INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




# Wonderbox

**STEFAN RODRIGUES GOMES**
Outubro de 2016

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

# Wonderbox

**Stefan Rodrigues Gomes**

**Dissertação para obtenção do Grau de Mestre em Engenharia Informática**

**Orientador: Professor Eng.º Jorge Manuel Canelhas Pinto Leite**

Porto, outubro 2016

*À minha Família, Namorada e Amigos*

# Resumo

Os Sistemas de Informação têm por base assegurar o processamento e distribuição de informação numa organização pelo que também se deve ter em consideração a qualidade com que essa informação chega ao destino.

A competitividade empresarial hoje em dia leva a que a informatização de processos corporativos em tecnologia seja cada vez mais uma necessidade. Desta forma exige-se que a informação esteja disponível para quando o seu acesso for solicitado dentro dos parâmetros de serviço previstos.

A garantia da disponibilidade da informação leva indiretamente à necessidade de se efetuar periodicamente cópias de segurança. A informação é delicada e poderá ter uma importância extrema ao ponto da sua ausência poder levar a grandes prejuízos financeiros.

Assim, pretendeu-se no trabalho realizado no âmbito desta dissertação de mestrado efetuar um estudo sobre os problemas inerentes às organizações, encontrando mecanismos para automatizar processos com vista a evitar intervenções e minimizar erros e esquecimentos, simplificando desta forma a vida aos administradores de sistemas. Equitativamente pretendeu-se efetuar uma investigação sobre ferramentas de mercado que permitam atingir tais objetivos.

Para tal foi implementado um sistema de análise à disponibilidade e à salvaguarda da informação dos utilizadores através de cópias de segurança num ambiente de simulação das realidades empresariais atuais.

Os resultados apurados foram satisfatórios, sendo possível verificar a relevância do produto numa prova de conceito.

**Palavras-chave**: Disponibilidade da informação, monitorização, cópias de segurança.

# Abstract

Information Systems are based on assure processing and distribution of information in an organization. It also should consider the information quality that reaches the destination.

Corporative competitiveness nowadays leads organizations to the computerization of processes into technology. This will be increasingly a necessity. So, it is demanded that information must be available when their access is requested.

The information availability guarantee takes indirectly to the necessity of performing regularly backups. The data is so sensitive that have an extreme importance, their absence could lead to huge financial losses.

So, in this master degree dissertation is intended to study the problems inherent to organizations, finding mechanisms to automate processes in order to avoid interventions, minimizing errors and omissions, thereby simplifying work for system administrators. Equally it is intended to conduct an investigation into marketing tools that achieve the goals of this report.

To do that, was implemented a system that analyzes information availability and protect it through backups in a simulation environment.

The results obtained were satisfactory, it is possible to verify the relevance of the product on a concept proof.

**Keywords**: Information availability, monitoring, backups.

# Agradecimentos

A execução desta dissertação só foi possível com a colaboração de várias pessoas às quais gostaria de expressar a minha gratidão.

Desde já aos meus pais, irmãos e namorada que me apoiarem desde o primeiro dia na elaboração desta dissertação, fazendo com que eu não desistisse da mesma.

À Visionware Sistemas de Informação pela possibilidade de elaboração e desenvolvimento do projeto dentro das suas instalações.

Aos meus colegas de trabalho e amigos Pedro Rodrigues e Bruno Lourenço que me apoiarem a nível técnico e juntos debatemos sobre estes temas.

Por fim também quero agradecer ao docente Eng.º Jorge Pinto Leite, pelo seu apoio, prontidão para esclarecimento de dúvidas e ajuda na construção deste documento.

# Índice

# Lista de Figuras

# Lista de Tabelas

# Acrónimos e Símbolos

## Lista de Acrónimos

| | |
|---|---|
| **AMANDA** | *Advanced Maryland Automatic Network Disk Archiver* |
| **BAREOS** | *Backup Archiving Recovery Open Source* |
| **BCP** | *Business Continuity Plan* |
| **BIA** | *Business Impact Analysis* |
| **CEO** | *Chief Executive Officer* |
| **CN** | *Common Name* |
| **DNS** | *Domain Name System* |
| **DMZ** | *Demilitarized Zone* |
| **DRP** | *Disaster Recovery Plan* |
| **EOS** | *European Organization for Security* |
| **FQDN** | *Fully Qualified Domain Name* |
| **GNS** | Gabinete Nacional de Segurança |
| **ICMP** | *Internet Control Message Protocol* |
| **IP** | *Internet Protocol* |
| **IT** | *Information Technology* |
| **LAN** | *Local Area Network* |
| **PKI** | *Public Key Infrastructure* |
| **PPP** | *Point-to-Point Protocol* |
| **RA** | *Risk Analysis* |
| **RPO** | *Recovery Point Objective* |
| **RTO** | *Recovery Time Objective* |
| **RAID** | *Redundant Array of Independent Disks* |
| **SMIG** | *Security Mission Industry* |

**SMTP** *Simple Mail Transfer Protocol*

**SNMP** *Simple Network Management Protocol*

**TCP** *Transmission Control Protocol*

**VSS** *Volume Snapshot Service*

# 1 Introdução

Neste capítulo irá ser feito o enquadramento da presente dissertação assim como uma apresentação da tese e da organização onde foi desenvolvida. Serão também abordados os contributos deste trabalho e a estrutura do relatório.

## 1.1 Enquadramento

Não existem dúvidas que atualmente as Tecnologias e Sistemas de Informação são um valor inquestionável na garantia e sustentabilidade e continuidade dos negócios das organizações (Silva P., 2015). Estas precisam de se manter atualizadas e enquadradas com a dinâmica e volatilidade deste mundo de inovação tecnológica.

As necessidades e exigências associadas à nova realidade económica e empresarial implicam que as plataformas de Tecnologias e Sistemas de Informação garantam índices de qualidade e estabilidade extremamente elevados. Independentemente da dimensão ou áreas de negócio das empresas, existem fatores comuns e fundamentais para o sucesso de qualquer organização que sustente o seu "*core business*" sobre Sistemas de Informação: os níveis de disponibilidade e integridade do seu principal ativo, a informação (Brochura VW, 2015).

Desta forma é necessário encontrar mecanismos para automatizar processos com vista a evitar intervenções, minimizando erros e esquecimentos, simplificando desta forma a vida aos administradores de sistemas.

A ausência na prevenção da perda de dados por parte de entidades é cada vez mais uma realidade, tornando-se importante criar um sistema de *backup* que permita à entidade salvaguardar cópias de segurança nos *endpoints*. Um *backup* local é inquestionável para a reposição rápida em caso de falha, mas de nada vale se houver um acidente natural que destrua a instalação. Contudo, se existir concomitantemente o local e um remoto, cumpre os requisitos mínimos de um plano de continuidade de negócio. As novas tecnologias de comunicação podem e devem ser utilizadas para otimizar os aspetos mencionados.

A monitorização constante da rede para uma deteção automática de anomalias é uma outra realidade. Há pelo menos duas razões para tal, descobrir problemas e identificar soluções.

A tecnologia hoje em dia é a espinha dorsal dos negócios (Kokemuller N., 2015). Em caso de falha pode levar uma entidade a grandes prejuízos. É necessário garantir a disponibilidade dos serviços críticos da empresa. Um serviço quando crítico jamais poderá estar indisponível. Ocorrendo um imprevisto, deverá existir um registo desse evento alertando imediatamente os administradores sobre a falha.

No caso da inexistência de uma monitorização e em caso de falha de sistema a primeira dúvida de um administrador de sistemas é: porque parou? O que está a acontecer?

Com uma monitorização adequada o administrador de sistemas saberá responder a estas questões minimizando desta forma o tempo da indisponibilidade dos serviços (Cirilo M. Luciano, 2010; Galstad E., 2015).

Aliando o gosto por este tipo de projetos às necessidades da Visionware surgiu esta oportunidade de elaborar esta dissertação de mestrado nas instalações da Visionware.

## 1.2 Apresentação da tese

Esta dissertação tem como seu principal objetivo o desenvolvimento de um produto que permita à Visionware cimentar o seu lugar como empresa de referência ao nível de sistemas no mercado nacional.

O produto tem por base um conjunto de ferramentas que permite a monitorização constante do parque informático e cópias de segurança aos utilizadores finais.

É expectável e desejável a introdução deste produto em parques informáticos para monitorização e cópias de segurança a informação de utilizadores finais.

## 1.3 Apresentação da organização

A Visionware Sistemas de Informação, sediada no Centro empresarial da Lionesa em Leça do Bálio, Matosinhos, define-se como um parceiro de excelência no mercado nacional e internacional das Tecnologias e Sistemas de Informação, disponibilizando para tal, serviços abrangentes e altamente especializados nas mais variadas vertentes tecnológicas e/ou aplicacionais.

Possuí uma equipa de consultores com elevado *know-how*, estabelecendo uma postura contínua na orientação para as necessidades e exigências dos seus clientes, com base em normas internacionais de referência. Apresenta certificações de alto-relevo nas diversas áreas em que atua, autenticando a experiência e conhecimento adquirido ao longo do ano.

A Visionware foi a primeira empresa a ser certificada pelo GNS – Gabinete Nacional de Segurança. A credenciação NATO SECRET permite à empresa participar em projetos internacionais de referência. Participa ativamente nos grupos europeus da área de segurança de informação, nomeadamente o ASD – *Aero Space and Defence*, EOS – *European Organization for Security* e SMIG – *Security Mission Industry Groups*.

A sua abordagem é orientada numa visão estratégica e integrada, em ordem à garantia da continuidade de negócio, com uma postura independente face a tecnologias ou sistemas de informação. A sua lógica de atuação prende-se com uma gestão criteriosa e minimizadora dos riscos associados aos investimentos em sistemas de informação nas organizações.

A gestão de serviços é baseada em processos. Estes não devem ser vistos de forma coordenada e orientada para o mesmo objetivo. O modelo de processos assenta numa abordagem cíclica "planear-executar-verificar-agir".

A Visionware tem 4 áreas de especificação entre as quais:

- Visionware *Security* – descoberta de falhas de segurança em aplicações de negócio, sistemas internos ou plataformas Web, até à identificação das necessárias ações de melhoria dos níveis de segurança praticados, a auditoria dá uma visão externa, não comprometida e totalmente focada na minimização do risco para o negócio.
  As auditorias são efetuadas com as melhores práticas internacionais (ISO 27001).
- Visionware *Integration*- Esta área de Integração está vocacionada para a conceção e arquitetura de soluções tecnológicas flexíveis e adequadas ao seu modelo de negócio.
- Visionware *Research & Development.*
- Visionware IT *Management*- A área de IT *Management* está vocacionada para a gestão e manutenção de plataformas de servidores físicos e virtuais, redes, comunicações e *desktops*, até às soluções de segurança de perímetro e corporativas, assente em aplicações de monitorização das infraestruturas tecnológicas.

Em termos de estrutura apresenta a seguinte estrutura de três níveis hierárquicos.

O primeiro nível é constituído pelo CEO – *Chief Executive Officer*. O segundo nível hierárquico corresponde aos diretores das diferentes áreas, diretor comercial, diretor de *consulting*, diretor IT – *Information Tecnology* e diretor financeiro. O terceiro e último nível hierárquico corresponde aos funcionários.

O organograma seguinte embutido no manual de acolhimento 2015 VW, ilustra os níveis hierárquicos existentes na Visionware Sistemas de Informação S.A.

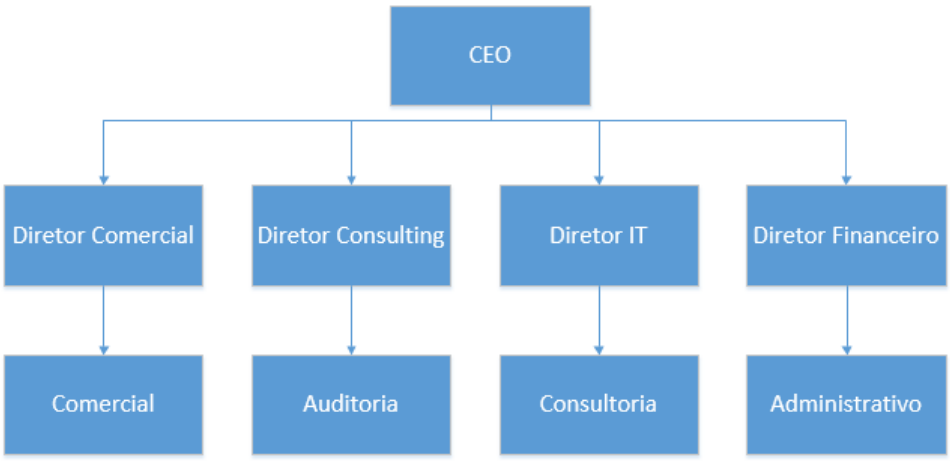



Figura 1 - Organograma da Visionware

## 1.4 Contributos deste trabalho

Apesar de ser discreto, este produto trará vantagens significativas tanto para os utilizadores finais como para os administradores de sistemas, pois permitirá um controlo da rede por parte dos *sysadmins*, diminuindo os tempos de indisponibilidade de serviços ou ativos. Aumentará também com certeza a probabilidade de reposição de dados aos utilizadores, reduzindo o RPO – *Recovery Point Objective* e o RTO – *Recovery Time Objective*.

Assim, podemos considerar a *Wonderbox* uma ferramenta que controla e vigia as necessidades para a continuidade de negócio por parte das empresas.

## 1.5 Estrutura da dissertação

Esta dissertação é constituída por 5 capítulos.

Neste primeiro capítulo, a Introdução, é feito um enquadramento sobre o tema deste trabalho, é feito uma apresentação da tese e da empresa Visionware. É igualmente abordado os contributos que se pretendem alcançar com esta dissertação.

No segundo capítulo, o Estado da Arte, é analisado o contexto e o problema desta dissertação. São igualmente estudadas tecnologias para a resolução deste problema.

No capítulo seguinte, a Descrição Técnica, pretende-se dar ênfase à arquitetura do produto, o que é utilizado e como é efetuado a abordagem à resolução do problema.

No quarto capítulo, a Avaliação do Produto, pretendemos avaliar a solução.

No quinto capítulo são apresentadas as Conclusões, descritas as principais limitações deste trabalho e apresentadas direções para trabalho futuro.

# 2 Estado da Arte

A monitorização afirma-se hoje em dia como uma vital importância para as empresas e/ou organizações (Alves A., 2012). O administrador de sistemas deve acompanhar em tempo real o estado dos equipamentos críticos como por exemplo espaço em disco, carga de CPU e largura de banda utilizada entre outros. Atualmente a monitorização não considera apenas o estado dos equipamentos, mas também os serviços.

O administrador de sistemas sabe que qualquer equipamento na infraestrutura seja ele recente ou não, não está imune a erros. Por isso, qualquer sistema crítico para o negócio deve ser alvo de uma constante monitorização a fim de evitar interrupções que prejudiquem o manuseio normal dos equipamentos informáticos por parte dos utilizadores finais. Quando uma entidade tem à sua mercê uma monitorização alarmística (NagiosCore), os administradores de sistemas responsáveis serão informados sobre possíveis falhas no sistema. Isso faz com que estes possam atuar de forma mais ágil, caso algum problema ocorra.

A monitorização torna a correção dos problemas mais rápida uma vez que já se sabe qual o foco do problema, não sendo necessária a sua identificação e reduzindo desta forma o tempo necessário para o seu despiste.

O controlo sobre os serviços e servidores é imprescindível pois em algum sistema poderão estar alocadas cópias de segurança necessárias para restauro em caso de falha. Desta forma, para além da monitorização torna-se imprescindível efetuar cópias de segurança ao sistema, pois é a única forma de recuperar dados (Duffy J., 2012, Sophos 2015).

As falhas são inevitáveis, mas o impacto das falhas, ou seja, o colapso do sistema e consequente perda de dados, pode ser evitado usando técnicas viáveis para o efeito.

Um estudo efetuado pela KrollOntrack indica-nos que a perda de informação se deve essencialmente a falhas de *hardware* como se pode ver na figura seguinte. A informação foi retirada com base em dados estatísticos efetuados pela KrollOntrack (KrollOntrack, 2015).

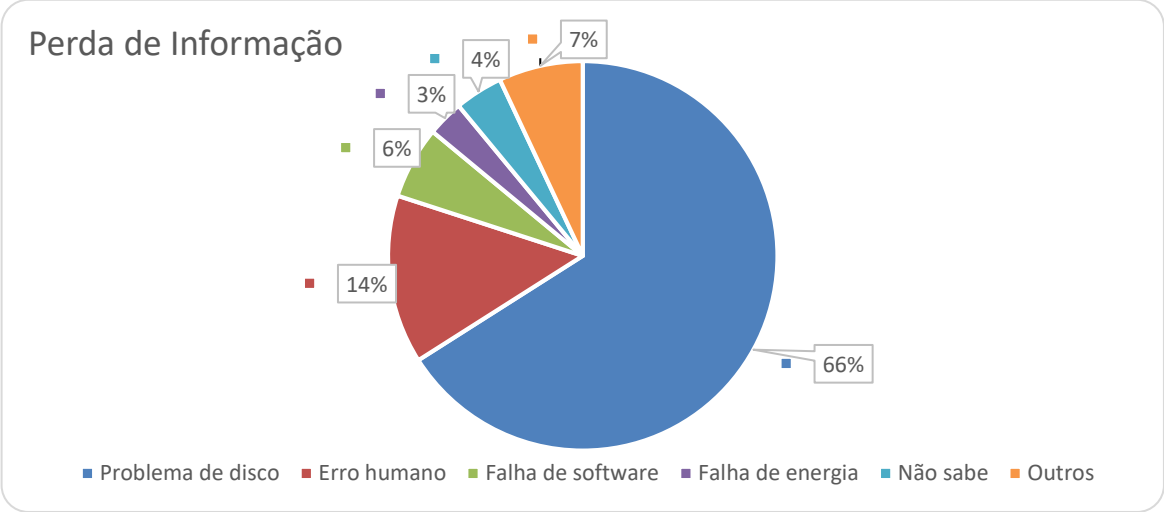



Figura 2 - Causas da perda de informação
(KrollOntrack, 2015, adaptado)

Segundo este estudo os problemas de disco estão em 66% dos casos na perda da informação dos utilizadores, sendo que essa perda acontece em 72% dos casos nos discos rígidos, 15% nos dispositivos móveis e 13% em discos com *RAID – Redundant Array of Independent Disks*, associado (KrollOntrack, 2015). Estas falhas tem um impacto de 42% em dados pessoais, 38% em informação de negócio e 20% em informação crítica de negócio (KrollOntrack, 2015).

Desta forma, a criação de um BCP - *Business Continuity Plan* – para estabelecer um conjunto de estratégias preventivas e planos de ação de maneira a garantir que os serviços essenciais sejam devidamente identificados e preservados após a ocorrência de uma falha, é essencial (António M. Adriano, 2014). Portanto, o objetivo é possibilitar o funcionamento da empresa a um nível aceitável nas situações de contingência onde há indisponibilidade dos recursos de informação - mitigar os efeitos de perdas, avarias e falhas críticas.

Aqui prendem-se outras questões nomeadamente a periodicidade, é necessário saber ajustar a periodicidade de forma a garantir os respetivos tempos de retenção. O que isto significa, é que não adianta efetuar uma cópia todos os dias se a retenção está mal ajustada. Vamos imaginar que o utilizador final pensa que pode recuperar ficheiros até 4 semanas e a retenção está ajustada a 2 semanas, só é possível recuperar ficheiros nas últimas 2 semanas. Existem empresas que necessitam por questões de qualidade e imposição da ISO 9001 a retenção da base de dados de produção durante 5 anos.

O local da cópia de segurança é vital em caso de desastre, não adianta que a cópia esteja no mesmo edifício que as máquinas, caso por exemplo haja um incêndio ou uma inundação, mas então é necessário ter em atenção as ameaças existentes hoje em dia relativo à interceção de informação.

Segundo Ícaro Mattes, a informação é o bem mais precioso (Mattes V. Ícaro et al., 2014). Cada vez mais existem relatos de ataques. Uma comparação ao número de ataques identificados

em 2014 e 2015 efetuada em 2016 (Passeri P., 2016), como podemos observar na figura 3, evidencia um aumento não desprezável.

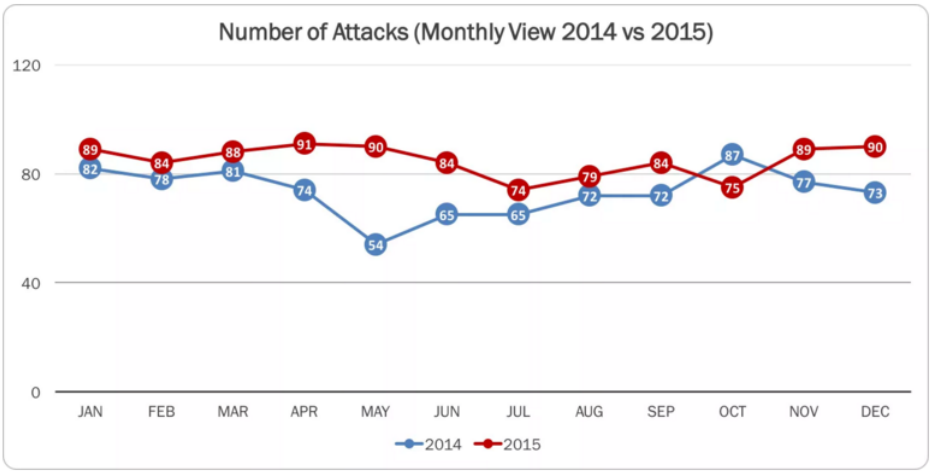



Figura 3 - Número de ataques
(Passeri P., 2016)

Mais preocupante é apurarmos que a principal motivação por de trás destes ataques é o *cyber crime*.

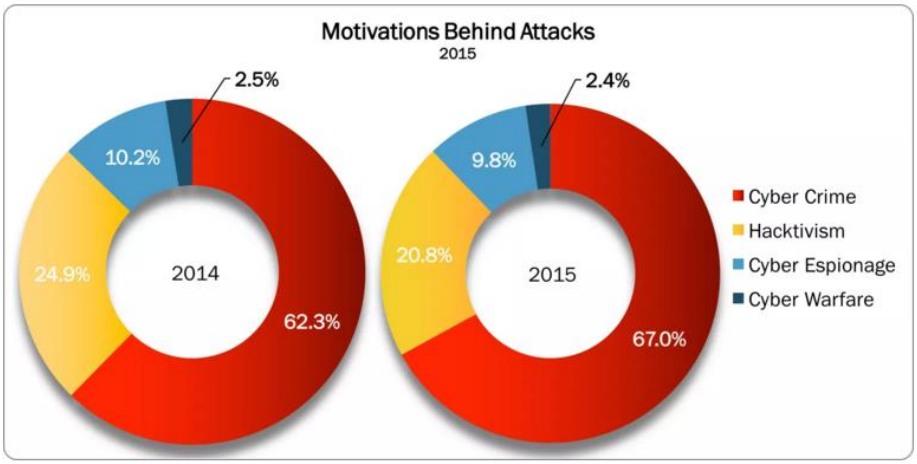



Figura 4 - Motivações para atacar
(Passeri P., 2016)

Um artigo da empresa Norton by Symantec refere que os *Cyber Crimes* se tratam de um episódio único que geralmente acontece através de um *download* de um cavalo de Troia. A ação poderá acontecer através de instalação de *software* malicioso ou vulnerabilidades de *software* já instalado no computador (Norton).
Portanto é necessário conduzir as empresas à adoção de mecanismos de aumento de fiabilidade, disponibilidade, integridade e segurança nos seus sistemas de informação. Desta forma a ISO 27001 tem sido importante para o panorama empresarial (Portal informativo ISO 27001). Segundo o portal informativo da norma ISO 27001 em Portugal, o seu objetivo é a

adoção pelas organizações de um conjunto de requisitos, processos e controlos com a finalidade de mitigar e gerir adequadamente o risco da organização (Portal informativo ISO 27001).

Existem para tais várias ferramentas para descobrir vulnerabilidades numa rede como por exemplo o Nessus[1] e o Openvas[2], sendo a última *Open Source* (Filho G., 2013). O Nessus também possuí uma versão de código aberto, sendo esta apenas para uso doméstico (Leonov V.). Tanto o Openvas como o Nessus são *scanners* de segurança que executam testes de vulnerabilidades que utilizando protocolos de comunicação aliados a *plugins* permitem manter as ferramentas atualizadas com as mais recentes descobertas de vulnerabilidades de segurança (Raymond G., 2009, Martinelo C. et Bellezi M., 2014).

Uma outra norma importante é a ISO 22301 que, quando implementada de forma correta, reduz o dano causado quando do aparecimento de incidentes (Fundamentos da ISO 22301) daí a monitorização ser importante.

Estas normas, caso cumpridas pelas empresas, pretendem sobretudo oferecer segurança, integridade e disponibilidade de sistemas de informação, dando uma certificação, sendo esta um selo de qualidade para as mesmas.

Importante também é o aspeto da automatização da infraestrutura. Durante muito tempo, as empresas apenas efetuavam poucas entregas do seu *software*/produto (Freire F., 2013). Esta prática por vezes ainda se verifica. Cada vez mais existe a necessidade da replicação de máquinas ou *softwares* para ambientes diferentes com configurações distintas. Caso exista um mecanismo que facilite essa mesma replicação, é possível agilizar tempo. Segundo Paul Duvall podemos dizer que a automatização da infraestrutura é o processo de criar *scripts* desde a instalação do sistema operativo até à configuração de *software* (Duvall P., 2012).

Para atingir esse objetivo existem as DevOps que ajudam as empresas a colocarem *software* e serviços mais rapidamente em produção, reduzindo os custos e aumentando a produção.

A figura seguinte mostra-nos uma comparação entre o desenvolvimento tradicional e o desenvolvimento através de DevOps (Contributor, 2015).

[1] www.tenable.com/Nessus
[2] http://www.openvas.org/

| | Dimensions | Traditional IT | DevOps |
|---|---|---|---|
| Planning & Organization | Batch Size | Big | Micro |
| | Organization | Skill Centric Silos | Dedicated Cells |
| | Scheduling | Centralized | Decentralized & Continuous |
| Performance & Culture | Release | High Risk Event | Non Event |
| | Information | Disseminated | Actionable |
| | Culture | Do Not Fail | Fail Early |
| Measure | Metric | Cost & Capacity | Cost, Capacity, and Flow (Time) |
| | Define "Done" | *"I did my job"* | *"Its ready to deploy"* |

Figura 5 - Comparação método tradicional vs. DevOps
(Contributor, 2015)

Após análise à figura anterior podemos constatar que o método tradicional, entrega uma grande quantidade de informação ao contrário das DevOps que entregam versões granulares e rápidas, desta forma respondendo facilmente aos impactos das mudanças. Ou seja, quanto maior for a *release* maior será o impacto em eventuais alterações de configurações e maior será o risco de gerar código defeituoso durante o seu desenvolvimento.

As DevOps mitigam os impactos das alterações de configuração, diminuindo o risco de código defeituoso tendo *a posteriori* uma maior facilidade e flexibilidade para implementar e distribuir novas configurações.

## 2.1 Análise de valor

A análise de valor tem por objetivo reduzir os custos e melhorar o desempenho do produto. Visa avaliar como aumentar ao valor de um item ou serviço com o menor custo, sem se sacrificar a qualidade do produto (Hughes D. & Chafin D., citado em: Nicola S.).

Segundo (Nicola S., Ferreira E. & Ferreira J., citado em: Nicola S.) "a criação de valor é a chave para qualquer empresa, e qualquer atividade de negócios é sobre troca de um bem ou serviço tangível e/ou intangível, sendo o seu valor aceite e recompensado pelos clientes." (adaptado).

No relacionamento tanto com o cliente como com o fornecedor torna-se importante saber como entregar e criar valor (Gil-Saura I., Del Toro M. et ai. 2009, p594, citado em: Nicola S.).

É necessário também saber distinguir os clientes, pois cada um percebe valor de maneira diferente (Ulaga & Eggert, 2006, citado em: Nicola S.). O mesmo acontece quando o valor percebido pelo produtor é diferente do valor percebido pelo cliente (Lindgreen & Wynstra,

2005, citado em: Nicola S.). Posto isto torna-se importante a elaboração de uma proposta de valor bem definida num negócio.

### 2.1.1 Valor (benefícios/sacrifícios)

Neste subcapítulo iremos abordar qual o valor (benefícios/sacrifícios) que esta solução trará para os seus clientes.

Um dos principiais valores (benefício) que a *Wonderbox* pretender oferecer é a simplificação das tarefas aos administradores de sistemas, aumentando essencialmente a disponibilidade da informação. Ou seja, visa monitorizar os serviços e servidores mais críticos do parque informático.

Outro valor (benefício) é o aumento da probabilidade de reposição de dados aos utilizadores após perda de informação, reduzindo o RPO e o RTO. Assim, podemos considerar a *Wonderbox* uma ferramenta que controla e vigia as necessidades para a continuidade de negócio por parte das empresas

Em termos de *target customers* podemos incluir qualquer empresa, pequena, média ou grande, que sinta necessidade de ter a informação disponível o maior tempo possível e/ou o aumento da probabilidade de reposição de dados com sucesso e num curto espaço de tempo (sendo "curto" proporcional ao volume de informação a repor).

Posto isto consideramos que o sacrifício para os *customers* será o custo da solução.

A tabela seguinte visa resumir os benefícios e os sacrifícios.

Tabela 1 - Valor (benefícios/sacrifícios)

| **Benefícios** | **Sacrifícios** |
|---|---|
| Simplificação das tarefas aos *sysadmins* | Custo |
| Continuidade de negócio | - ou n/a |

Só será possível quantificar a criação de valor com base numa comparação entre cenário atual e o cenário futuro após a implementação do produto no mercado

### 2.1.2 Modelo de Canvas

O autor desta dissertação criou um modelo de negócio para descrever a lógica de como uma organização cria, proporciona e obtém valor.

Com base nesta premissa preencheu-se o Modelo de Canvas que é uma ferramenta de gestão estratégica de negócio. É possível de se observar este modelo na figura 6.

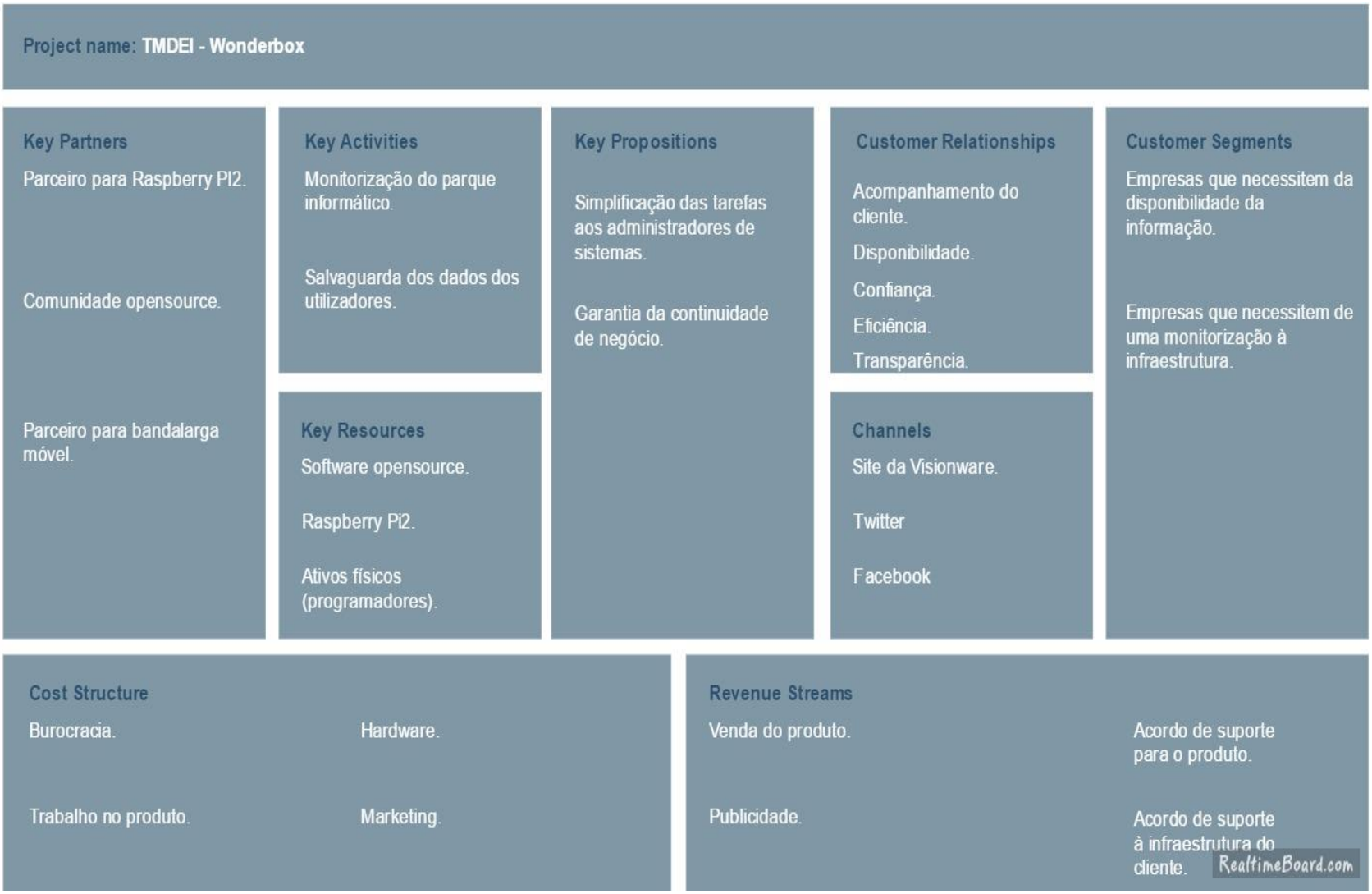



Figura 6 - Modelo Canvas

Este modelo está dividido em 9 partes distintas, segmentos de clientes, proposta de valor, canais de comunicação, relacionamento com clientes, fontes de receita, recurso-chave, atividades-chave, parcerias-chave e estruturas de custos.

Como se pode observar na figura anterior, os segmentos de clientes destinam-se a um grupo de pessoas ou organizações que a empresa pretende servir. Neste caso, todas as empresas que necessitem de ter a informação disponível e/ou empresas que necessitem de monitorização à rede da infraestrutura.

Na proposta de valor, a Visionware pretende com este produto oferecer os seguintes benefícios: Simplificação das tarefas aos administradores de sistemas e garantia na continuidade de negócio.

Os canais de comunicação para com o cliente são definidos num foco *online*, o sítio da Visionware na internet, o Twitter e o Facebook.

O relacionamento hoje em dia é importante para o negócio. Os clientes podem esperar por parte da Visionware confiança, eficiência, disponibilidade e total transparência. A Visionware também irá fazer um acompanhamento inicial com a passagem de conhecimento do funcionamento do produto.

Para que a relação seja profícua para ambos os lados, a Visionware gerará receita através da venda do produto, publicidade e acordo de suporte para o produto e infraestrutura.

Os recursos-chave para que o modelo de negócio funcione são os ativos físicos (programadores), o *software Open Source* e o Raspberry Pi. Sem isto não seria possível a elaboração do produto.

As atividades-chave são uma das peças mais importante para que o modelo de negócio funcione. No caso concreto, a monitorização do parque informático e a salvaguarda dos dados dos utilizadores.

Os parceiros-chave também são importantes, daí destacam-se um parceiro para o Raspberry Pi2, que poderá ser um fornecedor existente no mercado, um outro parceiro para a banda larga da internet, que poderá ser qualquer dos operadores existentes à data e depois toda a comunidade *Open Source* que servirá para esclarecimento de dúvidas.

Obviamente que esta operação também tem custos, entre os quais se pode classificar a burocracia com os contratos de venda, suporte, etc., sem deixar de mencionar o custo dos recursos humanos da Visionware envolvidos nem o preço do *hardware* e o *marketing*.

## 2.2 Gestão de Redes

Hoje em dia a área de IT não é apenas mais um departamento técnico, pois desempenha um papel fundamental para a área de negócios das empresas, procurando otimizar, reduzir custos e riscos intrínsecos à atividade de negócio.

### 2.2.1 Business Continuity Plan

A direção de uma empresa deve conhecer todas as partes e fases do BCP e aprovar as ameaças e riscos que podem afetar os ativos de informação. O BCP é da inteira responsabilidade dos cargos de gestão de uma organização e o que se espera da equipa IT é apenas auxílio, não podendo ser repassado a responsabilidade da implementação ou não.

A figura seguinte indica-nos as duas componentes de um BCP (PublicSafety Canada; Rodrigues C.).


Figura 7 - Componentes de um BCP
(Rodrigues C.)

Como foi possível de observar, saltam à vista duas componentes bastante importantes que são essenciais para a formação de um BCP. O RA – *Risk Analysis* e o BIA – *Business Impact Analysis*.

#### 2.2.1.1 Risk Analysis

O RA, em português, Avaliação de Risco, identifica as probabilidades de um risco vir a acontecer. Avalia esse risco e o impacto que poderá ter para a empresa, no caso de um determinado sistema ou serviço deixar de funcionar. Aqui, o objetivo é ter o conhecimento necessário para poder atenuar o risco (Kosutic D.; Rodrigues C.).

#### 2.2.1.2 Business Impact

O objetivo do BIA é manter o funcionamento correto do negócio. Este avalia a probabilidade de uma ameaça vir realmente a acontecer e o impacto que a ameaça poderá vir a ter (Kosutic D.; Rodrigues C.).

### 2.2.2 Disaster Recovery Plan

Em português, Plano de Recuperação de Desastre. Está intrinsecamente ligado ao BCP. É a possibilidade de uma empresa recuperar o seu funcionamento parcial ou total em caso de desastre. O DRP – *Disaster Recovery Plan* é, portanto, parte integrante do BCP (DisasterRecovery; Rodrigues C.).

Antes de se selecionar uma estratégia de recuperação de desastres, dever-se-á ter em atenção as principais métricas como se pode ver na figura nº 7 (Horseproject).

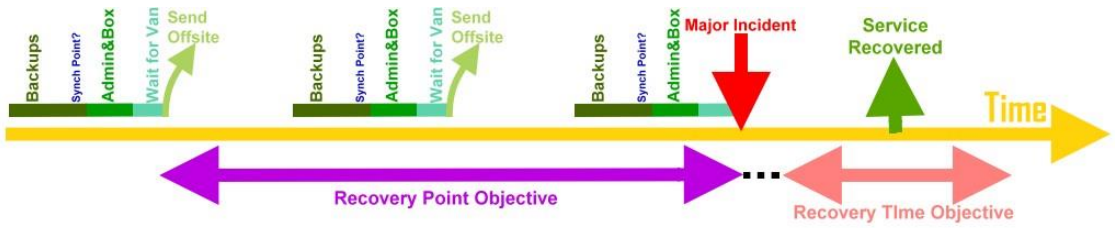



Figura 8 - Estratégia para recuperação de informação (Horseproject)

RPO – *Recovery Point Objective* e RTO – *Recovery Time Objective*. Em português, Objetivo e Ponto de Recuperação e Objetivo de Tempo de Recuperação respetivamente.

O RPO é a quantidade máxima de informação que se é aceitável perder durante um acidente, ou seja, é o ponto no tempo para o qual vai ser restaurada a informação. Já o RTO é o tempo necessário para que todo o sistema volte a estar completamente funcional.

## 2.3 Monitorização Alarmística

Neste capítulo vamos dar ênfase à monitorização alarmística, ferramentas que nos permitam atingir o objetivo. Existem várias ferramentas que disponibilizam monitorizações constantes à infraestrutura, mas uma das restrições atuais do local da elaboração da dissertação passa inevitavelmente por tecnologias *Open Source* e por um *design* apelativo para o cliente final.

Dadas estas imposições surgiram 3 opções. O Nagios, o Icinga, o Zabbix e o Observium.

### 2.3.1 Nagios

O Nagios é uma das ferramentas de monitorização mais utilizadas a nível mundial (Tabona A., 2015). É de código aberto e pode monitorizar *hosts* (*routers*, servidores, etc.) e serviços, alertando quando ocorrem problemas.

Estes serviços serão todos os inerentes ao funcionamento da empresa, ou seja, serão ajustados à atividade da empresa em causa. Permite agregar estes serviços a um conjunto de máquinas, *hosts.*

Foi originalmente criado sob o nome de NetSaint, mudando posteriormente o nome para NagiosAin'tGonnaInsistOnSainthood – Nagios. Foi escrito e é mantido atualmente por Ethan Galstad, junto com uma equipa de desenvolvimento que mantém *plugins* oficiais e não oficiais. Começou por atrair popularidade e uma grande comunidade de programadores devido a uma arquitetura de funcionamento que permitia o desenvolvimento de extensões simples por parte de terceiros criando uma grande flexibilidade em introduzir de forma simples e rápida no programa novas funcionalidades (NagiosArchitecture, 2015).

A arquitetura do Nagios como podemos ver na figura seguinte é constituída por duas componentes, o Nagios *Core* e o Nagios Web CGIs. Estes comunicam entre si através de uma *cache* e de um *pipe*. A interface WEB do Nagios é o Nagios Web CGIs. Esta recebe os dados de monitorização através da *cache* e envia comandos via *pipe* para o *command file*, ficheiro de comando. O sistema é controlado pelo *core*. Este lê os ficheiros de configuração, interpretando-os e depois executando os comandos (Casella G., 2011; Laurent A. Et Rémi B., 2008; Macedo V., 2011).

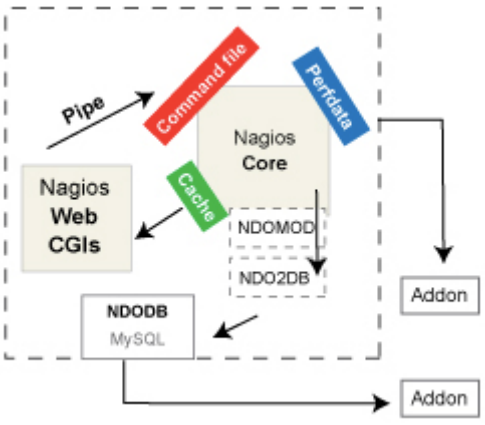



Figura 9 - Funcionamento do Nagios
(Casella G., 2011)

As monitorizações às máquinas e serviços são executadas através de *plugins*. Estes são chamados pelo *core* do Nagios para analisar os *hosts* e serviços. Os *plugins* podem ser *scripts*, escritos em Python, Shell Script ou Perl. Associado aos *scripts* podemos ter quatro níveis de

aviso. *Ok, Warning, Critical*, *Unknown*. Estes poderão ser definidos quando da sua configuração.

Os *addons* estendem as funcionalidades do Nagios e interpretam os dados recolhidos, mostrando-os de forma mais amigável aos utilizadores.

A comunicação com a base de dados é feita através de um módulo chamado NDOMOD que está ao mesmo nível do core e por um *daemon*, NDO2DB que escuta um *socket* da base de dados MySQL (Nagios to RuntimeDatabase).

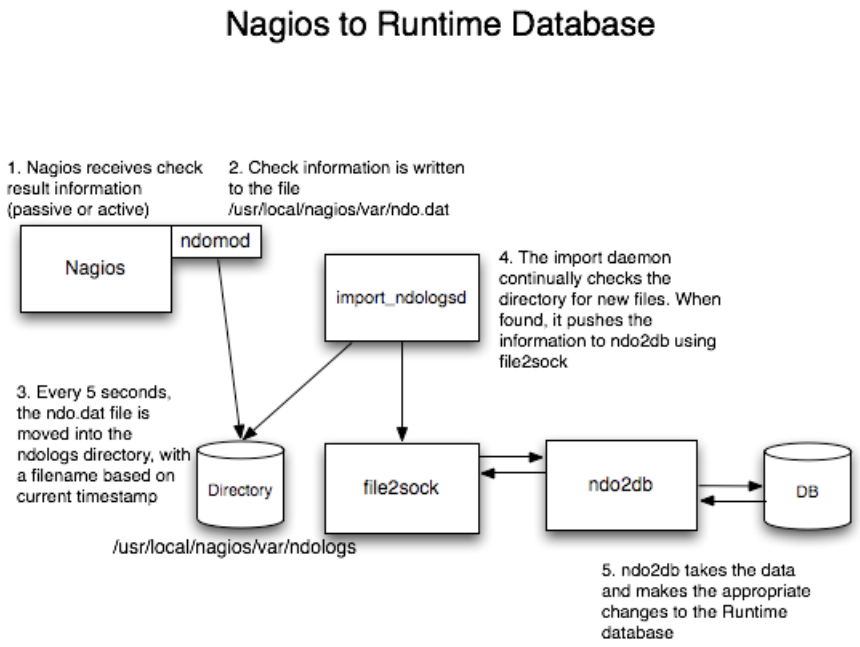



Figura 10 - Funcionamento da base de dados do Nagios
(Nagios to RuntimeDatabase)

Entretanto surgiu o Nagios XI que tem o suporte incluído. Esta versão é paga, contém alguns *addons* já incluídos e uma interface mais amigável do utilizador.

O Nagios também permite uma solução de monitorização distribuída, mas esta solução é paga, NagiosFusion. Isto torna a monitorização escalável e é possível de termos várias soluções Nagios Core e Nagios XI interligadas com o NagiosFusion como podemos ver na imagem seguinte (NagiosEnterprises, 2013).

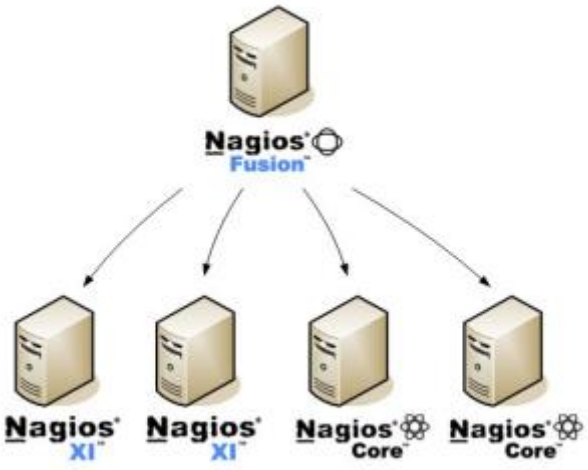



Figura 11- Monitorização distribuída no Nagios
(NagiosEnterprises, 2013)

O Nagios não é um *software* fácil de configurar, requer algum conhecimento técnico por parte do utilizador. Contudo visto as restrições assumidas nesta dissertação de mestrado, em termos de escalabilidade para implementações futuras não é possível ter uma solução centralizada que receba alertas de sítios diferentes, pois o NagiosFusion é pago. A interface gráfica também não é muito apelativa.

### 2.3.2 Icinga

O Icinga resulta de um desfasamento em relação ao desenvolvimento oficial do Nagios. Embora mantendo a compatibilidade com os *plugins* do Nagios, razão da popularidade da plataforma. O Icinga possui uma arquitetura diferente do Nagios, procurando melhorar o desempenho e ser mais modular.

Em 2012 foi criada uma nova versão que é mantida em paralelo com a versão inicial, chamada de Icinga2 (Icinga; Team Icinga, 2014).

O Icinga2 é um sistema de monitorização *Open Source* que verifica a disponibilidade dos recursos da rede e notifica os utilizadores sobre as interrupções, podendo gerar relatórios de dados. Corre em múltiplas versões de Linux (Fedora, Ubuntu e openSuSE) e também em algumas Unix (Solaris e HP)

Disponibiliza aos administradores de sistemas uma vasta gama de funcionalidades entre as quais se destacam as seguintes:

- Monitorização de serviços (SMTP, POP3, HTTP, ICMP, etc.).

- Monitorização de recursos de rede (Capacidade atual do CPU, utilização do disco, etc.).
- Facilidade de desenvolvimento dos administradores de sistemas para os seus próprios *plugins*.
- Execução de análises em paralelo.
- Possibilidade de receção de notificação quando um serviço ou *host* está indisponível via e-mail, *pager* ou outro método definido posteriormente.
- Possibilidade de receção de notificações durante a execução de um serviço ou *host* afim de proactivamente poder fazer uma intervenção antes que o serviço ou *host* tenha algum problema.
- Interface Web 2.0 *user-friendly*.

É escalável e extensível, ao contrário da primeira versão, podendo monitorizar grandes ambientes complexos em localizações diferentes. Utiliza dois tipos de base de dados, MySQL e PostgreSQL.

Apresenta a seguinte arquitetura como demonstrada no esquema seguinte (Hein M. et Friedrich M., 2013).

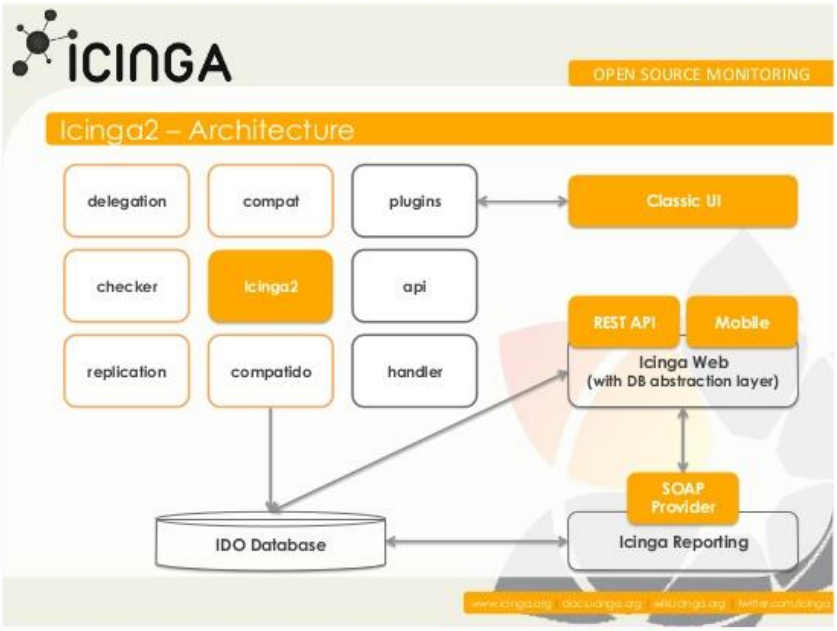



Figura 12 - Arquitetura do Icinga2

É composta por três componentes distintas que funcionam em simultâneo. O Icinga2 que é considerado o *core*, a IDODB – *Icinga Data Out Database* e a Interface Web (Team Icinga, 2014). A componente de Icinga2 gere as tarefas de monitorização, os serviços que vão ser monitorizados, ou seja, é a componente essencial para o funcionamento do sistema. Faz a interligação entre a base de dados (IDODB) e a interface gráfica (Classic UI). A componente de IDOBD é onde está alojada a informação, *reports*, dados de utilizadores para acesso à interface gráfica, entre outros. A *classic* UI é a interface gráfica do Icinga2. Recentemente foi

desenvolvida uma nova interface, mais moderna e com maior usabilidade para os utilizadores, podendo estes optar por quaisquer uma aquando na instalação.

Como já referimos anteriormente, o Icinga2 oferece escalabilidade. É possível termos um sistema distribuído de monitorização. Como podemos ver na figura seguinte (Icinga 2).

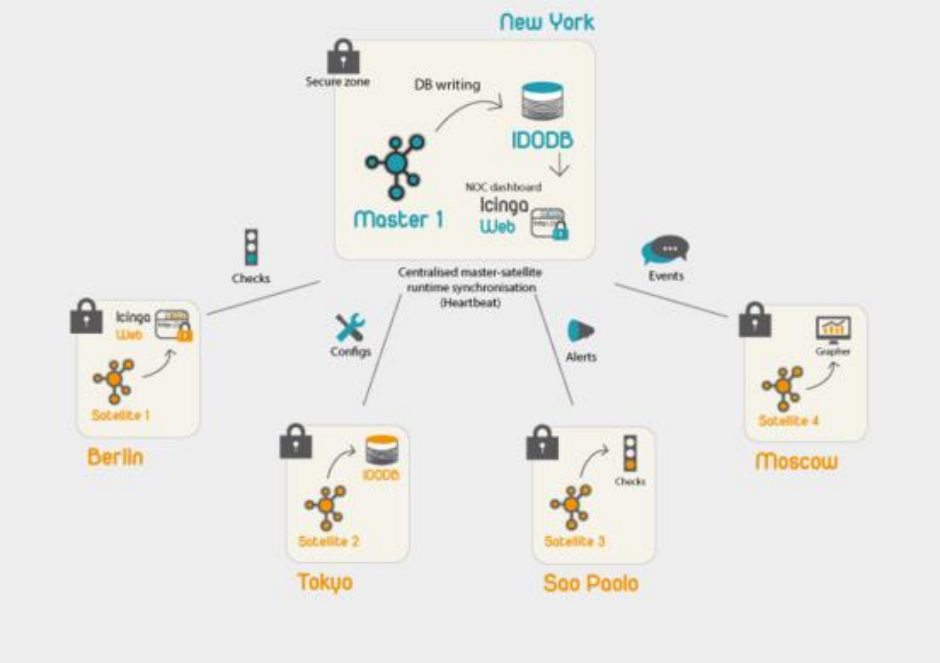



Figura 13 - Monitorização distribuída no Icinga2
(Icinga 2)

É possível termos satélites em diferentes localizações geográficas. Estes satélites podem ser simples ou totalmente equipados com sistema IDODB, interface WEB, entre outras.

Segundo o sítio (Icinga 2) "A replicação pode ser isolada a ocorrer entre a zona mestre e cada diferente satélite" (adaptado).

O Icinga como já referido resulta de um *fork*[3] do Nagios. Em termos técnicos para instalação também requer conhecimentos técnicos por parte do administrador de sistemas. Aqui, neste caso, a escalabilidade é possível sem quaisquer custos de *software* adicional, além de ter uma interface gráfica apelativa.

### 2.3.3 Zabbix

O Zabbix foi criado por Alexei Vladishev (Zabbix) e é atualmente suportado pela Zabbix SIA (Macedo V., 2011). À semelhança do Nagios e do Icinga também é um *software* de

[3] Desenvolvimento de um novo produto partindo de um produto existente.

monitorização. A informação é guardada em base de dados, tornando depois possível a geração de relatórios.

A figura seguinte mostra-nos a arquitetura do Zabbix.

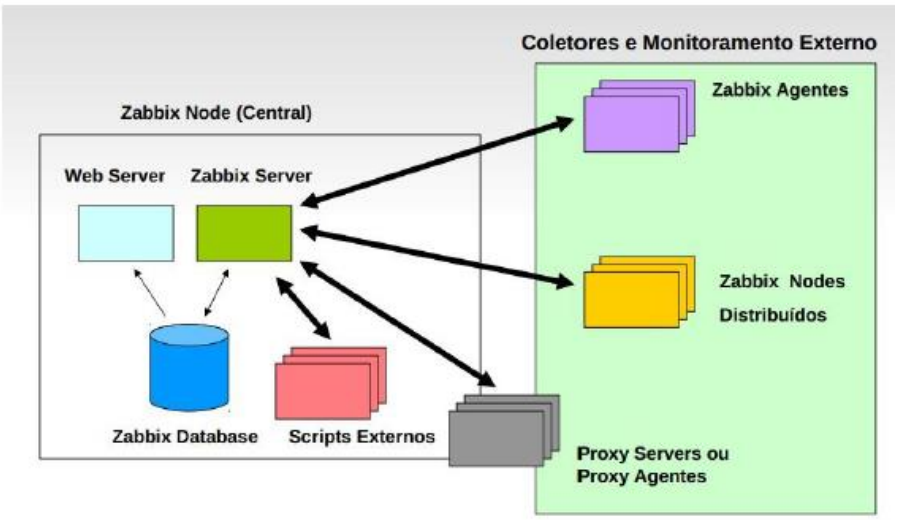



Figura 14 - Arquitetura do Zabbix
(Markovski M., 2016)

Como temos a possibilidade de observar, é uma solução distribuída em duas zonas, um componente central, que é constituída por um servidor, base dados onde ficam guardadas todas as configurações e o interface web, utilizado para efetuar configurações mas também para análise aos dispositivos que estão a ser monitorizados.

A outra zona refere-se aos agentes que são instalados nos dispositivos a monitorizar. Tem como função recolher dados e enviar para o servidor central.

### 2.3.4 Observium

À semelhança das ferramentas anteriores o Observium permite ao administrador de sistemas monitorizar a sua infraestrutura. É uma plataforma desenvolvida em PHP/MySQL e tem suporte para Linux, Windows, Cisco, HP, Dell, entre outras. Possui uma versão profissional, não gratuita, e uma versão *Open Source* onde à data a versão estável é a 0.15.6 (Observium, ver referências, Pinto P., 2014).

Esta ferramenta funciona por *autodiscover* através do protocolo SNMP - *Simple Network Management Protocol*, coletando de forma automática o ambiente da rede.

Poderá ser fundamental em caso de *disaster recover* uma vez que possuí várias métricas que podem ser coletadas, como por exemplo a voltagem e a velocidade da ventoinha dos ativos de rede, entre as habituais métricas já habitualmente disponibilizadas por este tipo de ferramenta como carga de servidores, etc., (Network Management & Monitoring, 2014).

Em termos gráficos como podemos ver na imagem seguinte, apresenta, na opinião do autor, uma interface limpa, mas pouco intuitiva para os utilizadores.

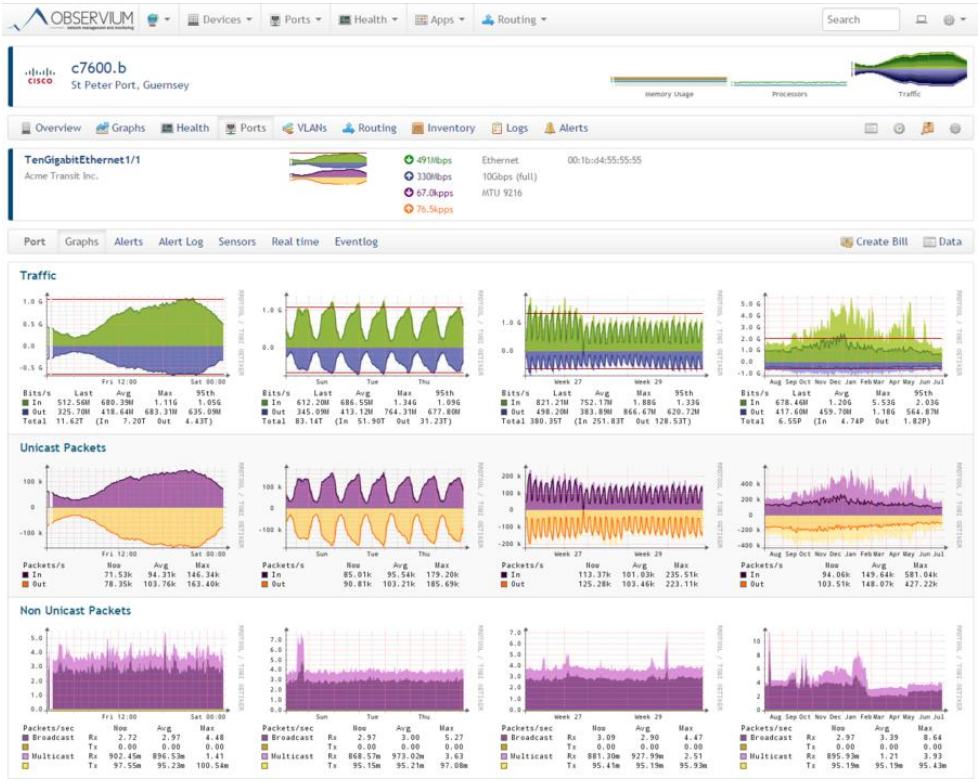



Figura 15 - Captura de imagem da ferramenta Observium

Após alguma pesquisa verificou-se que a comunidade de apoio para esclarecimento de dúvidas não é muito vasta, apostando o Observium na solução *professional*.

### 2.3.5 Nagios VS Icinga2 VS Zabbix vs Observium

Neste subcapítulo será elaborada uma breve comparação entre as tecnologias estudadas para responder aos problemas desta dissertação.

A tabela seguinte serve para este efeito.

Tabela 2 - Comparação entre ferramentas de monitorização

| **Termo de comparação** | **Nagios** | **Icinga 2** | **Zabbix** | **Observium** |
|---|---|---|---|---|
| **Usabilidade** | Não é apelativa | Bastante apelativa | Não é apelativa | Pouco apelativa |
| **Escalabilidade** | Permite, mas é pago. | Permite | Permite | Permite |

| | | | | |
|---|---|---|---|---|
| **Alertas para os administradores de sistemas** | Permite | Permite | Permite | Permite |
| **Sistemas Windows** | Permite | Permite | Permite | Permite |
| **Sistema centralizado** | Permite | Permite | Permite | Permite |
| **Dificuldade de configuração** | Difícil | Difícil | Difícil | Difícil |

Como tivemos oportunidade de analisar a tabela anterior, à exceção do Nagios que em termos de escalabilidade apresenta custos, o Icinga 2, o Zabbix e o Observium preenchem os requisitos, sendo que o Icinga 2 em termos de interface gráfica e usabilidade é mais apelativo.

## 2.4 Cópias de Segurança

As cópias de segurança são importantes para o nosso dia-a-dia, pois são a única maneira de salvaguardarmos os nossos documentos de forma segura (Ramos A.). Estas cópias devem realizar-se de forma periódica, devendo as mesmas ser realizadas manualmente ou automaticamente. Caso sejam automáticas o risco de esquecimento não existe.

Realizar cópias de segurança de forma periódica tem as suas vantagens, entre as quais, a vantagem da cópia de segurança guardada ser distinta da informação que está no nosso computador. Como já referido anteriormente permite uma proteção contra falhas no computador, nomeadamente falhas de disco e até mesmo em caso de portátil deterioração, apagamento acidental de documentos, entre outros.

Algo mais importante que os *backups* é a implementação da sua política. É necessário sensibilizar os utilizadores para tal. Eles precisam de saber a que horas deve ser efetuada a cópia, onde deve ser armazenada e a sua retenção.

### 2.4.1 Estratégias de cópias de segurança

Neste subcapítulo será abordado o tipo de cópias de segurança existentes, alguns meios para armazenar a informação.

#### 2.4.1.1 Tipos de cópias de segurança

Hoje em dia existem essencialmente três tipos de *backups*, os completos, os incrementais e os diferenciais (Macêdo D., 2012; Webmaster, 2012).

##### 2.4.1.1.1 Cópia de segurança completa

O *backup* completo, é simplesmente efetuar uma cópia completa a todo o sistema, independentemente de versões anteriores ou de alterações anteriores até à data da última cópia (Macêdo D., 2012; Webmaster, 2012).

A seguinte tabela indica-nos as principais vantagens e desvantagens no uso deste tipo de cópias.

Tabela 3 - Vantagens e desvantagens da cópia de segurança completa

| **Vantagem** | **Desvantagem** |
|---|---|
| Facilidade na localização de ficheiros para restauro. | Tempo necessário para a cópia. |
| Facilidade de manutenção e restauro de diferentes versões. | Consumo de espaço de armazenamento bastante significativo. |

#### 2.4.1.1.2 Cópia de segurança incremental

O *backup* incremental é uma cópia de todas as alterações de informação que exista desde o último *backup* (Macêdo D., 2012; Webmaster, 2012). Ele começa por primeiro verificar se a hora de modificação do ficheiro é mais recente que a do último *backup*, caso seja ele copia a informação, caso contrário, acaba por descartar a cópia.

Este tipo de cópia para que seja executada, necessita de uma cópia de segurança completa prévia e ainda de todos os *backups* incrementais já realizados.

A seguinte tabela indica-nos as principais vantagens e desvantagens no uso deste tipo de cópias.

Tabela 4 - Vantanges e desvantagens da cópia de segurança incremental

| **Vantagem** | **Desvantagem** |
|---|---|
| Velocidade da cópia. | Restauro lento. |
| Eficiência do uso do armazenamento de *backup*. Não existem ficheiros duplicados. | Restauro mais complicado uma vez que é necessário ter disponível quer o *backup completo* quer os incrementais efetuados. |

#### 2.4.1.1.3 Cópia de segurança diferencial

Da mesma forma que o *backup* incremental, o *backup* diferencial também só copia a informação modificada desde a última cópia de segurança (Macêdo D., 2012; Webmaster, 2012). A diferença aqui reside na não análise ao último *backup* diferencial, mas sim ao último completo. Isto quer dizer, que cada *backup* diferencial contém todos os arquivos modificados desde o último *backup* completo.

Este tipo de cópia para que seja executada, necessita de uma cópia de segurança completa.

A seguinte tabela indica-nos as principais vantagens e desvantagens no uso deste tipo de cópias.

Tabela 5 - Vantagens e desvantagens da cópia de segurança diferencial

| Vantagem | Desvantagem |
|---|---|
| Velocidade da cópia mais rápida que o *backup* completo. | Cópias mais lentas que o *backup* incremental. |
| Eficiência do uso do armazenamento de *backup*. Não existem ficheiros duplicados. | Restauro mais lento que o *backup* completo. |
| Restauro mais rápido que o *backup* incremental. | Restauro mais complicado uma vez que é necessário ter disponível o *backup* completo, mas mais simples que o incremental |

#### 2.4.1.2 Cópia integral ao disco

Este tipo de cópia é a mais fácil de efetuar e a mais fácil de compreensão para os utilizadores finais. É uma cópia total ao disco, podemos efetuar uma imagem, clone, ao disco através de *software* como o Acronis[4].

#### 2.4.1.3 Armazenamento

Esta questão de armazenamento é complexa, pois estamos a falar de informação sensível. Vamos imaginar que um utilizador apaga algum ficheiro acidentalmente e necessita de um restauro imediatamente? Deverá ele ter a informação perto dele?

E em caso de desastre natural valerá apenas ter a cópia de segurança nas mesmas instalações? Não. Se ocorrer um sismo, um incêndio, nada valerá apenas a existência da cópia de segurança.

Visto o autor da presente dissertação já ter aproximadamente 2 anos de trabalho nesta área, por experiência própria, uma das táticas que está a ser utilizada recentemente é uma cópia de segurança para um armazenamento nas instalações da organização e uma outra cópia de segurança para um ambiente remoto.

Posto isto, em ambiente empresarial, existem vários tipos de suporte para armazenamento das cópias de segurança (Macêdo D., 2012; Microcom).

- *Tape* – Uma *tape* é uma cassete de fita magnética. Necessitam de unidades próprias para a gravação. Aqui, os dados são gravados de forma sequência, o que significa que para retirar um determinado documento que se encontra no meio da cassete é necessário percorrê-la até chegar a esse ponto.
- Disco – Aqui um disco poderá ser um disco físico ou virtual (iSCSi).
- *Cloud* – Este tipo de suporte para armazenamento é cada vez mais utilizado, uma vez que previne a perda de informação em caso de desastre natural.

[4] http://www.acronis.com/

### 2.4.2 Análise ao software de mercado

Nesta secção vamos efetuar uma análise ao *software* de mercado, para efetuar cópias de segurança. Aqui, à semelhança da monitorização alarmística, a Visionware impõe as mesmas restrições, *Open Source* e uma usabilidade fácil, gestão centralizada e com um *design* apelativo para o cliente.

#### 2.4.2.1 Bareos

O Bareos- *Backup Archiving Recovery Open Source* é um *fork* do conhecido Bacula que é igualmente uma ferramenta de cópias de segurança. É *Open Source* e o seu código está atualmente disponível num repositório github e é suportada pela licença AGPLv3. Atualmente a versão estável mais recente é a versão 15.2 (Bareos 15.2).

Tem uma interface apelativa e intuitiva para o utilizador final como podemos observar na imagem seguinte.

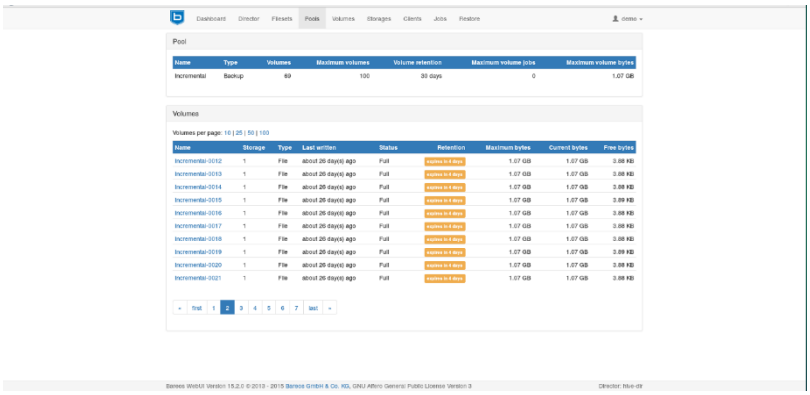


Figura 16 - Imagem do Bareos
(Bareos 15.2)

Esta interface é escrita em PHP e também é *Open Source*.

É uma *framework* flexível que permite a sua implementação em diferentes tipos de plataformas. Contém um conjunto de módulos que podem estar instalados em máquinas diferentes.

A figura nº 17 indica-nos o funcionamento do Bareos.

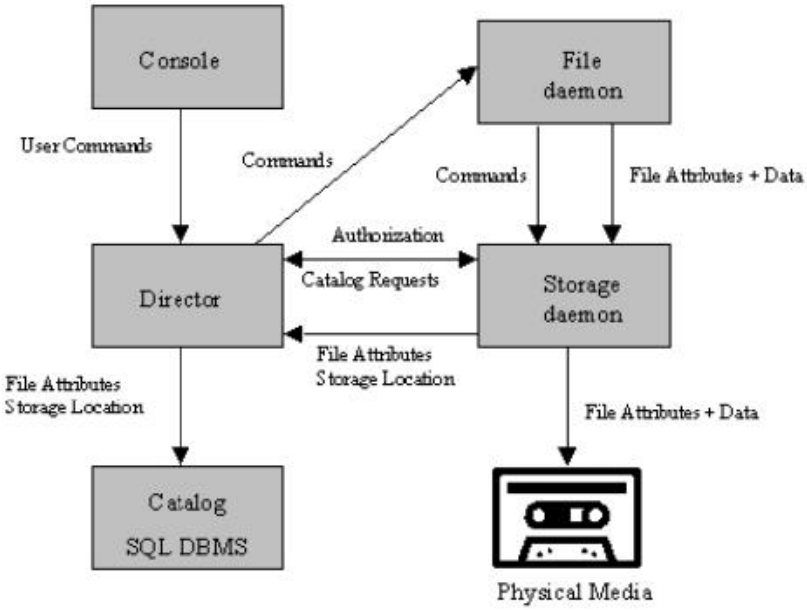



Figura 17 - Arquitetura do Bareos
(Bareos Main Reference, 2016; Gula I. et Beerheide T., 2015; Eric, 2011)

Como se observa, a *framework* é constituída por 5 módulos mais um mídia físico. Os 5 módulos são *Director Daemon*, *Console Manager*, *File Daemon*, *Storage Daemon* e *Catalog*.

O *Director Daemon* é responsável pelos processos de cópias de segurança, como por exemplo, a cópia, restauro e arquivo. É também aqui que se define o agendamento das tarefas de *backup*.

*File Daemon*, é responsável por enviar a informação solicitada pelo *Director Daemon* para a rede. É o agente instalado nas máquinas que necessitem de cópias de segurança.

*Console Manager*, ajuda na comunicação entre o utilizador e o *Director Daemon*. Pode ser executado a partir de qualquer computador na rede. Atualmente existem 3 versões, texto, em interface gráfica usando bibliotecas no *Gnome* e outra em interface gráfica usando bibliotecas *wxWidgets*.

*Storage Daemon*, permite administrar a gravação e restauro da informação das cópias de segurança em mídias.

*Catalog*, mantém a indexação de todos os arquivos que são armazenados e gera uma base de dados dos volumes geridos pelo *Director Daemon*.

Algumas características que se possam destacar deste *software* são as seguintes:

- Cliente, controlador das cópias de segurança e armazenamento, independentes.
- Gestão centralizada através de uma interface gráfica multiutilizador.
- Autenticação dos serviços de cópias através senha e chave criptográfica.

- Armazenamento de *backups* para diferentes dispositivos.
- Permissão para execução de *scripts* antes/depois do início de cada tarefa.
- Envio de informação para utilizadores após cada tarefa.
- Gestão das tarefas através da consola "*bconsole*".
- Vários canais de suporte.

O autor considera o Bareos pelas suas características uma ferramenta de gestão de *backup* de aplicabilidade a pequenas, médias e grandes empresas. Além disso é multiplataforma e poderá oferecer aos administradores de sistemas uma interface gráfica de gestão centralizada e apelativa.

#### 2.4.2.2 Amanda

O Amanda - *Advanced Maryland Automatic Network Disk Archiver* segundo a fonte (Amanda Network Backup, 2016) "é uma solução de cópias de segurança que permite aos administradores de sistemas terem uma solução centralizada, servidor, para fazer a gestão das cópias de segurança" (adaptado).

É uma solução *Open Source* e é escrita na linguagem C e Perl. Foi desenvolvido inicialmente na Universidade de Maryland, CollegePark, Zmanda (Amanda).

Atualmente a versão mais recente e estável é a 3.3.8.

Na imagem seguinte podemos verificar desde já, a disponibilidade do Amanda para suportar diferentes plataformas cliente (Amanda Community vs Enterprise Editions).

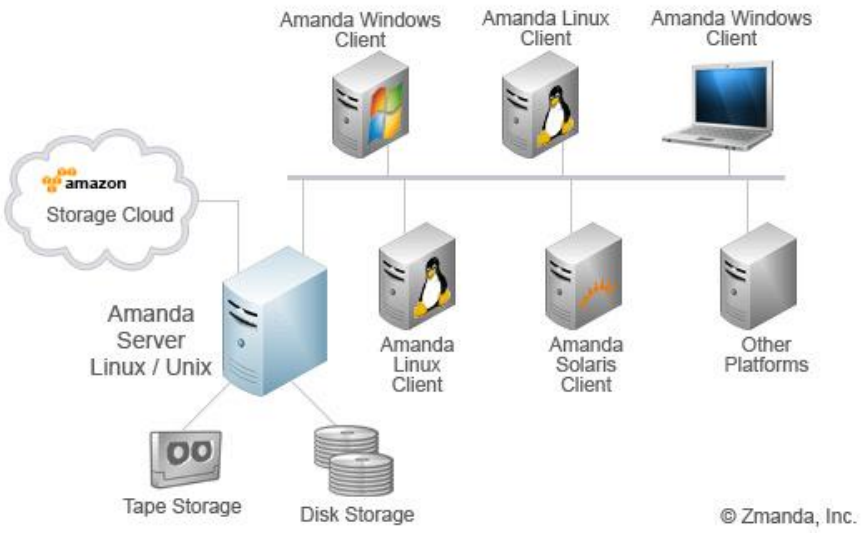



Figura 18 - Arquitetura do Amanda
(Amanda CommunityvsEnterpriseEditions)

Algumas das funcionalidades que podemos destacar desta ferramenta é a possibilidade de *backup* para NAS, SAN, iSCSI e até nuvem. Tem inteligência na programação da cópia de segurança.

O Amanda também contém uma versão empresarial que incluí *feedback* às cópias de segurança, replicação dos servidores de *backup* e suporte 24 x 7, entre outros (Amanda Community vs Enterprise Editions).

#### 2.4.2.3 Cobian

O Cobian é um *software* de *backup* para Windows criado por Luis Cobian em 2000 e escrito em Delphi (Cobian Backup).

Este permite agendar tarefas recorrentes com cópias de 4 tipos, completo, incremental, diferencial e *dummy*, este último permite a execução de programas, reinício do computador, entre outros (Cobian, 2006; Duarte H., 2014).

É possível comprimir as cópias em formato .zip ou .7zip, encripta-los em 3 métodos diferentes, nomeadamente AES128, AES192 e AES256 e protege-los através de uma senha.

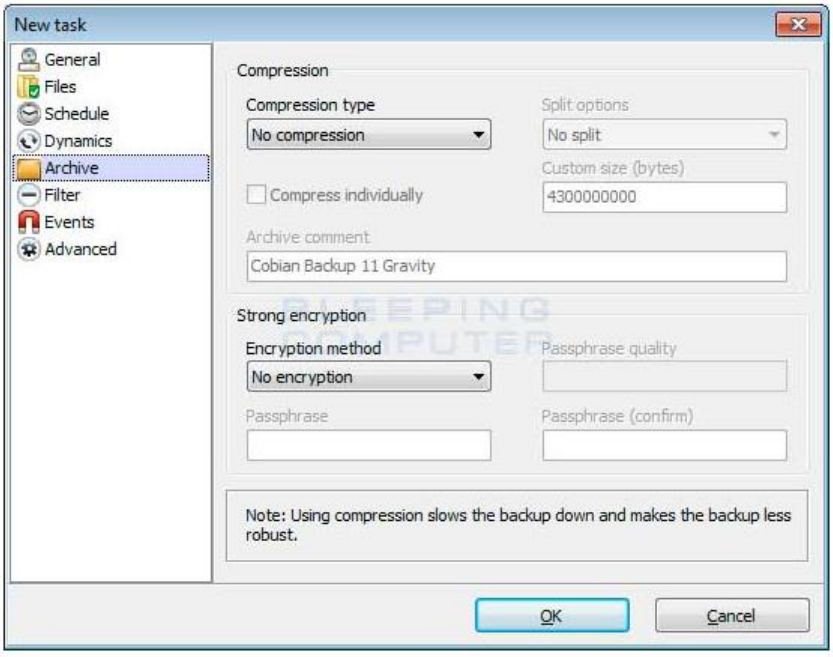



Figura 19 - Arquitetura do Cobian
(captura de imagem da interface do *software*)

Em relação à origem dos ficheiros podemos especificar, ficheiros, pastas, sítios FTP. É também possível indicar manualmente o caminho para as pastas que pretendemos selecionar para a cópia. O destino poderá ser uma pasta ou um sítio FTP.

Após teste, instalação e configuração v. 11, o autor considera a interface bastante confusa e o funcionamento mau, pois caso o Cobian não consiga contactar com o destino, ao fim de algum

tempo descarta a tarefa da cópia. Outra desvantagem é a ausência de uma interface gráfica centralizada de manuseamento para o administrador de sistemas.

#### 2.4.2.4 Duplicati

O Duplicati é um cliente de *backup* que armazena dados na nuvem. Ao contrário do Amanda que permite armazenar para a nuvem (Amazon) e para outros tipos de dispositivos, o Duplicati apenas permite o armazenamento para nuvem como podemos ver na captura de imagem seguinte.

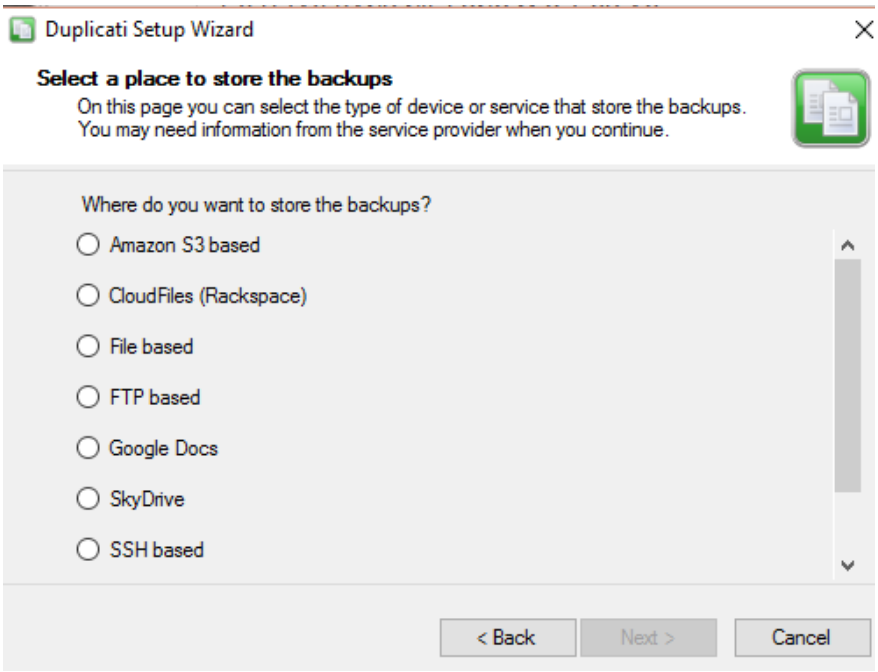



Figura 20 - Captura de imagem Duplicati

Esta ferramenta é gratuita e é disponibilizada para Windows, Linux e MacOS. Escrito por Kenneth Skovhede em 2008 na linguagem C#, atualmente a sua versão estável é a 1.3.4. O seu ponto forte é a grande variedade de armazenamento que oferece aos utilizadores para *cloud* (Duplicati).

Algumas das características desta ferramenta são as seguintes:

- Encriptação AES256;
- Possibilidade de cópias de segurança incrementais para não ocupar muito espaço;
- Armazenamento na Amazon, SkyDrive, Google Drive, Rackspace Cloud Files, WebDAV, via SSH e FTP.
- Utiliza o VSS - *Volume Snapshot Service* do Windows, desta forma é possível efetuar uma cópia aos ficheiros .pst com o Outlook aberto.

Na opinião do autor, esta ferramenta é bastante simples de utilizar, sendo bastante intuitiva para os utilizadores. Contudo carece de uma solução centralizada e armazenamento para outros suportes.

#### 2.4.2.5 Bareos VS Amanda VS Cobian VS Duplicati

Neste subcapítulo pretende-se fazer uma comparação entre as tecnologias estudadas para as cópias de segurança.

Tabela 6 - Comparação entre ferramentas de cópias de segurança

| **Termo de comparação** | **Bareos** | **Amanda** | **Cobian** | **Duplicati** |
|---|---|---|---|---|
| **Usabilidade** | Bastante apelativa | Não é apelativa | Apelativa | Apelativa |
| **Sistemas Windows** | Permite | Permite | Permite | Permite |
| **Sistema centralizado** | Permite | Permite | Não permite | Não permite |
| **Dificuldade de configuração** | Difícil | Difícil | Fácil | Fácil |

Após esta comparação efetuada pelo autor, apenas o Bareos e o Amanda cumprem os requisitos mínimos mencionados anteriormente, sendo que em termos de usabilidade o Bareos substancialmente melhor.

Ressalve-se que para o desenvolvimento deste projeto o objetivo não é efetuar cópias de segurança para armazenamento na nuvem.

## 2.5 Outras tecnologias

Ao longo desta dissertação tornou-se evidente para o autor que determinadas tecnologias deveriam ser usadas. No sentido de orientar o leitor apresentam-se aqui sem grande análise estas tecnologias.

### 2.5.1 Acesso Remoto

O acesso remoto é uma tecnologia que permite uma pessoa e ou dispositivo aceder a um destino díspar do ponto de origem. Pode ocorrer ou não via internet.

Esta tecnologia pode trazer diversas vantagens como por exemplo, filiais estarem em constante comunicação com a sede, acesso a pastas partilhadas, entre outros.

A principal vantagem que os administradores de sistemas tiram do acesso remoto são os custos economizados das deslocações para o destino. Desta forma não é necessária a deslocação deste, uma vez que é possível a sua simulação no destino (Pagés S., 2014).

Uma possível forma de se tirar vantagem do acesso remoto é através de um *modem* 3G. Para que isso seja exequível é necessário a criação de um endereço dinâmico uma vez que até à data o autor desta dissertação desconhece *modem's* 3G com IP – *Internet Protocol* fixo.

#### 2.5.1.1 Sakis

O Sakis[5] é um *software* que permite auxiliar o processo de instalação de um *modem* 3G. Foi especialmente criado para o uso deste em Raspberry's Pi. É um *script* que auxilia o processo de configuração da banda larga (Sakis).

Um dos requisitos para se poder instalar este *script* é o pacote PPP – *Point-to-Point Protocol*[6]. Este pacote serve para permitir a comunicação de dados entre dois nós distintos (Rouse M., 2016).

Até à data não foi encontrado mais nenhum *script* similar para efetuar esta tarefa.

#### 2.5.1.2 Endereçamento fixo e dinâmico

Quando um *modem* ou *router* se conecta à internet irá obter um endereço de IP. Este pode ser fixo ou dinâmico. Normalmente aos fixos acresce uma taxa na fatura mensal da operadora de internet.

Caso não se tenha um IP fixo, este irá alterar dinamicamente. Contudo é possível a adição de uma funcionalidade que permite a atualização do endereço de IP dinâmico para um determinado nome, ou seja, é possível aceder por um nome definido na criação do *dynamic* DNS, por exemplo **raspvw.ddns.net** , ao dispositivo mesmo que este altere o seu endereço IP.

### 2.5.2 Ferramentas de automatização

As DevOps como já referido anteriormente surgiram na sequência de melhorar a agilidade das *releases* no sector das Tecnologias de Informação. Estas focam o aperfeiçoamento da integração, colaboração e comunicação entre a equipa de desenvolvimento de *software* e administradores de sistemas.

Após recolha de opiniões de colegas de trabalho e professores, o autor decidiu efetuar uma breve análise a DevOps representativas. O Puppet, Chef e Ansible.

[5] http://www.sakis3g.com/
[6] https://www.rfc-editor.org/info/rfc1172

#### 2.5.2.1 Puppet

Criado em 2005, é O*pen Source* e está disponível para *download* gratuito sob a licença Apache 2.0. Tem em paralelo uma versão não gratuita chamada Puppet Enterprise, comercializada pela Puppet Labs.

O seu desenvolvimento é efetuado em Ruby. Possuí uma arquitetura de cliente/servidor, apesar de também ser possível utilizar o Puppet independentemente, via Puppet Apply.

O fluxo de dados funciona da seguinte forma como podemos verificar na imagem seguinte (Hillebrandt T., 2015).

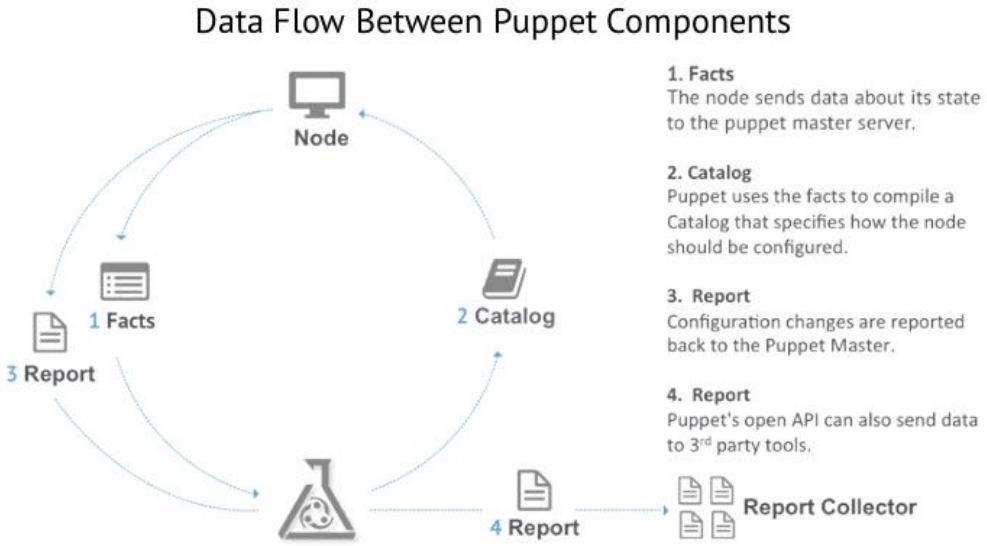



Figura 21 - Fluxo de dados no Puppet
(Hillebrandt T., 2015)

O cliente envia os dados coletados em formato YAML para o servidor. Seguidamente, o servidor vai compilar num *catalog* os dados recebidos e vai enviar o *catalog* para o cliente. O cliente ao receber o *catalog* irá executar o que está descrito nele, enviando um relatório com as mudanças realizadas para o servidor. O servidor também pode gerar relatórios externos através de APIs externas como por exemplo o Twitter.

#### 2.5.2.2 Chef

Esta ferramenta está disponível desde 2009 e oferece suporte para várias plataformas como por exemplo Ubuntu, Mac OS X e Windows. À semelhança do Puppet tem por base o Ruby (Duvall P., 2012).

A sua arquitetura tem por base três componentes que comunicam entre si como podemos apurar na imagem seguinte.

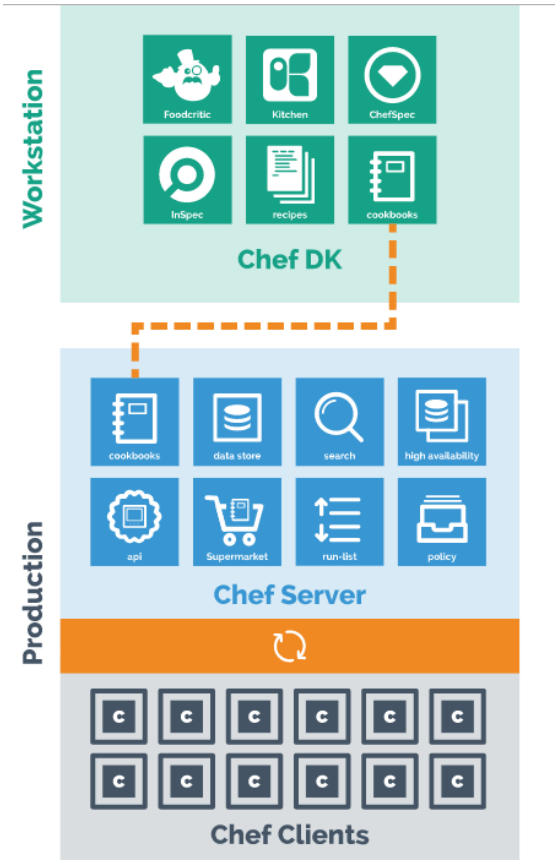



Figura 22 – Arquitetura do Chef (Chef)

O componente *Chef Development Kit* é executado numa máquina. Fornece as ferramentas necessárias para o desenvolvimento e teste de código antes de o colocarmos em produção.

O componente *Chef Server* é o repositório central do sistema, comunica com os *nodes* que executam o *Chef Client* para manusearem a infraestrutura.

Por fim, o último integrante desta infraestrutura, *Chef Client*, está instalado nos *nodes*. Estes *nodes* interpretam a política definida pelo *Chef Server* e alteram o seu estado mediante uma solicitação.

#### 2.5.2.3 Ansible

O Ansible é uma ferramenta semelhante ao Chef. Foi desenvolvido por Michael DeHaan em fevereiro de 2012 (DeHaan M., 2013). Segundo Daniel Bruno, a comunidade de administradores de sistemas e programadores que utilizam esta ferramenta tem crescido gradualmente.

Na figura seguinte está representada a arquitetura do Ansible.

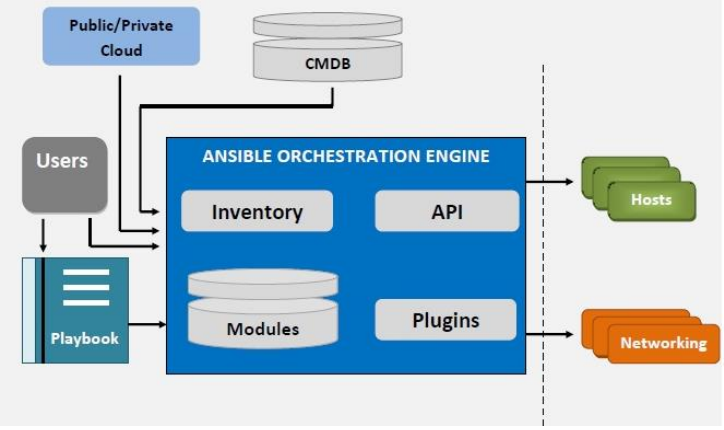



Figura 23 – Arquitetura do Ansible
(Seshachala S., 2015)

Algumas características que devem serem realçadas são o *Playbook*, *Modules* e *Inventory*.

O *Playbook* é o repositório onde ficam alojados os arquivos de gestão. É onde se define o perfil dos servidores e as respetivas tarefas a serem executadas. O seu formato é em YAML.

Os módulos podem ser escritos em linguagem *script*. São utilizadores para realizar diferente tipos de tarefa como por exemplo o teste da conetividade remota aos servidores.

O inventário é uma descrição dos *nodes* que são acedidos pela ferramenta.

#### 2.5.2.4 Conclusão

Da análise da informação anterior constata-se de mau grado as potencialidades das DevOps. No ambiente objeto desta dissertação não é exequível a sua utilização por falta dos recursos necessários no Raspberry Pi (Chef System Requirements, Ansible Installation, Puppet System Requirements).

### 2.5.3 Postfix

O Postfix é um servidor de entrega de e-mails de código aberto. Surgiu como alternativa ao SendMail. Foi originalmente escrito por Wietse Venema em 1997 no Centro de Pesquisa IBM Thomas J. Watson e foi lançado oficialmente em 1998 (4Linux).

Este servidor de e-mail permite ser usado como SMTP - *Simple Mail Transfer Protocol relay* (um *relay* é o envio de e-mail usando um serviço de e-mail externo) de e-mail.

# 3 Descrição técnica

No presente capítulo, o autor pretende demonstrar a arquitetura da solução e subjacentemente a respetiva implementação ao leitor.

## 3.1 Arquitetura da Solução

Neste subcapítulo irá ser demonstrada a arquitetura final da solução. Foi imposto pela Visionware que a solução terá de ser a menos dispendiosa possível, tendo sido, contudo, dado algumas ferramentas de trabalho para a execução do objetivo. Posto isto, o autor para atingir as metas traçadas pela Visionware optou pela instalação de um *software* de monitorização alarmística e de cópias de segurança, *Open Source*.

O *hardware* a utilizar será um Raspberry PI. É um pequeno computador que se conecta a um monitor. Foi desenvolvido no Reino Unido pela Fundação Raspberry PI e começou a ser comercializado no início de 2012 (Raspberry Pi Foundation). Este permite ser utilizado para os mesmos fins que um computador tradicional, apesar as características serem inferiores ao nível de processador e memória RAM.

As grandes vantagens deste dispositivo face aos computadores normais são as seguintes:

- Preço do produto;
- Tamanho, facilidade de transporte;
- Integração de *hardware* em placa única;
- Disponibilização de sistemas operativos por parte da Raspberry;
- Ajuda alargada devido a grande afluência de pessoas a utilizarem o mesmo dispositivo.

Existe atualmente duas gerações de Raspberry PI, a mais recente saiu em fevereiro de 2015 (Raspberry Pi 2 release). Entre estas duas gerações optamos pela segunda, pois como podemos observar na tabela seguinte, as características são melhores.

Tabela 7 - Comparação entre o Raspberry Pi 1 e Raspberry Pi 2

| Característica | Raspberry PI 1 | Raspberry PI 2 |
|---|---|---|
| **Processador** | 700MHz single-core ARM11 Broadcom BCM2835 CPU | 900MHz quad-core ARM Cortex-A7 Broadcom BCM2836 CPU |
| **RAM** | 512MB | 1GB |
| **SD Card** | Contém | Contém |
| **USB** | Contém | Contém |
| **HDMI** | Contém | Contém |

No computador iremos proceder à instalação de uma aplicação para monitorização e uma outra para efetuar cópias de segurança como podemos ver na figura seguinte.

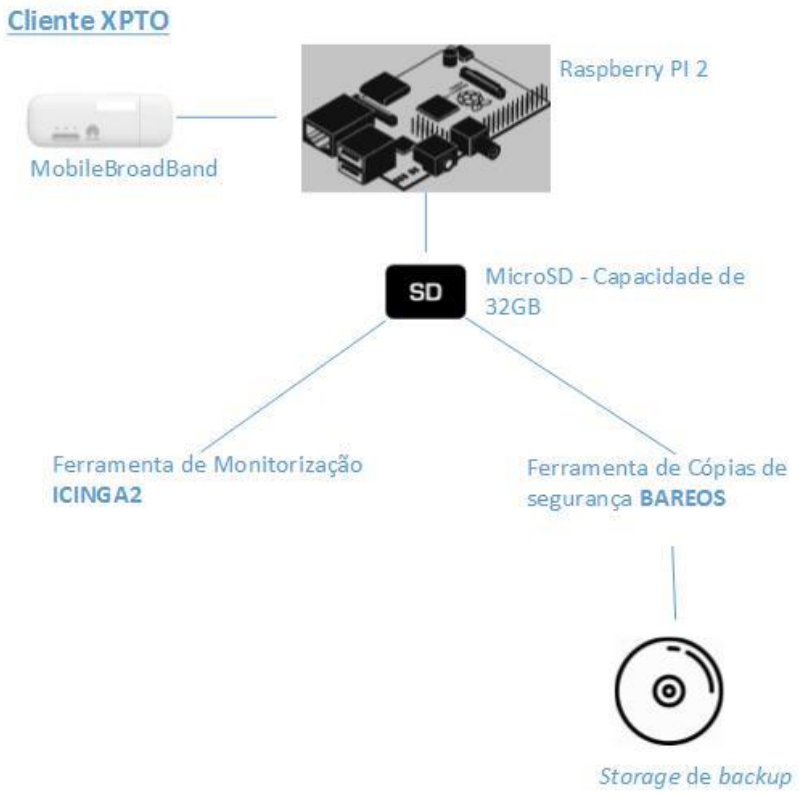



Figura 24 - Desenho para implementação da Wonderbox

A ideia é tirar partido da vantagem do Raspberry PI ser economicamente acessível, instalar aplicações *Open Source* em um cartão microSD. A incorporação de uma bandalarga 3G/4G irá permitir através de um sistema de resolução dinâmica de nomes desligar o equipamento. Está previsto um acordo com uma operadora para o fornecimento gratuito dos dispositivos para o acesso à internet.

Na figura ainda está associado à ferramenta de cópias de segurança uma *storage*. Este armazenamento terá de ser dimensionado para cada cliente já que nem todos tem a mesma quantidade de informação que necessitem de serem guardadas.

O sistema operativo a instalar optou-se pelo Ubuntu Mate 16.04 por conter uma interface gráfica amigável e intuitiva.

Ao longo do tempo foi possível aumentar a complexidade do produto como se pode ver na figura nº 25.

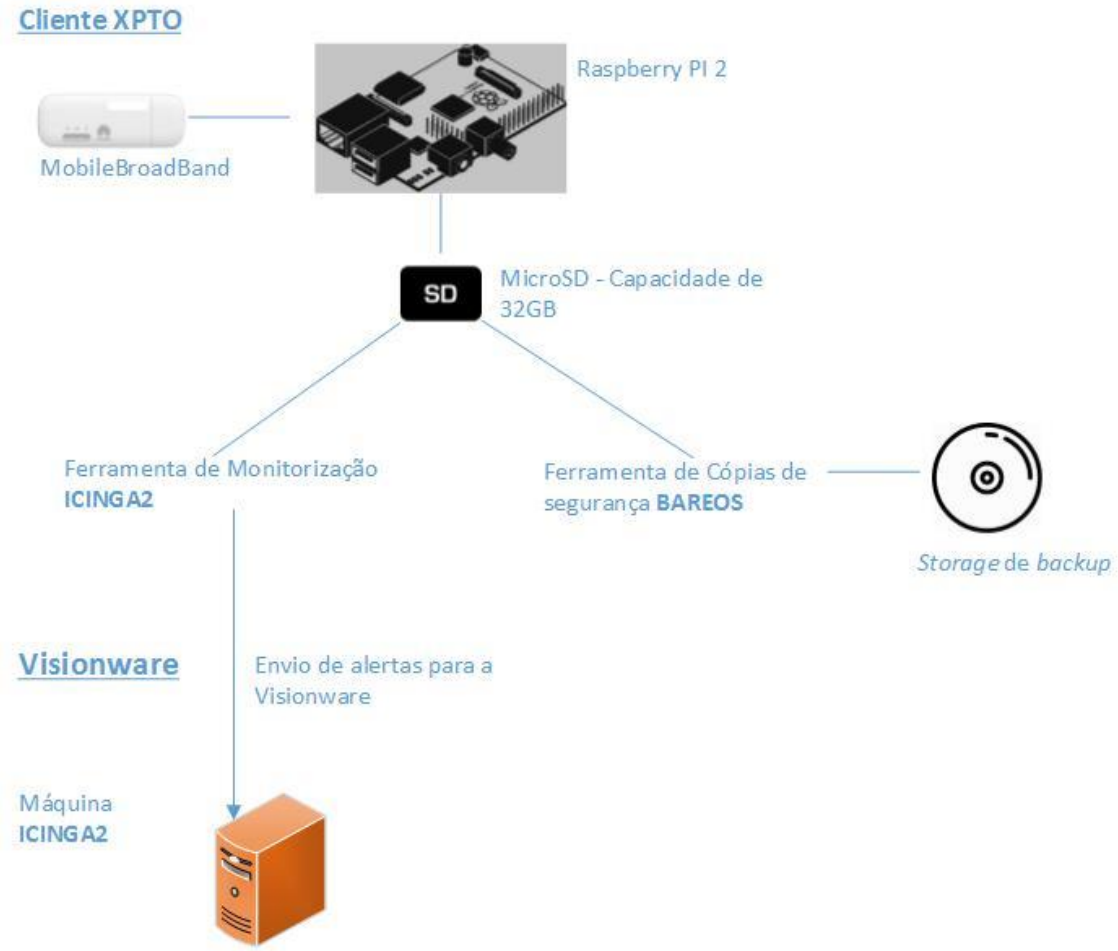



Figura 25 - Desenho para implementação da Wonderbox com solução distribuída

Desta forma foi utilizado o Icinga 2 como ferramenta de monitorização, pois este, como já exposto, tem capacidade de replicar dados entre o satélite (agente) e a consola central que

será uma máquina virtual com o Icinga 2. Assim, é possível ter um ecrã e receber os avisos de monitorização dos clientes. O resto das configurações mantém-se como na figura 24.

Para ferramenta de cópias de segurança optou-se pelo Bareos.

### 3.1.1 Diagrama de casos de uso

Para descrever melhor as funcionalidades do sistema, o autor desta dissertação apresenta diagramas de casos de uso. Um caso de uso é uma interação entre um ator e o sistema. Neste subcapítulo também terá uma breve descrição dos casos de uso.

A figura seguinte indica-nos as interações entre a Visionware, mais concretamente um membro da equipa de suporte e o sistema.

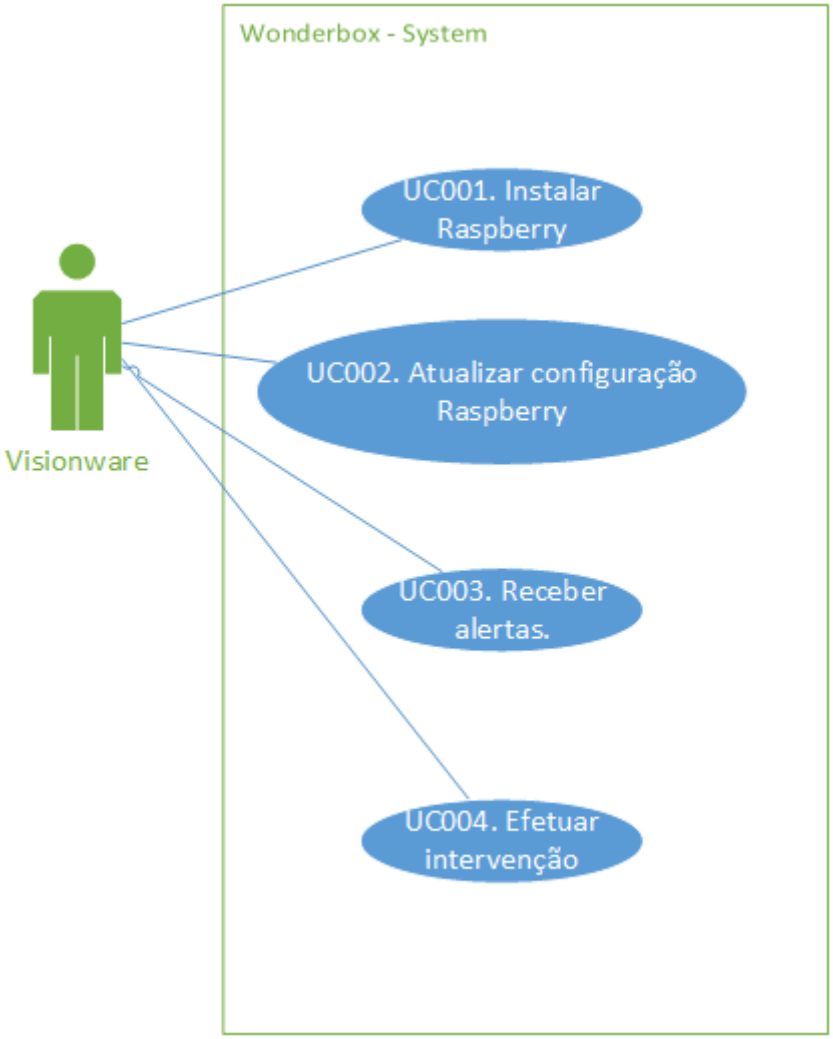



Figura 26 - Casos de uso Visionware

O ator "Visionware" terá 4 interações com o sistema. Instalar Raspberry, é a instalação de *software* e toda a configuração inicial do dispositivo. Atualizar configuração Raspberry, é a

remodelação de uma ligação ou configuração. Receber alertas, é o envio de alertas por parte do sistema para o ator e por fim, o último caso de uso, efetuar intervenção. O ator corrige o mau funcionamento do sistema.

O diagrama seguinte mostra-nos a interação entre o ator "Cliente" e o sistema.

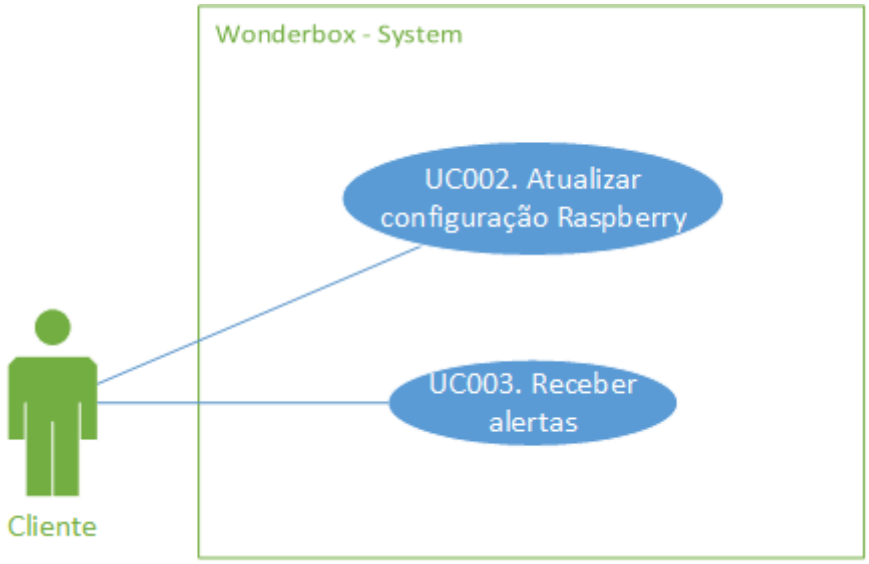



Figura 27 - Casos de uso Cliente

Este ator terá duas interações com o sistema. Receber alertas, ou seja, o envio de alertas por parte do sistema para o cliente. Atualizar configuração Raspberry, é a remodelação de uma ligação ou configuração.

Para o leitor ficar mais elucidado decidiu-se fazer uma breve descrição dos casos de uso. Posto isto, a seguinte tabela descreve o funcionamento do caso de uso "Instalar Raspberry".

Tabela 8 - Caso de uso – Instalar Raspberry

| **Número do Caso de Uso** | UC001 |
|---|---|
| **Nome do Caso de Uso** | Instalar Raspberry |
| **Ator(es)** | Visionware |
| **Descrição** | Este caso de uso tem por objetivo instalar o Raspberry. |
| **Cenário** | 1. O ator efetua a instalação do sistema operativo no Raspberry.<br>2. O ator instala o sistema de monitorização no Raspberry.<br>3. Parametrização dos servidores a monitorizar.<br>4. Parametrização dos serviços a monitorizar.<br>5. O ator instala o sistema de cópias de segurança.<br>6. Configuração do armazenamento.<br>7. Parametrização dos dados do cliente.<br>8. Configuração da periodicidade das cópias de segurança. |

| | |
|---|---|
| **Inclusão** | - |
| **Extensões** | - |

A seguinte tabela descreve o funcionamento do caso de uso “Atualizar configuração Raspberry”. De ressalvar neste caso de uso, que a profundidade da ação “Atualizar configuração Raspberry” vai ser distinta entre o ator Visionware e o ator Cliente.

Tabela 9 - Caso de uso - Atualizar configuração Raspberry

| | |
|---|---|
| **Número do Caso de Uso** | UC002 |
| **Nome do Caso de Uso** | Atualizar configuração Raspberry |
| **Ator(es)** | Visionware e Cliente |
| **Descrição** | Este caso de uso tem por objetivo a atualização das configurações do Raspberry. |
| **Cenário** | 1. O ator identifica que é necessário a atualização das configurações do Raspberry.<br>2. O ator atualiza as configurações. |
| **Inclusão** | - |
| **Extensões** | - |

Por sua vez, a seguir temos a tabela que descreve o funcionamento do caso de uso “Receber alertas”.

Tabela 10 - Caso de uso - Receber alertas

| | |
|---|---|
| **Número do Caso de Uso** | UC003 |
| **Nome do Caso de Uso** | Receber alertas |
| **Ator(es)** | Visionware e Cliente |
| **Descrição** | Neste caso de uso o ator recebe um alerta do sistema. |
| **Cenário** | 1. O ator recebe um alerta vindo do sistema. |
| **Inclusão** | - |
| **Extensões** | - |

Em último, a tabela seguinte mostra o funcionamento do caso de uso “Efetuar intervenção”.

Tabela 11 - Caso de uso - Efetuar intervenção

| | |
|---|---|
| **Número do Caso de Uso** | UC004 |
| **Nome do Caso de Uso** | Efetuar intervenção |
| **Ator(es)** | Visionware |
| **Descrição** | Este caso de uso tem por objetivo a realização de uma intervenção no sistema. |
| **Cenário** | 1. O sistema remete um aviso de uma anomalia para o ator.<br>2. O ator identifica a origem do problema.<br>3. O ator corrige o problema. |

| | |
|---|---|
| **Inclusão** | - |
| **Exclusão** | - |

Resumidamente, importa salientar que os casos de uso “Receber alertas” e “Atualizar configuração Raspberry” são ações cujas pertencem aos atores Visionware e Cliente.

## 3.2 Implementação

Para ser possível avaliar a solução e devido a questões alheias ao próprio autor desta dissertação, foi necessário que a implementação fosse realizada por outros colegas com base em *appointements* do autor desta dissertação.

Esta prova de conceito foi realizada num parque informático que simula uma realidade empresarial e se enquadra nos contributos que a *Wonderbox* acrescenta. Com base nas configurações genéricas da prova de conceito é possível ajusta-las a outras infraestruturas.

Pretendia-se efetuar testes de *stress* e escalabilidade para se analisar os limites do sistema *Wonderbox* e o seu possível crescimento aumentando o número de parques informáticos onde o sistema seria implementado. A ideia seria a medição de alguns atributos como por exemplo o tempo de resposta, medição do desempenho sobre carga avultada, uso do CPU, entre outros. Mas mais uma vez, estes testes não foram possíveis de serem efetuados devido a fatores externos que conduziram a uma carência de tempo.

### 3.2.1 Ambiente de Desenvolvimento e Teste

Por questões de confidencialidade optou-se por atribuir o nome XPTO ao parque informático da empresa onde se configurou a prova de conceito. Decidiu-se igualmente não desmitificar outros dados e/ou servidores que possam colocar em causa a integridade do negócio.

A figura seguinte dá-nos uma perspetiva do parque informático da empresa XPTO.

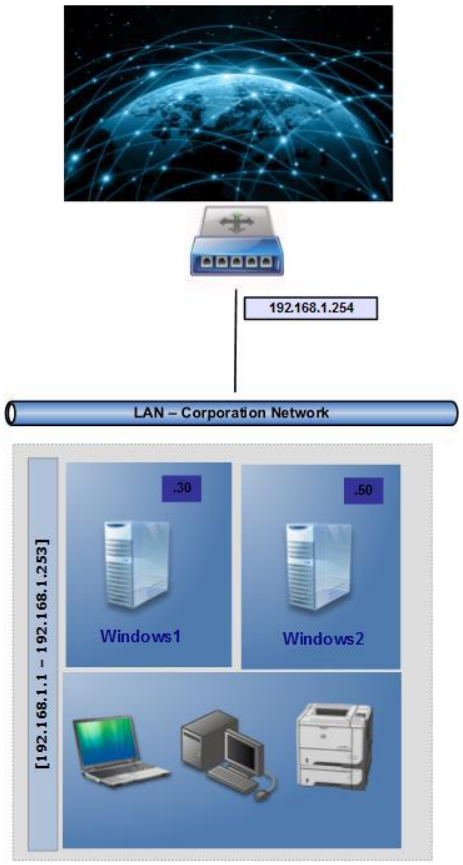



Figura 28 - Esquema de rede da Empresa XPTO

Como podemos observar na figura anterior, a LAN – *Local Area Network* engloba a rede de trabalho dos utilizadores.

A infraestrutura também engloba um *Router* que dá acesso à internet.

Em termos de serviços a implementar para a prova de conceito, o autor optou pelos seguintes que podem ser observados na tabela subsequente:

Tabela 12 - Servidores e serviços

| **Máquina** | **Serviço(s)** |
|---|---|
| Windows1 | Disponibilidade, Carga de CPU, Memória Física e Capacidade de Disco. |
| Windows2 | Disponibilidade, Carga de CPU, Memória Física e Capacidade de Disco. |
| VW-Rasp | HTTP, SSH e Login. |

Tabela 13 - Descrição dos serviços

| Serviço | Descrição |
|---|---|
| Disponibilidade | Analisa a disponibilidade da máquina através do ping. |
| Carga de CPU | Verifica a carga do processador. |
| Memória Física | Verifica a RAM física a ser usada. |
| Capacidade de Disco | Monitoriza o armazenamento a ser utilizado pelo disco. |
| HTTP | Verifica se a porta 80 http está aberta. |
| SSH | Analisa se a porta 22 SSH está disponível. |
| Login | Monitoriza quantas pessoas estão ligadas à máquina. |

Posta esta infraestrutura posicionou-se a *Wonderbox* também na LAN, com o endereço de IP 192.168.1.28, pois este deve ter acesso à comunicação com todo o meio envolvente.

De salientar que implementação serve como prova de conceito, sendo posteriormente necessário realizar os ajustes necessários a que sejam satisfeitas as necessidades das infraestruturas futuras.

### 3.2.2 Monitorização Alarmística Distribuída

Como já falado anteriormente, a ferramenta da monitorização ao qual se optou pela a sua instalação foi o Icinga2. Desta forma é possível termos satélites em localizações geográficas distintas a comunicarem com um serviço central, um servidor.

Para que logremos adicionar um satélite a um servidor é necessário existir uma comunicação entre os portos TCP 5665[7] em ambas as direções.

A ilustração seguinte ajuda a demonstrar a esquematização da comunicação entre o satélite e o servidor genericamente. Neste caso não é necessário efetuar-se *port forwarding* no *router* uma vez que o *link* da operadora está configurado na *firewall*.

[7] Considerado por omissão

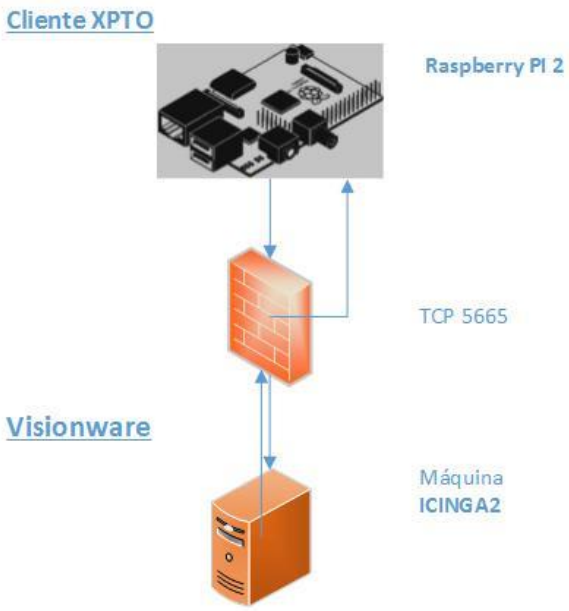



Figura 29 - Comunicação entre cliente e servidor

Devido a questões que impossibilitaram colocar a comunicação entre o cliente XPTO e a Visionware em funcionamento, para a prova de conceito colocou-se o servidor na mesma LAN que a *Wonderbox*.

Depois de a comunicação estar a funcionar, adiciona-se um nó ao servidor e ao cliente, neste caso um Raspberry.

Existem duas maneiras de adicionarmos um nó no Icinga2. Através da modificação manual dos ficheiros que servem para tal ou através de um *wizard* já embutido no sistema de monitorização.

#### 3.2.2.1 Servidor / Master

O excerto de código seguinte indica-nos como se adiciona um nó no Icinga2. Aqui, é possível indicar se o nó é *master*, ou seja, servidor, ou um satélite.

```
# Inicialização do wizard para a criação de um nó no Icinga2
root@srvwonderbox:~# icinga2 node wizard
Welcome to the Icinga 2 Setup Wizard!

We'll guide you through all required configuration details.

Please specify if this is a satellite setup ('n' installs a master setup)
[Y/n]: n
```

Código 1 – Adição de nó master

Se o administrador de sistemas introduzir “n”, automaticamente será despoletado um tutorial para a configuração do *master*. Este irá pedir o CN - *Common Name, bind host* e *bind port.*

Caso não se modifiquem estas opções, o Icinga2 automaticamente assume configurações por padrão como foi o caso seguinte.

Este passo criará também automaticamente o certificado para que a comunicação seja fidedigna com o *master*, cria um objeto do tipo zona com o nome do seu FQDN - *Fully Qualified Domain Name* e com o atributo endereço IP.

```
Please specify if this is a satellite setup ('n' installs a master setup)
[Y/n]: n
Starting the Master setup routine...
Please specifiy the common name (CN) [srvwonderbox]:
information/base: Writing private key to '/var/lib/icinga2/ca/ca.key'.
information/base: Writing X509 certificate to '/var/lib/icinga2/ca/ca.crt'.
information/cli: Initializing serial file in
'/var/lib/icinga2/ca/serial.txt'.
information/cli: Generating new CSR in '/etc/icinga2/pki/srvwonderbox.csr'.
information/base: Writing private key to '/etc/icinga2/pki/
srvwonderbox.key'.
information/base: Writing certificate signing request to '/etc/icinga2/pki/
srvwonderbox.csr'.
information/cli: Signing CSR with CA and writing certificate to
'/etc/icinga2/pki/ srvwonderbox.crt'.
information/cli: Copying CA certificate to '/etc/icinga2/pki/ca.crt'.
information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.
information/cli: Created backup file '/etc/icinga2/zones.conf.orig'.
Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:
information/cli: Enabling the APIlistener feature.
Enabling feature api. Make sure to restart Icinga 2 for these changes to
take effect.
information/cli: Created backup file '/etc/icinga2/features-
available/api.conf.orig'.
information/cli: Updating constants.conf.
information/cli: Created backup file '/etc/icinga2/constants.conf.orig'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
Please edit the constants.conf file '/etc/icinga2/constants.conf' and set a
secure 'TicketSalt' constant.
Done.

Now restart your Icinga 2 daemon to finish the installation!
```

Código 2 – Configurações para servidor

Depois deste processo é necessário a criação de uma chave PKI – *Public Key Infrastructure* que será introduzida posteriormente no cliente. Esta chave ou *token* permite a credibilidade entre a comunicação do cliente com o servidor e pode ser pedida da forma seguinte.

```
# Pedido do token PKI com o nome host Raspberry PI
root@srvwonderbox:~# icinga2 pki ticket --cn 'vw-rasp'
3d1c299bb100e30842fbfdc5777658dfafdfd778
```

Código 3 – Pedido de *token*

#### 3.2.2.2 Cliente / Satélite

No cliente / satélite o inicio processo de configuração do nó é semelhante como podemos ver no código seguinte.

```
# Inicialização do wizard para a criação de um nó no Icinga2
root@vw-rasp:~# icinga2 node wizard
Welcome to the Icinga 2 Setup Wizard!

We'll guide you through all required configuration details.

Please specify if this is a satellite setup ('n' installs a master setup)
[Y/n]:
```

Código 4 – Adição de nó satélite

A diferença reside na não colocação de um "n". Desta forma ele automaticamente inicia o processo de configuração de um satélite. A configuração do satélite pede ao utilizador o CN, endereço IP do *master* e também a sua *binding port,* neste caso a TCP 5665.

Como já visto anteriormente, é necessário então a introdução do *token* para a comunicação com o servidor ser fidedigna. Após este processo, o *wizard* irá automaticamente gerar os ficheiros de configuração.

```
Starting the Node setup routine...
Please specifiy the common name (CN) [vw-rasp]:
Please specifiy the local zone name [vw-rasp]:
Please specify the master endpoint(s) this node should connect to:
Master Common Name (CN from your master setup): srvwonderbox
Please fill out the master connection information:
Master endpoint host (optional, your master's IP address or FQDN):
192.168.1.94
Master endpoint port (optional) []:
Add more master endpoints? [y/N]
Please specify the master connection for CSR auto-signing (defaults to
master endpoint host):
Host [192.168.1.94]:
Port [5665]:
information/base: Writing private key to '/etc/icinga2/pki/vw-rasp.key'.
information/base: Writing X509 certificate to '/etc/icinga2/pki/vw-
rasp.crt'.
information/cli: Generating self-signed certificate:
information/cli: Fetching public certificate from master (192.168.1.94,
5665):
information/cli: Writing trusted certificate to file
'/etc/icinga2/pki/trusted-master.crt'.
information/cli: Stored trusted master certificate in
'/etc/icinga2/pki/trusted-master.crt'.

Please specify the request ticket generated on your Icinga 2 master.
 (Hint: # icinga2 pki ticket --cn 'vw-rasp'):
3d1c299bb100e30842fbfdc5777658dfafdfd778
information/cli: Processing self-signed certificate request. Ticket
'3d1c299bb100e30842fbfdc5777658dfafdfd778'.

information/cli: Writing signed certificate to file '/etc/icinga2/pki/vw-
rasp.crt'.
information/cli: Writing CA certificate to file '/etc/icinga2/pki/ca.crt'.
```

```
Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:
information/cli: Disabling the Notification feature.
Disabling feature notification. Make sure to restart Icinga 2 for these
changes to take effect.
information/cli: Enabling the Apilistener feature.
Enabling feature api. Make sure to restart Icinga 2 for these changes to
take effect.
information/cli: Created backup file '/etc/icinga2/features-
available/api.conf.orig'.
information/cli: Generating local zones.conf.
information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.
information/cli: Created backup file '/etc/icinga2/zones.conf.orig'.
information/cli: Updating constants.conf.
information/cli: Created backup file '/etc/icinga2/constants.conf.orig'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
Done.

Now restart your Icinga 2 daemon to finish the installation!
```

Código 5 – Configurações para cliente

Importa referir que o passo anterior modificou automaticamente o ficheiro **api.conf** para que este possa aceitar comandos e também o ficheiro **zones.conf** ao qual foi acrescentado a zona do satélite, o seu FQDN e o seu IP.

Uma vez completado ambos os *wizards* com sucesso, é possível através do servidor verificar os nós que ele tem acoplados, seja através da linha de comandos ou através da interface gráfica como podemos ver nas imagens seguintes.

```
root@srvwonderbox:/etc/icinga2# icinga2 node list
Node 'vw-rasp' (last seen: Thu Oct 13 12:39:50 2016)
 * Host 'Windows1'
        * Service 'disk'
        * Service 'load'
        * Service 'memory'
        * Service 'ping4'
 * Host 'Windows2'
        * Service 'disk'
        * Service 'load'
        * Service 'memory'
        * Service 'ping4'
 * Host 'vw-rasp'
        * Service 'http'
        * Service 'load'
        * Service 'ping4'
        * Service 'ping6'
        * Service 'procs'
        * Service 'ssh'
        * Service 'users'
```

Código 6 – Verificação de nó via linha de comandos

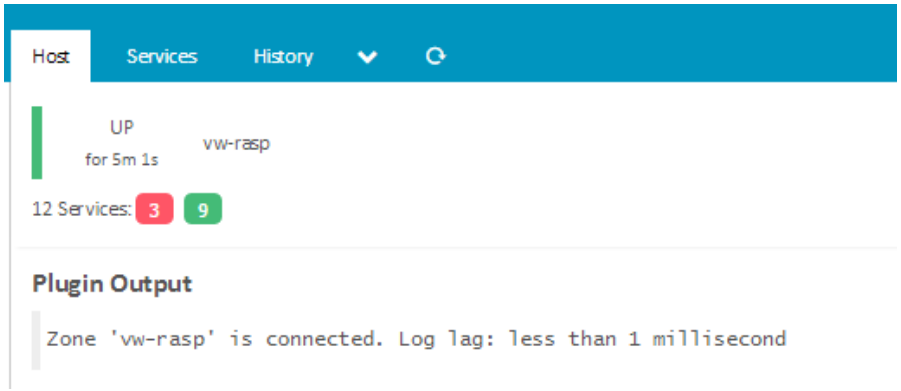



Figura 30 - Verificação de nó via interface gráfica

Por omissão, ao adicionarmos um nó, este automaticamente já vai ter a si associado alguns serviços de monitorização como por exemplo, a análise à disponibilidade da máquina, a carga do processador, entre outros.

#### 3.2.2.3 NSClient ++

Para podermos monitorizar máquinas que tem como base o sistema operativo Windows é necessário a instalação do NSClient++ ou do agente Icinga2[8]. Ambos os agentes permitem ao Windows interpretar os *plugins* que são executados no Icinga2.

Optamos pelo método do agente via NSClient++ que poderá ser descarregado através do endereço **http://nsclient.org/** e a sua instalação é trivial sendo necessário apenas ter em atenção a introdução do endereço IP do nosso cliente de monitorização, o Raspberry, e a ativação da interação do serviço NSClient++ com o ambiente de trabalho do sistema operativo Windows como podemos observar na imagem seguinte.

[8] http://packages.icinga.org/windows/

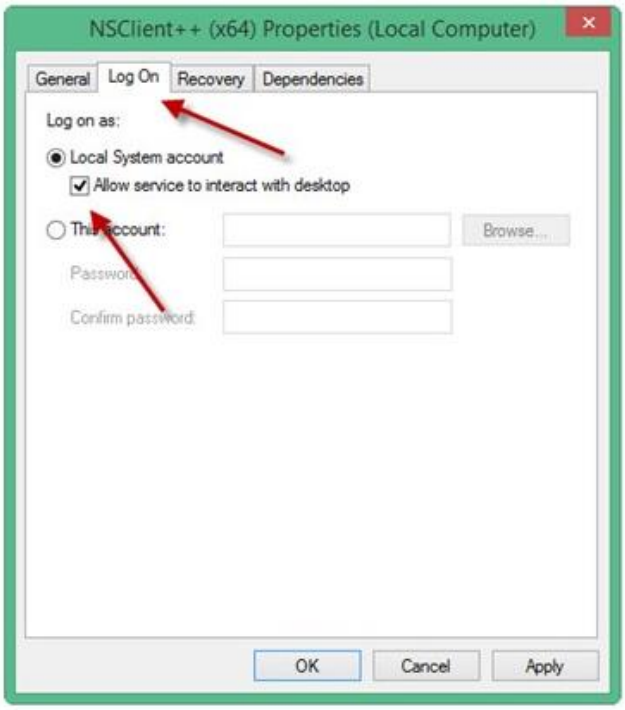



Figura 31 - Serviço NSClient++

A instalação do NSClient++ bem-sucedida deverá fazer com que o porto 12489 do Windows fique disponível para responder a pedidos.

#### 3.2.2.4 Hosts

Como já visto anteriormente, os *hosts* podem ser qualquer sistema que responda a SNMP e tenha conetividade com Raspberry.

Para a prova de conceito foram selecionadas duas máquinas com o sistema operativo Windows. A máquina Windows 1 e a máquina Windows 2. Desta forma, estas foram adicionados ao ficheiro **/etc/icinga2/conf.d/hosts.conf**.

```
object Host "Windows1" {
  import "generic-host"

  address = "192.168.1.30"
  check_command = "hostalive"

}
```

Código 7 – Criação do *host* Windows1

O excerto de código apresentado anteriormente, inicialmente cria um *host* com o nome Windows1, importa o *template generic-host*. Tem a si associado o endereço IP 192.168.1.30 e faz uma verificação da disponibilidade através do comando *hostalive*.

O *hostalive* através de pedidos ICMP[9] - *Internet Control Message Protocol* verifica se a máquina está *online* ou *offline*. Contudo o autor desta dissertação está consciente que nem sempre as máquinas respondem a estes pedidos, pelo que pode levar o Icinga2 ao erro.

Existem diversas formas de configurarmos os *hosts*, essas formas podem ser consultadas[10].

#### 3.2.2.5 Serviços

Existem vários formatos de se efetuar monitorização ao sistema operativo Windows[11]. Uma delas é editando o ficheiro **nsclient.ini** na pasta **C:\Program Files\NSClient++**, por forma a que este interprete comandos externos, criando *aliases,* para estes poderem ser interpretados pelos *plugins* no Icinga2.

Para a prova de conceito, o *plugin* escolhido para executar as *queries* foi o check_nrpe que opera no porto TCP 5666 do sistema operativo Windows.

Para tal editamos o ficheiro **/etc/icinga2/conf.d/commands.conf** por forma a que o Icinga2 possa interpretar os serviços de acordo com o *plugin* **/usr/lib/nagios/plugins/check_nrpe**.

```
object CheckCommand "check_nrpe" {
  import "plugin-check-command"

  command = [
    PluginDir + "/check_nrpe",
    "-H", "$address$",
    "-c", "$remote_nrpe_command$",
  ]
}
```

Código 8 – Criação do comando check_nrpe

O comando inicialmente importa o *template plugin-check-command*, procurando na diretoria *plugins* o *plugin* check_nrpe. Posto isto os parâmetros necessários a serem preenchidos nos serviços são o endereço de IP ou FQDN dá máquina e o *alias* inserido no ficheiro **nsclient.ini**. como podemos observador no excerto de código seguinte.

```
alias_mem = checkMem MaxWarn=80% MaxCrit=90% ShowAll=long type=physical
```

Código 9 – Criação do *alias* no ficheiro nsclient.ini

Este *alias* irá fazer uma verificação à memória da máquina Windows, irá despoletar um aviso *Warning* quando esta atingir os 80% e um aviso a *Critical* quando esta atingir os 90%. A memória por sua vez que pretendemos analisar é a memoria física.

[9] Serve para testar conetividade entre equipamentos
[10] http://docs.icinga.org/icinga2/latest/doc/module/icinga2/chapter/monitoring-basics
[11] http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/configuring-icinga2-first-steps#services-conf

Posto isto é necessário fazer a ponte entre o serviço Icinga2 a monitorizar a memória e o *alias* no ficheiro **nsclient.ini**. O autor achou conveniente colocar os serviços no ficheiro **/etc/icinga2/conf.d/services.conf**.

```
object Service "memory" {
  import "generic-service"

  host_name = "Windows1"
  check_command = "check_nrpe"
  vars.remote_nrpe_command = "alias_mem"
}
```

Código 10 – Criação do serviço *memory*

O pedaço de código anterior representa a monitorização de um serviço. Como podemos observar, este serviço irá monitorizar o *host* Windows1 criado na secção 3.2.2.4. Utilizará o comando check_nrpe criado neste subcapítulo. Aqui, faz-se referência ao alias_mem do Windows criado no ficheiro **nsclient.ini**.

Contudo, como já foi referido, esta é apenas um modo de configurarmos o serviço existindo inúmeras formas de o fazer[12].

#### 3.2.2.6 Notificações

Foram definidos dois utilizadores para receberem as alertas, o utilizador **icingaadmin** e o utilizador **icingacliente**. Ambos os utilizadores fazem parte do grupo de administradores e tem associados os seus respetivos endereços de e-mail.

Os utilizadores definem-se no ficheiro **/etc/icinga2/users.conf** da seguinte forma:

```
object User "icingaadmin" {
  import "generic-user"

  display_name = "Visionware"
  groups = [ "icingaadmins" ]

  email = "wonderboxvw@gmail.com"
}

object User "icingacliente" {
  import "generic-user"

  display_name = "Cliente XPTO"
  groups = [ "icingaadmins" ]

  email = "stefanrodrigues10@hotmail.com"
}

object UserGroup "icingaadmins" {
  display_name = "Icinga 2 Admin Group"
```

12 http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/monitoring-basics#using-apply-services

```
}
```

Código 11 – Criação de utilizadores para notificações

Como vimos no subcapítulo *Hosts*, é no objeto *host* que definimos quais os utilizadores ou grupo de utilizadores que são notificados.

#### 3.2.2.7 Criação de utilizador para cliente

Foi adicionado um utilizador para o cliente. O utilizador tem o nome de **icingacliente**. Desta forma é possível rastrear através de *logs* qual foi o utilizador que fez determinada ação.

O utilizador insere-se através de uma *password hash*. Essa *hash* é adquirida usando MD5 baseado no algoritmo BSD.

Através do comando seguinte criamos a *hash* da palavra-passe.

```
# Criação da hash da palavra-passe
root@vw-rasp:/home/sgomes# openssl passwd -1 cliente
$1$uPbBDHyp$uus7IhZeifyzdMn/5kSYF.
```

Código 12 – Criação de *hash* da palavra-passe

Seguidamente é necessário a introdução deste utilizador na base de dados icingaweb2.

```
# Inserção do utilizador icingacliente na base de dados da interface
gráfica. Active = 1 significa que o utilizador vai estar ativo.
root@vw-rasp:/home/sgomes# mysql -p icingaweb2
mysql> INSERT INTO icingaweb_user (name, active, password_hash) VALUES
('icingacliente', 1, '$1$uPbBDHyp$uus7IhZeifyzdMn/5kSYF.');
Query OK, 1 row affected (0,01 sec)

mysql> quit
Bye
```

Código 13 – Inserção do utilizador na BD

Após este processo é necessário a adição do utilizador **icingacliente** às permissões de administradores. Esta adição é uma imposição da organização onde se está a elaborar a presente dissertação de mestrado.

```
# Modificação do ficheiro roles.ini através do editor de texto vim
root@vw-rasp:/home/sgomes# vim /etc/icingaweb2/roles.ini
[admins]
users               = "icingaadmin, icingacliente"
permissions         = "*"
```

Código 14 – Adição do utilizador aos administradores

### 3.2.3 Cópias de Segurança

Como referido anteriormente a ferramenta para a execução das cópias de segurança escolhida foi o Bareos. Neste subcapítulo será efetuada uma prova de conceito num

computador com o sistema operativo Windows, configurando de raiz o agente, *storage* e o *job* de *backup*.

#### 3.2.3.1 Configuração de cliente Windows

Para a configuração de clientes Windows existe a necessidade da instalação de um agente no sistema operativo que faça a ligação com a *Wonderbox*. O seu *download* poderá ser efetuado em **http://download.bareos.org/bareos/release/latest/windows**.

A figura seguinte mostra os atributos necessários a preencher para que a comunicação entre o cliente e a *Wonderbox* seja um sucesso.

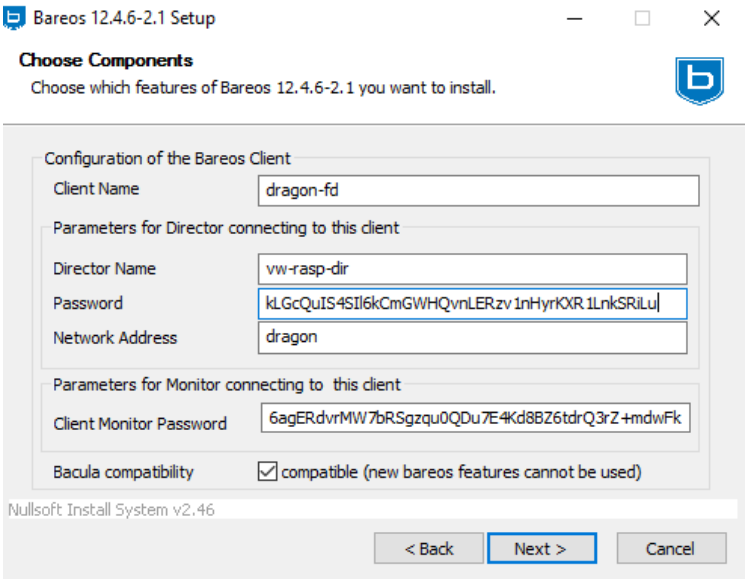



Figura 32 – Configuração cliente Bareos

É importante preencher os atributos *Director Name, Password* e *Network Address* de forma correta. Destes 3 à exceção do *Network Address* que poderá ser o FQDN ou o IP da máquina onde o agente está a ser instalado, é necessário copiar os atributos *name* e *password* do *Director* no ficheiro **/etc/bareos/bareos-dir.conf** do Raspberry.

A conclusão da configuração do cliente em ambiente Windows está exposta na Figura nº33.

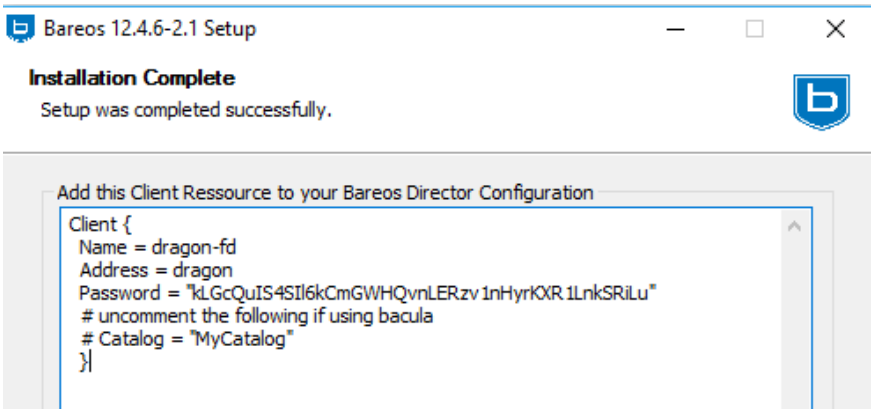



Figura 33 – Conclusão da configuração cliente Bareos

É necessário então copiar estas configurações para o ficheiro **/etc/bareos/bareos-dir.conf** do Raspberry.

Importa frisar que o agente Windows automaticamente vai criar uma exceção na *firewall* do Windows para que a comunicação possa existir.

#### 3.2.3.2 Configuração da storage para backup

Como já visto anteriormente a componente que controla o armazenamento para as cópias de segurança é a *storage deamon*. Esta componente tem o ficheiro configurável em **/etc/bareos/bareos-sd.conf**. Para a prova de conceito o objeto *Device* foi configurado da seguinte forma:

```
Device {
      Name = FileStorage
      Media Type = File
      Archive Device = /var/bareos_backup
      LabelMedia = yes;
      Random Access = yes;
      AutomaticMount = yes;
      RemovableMedia = no;
      AlwaysOpen = no;
}
```

Código 15 – Configuração do armazenamento de backup

A variável *Media Type* indica o nome do mídia a ser utilizado para a configuração da cópia de segurança. Deve ser do tipo “Rados” caso seja uma *pen* USB. No exemplo acima é do tipo *File* por a cópia ser efetuada para um ficheiro.

O *Archive Device*, explicita onde são gravados os dados das cópias de segurança. No exemplo acima são gravados no diretório **/var/bareos_backup**.

A variável *Random Access* ativa significa o uso da função Iseek[13]. Esta função permite operações aleatórias de escrita e leitura num diretório. Utiliza-se o valor *Yes* para escrita em diretórios e *pen's* USB e *No* para *tapes*.

*AutomaticMount* verifica se o dispositivo tem ficheiros Bareos. No exemplo acima utilizamos *Yes* para verificar se o armazenamento já contém ficheiros Bareos.

A varíavel *RemovableMedia*, usa-se para cópias de segurança com armazenamento em *tapes*. Acima a variável está definida a *No* uma vez que não estamos a utilizar *tapes*.

*AlwaysOpen* está definida a *No*, pois significa que o acesso ao *Media Type* não está sempre ativo. Apenas fica ativo quando uma tarefa é executada.

O *Device* é um objeto que pode ser definido de vários tipos. Todos estes tipos de configurações possíveis podem ser consultados[14].

#### 3.2.3.3 Configuração do *job* de backup

Para se configurar uma tarefa para a cópia de segurança é necessário parametrizar as configurações para que essa tarefa ocorra da forma como pretendemos. As parametrizações são definidas no objeto *JobDefs* que para serem consideradas tem de ser chamadas pelo objeto do tipo "*Job*". O *Director* para inicialização de uma tarefa de backup apenas consegue interpretar objetos do tipo "*Job*". Estas configurações são definidas no ficheiro **/etc/bareos/bareos-dir.conf**.

Para prova de conceito foram efetuadas as seguintes configurações para a execução de uma cópia de segurança.

```
JobDefs {
     Name = "WindowsDefJob"
     Type = Backup
     Level = Incremental
     Client = dragon-fd
     FileSet = "Windows"
     Schedule = "WeeklyCycle"
     Storage = File
     Messages = Notification
     Pool = Incremental
     Priority = 10
     Write Bootstrap = "/var/bareos/working/%c.bsr"
     Full Backup Pool = Full
     Differential Backup Pool = Differential
     Incremental Backup Pool = Incremental
}

Job {
     Name = "BackupDragon"
     JobDefs = "WindowsDefJob"
}
```

[13] http://www.rogercom.com/CursoOnlineLPT/Modulo03/Modulo003Aula006.htm
[14] http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-15400010.4

Código 16 – Definição do Job

Começamos pelo nome que é definido pela variável *Name*. Poderá ser atribuído arbitrariamente. Para o exemplo foi utilizado o nome "*WindowsDefJob*".

O *Type* define o tipo de trabalho, uma vez que no exemplo se pretende efetuar uma cópia, a variável foi definida com o tipo *backup*.

A variável *Level* define o tipo de cópia de segurança a ser executada. No exemplo acima pretende-se que seja incremental.

O *Client* define o *target* da cópia, neste exemplo a máquina que tem o agente instalado é o dragon-fd.

A variável *FileSet* uma vez que é considerada de extrema importância está detalhada no subcapítulo seguinte.

*Schedule* como o próprio nome indica, é o agendamento da hora/dia a que a cópia é realizada. Para esta prova de conceito está configurado uma cópia total no primeiro sábado de cada mês às 21h e depois cópias incrementais diárias de segunda-feira a sexta-feira às 21h.

É através da variável *Storage* que o *Director* fará a ponte para a *storage deamon*, ou seja, o armazenamento que está explicado em 3.2.3.2.

A *Pool* neste exemplo define apenas os tempos de retenção das cópias de segurança.

A *Priority*, em português prioridade, por omissão é 10. Quer isto dizer que caso tenha uma cópia de segurança com prioridade 10 e outra com 9 agendada para a mesma hora do mesmo dia, a que tem prioridade 9 será executada antes da que tem prioridade 10. Para a prova de conceito manteve-se a prioridade por omissão.

Na variável *Write Bootstrap* escolhemos a localização dos ficheiros que o Bareos cria automaticamente que servem para leitura quando é solicitado um restauro. Esses ficheiros contêm informação em ASCII sobre o volume onde foi armazenada a cópia de segurança e quais os ficheiros copiados.

*Full*, *Differential* e *Incremental Backup Pool*, estas variáveis não são necessárias no *job*, contudo são recomendadas e por omissão já estão colocadas nos exemplos disponibilizados pelo Bareos para evitar que a cópia não seja executada sem sucesso. A primeira cópia é sempre completa, pelo que se não tiver configurado por exemplo a *pool Full*, ele não irá executar a cópia de forma correta.

#### 3.2.3.3.1 FileSet

O *FileSet* define quais os ficheiros que devem ser incluídos ou excluídos de uma tarefa de *backups* e restauro, bem como várias opções como por exemplo, compressão, o uso de VSS por parte do Windows.

```
FileSet {
     Name = "Windows"
     Enable VSS = yes
     Include {
            Options  {
                 signature = MD5
                 WildFile = "*.exe"
                 exclude = yes

            }
            File = "C:/Users/Asus/Teste/"
      }
}
```

Código 17 – Definição do FileSet

A variável *Name* é o nome que atribuímos ao *FileSet*. O nome pode ser atribuído de acordo com a pretensão do administrador de sistemas.

A variável *Enable* VSS tem dois estados. *Yes* ou *No.* Se estiver em *Yes* utiliza o motor *Volume Shadow Copy Service* da Microsoft. Esta variável ativa apenas deve ser usada para fazer cópias aos sistemas operativos Windows.

A variável *WildFile* vai excluir para esta prova de conceito os ficheiros executáveis. É necessário também definir *exclude* = *yes*.

Por fim, a variável *File* define a pasta ou pastas onde vamos efetuar a cópia de segurança.

Existem uma série de configurações possíveis de colocarmos na definição de um *FileSet* que não são pertinentes para esta dissertação pelo que não são explanadas[15].

#### 3.2.3.4 Restauro das cópias de segurança

O restauro pode ser efetuado de duas formas. Através da interface gráfica clicando em "*Restore*" ou através da consola do *Director*.

```
Job {
     Name = "RestoreWindows"
     Type = Restore
     Client = dragon-fd
     FileSet = "Windows"
     Storage = File
     Pool = Incremental
     Messages = Notification
     Where = /tmp/bareos-restores
}
```

Código 18 – Definição do Job de restauro

A varíavel *Name* é o nome que atribuímos ao *job*.

O *Type* é o tipo de *job*, neste caso é restauro.

[15] http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-1320009.5

A variável *Client*, é o FQDN/IP da máquina onde está instalado o agente, neste caso para a prova de conceito utilizamos o dragon-fd.

O *FileSet* utilizado é o "Windows". Uma vez que a cópia utilizada utilizou o *FileSet* Windows, automaticamente teremos de utilizar o mesmo para fazer restauro da informação que foi copiada anteriormente.

A variável *Storage* aqui define para que tipo de mídia vai a informação ser restaurada, neste caso é do tipo *File* e vai para diretório na pasta **C:/tmp/bareos-restores**.

A *Pool* neste restauro vai verificar se a retenção do *job* ainda é válida.

Por fim a variável *Messages* são notificações, tem um nome arbitrário. Neste caso chama-se *Notification*. O funcionamento das notificações é explicado no subcapítulo seguinte.

#### 3.2.3.5 Notificações

As notificações via e-mail são de extrema importância para se saber que a tarefa da cópia de segurança foi executada de forma correta. Estas estão ligadas aos trabalhos de *backup*.

Para tal por omissão existe a componente *Messages* no ficheiro **/etc/bareos/bareos-dir.conf**. Esta componente utiliza o *script* **/bin/bsmtp**[16] para o envio de e-mails. Este por sua vez necessita de um serviço de e-mail configurado.

```
Messages {
  Name = Notification
  mailcommand = "/bin/bsmtp -h localhost -f \"\(Bareos\) \<%r\>\" -s
\"Bareos: %t %e of %c %l\" %r"
  operatorcommand = "/bin/bsmtp -h localhost -f \"\(Bareos\) \<%r\>\" -s
\"Bareos: Intervention needed for %j\" %r"
  mail = wonderboxvw@gmail.com = all, !skipped, !audit
  operator = wonderboxvw@gmail.com = mount
}
```

Código 19 – Definição das notificações

A variável *Name* que no caso tem o nome *Notification* não é fixa e pode ser definida de acordo com o nome pretendido pelo administrador de sistemas.

As variáveis *Mailcommand* e *Operatorcommand* tem de estar previamente definidas antes das variáveis *mail* e *operator*. *Mailcommand* e *Operatorcommand* vão utilizar os parâmetros definidos em *mail* e *operator* por forma a preencherem as *quoted strings* e enviarem a notificação através do *script* bsmtp.

O *Mailcommand* ocorre no fim da tarefa de backup ser executada e o *Operatorcommand* ocorre quando existe a necessidade de uma intervenção por exemplo na *storage*.

[16] http://manpages.ubuntu.com/manpages/xenial/man1/bsmtp.1.html

De referir que são nas variáveis *mail* e *operator* que definimos o nível de alertas que pretendemos que o sistema nos envie. Existem uma série de configurações exequíveis que podem ser consultadas[17].

#### 3.2.3.6 Criação de utilizador para cliente

Para que o cliente possa aceder à interface gráfica do Bareos por forma a proceder a algumas execuções de tarefas foi criado o utilizador **cliente**. Este utilizador poderá executar as tarefas acedendo à interface gráfica seguidamente *Director* e *Console*.

### 3.2.4 Acesso Remoto

Como já visto anteriormente o Sakis3G é um *software* que permite auxiliar o processo de instalação de um *modem* 3G. Para tal a figura seguinte explica os passos necessários para a sua instalação.

```
# Instalação do pacote Point-to-Point Protocol.
root@vw-rasp:/home/sgomes# sudo apt-get install ppp

# Download do software
root@vw-rasp:/home/sgomes# sudo wget
"http://www.sakis3g.com/downloads/sakis3g.tar.gz" -O sakis3g.tar.gz

# Descompressão do ficheiro sakis3g.tar.gz
root@vw-rasp:/home/sgomes# sudo tar -xzvf sakis3g.tar.gz

# Atribuição de permissões de execução ao ficheiro descomprimido sakis3g
root@vw-rasp:/home/sgomes# sudo chmod +x sakis3g

# Execução do ficheiro de forma interativa
root@vw-rasp:/home/sgomes# ./sakis3g –interactive
```

Código 20 – Configuração do Sakis3G

Uma vez que a banda larga a utilizar não dispõe de uma tecnologia com IP fixo, é necessário criar um *dynamic* DNS.

Para tal, é necessário registar uma conta (por exemplo em **http://www.noip.com/**) e efetuar os passos da figura seguinte.

```
# Download do ficheiro noip-duc-linux.tar.gz
root@vw-rasp:/home/pi/noip/# wget http://www.no-ip.com/client/linux/noip-
duc-linux.tar.gz

# Descompressão do ficheiro noip-duc-linux.tar.gz
root@vw-rasp:/home/pi/noip/# tar vzxf noip-duc-linux.tar.gz

# Navegação para a pasta noip-2.1.9-1
root@vw-rasp:/home/pi/noip/# cd noip-2.1.9-1
```

[17] http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-16500012

```
# Instalação do software. Aqui será pedido as credenciais de acesso à conta
criada anteriormente no no-ip.com e o tempo de verificação se o IP dinâmico
foi modificado.
root@vw-rasp:/home/pi/noip/noip-2.1.9-1# sudo make install
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
/usr/local/bin/noip2 -C -c /tmp/no-ip2.conf

Auto configuration for Linux client of no-ip.com.

Please enter the login/email string for no-ip.com  raspvw
Please enter the password for user 'raspvw'  ********

Only one host [raspvw.ddns.net] is registered to this account.
It will be used.
Please enter an update interval:[30]  30
Do you wish to run something at successful update?[N] (y/N)  n

New configuration file '/tmp/no-ip2.conf' created.
mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf
```

Código 21 – Configuração endereço dinâmico

Desta forma mesmo que o IP dinâmico se altere é sempre possível o acesso ao Raspberry Pi através do endereço **raspvw.ddns.net.**

### 3.2.5 Configuração do Postfix

O Postfix foi configurado por forma a fazer *relay* através do SMTP do Gmail. Este *relay* é usado pelo Icinga2 e pelo Bareos para notificações.

Para tal no ficheiro **/etc/postfix/main.cf** foram acrescentadas as seguintes variáveis.

```
# definir o smtp da Google
relayhost = [smtp.gmail.com]:587

#  uso de TLS pois a pois a porta 587 necessita de encriptação
smtp_use_tls=yes

# uso de sasl para autenticação SMTP externo
smtp_sasl_auth_enable = yes

# caminho para o ficheiro sasl
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd

# lista de certificados CA confiáveis
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

# elimina definições de segurança que não são compatíveis com o Gmail
smtp_sasl_security_options =
```

Código 22 – Alteração no ficheiro main.cf

É possível de se observar que o e-mail está a ser enviado através do SMTP do Gmail através da variável *relayhost*.

Uma vez que pretendemos que o envio do e-mail seja efetuado com autenticação e o servidor utilizado assim o exige, é necessário criar o ficheiro **/etc/postfix/sasl_passwd** com seguinte formato:

```
[smtp.gmail.com]:587  wonderboxvw@gmail.com:abc.1234
```

Código 23 – Formato do ficheiro sasl_passwd

Por fim atribuímos o *owner* do Postfix ao ficheiro **/etc/postfix/sasl_passwd** para o Postfix ter permissões de leitura sobre o ficheiro.

```
root@vw-rasp:/etc/postfix# chown postfix sasl_passwd*
```

Código 24 – Atribuição de permissões ao ficheiro sasl_passwd

De notar que as alterações indicadas só são assumidas quando se reinicia o Postfix.

# 4 Avaliação do Produto

Visto que os objetivos deste produto se prendem com um controlo à rede por parte dos administradores de sistemas por forma a diminuir os tempos de indisponibilidade de serviços ou ativos, fazendo com que estes ajam de forma mais acelerada e aumentado a probabilidade de recuperação dos dados dos utilizadores, foram consideradas as seguintes grandezas a avaliar:

- Tempos de indisponibilidade de serviços ou ativos;
- Reposição dos dados aos utilizadores;
- Tempo de RPO – *Recovery Point Objective* e RTO – *Recovery Time Objective*;

Os tempos de indisponibilidade de serviços ou ativos será possível medir efetuando uma comparação entre o cenário previamente à instalação do produto na organização e o cenário posterior à implementação.

A reposição de dados aos utilizadores ou é efetuado com sucesso ou insucesso. Podemos efetuar igualmente um paralelo entre o cenário anterior e o cenário atual, medindo a probabilidade de sucesso previamente à incorporação do dispositivo no cliente e posteriormente à incorporação do dispositivo no cliente.

O RPO e o RTO serão medido através de um inquérito de satisfação ao cliente. Desta forma será possível responder aos benefícios atingidos pelo produto.

## 4.1 Tempos de indisponibilidade de serviços ou ativos

Vejamos agora um cenário prévio à implementação do produto. O disco da máquina Windows2 está quase cheio. O administrador de sistemas do cliente XPTO não sabe. O disco ficando cheio pode levar a que aconteça um cenário catastrófico, causando inúmeros prejuízos para a empresa.

Tabela 14 - Cenário prévio à implementação nº 1

| **Ativo incapacitado** | **Tempo** |
|---|---|
| Disco da máquina Windows2 | Impossível de calcular |

De referir que no cenário prévio não se consegue ser pró-ativo, uma vez que está dependente da entrada de um pedido de intervenção.

Após a implementação do produto podemos constatar que de imediato foi recebido um alerta por parte da equipa de suporte da Visionware e do cliente. Assim contactou-se de imediato o responsável IT da empresa da empresa XPTO, tendo sido posteriormente efetuada uma intervenção.

Tabela 15 - Cenário após à implementação nº 1

| **Ativo incapacitado** | **Tempo** |
|---|---|
| Disco da máquina Windows2 | ~ 45 minutos |

Um outro cenário que esporadicamente acontece em clientes é uma máquina ter ficado inoperacional. A equipa de suporte da Visionware só é notificada quando o responsável IT do cliente avisa. Esta situação pode levar a prejuízos, pois os funcionários sem sistema poderão ficar incapacitados de exercerem as suas funções.

Analisando e transportando o último caso em que foi aberto um *ticket* devido à indisponibilidade de um servidor, o cliente notificou a Visionware após aproximadamente 1h da máquina estar inativa (pela forma de funcionamento operacional do cliente). Foi necessário efetuar uma intervenção que demorou 15 minutos. Ou seja, o tempo total da indisponibilidade do servidor foi de 1 hora e 15 minutos.

Tabela 16 - Cenário prévio à implementação nº 2

| **Ativo incapacitado** | **Tempo** |
|---|---|
| Disponibilidade Windows2 | ~ 1hora e 15 minutos |

Após a implementação, forçando a indisponibilidade da máquina Windows2, automaticamente foi recebido uma notificação por parte da Visionware e do cliente. Adaptando o tempo da intervenção gasto no cenário prévio ao cenário após o tempo da indisponibilidade do servidor foi de aproximadamente 15 minutos.

Tabela 17 - Cenário após à implementação nº 2

| **Ativo incapacitado** | **Tempo** |
|---|---|
| Disponibilidade Windows2 | 15 minutos |

Comparação:

Tabela 18 - Comparação entre cenário prévio e após monitorização

| Ativo incapacitado | Cenário prévio | Cenário após |
|---|---|---|
| Disco da máquina Windows2 | Impossível de calcular | ~ 45 minutos |
| Disponibilidade Windows2 | ~ 1hora e 15 minutos | 15 minutos |

Desta forma podemos constatar que o produto *Wonderbox* com o envio das respetivas notificações são de extrema importância, reduzindo o tempo da indisponibilidade de serviços e precavendo futuros prejuízos financeiros.

De ressalvar que os tempos calculados não contemplam o tempo da deslocação por esta ter comportamentos erráticos e não quantificáveis.

## 4.2 Reposição dos dados aos utilizadores

A reposição dos dados aos utilizadores também carece da nossa atenção em termos de avaliação.

Neste cenário prévio à implementação do produto, a utilizador A pediu para que fosse reposto o ficheiro **contabilidade1trimestre.xls** que se encontrava na diretoria **C:\Users\A\Contabilidade**. Uma vez que a empresa XPTO não tem nenhum mecanismo de cópias de segurança aos utilizadores, não foi possível restaurar o ficheiro.

Tabela 19 - Cenário prévio à implementação nº 3

| Utilizador | Reposição |
|---|---|
| A | Insucesso |

Aqui, o utilizador B apagou a pasta **RPI** no em **C:\Users\B\documents\Informatica**. Pelo mesmo motivo que o cenário anterior não foi possível recuperar a devida pasta.

Tabela 20 - Cenário prévio à implementação nº 4

| Utilizador | Reposição |
|---|---|
| B | Insucesso |

Cenário após à implementação do produto:

Simulando o erro no computador do A e apagando acidentalmente o ficheiro **teste.txt** na pasta **C:\Users\A\Contabilidade**, foi possível através das cópias de segurança efetuadas anteriormente pelo do Bareos, repor o ficheiro com sucesso.

Tabela 21 - Cenário após à implementação nº 4

| **Utilizador** | **Reposição** |
|---|---|
| A | Sucesso |

Voltando a forçar o erro, e criando uma pasta **Teste** em **C:\Users\B\documents\Informatica**, foi possível a sua recuperação com sucesso.

Tabela 22 - Cenário após à implementação nº 5

| **Utilizador** | **Reposição** |
|---|---|
| B | Sucesso |

Comparação:

Através desta prova de conceito foi possível concluir como podemos ver na tabela nº 24, que caso existam cópias de segurança é possível repor com sucesso os dados pretendidos pelos utilizadores caso a informação esteja selecionada para ser copiada.

Tabela 23 - Comparação entre cenário prévio e após backup

| **Cenário prévio** | **Cenário após** |
|---|---|
| 0% de sucesso na reposição | 100% de sucesso na reposição |

## 4.3 Tempo de RPO e RTO

Como já vimos anteriormente, pretende-se que o tempo de RPO e RTO sejam alvos de um inquérito de satisfação ao cliente. Este inquérito serve para aferir o grau de satisfação dos clientes que adquirem o produto, perceber o que falhou e também melhorar no futuro.

Contudo devido a motivos externos ao desenvolvimento desta dissertação não foi possível aferir resultados neste inquérito de satisfação. O modelo do inquérito pode ser consultado no apêndice C.

# 5 Conclusões

A crescente competitividade empresarial torna fundamental a que os serviços considerados críticos da organização estejam sempre em funcionamento e a reposição imediata das cópias de segurança em caso de erro é essencial.

Com base nesta premissa foi elaborado uma análise aos problemas e foram estudadas ferramentas por forma de os contornar.

Denota-se no mercado das ferramentas de monitorização uma evolução crescente. Cada vez mais existem soluções comerciais que permitem aos administradores de sistemas serem notificados em caso de falha de um serviço ou ativo da rede. No entanto também se tem verificado um desfasamento entre as soluções *Open Source* e as soluções pagas. Como no caso do Nagios, o "*Core*" apenas cumpre funções básicas enquanto a versão empresarial sofre um melhoramento da interface gráfica, permite escalabilidade na monitorização entre sítios diferentes.

Ao nível de cópias de segurança sente-se que o mercado *Open Source* não está muito evoluído, talvez por que possa existir falta de confiança não só pelos administradores de sistemas, mas também pelos utilizadores finais num sistema deste género.

Após o estudo de tecnologias que possam atingir os objetivos propostos, efetuou-se uma comparação entre elas e optou-se pelo conjunto de ferramentas que vai em conta aos requisitos nomeados pela entidade onde se efetuou a dissertação de mestrado. Consideram-se as ferramentas eleitas a implementar, o Icinga2 e o Bareos, bastante flexíveis e ajustáveis ao ponto de permitirem construir soluções independentemente do tipo e tamanho da rede.

Previamente à implementação deste projeto, o autor desta dissertação, por questões de força maior, viu-se limitado fisicamente, estando impossibilitado de implementar a *Wonderbox* num cliente, como era a ideia inicial. Optou-se então por fornecer o equipamento pré-configurado e *appointements* já elaborados a um colega da Visionware. Este implementou o sistema *Wonderbox* no cliente XPTO. Assim foi possível não só simular a instalação do equipamento

por parte de um responsável IT, mas também avaliar o produto em termos de resposta, integridade e funcionamento na prova de conceito.

Em termos de funcionamento, as grandezas possíveis de avaliar, nomeadamente os tempos de indisponibilidade de serviços ou ativos e a reposição dos dados aos utilizadores tiveram excelente apreciação. Foi possível reduzir não só os tempos de indisponibilidade de serviços ou ativos, mas também repor os dados dos utilizadores sempre que estes possuíssem cópias de segurança.

Contudo esta prova de conceito por questões já referidas anteriormente, não esteve instalado em cliente o tempo suficiente para se elaborarem testes de *stress*, escalabilidade e o preenchimento do questionário relativo à satisfação dos tempos de RPO e RTO.

Considera-se que este projeto contribuiu para o desenvolvimento das capacidades técnicas do autor desta dissertação e contribuiu para a melhoria do *know-how* da equipa técnica e suporte IT da Visionware. Também foi possível o desenvolvimento de um produto com um modelo de negócio que permitirá à Visionware implementá-lo em qualquer dos seus clientes, atuais e futuros, uma vez que o produto é ajustável ao tipo e tamanho da infraestrutura.

## 5.1 Trabalho Futuro

Para trabalho futuro é necessário utilizar esta prova de conceito em mais parques informáticos ajustando o sistema por forma a corresponder às necessidades de cada infraestrutura.

É também necessário efetuar testes de *stress* e escalabilidade. Este tipo de avaliação só é possível de se efetuar mediante o crescimento dos parques informáticos envolventes.

Em termos de segurança o Bareos e o Postfix deixam a desejar, pois são diversos os ficheiros de configuração em que a palavra-passe é colocado em *clear text*. É necessário encontrar mecanismos que mitiguem estes riscos de segurança. Provavelmente uma futura atualização destas duas ferramentas fará com que as palavras-passe sejam encriptadas à semelhança do Icinga2.

A disseminação desta solução dar-se-á em breve pelo autor, assim que este esteja totalmente operacional. Estas implementações poderiam ser realizadas por outros colegas, mas por motivos internos optou-se pelo autor da dissertação. Posto a disseminação, será possível aferir a satisfação dos clientes em termos dos tempos RTO e RPO.

# Referências


| | |
|---|---|
| (4Linux) | 4Linux. O que é Postfix. [Online] Disponível em: https://www.4linux.com.br/o-que-e-postfix [Acedido em Outubro 1, 2015]. |
| (Alves A., 2012) | Alves A., 2012. A importância da Tecnologia da Informação nas Empresas. [Online] Disponível em: http://www.webartigos.com/artigos/a-importancia-da-tecnologia-da-informacao-nas-empresas/95285/ [Acedido em Outubro 15, 2015]. |
| (Ansible Installation) | [Online] Disponível em: http://docs.ansible.com/ansible/intro_installation.html [Acedido em Abril 17, 2016]. |
| (António M. Adriano, 2014) | António M. Adriano, 2014. Gerenciamento da Continuidade de Negócio e Plano de Recuperação de Desastres. [Online] Disponível em: http://www.pmgacademy.com/pt/blog/artigos/gerenciamento-da-continuidade-de-negocio-e-plano-de-recuperacao-de-desastres [Acedido em Novembro 6, 2015]. |
| (Amanda) | [Online] Disponível em: https://en.wikipedia.org/wiki/Advanced_Maryland_Automatic_Network_Disk_Archiver [Acedido em Janeiro 30, 2016]. |
| (Amanda Community vs Enterprise Editions) | [Online] Disponível em: http://www.zmanda.com/Amanda-Enterprise-Amanda-Community-comparison.html [Acedido em Janeiro 30, 2016]. |
| (Amanda Network Backup, 2016) | *What is Amanda.*[Online] Disponível em: http://www.amanda.org/ [Acedido em Janeiro 26, 2016]. |
| (Bareos 15.2) | *What's new in Bareos.*[Online] Disponível em: https://www.bareos.org/en/whats_new.html [Acedido em Janeiro 5, 2016]. |
| (Bareos Main Reference, 2016) | [Online] Disponível em: http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-60001.3 [Acedido em Janeiro 5, 2016]. |
| (Brochura VW, 2015) | Brochura da Visionware para apresentação a clientes. |
| (Casella G., 2011) | Casella G., 2011. Sysadmin *tools* [Online] Disponível em: http://gcasella.blogspot.pt/2011/03/sys-admin-tools-01-icinga-vs-nagios.html [Acedido em Janeiro 15, 2016]. |
| (Cirilo M. Luciano, 2010) | Cirilo M. Luciano, 2010. Enviando e-mail de notificação no Nagios. [Online] Disponível em: https://nagiosnapratica.wordpress.com/2010/11/12/artigo9-enviando-e-mail-de-notificacao/ [Acedido em Outubro 20, 2015]. |
| (Chef) | *The Components of Chef and How It Works* [Online] Disponível em: https://www.chef.io/chef/#chef--resources [Acedido em Setembro 27, 2016]. |
| (Chef System Requirements) | *System Requirements.* [Online] Disponível em: https://docs.chef.io/chef_system_requirements.html [Acedido em Abril 17, 2016]. |
| (Cobian, 2006) | *How to backup and restore your data using Cobian Backup.* [Online] Disponível em: http://www.bleepingcomputer.com/tutorials/backup-and-restore-data-with-cobian-backup/ [Acedido em Janeiro 30, 2016]. |
| (Cobian Backup) | [Online] Disponível em: https://en.wikipedia.org/wiki/Cobian_Backup [Acedido em Janeiro 30, 2016]. |
| (Contributor, 2015) | Contributor, 2015. *Comparing DevOps to traditional IT.* [Online] Disponível em: http://devops.com/2015/02/17/comparing-devops-traditional-eight-key-differences/ [Acedido em Abril 17, 2016]. |

(Bruno D., 2015) Bruno D., 2015. Ansible – Conceitos básicos. [Online] Disponível em: http://dbruno.org/ansible/ [Acedido em Setembro 27, 2016].

(DeHaan M., 2013) DeHaan M., 2013. *The origins of Ansible*. [Online] Disponível em: https://www.ansible.com/blog/2013/12/08/the-origins-of-ansible [Acedido em Setembro 27, 2016].

(DisasterRecovery) [Online] Disponível em: http://www.disasterrecovery.org/ [Acedido em Janeiro 5, 2015].

(Duarte H., 2014) Duarte H., 2014. Mantenha os seus arquivos sempre à prova de perda com o Cobian Backup. [Online] Disponível em: http://www.techtudo.com.br/tudo-sobre/cobian-backup.html [Acedido em Janeiro 30, 2016].

(Duffy J., 2012) Duffy J., 2012. *Get organized: backup your most important data.* [Online] Disponível em: http://www.pcmag.com/article2/0,2817,2405876,00.asp [Acedido em Novembro 2, 2015].

(Duplicati) [Online] Disponível em: http://www.duplicati.com/home [Acedido em Abril 26, 2016].

(Duvall P., 2012) Duvall P., 2012. Agile DevOps. [Online] Disponível em: http://www.ibm.com/developerworks/br/library/a-devops2/ [Acedido em Abril 16, 2016].

(Eric, 2011) *What is Bacula.*[Online] Disponível em: http://www.bacula.org/5.1.x-manuals/en/main/main/What_is_Bacula.html [Acedido em Janeiro 6, 2016].

(Filho G., 2013) Filho G., 2013. Análise de Vulnerabilidades com o OpenVas em 12 passos. [Online] Disponível em: http://www.pviana.com.br/2013/08/analise-de-vulnerabilidades-com-o.html [Acedido em Abril 15, 2016].

(Freire F., 2013) Freire F., 2013. [Online] Disponível em: https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/o_que_devops?lang=en [Acedido em Abril 16, 2016].

(Fundamentos da ISO 22301) Fundamentos da ISO 22301. [Online] Disponível em: http://advisera.com/27001academy/pt-br/o-que-e-a-iso-22301/ [Acedido em Janeiro 15, 2016].

(Galstad E., 2015) Galstad E., 2015. *Notifications Examples and Troubleshooting.*[Online] Disponível em: http://docs.icinga.org/latest/en/notifications2.html [Acedido em Outubro 20, 2015].

(Gula I. et Beerheide T., 2015) *The smart solution with professional support.* [Online] Disponível em: https://www.bareos.com/files/bareos-documents/Bareos-brochure-en-2015-03.pdf [Acedido em Janeiro 6, 2016].

(Hein M. et Friedrich M., 2013) *Icinga Open Source Monitoring.* [Online] Disponível em: http://www.slideshare.net/icinga/icinga-clt-2013 [Acedido em Novembro 1, 2015].

(Hillebrandt T., 2015) Hillebrandt T., 2015. Entendendo o Puppet. [Online] Disponível em: https://tiagohillebrandt.eti.br/puppet-1-entendendo-o-puppet.html [Acedido em Abril 17, 2016].

(Horse Project) [Online] Disponível em: http://horseproject.wiki/?title=File:Schematic_ITSC_and_RTO,_RPO,_MI.jpg [Acedido em Janeiro 5, 2016].

(Icinga) [Online] Disponível em: https://www.icinga.org/ [Acedido em Novembro 1, 2015].

(Icinga 2) *Icinga 2 – Distributed Monitoring.* [Online] Disponível em: https://www.icinga.org/icinga/icinga-2/distributed-monitoring/ [Acedido em Novembro 10, 2015].

| | |
|---|---|
| (Kokemuller N., 2015) | Kokemuller N., 2015. *Why is technology important inbusiness*.[Online] Disponível em: http://www.ehow.com/about_6320228_technology-important-business_.html [Acedido em Outubro 15, 2015]. |
| (Kosutic D.) | Kosutic D. Risk assessment vs. Business impact. [Online] Disponível em: http://advisera.com/27001academy/knowledgebase/risk-assessment-vs-business-impact-analysis/ [Acedido em Dezembro 20, 2015]. |
| (Kroll Ontrack, 2015) | *Failure causes.*[Online] Disponível em: http://www.krollontrack.com/company/news-releases/?getPressRelease=62385 [Acedido em Novembro 5, 2015]. |
| (Laurent A. et Rémi B., 2008) | Laurent A. et Rémi B., 2008. [Online] Disponível em: http://www.aims-conference.org/issnsm-2008/07-nagios.pdf [Acedido em Janeiro 15, 2016]. |
| (Leonov V.) | Leonov V. *Tenable Nessus: registration, installation, scanning and reporting*. [Online] Disponível em : http://avleonov.com/2016/05/16/tenable-nessus-registration-installation-scanning-reporting/ [Acedido em Setembro 24, 2016]. |
| (Macêdo D., 2012) | Macêdo D., 2012. *Backup* Conceitos e Tipos. [Online] Disponível em: http://www.diegomacedo.com.br/backup-conceito-e-tipos/ [Acedido em Dezembro 28, 2016]. |
| (Macedo V., 2011) | Macedo V., 2011. Monitorização de sistemas de informação críticos. [Online] Disponível em: https://repositorio-aberto.up.pt/bitstream/10216/63403/1/000149170.pdf [Acedido em Janeiro 15, 2016]. |
| (Martinelo C. et Bellezi M., 2014) | Martinelo C. et Bellezi M., 2014. Análise de Vulnerabilidades com OpenVas e Nessus. [Online] Disponível em: http://revistatis.dc.ufscar.br/index.php/revista/article/viewFile/74/68 [Acedido em Abril 15, 2016]. |
| (Markovski M., 2016) | Zabbix *and You*. [Online] Disponível em: http://www.slideshare.net/martinmarkovski1/zabbix-40951433 [Acedido em Janeiro 28, 2016]. |
| (Mattes V. Ícaro et al. 2014) | Mattes V. Ícaro et al. 2014. *The value of the information.* [Online] Disponível em: http://www.infoteca.inf.br/contecsi/smarty/templates/arquivos_template/upload_arquivos/acervo/docs/PDFs/006.pdf [Acedido em Novembro 2, 2015]. |
| (Microcom) | *How to backup you data – Best practices.* [Online] Disponível em: http://www.data-master.com/BackupMediaTypes.html [Acedido em Dezembro 28, 2015]. |
| (Nagios Architecture, 2015) | [Online]. Disponível em: https://www.nagios.org/projects/nagios-core/ [Acedido em Dezembro 11, 2015]. |
| (NagiosCore) | Nagios Core *Notifications.* [Online] Disponível em: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/notifications.html [Acedido em Outubro 25, 2015]. |
| (Nagios Enterprises, 2013) | Nagios – *DistributedMonitoringSolutions*. [Online] Disponível em: https://assets.nagios.com/downloads/general/docs/Distributed_Monitoring_Solutions.pdf [Acedido em Janeiro 22, 2016]. |
| (Nagios to Runtime Database) | [Online] Disponível em: http://docs.opsview.com/lib/exe/detail.php?id=opsview4.0%3Aruntimedb&media=opsview4.0:nagios_to_runtime_architecture.png [Acedido em Janeiro 15, 2016]. |
| (Network Management & Monitoring, 2014) | Observium: all in one network graphing and monitoring. [Online] Disponível em: https://nsrc.org/workshops/2014/afnog-nmf/raw-attachment/wiki/Agenda/observium.pdf [Acedido em Abril 18, 2016]. |

| | |
|---|---|
| (Norton) | O que é o crime cibernético? [Online] Disponível em: http://pt.norton.com/cybercrime-definition [Acedido em Janeiro 17, 2016] |
| (Nicola S.) | Análise de valor de Negócio. [Online] Disponível em: https://moodle.isep.ipp.pt/pluginfile.php/91647/mod_resource/content/2/An%C3%A1lise_Valor_Aula1.pdf [Acedido em Janeiro 20, 2016]. |
| (Observium) | [Online] Disponível em: http://www.observium.org/ [Acedido em Abril 17, 2016]. |
| (Pagés S., 2014) | Pagés S., 2014. Vantagens de contratar suporte técnico remoto. [Online] Disponível em: https://www.workana.com/blog/pt/vantagens-contratar-suporte-tecnico-remoto-workana/ [Acedido em Setembro 25, 2016]. |
| (Passeri P., 2016) | Passeri P., 2016. 2015 *Cyber Attacks Statistics.*[Online] Disponível em: http://www.hackmageddon.com/2016/01/11/2015-cyber-attacks-statistics/ [Acedido em Janeiro 15, 2016]. |
| (Pinto P., 2014) | Pinto P., 2014. Observium: Esteja de olhos nos equipamentos da sua rede. [Online] Disponível em: http://pplware.sapo.pt/tutoriais/observium-esteja-de-olho-nos-equipamentos-da-sua-rede/ [Acedido em Abril 17, 2016]. |
| (Portal informativo ISO 27001) | O que é a norma ISO 27001?. [Online] Disponível em: https://www.27001.pt/index.html [Acedido em Janeiro 15, 2016]. |
| (PublicSafety Canada) | *A guide to business continuity planning.* [Online] Disponível em: http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/bsnss-cntnt-plnnng/index-eng.aspx [Acedido em Dezembro 15, 2015]. |
| (Puppet System Requirements). | [Online] Disponível em: https://docs.puppet.com/puppet/3.7/reference/system_requirements.html#hardware [Acedido em Abril 18, 2016]. |
| (Ramos A.) | Cópias de Segurança para documentos informatizados. [Online] Disponível em: http://www.drec.min-edu.pt/repositorio/manualcopiassegpdf.pdf [Acedido em Janeiro 20, 2016]. |
| (Raspberry Pi 2) | *Raspberry Pi 2 Model B.* [Online] Disponível em: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ [Acedido em Novembro 15, 2015]. |
| (Raspberry PI 2 release) | [Online] Available: https://www.raspberrypi.org/help/faqs/#generalFuture [Acedido em Fevereiro 15, 2015]. |
| (Raspberry Pi Foundation) | *What is Raspberry Pi.*[Online] Disponível em: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/ [Acedido em Novembro 15, 2015]. |
| (Raymond G., 2009) | Raymond G., 2009. *Scanner* de Segurança OpenVas. [Online] Disponível em: https://www.vivaolinux.com.br/dica/Scanner-de-seguranca-OpenVAS [Acedido em Abril 15, 2016]. |
| (Rodrigues C.) | Rodrigues C. ITIL – *Information Technology Infrastructure Library.* [Online] Disponível em: http://www.projetoderedes.com.br/artigos/artigo_itil.php [Acedido em Dezembro 15, 2015]. |
| (Rouse M., 2016) | Rouse M., 2016. PPP (Point-to-Point Protocol) – *definition*. [Online] Disponível em: http://searchnetworking.techtarget.com/definition/PPP [Acedido em Setembro 25, 2016]. |
| (Sakis) | *About* Sakis3g. [Online] Disponível em: http://www.sakis3g.com/#about [Acedido em Agosto 27, 2016]. |
| (Silva P., 2015) | Silva P., 2015. A importância da Tecnologia para as Empresas. [Online] Disponível em: http://faflor.com.br/administrasempre/?p=117 [Acedido em Outubro 16, 2015]. |

(Sophos, 2015) *Worldbackupday.* [Online] Disponível em: https://blogs.sophos.com/2015/03/31/world-backup-day-why-backups-are-so-important-and-some-data-protection-tips-for-businesses/ [Acedido em Outubro 25, 2015].

(SESHACHALA S., 2015) SESHACHALA S., 2015. *Ansible Best Practices: Automation, Provisioning and Configuration Management*. [Online] Disponível em: https://devops.com/2015/05/19/ansible-automation-provisioning-configuration-management/ [Acedido em Setembro 27, 2016].

(Tabona A., 2015) Tabona A., 2015. *The top 20 Free Network Monitoring and Analysis Tools for sysadmins.* [Online] Disponível em: http://www.gfi.com/blog/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins/ [Acedido em Outubro 10, 2015].

(Team Icinga, 2014) *Current state of Icinga.* [Online] Disponível em: http://wiki.monitoring-portal.org/_media/workshop/2014/monitoringworkshop_2014_icinga.pdf [Acedido em Novembro 9, 2015].

(Webmaster, 2012) Webmaster 2012. *Types of backup.* [Online] Disponível em: http://typesofbackup.com/ [Acedido em Dezembro 28, 2015].

(Zabbix) What is Zabbix. [Online] Disponível em: https://www.zabbix.com/documentation/2.0/manual/introduction/about [Acedido em Janeiro 27, 2016].

# Apêndice A:

Este apêndice tem por objetivo demonstrar como o utilizador efetuou a instalação do Icinga 2 com a respetiva interface gráfica. Assume-se ao longo deste apêndice que o leitor tem conhecimento dos comandos em ambiente Linux pelo que não serão explicados.

O primeiro passo foi a configuração do repositório Icinga e respetivo *update*. Note de que um repositório serve para o utilizador não efetuar *download* de *software* de sítios não seguros.

```
# add-apt-repository ppa:formorer/icinga
# apt-get update
```

Seguidamente instala-se o Icinga 2 no Raspberry.

```
# apt-get install icinga2
```

Após a instalação o Icinga 2 por omissão cria as seguintes pastas como podemos observar na tabela seguinte.

| **Caminho** | **Descrição** |
|---|---|
| /etc/icinga2 | Contém os ficheiros de configuração. |
| /usr/lib/Nagios/plugins/ | Alberga os *plugins* do Nagios. |
| /usr/sbin/icinga2 | Contém o binário. |
| /usr/share/doc/icinga2 | Contém documentação do Icinga2. |
| /var/run/icinga2 | PID *file*. |
| /var/lib/icinga2 | Ficheiros do estado do Icinga 2 e ficheiros de configuração. |
| /var/log/icinga2 | Ficheiros de registo. |

Após a instalação do Icinga 2, por omissão, este já está no estado de execução, mas podemos analisar o estado através do seguinte comando:

```
# systemctl status icinga2
```

A interface instalada é a mais recente até à data, a Icinga Web 2 interface e foi instalada e configurada da seguinte forma:

Instalação do servidor MySQL.

```
# apt-get install mysql-server mysql-client
# apt-get install icinga2-ido-mysql
```

Configuração da base de dados. Note de que -u é o utilizador e -p é a palavra-passe.

```
# mysql -u root -p
```

Criar uma base de dados chamada icinga.

```
mysql>  CREATE DATABASE icinga;
```

Atribuição de permissões do utilizador à base de dados.

```
GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON
icinga.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga';
```

Após a configuração da base de dados, é necessário efetuar-se a instalação do servidor de apache.

```
# apt-get install apache2
```

Depois da preparação da base de dados e da instalação do servidor de apache vamos instalar a interface Icinga Web 2.

Adição do repositório de Icinga Web 2.

```
# wget -O - http://debmon.org/debmon/repo.key 2>/dev/null | apt-key add -
echo 'deb http://debmon.org/debmon debmon-wheezy main'
>/etc/apt/sources.list.d/debmon.list
```

Atualização dos repositórios.

```
# apt-get update
```

Instalação.

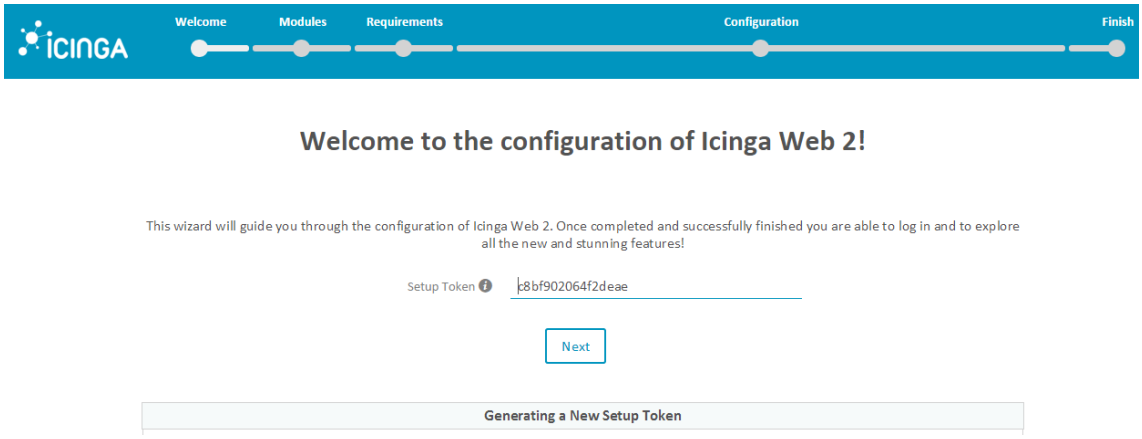
```
# apt-get install icingaweb2
```

Esta interface disponibiliza um tutorial para ajudar o utilizador a parametrizar a interface. Para tal a primeira vez será pedido ao utilizador um *token* que pode ser gerado da seguinte forma:

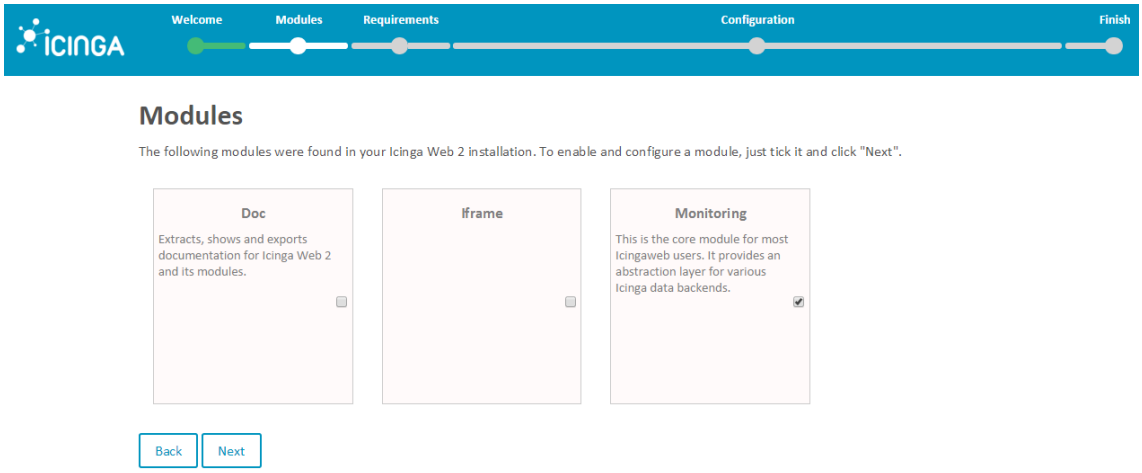
```
# icingacli setup token create
```

Para iniciar o processo de configuração da interface Web através de um *wizard*, pode-se aceder a ela através do endereço: **http://IP/icingaweb2/setup**.

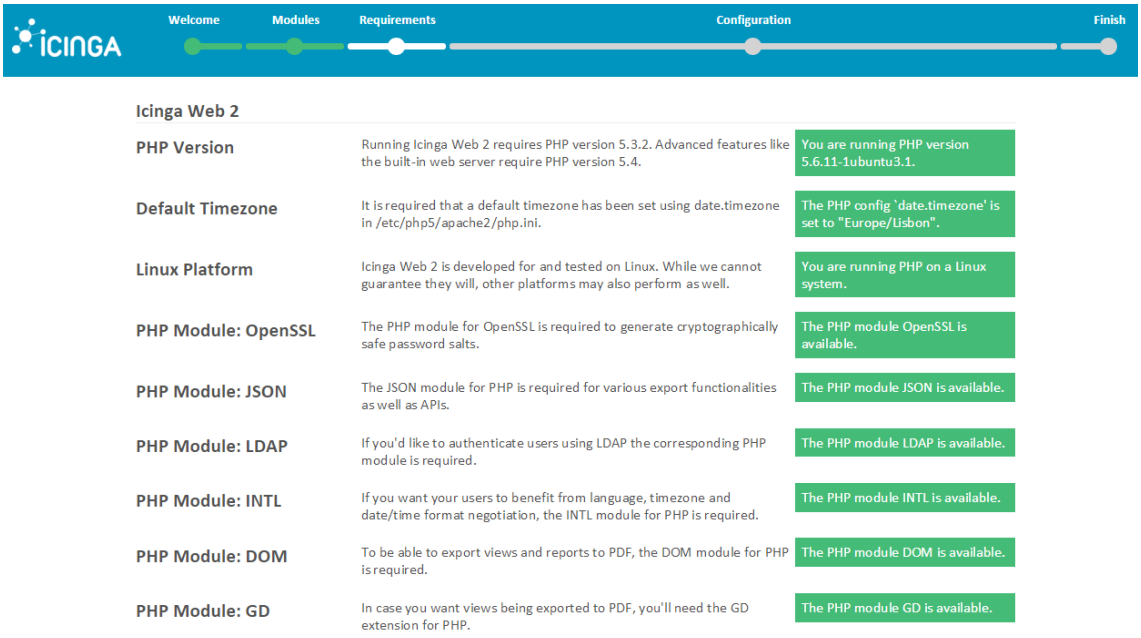
Após aceder à interface via http, será pedido é pedido ao utilizador a introdução do *token* como podemos ver na imagem seguinte.



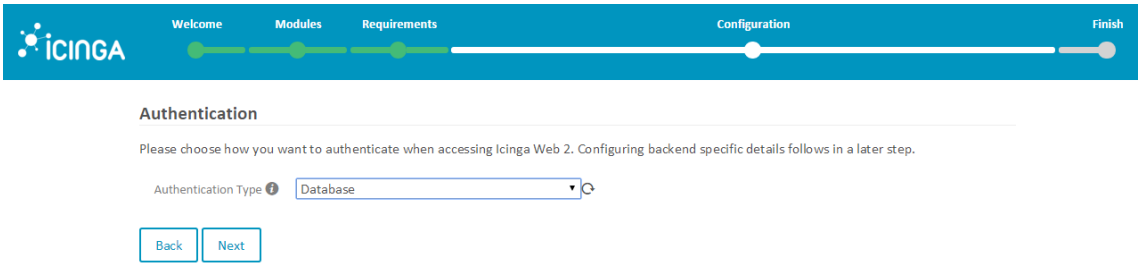

Após a introdução do *token*, é necessário a ativação dos seguintes módulos, doc (extrai, mostra e exporta a documentação do Icinga), *iframe* e *monitoring* (*core* da interface gráfica). Estes três módulos são essenciais para o funcionamento da interface.



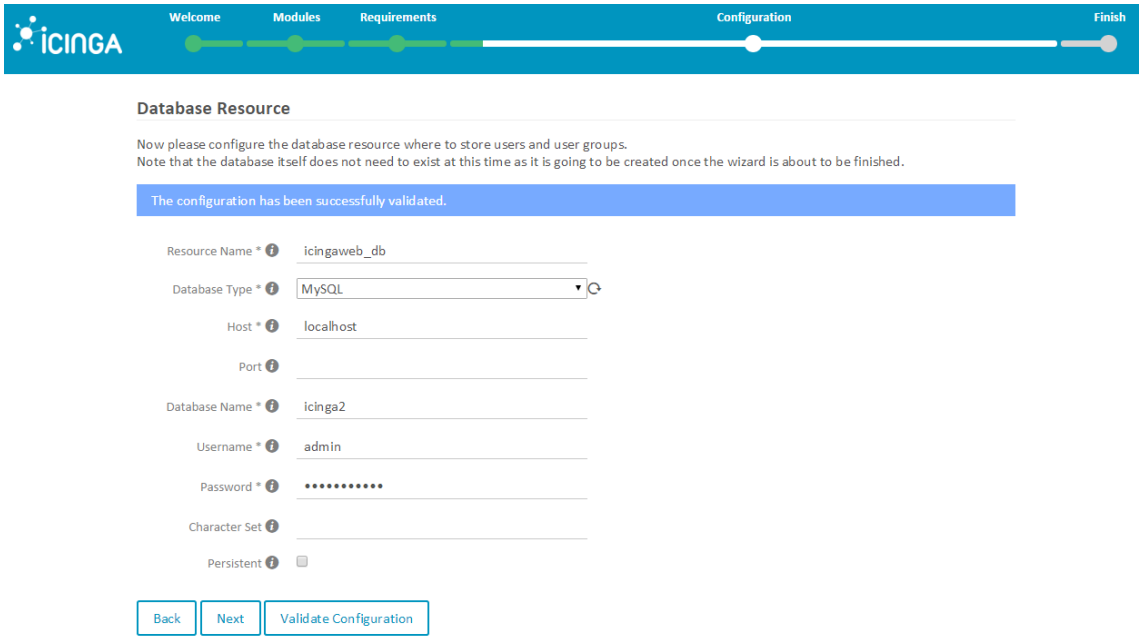

Na imagem seguinte podemos verificar se o sistema operativo tem requisitos mínimos para a instalação da interface gráfica. Caso não cumpra é necessário efetuar a sua instalação manualmente através da consola.



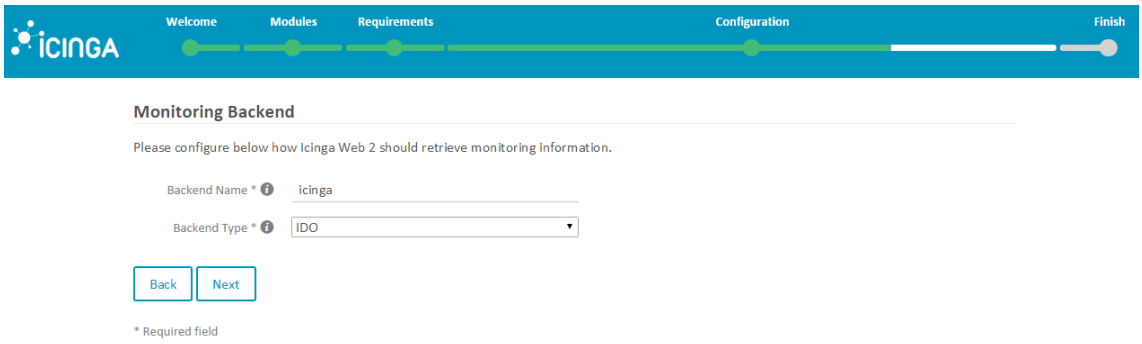

Nas imagens seguintes definimos a base de dados para a autenticação.



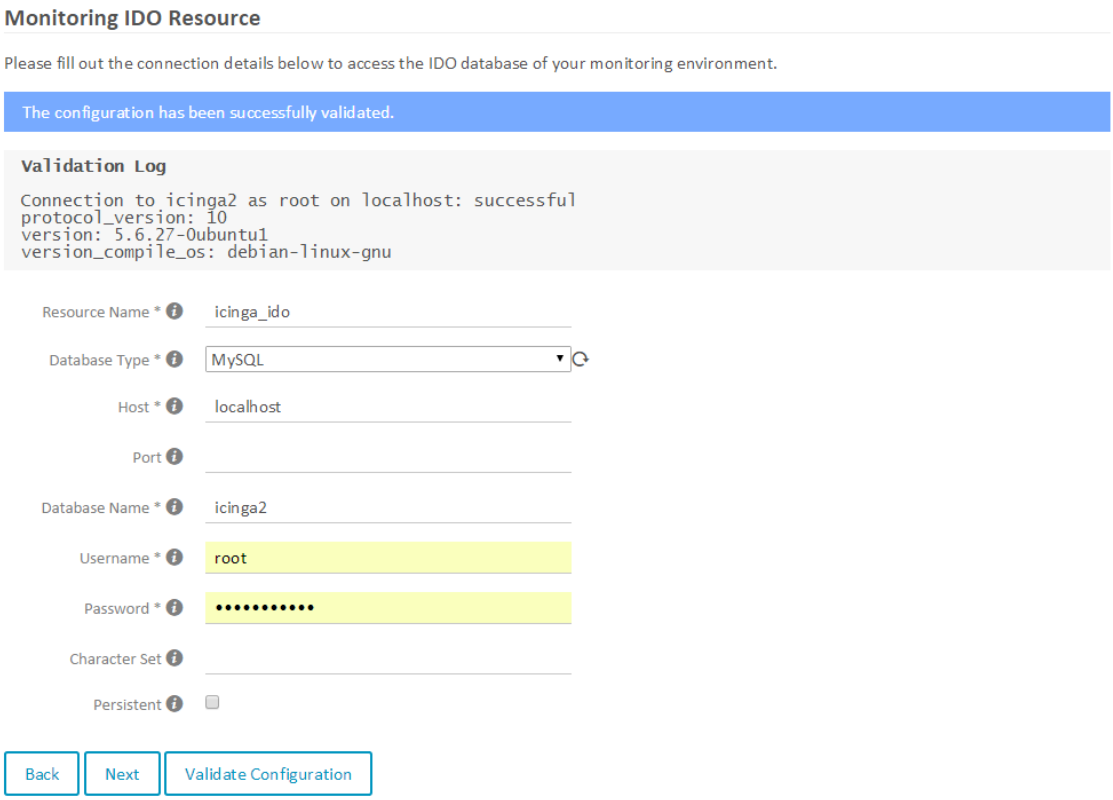

O preenchimento na imagem seguinte à exceção da palavra-passe é automático, podendo o administrador de sistemas caso assim o pretenda alterar os campos.



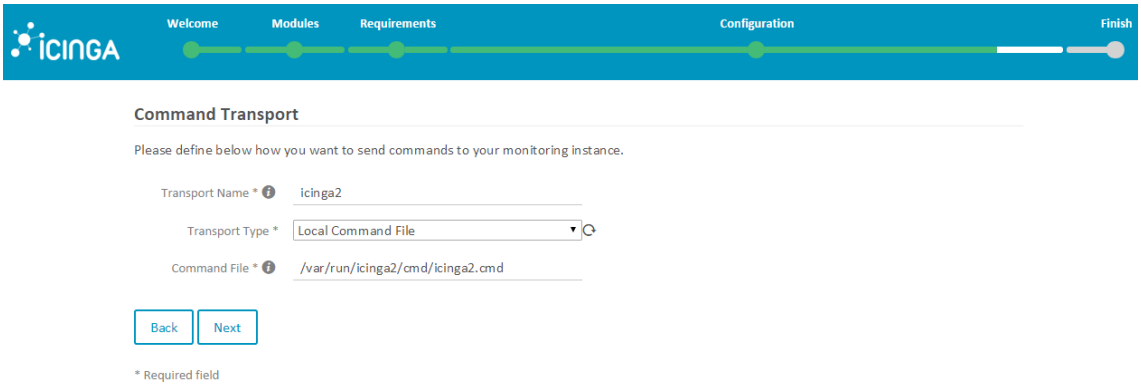

Aqui configuramos a ligação entre o *frontend* e o *backend* da aplicação.




Na imagem seguinte, igualmente de preenchimento automático à exceção da palavra-passe, o administrador de sistemas validará a conexão entre o *frontend* e o *backend,* observando-se que essa conexão está válida.




Na imagem seguinte permitimos que as ações levadas a cabo pela interface gráfica sejam refletidas no *backend* da aplicação.




Após as configurações surge-nos um resumo das mesmas.

## Monitoring Backend

Icinga Web 2 will retrieve information from your monitoring environment using a backend called "icinga" and the specified resource below:

**Database Resource**

| | |
|---|---|
| **Resource Name** | icinga_ido |
| **Database Type** | mysql |
| **Host** | localhost |
| **Port** | |
| **Database Name** | icinga2 |
| **Username** | root |
| **Password** | ********** |

## Command Transport

Icinga Web 2 will use the named pipe located at "/var/run/icinga2/cmd/icinga2.cmd" to send commands to your monitoring instance.

## Monitoring Security

Icinga Web 2 will protect your monitoring environment against prying eyes using the configuration specified below:

| | |
|---|---|
| **Protected Custom Variables** | *pw*,*pass*,community |

# Apêndice B:

Este apêndice tem por objetivo demonstrar como o utilizador efetuou a instalação do Bareos com a respetiva interface gráfica, assumindo que o servidor de *Apache* já está instalado. Tal como no apêndice anterior assume-se que o leitor está familiarizado com comandos em ambiente Linux.

O primeiro passo é escolher a distribuição da qual queremos fazer *download* na seguinte URL: **http://download.bareos.org/bareos/release/latest/**.

Depois de escolhida a distribuição vamos então adicionar o repositório ao nosso sistema operativo através do seguinte comando:

```
#
printf "deb http://download.bareos.org/bareos/release/latest/$DIST /\n" > /
etc/apt/sources.list.d/bareos.list em que $DIST é a nossa distribuição
```

Adição da chave do repositório para termos a certeza que o repositório é de fonte fidedigna.

```
# wget -
q http://download.bareos.org/bareos/release/latest/$DIST/Release.key -
O- | apt-key add –
```

Atualização dos repositórios.

```
# apt-get update
```

De seguida efetua-se a instalação do Bareos e respetiva base de dados. É possível definirmos dois tipos de base de dados, substituindo o parâmetro $DATABASE por postgresql ou mysql. Para o caso a base dados utilizada foi a postgresql.

```
# apt-get install bareos bareos-database-$DATABASE
```

Após a instalação do *core* e respetiva base de dados, é necessário criar a estrutura onde possam ser armazenados os dados. O *core* do Bareos já possuí 3 *scripts* que efetuam a criação da base de dados, tabelas e privilégios.

```
# su postgres -c /usr/lib/bareos/scripts/create_bareos_database
# su postgres -c /usr/lib/bareos/scripts/make_bareos_tables
# su postgres -c /usr/lib/bareos/scripts/grant_bareos_privileges
```

Posto isto podemos executar o *core* do Bareos.

```
# service bareos-dir start
# service bareos-sd start
# service bareos-fd start
```

De seguida efetua-se a instalação do Bareos-webui.

```
# apt-get install bareos-webui
```

A interface do Bareos fornece uma configuração *default* tanto do acesso à interface como os privilégios que o utilizador terá na interface gráfica.

Para tal é necessário colocar os ficheiros **webui-consoles.conf** e **webui-profiles.conf** na pasta **/etc/bareos/**. Este procedimento efetua-se da seguinte forma:

```
# echo "@/etc/bareos/bareos-dir.d/webui-
consoles.conf" >> /etc/bareos/bareos-dir.conf
# echo "@/etc/bareos/bareos-dir.d/webui
profiles.conf" >> /etc/bareos/bareos-dir.conf
```

De seguida é necessário alterar manualmente o FQDN ou IP para que seja possível aceder à interface gráfica do Bareos. O ficheiro onde se altera o FQDN ou IP é o seguinte: **/etc/bareos-webui/directors.ini**

; Fill in the IP-Address or FQDN of you director.
diraddress = "192.168.1.28"

# Apêndice C:

## Wonderbox

### Inquérito de satisfação – Tempos RTO e RPO

**Dados do Cliente**

Empresa:_____

Função: _____

Data: _____

Rúbrica: _____

(Assinale com uma cruz a sua resposta)

1 = Muito Insatisfeito, 2 = Insatisfeito, 3 = Pouco Satisfeito, 4 = Satisfeito e 5 = Muito Satisfeito.

* RTO – quantidade máxima de informação que se é aceitável perder durante um acidente

* RPO – tempo necessário para que todo o sistema volte a operar normalmente

| Fator | Grau de Satisfação | | | | | Sugestões de melhoria |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Tempo de RTO | | | | | | |
| Tempo de RPO | | | | | | |