para esses centros, fornecidos pela Symbol Technologies.

Como principais benefícios, os Correos reconhecem que com um sistema RFID se conseguiu melhorar a qualidade do serviço, dar um maior valor aos seus clientes e estabelecer uma vantagem competitiva em relação à concorrência. A empresa também admite que obtém um menor custo "devido à standardização da tecnologia e ao custo unitário das tags passivas em relação às tags activas", e uma maior funcionalidade, "já que além do controlo de qualidade podemos aplicar a outras funcionalidades como o controlo de elementos logísticos (carros, camiões, etc.)". 



Visual Basic.NET 4ª parte

Na edição anterior fizemos uma abordagem sobre a utilização de arrays e como usar as classes do namespace System.IO para a manipulação de ficheiros.

Como sabemos, qualquer ficheiro possui um conjunto de propriedades, propriedades estas que estão acessíveis usando a classe FileInfo.

```
Dim ficheiro As New
FileInfo("ficheiro.txt")
```

Propriedades do tipo, nome do ficheiro, localização, extensão, atributos, tamanho, etc, estão facilmente acessíveis recorrendo as propriedades incorporadas na classe FileInfo. Exemplo:

Função para obter as propriedades de um ficheiro

```
Private Function getProperties(ByVal
caminho As String) As Boolean
Try
Dim ficheiro As New FileInfo(caminho)
Dim propriedades As String = ""
With ficheiro
propriedades &= "Prop. do ficheiro" &
ficheiro.Name.ToString & vbCrLf
propriedades &= "=====" & vbCrLf
propriedades &= "Nome: " &
ficheiro.Name.ToString & vbCrLf
propriedades &= "Caminho: " &
ficheiro.FullName.ToString & vbCrLf
propriedades &= "Extensão: " &
ficheiro.Extension.ToString & vbCrLf
propriedades &= "Tamanho:" &
ficheiro.Length.ToString & vbCrLf
```

```
propriedades &= "Atributos:" &
ficheiro.Attributes.ToString & vbCrLf
propriedades &= "Criação:" &
ficheiro.CreationTime.ToString & vbCrLf
propriedades &= "Ultimo Acesso:" &
ficheiro.LastAccessTime.ToString & vbCrLf
propriedades &= "Modificação:" &
ficheiro.LastWriteTime.ToString & vbCrLf
End With
Console.WriteLine(propriedades)
Return True
Catch ex As Exception
Return False
End Try
End Function
```

Função Main

```
Sub Main()
Dim strCaminho As String = ""
While Trim(strCaminho) = String.Empty
Console.Write("Caminho do ficheiro: ")
strCaminho = Console.ReadLine()
Console.Clear()
End While
If Not getProperties(strCaminho) Then
Console.WriteLine("Erro")
System.Threading.Thread.Sleep(2000)
Else
Console.ReadLine()
End If
End Sub
```

Output

Caminho do ficheiro: c:\ficheiro.txt

Prop. do ficheiro

=====

Nome: ficheiro.txt

Caminho: c:\ficheiro.txt

Extensão: .txt

Tamanho 37

Atributos: Archive

Criação: 11-10-2006 16:27:14

Ultimo Acesso: 11-10-2006 16:28:14

Modificação: 11-10-2006 16:28:14

Como vimos, uma das vantagens em usar a classe FileInfo, é a possibilidade de obter propriedades de um ficheiro de uma forma rápida, mas não só. Esta classe incorpora também funções para criar ficheiros, mover ou copiar da sua localização actual para outra, apagar, encriptar, escrever, ler, etc.

Não são apenas os ficheiros que possuem propriedades, pois os directórios também as possuem e, por isso, temos disponível a classe `DirectoryInfo`, contendo quase as mesmas propriedades da `FileInfo`, não diferenciando muito a sua utilização da dos ficheiros. Quem tem vindo a acompanhar o seguimento dos artigos sobre esta linguagem, certamente que se apercebeu que o objectivo é dar uma ajuda a quem pretende iniciar-se em VB.NET, logo, todas as matérias não são extensivamente abordadas, ficando isso ao cargo de cada um, por isso, termina aqui a abordagem a ficheiros.

Outra das grandes implementações na .NET Framework, foi o namespace `System.String`. Na maioria dos projectos, existe uma situação em que precisamos de fazer o tratamento de uma string, daí a implementação de classes para cada vez mais facilitar a tarefa do programador. Neste artigo abordaremos as funções mais importantes para o tratamento de strings, e para exemplificar iremos utilizar uma string fixa, que será "Visual Basic .NET".

As funções que iremos exemplificar são as que, por norma, são mais utilizadas, sendo elas, `Contains()`, `IndexOf()`, `Insert()`, `Length()`, `LastIndexOf()`, `SubString()` e `Trim()`.

A função `Contains()` permite-nos saber se uma determinada string existe na string em questão. No exemplo seguinte, irá ser retornado o texto "Existe", pois como podemos verificar, a string "Ba" está contida na string exemplo.

```
Dim str As String = "Visual Basic
.NET"
If str.Contains("Ba") Then
    Console.WriteLine("Existe")
Else
    Console.WriteLine("Não Existe")
End If
```

A função `IndexOf()` permite-nos saber se uma determinada string existe no texto e devolve-nos a posição inicial dela caso exista. No exemplo seguinte, irá ser retornada a posição 7, pois é onde se encontra a letra "B".

```
Dim str As String = "Visual Basic .NET"
Console.WriteLine(str.IndexOf("B"))
```

A função `Insert()` permite-nos inserir em qualquer posição de uma string, outra string. No exemplo seguinte, irá ser adicionada a string "é fixe" no fim da string `str`, retornando uma string "Visual Basic .NET é fixe".

```
Dim str As String = "Visual Basic .NET"
str = str.Insert(17, " é fixe")
Console.WriteLine(str)
```

A função `LastIndexOf()` devolve-nos a última posição de uma string que se encontra num determinado texto. No exemplo seguinte, será retornada a posição 8, pois é onde se encontra a última letra "a" na string.

```
Dim str As String = "Visual Basic .NET"
Console.WriteLine(str.LastIndexOf("a"))
```

A função `Length()`, indica-nos o tamanho que uma string tem. No exemplo seguinte, irá ser retornado o valor 17, pois a string utilizada contém 17 caracteres.

```
Dim str As String = "Visual Basic .NET"
Console.WriteLine(str.Length)
```

A função `SubString()` permite-nos "cortar" partes de uma string.

```
Dim str As String = "Visual Basic .NET"
Console.WriteLine(str.Substring(7, 3)) 'OUTPUT: "Bas"
```

A função `Trim()` talvez seja a mais usada de todas. Esta função permite-nos remover os espaços que estão a mais numa string. Um exemplo da sua utilização:

```
Dim str As String = " Visual Basic .NET "
Console.WriteLine(str.Trim) 'OUTPUT: "Visual Basic .NET"
```

Como poderão ver, existem outras funções para tratamento de strings, que podem ser rapidamente descobertas na documentação. Quanto ao tratamento de strings é tudo. Na próxima edição iremos dar início à abordagem sobre base de dados, por isso estejam atentos ao lançamento da edição seguinte. 



Mounting File Systems

A montagem de sistemas de ficheiros nos sistemas GNU/Linux é algo que gera sempre em qualquer novato na matéria dúvidas e indecisões. Este artigo pretende explicar com simplicidade e desde o início, todos os passos requeridos para que tudo fique a funcionar correctamente.

É sem dúvida um entre muitos outros “choques” o que se tem ao experimentar um sistema GNU/Linux pela primeira vez, querer aceder àquela drive onde guardamos toda a nossa vida e não conseguir nem saber como o fazer. É certo que os Desktop Environments mais recentes têm facilitado muito o processo, contudo, nem sempre nos oferecem como opção gráfica aquilo que queremos exactamente fazer. Como tal, convém conhecer e dominar os principais mecanismos do sistema de montagem de ficheiros e usá-los como bem entendermos, sem limitações. Vamos a isso...

Primeiro, há que reter a ideia que em GNU/Linux qualquer drive, para poder ser acedida, tem de se encontrar montada, ou seja, a sua presença física teve de ter sido identificada pelo sistema, à drive ter sido associado um directório de montagem e, por fim, ter-se procedido à associação do conteúdo da drive ou sistema de ficheiros, ao directório de montagem definido. Para tal, usa-se o comando `mount`, que funciona da seguinte maneira:

```
# mount <directório representativo do hardware ou ficheiro> <directório de montagem> <opções>
```

Por exemplo,

```
# mount /dev/sda /mnt/pendrive -t vfat
```

Especifica que a pen drive (que, neste caso, é representada por `/dev/sda` no directório que contém os ficheiros que representam o hardware `/dev`) será montada no directório (que deverá já previamente existir) `/mnt/pendrive`, tendo como opção `-t vfat`, o que informa o `mount` de que se está a trabalhar com um sistema de ficheiro em FAT. Para saber mais facilmente a que `/dev/xxx` corresponde a drive que desejamos montar deverá correr o comando: `# fdisk -l`, ou qualquer correspondente gráfico.

No entanto, e como o leitor provavelmente estará a pensar, seria bastante cansativo sempre que se iniciasse o sistema ter de proceder à execução destes comandos para cada drive a que quiséssemos aceder, como tal, iremos também aprender a tornar tudo isto automático.

De qualquer modo, e antes de tudo isso, há algo a importante a ver. Todo este processo de montagem requer que o kernel da nossa distribuição já contenha carregados módulos que lhe permitirão, desde montar os sistemas de ficheiros habituais em GNU/Linux (como o `ext2/3`, o `ReiserFS`, o `JFS`, ...), bem com os de Windows (`FAT`, `NTFS`,...), assim como algumas propriedades destes (sistema de codificação de caracteres).

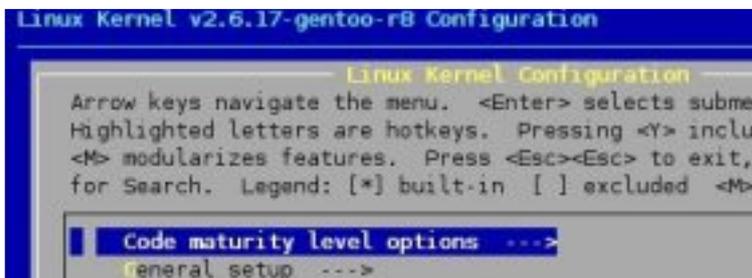
De um modo geral quase todas as distribuições já têm activos estes módulos de raiz e, portanto, só terá de se preocupar mesmo com o processo de montagem de sistemas de ficheiros, seja ele manual ou automático. Utilizadores de Ubuntu, SuSE, Mepis, Fedora Core, entre muitas outras podem passar directamente ao ponto seguinte. De qualquer modo, se algum comando lhe der um erro, poderá ser derivado da inexistência de suporte no kernel ao sistema de ficheiros pretendido e, como tal, deverá seguir o exposto a seguir.

Recompilando o Kernel para garantir suporte aos sistemas de ficheiros pretendidos

Poder-se-á pensar que este é um passo complicado ou mesmo arriscado, no entanto apenas requer um pouco de atenção e tudo correrá bem.

A sua distribuição deverá ter um método particular para proceder a esta actividade e, como nem sempre o método tradicional é universal para todas as distribuições, deixo a cargo do leitor procurar nos fóruns da sua distribuição um tutorial que lhe indique como proceder para a recompilação do kernel. Sublinho que para a grande maioria das distribuições nada disto será necessário.

Deverá, se seguiu um qualquer tutorial sobre como recompilar o kernel, encontrar-se neste momento numa interface semelhante à da imagem (geralmente acessível pelo comando `make menuconfig` ou `make xconfig`).

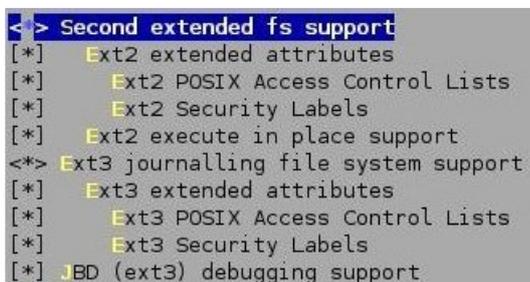


Movimente-se com as teclas de direcção, Enter para entrar nas pastas, a barra de espaços para seleccionar cada módulo que desejar (estará correctamente seleccionado quando aparecer `<*>` no princípio da linha) e Esc para sair.

Entre em: [File Systems](#)

Aqui terá de seleccionar cada sistema de ficheiros para o qual quiser ter suporte.

Um exemplo:



De seguida entre em:

```
Device Drivers
->DOS/FAT/NT Filesystems
```

E seleccione o suporte para os sistemas de ficheiros Windows que desejar. Em Default Charset for FAT deverá seleccionar Enter e inserir iso8859-15, caso o seu sistema de ficheiros seja acedido também num sistema Windows, de modo a não ter problemas nem na visualização, nem na escrita, do texto presente nesse tipo de partição. Acima desse valor encontra-se o de Default Codepage for FAT que deverá ser o 860 para a língua portuguesa. De seguida, se acabou de seleccionar o suporte para os sistemas de ficheiros Windows, deverá também activar o suporte aos sistemas de caracteres definidos anteriormente, em:

```
Device Drivers
->Native Language Support
```

E deverá activar:

```
<*> Codepage 860 (Portuguese)
<*> NLS ISO 8859-15 (Latin 9; Western
European Languages with Euro)
```

Opcionalmente pode dar jeito adicionar também:

```
<*> NLS UTF-8
```

Deverá também verificar se em:

```
Device Drivers
-> CDROM/DVD Filesystems
```

contém suporte para esses tipos de armazenamento.

Caso pretenda montar ficheiros de imagem de cd/dvd .iso, seleccione também nessa pasta:

```
<*> Microsoft Joliet CDROM extensions
```

Mais:

```
Device Drivers
-> Block devices
<*> loop-back filesystem support
```

Caso pretenda suporte USB terá de se dirigir a:

```
Device Drivers
-> USB Support
<*> USB Mass Storage Support
```

e tudo o mais se adequar ao hardware presente na sua máquina.

Use o TAB para seleccionar Exit, e repita várias vezes até sair do configurador e lhe ser perguntado se deseja gravar as alterações feitas, onde deverá responder Yes.

Prossiga o tutorial da sua distribuição de maneira a aplicar o novo kernel e activá-lo.

Montando...

Posto tudo isto, vamos finalmente passar à acção e a aprender a montar os sistemas de ficheiros que necessitamos sem qualquer problema. Primeiro, e como já foi referido, há um ficheiro de texto que define, aquando do carregamento do sistema, as drives a serem montadas automaticamente. Tal ficheiro é o `/etc/fstab` e é editável apenas pelo super utilizador root. Tudo aquilo que se faz pelo comando `mount` pode ser definido no `fstab` e deste modo montado desde o início sem intervenção directa do utilizador. O ficheiro consiste em 5 colunas que definem exactamente o mesmo que é definido no comando `mount`, sendo estas:

```
<fs> <mountpoint> <type> <opts>
<dump/pass>
```

Correspondendo o `<fs>` à imagem do hardware no directório `/dev/`, o `mountpoint` ao directório onde tal drive deve ser montada para que seja acedida, o `type` refere-se aos sistema de ficheiros (`fat`, `ext3`, etc), o `<opts>` a opções. Já o `dump` refere-se à necessidade ou não de ser feito um `dump` ao disco aquando do seu carregamento, e o `pass` refere-se à ordem pela qual deve ser feito um `check` às drives caso o sistema não tenha sido terminado correctamente (0 refere que não é necessário nunca fazer tal verificação, e a partição root deverá ter o valor 1).

Um exemplo para a partição root:

```
# <fs> <mountpoint> <type> <opts> <d/p>
/dev/hda1 / reiserfs noatime 0 1
```

O facto de uma drive se encontrar no `fstab`, não quer dizer exactamente que no momento do boot tenha de lá estar algum tipo de dados a serem montados. Por exemplo, pode adicionar-se uma drive de `cdrom's` ao `fstab` sem lá se encontrar nenhum `cd`, quando for para a aceder bastará fazer:

```
# mount /dev/cdrom0
```

ou

```
# mount /mnt/cdrom
```

que o comando automaticamente lerá o `fstab` e executará o processo com todas as opções lá definidas. Sendo por sua vez a definição na drive no `fstab`:

```
/dev/cdrom0 /mnt/cdrom iso9660
noauto,ro 0 0
```

O facto de apenas o utilizador root poder montar/desmontar a partição/drive pode ser alterado pela adição da opção `'user'` na coluna `<opts>` do `fstab`, na linha da drive pretendida. Já o `'ro'`, representa `read only`, ou seja, apenas será possível a leitura nessa drive `cd`.

Para montar uma partição FAT faríamos por exemplo, novamente tudo na mesma linha:

```
/dev/hda4 /mnt/particaofat vfat
user,rw 0 0
```



Que nos montará a drive com 'rw', ou seja, com a particularidade de se encontrar com acesso read and write (escrita e leitura). Se porventura esse acesso for apenas válido para o utilizador root, e desejarmos definir as permissões da drive em questão, basta nas opções adicionar o comando `umask=0xxx`, em que o xxx é um número que define as permissões da partição mas invertidas, e o 0 inicial o prefixo que indica que o número consequente estará em octal. Ou seja, se quisermos definir uma permissão 777 faremos `umask=0000`, se queremos 644 faremos `umask=0133`. Sempre 7 menos o número que define a permissão que pretendemos. Para saber definir permissões veja a edição passada da Revista PROGRAMAR.

Podemos também definir uma pen usb no `fstab`, fazendo para isso:

```
/dev/sda /mnt/pendrive vfat
user,rw 0 0
```

Não nos esqueçamos que na compilação do kernel foi definido, para os sistemas de ficheiros FAT, o sistema de codificação de caracteres default. Será com esse que a drive será montada. Se não o desejar use a opção `nls=`, por exemplo `nls=iso8859-15`, ou `nls=utf8`.

Para ntfs faríamos por exemplo:

```
/dev/hdb1 /mnt/windows ntfs
nls=utf8,user,ro,uid=1000,gid=1000 0 0
```

Que define que a drive a ser montada possui sistema de ficheiros em NTFS, com o sistema de codificação de caracteres definido para `utf8` (pode ser usada `iso8859-15`, mas é irrelevante visto que por defeito nos sistemas GNU/Linux nada pode ser escrito em ntfs), o utilizador normal poderá montá-la e desmontá-la ('user'), mas apenas poderá lê-la (ro).

Agora, o uid e gid, é uma maneira diferente de definir permissões, enquanto o `umask` faz o equivalente a um `chmod` (definir permissões), o uid e gid fazem o equivalente a um `chown` (definir a quem se aplicam as permissões definidas). Para saber os valores a pôr a seguir ao `=` de cada opção faça:

```
$ id <nome do user que quer que seja dono (owner) da drive>
```

E obterá a resposta.

Outra particularidade da montagem de sistemas de ficheiros é que poderá usar o comando `mount` para montar imagens de ficheiros em `.iso`, para isso:

```
# mount -t iso9660 -o loop,user
<imagem>.iso
/<directório de montagem>
```

Penso que estão desta maneira explicadas as opções básicas para a montagem de sistemas de ficheiros, como o fazer, e certos requerimentos necessários ao nível do kernel. Existem muitas mais opções, talvez tantas que dariam conteúdo para uma revista PROGRAMAR inteira, mas isso fica a cargo do utilizador que acabou de ler este artigo e deseja saber mais. O comando `man mount`, deverá responder mais detalhadamente a todas as opções possíveis e suas funções.

Boas montagens... 





A necessidade é a mãe de todas as invenções

weSpendMoney é uma aplicação web destinada a pessoas que querem controlar as suas despesas de forma fácil, rápida e segura. Não é preciso instalar em nenhum computador, apenas é necessário registar no site e ter uma ligação à Internet para começar a usar.

A ideia para o projecto começou numa conversa de café, entre colegas de trabalho, onde discutíamos a questão do controlo das despesas feitas ao longo dos dias e semanas. Poucos de nós realizamos com alguma periodicidade a actualização das cadernetas e chegamos à conclusão que os extractos bancários passado algum tempo são ilegíveis, pois a fraca descrição "humana" das despesas, por exemplo "COMP 98, 5 euros", não permite recordar exactamente qual a despesa em si. Outro problema consiste no uso de aplicações actualmente existentes para o Windows, normalmente com um interface confuso, com "features" que ninguém usava e difícil de aprender.

Lançado o problema na mesa percebemos que a melhor solução não era gravar cada despesa que fazemos em ficheiros de texto armazenados num pc, mas numa aplicação web que permitisse acesso em qualquer altura e lugar, sendo necessário ser intuitiva e segura (pois o dinheiro é sempre um assunto sensível na internet) e fácil de usar.

Depois de muitas sugestões decidi iniciar esta aventura de fazer uma aplicação que não só me resolvesse a falta de controlo sobre os gastos, mas que também servisse de treino para os meus skills com uma certa framework que já andava há estudar a algum tempo...

Duas semanas depois e com uma primeira versão funcional comecei a ponderar a hipótese de tornar o projecto disponível na Internet, ou mesmo criar uma versão comercial. Para confirmar todas estas questões, foi feito um plano de negócio básico e o projecto foi apresentado na Conferência Barcamp Pt, (<http://barcamp.org/BarCampPortugal>), conferência esta sobre novas tecnologias, programação, web 2.0 e outros assuntos. A apresentação foi um sucesso, várias pessoas demonstraram interesse no projecto e mais tarde foi feita outra apresentação pública na Shift 06, onde consegui efectuar muitos contactos e perceber o potencial que temos em Portugal para desenvolver para a web.

\$ my global situation

my.accounts

Title	Description	Balance	Create New
amoreiras	á ordem	\$652.10	
poupanças	a prazo	\$100.00	
visa	cartão de credito	\$2,500.00	
Total:		\$3,252.10	

my.recent.transactions

Article	Value	Date	Account
PES 6	\$-69.90	10/29/2006	amoreiras
Mac Mini	\$-900.00	10/29/2006	amoreiras
salario humilde	\$1,500.00	10/29/2006	amoreiras

WeSpendMoney foi desenhado de raiz pensando no utilizador. Não se pretende construir uma aplicação com centenas de "features" que não sejam usadas pelos utilizadores, mas procurar ser tão simples como um weblog e fornecer aos utilizadores as ferramentas que precisam para guardar e catalogar as suas despesas, definir orçamentos mensais e agendar transacções.

Também é possível consultar relatórios sobre o que se gastou, quando e onde, para além de se poder exportar a qualquer momento todos os nossos dados para um ficheiro ".csv" que pode ser abertos pelo Microsoft Excel. Isso quer dizer que os nossos dados não ficam "presos" pela aplicação.

my.calendar



veja em que dia gastou mais

Para começar o desenvolvimento do weSpendMoney não foram necessários muitos recursos, aliás, essa começa a ser a norma com a maioria dos projectos relacionados com a web 2.0. Foi utilizada a framework opensource Ruby On Rails como base da programação e o sistema de gestão de base de dados escolhido foi o MySQL, que é igualmente gratuito. O sistema de controlo de versões escolhido foi o SVN (mais conhecido por Subversion) pelas suas características superiores ao CVS e a gestão do projecto ficou facilitada pelo uso do Basecamp (<http://www.basecampHQ.com>), um gestor de projectos online também ele feito em Ruby On Rails.

Mas afinal o que é Ruby On Rails e qual a sua importância para o projecto?

Ruby On Rails é uma framework completa para o desenvolvimento de aplicações web, a sua filosofia assenta na felicidade do programador e em produtividade sustentável.

Rails é uma linguagem muito dinâmica, altamente produtiva, orientada a objectos com uma sintaxe coerente, familiar e muito simples que utiliza o padrão MVC (Model View Controller).

Rails segue dois conceitos que visam aumentar a produtividade de quem programa: DRY e Convention over Configuration. DRY (Don't Repeat Yourself, Não se repita) é o conceito por trás da técnica de escrever nomes, propriedades e código em somente um lugar e reaproveitar essas informações em outros sítios da nossa aplicação, é essencial para manter código limpo e a aplicação facilmente actualizáveis no futuro.

Convention Over Configuration (Convenção ao invés de Configuração) é um conceito onde se pretende o uso de convenções ou padrões e caso o programador deseje, pode-se sobrescrever essa convenção com o valor que deseja (ou foge à norma). Por exemplo, a classe Carro pode ter os seus dados armazenados na tabela Carros, mas se desejarmos podemos atribuir outro nome à tabela. Com estes dois conceitos e muitos outros pormenores intrínsecos da framework, o tempo de desenvolvimento é mais curto e o trabalho do programador é facilitado.

my.tags



organize as suas despesas



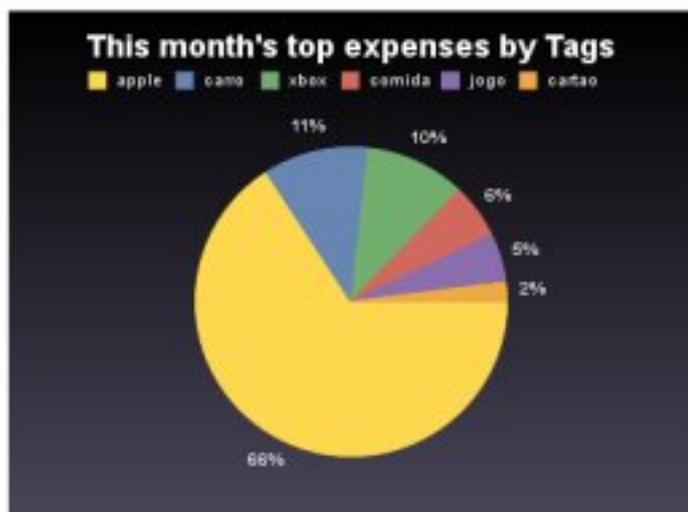
today's balance is **\$502.10**

31/10/2006 balance is \$301.10

saiba o saldo de hoje e amanhã

Poderia falar mais detalhadamente de Rails mas é melhor deixarmos isso para um futuro artigo, de qualquer forma qualquer pessoa pode fazer download e começar a programar em Ruby ou RubyOnRails, basta para isso ir ao endereço oficial (<http://www.rubyonrails.com>), ou se desejarem mais informações em português no Ruby<<portuguese (<http://www.ruby-pt.org>).

Esta alta produtividade fornecida pela framework (e a filosofia que a acompanha) foi a principal razão pela sua escolha na produção do weSpendMoney (admito que a integração com as bibliotecas prototype e script.aculo.us para gerar efeitos Ajax também pesou na escolha).



A aplicação ainda está em desenvolvimento constante, existem planos para incorporar weSpendmoney com o telemóvel (por exemplo para consultar o saldo), traduzir para várias línguas, criação de widgets para o Netvibes e o Yahoo!Widgets (de forma a interagir com o site a partir do desktop) e a possibilidade de importar transacções directamente dos bancos. No fundo tornar a vida dos utilizadores cada vez mais facilitada.

No início de Novembro deverá estar disponível a versão alpha para todas as pessoas que pedirem no site oficial (<http://www.wespendmoney.com>), é uma maneira de poder participar na construção da aplicação testando e contribuindo com sugestões ou críticas à forma como está desenvolvida.

Assim concluo, que este projecto tem sido um desafio muito enriquecedor e espero que inspire outros a participar em experiências semelhantes.

Quem é Pedro Sousa?

Pedro Sousa é Programador Senior na Accenture com vários anos de experiência em aplicações web e fundador do grupo "ruby<<portuguese" (ruby-pt.org) e nos tempos livres escreve para no seu blog (reinventar.com). 

Algoritmia e Estruturas de Dados



José Braga de Vasconcelos
João Vidal de Carvalho

Editora: CentroAtlantico.pt

Colecção: Tecnologias

Páginas: 328

1ª edição: Setembro de 2005

ISBN: 989-615-012-5

Algoritmia e Estruturas de Dados é um livro destinado a todos os interessados nas ciências da computação. Este livro foca-se essencialmente na algoritmia, como forma de resolução de problemas e como auxiliar à criação de programas de computador para aplicar essas soluções. São também muito focadas diversas estruturas de dados recorrentes actualmente. Por fim, é feita uma breve introdução a duas populares linguagens de programação, C e Java.

O livro começa com uma detalhada introdução ao mundo da algoritmia e à modelação de problemas. É relacionada a algoritmia com a engenharia de software, são apresentadas as suas componentes, algumas linguagens que permitem a sua representação. Também são mencionados dois diferentes métodos de concepção algoritmica, os conhecidos iteração e recursão. A seguir, são apresentados os tipos de estruturas de dados utilizados em computação e todas as características da linguagem (pseudocódigo) que será utilizada para a representação dos algoritmos propostos ao longo do livro.

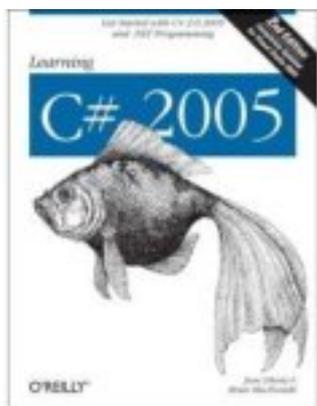
Grande parte do livro é dedicada à apresentação e conseqüente proposta de algoritmos referentes a variados problemas recorrentes em computação, como ordenação e pesquisa em listas, e referentes

à manipulação de importantes estruturas discretas como pilhas, filas, listas encadeadas simples e circulares, árvores e grafos. Todas estas estruturas de dados são bem explicadas, com o auxílio de esquemas simples e descrições detalhadas.

Finalmente, são introduzidas duas linguagens de programação muito utilizadas actualmente. É dada mais ênfase à linguagem C e às suas características para a representação e controlo de dados, sendo então transcritos alguns algoritmos anteriormente referenciados, nesta linguagem. São depois apresentadas as características únicas de Java, seguidas de exemplos de utilização de classes e da definição de tipos de dados abstractos.

Algoritmia e Estruturas de Dados é um livro bem conseguido, escrito por dois professores experientes em diversas áreas da ciência da computação. Este livro poderá servir tanto para entrar no mundo da algoritmia, como de referência para trabalhos futuros. Peca apenas por alguma complexidade, tornando-se assim apenas recomendado a quem já possuir algum "background" em programação de computadores. Mas não deixa de ser um óptimo livro para estudantes da área de informática, matemática aplicada e/ou cursos de engenharia. 

Learning C# 2005



Jesse Liberty
Brian MacDonald

Editora: O'Reilly Media
Linguagem: Inglesa
Páginas: 552
2ª edição: Fevereiro de 2006
ISBN: 0-596-10209-7

Actualmente, são muitos os utilizadores que aderem ao mundo da programação, quer para fins profissionais, quer para diversão ou como hobby. Muitos destes interessados deparam-se, de início, com a discutível e inevitável pergunta: “Que linguagem devo aprender primeiro?” Há diversas opiniões sobre o tema, mas uma grande parte dos programadores diria que C#.NET é uma linguagem poderosa e, simultaneamente, ideal para a iniciação.

É aqui que entra o livro Learning C# 2005. Destinado principalmente a utilizadores sem qualquer experiência nesta área, a referida publicação permite ao leitor construir bases sólidas, tanto nos conceitos teóricos do .NET Framework ou de Object Oriented Programming como nas aplicações desses conceitos para a construção de aplicação mais estáveis e funcionais. Através de exemplos de código, dicas e conselhos, a linguagem é apresentada ao iniciante de forma suave, permitindo a assimilação das ideias e a sua aplicação em contexto prático.

O texto é escrito num vocabulário muito acessível, sendo que todos os termos técnicos que o leitor possa não conhecer são explicados em detalhe. E é fazendo uso desta mesma acessibilidade que o livro consegue orientar e tornar perceptíveis mesmo os conceitos mais complexos.

Outro dos pontos a favor é o seu detalhe, permitindo que mesmo os iniciantes que programam há pouco tempo possam, depois da leitura, construir programas muito completos, utilizando o poder de todas as ferramentas e funcionalidades que lhes são oferecidas de forma prática e eficiente. O livro cobre quase tudo sobre a sintaxe e estrutura da linguagem, bem como as boas práticas no desenvolvimento de aplicações. Há um grande enfoque nos conceitos e aplicação do paradigma OO, uma das grandes vantagens do C# em relação a outras linguagens.

O desvio da atenção do leitor para temas como o aspecto visual e a apresentação dos controlos nas janelas não é problema, uma vez que todos os temas são abordados através da utilização de uma consola. Outro dos pontos a enaltecer é a apresentação de sumários, questionários e exercícios no fim de cada capítulo, que o leitor deve tentar resolver para certificar-se de que percebeu toda a informação.

O código dos projectos apresentados no livro está disponível na Internet, em <http://www.libertyassociates.com/pages/files/LearningCS2eSource.zip>. Fica então a ideia de que o livro é uma boa escolha para a iniciação em C#, mas para os que já têm experiência nesta linguagem outros livros, como Programming C#, dos mesmos autores, são mais aconselháveis. 



apatana é um IDE direccionado para as linguagens web: JavaScript, HTML, CSS; com características muito úteis para os programadores. Está disponível também como plugin para o conhecido Eclipse.

<http://www.aptana.com>

DistroWatch é um conhecido portal sobre distribuições de sistemas operativos Linux e BSD. Contém as últimas novidades e várias informações essenciais sobre as distribuições, assim como TOPs de popularidade.

<http://distrowatch.com>

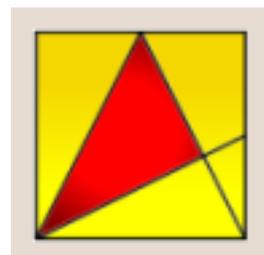


opensourceCMS possibilita uma demonstração de dezenas de CMSs (Content Management Systems) grátis e livres, destinados para portais, blogs, fóruns, wikis, entre muitos mais.

<http://www.opensourcecms.com>

MathsChallenge é a casa do Projecto Euler, um projecto que consiste na resolução de dezenas de diferentes problemas matemáticos, recorrendo à programação. Possui também estatísticas sobre as linguagens e os países dos participantes, sendo possível observar que ainda tem pouco peso em Portugal.

<http://mathschallenge.net>

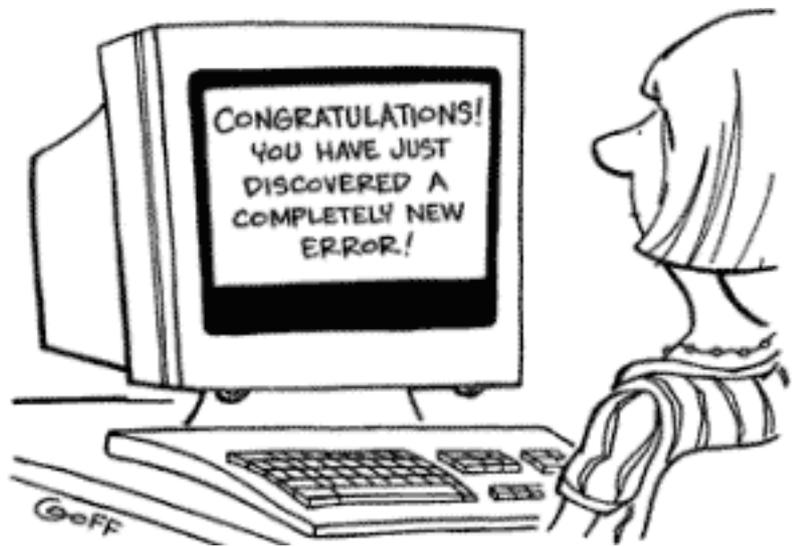




Mais um dos famosos erros



Foi encontrada a tecla mais procurada de todas



Uma descoberta inovadora



O novo modding portátil

Queres participar na revista PROGRAMAR? Queres integrar este projecto, escrever artigos e ajudar a tornar esta revista num marco da programação nacional?

Vai a

www.revista-programar.info

para mais informação como
participar
ou então contacta-nos por

revistaprogramar@portugal-a-programar.org

Precisamos do apoio de todos
para tornar este projecto ainda
maior...

contamos com a tua ajuda

Estamos sensivelmente a dois meses do fim do ano e o Portugal-a-Programar continua a tentar inovar em várias áreas ao mesmo tempo: nas que já estamos inseridos tivemos sucesso, nas que estão a ser planeadas esperamos ter ainda mais. Contamos com um fórum movimentado, com aproximadamente 2200 utilizadores, o que faz com que estejamos à vontade para nos concentrarmos noutros voos.

A revista bateu todos os recordes de estatísticas com a edição anterior e vamos certamente bater muitos mais com esta edição. Queremos fazer com que esta edição chegue a ainda mais interessados em programação, de modo a que possam conhecer o nosso fórum e o bom ambiente da nossa comunidade.

Em relação aos voos do futuro de que há tanto tempo tenho vindo a falar, podemos desde já adiantar mais alguns pormenores: vamos ter uma plataforma de e-learning a funcionar até ao fim de 2006 e esperamos arrancar rapidamente com uma televisão online técnica. Para que todos possam participar nestes projectos, devem dirigir-se ao fórum do P@P e ir acompanhando as novidades que todos os dias vão surgindo.

Estes projectos só se realizarão com a boa vontade e entrega de cada utilizador. Precisamos de gente capaz de criar bons conteúdos, de modo a que estes novos projectos tenham sucesso. Contamos com o vosso contributo. Vemo-nos pelo P@P..

www.portugal-a-programar.org