

# PROGRAMAR

A REVISTA PORTUGUESA DE PROGRAMAÇÃO

Revista nº18 - Fevereiro de 2009

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

## Cloud Computing

Artigo sobre uma das mais importantes tendências no futuro da informática

## Subversion

Controlo Total sobre o Software - 2.ª Parte

Continuação do tema de capa da edição anterior, sobre este sistema de controle de versões

## Packet Capture em Java com JPCAP

Utilizar a API Packet Capture para captura de pacotes de rede.

3º Aniversário da  
Revista PROGRAMAR!

3

## Índice

- 3 **notícias**
- 4 **tema de capa**
  - Cloud Computing
- a programar**
- 10 - Interacção Python-MS Office
- 14 - Subversion: Controlo Total sobre o Software - 2.ª Parte
- 22 - Package Capture em Java com JPCAP
- 28 - Algoritmo de Verhoeff
- segurança**
- 34 - Fundamentos de Segurança em Redes

## equipa PROGRAMAR

### coordenador

Miguel Pais

### coordenador adjunto

Joel Ramos

### editor

Pedro Abreu

### redacção

Vítor Tomaz  
Fernando Martins  
Miguel Rentes  
Fábio Correia  
Ciro Cardoso

### colaboradores

António Silva  
David Ferreira  
Daniel Correia  
João Matos  
José Fontainhas

### contacto

revistaprogramar  
@portugal-a-programar.org

### website

www.revista-programar.info

### issn

1647-0710

## O começo de uma nova etapa

Em Julho de 2007, já lá tão longe no tempo regular, mas tão perto na escala temporal da revista Programar, há portanto 9 edições, quase 2 anos, algo aconteceu na revista Programar. A sua primeira grande mudança havia ocorrido e a sua estabilidade seria posta à prova nas futuras edições. Isto porque uma das principais figuras deste projecto, o seu primeiro coordenador Sérgio Santos, acabava, não de abandonar, que projectos destes não se abandonam, mas sim passar o seu cargo juntamente com todos os seus conhecimentos acumulados ao longo de mais de ano e meio à frente deste projecto... a mim.

Quem era eu? Melhor, quem éramos todos nós quando este projecto começou? Nada mais que simples jovens no seu mais puro estado de amorismo (em que ainda nos encontramos) mas com uma vontade que os fazia vingar na realização dos seus projectos. Este foi um deles. Quem diria que rapazes ainda não na sua maioridade conseguiriam a perseverança, a vontade, ou simplesmente a carlice que faltou a muita gente grande. Por essa altura referia também que a revista Programar era um projecto composto de mudança. Neste, coordenadores não saem ou entram por falhas que tenham ocorrido, problemas, ou discussões, a mudança simplesmente ocorre porque é assim que deve ser, e a perpetuação de alguém à frente de um projecto como este não apenas ao fim de algum tempo começa a trazer a não inovação que não se quer, como rapidamente o "síndrome do fundador" se estabelece e entorpece inevitavelmente em seu redor a vista às mentes que doutro modo mais longe poderiam ver. Isto tudo tem uma razão, não é que as capacidades de alguém que antes traria mudança ao projecto tenham de súbito desvanecido, é apenas o confirmar das exigências de uma vida que cada vez deixa menos tempo para todos os hobbies, de modo a prosseguir essa carreira que em hobby começou.

É portanto, não com pena, mas como satisfação que anuncio que chegou à altura do Joel Ramos e do Pedro Caldeira, há praticamente tanto tempo neste projecto como eu, tomarem as tarefas que lhes sempre estiveram destinadas, e as conduzirem da maneira que estes longos meses de aprendizagem lhes ensinaram. Tenho a certeza que conduzirão este projecto com saber e dedicação, e que trarão todas as inovações que sabemos serem possíveis mas que, tal como este projecto na sua gestação, precisam de um pequeno salto de fé para que se possam realizar.

Há de facto ainda muito a ser possível. Quem sabe um dia a distribuição de umas quantas edições em papel seja possível. Quem sabe se um dia cada estudante não gostaria de ter escrito pelo menos um artigo para este projecto. Quem sabe se este projecto até não poderá evoluir ou mudar de rumo de modo a se adaptar a outros meios que façam mais sentido nessa altura que o actualmente adoptado. Mas para tudo isto ser possível, terão de continuar a haver pessoas cujos rostos se podem ver nas páginas seguintes, que participam e ajudam estes que agora coordenam a realizar a árdua tarefa que têm em mãos.

Espero que, da minha parte, tenha cumprido as vossas expectativas. Já os novos coordenadores tenho a certeza que cumprirão.

### COORDENADOR



Coordenador Adjunto desde a 4ª edição, é actualmente o Coordenador da Revista Programar. Frequenta o 2º ano do curso de Engenharia Informática e de Computadores no IST.

*Miguel Pais*

## Sapo Kids põe miúdos a programar sozinhos - Sistema Scratch apresenta, neste mês, ferramentas criativas às crianças

Imagine-se que a professora pede um trabalho sobre vulcões. Recorrendo ao sistema Scratch, integrado no Sapo Kids, que será lançado este mês, a criança pode criar um vídeo animado de forma a ilustrar a deslocação da lava.

Apresentado com mais uma janela do portal Sapo, a área dos miúdos oferecerá um painel de utensílios inovadores pensados especialmente para pequenos internautas entre os quatro e os 11 anos. "Trata-se de um espaço inteiramente dedicado a conteúdos pedagógicos e educativos", sublinha fonte da PT. O Sapo Kids permite acesso a múltiplos manuais escolares com respectivas tarefas, note-se, mas é a inclusão do sistema Scratch em português a sua grande mais-valia. Para exemplificar a sua utilidade a mesma fonte deu o exemplo do vulcão.

Criado pelo Instituto Tecnológico de Massachusetts (MIT) e inaugurado em Maio, o Scratch fornece gratuitamente o bê- a-bê da programação informática. "Apresenta comandos de fácil utilização, com que se consegue montar imagens", explica a mesma fonte. A adesão das crianças norte-americanas tem sido de tal ordem que a cada dois minutos tem nascido um novo projecto.

Pela primeira vez noutra língua, a sua importação para Portugal prevê ainda instalação no computador Magalhães. A equipa da PT que preparou o modelo nacional preparou, inclusive, alguns ajustes para esse efeito. Por ser o primeiro país a adaptá-lo, Portugal serve também de laboratório para futuras aplicações do sistema noutros países. O contrato com a MIT contempla ainda os direitos de distribuição para os todos países de língua portuguesa.

Outra das suas vantagens é a garantia que dá aos pais. "Tem controlo parental nos acessos aos messenger e e-mail, definindo estes a lista de nomes para quem os filhos podem contactar".

in Jornal de Notícias

## Amazon apresenta Kindle 2

A Amazon apresentou esta segunda-feira a nova versão do seu leitor de livros electrónicos. O Kindle 2 é mais fino e leve, apresentando nove milímetros de espessura e 300 gramas de peso. O preço mantém-se igual ao da versão anterior, 359 dólares.

Nas melhorias ao nível do design incluem-se o acabamento de metal na parte de trás, assim como pequenos botões de navegação, tanto do lado esquerdo quanto direito. Um joystick substitui o "volante" scroll da primeira versão do dispositivo.

Segundo a Amazon, o seu novo leitor de ebooks apresenta uma autonomia melhorada em 25 por cento, assim como maior capacidade de armazenamento, que cresceu para os dois gigas. O Kindle 2 tem igualmente uma opção para converter texto em voz, lendo livros aos seus utilizadores.

O ecrã mantém-se nas seis polegadas, com uma resolução de 600 por 800 pixels.

in Sapo Tek

## Windows 7 a 3 de Outubro

Os últimos dias têm sido ricos no que se refere a novidades acerca do Windows 7. A estratégia de mercado a utilizar tem sido uma constante no discurso da Microsoft e, actualmente, já se sabe que o próximo sistema operativo deverá sair a 3 de Outubro deste ano e em seis versões diferente.

A notícia surge com base na informação colocada no blog de Lotta Bath, da equipa de parceiros da Microsoft Suécia, onde o mesmo apresenta o roadmap até ao lançamento do software. Segundo o texto, que de acordo com o autor é confirmado pelo product manager Windows Michael Bohlin, o próximo sistema operativo terá uma Release Candidate a 7 de Abril deste ano, enquanto que a RTM do Windows 7 tem deadline marcado para 3 de Outubro. O lançamento de uma RC2 não faz parte dos planos da empresa.

Esta notícia surgiu um dia após a Microsoft anunciar que o próximo sistema operativo estará disponível em seis versões diferentes: Starter - para netbooks -, Home Basic, Home Premium, Professional, Enterprise e Ultimate Edition.

in Sapo Tek

# Cloud Computing

## Introdução

O Cloud Computing é uma das mais importantes tendências tecnológicas dos próximos tempos. Mas o que é afinal cloud computing? Não existe ainda uma definição clara, se quiser um exemplo disso faça esta pergunta a cinco pessoas ligadas à tecnologia e provavelmente irá obter cinco ou até mais respostas diferentes.

Existem opiniões que dizem que tudo o que está fora da firewall é cloud computing, ou seja, um simples webservice pode ser considerado cloud computing. Esta resposta até faz sentido porque na realidade o processamento desse webservice ocorre fora do nosso computador. No entanto, cloud computing é muito mais do que isso.

Essencialmente trata-se de afastar completamente a computação e os dados dos computadores de secretária e portáteis, e simplesmente mostrar ao utilizador os resultados da computação que ocorre numa localização centralizada (grandes datacenters) e é transmitida via Internet para o ecrã do utilizador. Mais ainda, é potenciar a interacção entre o software, que neste paradigma reside na nuvem, através de serviços.

Estaremos então a voltar à época dos terminais "estúpidos"? Iremos tentar perceber a resposta a esta pergunta mais à



frente neste artigo, mas neste momento torna-se necessário perceber as mudanças que ocorreram no passado para perceber o porquê dessas mudanças no futuro.

Olhando um pouco para a história dos computadores, não propriamente a partir do primeiro computador no sentido do termo (computador humano, pessoa que realizava

cálculos numéricos), mas sim a partir do primeiro computador electrónico, vamos tentar perceber as mudanças nas arquitecturas dos sistemas e as razões que levaram a essas mudanças.

## História

Os primeiros computadores eram máquinas enormes, muito pesadas e de elevado custo. Sempre que alguém pretendia realizar alguma computação tinha que se deslocar a esse computador, do tamanho de uma sala, para a poder realizar. O que é importante reter é que a computação começou, por razões económicas, com uma arquitectura de centralização da informação.

Mais tarde surgiram os terminais. O uso de terminais já não obrigava a que o operador se deslocasse ao computador para poder realizar operações e permitia ainda que estivessem várias pessoas a usar o computador em regime de partilha de tempo de computação. A arquitectura continua centralizada, todas as operações são realizadas no computador, sendo o terminal apenas um interface para aceder ao computador.

À medida que a Lei de Moore se vai manifestando, o tamanho e custo dos computadores desce e torna-se prática comum a utilização de computadores pessoais visto que já era economicamente viável ter um computador no escritório ou até em casa. Com o uso de computadores pessoais passa-se a utilizar uma arquitectura completamente descentralizada. Neste momento poderia optar-se pelas duas arquitecturas, a centralizada que ainda oferecia um custo mais baixo em termos de custo por utilizador, ou um sistema completamente descentralizado que oferecia uma maior autonomia e melhor interface aos utilizadores.

Posteriormente surge a arquitectura cliente-servidor em que existem computadores em ambos os lados. Algumas partes do programa correm no computador-servidor e outras correm no computador-cliente. Isto causa um aumento da complexidade do software dada a necessidade de comunicação e sincronização entre as partes. Estamos agora numa arquitectura mista.

Entretanto surge o primeiro Web browser que apenas mostrava texto e praticamente todo o processamento era realizado no lado do servidor. Voltamos novamente a uma arquitectura completamente centralizada. As capacidades deste tipo de aplicações eram muito limitadas. Os Web browsers evoluíram, sendo já é possível correr código de aplicações dentro destes, o que levou e leva a que apareçam cada vez mais aplicações Web em detrimento de aplicações desktop.

Esta arquitectura cliente-servidor oferece algumas

vantagens tecnológicas que naturalmente se tornam em vantagens económicas. A centralização de todos os dados no servidor é um exemplo disso, que leva a um aumento da segurança e à redução nos custos de manutenção.

Como pudemos observar, certa parte destas mudanças aparecem por razões económicas. Se uma arquitectura diferente apresenta as mesmas funcionalidades a um preço mais reduzido tende a ser cada vez mais adoptada.

## Motivação

Chegamos agora à parte mais importante. Com o aparecimento de Web browsers com cada vez mais capacidades e cada vez mais cumpridores dos standards, e com as capacidades para acesso à Internet que existem, torna-se cada vez mais possível a ideia de afastar completamente a computação e os dados dos computadores pessoais e movê-los para a nuvem.

Na base desta mudança estão, como mencionado anteriormente, a evolução tecnológica que agora existe e novamente as razões económicas, nomeadamente com a redução de custos que essa mudança pode proporcionar. Vamos agora dar uma vista de olhos nas próximas tendências na área das tecnologias de informação para tentar perceber se esta mudança terá ou não condições para acontecer.

Prevê-se que nos próximos anos, três das principais tendências na área das tecnologias de informação sejam a mobilidade, o software como serviço (SaaS) e o hardware como serviço (HaaS).

A mobilidade significa simplesmente que os notebooks vão ficar em casa. Ou seja, vamos passar a levar connosco dispositivos mais leves, mais pequenos e com a mesma eficiência. Esses dispositivos compactos terão preços relativamente baixos e serão munidos de aplicações Web.

## Software as a Service

Software como serviço significa que muitas das aplicações de negócios serão contratadas no modelo serviço em vez de licença de software. No modelo SaaS a empresa paga na proporção em que usa o software. Essa é uma diferença fundamental em relação ao preço fixo de uma licença. Com o suporte de empresas como Oracle, SAP e Microsoft e de líderes da internet, como Google e Amazon, o modelo de desenvolvimento e distribuição SaaS tende a crescer muito nos próximos anos.

Em vez de instalar software nos seus próprios servidores, as

empresas contratam o fornecimento desse software como um serviço. E em vez de pagarem uma licença de software, pagam mensalmente um custo por utilizador ou pelas horas em que os seus colaboradores estiveram a usar a aplicação. A interacção com este software é realizada da mesma forma que se realizam as interacções com as aplicações empresariais modernas, ou seja, através de um Web browser. As contas de webmail são um bom exemplo de um software muito utilizado e fornecido como serviço. Hoje em dia, praticamente todas as aplicações podem ser fornecidas como serviço.

Esta ideia não é nova, pois no passado já existiram Application Service Providers. No entanto, esta arquitectura não teve grande sucesso porque esses fornecedores dedicaram servidores e programas a empresas individualmente. O custo de operação e manutenção individual de cada um desses servidores impossibilitou o sucesso desta arquitectura. Os fornecedores de software como serviço trabalham com uma arquitectura diferente: existe apenas uma instância do software que serve os vários clientes. Apesar de cada cliente ter apenas acesso aos seus dados e a uma versão personalizada da aplicação, existe apenas uma instância desse software a correr no servidor. Esta arquitectura tem algumas vantagens tais como a redução de recursos ocupados no servidor e a redução dos custos de manutenção. Cada instância de um software por menor que seja requer logo à cabeça alguma memória e algum processamento. Essa necessidade multiplicada por muitos clientes pode tornar considerável esse custo inicial. A arquitectura de que vos estou a falar reduz substancialmente essa necessidade dado que apenas existe uma instância, embora mais "pesada", do software. Como já foi referido, a manutenção está, neste caso, muito mais facilitada dado que todas as correcções de erros e novas funcionalidades ficam imediatamente disponíveis para todos os utilizadores.

Os fornecedores disponibilizam ainda formas de os utilizadores poderem personalizar a sua "versão" da aplicação e até a partilha dessas personalizações com outros clientes. A maioria dos clientes deste tipo de fornecedores são pequenas e médias empresas, mas a constante complexidade e custo de manutenção de aplicações internamente faz com que o número de clientes destes fornecedores tenda a aumentar e a expandir-se a empresas de maior dimensão.

## Hardware as a Service

Hardware como serviço significa que a adopção de novas tecnologias de informação vai deixar de passar por investir grandes quantias de dinheiro com equipamentos, passando a adquirir a sua infra-estrutura como serviço. A banda larga de alta velocidade vai permitir colocar a infra-estrutura em

grandes datacenters com a mesma rapidez de resposta que teria com uma solução local.

A possibilidade de adquirir hardware como serviço já existe há muito tempo através dos fornecedores de alojamento. No entanto esta opção enfrenta alguns problemas complexos tais como a escalabilidade. Normalmente nestes casos é alugado um servidor dedicado o que leva a estarmos limitados à capacidade desse servidor.

Vamos agora pensar do ponto de vista empresarial. Imagine que tem uma empresa e compra um servidor para dar resposta às necessidades da sua empresa. Passado algum tempo a sua empresa está a ter cada vez mais sucesso no mercado, o seu servidor já não dá resposta às necessidades e tem que adquirir outro e depois mais outro... Por alguma razão o país entra em recessão e o seu volume de negócios desce bastante. Ora, para além dos problemas que a recessão lhe está a trazer tem ainda que se preocupar em acabar de pagar os servidores que teve que adquirir e que agora já não lhe fazem a mínima falta.

Outro exemplo da mais-valia que o hardware como serviço trás às empresas é a capacidade de resposta a picos temporários. Imagine que a sua empresa tem que processar um grande volume de dados no final do mês e tal não pode ser feito de outra forma. Provavelmente terá que adquirir tecnologia de servidor que estará a dar rendimento apenas no final de cada mês.

Como já pudemos perceber trata-se de uma arquitectura altamente escalável porque permite aumentar ou diminuir a capacidade do hardware numa questão de minutos. Além disso é muito flexível porque dá a possibilidade aos utilizadores de escolher quais as capacidades que cada instância terá. A fiabilidade é outra grande arma dado que tira partido de infra-estruturas gigantes, como por exemplo a infra-estrutura da Amazon ou da Google. Um dos exemplos mais conhecidos deste tipo de serviço é o Amazon EC2 (Amazon Elastic Compute Cloud).

## Desvantagens e/ou Preocupações

Até este momento só vimos vantagens neste paradigma, mas naturalmente também existem desvantagens. Algumas podem parecer desvantagens mas tratam-se apenas, até prova em contrário, de preocupações. O primeiro inconveniente tem logo a ver com a necessidade de conectividade, pois sem conectividade não é possível trabalhar. As ligações com baixa largura de banda também poderão limitar bastante o funcionamento deste tipo de serviço caso a informação a transportar entre as partes seja de elevada dimensão.

Outra das limitações a apontar trata-se da privacidade e de

toda a segurança que tem que existir. Existe alguma desconfiança quanto a deixar a carga de outra empresa a tarefa de guardar dados importantes, quer sejam pessoais ou empresariais. Talvez este tema não se trate de uma desvantagem mas, apenas e só, de uma preocupação. Empresas que forneçam estes serviços vão ter o seu sucesso ou insucesso muito intimamente ligados à sua reputação, e portanto os mecanismos de segurança serão uma das suas principais preocupações.

Outras existirão logo à partida mas que tenderão a desaparecer com a evolução, tais como a menor velocidade, menor capacidade e menos funcionalidades que este tipo de software ainda tem em relação ao software desktop. Existem ainda algumas questões muito importantes por resolver. Quem é o detentor legal da informação e da propriedade intelectual? É quem contrata o serviço ou quem o disponibiliza? É possível que o fornecedor negue ao utilizador acesso à sua própria informação? Estes e outros assuntos ainda estão a ser debatido por empresas, advogados e universidades.

## Cloud Platforms

Uma das componentes mais importantes desta tendência é a chegada das plataformas na nuvem. Este tipo de plataformas permite aos programadores desenvolver aplicações que correm na nuvem, usar serviços fornecidos pela nuvem, ou ambos. Para compreender esta importância pense como as aplicações são desenvolvidas actualmente. Quando uma equipa de programadores cria uma aplicação, a maioria das coisas que essa aplicação necessita já existem, desde um sistema operativo que fornece o suporte básico para executar essa aplicação a outros computadores que oferecem serviços como, por exemplo, o armazenamento em base de dados. Se para desenvolver um software tivéssemos que criar toda esta plataforma base existiriam muito menos aplicações nos dias de hoje.

Da mesma forma, se todas as equipas de desenvolvimento que pretendessem criar uma aplicação para correr na nuvem tivessem primeiro que criar a sua própria plataforma também não veríamos muitas aplicações a correr na nuvem. Felizmente, os fornecedores estão cada vez mais a responder a este desafio e já existem algumas plataformas de desenvolvimento na nuvem. Para conseguirmos entender melhor as plataformas na nuvem vamos tentar perceber como é que funcionam as plataformas locais.

## As plataformas locais

Independentemente de uma plataforma ser local ou na nuvem, pode ser vista como englobando três componentes: um software base para interagir com o hardware existente,

um grupo de serviços fornecido pela infra-estrutura e um conjunto de serviços fornecidos por outras aplicações.

Nos primeiros tempos da computação, a plataforma não era mais do que um conjunto de software base. Quase todas as aplicações utilizam algum software base na máquina onde correm, o que tipicamente inclui algumas funções de suporte, tais como um sistema operativo base, armazenamento standard e bibliotecas de tipos.

Nos anos 80 e 90, com a disseminação da computação distribuída, foram adicionados serviços disponibilizados pela infra-estrutura, com o armazenamento remoto em ficheiros e bases de dados, a integração e a identidade a tornarem-se muito utilizados.

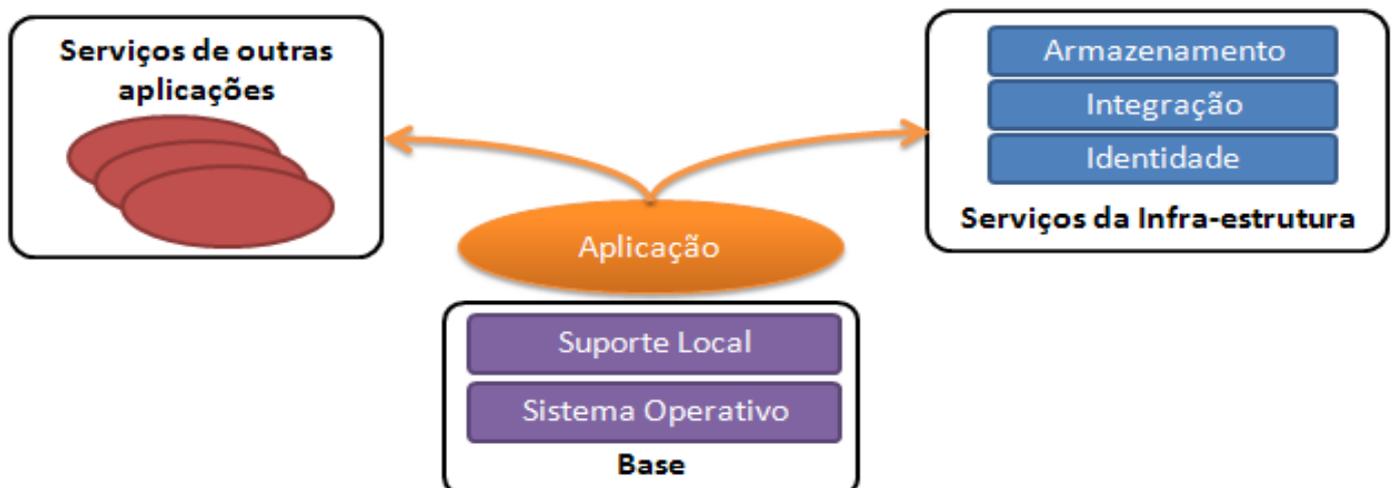
Hoje em dia apesar de as aplicações existirem acima de tudo para fornecer serviços a utilizadores finais, com a chegada das aplicações orientadas para serviços, os serviços fornecidos por outras aplicações locais tornaram-se também parte da plataforma.

Não fazendo directamente parte da plataforma, as ferramentas de desenvolvimento são outra importante parte desta história porque ajudam os programadores a construir aplicações usando as três componentes da plataforma.

terão que lidar com aplicações de uma escala bem maior, onde o número de utilizadores será também maior.

Vamos agora olhar para a camada base e identificar algumas diferenças estruturais. Começando pelo sistema operativo, serão fornecidas instâncias de sistemas operativos a correr sobre máquinas virtuais. A utilização de máquinas virtuais permite que o utilizador possa escolher as capacidades que quer ter na nuvem e, mais importante ainda, permite expandir muito rapidamente essas capacidades em caso de necessidade. Não se esperam grandes limitações quanto ao sistema operativo a utilizar nas máquinas virtuais, e numa primeira fase será Linux o sistema operativo mais utilizado. No entanto a Microsoft já está a desenvolver a sua plataforma para a nuvem, a plataforma Azure, sendo difícil neste momento prever se conseguirá recuperar as desvantagens que parece ter neste momento em relação a Linux.

As plataformas na nuvem actuais não fornecem as mesmas opções que uma plataforma local. Os programadores estão limitados a poucas, ou até apenas a uma, formas de poder obter suporte local. Relembro que ao estarmos a falar de suporte local estamos a falar de, por exemplo, acesso a um servidor de bases de dados a correr na própria máquina ou de frameworks sobre as quais as aplicações correm. Estas limitações também são, de alguma forma, consequências da

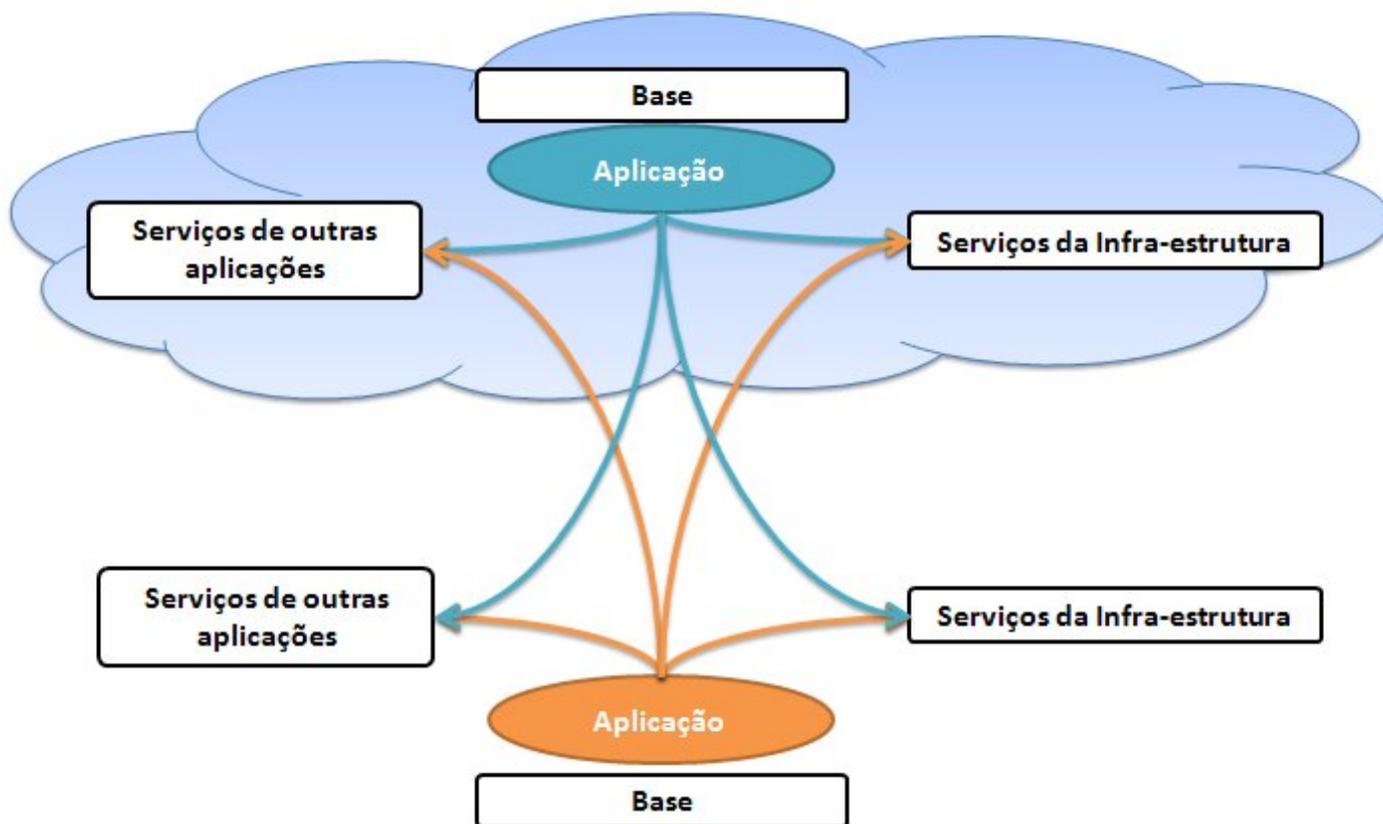


## As plataformas na nuvem

O modelo de plataforma usado na nuvem é muito semelhante ao modelo que acabámos de ver. No entanto, e dada as suas diferentes características, irá funcionar de forma diferente em cada camada. Percebemos facilmente isso se pensarmos na enorme necessidade de escalabilidade. Enquanto as plataformas locais lidam, no máximo, com aplicações de nível empresarial, as plataformas na nuvem

necessidade de escalabilidade mas também permitem aos fornecedores focarem-se no desenvolvimento e optimização desse tipo de suporte.

Antes de falarmos sobre as características dos serviços fornecidos pela infra-estrutura na nuvem e por outras aplicações na nuvem é importante perceber a interacção que poderá existir entre as duas plataformas



As plataformas locais poderão tirar partido dos serviços fornecidos pelas plataformas na nuvem, tal como as plataformas na nuvem poderão tirar partido dos serviços fornecidos pelas plataformas locais. A mudança para plataformas na nuvem será em muitos casos uma mudança progressiva. As aplicações locais vão cada vez mais inter-operar com serviços e aplicações que passarão a existir na nuvem até que um dia todas, ou pelo menos a maioria, estarão já a funcionar completamente na nuvem.

Quanto a serviços fornecidos pela infra-estrutura, muitas aplicações locais não tiram partido deste tipo de serviços. Quando falamos de aplicações na nuvem o mesmo poderá acontecer. Dada a normal existência de apenas uma instância na nuvem poderemos ser levados a pensar que funcionalidades como o armazenamento em bases de dados e gestão de identidade passem a ser fornecidos cada vez menos como serviço e passem a fazer parte do próprio software. Isto porque não existe a necessidade de partilhar essa informação com outras instâncias.

Este pode ser um caminho, mas o caminho contrário é ainda mais viável, ou seja, focar o desenvolvimento do software na lógica de negócio e deixar a resolução desse tipo de tarefas a cargo de serviços que ligamos ao nosso software e dos quais tiramos partido. Vamos pensar, por exemplo, no serviço de

gestão de identidade. O programador não tem que se preocupar na forma de guardar e verificar a identidade dos seus utilizadores, bastando-lhe apenas saber utilizar o serviço que lhe fornece solução para essa e outras preocupações. É uma mais-valia importante. As plataformas que existem hoje em dia na nuvem têm normalmente um serviço de gestão de identidade associado. Trata-se de uma limitação mas não deixa de ser um serviço muito importante para as aplicações, até porque a interacção com outros fornecedores de identidade é cada vez mais possível.

O serviço de armazenamento de informação na nuvem é um dos mais atractivos, mas mesmo este tem necessariamente que ter uma arquitectura diferente devido às características da nuvem. Isto poderá significar que os programadores poderão tirar partido deste tipo de serviço mas poderá implicar mais algum trabalho no desenvolvimento se o quiserem usar devidamente. Em relação a serviços fornecidos por outras aplicações não parecem existir grandes novidades. No entanto, a cada vez maior adopção de arquitecturas orientadas ao serviço vão potenciar ainda mais a interacção entre as aplicações, sejam elas locais ou residentes na nuvem.

## Conclusão

Como se pode constatar anteriormente, as plataformas não mudam muito ao longo do tempo. No entanto, sempre que aparece uma nova plataforma que apresenta reais vantagens em relação às existentes ela tende a ter um impacto muito grande. As plataformas na nuvem ainda não trazem todas as vantagens que as plataformas locais já disponibilizam mas apresentam sérias vantagens competitivas. A adoção destas plataformas fará parte do nosso futuro enquanto programadores ou simples utilizadores de informática. Fica prometido para uma futura edição uma análise mais aprofundada às plataformas existentes.



## Referências e Bibliografia

Wikipedia - Cloud Computing

HowStuffWorks.com – Cloud Computing

GigaOM – How cloud and utility computing are different

Cloud Computing Journal -What is Cloud Computing?

Linux Magazine - Computing In The Clouds: Setting Expectations

Chapell & Associates - A Short Introduction to Cloud Platforms

InfoWorld - What cloud computing really means

### SOBRE O AUTOR



Vitor Tomaz é estudante de Engenharia Informática e Computadores no ISEL bem como Trabalhador Independente na área de Tecnologias de Informação. Tem especial interesse por Programação Concorrente, Segurança Informática e Aplicações Web.

vitor.tomaz@portugal-a-programar.org

*Vitor Tomaz*

# Interacção Python-MS Office

## Introdução

Interagir de forma automática com as aplicações Excel e Word da Microsoft é algo que permite eliminar tarefas repetitivas e aborrecidas ao mesmo tempo que aumenta a produtividade e reduz a possibilidade de erro humano. Interagir com estas aplicações através de Python é bastante simples, basta algumas linhas de código para ler ou escrever informação nesses documentos.

Antes de mais, é necessário termos o Office da Microsoft correctamente instalados no Windows. De seguida, basta ter o Python 2.6 e a extensão Python Python Win32 Extensions para a versão 2.6. À data deste artigo a extensão está disponível para várias versões de Python, até à versão 2.6, pelo que se estiver a usar versões mais antigas de Python é provável que a extensão a suporte. Espera-se uma versão para Python 3.0 para breve.

## Interacção

Uma vez tudo instalado é necessário compreender o modelo de funcionamento da interacção envolvida. A interacção a partir de um script de Python efectua-se através da tecnologia COM e do Office Object Model, que permite simular a utilização da aplicação por parte de um utilizador humano. Tal facto implica algumas restrições. Dado que a interacção simula um utilizador, é necessário que haja um utilizador com sessão aberta no sistema e que o utilizador não interaja com as aplicações enquanto o script Python faz a sua tarefa, dado que as aplicações serão abertas durante a operação.

Uma vez compreendido o modelo de interacção, a codificação da interacção é bastante simples. Os exemplos seguintes seguem uma ordem lógica apenas por motivos de melhor compreensão do artigo.

Para iniciar a interacção, é necessário recorrer à extensão que foi previamente instalada. Tal obtém-se carregando a biblioteca win32com.client:

```
from win32com.client import Dispatch
```

Para invocar a aplicação recorre-se ao objecto Dispatch, importado no exemplo anterior, tendo como argumento a aplicação que se pretende manipular e recebe-se um handler que contém a ligação à aplicação.

```
xlApp = Dispatch("Excel.Application")
```

Para podermos observar a janela, é necessário torná-la visível, para tal atribui-se o valor 1 à propriedade Visible. Afectar o valor 0 à propriedade torna a janela invisível.

```
xlApp.Visible = 1
```

Para terminar a aplicação recorre-se ao método Quit(), sendo assim fechada a aplicação.

```
xlApp.Quit()
```

Por fim, liberta-se o handler que mantinha a ligação

```
del xlApp
```

## Exemplo

O exemplo seguinte abre o Excel, torna-o visível e por fim termina a aplicação. Para executar o exemplo, garante que não tem o Excel aberto. Provavelmente a execução do exemplo será tão rápida que o Excel quase não será visível no ecrã.

```
from win32com.client import Dispatch

xlApp = Dispatch("Excel.Application")
xlApp.Visible = 1
xlApp.Quit()
del xlApp
```

## Manipular um Excel

O Excel Object Model indica quais os objectos, propriedades e métodos que se encontram disponíveis para efectuar a manipulação de um documento Excel. De forma simples, Application refere a aplicação, Workbook refere um documento, Worksheet refere uma página, Cell refere uma

célula e Range refere um conjunto de células. Assim, para criar um novo documento, basta dizer à aplicação que esta deve adicionar um novo Workbook:

```
xlApp.Workbooks.Add()
```

Criar uma nova folha ao documento é bastante simples, recorrendo ao documento activo, referenciado por ActiveWorkbook, adiciona-se uma nova folha ao mesmo:

```
xlSheet =
xlApp.ActiveWorkbook.Worksheets.Add()
```

Atribuir ou ler valores às células de um folha é igualmente simples, basta referir a célula pela sua coordenada e atribuir-lhe o valor pretendido, ou requisitar o valor actual:

```
# Atribuir valor
xlSheet.Cells(1,1).Value = 'Exemplo'
# Ler valor
print xlSheet.Cells(1,1).Value
```

## Escrever um Excel

Eis um exemplo completo da criação de um documento Excel de nome excel-python-pap.xls na raiz do disco C:. O exemplo encontra-se comentado por forma a ser mais fácil acompanhar o código.

Antes de mais, é necessário efectuar uma ligação do Python ao Excel.

```
# -*- coding: iso-8859-15 -*-
from win32com.client import Dispatch
import datetime
import sys

# Efectuar ligação ao Excel
xlApp = Dispatch("Excel.Application")
xlApp.Visible = 1
```

De seguida trabalhamos com o documento de forma a ficarmos apenas com uma única folha, cujo nome é modificado.

```
# Adicionar um novo documento
xlWbook = xlApp.Workbooks.Add()

# Eliminar todas as folhas excepto a primeira
for i in range(xlWbook.Sheets.Count,
1, -1):
    xlWbook.Sheets(i).Delete()

# Seleccionar a primeira folha
xlWbook.Sheets(1).Select()
xlSheet = xlWbook.ActiveSheet

# Alterar o nome da folha para "Exemplo"
xlSheet.Name = "Exemplo"
```

Passando às células, é fácil a atribuição de valores de vários tipos, desde texto a datas ou até fórmulas do Excel.

```
# Atribuir strings
xlSheet.Cells(1,1).Value = 'Revista
Portugal-a-Programar'
xlSheet.Cells(2,1).Value = 'Python'
xlSheet.Cells(3,1).Value = 'Criada Em'
xlSheet.Cells(4,1).Value = 'Modificada
Em'

# Atribuir valor numérico
xlSheet.Cells(2,2).Value =
str(sys.version_info[0]) + "." +
str(sys.version_info[1])

# Atribuir uma data
xlSheet.Cells(3,2).Value =
datetime.date.today()

# Atribuir uma fórmula
xlSheet.Cells(4,2).Formula = '=Now()'
```

As funcionalidades de formatação estão também disponíveis, permitindo uma visualização mais atractiva ao documento.

Por fim, o documento é guardado e a ligação ao Excel é finalizada. Nota: Note-se que a execução consecutiva deste script fará com que a segunda execução apareça a janela de confirmação de sobrescrita do documento, uma vez que este já existe.

Após a execução do script o documento gerado terá o seguinte aspecto:

	A	B	C
1	Revista Portugal-a-Programar		
2	Python	2,6	
3	Criada Em	23-12-2008	
4	Modificada Em	23-12-2008 15:41	
5			
6			

## Ler um Excel

A leitura de um Excel efectua-se com a mesma simplicidade com que se efectua a escrita. Para uma melhor exemplificação da potencialidade da interacção, este exemplo contém algumas diferenças em relação ao anterior. Assim, neste exemplo, o Excel não é tornado visível, ou seja, a aplicação executa mas a janela nunca é visível para o utilizador. Mesmo que o utilizador possua um documento Excel aberto, o documento lido pelo script nunca é tornado visível. No entanto, o comando de terminar a aplicação é executado e o Excel será fechado.

Nota: O comando de término da aplicação fará com que o Excel termine e feche todos os documentos abertos, mesmo que sejam documentos abertos pelo utilizador.

O script fica em espera caso a aplicação necessite de interacção por parte do utilizador, como por exemplo se o Excel questiona o utilizador se este pretende guardar alterações a um documento. Caso se pretenda que a aplicação não seja terminada, então o comando de término da aplicação não deve ser executado. Também neste exemplo, as referências a alguns objectos do documento Excel serão feitas pelo seu nome, ao contrário do que aconteceu no exemplo anterior onde essas referências eram efectuadas através dos seus indices. Tal é possível porque o modelo suporta estas duas formas de referência para alguns dos objectos.

Eis o exemplo completo da leitura do documento Excel criado no exemplo acima. Este exemplo irá ler o documento e mostrará na shell os valores presentes no documento. De novo, o exemplo encontra-se comentado por forma a ser mais fácil acompanhar o código.

De novo, o exemplo começa por efectuar a ligação ao Excel e de seguida abre o documento.

```
# -*- coding: iso-8859-15 -*-
from win32com.client import Dispatch

# Efectuar ligação ao Excel
xlApp = Dispatch("Excel.Application")

# Abre o documento
xlWbook =
xlApp.Workbooks.Open('C:\\excel-
python-pap.xls')
```



A folha é seleccionada através do seu nome, em detrimento da sua posição.

```
# Seleccionar a primeira folha de nome
'Exemplo'
xlWbook.Sheets('Exemplo').Select()
xlSheet = xlWbook.ActiveSheet
```

A leitura dos valores presentes nas células faz-se de forma bastante simples, sendo o seu valor mostrado na consola.

```
# Ler o cabeçalho
print xlSheet.Cells(1,1).Value

# Ler a versão
print xlSheet.Cells(2,1).Value,
xlSheet.Cells(2,2).Value

# Ler as datas
print xlSheet.Cells(3,1).Value,
str(xlSheet.Cells(3,2).Value)[0:8]
print xlSheet.Cells(4,1).Value,
xlSheet.Cells(4,2).Value
```

Por fim, o documento é fechado sem guardar alterações e a ligação ao Excel é terminada.

```
# Fechar sem guardar alterações
xlApp.ActiveWorkbook.Close(SaveChanges=
0)

# Terminar aplicação
xlApp.Quit()

# Limpar memória
del xlApp
```

Após a execução do script a consola terá o seguinte aspecto:

```
c:\swsetup\scripts>python read_excel.py
Revista Portugal-a-Programar
Python 2.6
Criada Em 12/27/08
Modificada Em 12/27/08 12:09:31
c:\swsetup\scripts>
```

## Controlo Total

Os exemplos anteriores ilustram a simplicidade com que se interage com o Excel a partir de Python. Esta simplicidade está presente mesmo em tarefas menos comuns quando se automatizam operações de manipulação de documento como a leitura e escrita e informação. O exemplo seguinte ilustra como é possível imprimir informação de forma automática e, por brincadeira, interagir com o sintetizador de fala do sistema.

Nota: Este exemplo requer que o sistema possua o sintetizador de fala e uma impressora correctamente instalados. Em certas condições, o Excel questionará se pretende instalar o sintetizador de fala, pelo que é necessário responder afirmativamente de forma a ouvir a frase "Python rules!"

Como sempre, o exemplo começa por efectuar a ligação ao Excel e de seguida abre o documento.

```
# -*- coding: iso-8859-15 -*-
from win32com.client import Dispatch
# Efectuar ligação ao Excel
xlApp = Dispatch("Excel.Application")
# Abre o documento
xlWbook =
xlApp.Workbooks.Open('C:\\excel-
python-pap.xls')
```

A folha é seleccionada por nome e não por índice.

```
# Seleccionar a folha de nome
'Exemplo'
xlWbook.Sheets('Exemplo').Select()
xlSheet = xlWbook.ActiveSheet
```

A selecção da área a utilizar na impressão é efectuada através das coordenadas da folha, seguindo-se o comando de impressão do conteúdo dessa área.

```
# Seleccionar a informação a imprimir
xlSheet.PageSetup.PrintArea = "A1:B4"

# Imprimir as células seleccionadas
xlSheet.PrintOut()
```

De seguida é dada indicação para usar o motor de fala, uma funcionalidade um pouco menos comum.

```
# Falar
xlApp.Speech.Speak("Python rules!")
```

Por fim, a ligação ao Excel é finalizada sem que o documento seja actualizado.

```
# Fechar sem guardar alterações
xlApp.ActiveWorkbook.Close(SaveChanges=
0)

# Terminar aplicação
xlApp.Quit()

# Limpar memória
del xlApp
```

## Interação com Outras Aplicações

Os exemplos deste artigo centram-se na interacção com o Excel, mas é possível interagir com outras aplicações, como é o caso do Word e do Outlook. Essas interacções ficam como exercício lúdico para o leitor.

### SOBRE O AUTOR



Apaixonado pela tecnologia desde cedo, ganhou o gosto pela programação num ZX Spectrum 48K. Licenciou-se em Engenharia da Linguagem e do Conhecimento e encontra-se actualmente a trabalhar na dissertação do Mestrado de Informática na FCUL. Profissional na área das tecnologias e sistemas de informação desde 1999, tem especial interesse em manipulação de dados (ETL), integração de sistemas, limpeza de dados, *business intelligence* (*data mining*, *data warehouse*), inteligência artificial e processamento automático de língua natural.

fernando.martins@portugal-a-programar.org

*Fernando Martins*

# Subversion: Controlo Total sobre o Software - 2.ª Parte

Nesta 2.ª parte do artigo sobre Subversion (a 1.ª parte encontra-se na edição n.º 17), vamos ver como deveremos usar o Subversion no dia-a-dia e também algumas das ferramentas que existem actualmente e que se integram bastante bem com o Subversion. Com estas ferramentas poderemos tirar mais partido do Subversion, uma vez que nos vão fornecer um conjunto de funcionalidades acrescentadas (entre as quais: disponibilidade do código online, autenticação e autorização no repositório, gestão das alterações feitas a cada revisão, facilidade em aferir as diferenças entre as modificações feitas numa cópia local de trabalho e o repositório central de código), como veremos em seguida.

No fim deste artigo apresenta-se bibliografia recomendada para quem desejar extender os seus conhecimentos sobre este sistema de controlo de versões de software, cada vez mais utilizado hoje em dia.

## Ciclo Normal de Trabalho com SVN

### Checkout Inicial

Na maior parte das vezes, começa-se a usar um repositório de Subversion fazendo um checkout ao projecto em questão. Ao fazer isto, é criada na máquina local uma cópia do projecto. Esta cópia contém a HEAD (última revisão) do repositório SVN que é especificado na linha de comandos:

```
$ svn checkout
http://svn.exemplo.pt/repositorio/trunk

A trunk/subversion.dsw
A trunk/svn_check.dsp
A trunk/Makefile
A trunk/configure
A trunk/ficheiroA.c
...
Checked out revision 327.
```

Apesar do exemplo acima fazer checkout da directoria /trunk, pode-se igualmente escolher uma subdirectoría do repositório ao especificar o subdirectório no URL:

```
$ svn checkout
http://svn.exemplo.pt/repositorio/trunk
/documentacao/pdfs
A trunk/documentacao/pdfs/faq.pdf
A
trunk/documentacao/pdfs/userguide.pdf
A trunk/documentacao/pdfs/manual-de-
instalacao.pdf
...
Checked out revision 327.
```

Uma vez que o Subversion usa o modelo copia-modifica-unifica em vez do modelo bloqueia-modifica-desbloqueia, o utilizador pode logo começar a fazer alterações aos ficheiros e directorias na sua cópia de trabalho. Esta cópia de trabalho é igual a qualquer outra colecção de ficheiros e directorias no computador. O utilizador pode editar e alterar os ficheiros, navegar e mover pastas, ou até mesmo apagar a cópia de trabalho inteira e trabalhar numa nova. Apesar de realmente ser uma colecção de ficheiros e directorias como qualquer outra, é preciso sempre avisar o Subversion se se quiser rearranjar qualquer ficheiro ou directoria dentro da cópia de trabalho. Por isso, deve-se sempre usar os comandos `svn copy` ou `svn move` em vez dos comandos normais (`Ctrl-C` ou `Ctrl-X`, por exemplo) para que essas alterações sejam reflectidas a todos os utilizadores quando fizerem um `svn update` nas suas cópias de trabalho.

Cada directoria na cópia de trabalho local contém uma área administrativa, uma sub-directoria chamada `.svn`. Normalmente, um comando de listagem de directórios não mostra esta sub-directoria, mas é no entanto uma directoria muito importante que não se deve nunca apagar ou alterar. O Subversion depende desta directoria para gerir a cópia de trabalho de cada utilizador.

Pode-se também fazer um checkout especificando uma nova directoria que vai passar a ser a nova cópia de trabalho depois do URL do repositório, nos parâmetros do comando de checkout:

```
$ svn checkout
http://exemplo.efacec.pt/repositorio/trunk/documentacao/pdfs directoriaNova
A directoriaNova/faq.pdf
A directoriaNova/userguide.pdf
A directoriaNova/manual-de-
instalacao.pdf
...
Checked out revision 327.
```

Isto coloca a cópia de trabalho numa directoria chamada `directoriaNova` em vez da directoria `trunk` como tínhamos visto previamente.

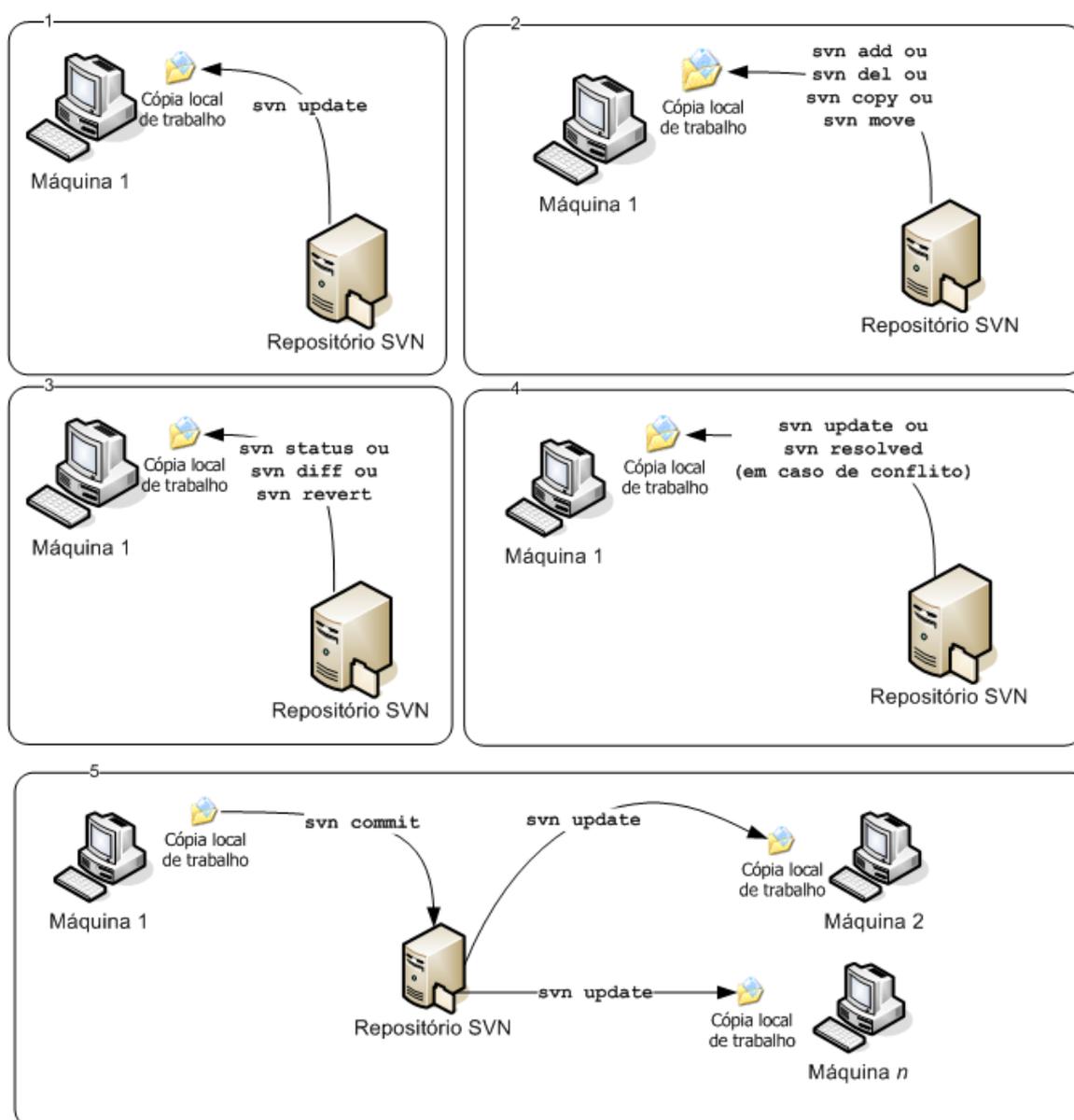
## Ciclo normal de trabalho

O Subversion tem bastantes funcionalidades, opções, possibilidades de uso, mas para uma utilização diária só se irá usar um sub-conjunto limitado das funcionalidades do Subversion. O ciclo de trabalho é tipicamente o seguinte:

- Fazer um update à cópia de trabalho
  - `svn update`
- Fazer as alterações necessárias aos ficheiros da cópia de trabalho
  - `svn add`
  - `svn delete`
  - `svn copy`
  - `svn move`

- Examinar as alterações feitas
  - `svn status`
  - `svn diff`
  - `svn revert`
- Unificar as alterações feitas por outros utilizadores na cópia de trabalho local
  - `svn update`
  - `svn resolved`
- E por fim, e só agora, fazer uma submissão das alterações feitas na cópia de trabalho local para o repositório de SVN
  - `svn commit`

O diagrama seguinte exemplifica melhor o ciclo normal de desenvolvimento com o Subversion.



## Propriedades do Subversion

O Subversion permite criar propriedades com versão nos ficheiros e directorias (com os nomes que se quiser), bem como propriedades sem versão nas revisões. A única restrição encontra-se nas propriedades com o prefixo `svn:` que são propriedades reservadas ao Subversion nesse espaço de nomes (namespace). Enquanto que as propriedades fora deste espaço de nomes podem ser usadas para controlar o comportamento do Subversion, os utilizadores não podem criar ou alterar propriedades do espaço de nomes `svn:`. Estas são:

### • Propriedades com versão

- `svn:executable` - Se está presente num ficheiro, o cliente de SVN fará o ficheiro executável em cópias de trabalho em ambiente Unix; Em ambiente Windows, para ser executável basta ter a extensão `.exe` ou `.bat`.

- `svn:mime-type` - Se está presente num ficheiro, o valor indica o tipo MIME do ficheiro. Isto permite ao cliente de SVN decidir se uma fusão (merging) contextual é possível de ser feita durante um update, e pode também afectar como o ficheiro se comporta quando é procurado via web browser;

- `svn:ignore` - Se está presente numa directoria, o valor é uma lista de padrões de ficheiros sem versão a serem ignorados pelo subcomando `svn status` ou outros; Por exemplo, `"*.log *.dmp core.*"`;

- `svn:keywords` - Se está presente num ficheiro, o valor diz ao cliente de SVN como expandir palavras-chave (keywords) particulares dentro do ficheiro;

- `svn:eol-style` - Se está presente num ficheiro, o valor diz ao cliente de SVN como manipular o fim de uma linha no ficheiro na cópia de trabalho; Por exemplo, CRLF (em Windows) ou CR (em Unix/Linux);

- `svn:externals` - Se está presente numa directoria, o valor é uma lista com vários caminhos (paths) e URLs a que o cliente de SVN deverá fazer checkout; Esta propriedade é muito útil quando se precisa de usar código-fonte de terceiras fontes mas que não precisamos que estejam versionadas no repositório de SVN;

- `svn:special` - Se está presente num ficheiro, indica que o ficheiro não é um ficheiro normal, mas um link simbólico ou outro tipo especial de objecto;

- `svn:needs-lock` - Se está presente num ficheiro, indica ao cliente de SVN para colocar o ficheiro só para leitura na cópia de trabalho, como uma nota de que esse ficheiro deverá ser bloqueado (locked) antes da sua edição (para desta forma ganhar propriedades de escrita e garantir que só um utilizador pode alterar esse ficheiro).

### • Propriedades sem versão

- `svn:author` - Se está presente, contém o nome do utilizador autenticado que criou a revisão (se não estiver presente, a revisão foi feita anonimamente);

- `svn:date` - contém o tempo UTC em que a revisão foi criada, no formato ISO. O valor vem do relógio do servidor;

- `svn:log` - contém a mensagem de log que descreve a revisão; Esta mensagem pode ser alterada após um commit, caso hajam enganos na mensagem que se queria ter colocado no commit;

- `svn:autoversioned` - Se está presente, a revisão foi criada com a funcionalidade de versionamento automático.

## URLs dos repositórios Subversion

Os repositórios de Subversion podem ser acedidos através de vários métodos – localmente no disco (directamente) ou através de vários protocolos de rede (indirectamente). A localização de um repositório, no entanto, é sempre um URL. Em seguida apresenta-se uma tabela que descreve como os diferentes URLs mapeam os diferentes métodos de acesso a um repositório Subversion.

Esquema	Método de acesso
<code>file://</code>	Acesso directo ao repositório (disco local).
<code>http://</code>	Acesso via protocolo WebDAV para o servidor Apache do Subversion
<code>https://</code>	O mesmo que para <code>http:</code> , mas com encriptação SSL (mais seguro)
<code>svn://</code>	Acesso via protocolo <code>svn</code> a um servidor a correr <code>svnserve</code>
<code>svn+ssh://</code>	O mesmo que <code>svn:</code> , mas através de um túnel SSH

Nota: Se se usar o SVN em DOS, as barras devem ser como numa directoria em Linux (/) e não da forma nativa (\). Exemplo: `file:///repositorio/pasta/` e não `file:///repositorio\pasta\`

## Ajuda do SVN

Um dos mais importantes comandos do SVN é o `svn help`. A qualquer momento, um

```
svn help sub-comando
```

descreve a sintaxe, os switches que podem ser usados e o comportamento normal desse sub-comando.

## Instalação do SVN

### Instalação em Linux (distribuição Ubuntu)

A instalação do subversion em Ubuntu é extremamente simples e funcional. Basta fazer numa linha de comandos:

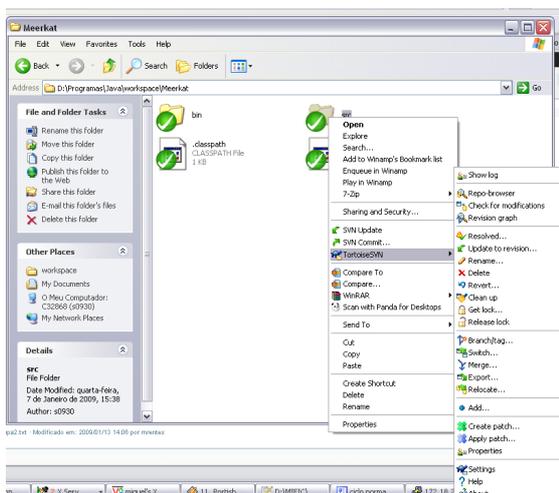
```
sudo apt-get install subversion
libapache2-svn
```

### Instalação em Windows (TortoiseSVN)

A instalação em Windows pode ser feita de duas maneiras: instalando os binários do cliente/servidor de SVN (e passa-se a usar apenas o subversion numa shell DOS, o que não é muito interessante para os utilizadores do Windows), ou instala-se um cliente gráfico para Windows: um dos melhores actualmente é o TortoiseSVN. Vejamos a seguir como instalar o TortoiseSVN (apenas para Windows):

1. Descarregar o TortoiseSVN no site oficial de downloads;
2. Duplo-clique no instalador .msi e resta apenas seguir as instruções apresentadas;
3. Um reboot será necessário para o TortoiseSVN ser correctamente instalado.

Uma das principais vantagens de uso do TortoiseSVN é a integração que tem no Windows explorer (clique no botão direito do rato faz surgir no menu de contexto as opções do TortoiseSVN) independentemente de outros clientes gráficos que estejamos a usar. Outra grande vantagem é a sobreposição de ícones no Windows: nas pastas que sejam de cópias locais de trabalho SVN, os ícones das pastas e dos ficheiros são alterados para indicar visualmente o estado da cópia local de trabalho; em suma, o programador vê instantaneamente o estado da sua cópia local de trabalho sem necessitar de fazer um `svn status`, o que é realmente muito bom.



## Componentes do Subversion

Uma vez instalado o Subversion, temos uma série de diferentes componentes deste software ao nosso dispôr:

- `svn`: o programa cliente de linha de comandos;
- `svnversion`: o programa para reportar o estado (em termos das revisões do software) de uma cópia de trabalho;
- `svnlook`: a ferramenta para inspecionar o estado de um repositório de Subversion;
- `svnadmin`: a ferramenta para criar, alterar ou reparar um repositório de Subversion;
- `svndumpfilter`: o programa para filtrar os dump streams do Subversion;
- `mod_dav_svn`: um módulo de plugins para o servidor HTTP da Apache, usado para colocar o repositório disponível a outros utilizadores através de uma rede;
- `svnserve`: um programa servidor, que corre como um processo (daemon) ou pode ser também invocado via SSH; é outra forma de tornar o repositório disponível a outros utilizadores através de uma rede, através do protocolo `svn`, em vez de se usar o servidor HTTP da Apache.

## Configuração do servidor de HTTP Apache

Nesta secção vamos ver como configurar o servidor HTTP da Apache para usar URLs que pertençam a repositórios SVN, usufruindo deste modo da autenticação e autorização fornecidos por este servidor. Os passos que se seguem são para Linux (distribuição Ubuntu), mas para Windows é em tudo semelhante (os comandos para criar e popular um repositório podem ser todos feitos recorrendo ao TortoiseSVN(<http://tortoisesvn.tigris.org/>)).

```
$ svnadmin create /home/projecto/ProjX
$ ls /home/projecto/ProjX
conf/  dav/  db/  format  hooks/
locks/  README.txt
```

Neste momento temos um repositório SVN para o ProjX, mas que está vazio, sem ficheiros. Para popular o repositório com ficheiros, teremos que ter outra pasta que tem todos os ficheiros que queremos adicionar ao projecto SVN:

```
$ ls -F /tmp/ProjX
trunk/  tags/  branches/
$ ls -F trunk
pasta1/  pasta2/  README.txt  CHANGELOG
ficheiro1.c  ficheiro1.h
```

Esta pasta `/tmp/ProjX` tem todo o código que vamos querer importar para o repositório. Depois disso, podemos apagar esta pasta, pois todo o seu conteúdo já estará no repositório SVN. Resta portanto, importar o conteúdo desta pasta para o nosso repositório SVN:

```
$ svn import /tmp/ProjX
file:///home/projecto/ProjX -m "import
inicial"
```

Notas: Não colocar / no fim dos caminhos (é /tmp/ProjX e não /tmp/Proj/; o mesmo para o segundo caminho indicado); O nome para o projecto, neste caso ProjX, deverá ser coincidente nos dois caminhos apresentados.

Em Windows, estes comandos para criar e popular um repositório são facilmente reproduzidos com o TortoiseSVN, carregando com o botão direito do rato para aceder às opções de criar um repositório e importar dados.

Depois de importar os dados com sucesso, vamos transferir o conteúdo de ProjX para uma cópia local de trabalho:

```
$ svn checkout
file:///home/projecto/ProjX/trunk
nome-do-projecto

A nome-do-projecto/pasta1
A nome-do-projecto/pasta2
A nome-do-projecto/README.txt
...
Checked out revision 1.
```

Nota: No checkout, o nome para a pasta local que vai conter o ProjX é o que o utilizador pretender, não tem de ser igual a ProjX.

Resta apenas configurar o servidor HTTP da Apache. Aceder ao ficheiro de configuração, `apache2.conf`, que fica em `/etc/apache2` e adicionar as entradas:

```
LoadModule dav_module
modules/mod_dav.so
LoadModule dav_svn_module
modules/mod_dav_svn.so
```

(na secção inicial de carregamento dos módulos)

```
<Location /ProjX>
  DAV svn
  SVNPath /home/projecto/ProjX
</Location>
```

(na secção das Locations; Neste caso, URLs do tipo `http://localhost/ProjX` vão ser tratados pelo Apache para carregar com o módulo `dav_svn` o conteúdo do projecto ProjX.)

Nota: Em versões anteriores à 2.2 do servidor HTTP da Apache, o ficheiro `apache2.conf` é substituído pelo ficheiro `httpd.conf`

Reiniciamos o apache:

```
sudo /etc/init.d/apache2 restart
```

E temos neste momento o servidor Apache configurado para usar o repositório SVN ProjX! Basta abrir um browser e apontar para o URL `http://localhost/ProjX`. Mas, neste exemplo, não estamos a usufruir da autenticação. Para tal, alteramos de novo o ficheiro `apache2.conf` e adicionamos:

```
<Location /ProjX>
  DAV svn
  SVNPath /home/projecto/ProjX
  AuthType Basic
  AuthName "Repositorio para o ProjX"
  AuthUserFile /etc/svn-auth-file
</Location>
```

Criamos o ficheiro `/etc/svn-auth-file` e adicionamos credenciais para os utilizadores que pretendemos que tenham acesso ao servidor SVN:

```
$ htpasswd -cm /etc/svn-auth-file
utilizador1
New password: *****
Re-type new password: *****
Adding password for user utilizador1
$ htpasswd -m /etc/svn-auth-file
utilizador2
New password: *****
Re-type new password: *****
Adding password for user utilizador2
```

Nota: A opção `-c` é usada para criar o ficheiro se ainda não existir, e a opção `-m` é para usar encriptação MD5 nas passwords, que é mais seguro.

Podemos ainda usufruir da autorização baseada em caminhos, em que dizemos ao servidor HTTP que determinado utilizador só tem acesso a determinada pasta. Para tal adicionamos ao ficheiro `apache2.conf`:

```
LoadModule authz_svn_module
modules/mod_authz_svn.so
```

(na secção de carregamento dos módulos)

```
<Location /ProjX>
  DAV svn
  SVNPath /home/projecto/ProjX

  Require valid-user

  AuthType Basic
  AuthName "Repositorio para o ProjX"
  AuthUserFile /etc/svn-auth-file
  AuthzSVNAccessFile /etc/access-file
</Location>
```

(na secção das Locations; Neste caso, URLs do tipo `http://localhost/ProjX` vão ser tratados pelo Apache para



carregar com o módulo `dav_svn` o conteúdo do projecto ProjX.)

E criamos o ficheiro `/etc/access-file` indicando que acesso por pasta cada utilizador tem:

```
[ProjX:/trunk/pasta1]
utilizador1 = rw
utilizador2 = r
```

Neste exemplo, indica-se que o utilizador1 tem permissões de escrita e de leitura à pasta1 mas o utilizador2 só pode ler os conteúdos desta pasta.

Existem bastante mais goodies e formas de configurar a autenticação e autorização do servidor Apache. Ver a documentação online aqui (<http://svnbook.red-bean.com/en/1.4/svn.serverconfig.htpd.html>).

## Clientes Gráficos/IDEs

### SVN no Eclipse (plugin Subclipse)

Nesta secção vamos ver como instalar o plugin Subclipse(<http://subclipse.tigris.org/>) para o editor Eclipse. Existem outros plugins, nomeadamente o Subversive, mas nesta secção iremos apenas detalhar a instalação do Subclipse.

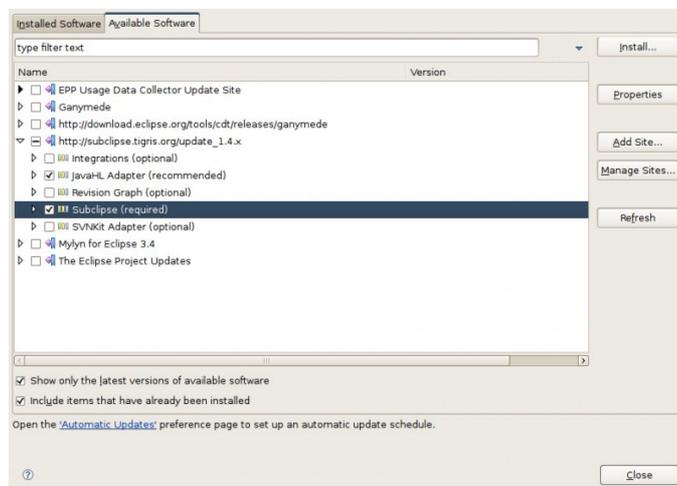
Para adicionar suporte ao SVN no editor Eclipse (versão Ganymede), basta seguir os passos:

1. Aceder ao menu Help Software Updates...
2. Aceder à tab Available Software e pressionar o botão Add site...
3. No URL escrever:

```
http://subclipse.tigris.org/update_1.4.
```

e carregar no botão OK

Uma nova entrada com o nome do URL vai surgir na lista de software disponível. Expandir a entrada e seleccionar o Subclipse e o adaptador JavaHL como se indica a seguir:

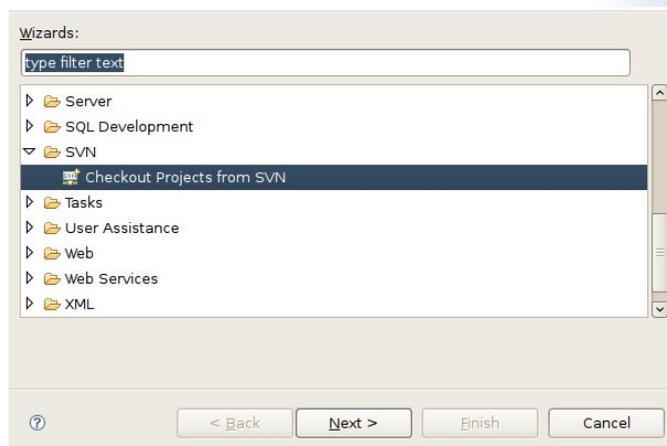


Pressionar o botão Install... para que o Eclipse calcule as dependências deste plugin e outros pacotes que sejam necessários descarregar. Na janela que surgir pressionamos o botão Next >, aceitamos os termos da licença e pressionamos no botão Finish para dar início à instalação. Por fim, surgirá uma janela para reiniciarmos o Eclipse. Escolher Yes e temos neste momento o plugin Eclipse correctamente instalado no ambiente de desenvolvimento Eclipse.

Podemos ver que temos o plugin Subclipse ao aceder ao menu File → New → Other... e vemos a entrada SVN para fazermos checkout de um projecto SVN, como se indica na figura seguinte:

#### Select a wizard

Create a new project by checking out an existing project from a SVN repository.



### SVN no trabalho colaborativo (Trac)

O Trac(<http://trac.edgewall.org/>) é um sistema de gestão de projectos e de bug-tracking que integra numa wiki o conteúdo de repositórios SVN para uma rápida visualização do que se passa nos mesmos. Para além de possibilitar aceder a todo o código de um repositório SVN (com

highlight de código), permite introduzir na wiki os objectivos e os prazos para o projecto, os bugs encontrados, e aferir o que cada utilizador tem alterado no projecto, entre outras funcionalidades muito interessantes para uma visualização intuitiva e rápida dos projectos. Vale a pena instalar este sistema, que para além de ser open source, ajuda no controlo e na análise das alterações de um projecto: possibilita ver que modificações foram feitas a dada revisão, ver diferenças entre diferentes projectos que estão num mesmo repositório, descarregar o conteúdo do projecto num arquivo .zip, para além de que é uma wiki. Antes de instalar o trac, é preciso ter instalado o Python, Genshi e uma base de dados com os drivers Python correspondentes. Depois, em Linux, basta fazer numa linha de comandos:

```
$ sudo apt-get install trac
```

Para mais informações sobre os requisitos de instalação do Trac, ver a página oficial de instalação do Trac aqui(<http://trac.edgewall.org/wiki/TracInstall>).

Em alternativa, em Windows, é preciso descarregar o arquivo .zip ou o instalador .exe do Trac. Se se optar por descarregar o instalador .exe, é apenas necessário executá-lo e seguir as indicações que surgem no ecrã. Por outro lado, se se optar por descarregar o arquivo é necessário extrairlo e seguir o passo (debaixo da pasta extraída):

```
python ./setup.py install
```

Este passo instala a ferramenta de administração trac-admin, usada para se criar e manter ambientes de projectos, bem como o servidor tracd.

Para se criar um ambiente para um projecto SVN previamente criado, em Windows ou em Linux, usa-se o comando trac-admin, como no se indica no exemplo seguinte para Linux (em Windows é semelhante mudando apenas o caminho para a pasta que vai ter o ambiente Trac):

```
$ trac-admin /home/projectos-trac/ProjX initenv
```

Nota: A pasta /home/projectos-trac/ProjX deverá estar previamente criada e vazia.

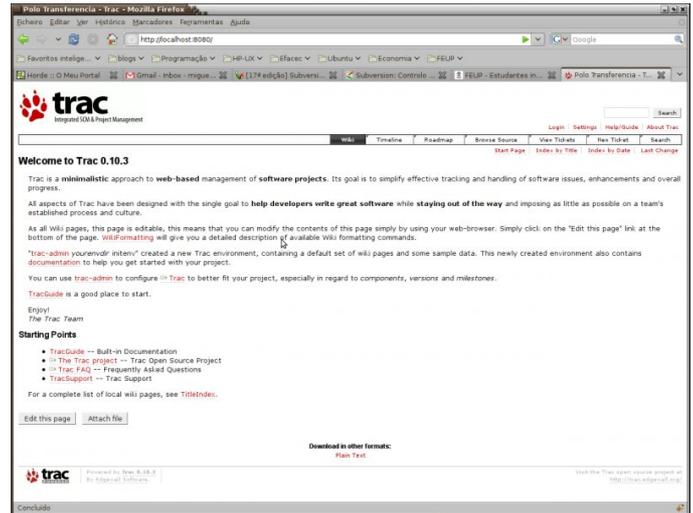
Durante este script será pedido para indicar a pasta onde se encontra o repositório SVN, o nome para o Projecto Trac, e mais informações (que podem ser deixadas com o valor por omissão, dando Enter para continuar a instalação do ambiente). Quando o script terminar, resta iniciar o servidor tracd e pondo-o a apontar para o projecto Trac já criado em /home/projectos-trac/ProjX:

```
$ tracd -port 8080 /home/projectos-trac/ProjX -d &
```

Desta forma, o servidor está à escuta na porta 8080. Quando abrirmos um browser e apontamos para o URL

<http://localhost:8080/>, obtemos a página inicial da nossa wiki recém-criada, com o código do projecto SVN disponível após se pressionar o botão Browse Source:

Resta apenas começar a editar as páginas da wiki e a usar os plugins(<http://trac.edgewall.org/wiki/PluginList>) disponíveis para usufruir de uma maior facilidade de trabalho colaborativo.



## Conclusões

O controlo de versões de software é imprescindível no desenvolvimento de software. Não só permite agilizar o mesmo como também fornece segurança aos programadores para avançar com as suas implementações sem risco de se perder código. O controlo de todo o software, desde a ideia inicial até às sucessivas iterações por que passa uma ideia, é registado pelo VCS. Se houver necessidade de voltar atrás, é rápido, simples e é garantido que se tem a versão anterior pretendida sem perda de código.

O Subversion na sua globalidade é um excelente VCS, que tem visto um aumento de popularidade entre programadores de todo o mundo. Para além de fornecer características importantes para a produtividade do software, tais como a atomicidade dos commits, a gestão das fusões de código de diferentes programadores, o versionamento de directorias e o custo das operações ser proporcional ao tamanho das alterações (e não ao tamanho dos ficheiros), é ainda software livre que pode ser usado com outras ferramentas (Apache, Trac) para se ter uma solução valiosa face a sistemas comerciais (sobre este aspecto, ver Testemunhos (<http://subversion.tigris.org/testimonials.html>)).

Para além disto, o Subversion é desenvolvido por milhares de programadores em todo o mundo, que trabalham para o aumento da qualidade deste VCS, e é suportado por uma extensa comunidade online. Uma forma rápida de ficar a

saber trabalhar com o Subversion é através dos exemplos do livro online SVN Red Book (<http://svnbook.red-bean.com/>) e das questões levantadas nas mailing lists (<http://subversion.tigris.org/mailling-lists.html>) do Subversion.

Como limitações, o Subversion tem apenas controlo de acessos dos utilizadores por directoria e não tem ainda uma maior "granularidade", no que toca ao acesso dos ficheiros.

Um exemplo disto é quando existe um projecto com pastas lib, include, bin, que deviam de ter segurança implícita (ou seja, o nome das pastas já diz implicitamente o que certos actores podem fazer, como os programadores só acederem à pasta lib e include, e os testers do projecto acederem exclusivamente à pasta bin, onde se encontram os binários a serem alvo de testes).

Outro problema actual com o Subversion é a administração de um repositório, em que não é de todo fácil apagar revisões que são lixo. Por exemplo, se uma revisão não interessar ou representar um retrocesso para outra que já existia, não é fácil apagar essa revisão do repositório. Isto é um pouco deslegante, uma vez que o repositório fica com revisões que não tem interesse de manter. Por fim, outro dos problemas prende-se com as pastas ocultas .svn, pois se o programador acidentalmente apagar esta pasta oculta, perde a forma de indicar ao servidor de SVN que fez modificações nessa pasta. Existe um procedimento para reverter esta situação, mas não é simples o suficiente e poderia estar já integrado à partida no Subversion.

Independentemente destas desvantagens, o Subversion é um VCS que representa uma mais-valia, não só para o ambiente empresarial, mas também para o dia-a-dia de um programador ou ainda aluno de uma faculdade (e por que não cada aluno ter uma conta SVN e fazer commits dos trabalhos para um repositório central SVN da faculdade?). De notar que o Subversion pode ser usado para qualquer projecto e não tem necessariamente de ser um projecto de software; o importante é a garantia que o Subversion fornece em guardar as alterações que se façam num projecto e o conjunto rápido de informação que se reúne

sobre essas mesmas alterações. A partir daí, o projecto em si pode ser de qualquer área de negócio (ver esta lista de projectos que adoptaram o Subversion (<http://subversion.tigris.org/testimonials.html#open-source-projects-using-svn>)).

Com este artigo espera-se que o utilizador tenha ficado a conhecer mais sobre a importância do controlo de versões do software e da segurança que advém deste controlo, para além de ficar a saber como instalar e usar o Subversion, bem como das ferramentas que são mais usadas actualmente com este software de controlo de versões.

## Recursos e Referências Electrónicas

- Site oficial do Subversion  
<http://subversion.tigris.org/>
- Subversion Red Book  
<http://svnbook.red-bean.com/>
- TortoiseSVN  
<http://tortoisesvn.net/>
- Trac  
<http://trac.edgewall.org/>
- Subclipse  
<http://subclipse.tigris.org/>
- Comparação entre diferentes VCS  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_revision\\_control\\_software](http://en.wikipedia.org/wiki/Comparison_of_revision_control_software)
- Comparação entre clientes de Subversion  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_Subversion\\_clients](http://en.wikipedia.org/wiki/Comparison_of_Subversion_clients)
- SVNKit biblioteca Java de Subversion  
<http://en.wikipedia.org/wiki/SVNKit>
- "The Forrester Wave: Software Change and Configuration Management, Q2 2007"  
[http://www.collab.net/forrester\\_wave\\_report/index.html](http://www.collab.net/forrester_wave_report/index.html)
- Testemunhos de empresas que usam o Subversion  
<http://subversion.tigris.org/testimonials.html#testimonials>
- Lista de projectos que adoptaram o Subversion  
<http://subversion.tigris.org/testimonials.html#open-source-projects-using-svn>

### SOBRE O AUTOR



Miguel Rentes é licenciado em Engenharia Informática e de Computação pela FEUP desde 2005 e integra actualmente a equipa de Desenvolvimento da divisão de Inovação e Desenvolvimento/Gestão de Redes da Efacec Engenharia. Actualmente está a trabalhar na tese "Novas Metodologias de Controlo de Versões de Software" do Mestrado Integrado em Engenharia Informática e Computação na FEUP. Fã do software livre, Matemática, Teoria da Computação, Protocolos de Redes e Segurança Informática. Aficionado da fotografia, livros, e cinema.

miguel.rentes@portugal-a-programar.org

*Miguel Rentes*

## Packet Capture em Java com JPCAP



PCAP (Packet Capture) consiste numa API para a captura de pacotes de rede. Em sistemas baseados em Unix o Pcap é implementado na biblioteca LibPcap, no caso dos sistemas Windows este encontra-se implementado na biblioteca WinPcap.

Estas bibliotecas permitem que software consiga capturar e filtrar pacotes que viagem pela rede, enviar pacotes e até listar todas as interfaces de rede existentes no sistema bem como obter informações sobre os mesmos, tais como o nome, IP ou MAC.

É ainda possível guardar os pacotes capturados num ficheiro e mais tarde voltar esses pacotes a partir do ficheiro onde foram guardados. Tais recursos são muito utilizados em ferramentas de monitorização e análise de rede como por exemplo packet sniffers, network monitors, network intrusion detection and traffic-generators.

Jpcap é no fundo uma biblioteca intermédia entre o Java e as bibliotecas LibPcap e WinPcap, permite ao programador criar aplicações Java utilizando as funcionalidades destas bibliotecas. Existem bibliotecas semelhantes para as mais diversas linguagens tais como Python, Ruby, .NET, Perl.

Ao longo deste artigo iremos abordar alguns dos principais recursos desta biblioteca tais como:

- Interfaces de rede disponíveis no sistema.
- Preparar a interface de rede para captura.
- Capturar pacotes da interface de rede.
- Aplicar filtros à captura de pacotes de rede.
- Gravar pacotes de rede capturados em ficheiros.
- Ler pacotes de rede gravados em ficheiros.

Para isso iremos criar pequenas aplicações, exemplos simples que demonstram como utilizar cada um dos recursos mencionados acima.

Mas antes de tudo vamos precisar de preparar o nosso ambiente, para isso precisar do Jpcap que podemos encontrar em

<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/download.htm>  
|

Aqui encontramos versões para vários sistemas operativos, podemos utilizar uma versão de instalação como é o caso do Windows ou Linux ou podemos utilizar a versão código-fonte, esta parte deixo ao critério de cada um, não existem vantagens ao nível de desenvolvimento.

Para instalar basta seguir os passos descritos na página <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/install.html>

Depois da instalação vamos por mão à obra!

### Interfaces de rede disponíveis no sistema

Nesta sessão vamos ver como é simples obter a lista de interfaces de rede que se encontram disponíveis no nosso sistema, para além disso vamos ver ainda como é fácil retirar alguma informação destas interfaces.

Para isso vamos ver um pequeno programa que demonstra como, usando o Jpcap, podemos obter todos estes dados de uma forma bastante simples e rápida.

```
import jpcap.JpcapCaptor;
import jpcap.NetworkInterface;
import jpcap.NetworkInterfaceAddress;

/**
 *
 * @author magician
 */
public class JpcapInterfaces {

    /**
     * Dado um array de bytes retorna a String
     * equivalente.
     * @param input - Array de bytes.
     * @return String com a representação
     * textual do array de bytes.
     */
    public static String hex2String(byte []
input) {
        String output = "";
        for (int i = 0; i < input.length-1;
i++) {
            output +=
Integer.toHexString(input[i] & 0xff) + ":";
        }
        output +=
Integer.toHexString(input[input.length-1] &
0xff);
    }
}
```

```

        return output;
    }

    public static void main(String args[]) {

        //Obtém a lista de interfaces de rede
        no sistema.
        NetworkInterface[] interfaces =
        JpcapCaptor.getDeviceList();

        for (NetworkInterface ni : interfaces)
        {
            System.out.println("-----
            -----");
            //Nome da interface.
            System.out.println("Nome: " +
            ni.name);
            //Descrição da interface caso
            exista.
            System.out.println("Descrição: " +
            ni.description);
            //DataLink da interface.
            System.out.println("Nome da
            DataLink : " + ni.dataLink_name);
            //Descrição do DataLink caso
            exista.
            System.out.println("Descrição da
            DataLink : " + ni.dataLink_description);
            //MAC Address da interface.
            System.out.println("MAC Address: "+
            hex2String(ni.mac_address));

            for (NetworkInterfaceAddress a :
            ni.addresses) {
                //Endereço de IP da interface.
                System.out.println("IP: " +
                a.address.getHostAddress());
                //Endereço de Broadcast da
                interface.
                System.out.println("BroadCast:
                " + a.broadcast);
                //Mascara de SubRede.
                System.out.println("SubNet: " +
                a.subnet.getHostAddress());
                //Em caso de ligações P2P o
                endereço de destino.
                System.out.println("Destinho
                P2P: " + a.destination);
            }
            System.out.println("-----
            -----\n");
        }
    }
}

```

Com o exemplo acima podemos ver como é simples obter as interfaces de rede disponíveis no sistema bem como todos os seus dados, alguns deles até bastante importantes como o Nome, IP, MAC e SubNet, dados esses que podem ser utilizados mais tarde das mais diversas formas. O resultado obtido após a execução desta pequena aplicação será algo como:

```

Destinho P2P: null
IP: fe80:0:0:0:216:36ff:feb0:5cd7
BroadCast: null
SubNet: ffff:ffff:ffff:ffff:0:0:0:0
Destinho P2P: null
-----
--
-----
--
Nome: any
Descrição: Pseudo-device that captures on all
interfaces
Nome da DataLink : LINUX_SLL
Descrição da DataLink : Linux cooked
MAC Address: 0:0:0:0:0:0
-----
--
-----
--
Nome: lo
Descrição: null
Nome da DataLink : EN10MB
Descrição da DataLink : Ethernet
MAC Address: 0:0:0:0:0:0
IP: 127.0.0.1
BroadCast: null
SubNet: 255.0.0.0
Destinho P2P: null

```

Este output corresponde a um sistema Linux, noutros sistemas podem aparecer diferenças nomeadamente nos nomes dos dispositivos. Por exemplo, pode aparecer algo como `\Device\NPF_{C3F5996D-FB82-4311-A205-25B7761897B9}` ao invés do simples etho.

Para que as nossas aplicações consigam utilizar o jpcap correctamente estas devem ser executadas com permissões de admin, caso contrário é possível que não sejam obtidos resultados, uma vez que o sistema pcap necessita dessas permissões para realizar as operações sobre as interfaces.

## Preparar a interface de rede para captura

Para preparar a captura de pacotes de rede basta utilizar o método static `openDevice` da classe `JpcapCaptor`, este método abre a interface escolhida e prepara a captura de pacotes. Abaixo podemos ver como usar este método

```

NetworkInterface[] interfaces =
JpcapCaptor.getDeviceList();

//openDevice(NetworkInterface
interface, int snaplen, boolean promics,
int to_ms)
JpcapCaptor captor =
JpcapCaptor.openDevice(interfaces[0],
65535, false, 20);

```

Como podemos ver são passados quatro argumentos ao método `openDevice`, o primeiro argumento é a interface a ser utilizada na captura, neste caso vamos utilizar o primeiro interface da lista ou seja o "eth0". O segundo argumento corresponde ao numero de máximo bytes a serem capturados de cada vez, o terceiro argumento irá dizer se a captura será em modo promiscuo (true) ou não (false). O modo promiscuo permite capturar pacotes de rede mesmo que a sua origem ou destino não seja a da interface aberto para captura. O modo não promiscuo apenas captura pacotes cujo destino ou origem se a interface escolhida. O último argumento corresponde ao timeout em milissegundos dado para a captura de pacotes.

## Capturar pacotes da interface de rede

Vamos agora passar à captura dos pacotes, o processo pode ser feito de duas formas diferentes mas que atingem o mesmo fim. Podemos utilizar a forma callback ou one-by-one.

A forma callback passa pela utilização dos métodos `processPacket` ou `loopPacket` da classe `JpcapCaptor` e pela implementação da interface `PacketReceiver`. De uma forma resumida o que vamos fazer é criar uma classe que implementa a interface `PacketReceiver` onde implementamos o método `receivePacket` que irá conter o que deverá ser feito a cada pacote capturado. Depois de implementada a interface, usamos o método `processPacket` ou `loopPacket`, ambos atingem o mesmo fim com a diferença que no `processPacket` é compatível com `timeout` e modo `non blocking`.

A forma one-by-one é a mais "primitiva" e a mais flexível dado que é possível controlar todas as ações feitas sobre o pacote. Para isso utilizamos o método `getPacket` da classe `JpcapCaptor`, este método como o nome indica retorna um pacote capturado, assim apenas temos que criar um mecanismo que irá repetir o processo quantas vezes quisermos.

Vamos para isso ver um pequeno exemplo que captura 20 pacotes e imprime na consola o IP de origem/destino bem como as portas correspondentes e o tamanho do pacote.

```
import java.io.IOException;
import jpcap.JpcapCaptor;
import jpcap.NetworkInterface;
import jpcap.packet.Packet;
import jpcap.packet.TCPPacket;
import jpcap.packet.UDPPacket;

/**
 * @author magician
 */

public class OpenInterface {
    public static void main(String args []){
```

```
        //Lista de interfaces de rede no
        sistema.
        NetworkInterface[] interfaces =
        JpcapCaptor.getDeviceList();

        try{
            //Abre a interface 0 da lista.
            JpcapCaptor captor =
            JpcapCaptor.openDevice(interfaces[0], 65535,
            false, 20);
            //Simples contador.
            int i = 0;
            Packet p = null;
            //Cliclo para capturar 20 pacotes.

            while(i < 20){
                //Captura um pacote.
                p = captor.getPacket();
                //Verifica se o pacote é do
                tipo TCPPacket
                if(p instanceof TCPPacket){
                    TCPPacket tcp = (TCPPacket)
                    P;
                    System.out.println("SRC: " +
                    tcp.src_ip.getHostAddress() + ":" +
                    tcp.src_port +
                    " \tDST: " +
                    tcp.dst_ip.getHostAddress() + ":" + tcp.dst_port
                    +
                    " \tSize = " +
                    tcp.length + " bytes");
                }
                //Verifica se o pacote é do
                tipo UDPPacket
                else if(p instanceof
                UDPPacket){
                    UDPPacket udp = (UDPPacket)
                    P;
                    System.out.println("SRC: " +
                    udp.src_ip.getHostAddress() + ":" +
                    udp.src_port +
                    " \tDST: " +
                    udp.dst_ip.getHostAddress() + ":" + udp.dst_port
                    +
                    " \tSize = " +
                    udp.length + " bytes");
                }
                i++;
            }
            //Fecha a captura de pacotes.
            captor.close();
        }
        catch(IOException io){
            System.out.println(io.getMessage());
        }
        catch(Exception e){
            System.out.println(e.getMessage());
        }
    }
}
```

Como podemos ver apenas foram processados os pacotes UDP e TCP, mas existem outros como por exemplo `ARPPacket`, `DatalinkPacket`, `ICMPPacket` entre outros. Este pequeno exemplo irá gerar um output semelhante ao que se segue a baixo sem os \* que foram colocados por motivos óbvios de privacidade e segurança.

SRC: 7*.177.*05.30:1670	DST: 19*.16*.*.1*3:49153	Size = 40 bytes
SRC: 19*.16*.*.1*3:49153	DST: 7*.177.*05.30:1670	Size = 1119 bytes
SRC: 19*.16*.*.1*3:1900	DST: 19*.16*.*.1*3:1900	Size = 247 bytes
SRC: *3.*5*.40.166:51457	DST: 19*.16*.*.1*3:49153	Size = 88 bytes
SRC: *3.30.6.103:*74*	DST: 19*.16*.*.1*3:49153	Size = 40 bytes
SRC: 19*.16*.*.1*3:49153	DST: *3.30.6.103:*74*	Size = 328 bytes
SRC: 19*.16*.*.1*3:579*5	DST: *1.111.49.15*:33655	Size = 445 bytes
SRC: *4.1*3.19*.*4:60000	DST: 19*.16*.*.1*3:49794	Size = 58 bytes
SRC: 19*.16*.*.1*3:49794	DST: *4.1*3.19*.*4:60000	Size = 160 bytes
SRC: *7.93.3.*55:16*4	DST: 19*.16*.*.1*3:49153	Size = 40 bytes
SRC: 19*.16*.*.1*3:49153	DST: *7.93.3.*55:16*4	Size = 550 bytes
SRC: 91.*1.*50.*11:*7075	DST: 19*.16*.*.1*3:49153	Size = 73 bytes
SRC: 19*.16*.*.1*3:49153	DST: 91.*1.*50.*11:*7075	Size = 93 bytes
SRC: *0*.99.194.194:49*90	DST: 19*.16*.*.1*3:50043	Size = 58 bytes
SRC: 19*.16*.*.1*3:50043	DST: *0*.99.194.194:49*90	Size = 260 bytes
SRC: *.*.*7.*00.1*:60053	DST: 19*.16*.*.1*3:469**	Size = 313 bytes
SRC: 19*.16*.*.1*3:469**	DST: *.*.*7.*00.1*:60053	Size = 58 bytes
SRC: *17.13*.7*.9*:600*0	DST: 19*.16*.*.1*3:504*7	Size = 52 bytes
SRC: 19*.16*.*.1*3:504*7	DST: *17.13*.7*.9*:600*0	Size = 1410 bytes
SRC: 19*.16*.*.1*3:504*7	DST: *17.13*.7*.9*:600*0	Size = 306 bytes

Embora neste exemplo apenas tenhamos extraído estes dados é possível extrair ainda mais dados dos pacotes, como por exemplo os dados (data) enviados no ficheiro, headers, version, etc.. Iremos ver mais à frente um exemplo em que retiramos os dados enviados.

## Aplicar filtros à captura de pacotes de rede

O Jpcap permite ainda a utilização de filtros de forma a limitar os pacotes capturados a um círculo restrito de possibilidades e não tudo o que passa pela interface como vimos até agora. Vamos começar por ver a sintaxe utilizada pelo tcpdump que é a mesma aplicada ao Jpcap.

### Filtragem por Host

Sintaxe	Descrição	Exemplo
host <host>	Captura todos os pacotes que entram e saem do host	h o s t 192.168.1.100
src host <host>	Captura todos os pacotes com origem no host	s r c h o s t 192.168.1.100
dst host <host>	Captura todos os pacotes com o host como destino	d s t h o s t 192.168.1.100

### Filtragem por Porta

Sintaxe	Descrição	Exemplo
port <port>	Captura todos os pacotes que entram ou saem da porta	port 80
src port <port>	Captura todos os pacotes com origem na porta	s r c port 80
dst port <port>	Captura todos os pacotes com a porta como destino	d s t port 80

### Filtragem por Network

Sintaxe	Descrição	Exemplo
net <net>	Captura todos os pacotes da rede <net>	net 192.168
src net <net>	Captura todos os pacotes que saem da rede <net>	s r c n e t 192.168
dst net <net>	Captura todos os pacotes que entram na rede <net>	d s t n e t 192.168

### Filtragem por Protocolo

Sintaxe	Descrição
ip	Captura todos os pacotes de IP
arp	Captura todos os pacotes ARP
rarp	Captura todos os pacotes ARP inversos
tcp	Captura todos os pacotes TCP
udp	Captura todos os pacotes UDP
icmp	Captura todos os pacotes ICPM

### Combinação de filtros

Sintaxe	Descrição	Exemplo
not	Negação	not src net 192.168
and	Concatenação	tcp and src host 192.168.1.100
or	Alternância	port 80 or port 8080

Acima temos a sintaxe principal que é possível utilizar para criar filtros, como podemos ver podem ser criados filtros por host, port, network, tipo de pacote e podemos ainda fazer combinações de filtros utilizando operações lógicas. Vamos agora ver um exemplo da utilização de filtros e iremos também ver neste exemplo como é possível ver os dados que são enviados nos pacotes.

```
import java.io.IOException;
import jpcap.JpcapCaptor;
import jpcap.NetworkInterface;
import jpcap.packet.TCPPacket;

/**
 * @author magician
 */

public class JpcapFilter {

    public static void main(String args []){

        //Lista de interfaces de rede no sistema.
        NetworkInterface[] interfaces =
        JpcapCaptor.getDeviceList();

        try{

            //Abre a interface 0 da lista.
            JpcapCaptor captor =
            JpcapCaptor.openDevice(interfaces[0], 65535,
            false, 20);

            //Captura apenas pacotes TCP com origem no host 192.168.1.100 e que tem como destino a porta 80
            captor.setFilter("tcp and src host 192.168.1.100 and dst port 80", true);

            //Simples contador.
            int i = 0;

            //Ciclo para capturar 20 pacotes.
            while(i < 20){
                //Captura um pacote e converte para TCPPacket dado que apenas a capturar pacotes TCP.
                TCPPacket p = (TCPPacket) captor.getPacket();
                //Gera o output com a informação sobre o pacote
                System.out.println("SRC: " +
```

```
p.src_ip.getHostAddress() + ":" + p.src_port +
        " \tDST: " +
        p.dst_ip.getHostAddress() + ":" + p.dst_port +
        " \tSize = " +
        p.length + " bytes");

        //Caso o pacote contenha dados este são impressos e o programa para.
        if(p.data.length > 0){
            System.out.println(new String(p.data));
            break;
        } i++;
    }

    //Fecha a captura de pacotes.
    captor.close();
} catch(IOException io){
    System.out.println(io.getMessage());
}

    catch(Exception e){
        System.out.println(e.getMessage());
    }
}
}
```



O que o programa vai fazer assumindo que o nosso IP corresponde ao 192.168.1.100 é capturar todos os pacotes TCP que são enviados pelo nosso computador e que tem como destino a porta 80. De uma forma resumida vai capturar todos os pedidos HTTP feitos por nós. Na realidade não irá capturar todos mas apenas alguns, ou seja, irá capturar até encontrar um pacote que contenha dados e irá gerar um output semelhante ao que se segue.

```
SRC: 192.168.1.100:49323      DST:
66.102.9.147:80      Size = 60 bytes
SRC: 192.168.1.100:49323      DST:
66.102.9.147:80      Size = 52 bytes
SRC: 192.168.1.100:49323      DST:
66.102.9.147:80      Size = 715 bytes
GET / HTTP/1.1
Host: www.google.pt
User-Agent: Mozilla/5.0 (X11; U; Linux
i686; en-US; rv:1.8.1.11) Gecko/20071204
```

```

Ubuntu/7.10 (gutsy) Firefox/2.0.0.11
Accept:
text/xml,application/xml,application/xhtml+xml,
text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: pt-pt,pt;q=0.8,en;q=0.5,en-us;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie:
NID=17=BNGeDayJAh13ZHF6XFJ2mwo5N3rNTRu6WaFD
UYmbgAC4pyT0kzPnSKN0qKAz_ajH2WjpdZ92DXTdjh
NyiG_s-Xwd4Mwr;
PREF=ID=8ed9295:TM=197383:LM=213911:IG=8:S=
T7z-r65IhJ

```

O exemplo acima mostra o pedido feito pelo Firefox ao servidor [www.google.pt](http://www.google.pt), como podemos ver trata-se de um pedido HTTP ao servidor.

Assim encerramos este artigo, neste momento estamos prontos a criar desde grandes aplicações a pequenos utilitários de análise de pacotes de rede.

## Links

### JPCAP Home

<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>

### JPCAP API

<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/javadoc/index.html>

### JPCAP Samples

(<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/samples.html>)

### LibPcap/TCPDump

<http://www.tcpdump.org/>

### Winpcap

<http://www.winpcap.org/>



## SOBRE O AUTOR



Fábio Correia é estudante de Engenharia Informática na Universidade de Évora. Partilhando o estudo com a moderação do fórum Portugal-a-Programar e a participação na Revista Programar, como um dos redactores mais activos, ainda tem tempo para explorar algumas das suas linguagens preferidas: Java, PHP e a recente D.

[fabio.correia@portugal-a-programar.org](mailto:fabio.correia@portugal-a-programar.org)

*Fábio Correia*

# Algoritmo de Verhoeff

## Introdução

Na edição 16 da Revista PROGRAMAR em Outubro de 2008 foi publicado o artigo "Algoritmos para o Cálculo de Dígito Verificador" que descreveu o cálculo de dígito verificador para códigos de identificação a partir dos métodos de módulo 10 e 11. Além dos algoritmos já apresentados existem outros mecanismos para cálculo e obtenção de dígitos verificadores, podendo-se destacar o algoritmo de Verhoeff, tema deste artigo.

O algoritmo ora apresentado foi desenvolvido por Jacobus (Koos) Verhoeff, matemático holandês para sua tese de doutorado em 1969 (GARCIA, et. al., 2007, p. 47). Verhoeff escreveu um algoritmo capaz de operacionalizar sequências de dígitos decimais de qualquer tamanho, evitando desta forma a ocorrência de erros de dados (HAMMING, 1986, p. 27) e de transposição de dígitos adjacentes quando efectuados por utilizadores humanos (WAGNER & PUTTER, 1989), tais como:

- Substituir um dígito por um outro dígito. Por exemplo, invés de informar os dígitos 5678 acaba-se por informar 4678, onde o dígito que deveria ser 5 foi substituído pelo dígito 4. Este tipo de erro ocorre quando se acciona no teclado uma tecla adjacente a tecla que deveria ter sido realmente accionada. Ocorre em cerca de 60% a 90% dos casos.

- Omissões ou adições de dígitos. Este tipo de erro ocorre normalmente por falta de atenção do utilizador. Ocorre em cerca de 10% a 20% dos casos.

- Trocar a ordem entre dois dígitos. Por exemplo, invés de informar os dígitos 5678 acaba-se por informar 6578, onde os dígitos 56 foram trocados pelos dígitos 65. Este tipo de erro ocorre quando se escreve muito rápido em um teclado. Ocorre em cerca de 10% a 20% dos casos.

Além das formas habituais de detecção de erros que podem ser equacionadas com o algoritmo de Verhoeff, é possível também descobrir algumas formas menos comuns de erros (WAGNER & PUTTER, 1989), tais como:

- Erros gémeos contínuos. Por exemplo, invés de informar os

dígitos 1123 acaba-se por informar 7723, onde os dígitos 11 foram trocados pelos dígitos 77. Este tipo de erro pode ocorrer por distração do utilizador na entrada da sequência de dígitos ou por ilegibilidade na escrita manual do código a ser verificado. Ocorrem em cerca de 0,5% a 1,5% dos casos.

- Erros gémeos irregulares. Por exemplo, invés de informar os dígitos 1214 acaba-se por informa a sequência 7274, onde os dígitos 121 foram trocados pelos dígitos 727. Este tipo de erro pode ocorrer pelas mesmas razões apontadas na ocorrência de erros gémeos contínuos ou por ilegibilidade na escrita manual dos dígitos numéricos. Ocorrem em menos de 1% dos casos.

- Transposição irregular. Por exemplo, invés de informar os dígitos 1234 acaba-se por informar a sequência 3214, onde os dígitos 123 foram transpostos de forma irregular pelos dígitos 321. Este tipo de erro ocorre pode ocorrer por distração do utilizador. Ocorre em cerca de 0,5% a 1% dos casos.

- Erros fonéticos. Por exemplo, invés de informar os dígitos 6-123 acaba-se por informar 3-123, onde o dígito 6 foi trocado pelo dígito 3. Este tipo de erro ocorre normalmente quando o código numérico é transmitido por voz por meio de um telefone, o interlocutor fala "seis" e o utilizador do outro lado entende "três". Ocorre em cerca de 0,5% a 1,5% dos casos.

Os percentuais de erros apontados por Paul Putter e Neal R. Wagner basearam-se num estudo de erros com 12.000 ocorrências de erros humanos quando da utilização de códigos de identificação.

O maior questionamento encontrado a respeito do uso do algoritmo de Verhoeff é sua complexidade de implantação. Pelo fato de ser baseado em tabelas numéricas, essas tabelas necessitam ser implementadas em memória e de forma correcta. Normalmente se utiliza técnicas de uso de matrizes ou faz-se o armazenamento dos valores em uma tabela de banco de dados. Neste artigo estará sendo considerado a primeira forma de implementação.

## Método Algorítmico

O método algoritmo apresentado por Verhoeff (1969) baseia-se no uso de três tabelas de valores fixos, sendo: tabela de localização do inversor, tabela do índice de inversão e tabela do dígito verificador. Há referência a essas tabelas, no Wikipédia, como: tabela de permuta, tabela de multiplicação e tabela de inversão. No entanto, essas denominações não representam seu sentido operacional, dificultando seu entendimento didáctico. Por esta razão neste artigo as referências as três tabelas obedecerá a seguinte nomenclatura:

Forma em alguns documentos	Forma documentada neste artigo
tabela de permuta	tabela de localização do inversor
tabela de multiplicação	tabela do índice de inversão
tabela de inversão	tabela do dígito verificador

A primeira tabela: tabela de localização do inversor possui os valores dos índices que são obtidos a partir do fornecimento de um código de identificação que terá o dígito verificador conferido ou mesmo gerado e que serão usados na tabela do índice de inversão.

Tabela de localização do inversor										
	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

Para que esta tabela seja usada é necessário inverter-se a disposição (da esquerda para à direita) dos valores constantes do código de identificação, que serão usados como coordenadas das colunas.

Assim que o código de identificação é invertido numera-se a partir de 0 (zero) até 7 (sete) da esquerda para a direita toda a sequência de código de identificação, repetindo-se a sequência de zero a sete por toda a extensão do código de identificação não importando o seu tamanho para que desta forma tenha-se os valores das coordenadas das linhas.

A segunda tabela: tabela do índice de inversão possui os valores de inversão de retorno para auxiliar a obtenção do dígito verificador da tabela do dígito verificador. Para que esta tabela seja utilizada é necessário ter primeiramente os valores de localização da tabela de localização do inversor.

Tabela do índice de inversão										
	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Para uso da tabela: tabela do índice de inversão é necessário obter as coordenadas de linha e coluna. A coordenada de coluna da tabela do índice de inversão é obtida a partir do índice de localização seleccionado na tabela de localização do inversor. Já a coordenada de linha é obtida por meio da definição do valor de verificação.

O primeiro valor de verificação é por definição o (zero), os demais valores são obtidos a partir da obtenção do índice de localização e do índice verificador localizados na própria tabela.

O índice verificador obtido na tabela do índice de inversão de um dos dígitos do código de identificação torna-se o próximo valor de verificação para o próximo dígito do código de identificação. O último valor obtido como índice verificador será utilizado como índice de localização do dígito verificador na tabela do dígito verificador.

A terceira tabela: tabela do dígito verificador apresenta os valores que serão utilizados como dígitos verificadores a partir da definição do índice verificador (que é o último valor obtido como índice verificador) extraído a partir da tabela do índice de inversão.

Tabela do dígito verificador										
Índice verificador	0	1	2	3	4	5	6	7	8	9
Dígito verificador	0	4	3	2	1	5	6	7	8	9

## Determinar dígito verificador

No sentido de entender o mecanismo de funcionamento do algoritmo de Verhoeff considere como código de identificação 9.234.876-7. O objectivo será estabelecer as transposições necessárias no sentido de determinar se o dígito verificador informado é correcto ou não.

Como primeira etapa de trabalho é necessário inverter a sequência do número do código de identificação excluindo-se desta inversão o dígito verificador, neste caso 7. Assim sendo, considere apenas do código de identificação fornecido o valor 9234876. Após efectuar a inversão da esquerda para à direita fica o código de identificação invertido grafado como 6784329. Em seguida acrescente o valor 0 (zero) a frente do código de identificação invertido de forma que fique assim definido 06784329.

Distribua a sequência do código de identificação invertido linearmente como sendo a informação de definição da coordenada de coluna na tabela de localização do inversor.

Linha	0	1	2	3	4	5	6	7	Sequência numérica
Coluna	0	6	7	8	4	3	2	9	Código de identificação invertido

Para a definição da coordenada de linha relacione a partir do primeiro valor do código de identificação invertido a sequência numérica de 0 (zero) até o valor máximo de tamanho do código de identificação invertido, limitando o valor máximo da sequência em 7 (sete). Chegando-se em 7 (sete) reinicia-se a contagem a partir de (zero) se assim for necessário.

Observe a definição dos valores da coordenada de coluna (código de identificação invertido) e da coordena de linha (sequência numérica de 0 até 7).

Acompanhe a extracção dos valores de localização da tabela de localização do inversor por meio da definição das cores.

A partir do momento em que se tem definido as coordenadas de linha e coluna basta obter na tabela de localização do inversor o valor de índice de localização. A sequência a seguir mostra a obtenção de cada um dos valores de localização da tabela de localização do inversor.

Valores de linha 0 e 1 com valor de coluna 0 e 6 e índice de localização obtido 0 e 3, respectivamente:

O processo continua de forma idêntica para as restantes linhas.

A partir do momento em que se possui os valores dos índices de localização basta relacionar os valores das coordenadas de linha e coluna e deixar registado os valores obtidos na tabela de localização do inversor como sendo os valores de índice de localização.

Linha	Coluna	Índice de localização
0	0	0
1	6	3
2	7	1
3	8	2
4	4	1
5	3	6
6	2	9
7	9	8

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

A próxima etapa consiste em obter o índice verificador na tabela do índice de inversão a partir da definição das coordenadas de linha (valor de verificação) e coluna (índice de localização). Para tanto, é necessário obter os valores das coordenadas de linhas a partir da tabela do índice de inversão.

O primeiro valor de verificação a ser utilizado com o primeiro índice de verificação é por definição o (zero). Assim sendo, fica definido que o valor do índice verificador é por definição inicial o (zero), uma vez que é o (zero) também o índice de localização. Se for feito o posicionamento da linha 0 (zero) com a coluna 0 (zero) na tabela do índice de inversão obter-se-á o valor 0 (zero).

Linha	0								Valor de verificação
Coluna	0	3	1	2	1	6	9	8	Índice de localização
Índice	0								Índice verificador

A partir do valor 0 (zero) do índice verificador faz-se a transposição deste valor para a próxima posição livre para o valor de verificação de forma que se possa buscar na tabela de inversão o próximo valor de verificação.

Linha	0	0							Valor de verificação
Coluna	0	3	1	2	1	6	9	8	Índice de localização
Índice	0								Índice verificador

Neste momento têm-se o valor de verificação 0 e o índice de localização 3 que deverão se posicionados na tabela do índice de inversão a fim de obter-se o valor do índice verificador que neste caso é o valor 3, como se apresenta a seguir:

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Linha	0	0							Valor de verificação
Coluna	0	3	1	2	1	6	9	8	Índice de localização
Índice	0	3							Índice verificador

A partir desta sequência pega-se o valor de índice verificador 3 e faz-se a transposição deste valor para a próxima posição livre para o valor de verificação.

Linha	0	0	3						Valor de verificação
Coluna	0	3	1	2	1	6	9	8	Índice de localização
Índice	0	3							Índice verificador

O processo continua da mesma forma, até termos a tabela completamente preenchida:

Linha	0	0	3	4	1	2	8	4	Valor de verificação
Coluna	0	3	1	2	1	6	9	8	Índice de localização
Índice	0	3	4	1	2	8	4	7	Índice verificador

As operações anteriormente efectuadas para a extracção dos índices de inversão a título de ilustração podem ser representadas da seguinte forma:

Índice verificador	Valor de verificação	Índice de localização
0	0	0
3	0	3
4	3	1
1	4	2
2	1	1
8	2	6
4	8	9
7	4	8

O último valor obtido, neste caso 7 é o valor de índice verificador que será utilizado para indicar o valor do dígito verificador. Observe a tabela do dígito verificador seguinte:

Índice verificador	0	1	2	3	4	5	6	7	8	9
Dígito verificador	0	4	3	2	1	5	6	7	8	9

Note que o índice verificador 7 indica o dígito verificador 7 que é o dígito verificador do código de identificação 9.234.876-7. Neste caso, o valor obtido de dígito verificador é igual ao valor previamente informado. A obtenção de um valor diferente indica que o código de identificação está incorrecto.

### Obtenção de dígito verificador

Por exemplo, o código de identificação 2468019753 necessita ter obtido seu dígito verificador. Assim sendo, deve este ser representado de forma inversa: 3579108642. Os valores invertidos serão usados como valores de índices de localização de valores na coluna da tabela índice de localização. Para tanto, acrescente o valor 0 (zero) a partir da primeira posição do código de identificação invertido de forma que o código de identificação fique expresso como: 03579108642. Desta forma, cada um dos valores que compõe o código de verificação invertido será utilizado como valor de índice de coluna da tabela de localização do inversor.

É necessário estabelecer também os valores de índices de localização de linhas da tabela de localização do inversor. Assim sendo, basta serializar o código de verificação invertido com valores de 0 (zero) até 7 (sete), repetindo-se esta sequência quantas vezes for necessária até atingir o tamanho total do código de identificação. Observe a seguir a definição das coordenadas (linha x coluna) que serão utilizadas:

Linha	0	1	2	3	4	5	6	7	0	1	2
Coluna	0	3	5	7	9	1	0	8	6	4	2

A identificação da coordenada Linha representa o valor serial de 0 a 7 do código de identificação invertido que se encontra definido acima da coordenada Coluna que é o valor do código de identificação invertido. A partir da definição dos valores de índices das coordenadas linha e coluna é possível descobrir os valores de localização na tabela de localização do inversor que serão usados como coordenadas de coluna na tabela do índice de inversão Assim sendo, observe os valores seguintes:

Obtenção do índice de localização		
Linha	Coluna	Índice de localização
0	0	0
1	3	6
2	5	9
3	7	5
4	9	0
5	1	2
6	0	2
7	8	5
0	6	6
1	4	2
2	2	0

Observe que os valores de índice de localização são obtidos a partir das coordenadas de linha e coluna existentes na tabela de localização do inversor. O processo de obtenção de um índice de localização deve ser efectuado um a um para todos os valores da composição do código de verificação invertido.

Obtenção do índice verificador		
Índice verificador	Valor de verificação	Índice de localização
0	0 (inicial é zero)	0
6	0 (índice verificador anterior)	6
2	6 (índice verificador anterior)	9
7	2 (índice verificador anterior)	5
7	7 (índice verificador anterior)	0
5	7 (índice verificador anterior)	2
8	5 (índice verificador anterior)	2
3	8 (índice verificador anterior)	5
9	3 (índice verificador anterior)	6
7	9 (índice verificador anterior)	2
7	7 (índice verificador anterior)	0

O último valor de índice verificador obtido na tabela é o valor 7. Este valor caracteriza-se em ser o valor de índice da tabela do dígito verificador. A partir do valor do índice verificador basta localizar o dígito verificador na tabela do dígito verificador. No caso

apresentado o dígito verificador do índice verificador 7 é o valor 7. Assim sendo, o código de identificação 2468019753 possui o dígito verificador 7, podendo este código de identificação ser representado como **2468019753-7**.

## Função Verhoeff() em LPP

A título de ilustração e no sentido de facilitar a implementação em programação de computadores do algoritmo de Verhoeff segue o projecto de um código de função escrito em LPP (Linguagem de Projecto de Programação) com a finalidade de retornar o valor de dígito verificador de um código de identificação fornecido.

O algoritmo a seguir pode ser facilmente implementado em várias linguagens de programação, desde que respeitadas as limitações e estilos de cada uma das linguagens escolhidas.

```

função Verhoeff(CÓDIGO : cadeia) : inteiro
constante {definição das tabelas de
Verhoeff}

    TLOCAL = [ 0, 1, 2, 3, 4, 5, 6, 7, 8,
9], {tabela de localização do inversor 8 x
10}
    [ 1, 5, 7, 6, 2, 8, 3, 0, 9,
4],
    [ 5, 8, 0, 3, 7, 9, 6, 1, 4,
2],
    [ 8, 9, 1, 6, 0, 4, 3, 5, 2,
7],
    [ 9, 4, 5, 3, 1, 2, 6, 8, 7,
0],
    [ 4, 2, 8, 6, 5, 7, 3, 9, 0,
1],
    [ 2, 7, 9, 3, 8, 0, 6, 4, 1,
5],
    [ 7, 0, 4, 6, 9, 1, 3, 2, 5, 8]

    TINVER = [ 0, 1, 2, 3, 4, 5, 6, 7, 8,
9], {tabela do índice de inversão 10 x 10}
    [ 1, 2, 3, 4, 0, 6, 7, 8, 9,
5],
    [ 2, 3, 4, 0, 1, 7, 8, 9, 5,
6],
    [ 3, 4, 0, 1, 2, 8, 9, 5, 6,
7],
    [ 4, 0, 1, 2, 3, 9, 5, 6, 7,
8],
    [ 5, 9, 8, 7, 6, 0, 4, 3, 2,
1],
    [ 6, 5, 9, 8, 7, 1, 0, 4, 3,
2],
    [ 7, 6, 5, 9, 8, 2, 1, 0, 4,
3],
    [ 8, 7, 6, 5, 9, 3, 2, 1, 0,
4],

```

```

[ 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

TDIVER = [ 0, 4, 3, 2, 1, 5, 6, 7, 8, 9]
{tabela do dígito verificador 1 x 10}

variável {definição das variáveis de
auxílio a obtenção do dígito verificador}

I, POSIÇÃO : inteiro
ÍNDLOC : inteiro
{índice de localização}
ÍNDVER : inteiro
{índice verificador}
CÓDINV : conjunto[0..Tamanho(CÓDIGO)] de
cadeia {código invertido}

início

{trecho de algoritmo para inversão do
código de identificação}

para I de 1 até Tamanho(CÓDIGO) passo 1
faça
  CÓDINV[I] ← CÓDIGO[Tamanho(CÓDIGO) + 1
- I]
fim_para

{trecho de algoritmo para obtenção de
dígito verificador}

ÍNDVER ← 0
para I de 0 até Tamanho(CÓDINV) - 1
passo 1 faça
  POSIÇÃO ← I + 1
  INDLOC ← TLOCAL[POSIÇÃO - 8 * (POSIÇÃO
div 8), CarParaNum(CÓDINV[I])]
  ÍNDVER ← TINVER[ÍNDVER, INDLOC]
fim_para

Verhoeff ← TDIVER[ÍNDVER]

fim

```

O algoritmo da função **Verhoeff()** proposto faz uso de alguns recursos que depende da disponibilidade destes nas bibliotecas da linguagem de programação em uso como as funções **Tamanho()** e **CarParaNum()**. A função **Tamanho()** retorna o tamanho de um *string* e a função **CarParaNum()** converte em dado numérico inteiro um dado numérico do tipo carácter fornecido.

O trecho **POSIÇÃO - 8 \* (POSIÇÃO div 8)** permite obter-se o cálculo do resto da divisão do dividendo **POSIÇÃO** pelo divisor **8**. Este recurso pode ser definido como: **mod** na linguagem Pascal (**POSICAO mod 8**) e **%** nas linguagens C, C++, JavaScript, Java, ActionScript, PHP entre outras (**POSICAO % 8**).

Observe alguns códigos de identificação e seus dígitos verificadores obtidos a partir do algoritmo de Verhoeff: 123-3, 321-1, 26041965-3, 1-5 e 1234567890-2.

## Referências Bibliográficas

GARCIA, A. C. dos S, et. al. **A Matemática dos Códigos de Barras**. Osasco: Centro Universitário FIEO, 2007. 58 p.

HAMMING, R. **Coding and Information Theory**. 2nd ed. USA: Prentice-Hall, 1986. 272 p. ISBN-13: 978-0131390720.

VERHOEFF, J. **Error Detecting Decimal Codes**. Amsterdam: The Mathematical Centre, 1969. 118 p.

WAGNER, N. R. & PUTER, P. **Error Detecting Decimal Digits**. CACM Publisher ACM, New York, v. 32, issue 1, p. 106-110. jan. 1989. ISSN: 0001-0782.

### SOBRE O AUTOR



Natural da Cidade de São Paulo, Augusto Manzano tem 23 anos de experiência em ensino e desenvolvimento de programação de software. É professor da rede federal de ensino no Brasil, no Centro Federal de Educação Tecnológica de São Paulo. É também autor, possuindo na sua carreira mais de cinquenta obras publicadas.

augusto.manzano@portugal-a-programar.org

*Augusto Manzano*

# Fundamentos de Segurança em Redes

## Parte III (IDS)

Na continuação dos dois últimos artigos sobre alguns fundamentos de segurança em redes, este irá focalizar-se em alguns componentes mais comuns de uma rede e assim como também em Sistemas de Detecção.

### Mais um pouco de arquitectura

Na arquitectura de uma rede estão presentes diversos componentes, cada qual com as suas funcionalidades e também responsabilidades. Já tocamos um pouco neste aspecto da importância da arquitectura da rede no primeiro artigo, mas nesta parte vamos cingir-nos aos componentes mais comuns que podemos encontrar numa rede.

### Gateways

Podemos começar pelo último dispositivo que

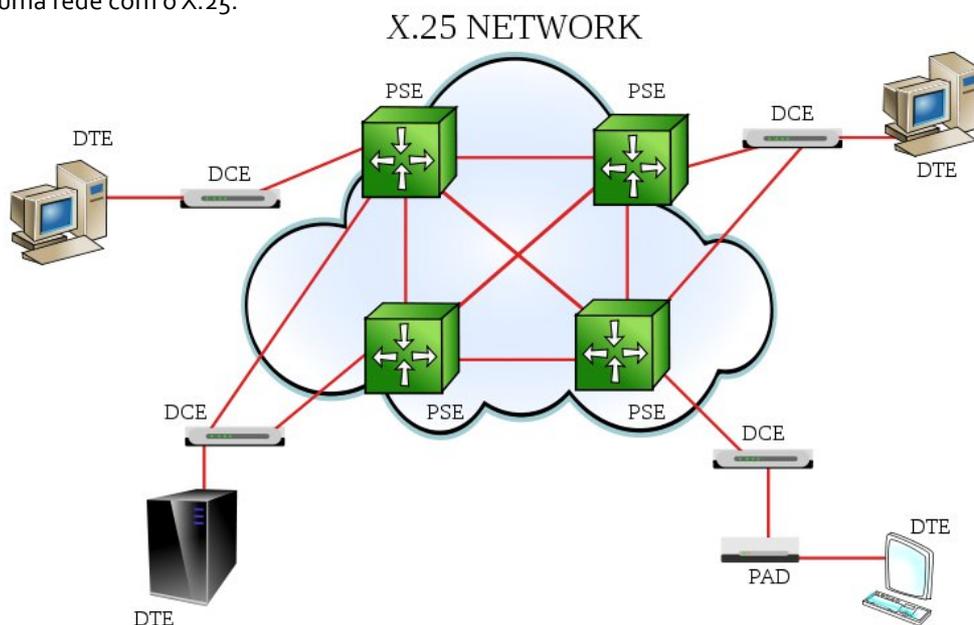
provavelmente vamos encontrar numa rede. Uma gateway é um ponto da rede que funciona como entrada para uma rede e faz de intermediário entre duas redes que falam protocolos diferentes. Geralmente as gateways trabalham nas camadas Transporte e Sessão do modelo OSI, mas dependendo do tipo de protocolo que suportem podem operar praticamente a qualquer nível ou camada do modelo OSI. Exemplos de protocolos e suites normalmente usados pelas Gateways são o X.25, o SNA ( System Network Architecture – IBM) e o ATM (Asynchronous Transfer Mode).

Dado que uma gateway é colocada na fronteira da rede é também necessário que ela desempenhe algumas funcionalidades importantes tais como Firewall, Proxy, VPN (Virtual Private Network), onde também podem ser incorporados alguns mecanismos de defesa.

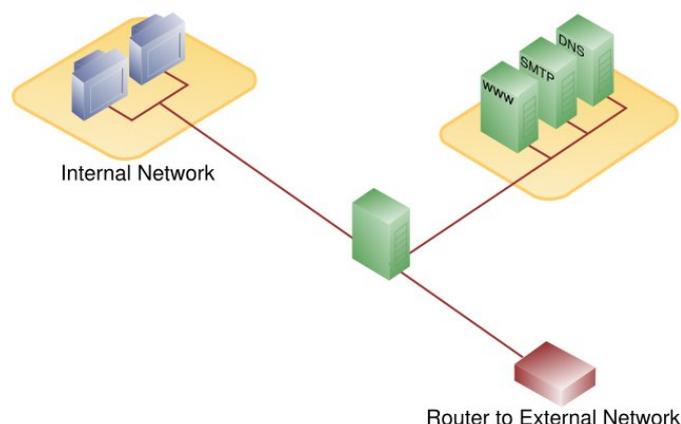
### DMZ

De seguida temos uma DMZ (DeMilitarized Zone) ou Zona Desmilitarizada. Como certamente é óbvio, este é um termo militar, que teve origem na Guerra das Coreias. Uma DMZ designa uma área geralmente entre dois países, onde actividades militares não são permitidas. Analogamente uma DMZ numa rede é uma região segura que é designada na fronteira entre a rede interna e a rede externa que na maioria dos casos é a Internet. Tipicamente usamos o termo DMZ para nos referirmos a uma pequena rede que contém serviços públicos tais como servidores de FTP, DNS e E-mail.

Exemplo de uma rede com o X.25.



O que assegura que os serviços que requerem acesso externo fiquem separados da rede local. Por estar separada da rede local, qualquer dano ou ataque que seja feito não irá afectar a rede local.



É possível também a utilização de uma DMZ e de uma Screened Subnet. O que é então uma Screened Subnet?

Uma Screened Subnet é uma firewall que possui 3 interfaces de rede. A primeira interface é a interface pública e que liga à Internet. A segunda interface conecta à DMZ e por fim a terceira interface conecta à intranet e às redes internas. Qual é o objectivo de uma screened subnet?

Bem o propósito da Screened Subnet é isolar a DMZ e os seus recursos da intranet e portanto proteger a rede interna tornando mais difícil que esta sofra as consequências de um ataque.

## Routers

Passando para os routers. Apesar de termos abordado algumas estratégias que podem ser usadas na melhoria da segurança de uma rede, não podemos deixar de falar num dos dispositivos mais comuns numa rede mas que tem um peso considerável. Os routers têm como função principal direccionar o tráfego que chega e que sai de uma rede, mas não se resumem apenas a esta função. Um router pode ser uma ferramenta bastante poderosa se for bem configurada e inserida na rede. Com um router podemos concentrar num único dispositivo diversas funcionalidades tais como IP Security, VPN, Stateful Firewall, Intrusion Detection e Quality of Service (QoS).

Existem diversas funcionalidades que podemos usar ou não num router convencional. Uma das primeiras coisas que podemos fazer é activar a DMZ e desactivar o UPnp. Porquê? Porque esta funcionalidade permite que um dispositivo de rede se auto-configure, apesar de ser aparentemente uma boa funcionalidade é também prejudicial. Por exemplo um vírus pode usar o UPnp para

abrir um buraco na firewall do router. Outro aspecto também importante é usarmos o Filtro de endereços MAC. Assim podemos restringir os dispositivos que podem se ligar à nossa rede. Por exemplo, se apenas temos 10 endereços MAC na rede, uma boa medida de segurança seria configurar o router de maneira a que apenas esses 10 endereços possam conectar à rede.

Se usamos wireless podemos modificar o nome SSID (Service Set Identifier) que vem por defeito, assim como configurar o WEP/WPA e desactivar o SSID broadcast.

## NAT – como defesa

Geralmente é usado um firewall ou um router para proteger cada segmento da rede. Estes são dispositivos ideais para proteger o perímetro da rede, mas não são os únicos a ser usados. Talvez associemos o Network Address Translation (NAT) a um método de aproveitar ao máximo a falta de endereços no IPv4, por conectar múltiplos computadores à Internet usando apenas um endereço IP. Mas o NAT tem um bom potencial no que toca a segurança, podendo ser usado para complementar o uso dos firewalls e routers. Como? NAT complementa o uso de routers ao providenciar medidas extras de segurança para uma rede interna. Usualmente, os hosts de dentro da rede são capazes de comunicar com o exterior, mas sistemas de fora necessitam passar pelo NAT para chegar à rede interna o que confunde o atacante quanto a qual é o host que quer atingir. Outra vantagem da utilização do NAT diz respeito ao bloqueamento de ataques como o SYN flood e ping of death.

## IDS (Intrusion Detection System)

Existem dois tipos principais de Intrusion Detection System, temos o NIDS (Network Intrusion Detection System) e o HIDS (Host Intrusion Detection System). Basicamente tanto num caso como noutro o objectivo é a detecção e o alerta de eventos prejudiciais ou maliciosos que possam comprometer a rede ou um computador. O alerta pode ser feito de diversas formas, desde o enviar um e-mail a apenas indicar a ocorrência num log.

Também é possível a utilização de IDS híbridos no sentido de usarmos diversos tipos de abordagem como solução para uma situação específica.

Alguns dos IDS mais conhecidos e talvez usados são:

Snort Cisco Secure IDS Dragon Sensor E-Trust IDS Audit-Guard Symantec Snort

Como podemos ver, existem diversos IDS que podem ser inseridos em diversas categorias dependendo da sua funcionalidade e arquitectura. Já referimos a existência dos

HIDS e dos NIDS, vamos falar um pouco sobre cada um deles.

## Host-based Intrusion Detection Systems

Este tipo de IDS tem como objectivo monitorizar, detectar e responder correctamente a alguma actividade suspeita ou a um ataque num computador. Um HIDS permite que um determinado computador ou computadores possam ser protegidos não só contra ataques mas também contra falhas de segurança ou vulnerabilidades. Por isso alguns HIDS trazem incorporado diversas ferramentas de gestão de políticas, análise de estatísticas e outras ferramentas importantes para um bom e eficaz funcionamento do IDS. Na maior parte dos casos o HIDS é incorporado no sistema operativo do computador em questão. No entanto, este tipo de IDS apresentam desvantagens, nomeadamente quando o computador em questão é invadido e o IDS fica comprometido e muitas vezes inutilizado para futuras acções. Essa situação leva-nos à existência em parte de outro tipo de IDS.

Exemplos de HIDS:

OSSEC Samhain Osiris Aide ThirdBrigade

## Network-based Intrusion Detection System

O funcionamento do NIDS baseia-se na análise do tráfego da rede e verificar operações suspeitas. Basicamente funcionam como Packet Sniffers que lêem o tráfego e usam regras ou outros parâmetros de maneira a verificar se a rede está ou não comprometida. Apesar de ser uma boa opção para analisar protocolos como TCP/IP, XNS e muitos outros, os NIDS não apresentam tão bons resultados face a tráfego encriptado. Outro factor constrangedor é a velocidade de análise do tráfego.

Uma boa utilização de IDS passa pela utilização conjunta de um NIDS e de um HIDS. Ficando o NIDS centralizado e analisando os dados que serão fornecidos pelos outros IDS (HIDS).

Portanto, apesar de o NIDS e o HIDS terem o mesmo objectivo a maneira como actuam é diferente, no caso do NIDS é monitorizado o tráfego da rede e no caso de surgirem situações suspeitas é dado o alarme. No HIDS essa monitorização é aplicada apenas a um host. Existem também IDS que trabalham com base na monitorização de protocolos e de aplicações, sendo estes IDS mais apropriados para redes com server's e datacenter's

## IPS (Intrusion Prevention System)

Um Intrusion Prevention System tem como função a detecção e impedir ataques contra certos recursos da rede.

A diferença de um IPS para um IDS, é que um IPS concentra-se em automaticamente defender o alvo do ataque sem a intervenção de um administrador.

A prevenção é conseguida pela incorporação de tecnologias de firewall com IDS. Dado que um IPS incorpora este género de tecnologias pode praticamente trabalhar a qualquer nível do modelo OSI. No entanto, poderá não ser uma boa solução porque um IPS pode não conseguir apresentar a mesma rapidez e robustez que um firewall e um IDS convencionais.

## IDS - Métodos Detecção e Acção

Como podemos verificar nas secções anteriores os IDS funcionam de maneiras diferentes, mas o princípio básico de operacionalidade mantém-se ou seja analisar pacotes do tráfego da rede. No entanto existem duas maneiras básicas de detecções que se baseiam na detecção de anomalias e detecção com base em assinaturas.

No caso da detecção por anomalias é feita uma análise à procura de actividades fora do comum e é feita uma comparação com logs de actividades ditas normais ou com regras definidas pelo administrador. Portanto se existir um contraste de actividade daquilo que é habitual aquele computador fazer é dado o alarme e é indicada uma anomalia que é transmitida ao administrador.

As regras por exemplo podem ser definidas de maneira a verificar se por exemplo certos ficheiros são acedidos, se certos recursos são usados, etc. São diversos os parâmetros os quais podemos usar para verificar anomalias.

Apesar de ser uma boa solução apresenta algumas desvantagens como é lógico. Uma desvantagem são os falsos alarmes que poderão ser activados mesmo quando acções legítimas são efectuadas, mas dado que diferem do padrão habitual despoletam um alarme.

Outra metodologia empregada pelo IDS é a detecção com base em regras ou assinaturas já predefinidas. Essas assinaturas representam um padrão que é característico de um determinado ataque. Portanto esses IDS possuem uma base de dados dessas assinaturas que são usadas para comparação com a actividade da rede.

Como é que os IDS respondem às detecções de intrusão?

O tipo de resposta depende do tipo de ataque que é feito e do tipo de alarme gerado e por isso existe dois tipos de resposta, modo passivo e activo que podem ser incorporadas nos IDS.

Alguns desses modos podem passar desde o bloquear o endereço de IP, bloqueando assim o endereço do atacante o



que permite em certa forma travar o SPAM e os ataques de Denial of Service. Outro modo interessante passa por analisar o ataque por algum tempo, retirar informação e usar esses dados para tornar o IDS mais eficaz.

## Conclusão

Como podemos ver ao longo destes artigos, a segurança é uma parte fundamental numa rede. Para termos relativa segurança não podemos basear a nossa confiança apenas num software ou dispositivo pensando que isso basta por si só. Uma boa solução de segurança para uma rede pode não ser o melhor para outra. É preciso fazer uma análise da rede, daquilo que queremos proteger dos objectivos que

queremos atingir e por isso é preciso encarar a segurança de uma rede como sendo um projecto que baseia a sua capacidade em diversas camadas, cada qual com o seu objectivo.

### A ver:

Cisco's Virtual LAN Communications white paper - [http://www.cisco.com/warp/public/cc/pd/wr2k/cpbn/tech/vlan\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/wr2k/cpbn/tech/vlan_wp.htm)

Top 5 Intrusion Detection Systems <http://sectools.org/jids.html>

Linux IDS [http://www.linuxsecurity.com/resources/intrusion\\_detection-1.html](http://www.linuxsecurity.com/resources/intrusion_detection-1.html)

### Referências:

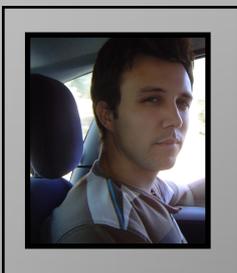
Sybex CCNA Cisco Certified Network Associate Study Guide

### Imagens:

<http://en.wikipedia.org/wiki/File:X25-network-diagram-0a.svg>

[http://en.wikipedia.org/wiki/File:DMZ\\_network\\_diagram\\_1\\_firewall.svg](http://en.wikipedia.org/wiki/File:DMZ_network_diagram_1_firewall.svg)

#### SOBRE O AUTOR



Residente em Lisboa, **Ciro Cardoso** é um grande amante do vasto mundo que é a informática em especial a segurança. Está actualmente a trabalhar na próxima etapa do percurso Cisco com a certificação CCNP. Gosta bastante da área de redes, configuração e implementação de protocolos.

[ciro.cardoso@portugal-a-programar.org](mailto:ciro.cardoso@portugal-a-programar.org)

*Ciro Cardoso*



## RoutingPT

O RoutingPT é um fórum português criado recentemente dedicado, principalmente, a redes e sistemas informáticos. Apesar da sua juventude, é já possível encontrar aqui várias informações relativamente a esta temática, nomeadamente em termos de Software, Hardware, Segurança ou, em termos mais

específicos, sobre Protocolos existentes e mesmo VoIP. A curva de evolução do fórum tem sido bastante boa, sendo que é esperado que dentro de em breve comece a tornar-se um bom repositório de informação sobre este tema.

<http://www.routingpt.com/>

## Concurso de Projectos de Programação P@P

A comunidade Portugal-a-Programar decidiu organizar o "Concurso de Projectos de Programação P@P". Este concurso pretende a dinamização e divulgação das diversas linguagens de programação, assim como reconhecer os programadores pelos seus trabalhos.

Este concurso está aberto a qualquer programador e encontra-se dividido em duas categorias "Projectos de software livre/aberto" e "Projectos de software proprietário". Irá ser realizado de dois em dois meses, alternando entre as duas categorias acima mencionadas, encontrando-se as datas dos concursos e prazos de entrega de trabalhos, disponíveis no regulamento.

Para eleger o melhor projecto irá ser efectuada uma pré-selecção pelo júri do concurso, de onde saíam 5 projectos, que serão alvo de votação pública por todos os membros da comunidade (com mais de 20 mensagens no fórum).

O vencedor terá direito a alguns prémios, entre os quais uma T-Shirt do P@P, alojamento para o projecto no servidor do P@P, entre outros de possíveis patrocinadores.

Para informação mais detalhada sobre o concurso podem consultar o regulamento, que se encontra em anexo, e/ou contactar um moderador da secção de "Projecto de Programação" por mensagem privada, através do fórum, ou através do endereço [projectos@portugal-a-programar.org](mailto:projectos@portugal-a-programar.org).

### Calendário do próximo concurso

Categoria: Código Aberto

Início do Concurso:	01-01-2009
Prazo de Entrega:	15-02-2009
Anuncio de Finalistas e Votação do Público:	22-02-2009
Fim de Votação e Anuncio do Vencedor:	01-03-2009



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](http://www.revista-programar.info)

para mais informações em como participar,  
ou então contacta-nos por

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

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

contamos com a tua ajuda!



**Equipa PROGRAMAR**

Um projecto Portugal-a-Programar.org

