Instituto Superior Politécnico Gaya

Escola Superior de Ciência e Tecnologias

Mestrado de Administração de Redes e Sistemas Informáticos

2012/2013

Monitorização de eventos numa PME

Nuno Miguel do Carmo Quaresma
nmcq@ispgaya.pt
http://www.projetomarsi.pt/nmcq/

Junho 2013

Monitorização de eventos numa PME

Nuno Miguel do Carmo Quaresma

Relatório de Estágio

Mestrado de Administração de Redes e Sistemas Informáticos

Projeto de Investigação Aplicada e de Desenvolvimento/ Estágio Profissional

Orientador externo: Sr. António Baixinha

Orientador interno: Eng.º Vasco Miranda

Junho 2013

Agradecimentos

O Homem como ser social, está inserido num conjunto de redes sociais mais amplas (família, amigos, colegas de trabalho, etc.) Inserido nesses grupos sociais, adquire a sua identidade enquanto ser humano e os meios fundamentais para a sua sobrevivência e necessitando de outras pessoas, resta-me então agradecer a todos os que foram agentes facilitadores durante o desenvolvimento deste exigente projeto e dele fizeram parte, direta ou indiretamente.

À Administração da DURIT, na pessoa do Sr. Flausino que permitiu a realização do estágio.

Aos funcionários do departamento de Informática da Durit que demonstraram disponibilidade, capacidade de interajuda, facultando apoio técnico e partilha de conhecimento.

Ao professor Vasco Nuno Capitão Miranda, meu orientador, por todo o empenho, dedicação, conhecimento e pelas sugestões que levaram à conclusão desta dissertação.

Ao professor Fernando Almeida, responsável pelo Mestrado pela sua colaboração e compreensão inestimáveis.

Ao Sr. António Baixinha, meu orientador externo, por todo o apoio, dedicação e conhecimento, fundamentais para a conclusão deste documento.

A todos os que leram este trabalho e manifestaram as suas opiniões, com especial relevo ao amigo Carlos Reis.

À minha família, com especial destaque para os meus pais, por todo o apoio ao longo dos últimos meses e pelos valores que me transmitem. À minha sogra e ao meu sogro pela disponibilidade que têm demonstrado.

À minha esposa, Sandra, por todo o amor, dedicação e motivação que me concedeu, nos momentos mais difíceis. Ao Miguel, meu filho, que suportou as ausências do pai necessárias durante este desafio.

Declaração de originalidade e respeito pelos direitos de autor

Nuno Miguel do Carmo Quaresma portador do cartão cidadão 11114189 declara que este trabalho foi por si realizado na íntegra e é original. Confirmo também que o material proveniente de fontes consultadas está devidamente assinalado e foi referenciado na sua totalidade.

_____

[Nuno Quaresma]

Vila Nova de Gaia, 19 de Junho de 2013

Índice

Índice de Ilustrações

Índice de Tabelas

Sumário executivo

O presente relatório tem a finalidade de contextualizar o desafio proposto no campo de ação da disciplina de Projeto de Investigação Aplicada e de Desenvolvimento/ Estágio Profissional: Monitorização de eventos numa PME.

Neste caso específico pretendeu-se a implementação de uma solução na empresa DURIT Metalurgia Portuguesa do Tungsténio, Lda, para monitorizar eventos na rede informática e no sistema de informação.

O desenvolvimento deste projeto visou a preparação de um documento de estudo sobre as funcionalidades, capacidades e vantagens criadas pela utilização de um software livre de licenciamento por parte de pequenas e médias empresas, já com alguma dimensão. Pretendeu-se a implementação de uma solução generalista para que o projeto desenvolvido, possa vir a ser implementado noutras pequenas e médias empresas, sendo uma base de estudo e de trabalho no futuro.

Este projeto aborda o estudo dos princípios de monitorização de redes informáticas, bem como o estudo e aplicação do Software Zabbix, uma solução distribuída sobre a licença GPL (General Public License), aplicada na monitorização constante de postos de trabalho, servidores, impressoras e demais serviços existentes na empresa.

O conteúdo deste, foi formado através de pesquisas bibliográficas, artigos, revistas, dissertações, troca de informações e testes realizados na empresa Durit, com o intuito de mostrar a administradores de rede, estudantes e demais interessados no tema, como melhor utilizar os recursos oferecidos pelo software de monitorização Zabbix, aumentando desta forma a confiança, desempenho e produtividade dos serviços existentes na infraestrutura de rede. Para além disso, o presente estudo trata de aspetos como instalação, configuração e interface gráfica do Zabbix. A utilização de um software com estas características, pretende também agilizar o processo de correção dos problemas ocorridos na infraestrutura, possibilitando que o administrador desta centre sua atenção para outras necessidades ou atividades da empresa.

Abstract

This report aims to contextualize the challenge in the field of applied research and Project development/Traineeship: event monitoring in a SME.

In this particular case was intended to implement a solution in the DURIT Metalurgia Portuguesa do Tungsténio, Lda, to monitor events in the computer network and information system.

The development of this project aimed at the preparation of a study on the features, capabilities and benefits created by the use of free licensed software for small and medium-sized companies, with considerable dimension. It was intended to implement a general solution so that the developed project could be implemented in other small and medium-sized enterprises, as a base of study and work in the future.

This project deals with the study of the principles of network monitoring, as well as the study and application of Zabbix Software, a distributed solution on LPG (General Public License), applied to the constant monitoring of workstations, servers, printers and other services that exist within the company.

The contents of this project, was based on bibliographical research, articles, magazines, dissertations, the exchange of information and tests performed within the company Durit, in order to show the network administrators, students and others interested in the topic, the best way to use the resources offered by the Zabbix monitoring software, and this way increasing the confidence, performance and productivity of the existing infrastructure services of the network. The use of a software with these characteristics, also enables you to speed up the correction process of possible infrastructure problems, enabling the administrator to focus his/her attention on other needs or activities within the company.

Résumé

Le présent rapport a pour but de contextualiser le défi propos dans le domaine d'action de la discipline di projet de la Recherche Appliquée et du Développement/Stage Professionnel: Surveillance d'évènements dans une PME.

Dans ce cas précis il est envisagé de mettre en œuvre une solution dans l' entreprise DURIT Metalurgia Portuguesa de Tungsténio, Lda pour surveiller les évènements dans le réseau informatique et le système d'information.

Le développement de ce projet a vise la préparation d'un document d'étude sur les traits, capacités et les avantages crées par l'utilisation d'un logiciel libre de licence de la part de petites et moyennes entreprises, qui ont déjà une certaine dimension. Il a été conçu la mise œuvre d'une solution généraliste pour le projet développé puisse être mis en œuvre dans d'autres petites et moyennes entreprises, qui dans le futur sera sujet a une base d´ études et de travail.

Ce projet aborde l'étude des principes de surveillance des réseaux informatiques, ainsi que l'étude et l'application du logiciel Zabbix, une solution distribuée sur la licence GPL (General Public License), appliquée sur la surveillance la constante des emplois, serveurs, imprimantes et autres services existant dans l' entreprise.

La teneur de celui-ci à travers de recherches bibliographiques, articles, magazines, dissertations, échangé d'informations, et essais réalisés dans l'entreprise DURIT, avec l'intention de montrer aux administrateurs de réseau, étudiants et autres intéressés par le sujet, comment utiliser les ressources offertes par le logiciel de surveillance Zabbix, augmentant la confiance, les résultats, la productivité des services existants dans l'infrastructure du réseau. Par ailleurs la présente étude traite des aspects tels que l'installation, la configuration et l'interface graphique du Zabbix.

L'utilisation d'un logiciel avec ses caractéristiques prétend également simplifier les processus de correction des problèmes que si sont produits dans l'infrastructure, rendant possible que l´administrateur de celui-ci centre attention sur d'autres besoins ou activités de l'entreprise.

Abreviaturas

CDP - Cisco Discovery Protocol

CLI - Command-Line Interface

CPU - Central Processing Unit

DNS - Domain Name System

FTP - File Transfer Protocol

GPL - General Public License

HTTP - Hypertext Transfer Protocol

IP - Internet Protocol

ISO - International Organization for Standardization

LAN - Local Area Network

LLDP - Link Layer Discovery Protocol

LTS - Long Term Support

MAC - Media Access Control

MIB - Management information base

MPLS - Multiprotocol Label Switching

NAT - Network Address Translation

OSI - Open Systems Interconnection

PC - Personal Computer

PME - Pequena e Média Empresa

POP - Post Office Protocol

RAM - Random-access memory

SLA - Service Level Agreement

SMS - Short Message Service

SMTP - Simple Mail Transfer Protocol

SNMP - Simple Network Management Protocol

SO - Sistema Operativo

SQL - Structured Query Language

SSH - Secure Shell

TCP - Transmission Control Protocol

TI - Tecnologias de Informação

WAN - Wide Area Network

Glossário

FTP - Protocolo que permite a transferência de ficheiros

GPL - Designação da licença para software livre

Host - Máquina ou computador ligado a uma rede

HTTP - Protocolo de Transferência de Hipertexto - é um protocolo de comunicação

Interface - Interface visual, forma do utilizador receber informação através de imagens

IPADDR - Endereço IP

Largura de Banda - Débito ocupado na transferência de dados

Linux - Sistema operativo open source, inspirado no sistema Minix

LTS - Long Term Support

Modelo OSI - Modelo que divide as redes de computadores em 7 camadas, de forma a se obter camadas de abstração

Router - Dispositivo de encaminhamento de pacotes de dados em redes de computadores

Script - Pequeno programa escrito numa linguagem de extensão que é executado no interior de outro programa

Switch - Comutador, hardware com multiportas, é um elemento de rede ativo

Ubuntu - Sistema operativo de código aberto construído a partir do núcleo do Linux

Virtualbox - Software de virtualização da Oracle

1 Introdução

Este ponto pretende dar uma visão geral sobre o trabalho realizado, justificando as escolhas efetuadas, expondo o tema, os objetivos e a estrutura do documento.

1.1 Enquadramento

As pequenas e médias empresas atualmente têm uma infraestrutura informática mais ou menos complexa, dependendo da área de atividade e das suas necessidades. Com o passar do tempo as suas redes informáticas foram crescendo e foi sendo cada vez mais exigente a administração da sua infraestrutura, começando a ser necessário existir uma pró-atividade na resolução de problemas à medida que estes iam surgindo.

Como reflexo deste aumento das infraestruturas, o número de ocorrências de problemas manteve o mesmo sentido.

Neste sentido a monitorização da infraestrutura ganhou uma importância, muitas vezes descurada, mas cada vez mais essencial nas pequenas e médias empresas portuguesas, que estão cada vez mais dependentes das novas tecnologias como suporte da sua atividade principal.

Esta foi a principal razão da escolha do tema, a gestão e monitorização de ativos numa infraestrutura de rede, optando por ferramentas de código aberto, ou de utilização livre, tornando o projeto menos dependente de questões financeiras que o pudessem vir a inviabilizar.

Só sentimos a falta das novas tecnologias de informação e comunicação quando estas nos falham. A monitorização da infraestrutura pretende minorar a possibilidade de ocorrências que possam vir a provocar a falha dos serviços, que são fundamentais no dia a dia das empresas.

1.2 Empresa

A empresa onde foi desenvolvido o estágio é a DURIT Metalurgia Portuguesa do Tungsténio, Lda, uma empresa industrial de referência já com vários prémios e

distinções obtidos, referentes à sua área de atuação. A DURIT está incluída num grupo de empresas, o grupo DURIT, onde já se pode contar uma dúzia de empresas estando já representado em diversos países como são exemplos a Alemanha, o Brasil, Espanha, Angola e Moçambique.

A infraestrutura informática da empresa engloba vários servidores, que servem como garante dos diversos serviços necessários, desde a partilha de ficheiros, bases de dados, servidor de email, etc. As diversas empresas do grupo em território nacional encontram-se interligadas existindo a possibilidade de brevemente ser implementada uma rede MPLS (Multi Protocol Label Switching), para melhorar os serviços existentes e garantir uma melhor qualidade e menor tempo de inoperabilidade da rede, esta implementação esteve prevista para ser realizada ainda durante o estágio, situação que não se veio a verificar, e sem implicação para o desenvolvimento do estágio.

A DURIT é uma empresa onde as novas tecnologias não são descuradas pela administração, havendo uma perfeita consciencialização das necessidades e vantagens inerentes à sua utilização.

1.3 Objetivos

O presente documento visa sobretudo o estudo do estado da arte no que se refere à administração de redes informáticas, e sistemas assegurando uma abordagem sobre as arquiteturas e protocolos existentes que sustentam todas as ferramentas que permitem a monitorização de infraestruturas.

Relativamente às diversas ferramentas existentes para monitorização de eventos em infraestruturas informáticas, pretende-se fazer um comparativo entre as principais ferramentas, focando especial atenção, nas ferramentas de utilização gratuita ou de código aberto.

Depois do estudo sobre as ferramentas aplicáveis à infraestrutura da DURIT, pretendeu-se efetuar a implementação de uma das soluções cumprindo as metas estabelecidas entre as entidades envolvidas no projeto, mas sempre de acordo com as questões cientificas e técnicas associadas, ao trabalho realizado.

A infraestrutura informática da empresa DURIT, é um suporte a toda a atividade da empresa, como tal pretendeu-se a implementação de uma solução para a monitorização e envio de alertas em caso de anomalias da infraestrutura da empresa de forma a garantir um menor tempo de *downtime* possível

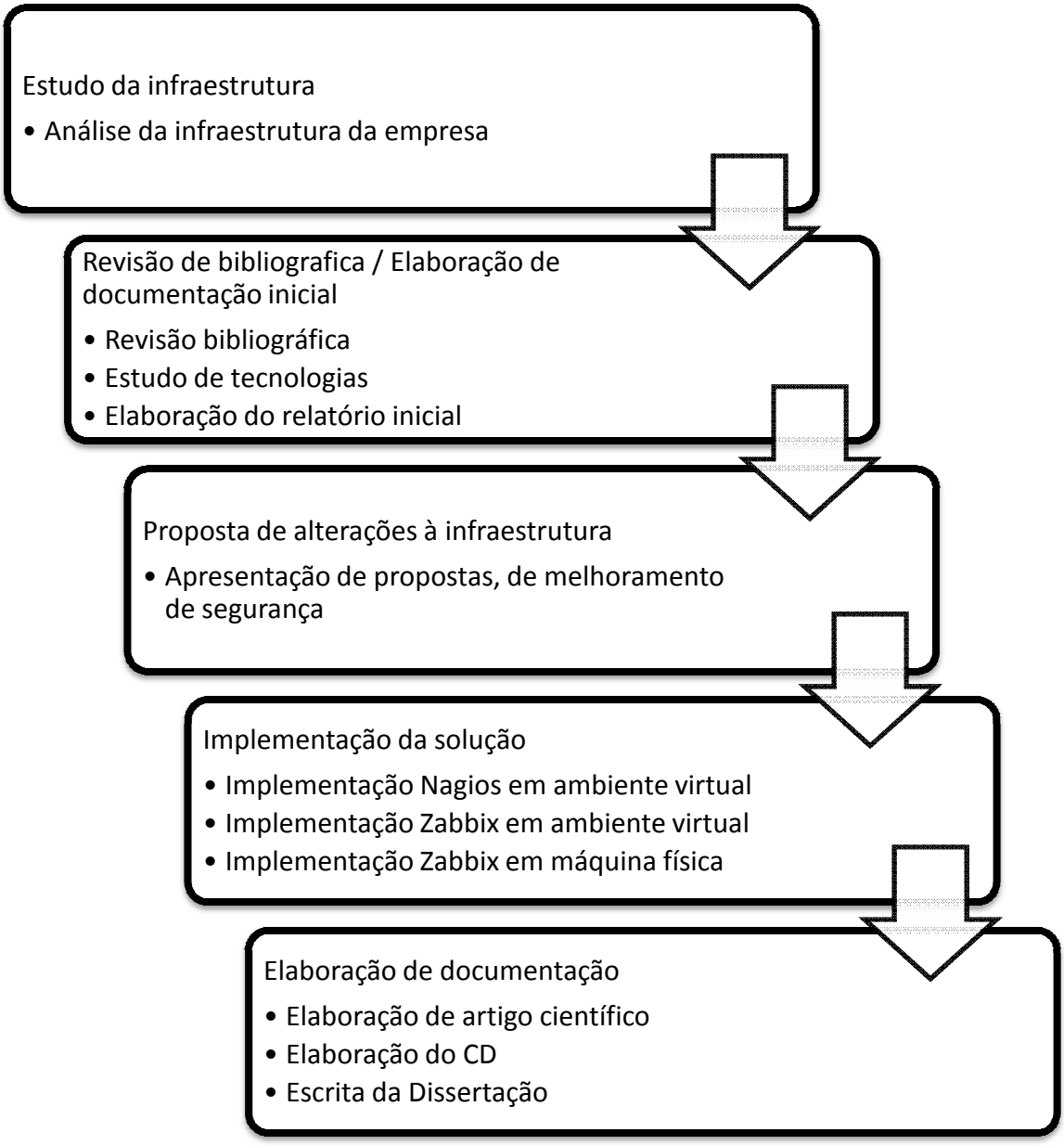
Pretendeu-se a implementação de algumas metodologias baseadas na norma ITIL, nomeadamente na gestão de incidentes, gestão de problemas e gestão de configurações.

A implementação efetuada permite ter relatórios da atividade de rede e recursos utilizados pelos diversos serviços, para desta forma ser mais fácil a prevenção de problemas e poder-se ter uma noção do crescimento das necessidades evitando uma "navegação à vista".

1.4 Metodologia de trabalho

O desenvolvimento do projeto teve várias fases, o diagrama seguinte, é elucidativo de cada uma das fases e qual a sua sequência, sendo que por vezes algumas acabaram por se sobrepor.

Ilustração 1 - Diagrama de etapas



Etapa 1: Estudo da infraestrutura.

Nesta etapa foi feito um levantamento das características da infraestrutura de rede, servidores existentes e serviços associados, ativos de rede, e políticas de segurança implementadas. Foi também nesta fase que foram identificados quais os serviços que são considerados críticos para o correto funcionamento de toda a rede informática da empresa.

Etapa 2: Revisão de bibliografia / Elaboração de documentação

Com a revisão de bibliografia pretendeu-se pesquisar sobre soluções de

monitorização existentes. Foi elaborado um relatório descritivo das soluções de monitorização existentes e das suas funcionalidades, bem como, foi também tido em consideração os custos de cada solução e o tipo de licenciamento.

Foram analisados projetos com características semelhantes e documentadas as diversas soluções código aberto existentes para monitorizar redes informáticas. Foram ainda documentados os conceitos a ter em conta para a implementação correta de uma solução de monitorização.

Nesta etapa foi feito um estudo preliminar sobre as diversas tecnologias, associadas à monitorização de redes informáticas, e elaborado o relatório inicial.

Etapa 3: Proposta de alterações à infraestrutura.

Nesta fase foram feitas propostas de alterações à infraestrutura, ficando a sua implementação condicionada à aprovação por parte dos responsáveis da empresa. Foi feito um relatório descritivo com as alterações propostas e quais as vantagens inerentes à sua aplicação.

Antes de se pensar na implementação da solução para monitorizar a infraestrutura foi fundamental responder às questões, sobre o que vamos efetivamente monitorizar, se será só hardware, serviços ou aplicações, em que períodos, e como vamos fazer essa monitorização.

Etapa 4: Implementação da solução.

Nesta fase foi feita a implementação da solução, esta implementação, foi feita numa máquina física depois de alguns testes efetuados em máquinas virtuais, e foi condicionada ao hardware que a empresa pode disponibilizar para o projeto.

Etapa 5: Recolha de informação.

Nesta fase foram tiradas conclusões sobre a implementação da solução. Pretendeu-se também analisar os relatórios fornecidos pela solução escolhida e implementada durante a realização do projeto.

1.5 Estrutura da dissertação

O presente documento encontra-se dividido em nove capítulos.

No primeiro, pretendeu-se dar uma visão geral sobre o trabalho realizado, justificando as escolhas efetuadas, expondo o tema, os objetivos e a estrutura do documento.

No segundo, foi justificada a documentação que foi considerada para a realização deste projeto, o capítulo tem por objetivo contextualizar em termos teóricos e o estado da arte da problemática da gestão e administração de redes.

No capítulo seguinte foi feito uma exposição sobre os requisitos técnicos adjacentes à elaboração do projeto.

O quarto capítulo expõe a solução escolhida, fazendo uma descrição exaustiva do trabalho efetuado.

No quinto capítulo é feito um paralelismo entre o cronograma inicialmente previsto, e o que efetivamente foi conseguido justificando os desfasamento existentes.

O sexto capítulo expõe os meios que foram necessários à elaboração do projeto, e quais tinham sido os inicialmente previstos.

O sétimo capítulo apresenta um conjunto de resultados, e é feita uma análise dos mesmos, resumindo todo o trabalho desenvolvido.

O oitavo e ultimo capítulo, descreve as conclusões gerais do projeto, bem como as evoluções que este poderá vir a ter.

2 Bases teóricas

Para a realização deste projeto baseei-me na documentação oficial das soluções de monitorização de infraestruturas informáticas existentes, não menosprezando a documentação técnica dos protocolos envolvidos nas redes de comunicação.

2.1 Redes de Comunicação

A evolução tecnológica não pára e as necessidades continuam a crescer, percebe-se que é cada vez maior a necessidade de, transporte, armazenamento e processamento de dados dentro de uma organização empresarial, o que também leva à necessidade de melhorar tempos de resposta e monitorização de toda a infraestrutura. As trocas de informação distantes no espaço, num curto período de tempo são possíveis graças às redes de comunicação.

Segundo Tanenbaum (Andrew S. Tanenbaum, Computer Networks 4ed, 2009) "Podemos definir redes de computadores, simplesmente como o conjunto de computadores autónomos interligados." É importante recordar que a expressão redes de computadores não pode ser confundida com sistemas distribuídos. Sendo a principal diferença o facto de num sistema distribuído, existirem diversos computadores autónomos o que é transparente ao utilizador, ou seja, este não tem conhecimento deles. Além disso, será responsabilidade do sistema operativo selecionar o melhor processador, localizar e transportar todos os ficheiros de entrada necessários e colocar os resultados no local apropriado.

O facto da infraestrutura do Grupo Durit, ter sofrido recentemente alterações levou também a algumas pesquisas sobre redes MPLS, em termos de projeto estas apenas fazem o transporte sendo transparente à implementação. Faucheur (2002) faz um estudo exaustivo dos protocolos associados, sendo que a nível do modelo OSI estes irão aparecer numa camada intermédia às definições tradicionais do Layer 2 (Enlace) e Layer 3 (Rede) (ver Ilustração 3).

Ilustração 2 - modelo OSI

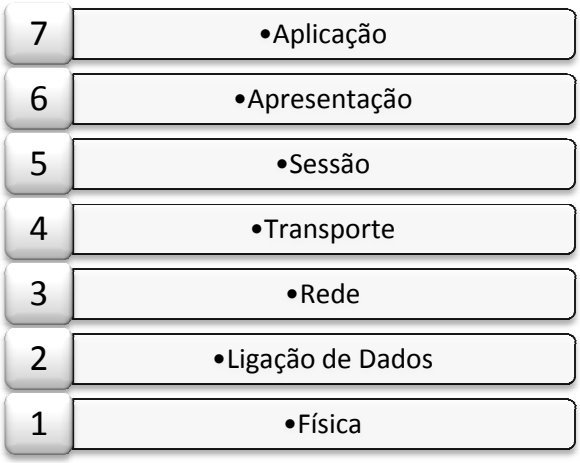
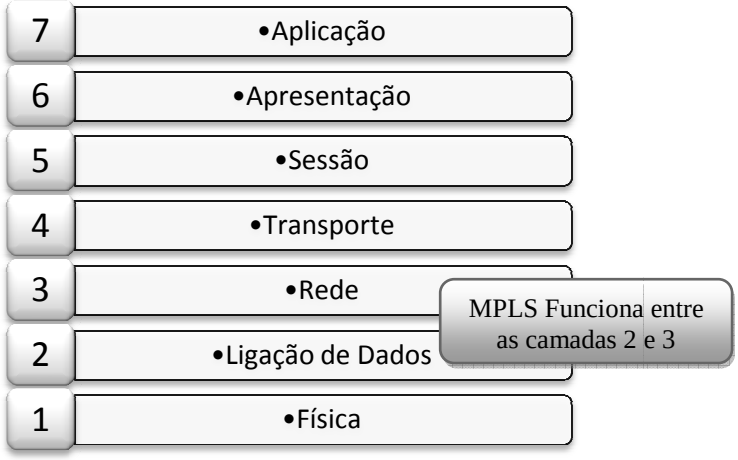
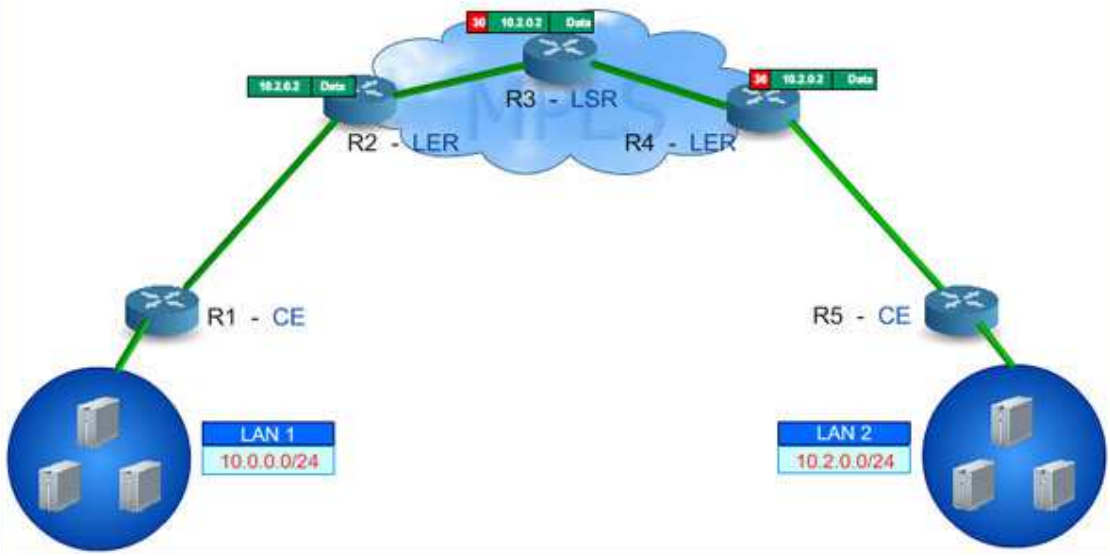


Ilustração 3 - MPLS no modelo OSI



Para o projeto a implementação desta rede seria apenas transporte, não trazendo para a solução de monitorização qualquer maleficio, pelo que não foi feito um estudo muito exaustivo desta tecnologia, no entanto esta arquitetura recentemente implementada, facilita a comunicação entre as diversas empresas, sendo muito mais simples a comunicação entre o servidor e os seus clientes (*hosts*).

Ilustração 4 - Exemplo de Arquitetura MPLS

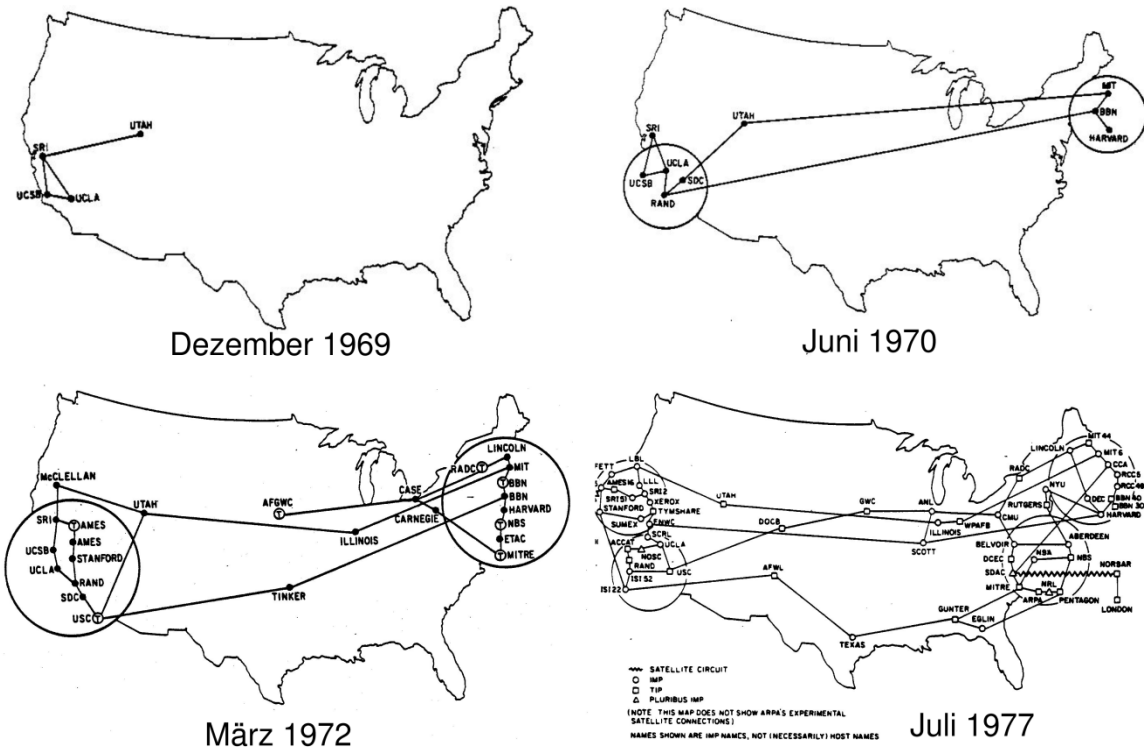


fonte: www.mplsinfo.org

Em termos de implementação a alteração feita na infraestrutura já numa fase de conclusão do projeto, não teve impacto já que ficámos perante uma rede privada virtual, o que em termos de funcionamento quer dizer que continuamos a estar numa rede local (LAN). Tanenbaum, afirma que redes locais, geralmente chamadas de LANs, são redes privadas de capital dentro de um único prédio ou campus de até alguns quilômetros de tamanho. Estas são amplamente utilizadas para conectar computadores pessoais e estações de trabalho. Como os escritórios das empresas do grupo e fábricas irão partilhar recursos (por exemplo, impressoras) e trocar informações ficaremos perante uma rede local.

Apesar de "aparentemente" estarmos na presença de uma LAN, a verdade é que a comunicação entre segmentos distantes só é possível com a utilização da Internet. Os autores Leiner, Barry; Vinton, Cerf; Clark, David; Kahn, Robert; Kleinrock, Leonard; Lynch, Daniel; Postel, Jon; Lawrence, Roberts; Wolff, Stephen, fazem uma excelente retrospetiva histórica da Internet na publicação "Communications of the ACM" de fevereiro de 1997. As redes de comunicação como hoje as conhecemos surgiram a partir de uma instituição de investigação criada nos Estados Unidos da América no final dos anos 50 denominada por ARPA (Advanced Research Project Agency).

Ilustração 5 - Arpanet de 1969 A 1977



Fonte: www.fibel.org

O objetivo era criar uma rede de comunicações robusta e fiável entre os locais mais críticos do sistema de defesa Norte-Americano. Pela primeira vez os dados foram divididos em pacotes que iriam ser encaminhados, por uma das várias vias que estivesse disponível. Foi então criada uma rede experimental, a qual viria a ser designada por ARPANET. Um dos pressupostos da rede, é que esta deveria ser uma rede descentralizada, todos os nós da rede deveriam ter um estatuto semelhante no que diz respeito às comunicações. Esta rede acabou por ser o ponto de partida para a Internet, termo que começou a ser utilizado em 1973, numa altura que na ARPA se tinha iniciado um novo conceito de investigação "internetworking", mas só no inicio da década de 80 é que foram definidos protocolos standard TCP-IP, altura em que a Internet foi definida como o conjunto de redes que utilizava este protocolo. Em 1990 surgiu o primeiro de muitos ISP (Internet Service Provider) comercial, altura em que a ARPANET deixou formalmente de existir.

2.2 Protocolos necessários

A implementação com sucesso de uma solução de monitorização, só será possível com a utilização do protocolo SNMP (Simple Network Management Protocol), existe algum hardware a título de exemplo impressoras, ou switchs, que a recolha de informação é feita utilizando este protocolo.

O SNMP é um protocolo ao nível da aplicação que facilita a troca de informações entre o servidor onde fica implementada a solução e os agentes residentes nos componentes geridos, como routers, switchs, impressoras, entre outros. Uma rede gerida por SNMP possui três elementos chave: os dispositivos geridos, os agentes e as estações de gestão. Através desta infraestrutura, o SNMP disponibiliza aos administradores da rede informações sobre a utilização dos recursos e alarmes, o que permite identificar e solucionar problemas, e planear o crescimento da rede. Os dispositivos geridos, recolhem e armazenam as informações na sua base de dados (Management Information Base - MIB) e disponibilizam-na, via SNMP, para as estações de gestão. O agente tem função específica de recolher os dados armazenados na MIB e transformá-los em informações compatíveis com SNMP.O padrão MIB que contempla a base de dados relacionado com dados referentes a de tráfego é a MIB-II. A sua organização contém um conjunto de grupos que monitorizam a execução de vários protocolos no elemento de rede, como o IP, TCP/UDP, BGP, OSPF, fornecendo o estado (ativo/inativo) das interfaces de um equipamento ou contabilizando as estatísticas dos contadores de bytes e pacotes enviados, recebidos ou perdidos para cada interface. Disponibilizam ainda, informações sobre dados relevantes relativos ao desempenho do dispositivo, como utilização de CPU e memória. Avaliação de Ferramentas de Monitorização e Gestão de Redes.

A EG Innovations, é uma empresa que foca a sua atividade na manutenção e monitorização de infraestruturas, com serviços de excelência e expõe na sua documentação oficial algumas considerações que devem ter-se em conta na implementação de soluções de monitorização. Antes de pensarmos em implementar uma solução devemos definir concretamente o que pretendemos, fazer testes de *stress*? Otimização? Monitorização? Diagnósticos? Administração?

Neste caso concreto a minha implementação irá no sentido de poder ter uma monitorização constante, vinte e quatro horas por dia, sete dias por semana, o que só se

torna viável com uma ferramenta que o possa fazer de forma automática. Tipicamente a monitorização de eventos numa infraestrutura deve ter o mínimo de intervenção humana possível, após o serviço inicial, para implementação da solução, definição de sensores e configuração de alertas, a solução deve ficar atomizada, estando apenas sujeita a pequenos ajustes.

Outro aspeto não menos importante a ter em conta, é que a solução tenha o mínimo de impacto possível na infraestrutura, isto é que não aumente de forma substancial a utilização de processador das máquinas, memória, ou que provoque constrangimento na largura de banda.

2.3 Ferramentas de monitorização

A GÉANT que é fundamental para a visão da União Europeia sobre um Espaço Europeu de Investigação (ERA European Research Area), elaborou um manual de boas práticas intitulado "Ferramentas de monitorização para Serviços de Rede e Sistemas" (Monitoring Tools for Network Services and Systems). Este manual fornece uma lista de soluções de monitorização que serão a base de estudo deste projeto. O manual indica ainda as principais características de cada uma das soluções:

2.3.1 Nagios (também Icinga):

Especialmente adequado para a monitorização de sistemas de servidores e serviços em execução neles.

O Nagios é uma solução Open Source.

Esta ferramenta é versátil, escalável e modificável, não é a solução mais fácil de implementar e manter, mas é uma solução extremamente poderosa, e existem versões comerciais, bastante mais amigáveis.

Executa verificações programadas ou recebe apenas os resultados.

Existe uma extensa coleção "plug-ins" prontos para a monitorização de vários serviços e sistemas.

Tem um interface simples entre o núcleo do Nagios e os *plug-ins* de verificação; é de fácil implementação usando os seus próprios *plug-ins* de verificação.

Existe a possibilidade de implementar outras funções de informação, mas são difíceis de utilizar.

Não há suporte direto para a criação de estatísticas e gráficos das medições recolhidas durante a verificação (por exemplo: sistema de ficheiros 19% cheio).

Existe uma base de utilizadores extensa, e pode ser contratado suporte comercial.

Icinga, é um ramo do Nagios. É compatível com o Nagios mas com uma melhor informação e GUI (Graphical User Interface).

2.3.2 Munin

Munin é uma ferramenta de monitorização de rede que ajuda a analisar as tendências dos recursos e tende a mostrar "O que aconteceu para diminuir o nosso desempenho?". O Munin é projetado para ser muito plug and play. Uma instalação padrão fornece uma série de gráficos com muito pouco trabalho.

Utilizando o Munin pode-se monitorizar o desempenho de servidores, computadores, redes, SANs, aplicações, e tudo o que vem à mente. Isso torna muito mais fácil determinar "o que é diferente hoje", de quando um problema de desempenho começou a surgir. Assim será mais fácil gerir todos os recursos, e ter uma melhor perceção das necessidades que possam vir a surgir.

Esta ferramenta é usada para a monitorização de servidores.

É uma Solução Open Source.

2.3.3 ManageEngine, OpManager

Uma ferramenta de monitorização para a rede, servidores e serviços.

É um software comercial.

2.3.4 ManageEngine, DeviceExpert

Manage Engine é instrumento de rastreamento dos dispositivos ativos.

É um software comercial

2.3.5 MetaNav

NAV - Network Administration Visualized é um conjunto de software avançado para monitorizar grandes redes de computadores que atualmente inclui 20 ferramentas.

O NAV deteta automaticamente a topologia da rede, monitoriza a carga da rede e interrupções. Pode enviar alertas sobre os eventos de rede via e-mail e SMS, permitindo a configuração flexível de perfis de alerta.

É uma ferramenta utilizada para monitorizar redes de campus.

É uma solução Open Source.

Tona-se uma ferramenta bastante trabalhosa para ter em utilização.

2.3.6 Netdisco

Netdisco é uma ferramenta Open Source baseada num interface web para gestão de redes, tendo sido lançada publicamente a primeira versão em 2003.

O público-alvo são empresas de grande/média dimensão e administradores de redes universitárias. Os dados são recolhidos para uma base de dados Postgres utilizando o protocolo SNMP e apresentados num interface web limpo que utiliza Mason que é um poderoso sistema de Templates baseado em Perl, projetado para gerar conteúdo dinâmico de todos os tipos. Os dados das configurações e de conexões dos dispositivos de rede são recebidos via SNMP e armazenados utilizando uma base de dados SQL para permitir escalabilidade e velocidade. Protocolos da camada 2 do modelo OSI (Ligação de dados) como CDP e LLDP proporcionam a descoberta automática da topologia da rede.

Esta ferramenta pode ser usada para, por exemplo, encontrar a localização de dispositivos na rede com através do endereço IP (Internet Protocol) ou MAC (Media Access Control), e informar qual a porta do switch em que se encontra ligada.

É capaz de criar um mapa da topologia da rede, por exemplo. Exibe os dados de inventário de diverso hardware.

Algumas das funcionalidade preferidas desta ferramenta:

A ferramenta permite desligar uma porta do switch, deixando um caminho de auditoria. Efetua um registo para o administrador do porquê de uma porta ter sido fechada.

Permite efetuar um inventário do hardware de rede, sistema, modelo, marca, e versão do firmware operativo.

Permite obter relatórios sobre o endereço IP e porta do switch em utilização, histórico e atual.

Netdisco recebe todos os seus dados, incluindo informações sobre a topologia, com pesquisas e consultas feitas via SNMP DNS. Não utiliza o acesso CLI e por isso não tem necessidade de palavras-chave para obter privilégios.

2.3.7 Smokeping

É uma ferramenta utilizada para análise de disponibilidade e tempos de resposta de um determinado dispositivo.

Solução Open Source.

É uma ferramenta bastante útil com boa visualização de latência na rede e capaz de enviar emails de alerta.

2.3.8 What's UP

É uma ferramenta que permite a monitorização de conexões na rede.

É um software comercial.

É um software bastante amigável, acessível e fácil de usar.

A ferramenta é capaz de criar um mapa da topologia da rede e recolher vários tipos de informações dos logs.

Estão disponíveis Add-ons para a ferramenta, mas também existe a possibilidade de ser o próprio administrador do sistema a criá-los.

2.3.9 Zabbix

Uma ferramenta de monitorização para servidores e aplicações.

É uma solução Open Source, pelo que será uma das soluções abordadas para a minha implementação.

É uma ferramenta muito rápida a detetar problemas.

Existe a possibilidade de um contrato de suporte comercial, mas a solução com suporte comercial é exatamente a mesma que está disponível para implementações gratuitas.

2.3.10 Zino

É uma solução para recolha de informações, tais como os contadores de ligação de dispositivos de rede. Apresenta a informação na forma de gráficos sendo o tempo o eixo horizontal.

É uma solução Open Source.

Possibilita a criação de mapas de topologia em camadas diferentes e gráficos de contador das ligações individuais podem ser interligados uns com os outros.

Topologia de mapas com coloração das ligações característica de acordo com a carga.

Permite o envio de mensagens de correio eletrónico com relatórios de erros de conexão de rede.

2.4 Virtualização, porquê?

Com tantas soluções existentes seria extremamente complicado a sua implementação sem se usar a "Virtualização". A virtualização é uma camada que separa o hardware do sistema operativo. Possibilita a otimização de recursos, uma vez que estarão vários Sistemas Operativos suportados pelo mesmo hardware. Cada máquina virtual terá o seu próprio hardware "apresentado" ao sistema operativo, mesmo as placas de rede terão um macaddress próprio para cada sistema instalado.

Para a realização de testes de implementações, a virtualização tem um papel fundamental, pois permite realizar diversos testes, e quando necessário repor configurações anteriores.

2.4.1 VMware Workstation

O VMware Worksation, é uma ferramenta de virtualização comercial, direcionada para ser usadas em computadores desktop. Permite a instalação de máquinas virtuais num sistema operativo utilizado em máquinas desktop como o Windows, versões GNU/Linux ou MAC OS. Possibilita a união de várias máquinas virtuais, criando uma rede virtual, que permite a interligação das mesmas. As máquinas configuradas podem utilizar interfaces de rede virtuais, que podem ser configurados como bridge (a máquina é vista como um outro computador na rede

externa do PC), NAT (a máquina liga-se ao computador que por sua vez faz a ligação para a rede externa) ou Host-only (onde a máquina virtual só comunica com o *host* ficando uma rede completamente privada). O VMware Workstation, tem a possibilidade de criar snapshots, que são registos instantâneos do estado da máquina, tornando possível a sua reposição de uma forma simples, evitando ter de configurar novamente a máquina desde o início, esta funcionalidade é extremamente útil quando ainda estamos perante ambientes de testes.

2.4.2 VirtualBox

VirtualBox é um produto de virtualização de x86 e AMD64/Intel64 poderoso para empresas e uso doméstico. O VirtualBox é extremamente rico em recursos, sendo uma ferramenta de alto desempenho para clientes corporativos. Neste momento é também a única solução profissional que está disponível gratuitamente como software de código aberto sob os termos da GNU General Public License (GPL) versão 2.

Atualmente, o VirtualBox corre em Windows, Linux, Macintosh e Solaris e suporta um número vasto de sistemas operativos convidados, incluindo mas não limitado aos Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7), DOS / Windows 3 . x, Linux (2.4 e 2.6), Solaris e OpenSolaris, OS / 2, e OpenBSD.

VirtualBox está a ser ativamente desenvolvido com lançamentos frequentes e tem uma lista sempre crescente de funcionalidades, de sistemas operativos suportados e de plataformas onde pode correr. VirtualBox é um esforço da comunidade apoiada por uma empresa dedicada: todos são incentivados a contribuir enquanto a Oracle garante que o produto sempre atende aos critérios de qualidade profissional.

2.4.3 Opção pelo VirtualBox

Por questões de licenciamento a opção escolhida caiu no VirtualBox propriedade da Oracle. Na página web do produto encontra-se diversa documentação sobre as funcionalidades, requisitos de hardware, e ligações a uma vasta comunidade onde a troca de experiências e conhecimentos são uma mais valia para todos os que querem utilizar a ferramenta.

2.5 Licenças de Software

A GNU GPL (Licença Pública Geral) é a licença que acompanha o software distribuído pelo Projeto GNU, e mais uma grande variedade, incluindo o núcleo do sistema operativo Linux. Conforme a Equipa de Recursos e Tecnologias Educativas / Plano Tecnológico da Educação do Ministério da Educação (Software livre | o que é?, 2007-2008), a formulação da GPL é tal que ao invés de limitar a distribuição do software por ela protegido, impede que este software seja integrado num software proprietário. A GPL é baseada na legislação internacional de copyright, que deve garantir cobertura legal para o software licenciado com a GPL.

Engelfriet (2002-2005), dá particular importância ao facto de que o software livre pode ser comparado com qualquer outro produto derivado de atividade intelectual e por isso deve ser protegido por leis que tratam de tal tipo de propriedade. Este conjunto de leis de proteção é conhecido como copyright. O conceito fundamental do copyright é utilizado pelo autor original do trabalho e pode determinar a maneira e a forma pela qual a sua obra poderá vir a ser utilizada, ou seja, o conceito de copyright permite ao autor determinar direitos de uso, cópia, modificação e distribuição.

Silveira (2003), destaca que a maioria das licenças usadas na publicação de software livre permite que os programas sejam modificados e redistribuídos. Estas práticas são geralmente proibidas pela legislação internacional de copyright, que tenta impedir que alterações e cópias sejam efetuadas sem a autorização do autor original da obra. As licenças que acompanham software livre fazem uso da legislação de copyright para impedir utilização não-autorizada, mas estas licenças definem clara e explicitamente as condições sob as quais cópias, modificações e redistribuições podem ser efetuadas, para garantir as liberdades de modificar e redistribuir o software licenciado desta forma. A esta versão de copyright, dá-se o nome de copyleft. Um software livre sem copyleft pode ser tornado não-livre por um utilizador, caso assim o deseje. Já um software livre protegido por uma licença que ofereça copyleft, se distribuído, deverá ser sob exatamente a mesma licença, ou seja, repassando os direitos.

2.6 Open Source

O software Open Source, também conhecido como Software Livre ou de Código Aberto, é o software fornecido com o seu código fonte visível publicamente, ou seja, segundo a definição criada pela Free Software Foundation é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído com

algumas restrições. Este modelo de software geralmente é desenvolvido por comunidades, onde indivíduos geograficamente dispersos, partilham conhecimentos, utilizando ferramentas simples para coordenar e comunicar o seu trabalho através da Internet.

Este tipo de Software, respeita as quatro liberdades definidas pela Free Software Foundation, porém, não estabelece certas restrições como as contidas na GPL. São elas:

A liberdade para executar o software, para qualquer propósito (não possui restrições por parte do fornecedor);

A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Acesso ao código-fonte é um pré-requisito;

A liberdade de redistribuir cópias;

A liberdade de aperfeiçoar o software, e publicar os seus melhoramentos, de modo a que toda a comunidade possa beneficiar.

Acesso ao código-fonte é um pré- requisito;

Nesses últimos anos, grandes empresas como a Intel, IBM, HP, Fujitsu, entre outras, têm juntando esforços e financiaram a criação do OSDL (Open Source Development Lab) – instituição cujo objetivo é a criação de tecnologias de código aberto - através de investimentos em softwares de código aberto.

O grande poder do software livre, é sem dúvidas o seu potencial de cooperação entre vários programadores e analistas, para depuração em conjunto, capaz de neutralizar pressões dos mercados e melhor subjugar complicações que possam surgir, em curtos espaços de tempo e de maneira mais produtiva e eficaz.

2.7 O modelo ITIL

Todo o projeto terá como base o modelo ITIL (Information Technology Infrastructure Library) que representa um conjunto de regras de boas práticas para a gestão de Tecnologias de Informação que foi criado no final dos anos 80 pela CCTA (United Kingdom's Central Computer and Telecommunications) para dar resposta à dependência crescente das TI sentida em qualquer organização

Ilustração 6 - Esquema do Modelo ITIL V3

fonte: www.itil-itsm-peoplecert.com

Na página de internet oficial do ITIL, http://www.itil-officialsite.com encontra-se toda a documentação necessária à implementação do modelo, bem como informação sobre os passos necessários para se poder obter uma certificação ITIL.

Na versão 3 o modelo é baseado em ciclos de vida das boas práticas de serviços, incorpora o melhor do ITIL V1 e V2 e as já testadas melhores práticas para a gestão de serviços de TI. Cinco títulos de ciclo de vida formam o núcleo das práticas do ITIL:

Estratégias de serviços

Desenho de serviços

Transição de serviços

Operação de serviços

Melhoria contínua de serviços

O núcleo é suportado por uma introdução e orientações de elementos chave, junto com orientações complementares específicas de múltiplos tópicos, e um modelo integrado do ciclo de vida do serviço, incluindo mapas do serviço, mapas organizacionais, e mapas de processos e tecnologias.

O "ciclo de vida" é para um qualquer serviço de TI, o momento desde a sua conceção até à sua retirada. A isto chama-se ciclo de vida de gestão do serviço. Há vários indivíduos e várias partes da organização envolvidos no ciclo de vida de um serviço, desde a planificação, desenho, construção, teste, versões, operação,

melhoramento, etc. Diferentes níveis da organização e pessoas com papéis diferentes realizam a tomada de decisão, o desenvolvimento e a entrega dos serviços.

2.8 Análise de projetos similares

Miguel Costa elaborou em Junho de 2011 o relatório do seu Estágio Informático em Contexto Empresarial do curso Engenharia Informática do Instituto Superior politécnico de Gaya, onde entre outras soluções implementou uma solução para a Monitorização de Sistemas Informáticos, o NTOP que é uma ferramenta para monitorizar e gerir de forma centralizada redes de computadores, tendo através de gráficos, tabelas e outros meios informação detalhada da rede a ser monitorizada. Esta aplicação funciona em ambientes Unix e Windows, monitoriza e cria relatórios sobre o tráfego e suporte dos clientes pertencentes à rede em questão utilizando os protocolos: TCP/UDP/ICMP; (R)ARP; IPX; DLC; Decnet; AppleTalk; Netbios; TCP/UDP.

Neste projeto foi ainda implementado o Cacti que é uma ferramenta especializada na área de monitorização de performance para monitorizar o desempenho de dispositivos remotos. Esta ferramenta faz a recolha, o armazenamento em base de dados, e a apresentação gráfica de dados próprios da performance de cada dispositivo num interface.

Thiago Fachini a 27 de novembro de 2010 publicou um artigo que descreve a implementação da ferramenta de monitorização Zabbix, numa empresa de desenvolvimento de software e prestadora de serviços de TI. Neste trabalho foi dado particular enfase às normas ITIL, de forma a garantir a qualidade do trabalho desenvolvido.

Segundo o autor, neste projeto foi efetuada a implementação da ferramenta Zabbix por ser uma ferramenta de monitorização com capacidade para um vasto número de parâmetros, esta é idealizada para monitorizar e controlar o funcionamento dos ativos e dos seus serviços. O Zabbix possui um mecanismo flexível de alarmes que permite aos utilizadores configurar emails, mensagens instantâneas e SMS para receber os alertas se algum evento ocorrer com os sensores monitorizados, e sendo corretamente configurada pode executar comandos remotos permitindo uma fácil resolução do problema encontrado nos ativos. Zabbix oferece excelentes características de relatórios e visualização dos dados armazenados, isso faz da ferramenta uma das melhores para a

monitorização de infraestruturas TI. Todos os relatórios e estatísticas do Zabbix, bem como as configurações dos parâmetros são acedidos através de um interface gráfico numa página web. Depois de bem configurado o Zabbix pode ter um papel relevante na monitorização de toda a infraestrutura de TI, isto pode ser aplicável tanto para pequenas empresas com uma infraestrutura pequena, com poucos servidores, quanto para grandes empresas com muitos servidores e uma infraestrutura de grandes dimensões.

3 Requisitos

Neste capítulo são apresentados os requisitos inerentes ao projeto.

3.1 Requisitos funcionais

Pretendeu-se a implementação uma solução que permita aos administradores de rede e sistemas da empresa conhecerem de forma clara o que se passa na infraestrutura que gerem. A solução tendeu melhorar as capacidades para se poder reagir de uma forma mais rápida e adequada a um eventual problema.

Por outro lado, à medida que a infraestrutura vai crescendo, e aumenta o número de sistemas ligados, aumenta também a probabilidade de eventos problemáticos, sendo que se pretende que estes sejam cada vez menos com uma monitorização constante com um sistema que envie alertas para os administradores, via email.

Pretende-se uma ferramenta adequada de monitorização, não só para permitir reações adequadas aos problemas que possam surgir, mas também que possa ser utilizada como instrumento de políticas de carácter preventivo.

A seguinte tabela tem um resumo dos principais ativos, quais os serviços que suportam, a sua criticidade, as métricas de monitorização, o critério para despoletar ações, e quais as ações a executar.

Tabela 1 - Ativos a monitorizar

Ativos	Serviços	Criticidade	Métrica de monitorização	Critério para ações	Ações
Aramis	Servidor de Email	Média	Consumo geral de processador, memória física e virtual. Largura de banda da atividade de rede. Espaço de armazenamento.	Inatividade do Servidor	Comunicar por email.
				Inatividade dos serviços	Comunicar por email.
				Espaço disponível inferior a 20%	Comunicar por email.
				Utilização de memória e processador elevados por um períodos	Comunicar por email.

Ativos	Serviços	Criticidade	Métrica de monitorização	Critério para ações	Ações
				superior a 20 minutos.	
Athos	Webserver	Média	Consumo geral de processador, memória física e virtual. Largura de banda da atividade de rede. Espaço de armazenamento.	Inatividade do Servidor	Comunicar por email.
	Servidor DNS secundário	Média		Espaço disponível inferior a 20%	Comunicar por email.
				Inatividade dos serviços	Comunicar por email.
Copernico	Base de Dados SQL	Alta	Consumo geral de processador, memória física e virtual. Largura de banda da atividade de rede. Espaço de armazenamento.	Inatividade do Servidor	Comunicar por email.
	Desenvolvimento SAP	Alta		Espaço disponível inferior a 20%	Comunicar por email.
Kepler	Controlador de Domínio	Alta	Consumo geral de processador, memória física e virtual. Largura de banda da atividade de rede. Espaço de armazenamento.	Inatividade do Servidor	Comunicar por email.
	Servidor de Antivírus	Alta		Espaço disponível inferior a 20%	Comunicar por email.
	Servidor de Impressão	Alta			
	Servidor DHCP	Alta		Inatividade dos serviços	Comunicar por email.
	Servidor DNS	Alta		Utilização de memória e processador elevados por um períodos superior a 20 minutos.	Comunicar por email.
Porthos	Servidor de Backups	Baixa	Consumo geral de processador, memória física e virtual.	Inatividade do Servidor	Comunicar por email.

Ativos	Serviços	Criticidade	Métrica de monitorização	Critério para ações	Ações
			Espaço de armazenamento.		
Frodo	Web Proxy	Média	Consumo geral de processador, memória física e virtual. Largura de banda da atividade de rede. Espaço de armazenamento.	Inatividade do Servidor.	Comunicar por email.
	WSUS - Windows Server Update Services	Média		Espaço disponível inferior a 20%.	Comunicar por email.
				Utilização de memória e processador elevados por um períodos superior a 20 minutos.	Comunicar por email.

3.2 Casos de uso

Atores dos casos de uso:

a) Administrador do sistema: tem por função administrar o sistema. A sua função inclui todas as tarefas de administração desde a criação e manutenção do sistema, atualizações, configuração de alertas a adicionar ou remover *host*s e sensores,

b) Utilizador do sistema: todos os utilizadores que façam uso da solução, apenas no intuito de obter informação sobre o estado da infraestrutura.

Ilustração 7 - Caso de Uso- Administrador

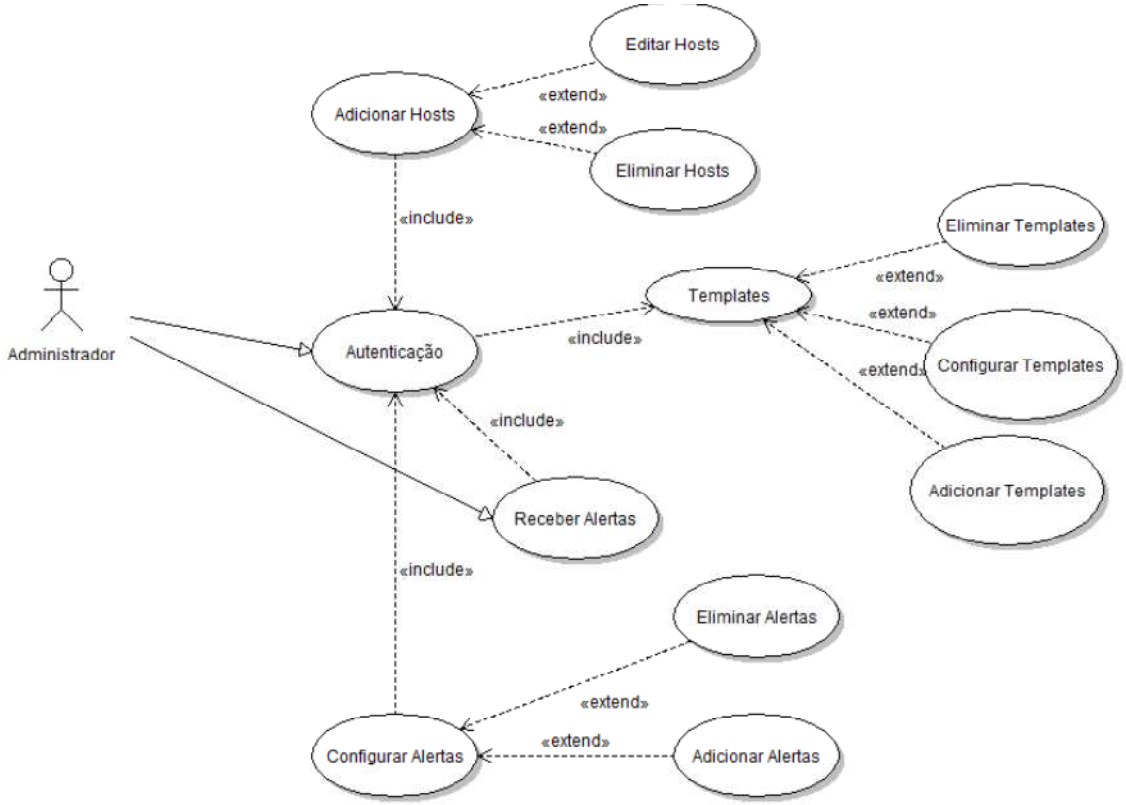
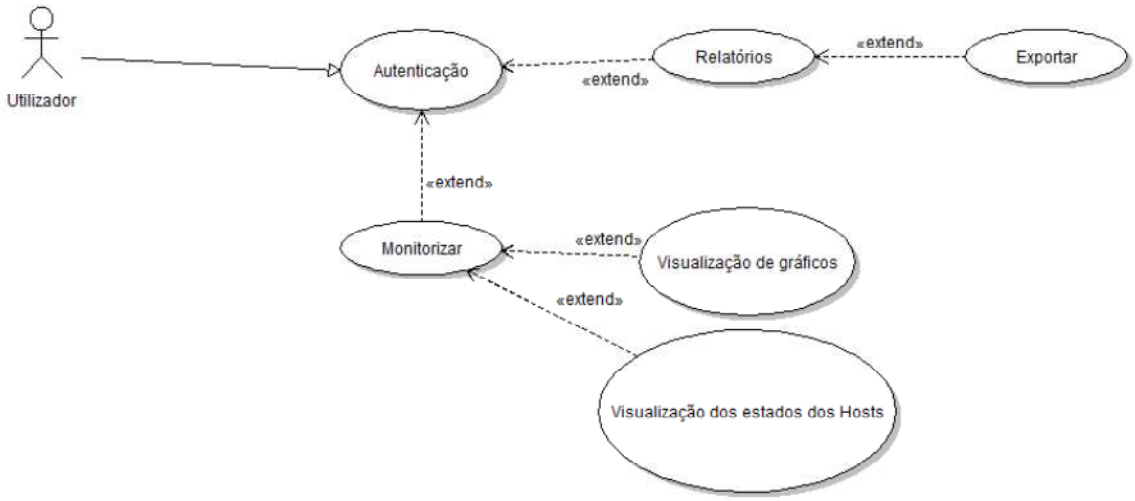


Ilustração 8 - Caso de Uso- Utilizador



3.3 Tabelas de Casos de Uso

Tabela 2 - Caso de Uso Autenticação

Nome do Caso de Uso	Autenticação
Descrição	Este caso de uso consiste na inserção das credencias (Username e Pasword) pelo utilizador, que permite ser reconhecido no sistema e obter as permissões do mesmo, consoante o grupo a que pertença
Ator	Utilizador / Administrador
Pressuposto	O servidor encontrar-se ativo com os serviços "apache" e "mysql" em funcionamento e ligado a rede local.
Pré-condição	Os dados de "username" e password têm de ser válidos.
Pós Condição	Dependendo dos dados introduzidos, as permissões de utilização poderão ser diferentes

Tabela 3 - Caso de Uso - Monitorizar

Nome do Caso de Uso	Monitorizar
Descrição	Monitorizar o estado do sistema, sem ter a possibilidade de fazer qualquer alteração às suas configurações. Permite a visualização do estado dos *hosts* no momento ou os seu gráficos para uma melhor perceção da sua atividade.
Ator	Utilizador
Pressuposto	Ter permissões e ter feito a sua autenticação no sistema.

Tabela 4 - Caso de Uso - Adicionar *Hosts*

Nome do Caso de Uso	Adicionar *hosts*
Descrição	Adicionar novos sensores para monitorização pelo sistema.
Ator	Administrador
Pressuposto	Ter permissões e ter feito a sua autenticação no sistema.

Tabela 5 - Caso de Uso - Configurar Alertas

Nome do Caso de Uso	Configurar alertas
Descrição	Configurar quais o alertas que o sistema deve mandar e quando, tendo em atenção os sensores que se pretende, a periocidade e
Ator	Administrador
Pressuposto	Ter permissões e ter feito a sua autenticação no sistema

Tabela 6 - Caso de Uso - Receber Alertas

Nome do Caso de Uso	Receber Alertas
Descrição	Receber alertas enviados pelo sistema, por e-mail ou sms.
Ator	Utilizador
Pressuposto	Pressupõe-se que foram configurados alertas a enviar para este utilizador.

3.4 Requisitos não funcionais

Relativamente a requisitos não funcionais, pretende-se uma implementação que seja:

- ✓ De usabilidade simples com um interface atrativo, que não se faça depender de conhecimentos específicos pelo utilizador;
- ✓ Pretende-se que a implementação não tenha implicações na infraestrutura, sendo o mais transparente possível aos demais serviços e utilizadores.
- ✓ De elevada disponibilidade, a solução deverá estar sempre disponível com fácil acesso pelos utilizadores;
- ✓ Deverá ser uma implementação feita sem custos de software associados;
- ✓ Segura de forma a assegurar que não seja permitido o acesso ao sistema por utilizadores não autorizados;
- ✓ Acessível remotamente, de forma a rapidamente poderem ser detetados problemas na infraestrutura;
- ✓ Acessível através um interface web, independente do browser utilizado.

- ✓ Baseada em Sistemas Open Source utilizando o Sistema Operativo: Linux Ubuntu Server LTS;
- ✓ Assente em bases de dados MySQL;
- ✓ Baseada no servidor web Apache, uma vez que a grande maioria das implementações Open Source, utilizam a linguagem de programação PHP.
- ✓ Documentada corretamente para facilitar posteriores alterações à implementação.

4 Desenvolvimento do projeto

Nesta secção serão descritos todos os pressupostos de implementação de acordo com os objetivos.

Face à definição de objetivos, a escolha iria recair sobre uma ferramenta de utilização livre, sendo que foram feitas implementação inicialmente em máquinas virtuais, e posteriormente foi feita a implementação final numa máquina física.

4.1 Implementações iniciais

Inicialmente foram feitos testes em máquina virtuais, com duas das principais soluções existentes, o Nagios na sua versão Core, uma vez que é a sua versão opensource, e o Zabbix, que logo à partida tem a vantagem de ser totalmente gratuito, uma vez que em termos comerciais apenas iremos encontrar formação e suporte.

4.2 Nagios Core

4.2.1 O que é o Nagios Core?

Ilustração 9 - Logo Nagios Core

Fonte: www.nagios.org

Nagios é uma popular ferramenta de monitoração de redes informáticas de código aberto distribuída sob a licença GPL. O Nagios permite monitorizar tanto *hosts* como serviços, enviando alertas quando ocorrerem problemas e também quando os problemas forem resolvidos, desde que sejam configurados estes alertas.

A versão utilizada no projeto foi a versão "Core 3.X", a denominação "Core" que surgiu em 2009, com o advento da versão comercial, o Nagios XI.

O Nagios Core coordena e dirige todas as tarefas associadas à monitoração, e isso pode ser entendido como administrar a periodicidade das recolhas de informação,

limitar o período máximo de execução de uma tarefa, selecionar os utilizadores que serão notificados por uma determinada ocorrência, criar relatórios com base nas informações recolhidas ou criar mapas topológicos.

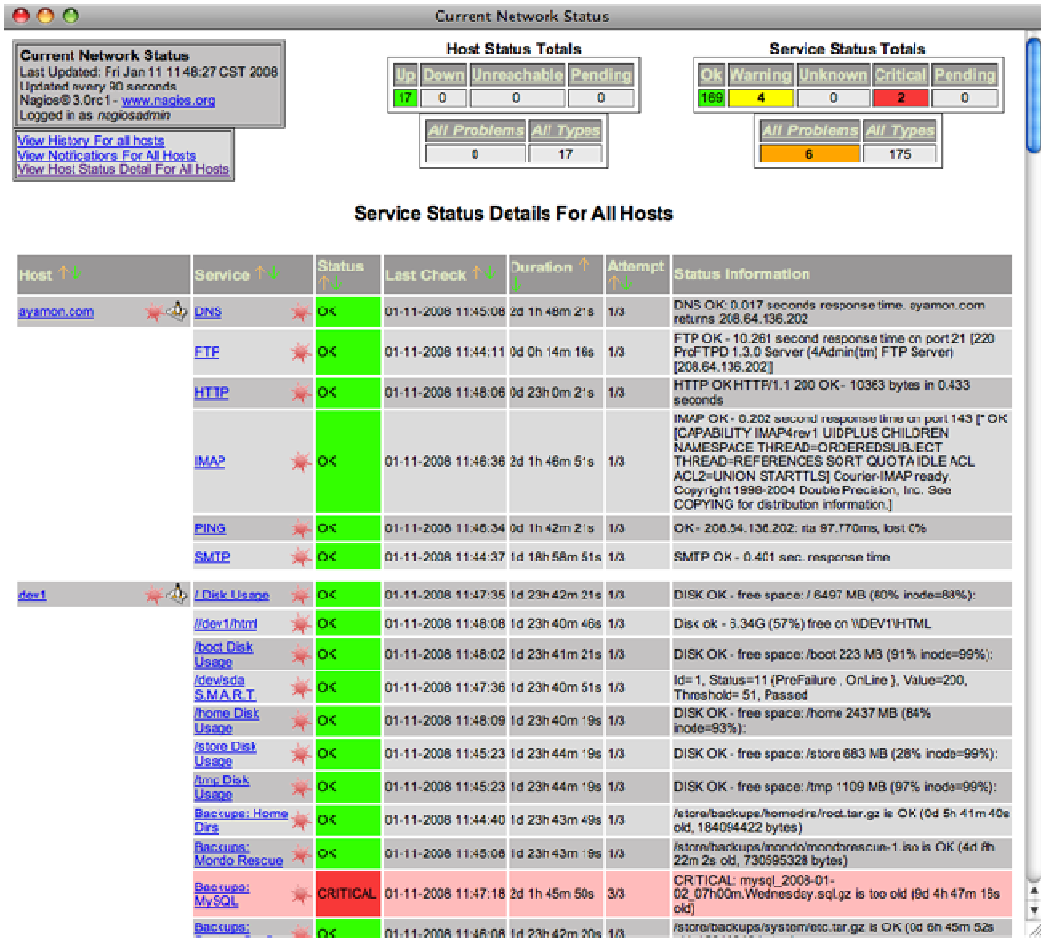
Ilustração 10 - Interface Nagios

Fonte: http://www.nagios.com

O Nagios é uma ferramenta extremamente poderosa que não possui limites com relação ao número de *hosts* e serviços que pode monitorizar, esse número é definido pelo hardware. Quanto mais poderosa for a máquina onde é instalada a ferramenta, mais tarefas poderá monitorizar com excelente precisão.

A quantidade de informação que podemos retirar, pode ser extremamente detalhada, sendo que deve ser o administrador do sistema a definir, qual o nível de detalhe que pretende, e para que será usada a informação.

Ilustração 11 - Interface Nagios - Detalhe de Serviços



Fonte: http://www.nagios.com

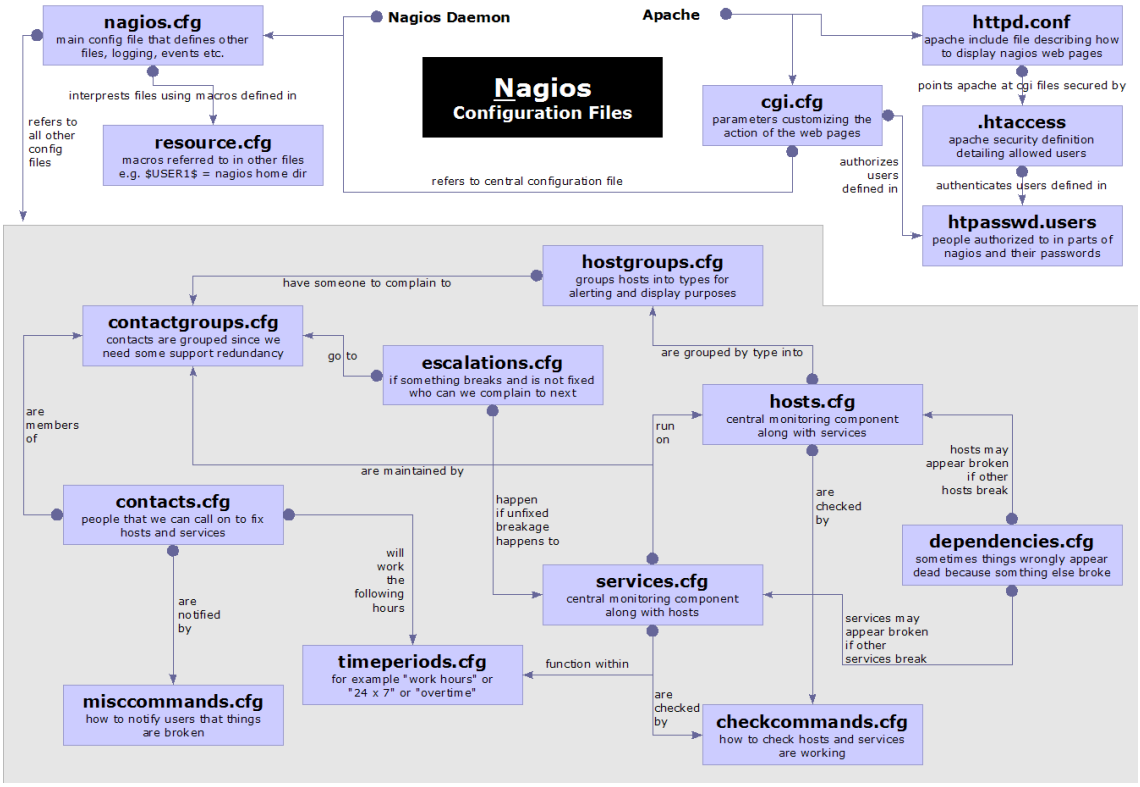
Neste projeto pretendi implementar uma solução que fosse fácil de gerir no futuro, não só gerir os *hosts* que existem atualmente na infraestrutura da Durit, mas também todos aqueles que possam vir a surgir.

A configuração de novos *hosts* no Nagios, é um processo algo complexo, sendo que foi esta a principal razão de abandonar esta solução, e optar pelo Zabbix, a outra ferramenta de monitorização que escolhi.

Para configurarmos um novo *host* no Nagios, teremos de criar um ficheiro ".cfg", que pode ser nomedohost.cfg, na diretoria "/usr/local/nagios/etc/objetos/" do servidor. Nesta diretoria ficam todos os ficheiros de configuração dos *hosts* que monitorizamos com o nosso Nagios. Posteriormente teremos sempre de editar o ficheiro "nagios.cfg", que podemos encontrar em " /usr/local/nagios/etc/nagios.cfg", e adicionar a linha " cfg _file=/usr/local/nagios/etc/objetos/nomedohost.cfg", este processo fará com que o Nagios reconheça o novo *host*, mas para isso ainda teremos de

reiniciar o serviço que pode ser feito digitando na consola: " sudo /etc/init.d/nagios restart"

Ilustração 12 - Estrutura de ficheiros do Nagios



Fonte: www.the-tech-tutorial.com

Todo este processo necessita que o administrador do sistema configure corretamente o ficheiro "nomedohost.cfg", e para isso necessita de perceber bem o seu conteúdo, ver Anexo I.

É necessário configurar o nome do *host*, o seu endereço, e muito importante, qual a informação que o Nagios vai recolher através do "check_command", toda esta configuração, acaba por ser algo penosa, principalmente, para um administrador de sistemas, onde praticamente toda a infraestrutura é baseada em sistemas Windows.

4.3 Zabbix

4.3.1 Zabbix, o que é?

Ilustração 13 - Logo Zabbix

Fonte: www.zabbix.com

Zabbix é uma ferramenta, dirigida à monitorização de redes informáticas, gratuita e opensource. A monitorização considera diversos aspetos, desde o desempenho/largura de banda de links de rede, até à disponibilidade/conectividade de equipamentos (*hosts*, *switchs*, *routers*, etc.) e serviços.

O Zabbix consegue recolher informações dos alvos monitorizados por intermédio de scripts, via agente ou através do protocolo SNMP. Para além destas funcionalidades, permite que as informações sejam guardadas numa base de dados que pode ser MySQL, PostgreSQL, SQLite ou Oracle. Na implementação final optou-se por utilizar o MySQL.

4.3.2 Proveniência do Zabbix

Um dos principais responsáveis pelo desenvolvimento desta ferramenta é Alexei Vladishev, criador e principal programador do Zabbix. O projeto Zabbix foi iniciado em 2001, na Letónia, usando a linguagem PHP, com ligação a uma base dados e utilizando um interface web.

A ferramenta Zabbix é considerada como uma das melhores de monitorização de redes, da atualidade. As suas funcionalidades, em grande parte, foram baseadas no Nagios, o que tornou o Zabbix uma das ferramentas mais completas e poderosas disponíveis.

4.3.3 Estrutura do Zabbix

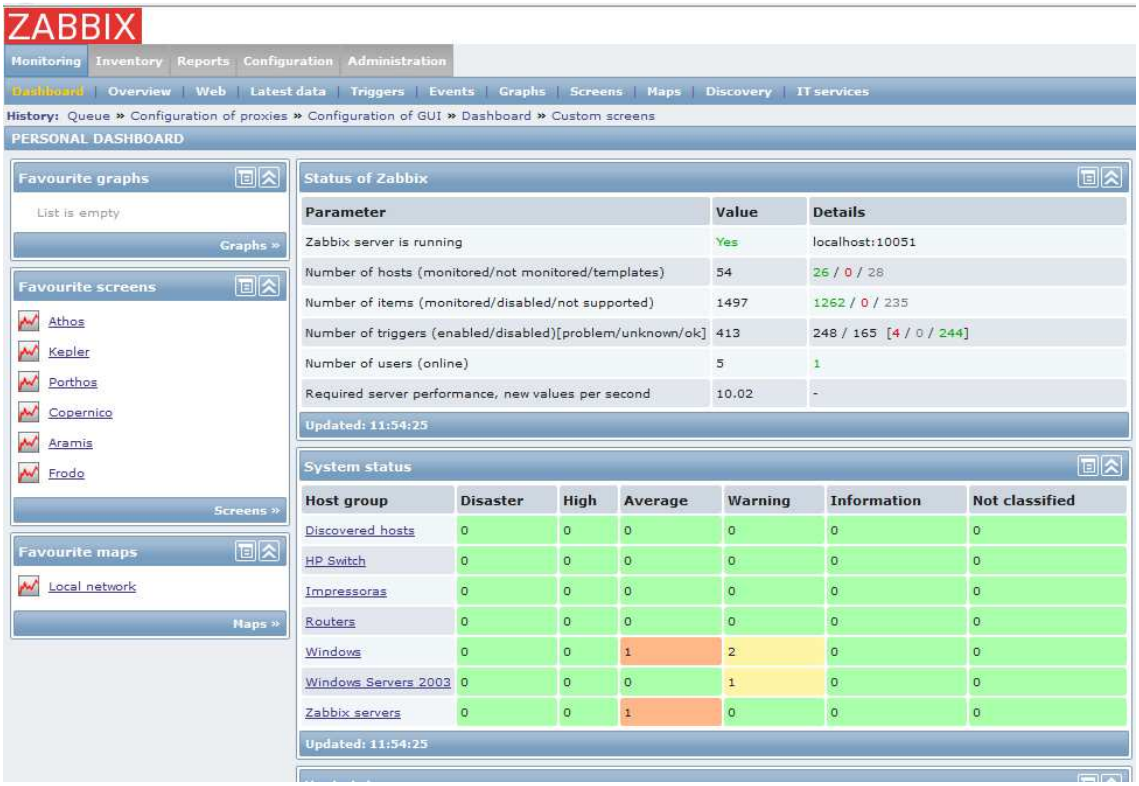
O sistema está dividido em três grandes componentes:

Servidor: responsável pela recolha e armazenamento dos dados que são monitorizados; a máquina onde é instalado o servidor Zabbix, deverá ter um sistema operativo Unix/Linux;

Agente Zabbix: encarregado por conduzir ao servidor todas as informações que foram recolhidas no sistema onde o agente está a ser executado. Normalmente, o agente executado de forma permanente, como serviço. Desta forma, quando o servidor colocar um pedido ao agente, este efetua o pedido e retorna os dados solicitados, como por exemplo: consumo de memória e do disco rígido, estatísticas do processador, etc. Nem todos os sistemas monitorizados, necessitam de ter o agente instalado, alguns *hosts*, são monitorizados utilizando o protocolo SNMP.

Interface web: permite que o administrador do sistema possa interagir e administra-lo de uma forma mais fácil, simplificando o acesso a dados e configurações, o interface foi projetado para que o sistema possa ser acedido via Web.

Ilustração 14 - Vista principal do Zabbix



4.3.4 Funcionalidades do Zabbix

A lista seguinte enumera as principais funcionalidades do Zabbix:

• Gestão centralizada;

• Acesso centralizado às informações;

• Número ilimitado de proxies (servidor intermediário que atende requisições passando os dados do cliente à frente);

• Monitorização em tempo real;

• Monitorização de alertas para disponibilidade, integridade, entre outros;

• Alertas enviados via e-mail, SMS, mensagem instantânea e via script configurado;

• Log de auditoria;

• Visualização via separadores web e mapas;

• Execução de comandos de forma remota;

• Suporte a serviços de IT hierárquico (capacidade de dispor hierarquicamente a rede, definir *hosts* pai e filhos dentro da rede e distinguir clientes inativos);

• Relatórios em tempo real de SLA's (ex. relatórios sobre o desempenho do processador);

• Facilidade de integração com sistemas de terceiros;

• Modelos pré-configurados de *hosts*;

• Facilidade de partilha de modelos;

• Sistema de pesquisa de dispositivos a serem monitorizados;

• Monitorização de páginas web;

• Suporte a qualquer plataforma;

• Suporte ao protocolo SNMP;

• Agente próprio com elevada performance e desempenho;

• Aprendizagem rápida;

• Multiutilizador com níveis de acessos configuráveis.

4.3.5 Pré-Requisitos

O servidor do Zabbix, naturalmente tem alguns pré-requisitos de software e hardware, a levar em atenção para que a sua instalação seja bem-sucedida e para que o seu funcionamento não padeça de problemas de desempenho.

Na Tabela 7, são listadas todas as plataformas que são suportadas pelo Zabbix, e na Tabela 8, são mostrados os requisitos mínimos de Software para a instalação do servidor Zabbix.

Tabela 7 - Plataformas suportadas pelo Zabbix

Plataforma	Servidor ZABBIX	Agente ZABBIX
AIX	Suportado	Suportado
FreeBSD	Suportado	Suportado
HP-UX	Suportado	Suportado
Linux	Suportado	Suportado
Mac OS X	Suportado	Suportado
Novell Netware	-	Suportado
Open BSD	Suportado	Suportado
SCO Open Server	Suportado	Suportado
Solaris	Suportado	Suportado
Tru64/OSF	Suportado	Suportado
Windows NT 4.0, Windows 2000, Windows 2003, Windows XP, Windows Vista	-	Suportado

Tabela 8 - Pré-requisitos de Software

Software	Versão
Apache	1.3.12 ou seguintes
PHP	5.0 ou seguintes
PHPmodules: php-gd	GD 2.0 ou seguintes
PHPTrueType support	
PHP bc support	
PHP XMLsupport	
PHPsession support	
PHP socket support	
PHP multibyte support	

Software	Versão
IBM DB2 ibm_db2	
MySQL **php-mysql**	3.22 ou seguintes
Oracle **oci8**	
PostgreSQL **php-pgsql**	7.0.2 ou seguintes se Zabbix < 1.8.9 7.4 ou seguintes se Zabbix >= 1.8.9
SQLite **php-sqlite3**	3.3.5 ou seguintes

Na Tabela 9, são mostrados os requisitos mínimos de hardware do sistema que aloja o Servidor, sendo que os requisitos de hardware e software variam de acordo com a dimensão da infraestrutura monitorizada. Na Tabela 10, temos um pequeno resumo do software necessário, hardware, e base de dados, de acordo com a dimensão da infraestrutura, relação direta com o número de *hosts* que são monitorizados.

Tabela 9 - Requisitos mínimos do hardware para Servidor Zabbix

Requisitos	Minino	Recomendado
Espaço em Disco	10 MB	100 MB
RAM	64 MB	256 MB
CPU	Pentium	Pentium IV ou equivalente

Tabela 10 - Resumo de requisitos do Zabbix de acordo com dimensão da infraestrutura

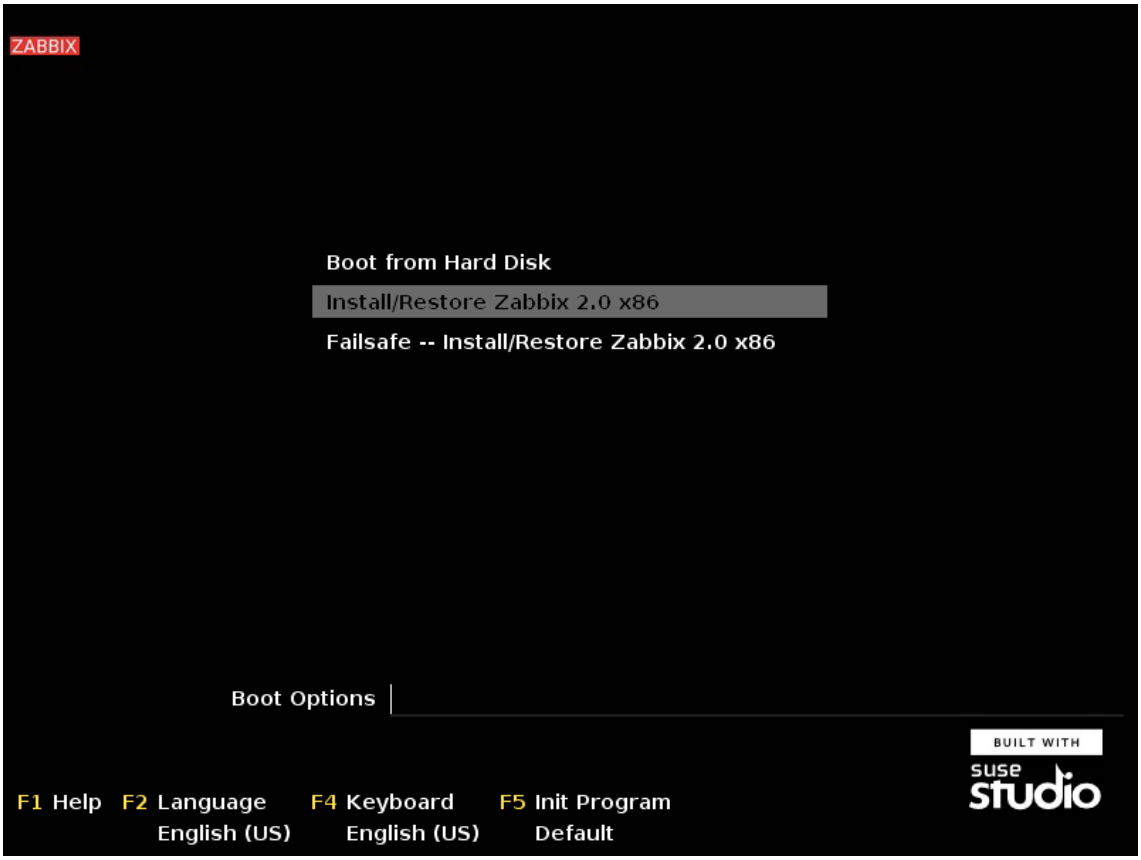
Dimensão	Plataforma	CPU/Memória	Base de dados	*Hosts* a Monitorizar
Pequena	Ubuntu Linux	PII 350MHz 256MB	SQLite	20
Média	Ubuntu Linux 64 bit	AMD Athlon 3200+ 2GB	MySQL InnoDB	500
Grande	Ubuntu Linux 64 bit	Intel Dual Core 6400 4GB	RAID10 MySQL InnoDB or PostgreSQL	>1000
Muito grande	RedHat Enterprise	Intel Xeon 2xCPU 8GB	Fast RAID10 MySQL InnoDB or PostgreSQL	>10000

4.3.6 Instalação

Durante a instalação não foram encontradas dificuldades de maior no que diz respeito ao servidor Zabbix, pois no site oficial da ferramenta existe documentação de

qualidade, apesar disso nos testes iniciais foi usada uma appliance disponível para download na página oficial da ferramenta. Optei então pelo download da máquina virtual para VMware / VirtualBox (.vmdk), de forma a minimizar o tempo nas configurações iniciais.

Ilustração 15 - Vista inicial da appliance Zabbix



Por defeito a appliance utiliza DHCP para obter um endereço IP. Para definir um endereço IP estático, foi necessário fazer a autenticação com o utilizador "root", ver Ilustração 16, utilizando para isso a palavra passe que vem por defeito "zabbix", ver Tabela 11. Depois foi necessário editar o ficheiro "/etc/sysconfig/network/ifcfg-eth0", que contem as configurações do interface de rede eth0, e definir a variável BOOTPROTO para "static". Defini o endereço IP e a máscara de rede, de acordo com os parâmetros de rede da DURIT, de forma a poder ter um IP fixo na máquina virtual. Foi ainda necessário criar o ficheiro "/etc/sysconfig/network/routes", e definir que a rota por defeito, "default 192.168.0.38" gateway que a máquina virtual iria utilizar.

Depois, destas configurações foi necessário reiniciar os serviços de rede utilizando o comando: "rcnetwork restart".

Ilustração 16 - Vista após autenticação no Zabbix com o utilizador "root"

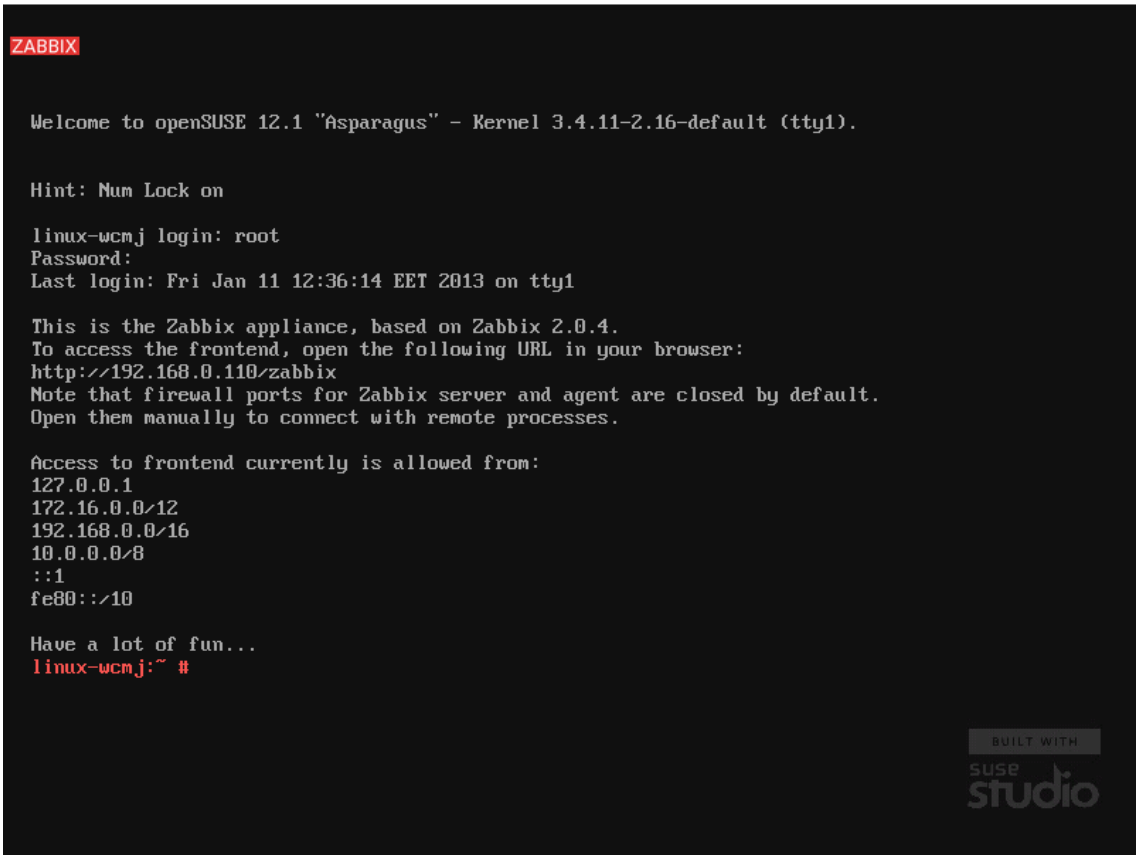


Tabela 11 - Passwords por defeito da appliance Zabbix

Local	Utilizador:password
Sistema	root:zabbix zabbix:zabbix
Base de dados	root:zabbix zabbix:zabbix
Interface web Zabbix	Admin:zabbix

Nas configurações que a appliance tem por defeito, apenas duas portas estão abertas da firewall, a 22 (SSH) e a 80 (HTTP).

Para abrir portas adicionais, por exemplo, para o servidor Zabbix e para o agente zabbix - modifiquei as regras de iptables com o utilitário SuSEfirewall2, utilizando o seguinte comando: "SuSEfirewall2 open EXT TCP zabbix-trapper zabbix-agent".

Para efetivar a alterações foi necessário reiniciar os serviços da firewall, através dos comandos:

"SuSEfirewall2 stop"

"SuSEfirewall2 start"

Simplicidade das configurações iniciais, foram extremamente úteis, pois permitiram começar a obter resultados muito rapidamente.

A partir deste momento, o Zabbix está pronto para poder ser acedido via browser, bastando colocar na barra de endereços o IP, da máquina virtual seguido de "/zabbix".

Ilustração 17 - Vista de autenticação no interface web do Zabbix

4.4 Escolha da solução para produção

Pode-se afirmar que a nível de instalação o Nagios “perde um pouco” em relação ao Zabbix, principalmente porque para o ambiente de testes em máquinas virtuais, optei pela utilização de uma *appliance* do Zabbix, o que proporcionou, a obtenção de resultados muito rapidamente.

Ao nível da configuração e adição de *hosts*, o Zabbix impõe-se claramente, quer pela simplicidade com que se adicionam *host*s, quer pelo seu acessível manuseio no interface web, sendo esta a principal razão da escolha desta solução, pois pretendia-se uma ferramenta que no futuro fosse mais fácil de gerir.

No caso dos agentes, no Zabbix, o processo limitou-se a fazer o download dos agentes disponibilizados no site oficial e na alteração de um ficheiro de configuração do agente (que contem informação do servidor, e configurações relativos à própria máquina). O Nagios, tem a instalação dos agentes nos variados sistemas relativamente simples, pois tal como para o servidor, a documentação existente é percetível e de fácil seguimento, no entanto, é preciso ter um mínimo de conhecimentos de Linux, e como a infraestrutura da Durit, é baseada em sistemas Windows, a familiarização com sistemas

operativos Linux, acaba por ser menor, dificultando sempre um pouco mais o adicionar de novos *hosts*. Como já tinha sido referenciado, é necessário criar um ficheiro conforme o Anexo I, para cada *host* no próprio servidor, ou em alternativa criar o ficheiro num cliente de rede e fazer o seu envio por FTP.

A possibilidade de utilização de *templates*, é também uma das grandes armas do Zabbix, graças a eles podemos rapidamente, gerir os parâmetros de monitorização de máquinas semelhantes, e se for necessário alguma alteração, esta é automaticamente replicada para todos os *hosts* que utilizem o mesmo *template*.

A nível do interface *web*, o do Zabbix, está muito bem estruturado, e os menus de navegação facilitam muito a vida do administrador do sistema, pela sua fácil compreensão. O Nagios, perde bastante neste aspeto, pois a distribuição da informação acaba por ser algo confusa, na sua versão comercial (Nagios XI), o interface torna-se bastante mais amigável, mas passamos a ter um custo associado à ferramenta, que se pretendeu evitar neste projeto.

4.5 Implementação da solução para produção

Após terem sido feitos vários testes nas duas ferramentas instaladas em ambientes virtuais, foi então iniciado o processo de instalação do Zabbix, para ficar em produção.

4.5.1 Instalação do Sistema Operativo

Para a instalação do ambiente onde o servidor Zabbix ficará em produção, foi utilizado o sistema operativo Ubuntu 12.04 LTS, dado que para servidores, o período de suporte às versões LTS, é muito superior (5 anos). Logo após a instalação do sistema operativo, foi definido um endereço IP fixo, na placa de rede que liga à rede local, para isso foi necessário editar o ficheiro "/etc/network/interfaces" com privilégios de root. O interface eth0, ficou então com o endereço IP 192.168.0.33, na rede 192.168.0.255.

```
auto eth0
iface eth0 inet static
    address 192.168.0.33
    netmask 255.255.255.0
    network 192.168.0.255
```

O interface eth1, ficou configurado para obter IP de forma dinâmica por DHCP, na rede 10.10.10.255, que é utilizada apenas para fazer atualizações ao sistema.

Tendo conectividade com o exterior, foi então necessário preparar o servidor, dotando-o dos pacotes necessários ao Zabbix.

Antes de iniciarmos a instalação, atualizam-se os pacotes do sistema, utilizando os comandos:

```
# apt-get update
# apt-get upgrade
```

Seguidamente, foram instalados todos os pacotes necessários ao Zabbix.

```
# apt-get install mysql-server mysql-client libmysql++-dev gcc make libmysqlclient-dev apache2 libiksemel-dev libiksemel-utils libsnmp-dev fping snmpd lm-sensors libsysfs2 php5 libapache2-mod-php5 php5-gd php5-snmp
```

```
php5-mysql php-pear perl-base liburi-perl libapache2-mod-perl2 libwww-perl
libtool libextutils-pkgconfig-perl pkg-config libsnmp9-dev libcurl4- openssl-dev
libcurl3 rcconf libgd-text-perl php5-cgi perl-modules libpdf- api2-perl libssh2-1-
dev
```

4.5.2 Instalação do Zabbix

Com o servidor preparado para a instalação do Zabbix, procedeu-se ao *download* da versão mais recente e estável do Zabbix, a partir do site oficial: http://www.zabbix.com/download.php.

```
# cd /usr/src/

# wget –c <link obtido na página de download>
```

Neste momento, a máquina encontrava-se pronta para se iniciar verdadeiramente a instalação do Zabbix. A instalação do Zabbix pressupõe vários passos, criação de uma conta para o serviço, a compilação do Zabbix, a criação da sua base de dados no MySQL, a criação das tabelas na base de dados, e por fim à instalação da aplicação. Todos estes passos foram executados conforme o Anexo II.

A instalação do Zabbix, acabou por ser um processo pouco moroso, mas foi necessário efetuar alguns ajustes, ao nível dos serviços, ou permissões em pastas, conforme documentado no Anexo III.

No Anexo IV, podemos ver a configuração do servidor Zabbix, passo a passo. Todas as fases e parametrizações efetuadas encontram-se documentadas neste anexo. Após a conclusão da instalação do Zabbix, este fica então pronto para ser acedido via browser, através do <endereço ip da máquina>/zabbix, aparecendo um formulário de autenticação conforme ilustração 18.

Ilustração 18 - Formulário de autenticação no Zabbix

4.5.3 Templates no Zabbix

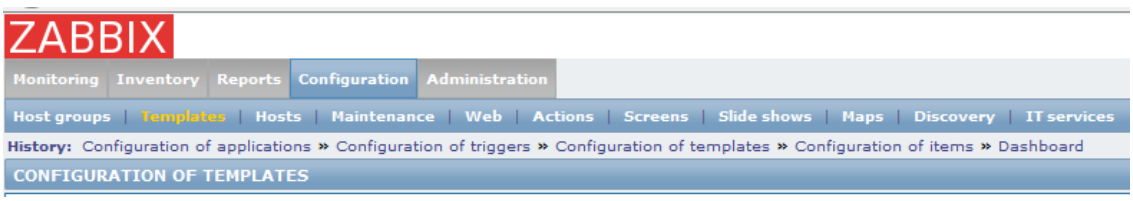
Para a monitorização dos ativos, foi necessário proceder à criação dos *templates*, no Zabbix, *templates* são a definição de um modelo de regras de recolha, níveis de alertas e representações gráficas que podem ser aplicadas facilmente a elementos monitorizados.

O Zabbix, tem por defeito alguns *templates* criados, que são um ótima base de trabalho. Os *templates*, permitem agrupar máquinas semelhantes, para que os critérios de monitorização e alertas sejam semelhantes, de acordo com as características das máquinas. Na ilustração 18, encontra-se a vista onde podemos ter uma perceção de todos os *templates* que temos instalados no nosso sistema.

Para se conseguir visualizar as configurações dos *templates* (e todo o restante menu de configurações) é necessário utilizar um utilizador com perfil, no mínimo, de administrador. Para editar e criar *templates* é ainda necessário possuir privilégios de gravação sobre o grupo ao qual o *template* está associado.

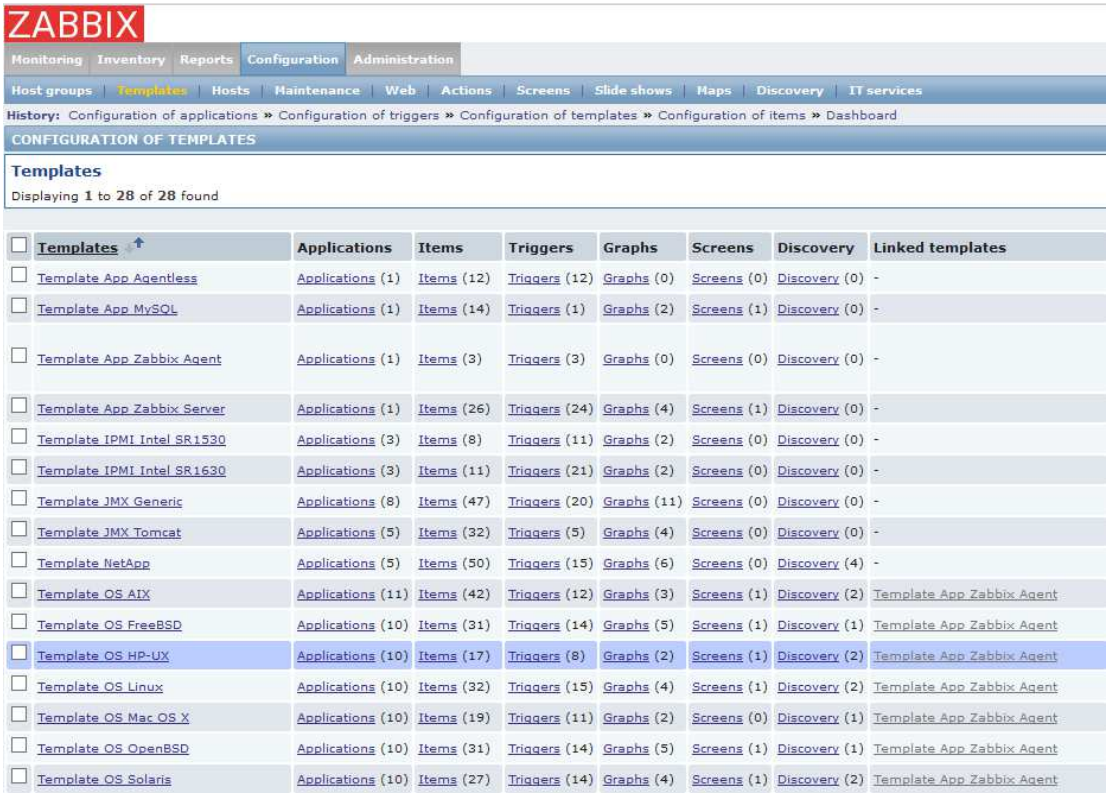
Para se aceder ao módulo de *templates* é necessário selecionar as opções: **Configuration -> Templates.** (Ver Ilustração 20).

Ilustração 19 - Vista de acesso aos *Templates*



Ao se clicar no menu *Templates* é-nos apresentada uma lista contendo os *templates* que estão registados no servidor e que o utilizador tenha permissões para visualizar. A lista possui colunas com resumo das características de cada *template* (Ver Ilustração 21).

Ilustração 20 - Lista de *Templates* do Sistema



Templates	Applications	Items	Triggers	Graphs	Screens	Discovery	Linked templates
Template App Agentless	Applications (1)	Items (12)	Triggers (12)	Graphs (0)	Screens (0)	Discovery (0)	-
Template App MySQL	Applications (1)	Items (14)	Triggers (1)	Graphs (2)	Screens (1)	Discovery (0)	-
Template App Zabbix Agent	Applications (1)	Items (3)	Triggers (3)	Graphs (0)	Screens (0)	Discovery (0)	-
Template App Zabbix Server	Applications (1)	Items (26)	Triggers (24)	Graphs (4)	Screens (1)	Discovery (0)	-
Template IPMI Intel SR1530	Applications (3)	Items (8)	Triggers (11)	Graphs (2)	Screens (0)	Discovery (0)	-
Template IPMI Intel SR1630	Applications (3)	Items (11)	Triggers (21)	Graphs (2)	Screens (0)	Discovery (0)	-
Template JMX Generic	Applications (8)	Items (47)	Triggers (20)	Graphs (11)	Screens (0)	Discovery (0)	-
Template JMX Tomcat	Applications (5)	Items (32)	Triggers (5)	Graphs (4)	Screens (0)	Discovery (0)	-
Template NetApp	Applications (5)	Items (50)	Triggers (15)	Graphs (6)	Screens (0)	Discovery (4)	-
Template OS AIX	Applications (11)	Items (42)	Triggers (12)	Graphs (3)	Screens (1)	Discovery (2)	Template App Zabbix Agent
Template OS FreeBSD	Applications (10)	Items (31)	Triggers (14)	Graphs (5)	Screens (1)	Discovery (1)	Template App Zabbix Agent
Template OS HP-UX	Applications (10)	Items (17)	Triggers (8)	Graphs (2)	Screens (1)	Discovery (2)	Template App Zabbix Agent
Template OS Linux	Applications (10)	Items (32)	Triggers (15)	Graphs (4)	Screens (1)	Discovery (2)	Template App Zabbix Agent
Template OS Mac OS X	Applications (10)	Items (19)	Triggers (11)	Graphs (2)	Screens (0)	Discovery (1)	Template App Zabbix Agent
Template OS OpenBSD	Applications (10)	Items (31)	Triggers (14)	Graphs (5)	Screens (1)	Discovery (1)	Template App Zabbix Agent
Template OS Solaris	Applications (10)	Items (27)	Triggers (14)	Graphs (4)	Screens (1)	Discovery (2)	Template App Zabbix Agent

A informação da lista está distribuída da seguinte forma:

Nome do *template* e o link para editar as suas propriedades, links com outros *templates*, macros do *template*, etc;

Número de aplicações e o link para manter o seu registo no *template* (as aplicações são uma forma de agrupar itens no Zabbix);

Quantidade de Itens e o link para os seus dados no *template*;

Número de *Triggers* (gatilhos)e o link para os gerir no *template*;

Total de Gráficos e o link para os gerir no template;

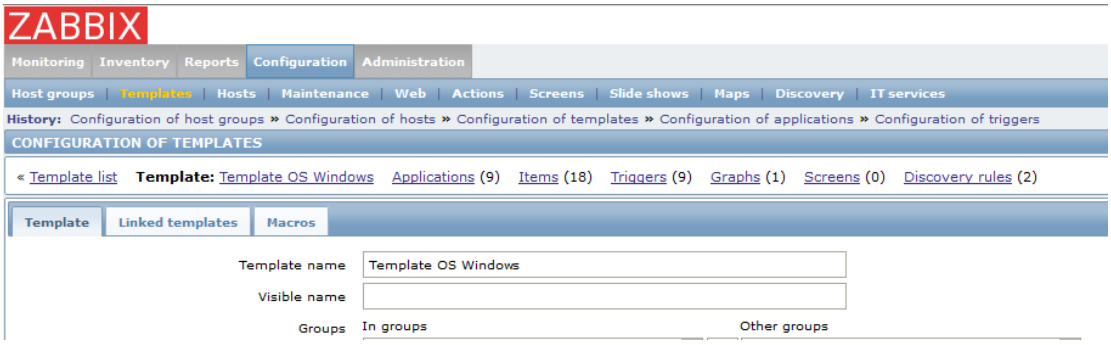
Número de *Screens* (écrans) e o link para os gerir dentro do template;

Número total de pesquisa automáticca (Low Level Discovery – LLD) e o link para gerir no próprio *template*;

Link para acesso rápido a outros *templates* que são herdados e fazem parte do template atual;

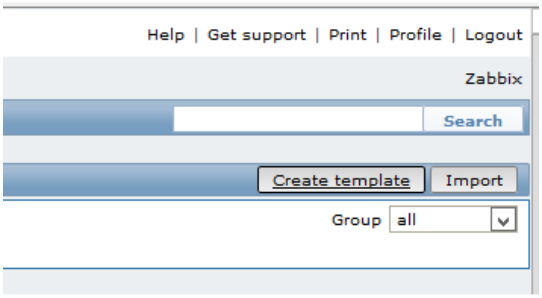
Lista de *hosts* associados ao *template*;

Ilustração 21 - Vista de configuração de Templates



A criação de novos *templates* no Zabbix, é um processo simples, já que quando estamos na gestão de *templates*, existe um botão "Create template"(Ver Ilustração 22), que irá abrir uma nova vista para esse efeito.

Ilustração 22 - Criar novo *Template*



O processo de criação de templates está documentado no Anexo VII.

Na Ilustração 23 - Exemplo de Template - Sistema Operativo Windows, temos um exemplo do tamplate para máquinas com Sistema Operativo Windows. Nesta vista aparece-nos uma lista de ítens, e é onde podemos verificar cada parametro que irá ser monitorizado. Aqui serão ajustados parametros como a "chave" (Key), que é utilizada pelo Servidor Zabbix, para solicitar a informação ao agente, o "intervalo" (Interval) de tempo, que é em segundos, a periocidade com que será feita a recolha de informção, o parametro histórico (History), permite definir quantos dias serão guardados os dados recolhidos e o parametro de estatisticas (Trends), onde é possivel definir durante quanto

tempo se guarda a informação em termos estatisticos. Existe também um parametro "estado" (Status), onde podemos definir o item monitorizado com "ativo" ou "destivado" (Enabled/Diasabled). Este ultimo iten ao ser definido no template, é definido para os *hosts* associados a ele, pelo que é recomendável que este ajuste seja feito ao nível do *host*.

Ilustração 23 - Exemplo de Template - Sistema Operativo Windows

« Template list **Template:** Template OS Windows Applications (9) Items (18) Triggers (9) Graphs (1) Screens (0) Discovery rules (2)

Wizard	Name	Triggers	Key	Interval	History	Trends	Type	Applications
	Template App Zabbix Agent: Agent ping	Triggers (1)	agent.ping	60	7	365	Zabbix agent	Zabbix agent
	Average disk read queue length		perf_counter[\234(_Total)\1402]	60	7	365	Zabbix agent	Filesystems, Performance
	Average disk write queue length		perf_counter[\234(_Total)\1404]	60	7	365	Zabbix agent	Filesystems, Performance
	File read bytes per second		perf_counter[\2\16]	60	7	365	Zabbix agent	Filesystems, Performance
	File write bytes per second		perf_counter[\2\18]	60	7	365	Zabbix agent	Filesystems, Performance
	Free memory	Triggers (1)	vm.memory.size[free]	60	7	365	Zabbix agent	Memory
	Free swap space	Triggers (1)	system.swap.size[,free]	60	7	365	Zabbix agent	Memory
	Template App Zabbix Agent: Host name of zabbix_agentd running	Triggers (1)	agent.hostname	3600	7		Zabbix agent	Zabbix agent
	Number of processes	Triggers (1)	proc.num[]	60	7	365	Zabbix agent	Processes
	Number of threads		perf_counter[\2\250]	60	7	365	Zabbix agent	OS
	Processor load (1 min average)	Triggers (1)	system.cpu.load[,avg1]	60	7	365	Zabbix agent	CPU, Performance
	Processor load (5 min average)		system.cpu.load[,avg5]	60	7	365	Zabbix agent	CPU, Performance
	Processor load (15 min average)		system.cpu.load[,avg15]	60	7	365	Zabbix agent	CPU, Performance
	System information	Triggers (1)	system.uname	3600	7		Zabbix agent	General, OS

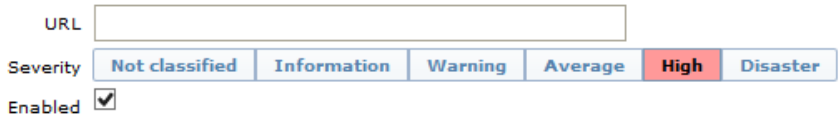
Os *triggers* são um dos parâmetros mais importantes do nosso *template*, por isso mereceram uma atenção especial na implementação da solução.

A tradução de *trigger* para português é gatilho, este conceito de *trigger* (gatilho), é a terminologia adotada para a realização automatizada de procedimentos sempre que um evento acontecer. No Zabbix, os *triggers* são como uma “marca d´agua”, limites configuráveis para representar situações que temos que estar atentos.

Quando um *trigger* é acionado automaticamente o Zabbix replica esta informação para todos os mapas e *dashboards* onde o *host* ou os grupos ao qual o *host* está relacionado estejam representados. Além da representação gráfica do incidente ocorrido é também possível a representação sonora (beep) e a configuração de ações específicas, com o intuito de notificar os administradores ou até mesmo executar procedimentos automatizados de forma a minimizar o problema.

Os *triggers* são classificados de acordo com a criticidade no campo "Severity", podendo o seu risco ser considerado: "Information", "Warning", "Average", "High", "Disaster" ou "Not classified", conforme podemos verificar na Ilustração 24.

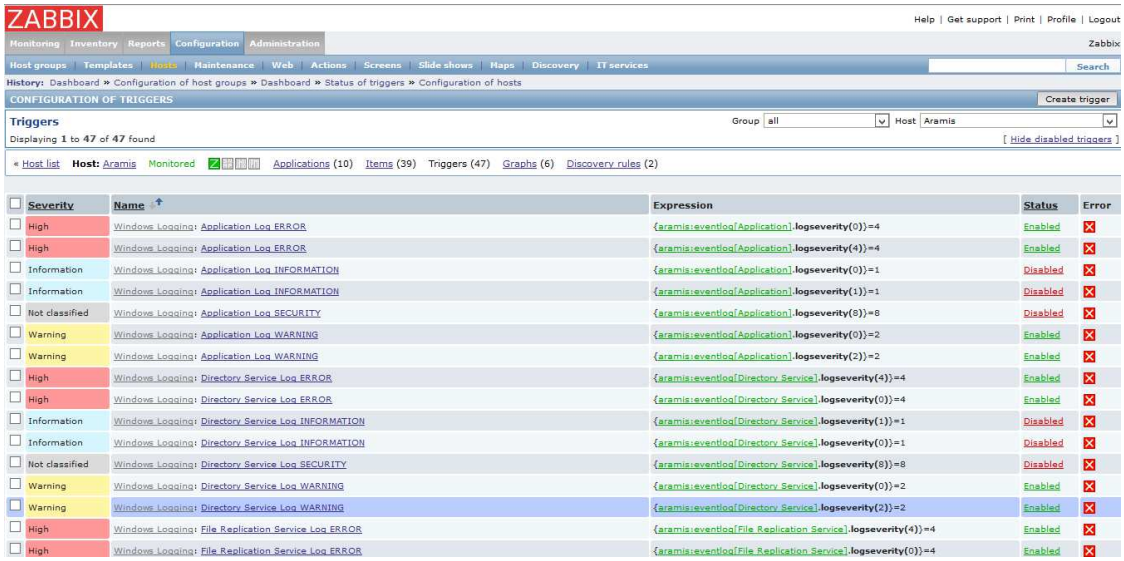
Ilustração 24 - Classificação do risco de um *Trigger*



A expressão que classifica o risco do *trigger*, está relacionada a um ou vários itens, esta funcionalidade permite juntar vários itens, num só *trigger* sendo estes separados por um "OU" lógico, sem ser necessário criar um *trigger* para cada item.

Na Ilustração 25, temos um exemplo de uma listagem dos *triggers* associados a um *host*. Na coluna "Severity" são usadas cores que podem ser personalizadas, para que graficamente seja mais fácil identificar alertas.

Ilustração 25 - *Triggers* associados a um *Host*



4.5.4 Registo de *Hosts*

O registo de novos *Hosts*, é um processo simples e bastante intuitivo, muito por força do formulário próprio para o registo de novos *hosts* (ver Ilustração 27) e da existência *templates*, sendo as informações mais importantes o "DNS name" ou o "IP address", para que possamos dizer ao Zabbix, como deve ligar-se ao *host*, no parâmetro " Connect to".

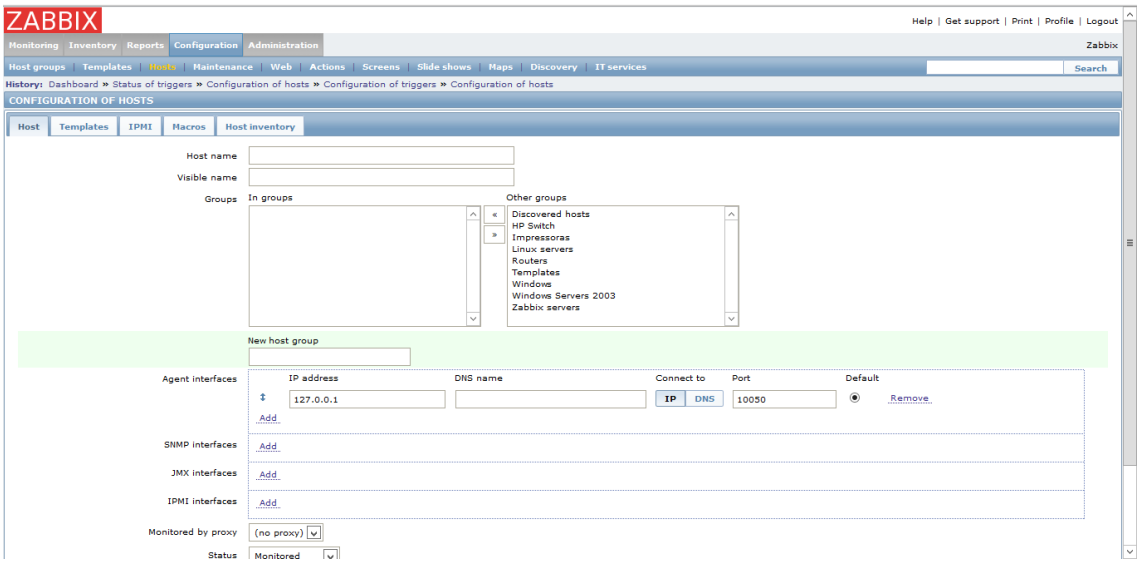
No separador *templates*, podemos associar um *host*, a um dos *templates* anteriormente criados, sendo que para isso temos uma interface gráfica com uma *checklist*, (ver Ilustração 26).

De uma forma geral os *templates* existentes para os Sistemas Operativos Linux e Microsoft Windows, cingem a recolha de informação sobre o estado e desempenho do *host* monitorizado, com utilização de processador e memória, utilização das placas de rede, espaço em disco disponível, integridade de ficheiros e informações sobre o *host*.

Por outro lado os templates criados para monitorizar serviços, para além de monitorizarem o consumo deles, também verificam a sua disponibilidade, e o tempo de resposta para os seus utilizadores.

Ilustração 26 - *Checklist* para atribuição de *templates* a um *Host*

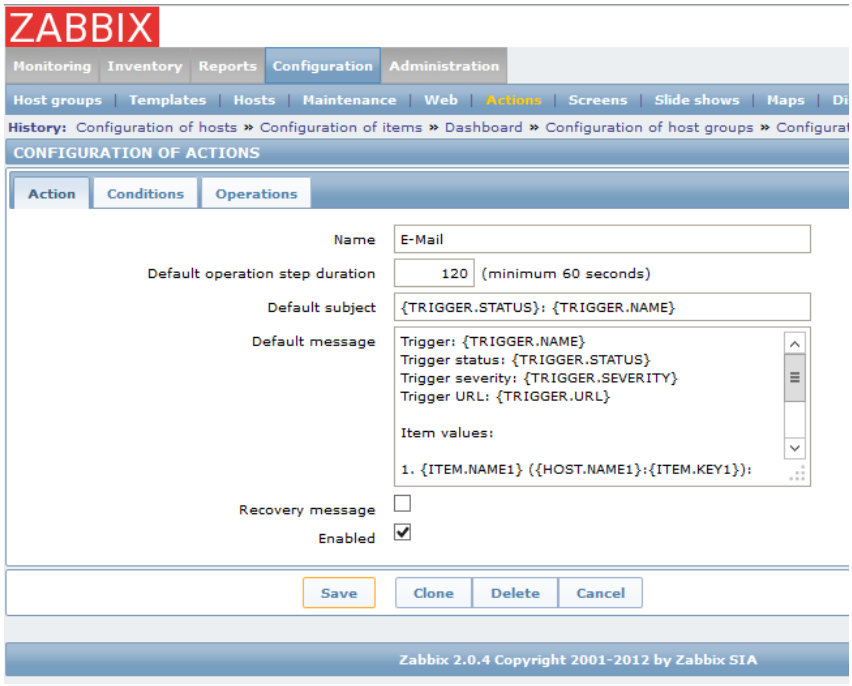
Ilustração 27 - Formulário para adicionar um novo *Host*



4.5.5 Ações

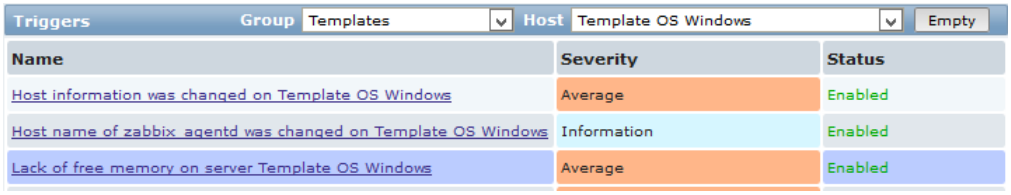
O registo de ações que devem ser executadas, feito através de um formulário (ver Ilustração 28), onde definimos um nome para a ação e os parâmetros para envio da informação por email. Neste menu existem dois separadores adicionais, um "Conditions" e outro "Operations", o primeiro, define as condições em que a nossa ação decorre, podendo estas serem ajustadas de acordo com o *host*, que pretendemos associar à ação.

Ilustração 28 - Vista para adicionar uma nova Ação



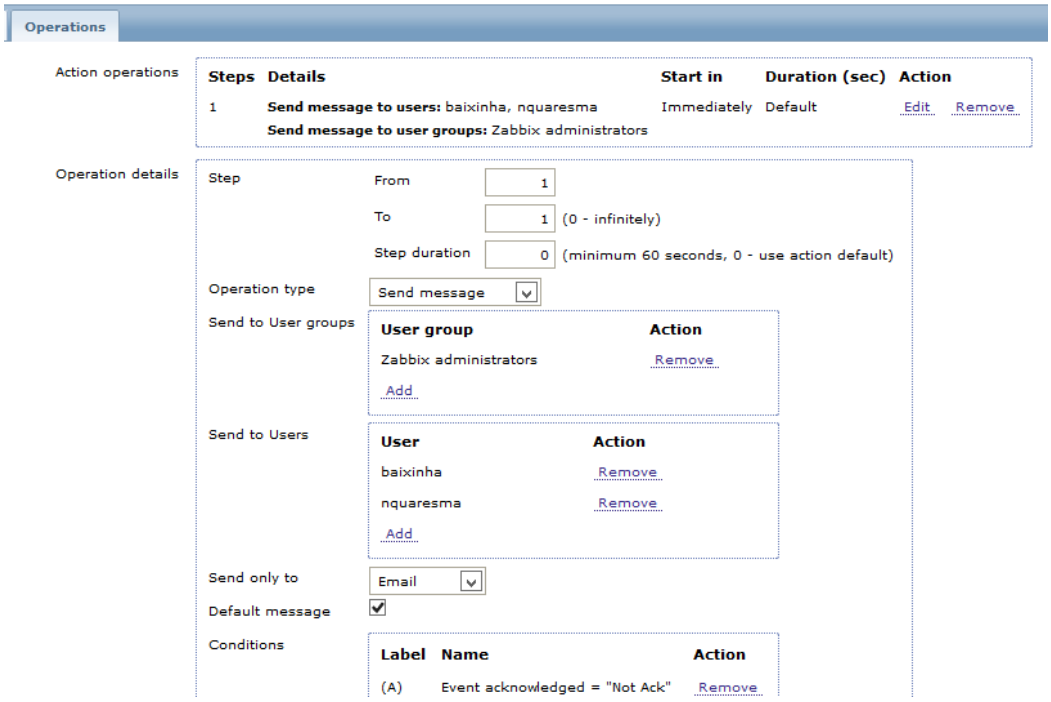
As condições, são os parâmetros definidos em que a ação decorre, podendo existir uma combinação de condições que levam a que um ação se realize. As condições estão associadas aos *triggers*, a título de exemplo pode ser definida uma ação, para quando a memória livre num *host*, for demasiado baixa. Esta escolha de condições é feita com base numa lista, onde surgem todos os *triggers* associados a um determinado *template*.

Ilustração 29 - Atribuição de *Triggers* a uma Ação



No separador "Operations", surgem as operações que a ação despoleta, sendo neste separador que definimos os utilizadores que serão notificados, e de que forma são notificados, quando a nossa ação decorrer (ver Ilustração 30).

Ilustração 30 - Vista de configuração de Operações



4.5.6 Resultados

No Zabbix, os resultados podem ser obtidos em tempo real, de uma forma geral pela *dashboard* (ver Ilustração 31), ou em alternativa de uma forma mais detalhada no separador "Latest Data", que nos permite visualizar a ultima informação recolhida (ver Ilustração 32). No separador "Latest Data", temos informação de quando foi executada a recolha de informação pela ultima vez, o último valor da captura, e a variação deste valor em relação à recolha feita anteriormente. Na nossa listagem temos uma coluna "History", onde é gerado um gráfico referente ao parâmetro em causa, em alternativa, é possível visualizar apenas os valores numa tabela. Todos os itens associados aos *Hosts*, podem ser visualizados de uma destas formas, tornando-se mais fácil prevenir problemas e prever as necessidades que se possam vir a verificar no futuro.

Ilustração 31 - *Dashboard* do Zabbix implementado

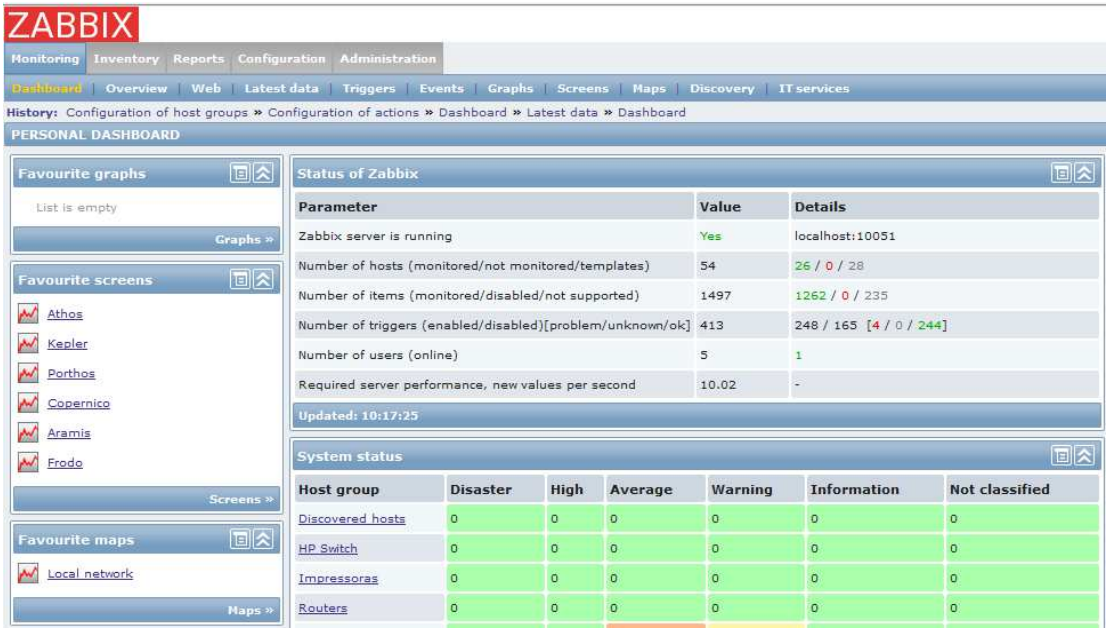
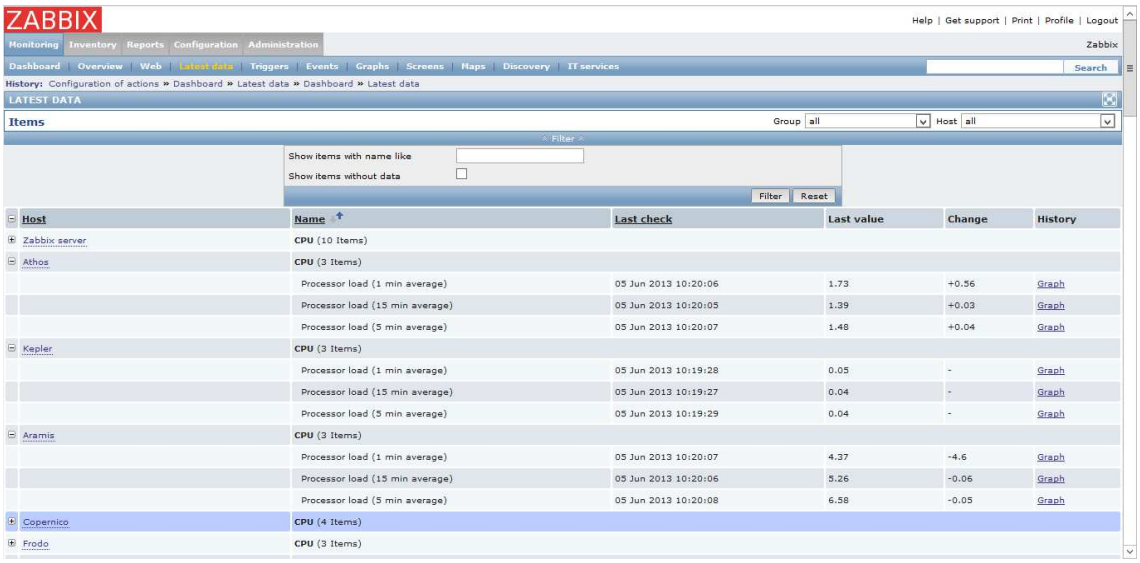


Ilustração 32 - Vista da ultima informação recolhida



5 Cronograma

5.1 Calendarização

O cronograma imediato realça o esforço estimado para cada uma das fases previstas, bem como a sua sequência:

ID	*Nome da Tarefa*	*Inicio*	*Fim*	*Duração*
	Estágio DURIT	**17-09-2012**	**28-06-2013**	**205d**
	Inicio	17-09-2012	17-09-2012	1d
	Criação do sitio web para o projeto	01-10-2012	05-10-2012	5d
	Pesquisa e Levantamento das Necessidades	17-09-2012	26-10-2012	30d
	Elaboração do esquema da rede	29-10-2012	02-11-2012	5d
	Levantmento dos serviços implementados em cada servidor e definição de serviços críticos	05-11-2012	30-11-2012	20d
	Elaboração do relatório Inicial	03-12-2012	17-01-2013	34d
	Entrega do relatório Inicial	18-01-2013	18-01-2013	1d
	Implementação	**17-12-2012**	**22-01-2013**	**26d**
	Preparação das máquinas	17-12-2012	20-12-2012	3d
	Instalação da solução de monitorização	20-12-2012	27-12-2012	5d
	Configuração de sensores a monitorizar	27-12-2012	17-01-2013	15d
	Realização de testes e análise de relatório obtidos	17-01-2013	22-01-2013	3d
	Fase de testes	**22-01-2013**	**14-06-2013**	**103,5d**
	Testes finais do projeto	22-01-2013	15-02-2013	18d
	Conclusão do projeto	15-02-2013	21-02-2013	4,5d
	Actualização do SO e últimos testes	10-06-2013	14-06-2013	5d
	Elaboração do relatório final	20-02-2013	28-06-2013	93d
	Elaboração da apresentação do Projeto	20-06-2013	28-06-2013	6,5d

Para a criação do sitio Web do projeto, foi utilizado o Wordpress, um dos muitos CMS's existentes pelo que a sua implementação será rápida e de fácil gestão. A escolha recaiu sobre o Wordpress, apenas por ser uma escolha pessoal, uma vez que não será alvo de estudo durante o projeto.

A elaboração do esquema de rede, tornou-se algo complexa, uma vez que o Grupo DURIT, engloba 8 empresas em território nacional, no entanto esta fase foi de extrema importância, para o sucesso da implementação.

A elaboração do relatório inicial foi iniciada logo após a recolha de informação sobre a infraestrutura da empresa, esta fase foi algo morosa, pois foi necessário a leitura de numerosa documentação sobre as tecnologias e conceitos envolvidos. Logo que foi possível, foi entregue o relatório inicial que serviu como base ao restante projeto.

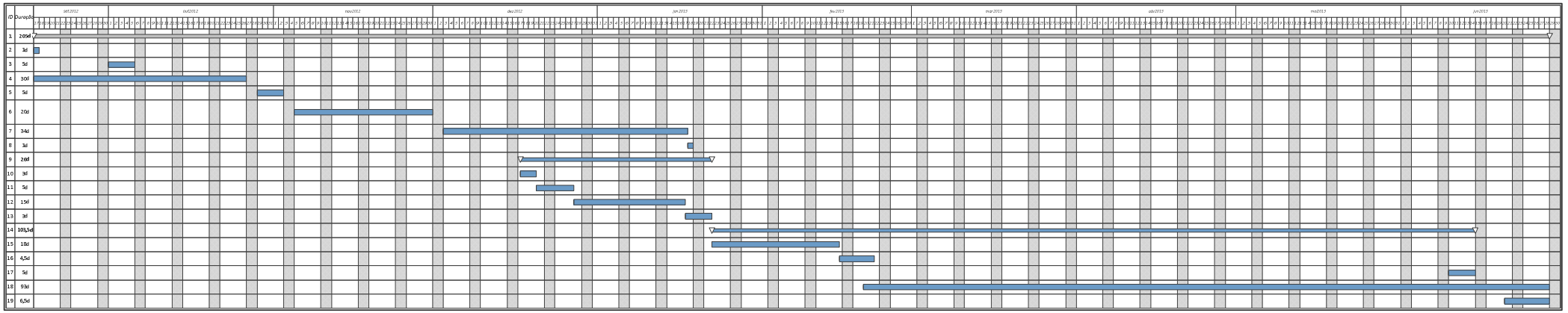
A implementação foi uma fase algo demorada, visto que nesta fase foram feitos testes a soluções possíveis, de forma a poder ser feita a melhor escolha para a infraestrutura de empresa.

Depois da implementação da solução escolhida para foram feitos testes em produção, com inúmeras simulações. Esta fase acabou por demorar bastante tempo, pois só com estes foi possível recolher informação fidedigna dos ativos da infraestrutura. A fase de testes, acabou por ser um período longo, uma vez que foram feitos pequenos ajustes na solução, para obtenção dos resultados desejados. Não foi uma tarefa feita de forma sistemática, acabando por ter um interregno relativamente longo, entre a os testes feitos no Zabbix, e a atualização final da solução, que ficará em funcionamento na empresa.

A elaboração do Relatório final foi outra fase algo demorada, pois como era previsível, aquando da implementação da solução escolhida foi necessário fazer ajustes, e nesta fase houve também necessidade de alguma revisão de literatura sobre as tecnologias adotadas.

A preparação da apresentação é a ultima fase do cronograma previsto, com a entrega da documentação final para defesa do trabalho realizado.

5.2 Mapa de Gantt



6 Meios necessários

Os meios necessários à realização deste projeto podem ser divididos entre recursos humanos e materiais.

Os recursos materiais prendem-se com as necessidades, relacionadas com a tecnologia a implementar, estando subdivididos em Software e Hardware, sendo as necessidades, as seguintes:

Software:

Microsoft Windows Server;

Clientes Microsoft Windows XP (Service Pack 3) / Windows 7;

Ubuntu LTS Server Edition;

Apache Web Server;

Bases de dados MySQL

VirtualBox;

FreeRadius;

Nagios Core;

Zabbix.

A escolha recaiu sobretudo sobre software livre, de forma a que não existam custos associados à utilização destas tecnologias, a licença do software é GNU (General Public License).

Hardware:

Servidores (da empresa);

Computador portátil pessoal;

Computador para instalação de todas as máquinas virtuais;

Software de Virtualização;

Router Wireless com suporte de autenticação Radius;

Restante equipamentos que serão monitorizados;

Switch Layer 3 (da empresa).

Os recursos humanos para o projeto foram, o aluno que desenvolve o projeto, o orientador interno e a orientador externo (empresa).

A estes recursos, estão associados os custos relativos ao tempo despendido, no desenvolvimento do projeto, tempo utilizado pelo orientador, na sua função, bem como o orientador da empresa onde me encontro a efetuar o estágio.

7 Análise de resultados

Neste capitulo é feita uma análise critica aos sistemas testados, e divulgados os resultados obtidos pela comparação dos sistemas. No final uma tabela resume as principais conclusões tiradas.

7.1 Implementação / divulgação

Através das inúmeras pesquisas feitas ao longo do projeto, foi possível concluir que o Nagios e o Zabbix, estão no topo da lista das melhores ferramentas de monitorização, sendo que em alguns aspetos, o Zabbix surge melhor colocado que o Nagios e noutros acontece exatamente o contrário.

O Nagios tem um comunidade de partilha de conhecimentos muito superior, sendo mais popular entre os entusiastas das ferramentas *opensource* e de monitorização de redes informáticas.

7.2 Facilidade de instalação

No que diz respeito à instalação dos sistemas de monitorização, é possível concluir que a instalação do servidor Zabbix em Ubuntu 12.04 LTS, possui pouca documentação que sirva de auxilio à implementação, pois a distribuição escolhida ainda é relativamente recente. No entanto, a existência de documentação de distribuições anteriores, permitiu a concretização da instalação, sendo que acabou por ser necessário um maior esforço e empenho.

Contrariamente o Nagios, teve uma instalação relativamente simples, visto que existe muito mais documentação, quer aquela que é possível encontrar em inúmeros *web sites*, relacionados, quer nos diversos livros editados, que podemos facilmente encontrar numa livraria, ou biblioteca.

Em suma é possível concluir que fruto da elevada qualidade e quantidade de documentação existente, a instalação do Nagios, tornou-se mais simples.

7.3 Configuração

No ponto anterior, foi referido que existe muito mais documentação sobre o Nagios, no entanto a configuração do Zabbix, é bastante simples e intuitiva. A instalação dos agentes nos *hosts*, é um processo tão simples, como fazer o *download* do agente de acordo com o sistema operativo, editar um pequeno ficheiro de configuração, e correr o aplicativo que instala o serviço.

O processo de adicionar *hosts*, no Zabbix, pode resumir-se a simples cliques de rato, sendo que no Nagios, é sempre necessário editar um ficheiro para cada *host*, obrigando o administrador do sistema a ter um grau de conhecimento muito maior, que tem de saber o que editar nos ficheiros e quais deve editar. processo este que obriga sempre a reiniciar o serviço.

Conclui-se que apesar da documentação superior existente sobre o Nagios, o processo de configuração e adição de novos *hosts*, impõe-se claramente no Zabbix, pela simplicidade que o seu interface web nos proporciona.

Também o processo de expansão, quer seja com a adição de novos *hosts* ou replicação da informação para um outro servidor, é muito mais simples no Zabbix, e obriga a um menor conhecimento da ferramenta.

7.4 Informação

O Zabbix, tem o seu interface web, bem estruturado e organizado, servindo-se de menus que facilitam a sua compreensão e administração. A informação surge-nos bem estruturada, com uma barra inicial, que permite aceder a submenus com um simples passar do rato, tornando-se muito fácil e rápido encontrar a opção pretendida. Esta simplicidade de funcionamento permite que a ferramenta possa ser usada por alguém menos experiente, sendo a sua aprendizagem muito fácil. O Zabbix, permite ainda criar utilizadores, que pelas suas permissões vão poder ou não aceder a determinadas opções do sistema.

Em contrapartida a toda a simplicidade evidenciada pelo Zabbix, surge-nos o Nagios, que na sua versão Core, acaba por ter um acesso à informação algo complicado e confuso, uma vez que a organização dos menus e a forma como estes são utilizados torna o uso da ferramenta, muito menos intuitivo. O interface web do Nagios Core, tem um aspeto muito mais rudimentar e menos amigável para o seu utilizador.

No que diz respeito à informação que se pode obter, qualquer das ferramentas disponibilizam muita informação, sendo que no Nagios, sempre que pretendemos adicionar mais itens a monitorizar, é necessário proceder à edição de ficheiros. Outro aspeto menos positivo é que alguns valores, podem tornar-se pouco percetíveis, já que a informação gerada, é apresentada em forma de texto.

Resumindo, pode-se afirmar que em termos de informação gerada, e forma de aceder à informação, o Zabbix, é superior ao Nagios, sendo uma ferramenta mais amigável e com facilidade de aprendizagem.

7.5 Custos de implementação

Um dos requisitos do projeto era utilizar ferramentas gratuitas ou *opensource*, o que foi cumprido em ambas as plataformas implementadas. No entanto, concretamente no caso do Nagios, existe uma versão paga, o Nagios XI, que poderia aproximar de forma significativa a ferramenta, das principais vantagens evidenciadas no Zabbix.

7.6 Sistemas Operativos suportados

No que diz respeito aos Sistemas Operativos suportados pelas ferramentas, estas são ferramentas semelhantes, a nível dos *hosts* que permitem monitorizar ambas suportam sistemas operativos Windows, o que era fundamental para o sucesso do projeto. Outro fator era o de permitirem a instalação do servidor numa distribuição Linux, devido aos custo de software, algo que se tornou irrelevante na escolha da ferramenta uma vez que ambas necessitam de ser instaladas numa distribuição Linux.

7.7 Considerações Finais

A implementação da ferramenta Zabbix, trouxe ganhos na capacidade de prever a evolução de necessidades, e de reagir mais rapidamente a problemas que possam surgir, quer através do envio de alertas, quer com a possibilidade de corrigir falhas nos *hosts*, quando estes permitem ações.

Dado que o Grupo DURIT, possui várias empresas espalhadas geograficamente, a utilização do Zabbix, pode vir a possibilitar uma gestão centralizada, mesmo que seja

implementado um servidor em cada empresa, já que é uma ferramenta escalável e permite a interligação de vários servidores.

Com o intuito de facilitar a compreensão, da comparação das duas ferramentas, foi elaborada a Tabela 12, onde as ferramentas foram classificadas com os níveis (Mau, Razoável, Bom e Excelente).

Tabela 12 - Comparativo entre o Nagios e o Zabbix

	Nagios	**Zabbix**
Implementação / Divulgação	Excelente	Bom
Instalação	Bom	Bom
Configuração	Razoável	Excelente
Informação	Bom	Excelente
Custos de implementação	Excelente	Excelente
Sistemas Operativos	Excelente	Excelente

Como é possível verificar na tabela anterior, facilmente se percebe o porquê do Zabbix ter sido a ferramenta escolhida para a implementação final. O Zabbix, é acima de tudo uma ferramenta muito mais amigável, de configuração mais simples, e fundamentalmente de onde é mais fácil recolher a informação de uma forma percetível. O Zabbix, acabou por se mostrar uma ferramenta mais completa e intuitiva.

8 Conclusões

Da realização deste projeto, foi possível retirar várias conclusões. A monitorização de uma infraestrutura informática, é uma área muito abrangente, e obriga a um grande estudo das tecnologias envolvidas. Bons conhecimentos a nível técnico são fundamentais, mas um estudo prévio da organização e da infraestrutura onde se pretende implementar o sistema de monitorização, também são imprescindíveis para o sucesso do trabalho final.

Ao longo do desenvolvimento do projeto, houve uma forte aprendizagem. O progresso foi moroso e lento, em grande parte devido ao processo de aprendizagem, mas também devido à muita informação de qualidade duvidosa que se encontra nas pesquisas feitas na internet. Mesmo com todas as dificuldades encontradas estou ciente de poder afirmar que foram atingidos os objetivos inicias do projeto.

O Zabbix, foi configurado com um conjunto de sensores importantes, que permitirão no futuro, uma melhor gestão da infraestrutura, permitindo uma boa perceção do seu crescimento, e antever necessidades que possam surgir.

A correta configuração do Zabbix, passou a permitir uma melhor gestão da infraestrutura, possibilitando em muitos casos uma ação preventiva, evitando e prevendo possíveis problemas.

A implementação desta solução para monitorizar a rede informática, deixou em aberto a possibilidade de outros trabalhos futuros, quer na manutenção da infraestrutura, quer na previsão de recursos, que possam vir a ser necessários.

A base para uma configuração de *hosts*, mais abrangente, de forma a abarcar todos os pontos críticos da infraestrutura foi conseguida, no entanto com as limitações de hardware existentes, não foram albergados todos os pontos críticos da infraestrutura, caindo a opção em pontos de referência, que permitiram o sucesso da implementação.

As funcionalidades mais avançadas da plataforma Zabbix, que acabou por ficar em produção, foram exploradas como referência, ficando pendentes para a restante infraestrutura, mas implementações que só serão possíveis, com melhorias no hardware disponível para a plataforma.

A alterações previstas para a infraestrutura de rede do grupo de empresas, nomeadamente a implementação de uma rede MPLS, que permitirá interligar as

diversas empresas do grupo, deixa em aberto a possibilidade de alargar a quantidade de *hosts* a monitorizar.

A realização deste projeto possibilitou um enriquecimentos pessoal, quer a nível de conhecimentos teóricos e práticos, quer ainda a nível de competências de trabalho em equipa e da capacidade de contornar obstáculos.

9 Bibliografia

(s.d.). (Zabbix SIA) Obtido em 17 de 05 de 2013, de Zabbix - The Enterprise-class Monitoring Solution for Everyone: http://www.zabbix.com

Allen, T. (s.d.). *Adding Hosts to Nagios*. Obtido em 21 de Março de 2013, de The Tech Tutorial: http://www.the-tech-tutorial.com/?p=414

Engelfriet, A. (2002-2005). *Choosing a software license.* Obtido em 23 de Abril de 2013, de Ius mentis Law and technology explained: http://www.iusmentis.com/computerprograms/licenses/choosing/#Standardlicenses

Faucheur, F., Wu, L., Davari, B., Vaananen, P., Krishnan, R., Cheval, P., & Heinanen, J. (Maio de 2002). *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*. Obtido em 18 de 12 de 2012, de http://www.hjp.at/doc/rfc/rfc3270.html

Free Software Foundation. (s.d.). Obtido em 11 de 03 de 2013, de http://www.fsf.org

ITIL - Information Technology Infrastructure Library. (s.d.). Obtido em 27 de 02 de 2013, de http://www.itil-officialsite.com/

Leiner, B., Vinton, C., Clark, D., Kahn, R., Kleinrock, L., Lynch, D., . . . Wolff, S. (Fevereiro de 1977). The Past and Future History of the Internet.

MPLSINFO. (s.d.). Obtido em 20 de 12 de 2012, de http://www.mplsinfo.org/

MUNIN. (s.d.). Obtido em 05 de 03 de 2013, de http://munin-monitoring.org/

NAV - Network Administration Visualized. (s.d.). Obtido em 05 de 03 de 2013, de https://nav.uninett.no/

NETDISCO - Network Management Tool. (s.d.). Obtido em 04 de 03 de 2013, de http://netdisco.org/

Olups, R. (2010). *Zabbix 1.8 Network Monitoring.* Packt Publishing.

Oracle. (s.d.). *VirtualBox*. Obtido em 10 de 1 de 2013, de https://www.virtualbox.org/

Silveira, S. A. (2003). *Software Livre e Inclusão Digital* (1ª Edição ed.). Conrad.

Simple Network Management Protocol. (s.d.). Obtido em 20 de 12 de 2012, de http://www.snmp.org

Software livre | o que é? (2007-2008). (ERTE/PT - Equipa de Recursos e Tecnologias Educativas / Plano Tecnológico da Educação) Obtido em 24 de Abril de 2013,

de Software Livre DGIDC: http://softlivre.dgidc.min-edu.pt/index.php?option=com_content&task=view&id=13&Itemid=81

Tanenbaum, A. (2003). *Computer Networks* (4ª Edição ed.). Pearson Education, Inc.

Teixeira, R. (1999). *Redes de Computadores, serviços, administração e segurança.* Makron Books.

Zabbix documentation for version 2.0. . (s.d.). Obtido em 08 de 01 de 2013, de Zabbix documentation: https://www.zabbix.com/documentation/2.0

ANEXOS

ANEXO I

Exemplo de configuração de um *host* no Nagios "nomedohost.cfg"

```
# Define a host for the local machine

define host{
      use                linux-server          ; Name of host template to use
                                               ; This host definition will inherit all variables that are defined
                                               ; in (or inherited by) the linux-server host template definition.
      host_name            google.com
      alias              google.com
      address              www.google.com
      }

##############################################################################
##############################################################################
#
# SERVICE DEFINITIONS
#
##############################################################################
##############################################################################

# Define a service to "ping" the local machine

define service{
      use                      generic-service       ; Name of service template to use
      host_name                google.com
      service_description      PING
      check_command            check_ping!100.0,20%!500.0,60%
      }

# Define a service to check HTTP on the local machine.
# Disable notifications for this service by default, as not all users may have HTTP enabled.

define service{
      use                      generic-service       ; Name of service template to use
      host_name                google.com
      service_description      HTTP
      check_command            check_http
      notifications_enabled    0
      }
```

ANEXO II

Instalação do Zabbix no servidor de produção.

Tendo sido efetuado o download da versão mais recente e estável do Zabbix, a partir de: http://www.zabbix.com/download.php, que é o site oficial da ferramenta, segue-se os seguintes passos:

```
# cd /usr/src/
# wget –c <link obtido na página de download>
```

Depois, foi necessário criar a conta do serviço zabbix, com o comando:

```
# adduser --no-create-home --disabled-password --disabled-login--shell=/bin/false zabbix
```

Compilação e instalação do Zabbix, e criação da base de dados.

Inicialmente descompactou-se e compilou-se o Zabbix com os comandos:

```
# cd /usr/src/
# tar -xzvf <nome_ficheiro_baixado>
# cd <nome_pasta_resultante_descompatacao>
# ./configure --enable-server --enable-agent --with-mysql --with-net-snmp --with-libcurl --with-ldap --with-ssh2 --enable- proxy --with-jabber --prefix=/usr/local/zabbix PKG_CONFIG_PATH=/usr/lib/pkgconfig/ PKG_CONFIG=/usr/bin/pkg- config
# make
```

Foi então criada a base de dados em MySQL:

```
# mysql -u root -p -e "create database zabbix;"
```

Configura-se acesso da conta zabbix à base de dados

```
# mysql -u root -p -e "GRANT ALL PRIVILEGES ON zabbix.* TO
zabbix@localhost IDENTIFIED BY 'NOVA_SENHA';"
```

Executam-se as scripts SQL para criar as tabelas da base de dados do zabbix:

```
# mysql -u zabbix -p zabbix < create/schema/mysql.sql
# mysql -u zabbix -p zabbix < create/data/data.sql
# mysql -u zabbix -p zabbix < create/data/images_mysql.sql
```

Finalmente procede-se à instalação do Zabbix:

```
# make install
```

ANEXO III

Ajustes no servidor pós-instalação do Zabbix

Foi necessário fazer a edição do ficheiro /etc/services, para isso executou-se o seguinte código:

```
# nano /etc/services
```

No final do ficheiro foram adicionadas as seguintes linhas:

```
zabbix_agent 10050/tcp # Zabbix Agent
zabbix_agent 10051/tcp # Zabbix Server
```

De seguida criei o diretório /etc/zabbix e foram copiados os ficheiros de configuração para lá:

```
 # mkdir /etc/zabbix
# cp misc/conf/zabbix_server.conf misc/conf/zabbix_agent*
/etc/zabbix/
```

Foram então criados as pastas para o armazenamento dos ficheiros de log e pid:

```
# mkdir/var/log/zabbix
# mkdir /var/run/zabbix
```

De seguida foram alteradas as permissões dos diretórios /var/log/zabbix e /var/run/zabbix, com o comando:

```
# chown zabbix: /var/log/zabbix /var/run/zabbix
```

Criou-se a pasta /var/tmp/zabbix e deram-se as respetivas permissões para o utilizador zabbix:

```
# mkdir/var/tmp/zabbix
# cd/var/tmp/
# chown –R zabbix.zabbix zabbix
```

ANEXO IV

Configuração do servidor Zabbix

Editou-se o ficheiro /etc/zabbix/zabbix_server.conf, utilizando o "nano" para esse efeito:

```
# nano /etc/zabbix/zabbix_server.conf
```

Alteraram-se ou descomentaram-se as linhas abaixo e configuraram-se de acordo com as necessidades da nossa infraestrutura.

```
# Define um Node ID como sendo unico
### Option: NodeID
#       Unique NodeID in distributed setup.
#       0 - standalone server
NodeID=0

# Frequencia de envio de alertas
### Option: SenderFrequency
#       How often Zabbix will try to send unsent alerts (in seconds).
SenderFrequency=30

# Nível do debug no Log File
### Option: DebugLevel
#       Specifies debug level
#       0 - no debug
#       1 - critical information
#       2 - error information
#       3 - warnings
#       4 - for debugging (produces lots of information)
DebugLevel=3
```

```
# Timeout de conexão com o agente

### Option: Timeout

#       Specifies how long we wait for agent, SNMP device or external check (in seconds).

Timeout=5


# Caminho do arquivo pid do Zabbix Server

PidFile=/var/tmp/zabbix/zabbix_server.pid


# Caminho do arquivo de log do Zabbix Server

LogFile=/var/log/zabbix/zabbix_server.log


# Tamanho do arquivo de log

LogFileSize=2

memory_limit = 512M


# Caminho dos scripts personalizados

AlertScriptsPath=home/zabbix/alertscripts


# Servidor MySQL DBHost=localhost


# Nome da base de dados no MySQL DBName=zabbix


# Utilizador da base de dados no MySQL DBUser=zabbix


# Senha do utilizador Zabbix no MySQL DBPassword=<password>
```

Configurou-se o serviço zabbix-server para inicializar automaticamente no sistema.

```
# cp misc/init.d/debian/zabbix-server /etc/init.d/
```

```
# chmod a+x /etc/init.d/zabbix-server
```

Através do comando rcconf habilitamos o zabbix-server:

```
# rcconf
```

De seguida, editou-se o ficheiro /etc/init.d/zabbix-server:

```
# nano /etc/init.d/zabbix-server
```

Editaram-se as linhas do ficheiro de acordo com o que se mostra a seguir:

```
NAME=zabbix_server DAEMON=/usr/local/zabbix/sbin/$NAME DESC=”Zabbix server daemon” PID=/var/run/zabbix/$NAME.pid

PATH=/bin:/usr/bin:/sbin:/usr/sbin:/usr/local/zabbix/sbin:/usr/

local/zabbix/bin
```

Finalmente, iniciou-se o servidor Zabbix:

```
# /etc/init.d/zabbix-server start
```

(se o arranque tiver algum problema e falhar, analisar os motivos para a falha nos ficheiros de logs, em /var/log/zabbix)

ANEXO V

Configuração do *front-end* do Servidor Zabbix

Inicialmente editou-se o ficheiro `/etc/php5/apache2/php.ini`:

```
# nano /etc/php5/apache2/php.ini
```

E alteraram-se as seguintes linhas:

```
date.timezone = "Europe Lisbon"
max_execution_time = 300
max_input_time = 300
memory_limit = 512M
post_max_size = 32M
upload_max_filesize = 16M
max_execution_time = 600
```

Reiniciou-se por fim o Apache para atualizar as novas configurações, PHP:

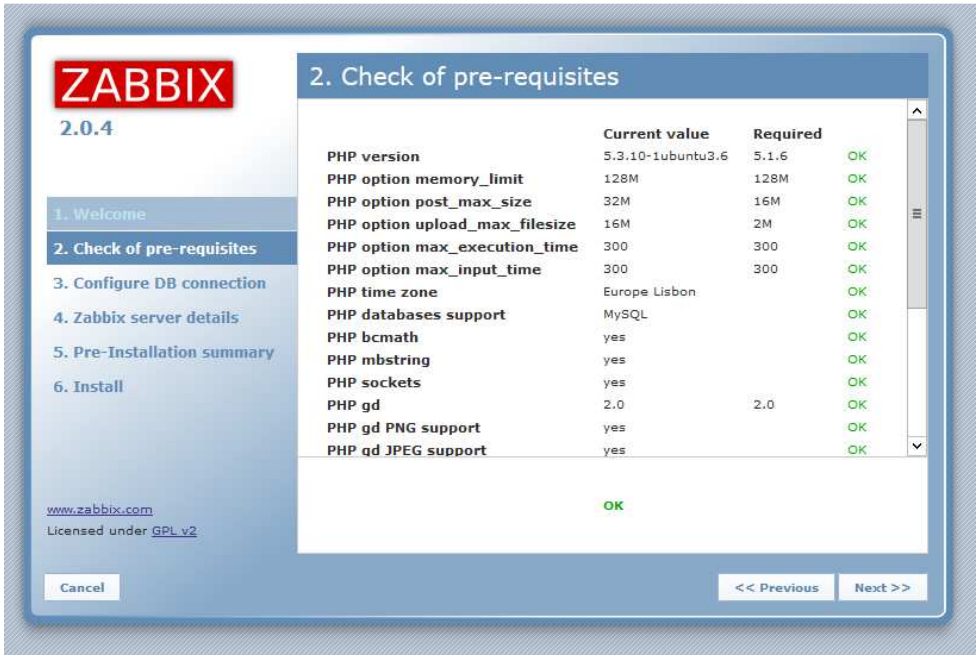
```
# /etc/init.d/apache2 restart
```

Para se poder ter acesso ao *front-end* do Zabbix, copiou-se o código fonte do PHP do front-end para a pasta /var/www/zabbix:

```
# cp -r /usr/src/zabbix-1.8.8/frontends/php//var/www/zabbix
# chown -R www-data:zabbix /var/www/zabbix
```

Acedeu-se ao *front-end*, no endereço http://<endereco_servidor>/zabbix/ onde podemos aceder ao interface de configuração do Zabbix com uma mensagem de boas vindas do Zabbix será exibida:

Na imagem seguinte surge o resumo de uma verificação dos pré-requisitos.



De seguida foram configurados os parâmetros para fazer a ligação à base de dados.

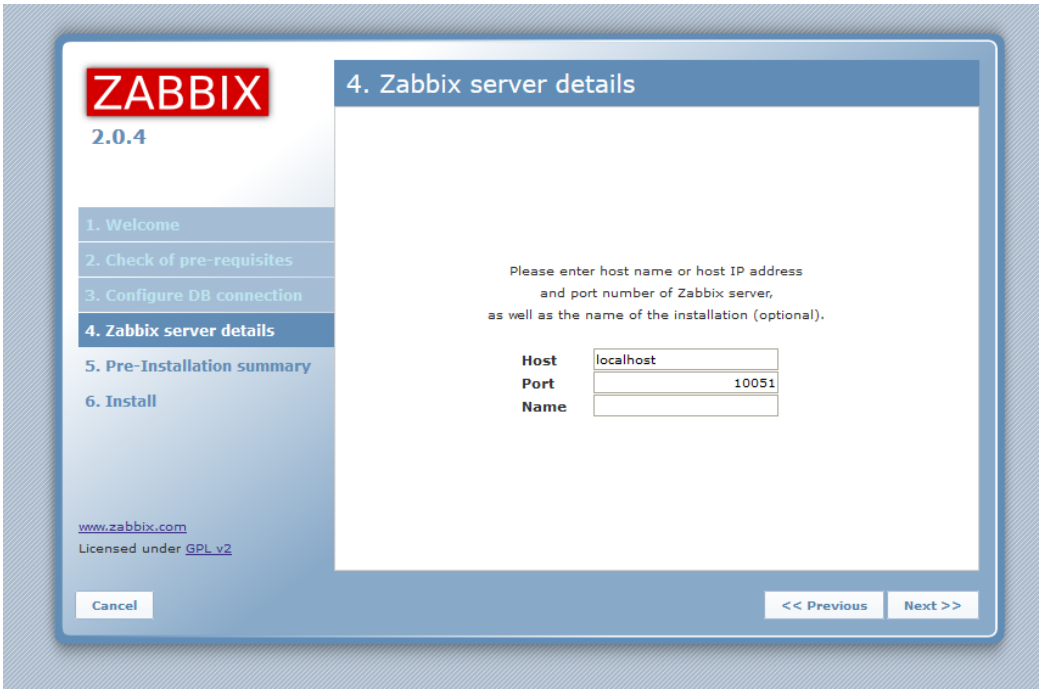
Procedeu-se então à verificação dos teste de conexão à base de dados, e só após as configurações serem dadas como válidas é que foi possível avançar ao passo seguinte.

Na opção *Type*, selecciona-se “MySQL”. Na opção *Host*, escreve-se “localhost”. Na opção *Port*, deixa-se a padrão “0”.

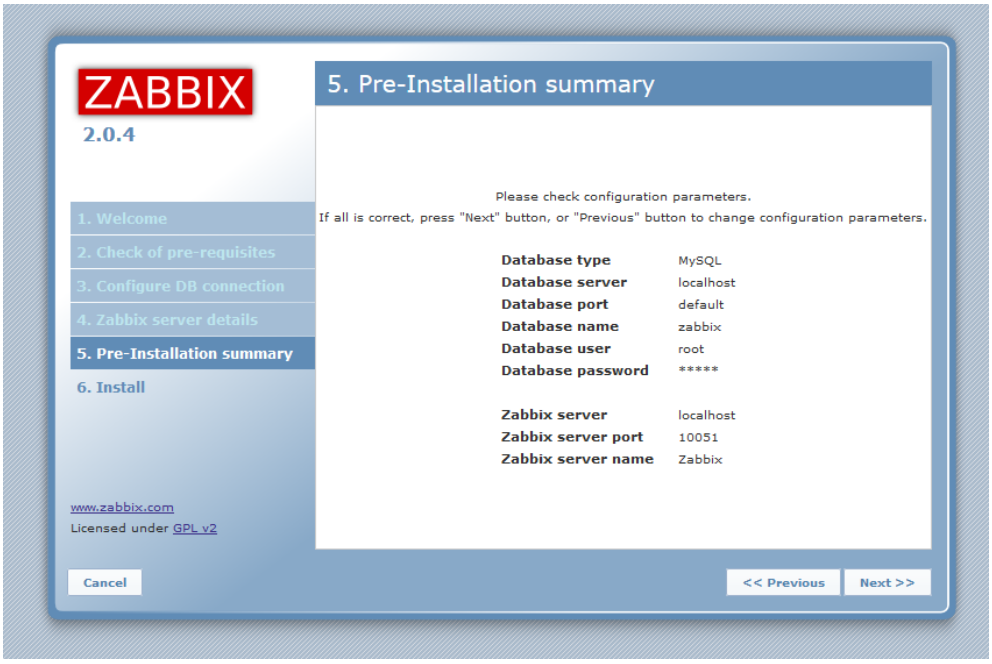
Na opção *Name*, insere-se o nome da base de dados no MySQL (“zabbix”). Na opção *User*, dá-se o nome da conta com permissão de acesso (“root”) Na opção *Password*, colocamos a password da conta anterior.

Depois, testa-se a conexão (botão *Test Connection*) para validar as configurações (se tudo estiver bem, aparecerá um OK a verde acima do botão *Test Connection*).

De seguida (Ilustração 20 e Ilustração 21) confirma a correcção de configurações efectuadas antes na consola.

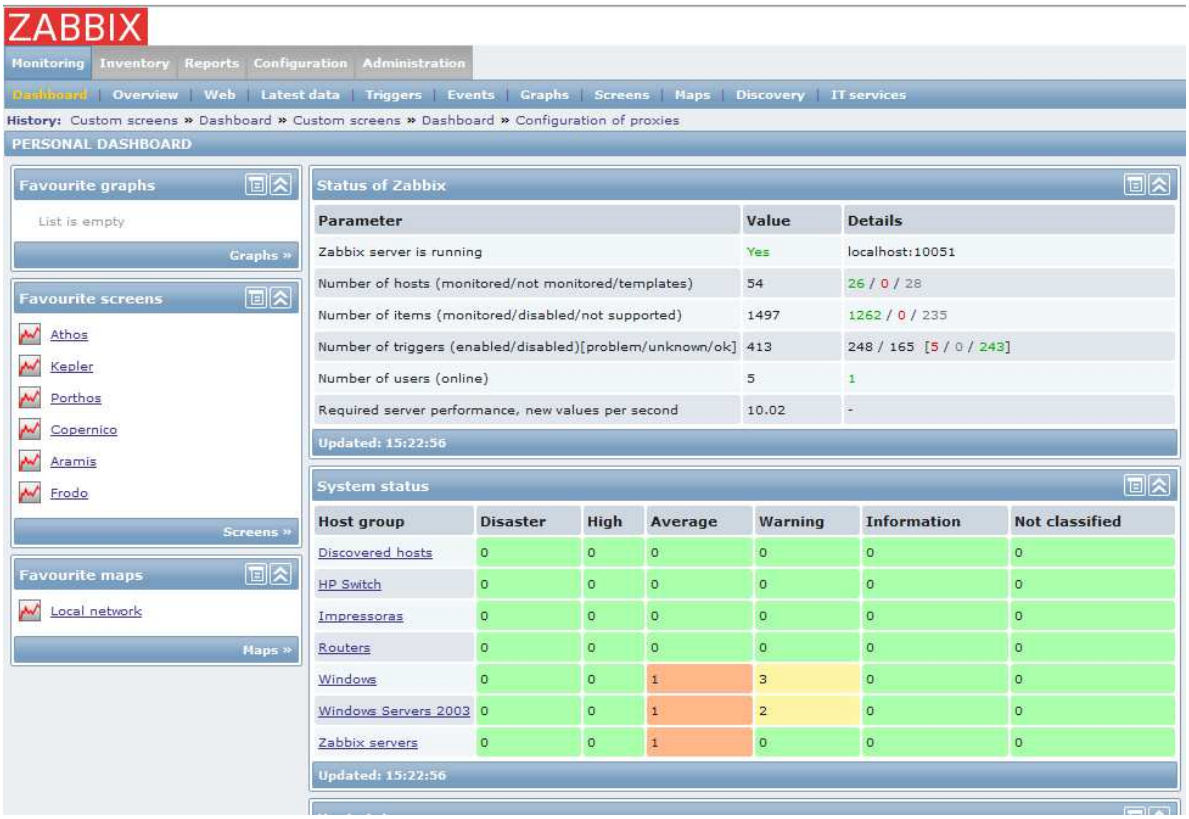


De seguida foi feita a confirmação dos parâmetros de configuração:

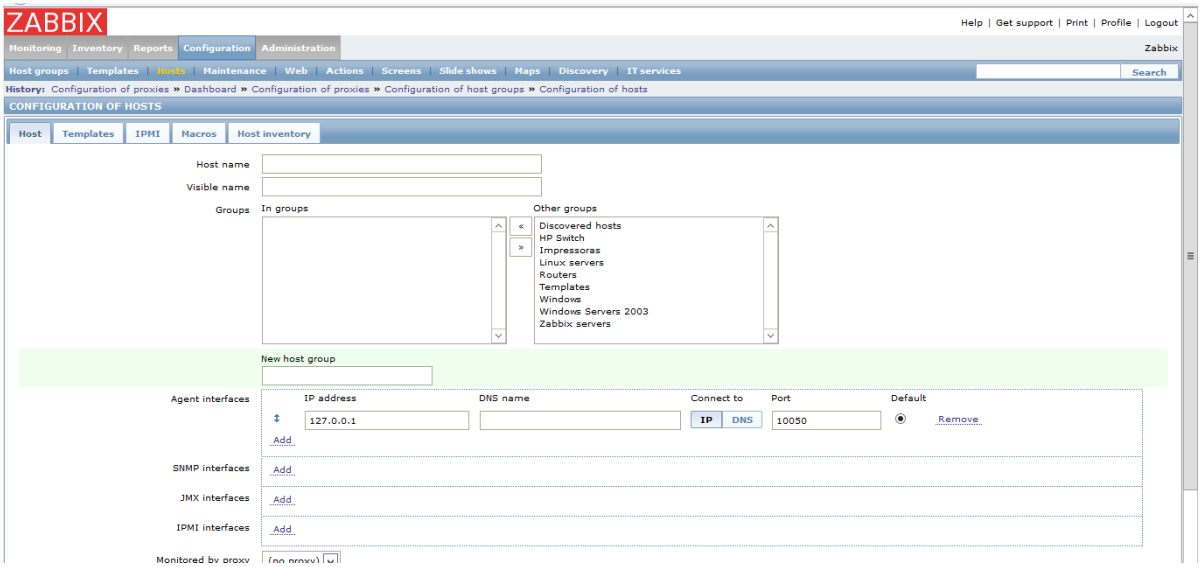


Temos então a nossa configuração efectuada, e para a concluir apenas foi necessário escolher a opção "Finish".

A partir de agora possível aceder ao *front-end* com as credenciais por omissão (“admin” + “zabbix”).



Depois de instalados os agentes nos *host*s a monitorizar (ver anexo V), foi necessário adicionar esses *hosts* no Zabbix. Para isso acede-se ao separador *Configuration*, seguido do separador *Hosts* e de seguida carrega-se no botão *Create host*, o que fará aparecer a janela



ANEXO VI

Instalação do agente Zabbix em máquinas Windows.

Download dos ficheiros necessários

O download do agente deve ser feito no site do Zabbix, ou em alternativa, copiado do diretório que contém os fontes de instalação do Zabbix Server, neste caso os ficheiros encontram-se em zabbix-versão/bin e dentro deste diretório temos 2 subdiretórios (win32 e win64) respetivos à arquitetura do processador. A cópia dos ficheiros do Linux para o Windows pode ser feita através do software "WinSCP" ou do "Secure File Client Transfer".

Instalação:

Copiar ou descompactar os arquivos, dependendo da sua escolha no passo anterior, na raíz do C:\, renomear a pasta para Zabbix.

Copiar ou criar o arquivo zabbix_agentd.conf dentro da pasta C:\Zabbix, um arquivo de exemplo pode ser encontrado no diretório que contém os fontes de instalação do Zabbix Server, no diretório zabbix-versão/misc/conf.

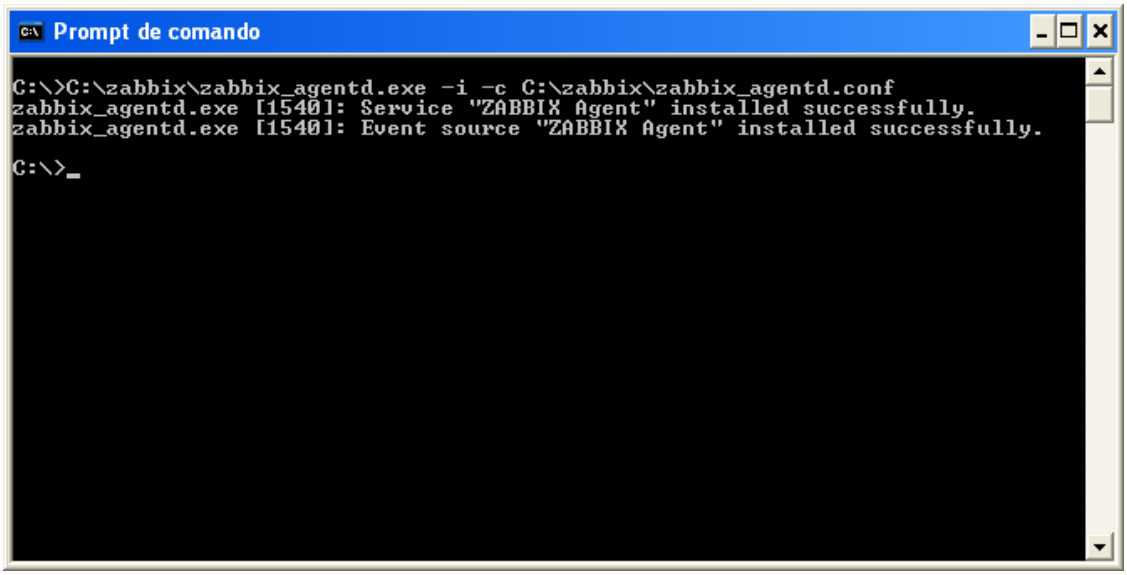
Conteúdo mínimo do arquivo zabbix_agentd.conf:

Server=IP do Servidor do Zabbix
Hostname=Nome da máquina Cliente
StartAgents=5
DebugLevel=3
LogFile=C:\Zabbix\zabbix_agentd.log
Timeout=3

Criando um serviço ZABBIX Agent no Windows:

Abrir um prompt (Linha de comandos) de comando e executar o seguinte comando:

C:\Zabbix\zabbix_agentd.exe -i -c C:\Zabbix\zabbix_agentd.conf



Após a instalação do agente devem aparecer as mensagens:

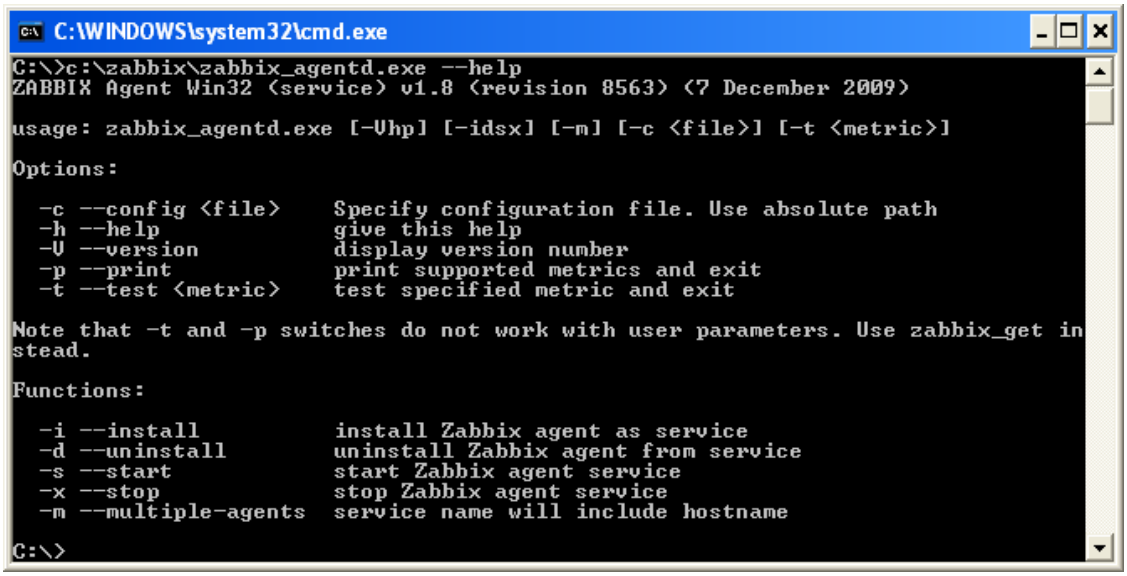
zabbix_agentd.exe [1540]: Service "ZABBIX Agent" installed successfully.

zabbix_agentd.exe [1540]: Event source "ZABBIX Agent" installed successfully.

Estas são a confirmação de que a instalação foi efetuada com sucesso.

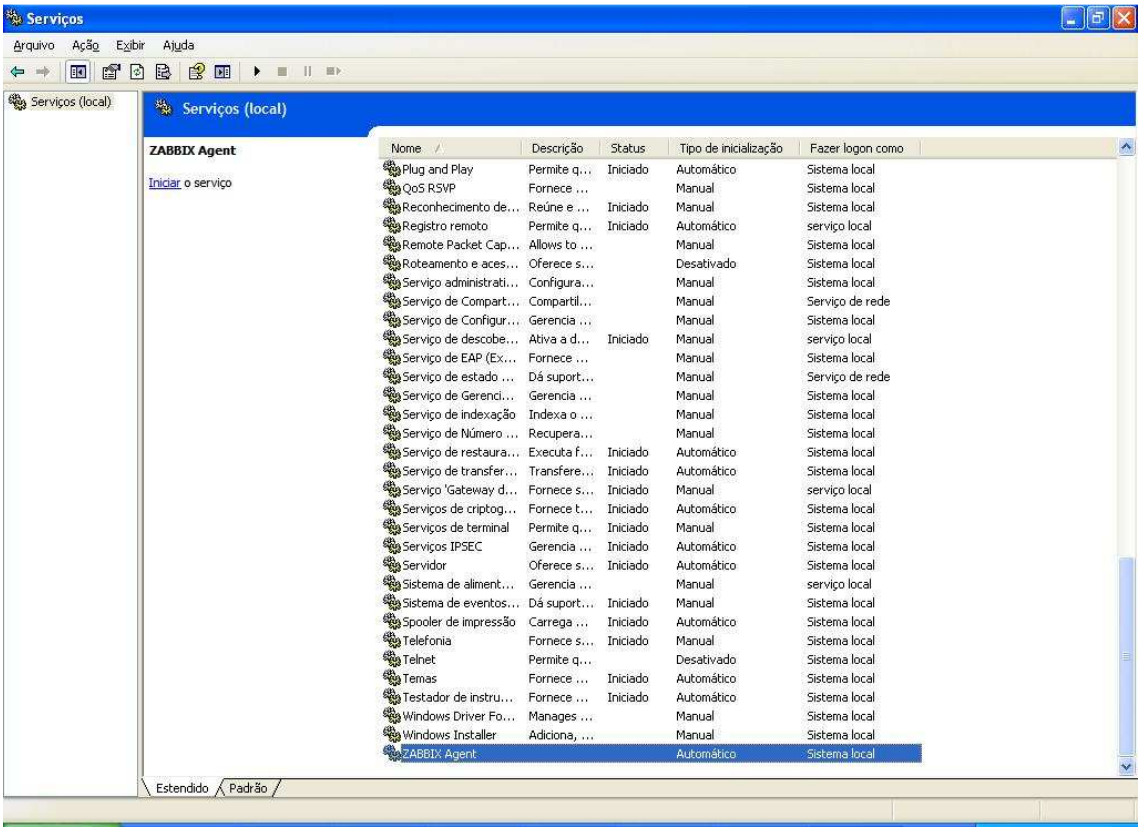
Pode-se visualizar as demais opções do agente executando o comando:

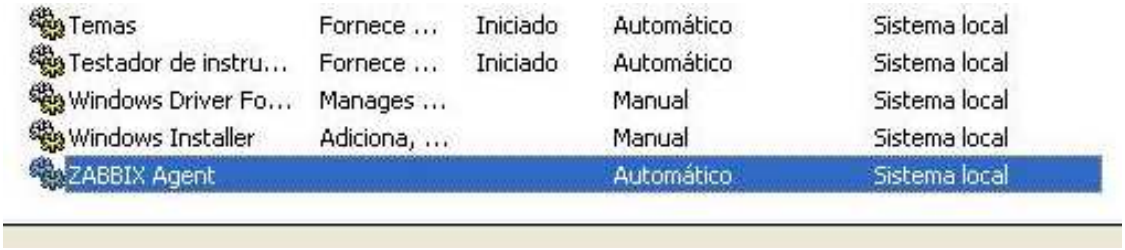
C:\Zabbix\zabbix_agentd.exe --help



Verificar o estado do serviço do Zabbix:

Iniciar // Painel de Controle // Ferramentas administrativas // Serviços:





Verificar as propriedades do serviço:

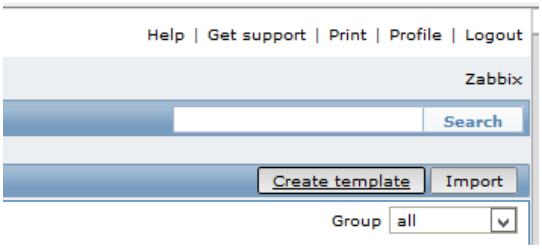
Duplo clique em "ZABBIX Agent", e confirma se a inicialização está em automático.


Propriedades ZABBIX Agent (Computador local)
Geral
Logon
Recuperação
Dependências
Nome do serviço: ZABBIX Agent
Nome para exibição:
ZABBIX Agent
Descrição:
Caminho do executável:
"C:\zabbix\zabbix_agentd.exe" --config "C:\zabbix_agentd.conf"
Tipo de inicialização:
Automático
Status do serviço: Parado
Iniciar
Parar
Pausar
Continuar
Você pode especificar os parâmetros de inicialização aplicáveis quando o serviço é iniciado aqui.
Parâmetros de inicialização:
OK
Cancelar
Aplicar

ANEXO VII

Criação de *Templates*

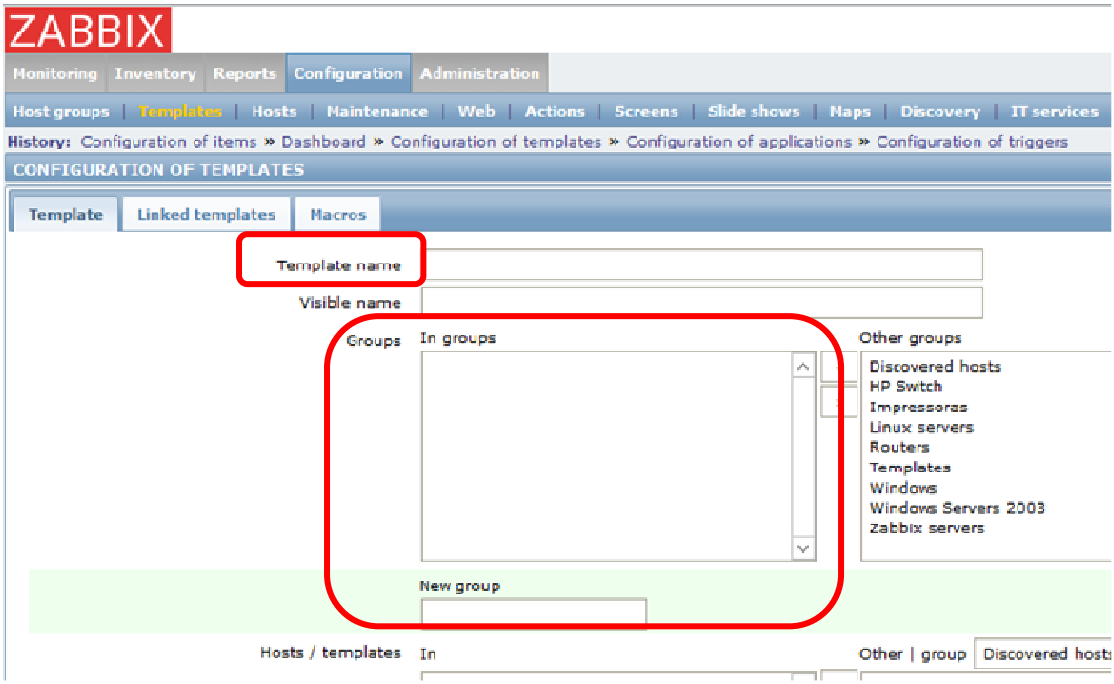
A criação de novos *templates* no Zabbix, é um processo simples, já que quando estamos na gestão de *templates*, existe um botão "Create template", que irá abrir um formulário para esse efeito.



O formulário que possibilita a definição básica do *template*. No separador **Template** é necessário o preenchimento de, no mínimo dois campos:

Template Name – Identificador único do *template*. O Zabbix também não permite que exista um *template* ou um *host* com o mesmo nome.

Groups – Grupo ou grupos ao qual o *template* é atribuído. Pode-se selecionar um dos grupos já existentes a partir de uma lista ou criar um novo. Para criar novo grupo de *templates* é necessário que o utilizador tenha o perfil de super administrador do sistema.



O separador **"Linked Templates"** tem a finalidade de permitir associações entre *templates*. Este recurso adiciona ao Zabbix o conceito de herança.

Ao associar um *template* a outro, herda-se automaticamente qualquer definição que exista no *template* **"pai"** e não é permitida a eliminação de definições, apenas é possível a desativação e alguns ajustamentos nas configurações.

O separador "**Macros**" acrescenta outro recurso muito poderoso ao Zabbix que na maioria dos sistemas é pouco explorado.

As macros podem aparecer em três níveis:

Nível de sistema – Acessíveis em: "**Administration – General – Macros**";

Nível de *template* – Acessível através do separador **Macros** do *template*;

Nível de *host* - Acessível através do separador **Macros** do *host*;

É possível a utilização das macros nos *ítens* e nos *triggers*. A utilização consciente e bem planeada deste recurso pode reduzir significativamente o número de *templates* e a dificuldade de gestão destes.

Terminada a definição das características do *template* resta-nos gravá-las, para que sejam validadas. Após salvar o registro o Zabbix irá retornar para a tela com a lista de *templates*.