Mestrado em Engenharia Informática
Dissertação/Estágio
Relatório Final

csSECURE- Automated Data Discovery and Protection

Nuno Escada
nescada@student.dei.uc.pt

Orientador:
Dr. Edmundo Monteiro
Eng. Bernardo Patrão
Data: 12 de Julho de 2012

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Agradecimentos

Terminada esta importante fase da minha vida académica torna-se indispensável dirigir as minhas palavras de apreço e agradecimento a todos aqueles que contribuíram para o meu crescimento, quer a nível pessoal, quer a nível profissional.

Agradeço aos meus orientadores Eng.º Bernardo Patrão e Dr. Edmundo Monteiro por toda a ajuda prestada e paciência demonstrada durante a realização do estágio uma vez que a sua ajuda foi fundamental quer a nível técnico quer ao nível pessoal.

Agradeço ainda à ITGROW pela oportunidade proporcionada e a todos os colegas do projeto csSECURE que me receberam de braços abertos, proporcionando uma primeira experiencia de trabalho fantástica, num ambiente familiar, o que tornava os dias menos cansativos e longos.

Por ultimo mas não menos importante, agradeço toda a minha família, e aos que me são próximos, em especial à Patrícia Lopes pelo orgulho e entusiasmo que sempre manifestaram. Por todo o seu apoio, não só perante todas as conquistas e resultados apresentados, mas também quando mostravam acreditar em mim mesmo quando as coisas não corriam pelo melhor. Foram eles que tornaram o fim deste curso uma realidade.

Resumo

Este Projecto tem comoprincipal objectivo o design de uma solução que efectue a pesquisa automática de informação produzida numa dada organização, de forma a permitir a protecção da mesma Como tal, pretende-se uma solução que procure de forma transparente documentos na rede, de modo a permitir a sua análise e classificação automática. Desta forma o conteúdo produzido ficará protegido de acessos indevidos.

De forma a atingir este objectivo foi feito um estudo aprofundado e implementada uma solução que actua na rede à procura de conteúdo disponível repositórios partilhados, possibilitando assim a sua protecção. Para além disso a solução apresentada possibilita ao administrador de segurança conhecer os repositórios que são partilhados na rede, facilitando assim a criação de normas de segurança de modo a precaver eventuais fugas de informação.

A solução desenvolvida resultou de um trabalho realizado em várias fases. Este relatório visa descrever o trabalho realizado, desafios e soluções encontradas por forma a cumprir os objectivos traçados.

Deste trabalho resultou não só a ferramenta acima descrita, como também documentação técnica relativa ao desenvolvimento de futuras funcionalidades para o produto, tal como será demonstrado ao longo deste documento.

Índice

1 Introdução

1.1 ENQUADRAMENTO

O presente estágio foi desenvolvido no âmbito da cadeira de dissertação/estágio do Mestrado em Engenharia informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra. O projecto foi iniciado a 12 de Setembro de 2011 e terminou a 12 de Julho de 2012.

Este trabalho decorreu nas instalações da empresa Critical Software S. A., em Coimbra, no âmbito do desenvolvimento de um módulo de Automated Data Discovery and Protection para o produto csSECURE. Este módulo visa a descoberta e protecção da informação partilhada numa organização.

Apesar de estar integrado na equipa de um projecto, o módulo desenvolvido teve o estagiário como único elemento da equipa. Este realizou tarefas de pesquisa e desenvolvimento, auxiliado pelo seu orientador de estágio na empresa, que desempenhou a tarefa de gestão do projecto.

A orientação deste estágio esteve a cargo do Doutor Edmundo Monteiro e do Eng.º Bernardo Patrão, representantes da Universidade de Coimbra e da Critical Software S.A., respectivamente.

1.2 CSSECURE

O csSECURE é uma ferramenta de protecção de informação não estruturada que visa a prevenção de fugas de informação através da encriptação de ficheiros.

Esta ferramenta baseia-se num conceito de segurança multinível, que possibilita a atribuição de diferentes níveis de acesso a documentos, consoante o nível atribuído e o utilizador que pretende aceder à informação. Para tal é feita uma associação entre os direitos de um dado nível de classificação e as credenciais de segurança do utilizador. Ao contrário da maioria das soluções existentes no mercado este sistema centra-se nos dados e não no sistema de transmissão e suporte dos mesmos. Desta forma obtém-se uma protecção permanente da informação, quer esta esteja a ser enviada, guardada ou acedida, independentemente do dispositivo utilizado.

Resumindo, neste produto encontra-se características de Data Loss Prevention (DLP) e de *enterprise Rights Management* (ERM), uma vez que permite a definição de políticas de segurança obrigatórias para o acesso à informação da organização (DLP), ao mesmo tempo que garante que a informação está protegida independentemente do seu estado (ERM).

1.3 OBJECTIVO

Com a proliferação da tecnologia e da utilização de documentos digitais, torna-se cada vez mais difícil controlar a informação partilhada dentro das organizações. Como tal, estas utilizam cada vez mais repositórios de dados, para armazenamento de informação, de forma a terem um maior controlo da informação produzida pelos seus colaboradores. Apesar da existência destes repositórios centrais, os colaboradores nem sempre se cingem à exclusiva utilização dos mesmos. Frequentemente optam pela criação de novos repositórios sem informarem a administração, o que dificulta a tarefa de gestão e controlo de acesso à informação.

Além do problema acima enunciado, existe ainda o facto de cada vez mais existirem fugas de informação confidencial (de forma intencional ou não) dentro das organizações. Apesar de estes riscos já poderem ser minimizados utilizando o csSECURE, ainda existem lacunas a ser preenchidas – nomeadamente, o facto de neste momento, o mesmo, não possuir mecanismos que garantam a protecção automática de informação pré-existente, ou de proveniência externa à organização.

É com o intuito de cobrir estas limitações que surge o módulo de Automated Data Discovery and Protection do csSECURE. Este módulo deve possuir capacidades para:

- Pesquisar de forma automática os Repositórios partilhados na Organização (pastas partilhadas e repositórios CVS);
- Pesquisar de forma automática os ficheiros armazenados nos repositórios encontrados;
- Suportar criação de plug-ins para:
    - Pesquisar e analisar repositórios não suportados nativamente pela ferramenta;
    - Analisar e/ou proteger de ficheiros.

1.4 ESTRUTURA DO DOCUMENTO

A secção actual (Introdução) apresenta a descrição do âmbito e objectivos traçados para o estágio

A secção 2 (Gestão) apresenta a equipa e calendarização do projecto, incluindo reuniões e planeamento.

A secção 3 (Desenvolvimento) descreve as actividades desenvolvidas – desde o levantamento do estado da arte, até ao culminar das tarefas de design da aplicação.

A secção 4 (Resultados) apresenta os resultados finais do trabalho desenvolvido e a documentação produzida.

A secção 5 (Conclusões), tal como o nome indica, apresenta as conclusões retiradas sobre o trabalho e competências adquiridas ao longo do estágio. Nesta secção também podem ser encontradas algumas considerações em relação ao trabalho futuro.

A secção 6 (Glossário) apresenta algumas definições e acrónimos utilizados neste documento.

Finalmente, a secção7 (Referências) apresenta as referências e documentos que devem ser consultados para uma melhor compreensão do trabalho realizado ao longo do estágio.

2 Gestão

Esta secção descreve a metodologia de desenvolvimento utilizada, a equipa a nível organizacional e o planeamento realizado para este projecto. Para uma descrição mais detalhada, deve ser consultado o documento referente ao planeamento [AD-3].

2.1 METODOLOGIA DE DESENVOLVIMENTO

Para a execução do módulo *Automated Data Discovery and Protection* que foi implementado durante o estágio utilizou-se a metodologia de desenvolvimento de *software* Scrum.

Scrum é uma metodologia ágil de desenvolvimento de *software* e de gestão de projectos. Esta metodologia foi desenhada para se adaptar a projectos que estejam sujeitos a constantes mudanças, uma vez que utiliza ciclos de desenvolvimento curtos (tipicamente duas a quatro semanas) conhecidos como *sprints*. Cada *sprint* começa com uma reunião de planeamento e termina com uma reunião de *review*. No final de cada *sprint* deve ser possível fazer uma nova *release* do produto.

Optou-se pela metodologia Scrum devido à sua natureza ágil permitir um acompanhamento diário da evolução do estágio e ao mesmo tempo imprimir um ritmo elevado de prototipagem. O facto de a equipa do csSECURE se encontrar a desenvolver em Scrum facilitou a adopção desta metodologia.

2.2 EQUIPA

O módulo *Automated Data Discovery and Protection* foi desenvolvido num projecto de estágio. A sua equipa é composta por três elementos:

- Estagiário Nuno Escada – responsável pela investigação e desenvolvimento do módulo *Automated Data Discovery and Protection*;
- Eng.º Bernardo Patrão – *Scrum Master* e responsável por SQA (*Software Quality Assurance*);
- Eng.º Sérgio Cruz – *Product Owner*.

2.3 CALENDARIZAÇÃO E ACOMPANHAMENTO

2.3.1 REUNIÕES DE PROJECTO

Este projecto está a ser desenvolvido segundo a metodologia de desenvolvimento de *software Scrum*. Como tal, as reuniões estabelecidas ao abrigo desta metodologia são:

- *Sprint Planning* – Reunião efectuada no início de cada *Sprint.* Nesta reunião é decidido o âmbito da *Sprint,* são seleccionadas e estimadas as tarefas necessárias para a conclusão do âmbito da *Sprint;*
- *Daily Scrum* – Reunião diária com a equipa do projecto e o *Scrum Master*, com uma duração máxima de 15 minutos, em que os participantes dizem quais as tarefas que realizaram desde a última reunião, que tarefas esperam realizar até à próxima reunião, e os obstáculos encontrados até ao momento;

- *Sprint Review* – Reunião realizada no final de cada *Sprint*, onde é demonstrado o trabalho realizado ao longo da *mesma* e verificado se o seu âmbito foi cumprido;

- *Sprint Retrospective:* Reunião que ocorre no final da *Sprint* em que é analisado o desempenho ao longo da mesma. Trata da análise do que correu bem, o que pode ser melhorado e o que correu mal. Desta reunião resulta um conjunto de acções que devem ser tomadas ao longo das *Sprints* futuras.

2.3.2 REUNIÕES DE ESTÁGIO

Reuniões entre o estagiário e respectivos orientadores de estágio (do DEI e da Critical Software S. A.) em que se dá a conhecer o trabalho realizado durante o estágio. Nestas reuniões foram ainda abordadas algumas questões técnicas relacionadas com o projecto.

2.3.3 PLANEAMENTO

Nesta secção será apresentado o planeamento macro das tarefas realizadas Para uma consulta mais detalhado do planeamento, com recurso a user stories, consultar o Anexo C – Planeamento e Metodologia de Desenvolvimento [AD-3].

2.3.3.1 Planeamento inicial

No início do estágio foi feito um planeamento global para a execução das várias tarefas do estágio. Tal planeamento é descrito pelo diagrama de Gantt apresentado na Figura 1. Este diagrama representa uma estimativa inicial de alto nível, uma vez que foi feita antes da especificação dos requisitos. Como tal, na seguinte subsecção será apresentado um diagrama mais detalhado.

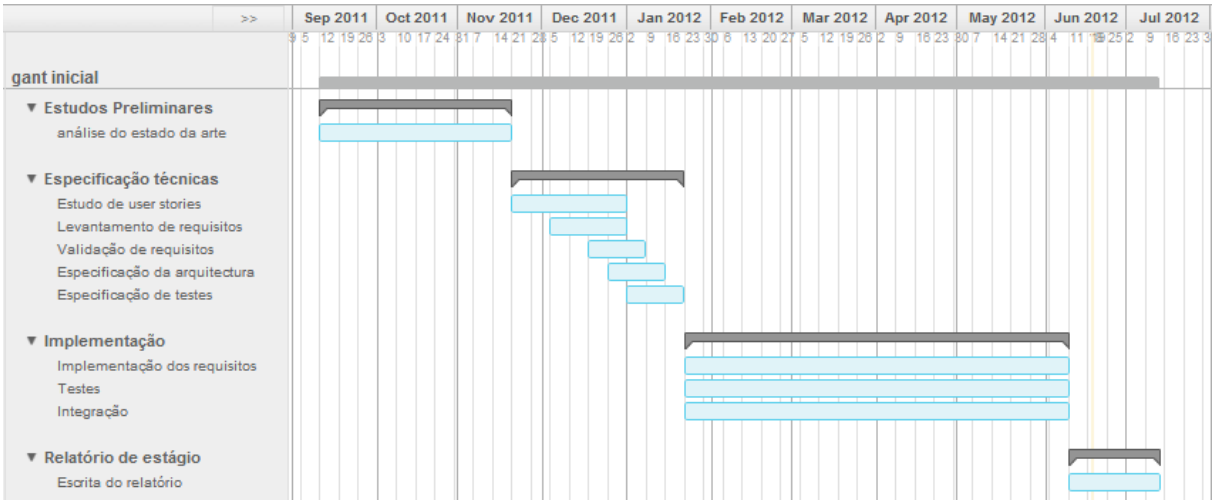

Figura 1 Diagrama de Gantt do planeamento inicial

2.3.3.2 Planeamento intermédio

Após o levantamento de requisitos foi feito um novo planeamento, mais detalhado e tendo em conta a maior maturidade que a equipa adquiriu em relação às necessidades do produto. Desse planeamento resultou um planeamento descrito pelo diagrama de Gantt apresentado na Figura 2. Apesar de o estágio terminar em Julho verificou-se que para a implementação de todas as *user stories definidas no product backlog*

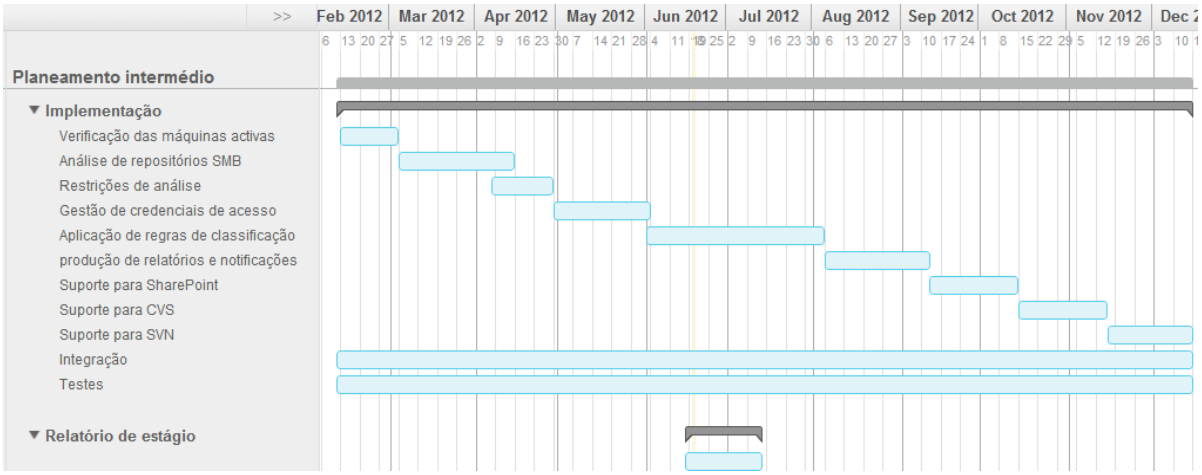

Figura 2 Diagrama de Gantt do planeamento da fase de implementação

2.3.3.3 Execução

Finalizado o estágio, foi criado o diagrama de Gantt ilustrado na Figura 3. Desta forma, tornou-se mais fácil efectuar a comparação entre o que foi planeado e executado. As conclusões desta comparação encontram-se na secção seguinte (Conclusão).

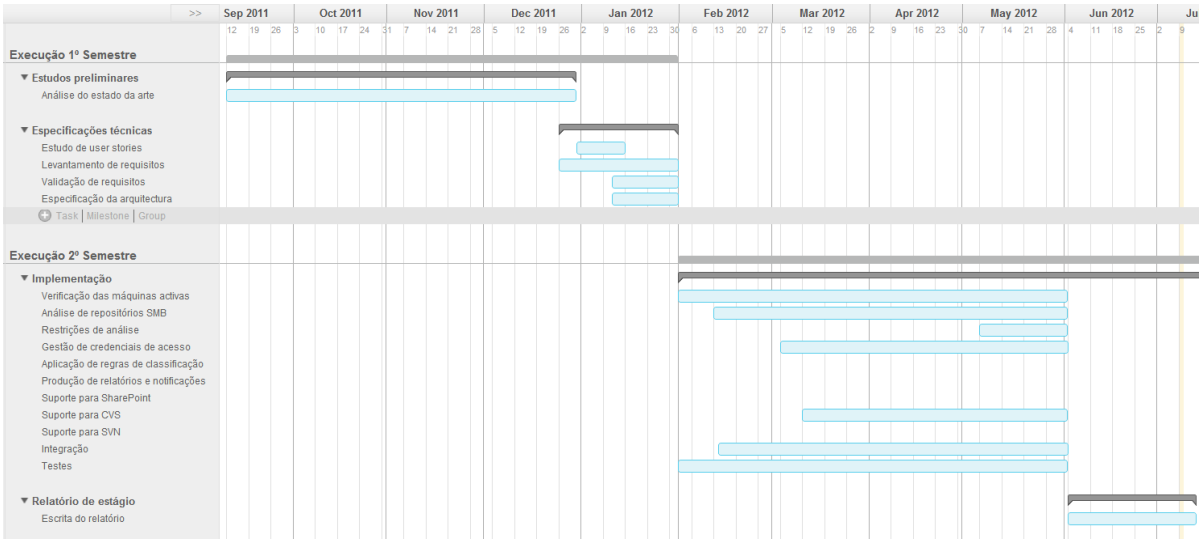

Figura 3 Diagrama de Gantt da execução do estágio

2.3.3.4 Conclusão

Após uma análise à execução realizada, verifica-se alguma disparidade entre o planeado e o executado. Essa disparidade ocorreu maioritariamente devido ao alongar das tarefas relacionadas com o levantamento do estado da arte e à necessidade constante de adaptação entre o planeamento e as necessidades do cliente. Essa adaptação levou por exemplo ao aumento da prioridade atribuída ao suporte para repositórios CVS, fazendo com que tivesse sido incluída a sua implementação durante o estágio.

3 Desenvolvimento

3.1 ESTADO DA ARTE

3.1.1 INTRODUÇÃO

De forma a cumprir os objetivos acima mencionados (secção 1.3) foram analisados vários tipos de ferramentas dentro da área de Data Discovery. Dentro das ferramentas que permitem a realização de descoberta de dados destacam-se as ferramentas de Enterprise Search e os crawlers. É sobre este tipo de ferramentas que incidiu o estudo do estado da arte.

3.1.2 CRITÉRIOS DE SELECÇÃO

Dado o elevado número de ferramentas de Enterprise Search existentes no mercado, foram apenas seleccionadas as consideradas mais relevantes, quer em termos de quota de mercado, como em termos de funcionalidades oferecidas [RD-1]. No caso das ferramentas comerciais foram analisadas aquelas que apresentaram melhores resultados em análises de mercado feitas quer pela Gartner [RD-1] quer pela Forrester [RD-3]. No caso das ferramentas não comerciais foram analisadas duas das mais utilizadas actualmente, cujo tipo de licenciamento fosse compatível com a sua utilização no csSECURE [RD-4]

Para além das ferramentas direccionadas à pesquisa de dados, foram também analisadas três ferramentas direccionadas à descoberta de falhas de segurança (Nessus, NMAP e OpenVAS). Estas ferramentas foram alvo de análise uma vez que uma das suas funções é a descoberta de *hosts* activos, o que faz parte dos objectivos acima traçados.

3.1.3 FUNCIONALIDADES ANALISADAS

Após a análise das ferramentas acima mencionadas foi elaborada uma matriz comparativa que faz a correspondência entre estas e as funcionalidades requeridas a fim de cumprir os objetivos traçados para o estágio.

Para a elaboração desta matriz foram consideradas as seguintes características:

- Política de licenciamento;
- Plataformas;
- Tipos de ficheiros Suportados:
    - PDF;
    - MS Office 2003;
    - MS Office 2010;
    - Open Document.

- Descoberta automática de *hosts;*
- Descoberta e análise automática a automática dos repositórios de dados:
    - CVS;
    - Samba;
    - SharePoint;
    - SVN

Desta análise resultou a matriz de funcionalidades descrita pela Tabela 1.

		Cyclops	Endeca	FAST	GSA	OSES	SOLR	Sphinx	Nutch	Nessus	NMAP	OpenVAS
Política de licenciamento		Prop	Prop	Prop	Prop	Prop	Apache 2	GPL	Apache 2	Prop	LGPL	GPL
Plataformas	Windows	✓	✓	✓	NA	✓	✓	✓	✓	✓	✓	
	Linux	✓	✓		NA	✓	✓	✓	✓	✓	✓	✓
	MacOS	✓	✓		NA	✓	✓	✓	✓			
	outros	✓	✓		NA	✓	✓	✓	✓	✓	✓	✓
Descoberta automática de Hosts										✓	✓	✓
Repositórios de dados Suportados	CVS	✓										
	Samba/SMB	✓	✓	✓	✓	✓	✓		✓			
	SharePoint		✓	✓	✓							
	SVN											
	Desenvolvimento de plug-ins	✓	✓	✓	✓	✓	✓	✓	✓			
Descoberta automática de repositórios	CVS											
	Samba/SMB			✓	✓	✓				✓	✓	
	SharePoint			✓	✓	✓						
	SVN											
Ficheiros indexáveis	PDF	✓	✓	✓	✓	✓	✓		✓			
	MS Office 2003	✓	✓	✓	✓	✓	✓		✓			
	MS Office 2007		✓	✓	✓	✓	✓		✓			
	Open Document	✓	✓	✓	✓	✓	✓		✓			

Tabela 1- Matriz de funcionalidades

3.1.4 CONCLUSÕES

Através do levantamento do estado da arte (tal como pode ser verificado na Tabela 1) foi possível verificar que nenhuma das ferramentas analisadas cumpre todos os objetivos traçados. Foi também possível verificar que das soluções existentes no mercado, as que mais se aproximam do pretendido no âmbito do csSECURE são as soluções comerciais de Enterprise Search.

As ferramentas de Enterprise Search comerciais na sua maioria realizam a descoberta automática de ficheiros, contudo estas aplicações não realizam nem as tarefas de auditoria, nem a classificação automática de ficheiros. Estas limitações também se aplicam às ferramentas não comerciais, sendo que a estas ainda acresce o facto de ou necessitarem de um *crawler* ou necessitarem do desenvolvimento de funcionalidades de acesso a alguns tipos de repositórios de dados.

No que toca aos *crawlers* foi possível aferir que estes são maioritariamente direccionados para a Web e não para um ambiente empresarial. Para além disso os existentes estão integrados em ferramentas de *Enterprise Search*, sendo que mesmo aquele que foi analisado (Nutch) está a ser incluído numa dessas ferramentas. Para além disso são o crawler analisado apresenta o mesmo tipo de limitações que as ferramentas de Enterprise Search não comerciais.

Relativamente às ferramentas vocacionadas à descoberta de falhas de segurança, foi possível verificar que possuem características que preenchem alguns dos objetivos traçados para o módulo em desenvolvimento. Todas as ferramentas vocacionadas à descoberta de falhas estudadas realizam com sucesso a descoberta automática de máquinas. Para além disso duas das ferramentas analisadas (NESSUS e NMAP) conseguem detectar a existência de repositórios Samba e SMB nas máquinas analisadas.

Do levantamento do estado da arte foi possível retirar algumas ideias para o desenvolvimento da aplicação requerida nomeadamente relativamente a qual o tipo de arquitectura que poderia ser utilizada e qual o tipo de expansibilidade que deveria ser permitido.

Estas conclusões advêm do número elevado de ferramentas que possibilitam a criação de plug-ins para a análise dos repositórios de dados não suportados e de a maioria das ferramentas estudadas actuarem a partir de um servidor central para fazer a pesquisa na rede.

Foi ainda possível verificar que seria possível a utilização do NMAP para a realização de alguns objectivos do estágio, mais concretamente ao nível da descoberta de máquinas.

3.2 ESPECIFICAÇÃO DE REQUISITOS

3.2.1 INTRODUÇÃO

Ao iniciar a fase de requisitos foi necessário tomar uma decisão fundamental ao nível do tipo de arquitectura que se iria utilizar no desenvolvimento da aplicação que iria condicionar todo o processo de *design* da aplicação. Foi então necessário decidir entre duas abordagens distintas:

- Criação de clientes (Windows e Linux) que seriam instalados nas diferentes máquinas e que alimentariam um servidor central com a informação recolhida sobre a máquina e conteúdo presente na mesma;

- Implementação de um servidor central que fizesse a descoberta de repositórios de dados partilhados.

Nas secções seguintes serão apresentadas com mais detalhe as abordagens consideradas.

3.2.2 ABORDAGENS CONSIDERADAS

A primeira opção considerada deveu-se à necessidade da ferramenta estar integrada no csSECURE. Nesta abordagem existiria um servidor central que armazenaria os dados sobre o conteúdo presente nas máquinas em que o módulo de data discovery estivesse instalado. A Figura 4 ilustra de forma esquemática o funcionamento desta abordagem.

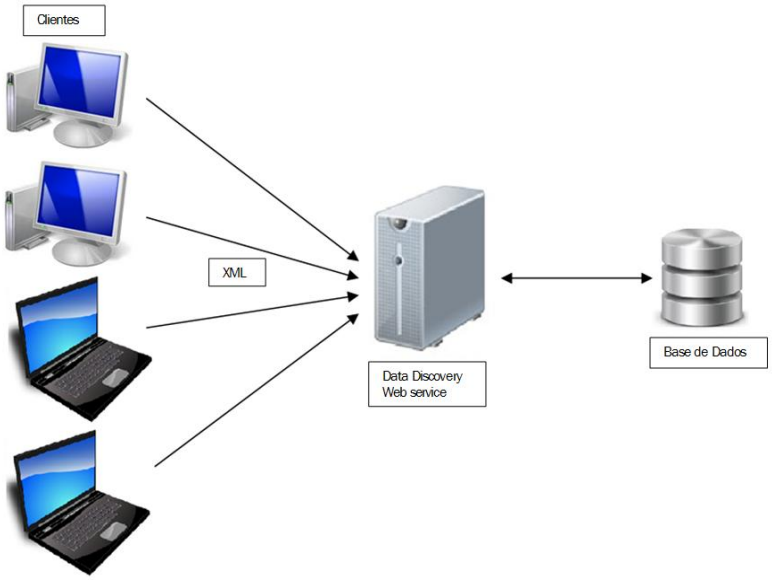

Figura 4 Implementação distribuída pelos clientes

Esta alternativa funcionaria através da instalação de um serviço que :

- Periodicamente procurasse por alterações a ficheiros e pastas;
- Comunicasse com um *web server* que registaria as alterações detectadas;
- Executasse *plug-ins* de análise e protecção de ficheiros.

Esta abordagem apresenta como principais vantagens o seu elevado grau de escalabilidade, e a pequena quantidade trafego de rede gerado pela aplicação. Estas vantagens advêm do facto de as tarefas de descoberta e análise de ficheiros serem realizadas em cada um dos computadores em que a aplicação estaria instalada, sendo apenas necessário comunicar com um web service que trataria da persistência dos dados obtidos,

Por outro lado foram encontradas as seguintes desvantagens: necessidade de instalação em todas as máquinas que sujeitas a análise, a solução não tem em conta se a informação está partilhada ou se o seu uso é apenas local, e a impossibilidade de conhecer de todas as máquinas ligadas à rede.

Outra solução abordada consiste na criação de uma ferramenta, que é instalada num servidor central. Esta solução deve pesquisar toda a rede, à procura de máquinas e do respectivo conteúdo partilhado, tal como descreve a Figura 5.

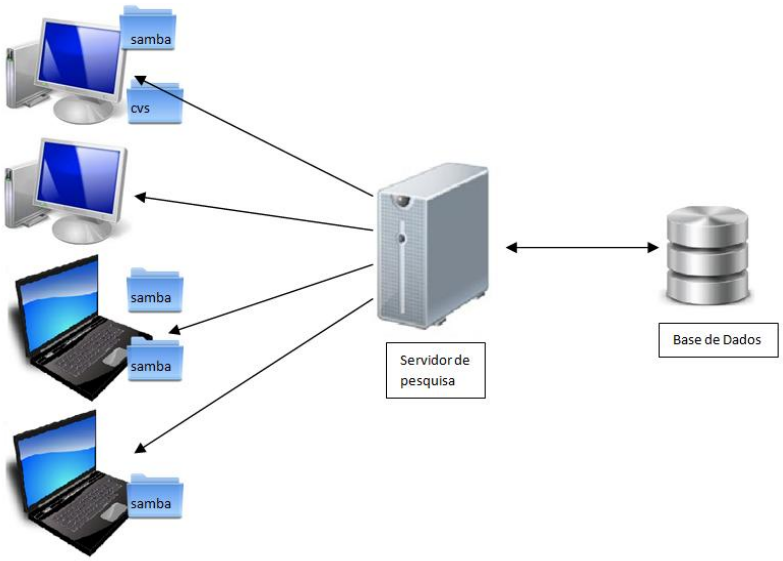

Figura 5 Implementação através de servidor central

Nesta solução existiria um servidor central que iria realizar a pesquisa por máquinas ligadas na mesma rede do servidor, fazendo posteriormente a descoberta dos repositórios de dados que são disponibilizados pelas mesmas. Durante a análise ao conteúdo dos repositórios encontrados o servidor central é responsável pela execução de possíveis plug-ins de análise e protecção de ficheiros para os quais será necessário fornecer o conteúdo dos mesmos e eventualmente proceder à sua substituição.

Esta solução tem como principal vantagem a experiencia que é possível extrair das soluções com características analisadas durante o levantamento do estado da arte (uma vez que a maioria também funciona com este tipo de arquitectura). Para além disso, não é necessário instalar nada nas máquinas a analisar, nem conhecer previamente a rede onde a pesquisa será efectuada. No entanto, esta abordagem também possui desvantagens relativamente ao nível to trafego de rede gerado, e ao tempo gasto para a realização da análise ao conteúdo partilhado - uma vez que é necessário aceder remotamente à informação (no caso de repositórios que comunicam via NetBEUI), ou fazer *download* do contudo dos repositórios (no caso de repositórios CVS).

3.2.3 ABORDAGEM ESCOLHIDA

Após ponderação e reuniões com o orientador do estágio, foi escolhida a segunda(B)) opção, em detrimento da primeira(A)) uma vez que esta opção permite:

- Conhecer os dispositivos ligados à rede – com a abordagem escolhida é possível conhecer os dispositivos ligados na rede, possibilitando assim gerar relatórios e gráficos de disponibilidade dos mesmos;

- Evitar a instalação de *software* adicional - Ao contrário da opção A esta não necessita de instalação de software nas máquinas alvo de análise;

- Conhecer o conteúdo partilhado – permite definir se apenas deve ser conhecido o conteúdo partilhado, ou se todo o conteúdo do dispositivo (no caso de um computador com conta de domínio), bastando para tal o fornecimento de credenciais de administração de domínio,

- Escalabilidade – Apesar de esta solução ser menos escalável do que a solução A), A opção permite adicionar servidores de pesquisa e análise de dados. Esta adição resulta num aumento de performance. Neste cenário os servidores partilham a mesma base de dados.

Devido à necessidade de implementação de diferentes tipos de repositórios de dados (que usam protocolos de comunicação diferentes) foi decidido pelo *Product Owner* que seria mais prioritário a implementação nativa de suporte para repositórios que comuniquem por NetBEUI (mais concretamente em repositórios Samba e Pastas partilhadas Windows) e repositórios CVS.

3.2.4 ARQUITECTURA PRELIMINAR DE ALTO NÍVEL

Antes de iniciar a fase de análise de requisitos foi definida uma arquitectura preliminar de alto nível de forma a ajudar à identificação de sistemas e actores envolvidos, tal como é demonstrado pela Figura 6.

Resumidamente, o sistema é subdividido em três módulos: O primeiro trata da descoberta automática dos dispositivos ligados à rede em que o servidor está presente. O Segundo trata da descoberta e análise de conteúdo de repositórios de dados. O terceiro trata da interface gráfica, mais concretamente através de consolas *web* de administração, que serão manuseadas pelo chefe do departamento de segurança (utilizador a quem se destina esta ferramenta).

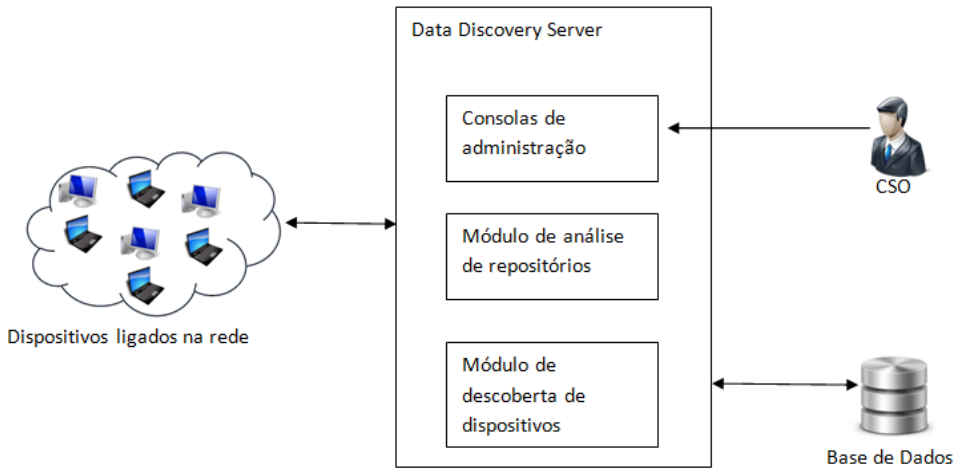

Figura 6 Arquitectura preliminar de alto nível do sistema

3.2.5 ANÁLISE DE REQUISITOS

O módulo implementado durante o estágio foi realizado segundo a metodologia de desenvolvimento *Scrum*, como tal a especificação de requisitos é feita sob a forma de *user stories*. Neste documento apenas serão apresentadas na Tabela 2 as user stories que foram implementadas durante o estágio. No entanto as restantes user stories estão disponíveis para consulta no product backlog presente no Anexo C [AD-3].

ID	User Story	Story points	Prioridade
E1.T3.US12	Como CSO quero poder associar uma gama de IPs à lista de Inclusão/exclusão para que possa cingir a pesquisa de máquinas	8	10
E1.T3.US14	Como CSO quero poder associar um MAC à lista de inclusão/exclusão para que possa cingir a análise de máquinas	3	11
E2.T4.US17	Como CSO quero que o sistema analise a rede para que possa conhecer as máquinas presentes na rede	13	1
E2.T4.US18	Como CSO quero que os dados de máquinas encontradas pelo crawler fiquem armazenados de forma persistente	8	2
E2.T4.US20	Como CSO quero poder ver uma a listagem das máquinas para que possa conhecer as máquinas encontradas na rede, mostrando e possibilitando organização por nome, host, IP, estado na última execução do sistema	5	12
E2.T7.US38	Como CSO quero que o sistema identifique as pastas partilhadas presentes numa máquina utilizando acesso GUEST/ANONYMOUS	8	3
E2.T7.US39	Como CSO quero que o sistema identifique os repositórios CVS presentes numa máquina utilizando acesso GUEST/ANONYMOUS	8	7
E2.T7.US51	Como CSO quero poder ver uma listagem dos repositórios encontrados/configurados ordenável por nome, grupo de máquinas, data de descoberta, data de último acesso	3	20
E2.T8.US56	Como CSO quero poder adicionar Credenciais de acesso para que estas possam ser utilizadas em repositórios, pastas ou ficheiros	3	18
E2.T8.US57	Como CSO quero poder apagar uma credencial de forma a remover credenciais que não sejam usadas	3	19
E2.T8.US58	Como CSO quero poder ver uma listagem de credenciais ordenável e filtrável por nome, número de utilizações, tipo de utilizações e uso por default	5	17
E2.T9US66	Como CSO quero que o sistema conheça o conteúdo presente nos repositórios CVS encontrados utilizando acesso GUEST/ANONYMOUS	5	8
E2.T9US67	Como CSO quero que o sistema conheça o conteúdo presente nas pastas partilhadas encontrados utilizando acesso GUEST/ANONYMOUS	5	4
E2.T9US70	Como CSO quero que o sistema conheça o conteúdo presente nos repositórios CVS encontrados utilizando uma credencial específica	2	14

ID	User Story	Story points	Prioridade
E2.T9US71	Como CSO quero que o sistema conheça o conteúdo presente nas pastas partilhadas (via NetBEUI) encontrados utilizando uma credencial específica	5	13
E2.T9US74	Como CSO quero que o sistema identifique o tipo de acesso permitido a pastas e ficheiros presentes num repositório CVS utilizando acesso GUEST/ANONYMOUS	5	9
E2.T9US75	Como CSO quero que o sistema identifique o tipo de acesso permitido a pastas e ficheiros presentes numa pasta partilhada (via NetBEUI) utilizando acesso GUEST/ANONYMOUS	5	5
E2.T9US78	Como CSO quero que o sistema identifique o tipo de acesso permitido a pastas e ficheiros presentes num repositório CVS utilizando uma credencial específica	3	16
E2.T9US79	Como CSO quero que o sistema identifique o tipo de acesso permitido a pastas e ficheiros presentes numa pasta partilhada (via NetBEUI) utilizando uma credencial específica	5	15

Tabela 2 User stories implementadas

Além dos requisitos funcionais abordados pelas user stories existem ainda os requisitos não funcionais traçados para o projecto, mais concretamente ao nível de:

- Expansibilidade – A arquitectura da ferramenta implementada deverá permitir a expansão para outros tipos de repositórios de dados;
- Performance – A ferramenta criada deverá permitir a adição de servidores adicionais de forma a aumentar a performance do sistema;
- Auditoria – A ferramenta deverá registar todas as operações realizadas automaticamente pela aplicação, sejam estas análises, execuções de plug-ins ou substituição de ficheiros. Deverão ainda ser registadas as operações feitas na interface gráfica que interfiram com as análises feitas à rede, como por exemplo: alteração das zonas da rede analisáveis, alteração das máquinas analisáveis, etc.

3.2.6 ESPECIFICAÇÃO DE TESTES DE ACEITAÇÃO

De forma a garantir o correcto funcionamento da aplicação durante as *Sprints, foram definidos* testes de aceitação para cada uma das user stories implementadas. Estes testes permitiram validar a correcta implementação das user stories e avaliar o seu grau de *achievement* no final de cada *Sprint*.

Foi então definido um plano de testes onde constam as seguintes informações:

- Definição de ambiente de testes;
- *User stories* a testar;
- Abordagem;
- Critérios de aceitação dos testes;
- Riscos e respectivos planos de contingência.

Para consulta da documentação detalhada da especificação relativa aos casos de teste deve ser consultado o Anexo B – Especificação de casos de Teste [AD-2]. Estes testes foram executados pelo estagiário durante as *sprints* realizadas. Os seus resultados foram validados pelo *Product Owner* durante as reuniões de *Sprint Review*.

3.3 ARQUITECTURA E DESENHO DETALHADO

3.3.1 INTRODUÇÃO

Nesta secção (3.3) é apresentada a descrição sobre a forma como a aplicação foi projectada, nomeadamente o seu desenho detalhado e arquitectura. Estes elementos serviram como guia durante a implementação *das user stories* seleccionadas, garantindo assim a sua compatibilidade com futuras funcionalidades (presentes no product backlog). Além disso, serão também apresentados as várias soluções encontradas, de forma a responder aos desafios técnicos que surgiram durante a fase de desenvolvimento da aplicação.

3.3.2 ARQUITECTURA

O desenho da arquitectura de um projecto é uma questão fundamental, uma vez que tem implicações que podem afectar o desenvolvimento das futuras versões do produto. Como tal , esta deve ser o mais flexível e extensível possível. Assim, e de forma a facilitar a compreensão do sistema, optou-se pela apresentação de duas vistas de arquitectura:

- Vista física/instalação – mostra o que é necessário em termos de *hardware* e *software* para a utilização do sistema;

- Vista Lógica/Funcional – mostra a forma dos diferentes componentes comunicarem entre si;

3.3.2.1 Arquitectura Física

O projecto desenvolvido encontra-se dividido em vários módulos distintos. Cada módulo foi desenvolvido de forma a ser independente dos restantes.

A aplicação pode ser decomposta em quatro (4) servidores distintos: O *web server,* o servidor de descoberta de máquinas, o servido de análise de repositórios e o servidor de bases de dados. De uma forma resumida:

- Servidor de descoberta de máquinas – este servidor é o responsável pela realização da análise à disponibilidade das máquinas na rede. Posteriormente comunica com o servidor de base de dados, por forma a garantir a persistência da informação recolhida. Este servidor necessita da instalação do NMAP para a realização das tarefas de pesquisa.
- Servidor de análise de repositórios – este servidor é o responsável pela descoberta e análise do conteúdo dos repositórios partilhados na rede. Neste servidor existe um serviço que realiza

periodicamente a procura e análise de repositórios partilhados na rede. Este servidor contém os *plug-ins* de descoberta e análise de repositórios, e de análise e protecção de ficheiros. Para o correcto funcionamento destes componentes, é necessária a instalação do CVSNT (cliente CVS) no servidor

- Web Server – Este servidor alberga os módulos associados à interface gráfica do utilizador da aplicação, especificamente, as consolas de administração. Este servidor necessita de ter o IIS instalado e configurado, uma vez que as consolas de administração foram desenvolvidas em ASP.NET.
- Servidor de base de dados – Esta máquina alberga as bases de dados requeridas pela aplicação. Neste deverá estar instalado o SQL Server 2008 uma vez que a aplicação desenvolvida o utiliza para suportar as bases de dados.

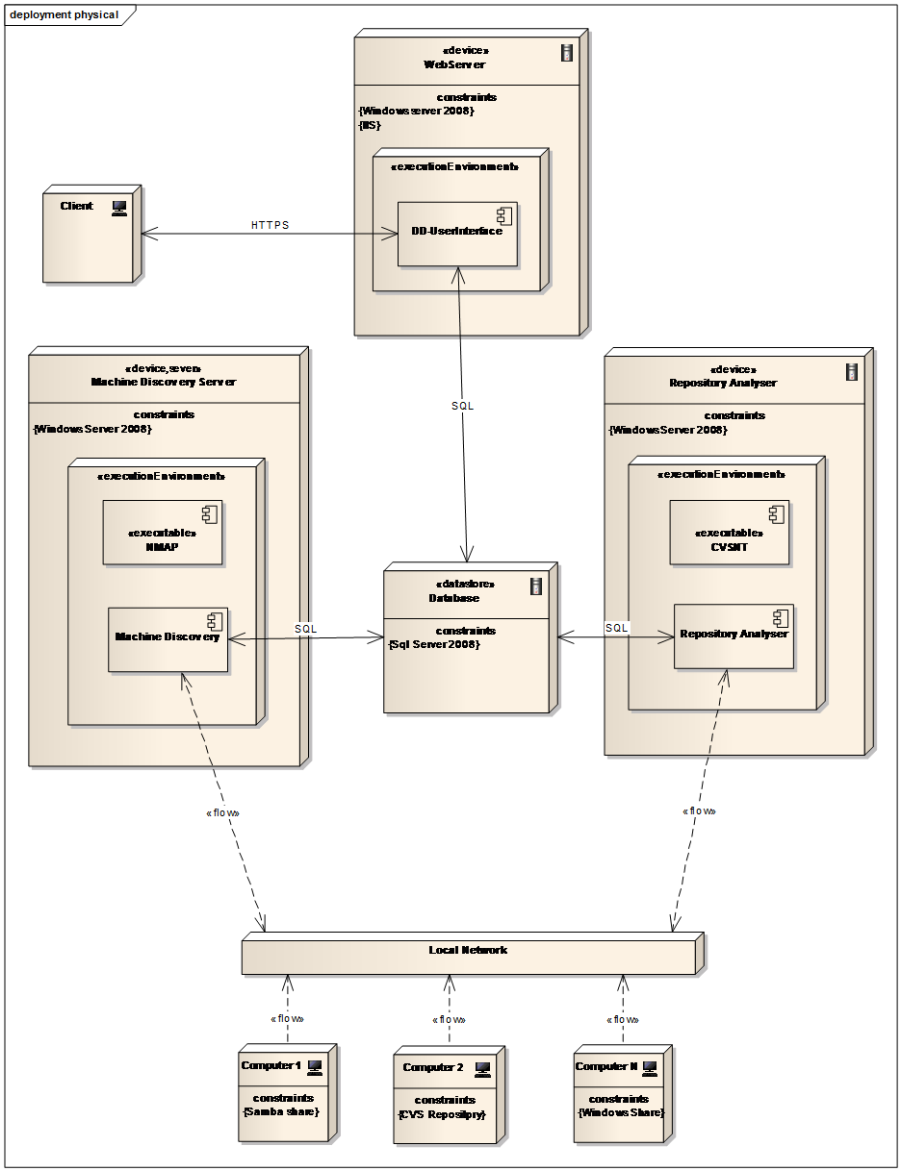

Figura 7 Vista física da arquitectura - Diagrama de Instalação

Na Figura 7 é possível analisar com mais detalhe a descrição que foi feita anteriormente. Esta divisão em diferentes componentes permite a sua distribuição por diferentes máquinas, aumentando assim a performance geral da aplicação. A arquitectura desenhada representa um dos possíveis cenários de utilização, uma vez que o sistema desenhado permite a adaptação a diferentes condições e composições de *hardware*. Além disso é possível combinar todos os componentes de diferentes formas, podendo inclusivamente estar todos em apenas uma máquina, centralizando assim toda a lógica.

3.3.2.1.1 Tecnologias utilizadas

O módulo de Automated Data Discovery and Protection do csSECURE é suportado pelas seguintes tecnologias:

- Windows Server 2008 – sistema operativo utilizado para correr a ferramenta desenvolvida. Este sistema operativo foi escolhido devido ao facto de a ferramenta desenvolvida estar integrada no projecto csSECURE, que por sua vez pressupõe um ambiente Windows nos seus servidores de suporte.

- NMAP – Esta ferramenta é utilizada para a descoberta de dispositivos na rede. Tem por objectivo apenas a descoberta das máquinas activas e respectivos o seu *hostname*, IP e *mac address*). Apesar de esta ferramenta realizar *port scanning* para descoberta dos serviços que correm nas máquinas não foram utilizadas estas funcionalidades.

- CVSNT- Cliente CVS utilizado para o acesso ao conteúdo de repositórios cvs. Esta ferramenta foi utilizada como cliente cvs de forma a obter informação acerca do conteúdo dos repositórios CVS.

- C# - A aplicação foi desenvolvida na linguagem de programação C#. A razão subjacente a esta escolha deve-se ao facto deste módulo fazer parte do csSECURE, que pressupõe a utilização de servidores Windows. Como tal existia a necessidade de desenvolver especificamente para este tipo de plataformas Windows, levando assim a esta escolha.

- SQL Server 2008 – Tecnologia usada para suportar a base de dados da aplicação. A escolha recaiu nesta plataforma uma vez que também é utilizada no projecto csSECURE.

- IIS+ASP.NET – Foi utilizado um servidor ISS com ASP.NET de forma a suportar as consolas de administração. Foram escolhidas estas tecnologias uma vez que se enquadram com o ambiente de desenvolvimento (Windows). Para além disso, esta é a tecnologia utilizadas no csSECURE, o que facilitará uma futura integração.

3.3.2.2 Arquitectura Funcional

A definição da arquitectura funcional pode ser descrita pelo diagrama de componentes ilustrado na Figura 8. Este diagrama contém os principais componentes utilizados para o desenvolvimento da solução apresentada. Seguidamente é apresentada uma descrição de cada um dos componentes desenhados.

- Componentes comuns – conjunto de componentes partilhados pelos módulos de interface gráfica, Machine Discovery e Repository Analyser. Nestes componentes estão incluídas a representação das entidades (máquina, pasta, ficheiro, etc.)as operações de *logging,* e de persistência dos dados que são partilhados pelos diferentes módulos.
- Machine Discovery – Este componente, em parceria com o NMAP, realiza a análise às máquinas ligadas à rede local
- Repository Analyser – Este módulo é o responsável pela descoberta e análise de pastas Samba, pastas partilhadas Windows e CVS. É através deste componente que se faz a ligação quer com os *plug-ins* de análise e protecção de ficheiros quer com os *plug-ins* de descoberta e análise de repositórios não suportados.
- *Interface* para *plug-in* de análise de repositórios não suportados – Esta *interface* oferece a possibilidade de criação de *plug-ins* com vista à descoberta e análise de repositórios não suportados nativamente pela aplicação. Estes *plug-ins* devem realizar tarefas de descoberta e análise de um determinado tipo de repositório de dados. Para além disso também devem implementar métodos para fazer a obtenção e substituição de um determinado ficheiro presente num repositório de dados analisado pelo plug-in;
- Interface para *plug-in* de análise e protecção de ficheiros – Esta *interface* disponibiliza a possibilidade para a criação de *plug-ins* de análise de conteúdo e protecção de ficheiros. Para tal são fornecidos aos *plug-ins* os ficheiros, juntamente com as informações recolhidas sobre os mesmos de modo a que estes os possam analisar e em caso de necessidade pedir a sua substituição nos repositórios em que estes se encontram.
- DD-Interface – Este componente é responsável pela interacção com o utilizador. É através deste componente que são realizadas as tarefas de configuração da aplicação, apresentação dos resultados das análises e monitorização das operações realizadas.

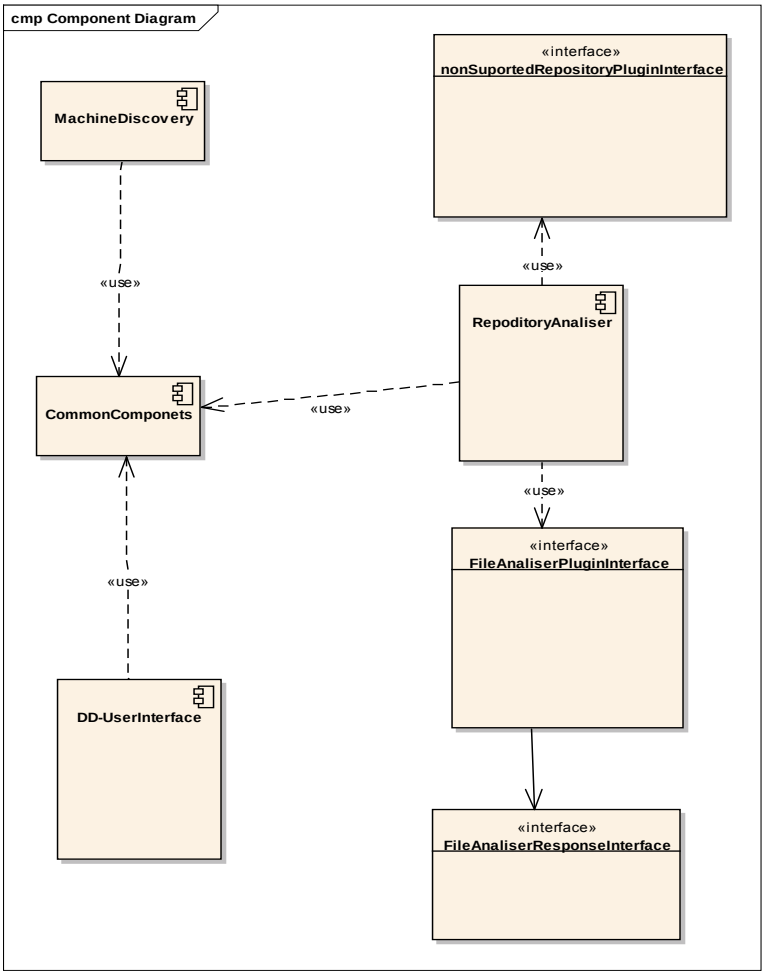

Figura 8 Diagrama de componentes

3.3.3 DESENHO DETALHADO

Para a especificação detalhada do *design* recorreu-se ao desenho de diagramas de *packages* e classes de modo a representar os diferentes componentes identificados na arquitectura.

3.3.3.1 Componentes Comuns

Na Figura 9 encontra-se representado o diagrama de *packages* referente aos componentes comuns. Este componente encontra-se dividido em 4 packages:

- BLL (*Business Logic Layer*) – Camada responsável pelo tratamento dos pedidos de dados da aplicação feitos pelos restantes componentes da aplicação. As classes presentes nesta *package* recebem os pedidos de dados vindos dos restantes componentes da aplicação e tratam dos pedidos recorrendo às classes presentes na *package* DAL, garantindo assim uma camada de abstracção entre os restantes componentes e os acessos à base de dados;

- DAL (*Data Access Layer*) – Camada responsável pela comunicação com a base de dados. É através das classes representadas nesta camada que se realizam todas as operações com a base de dados da aplicação.

- BO (*Business Object*) – Camada que contem os objectos representantes das entidades de negócio. Estas entidades são partilhadas por todos os componentes da aplicação.

- LOG – Camada responsável pelo tratamento das tarefas de *logging da* aplicação.

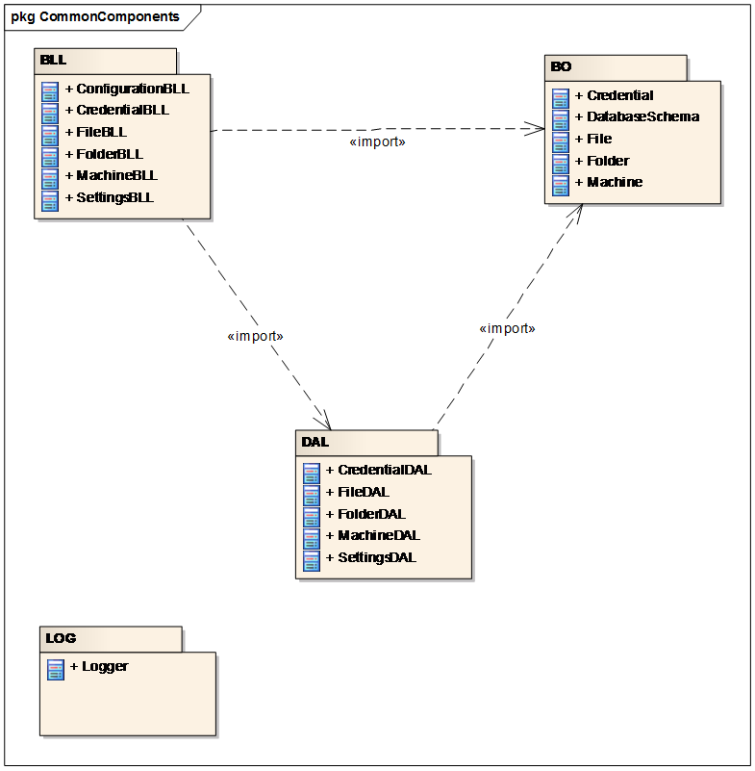

Figura 9 Diagrama de *packages* dos componentes comuns da aplicação

3.3.3.2 Descoberta de Máquinas

Este componente é responsável pela descoberta de máquinas na rede. Para o desenho deste componente optou-se pela separação em 3 *packages*:

- BLL (*Business Logic Layer*) – Camada responsável pela implementação das funcionalidades. É através das classes desta package que se realizam as tarefas de descoberta de máquinas e são feitos os pedidos de persistência dos dados encontrados (recorrendo quer aos componentes comuns quer através da classe RestrictionsBLL)

- DAL (*Data Access Layer*) – Camada responsável pela comunicação com a base de dados. Esta camada realiza as tarefas de comunicação com a base de dados não incluídas nos componentes comuns.

- BO (*Business Object*) – Camada que contem as instâncias das classes representantes das entidades de negócio necessárias ao componente em questão e que não se encontram incluídas nos componentes comuns.

Esta separação pode ser observada na Figura 10.

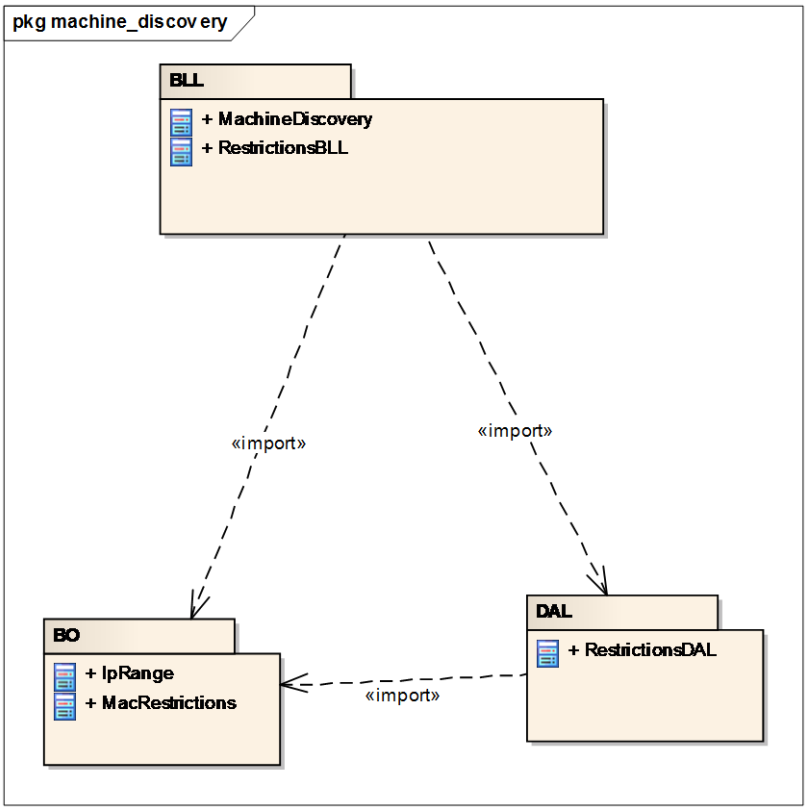

Figura 10 Diagrama de *packages* do componente de descoberta de máquinas

3.3.3.3 Análise de repositórios

Este componente encontra-se dividido em 4 packages, tal como pode ser verificado no diagrama ilustrado pela Figura 11. Estas *packages* implementam as seguintes funcionalidades:

- SMB – trata das funcionalidades relacionadas com a descoberta e análise de repositórios que comunicam por NetBEUI;
- PluginSuport – package responsável pela comunicação com os plug-ins existentes;
- CVS – Package responsável pelas operações de descoberta e análise de repositórios cvs;
- RepositoryAnalyser – Package responsável pela gestão da execução das análises feitas aos repositórios.

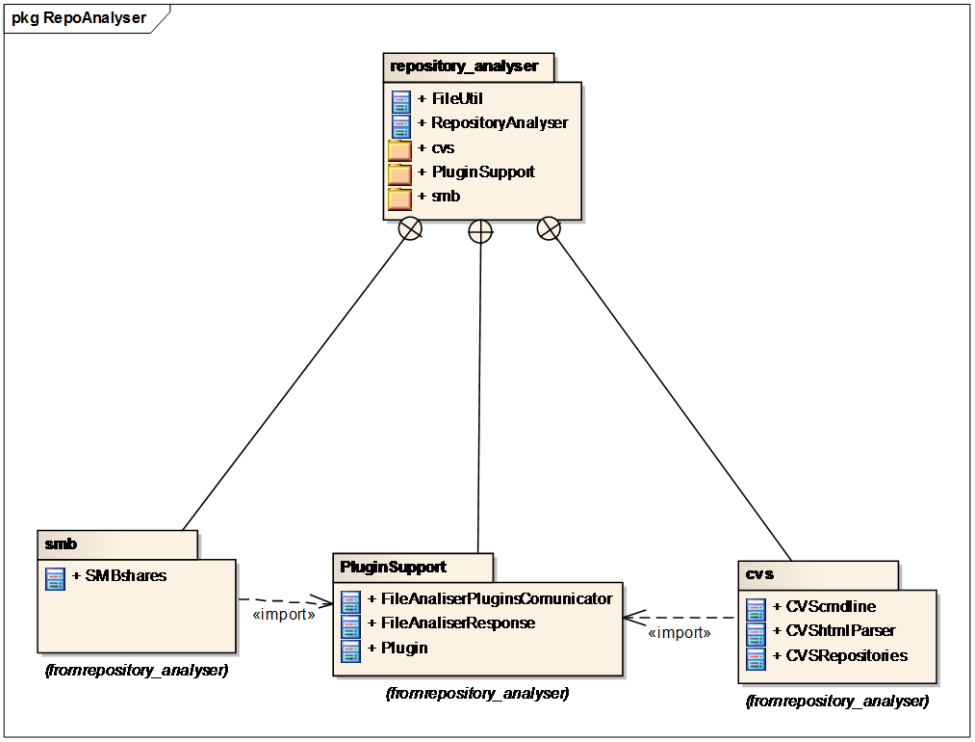

Figura 11 Diagrama de packages do componente de análise de repositórios

3.3.3.4 Consolas de administração

Este componente pode ser dividido em duas *subpackages*:

- Aspxop – Nesta subpackage são tratados os pedidos realizados pela interface gráfica.
- App_GlobalResources – Nesta subpackage encontram-se os recursos necessários para a globalização da interface gráfica.

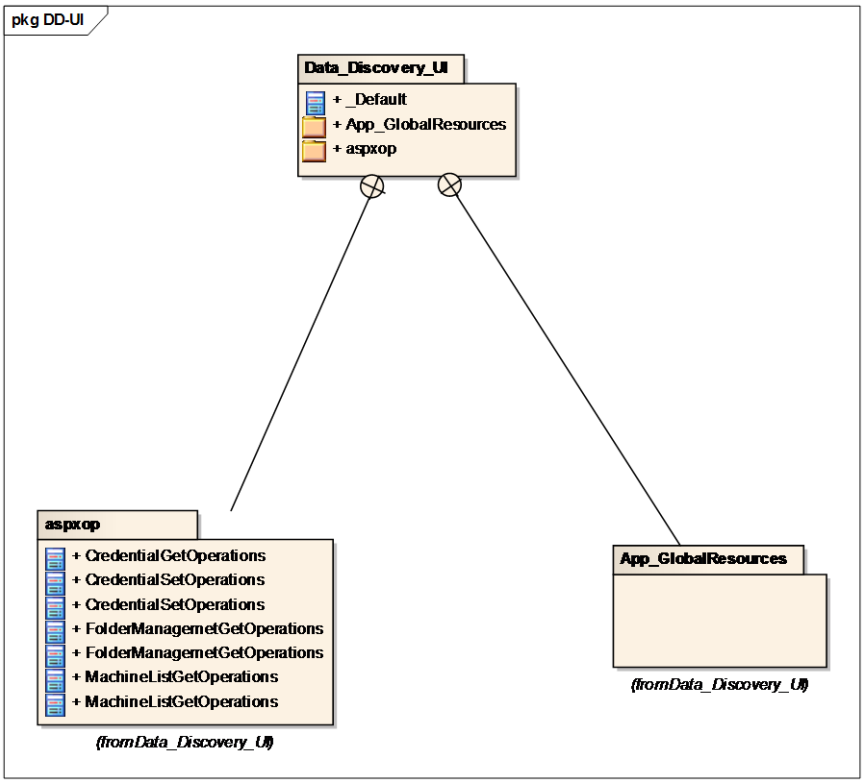

Figura 12 Diagrama de packages do componente de interface gráfica

3.3.3.5 Interfaces de suporte a plug-ins

Para especificação detalhada do design das interfaces para suporte de plugins (de análise de repositórios e de análise e protecção de ficheiros) recorreu-se a diagramas de classes que serão apresentados nas subsecções seguintes.

3.3.3.5.1 Análise de repositórios

De forma a permitir a análise de repositórios sem suporte nativo foi criada uma *interface* que define quais as propriedades e métodos que devem ser implementados por eventuais *plug-ins*. Deste modo a permitiu-se uma expansão das funcionalidades de forma mais fácil. Da especificação da *interface* resultou a classe ilustrada pela Figura 13.

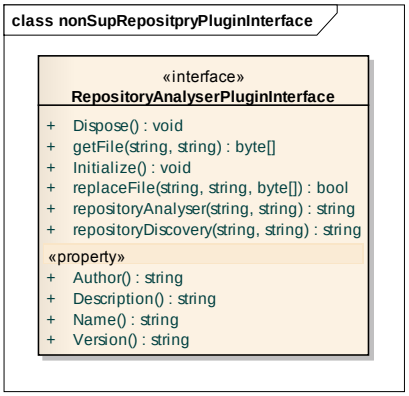

Figura 13 *Interface* para análise de repositórios de dados

Foram também definidos esquemas XSD de forma a garantir a uniformização e validação dos dados transmitidos entre aplicação e os *plug-ins*.

3.3.3.5.2 Análise e protecção de ficheiros

De modo a possibilitar a criação de *plug-ins* de análise e protecção de ficheiros, foi definida uma *interface* para servir de guia ao seu desenvolvimento. Essa *interface* encontra-se retratada no diagrama de classes representado pela Figura 14.

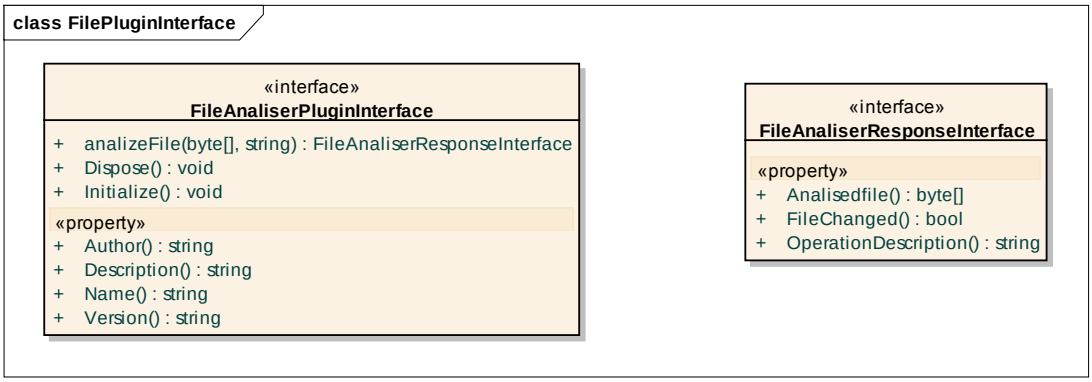

Figura 14 Diagrama de Classes das *interfaces* necessárias aos plug-ins de análise e protecção de ficheiros

Os *plug-ins* necessitarão de implementar os métodos e propriedades definidas pelas interfaces. Para além disso terão ainda de obedecer a esquemas XSD definidos para a transmissão de dados entre a aplicação e o *plug-in*. Seguidamente é apresentado um exemplo dos esquemas XSD acima mencionados.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="file-description">
        <xsd:complexType>
            <xsd:element name="file_id" type="xsd:int" />
            <xsd:element name="file_name" type="xsd:string" />
            <xsd:element name="location" type="xsd:string" />
            <xsd:element name="repository_type" type="xsd:string" />
            <xsd:element name="last_access" type="xsd:date" />
            <xsd:element name="read_permissions" type="xsd:boolean" />
            <xsd:element name="write_permissions" type="xsd:boolean" />
            <xsd:element name="classification" type="xsd:int" />
            <xsd:element name="folder">
                <xsd:complexType>
                    <xsd:element name="read_permissions" type="xsd:boolean" />
                    <xsd:element name="write_permissions" type="xsd:boolean" />
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="machine">
                <xsd:complexType>
                    <xsd:element name="ip" type="xsd:string" />
                    <xsd:element name="mac" type="xsd:string" />
                    <xsd:element name="host" type="xsd:string" />
                </xsd:complexType>
            </xsd:element>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Figura 15 XSD de validação de ficheiro XML com informação sobre um ficheiro

3.4 IMPLEMENTAÇÃO

Após a fase de desenho da arquitectura que viria a servir de base à implementação da ferramenta, segue-se a fase de implementação da mesma. Durante a fase de implementação, devido ao tempo limitado para a mesma, decidiu-se executar apenas as user stories mais prioritárias para o produto. Estas prioridades resultaram na implementação de apenas algumas partes dos módulos desenhados.

No que refere ao módulo de descoberta automática de máquinas na rede, foram implementados os comportamentos considerados mais importantes, ou seja, foram implementadas as funcionalidades de descoberta de *hosts,* e de limitação das zonas de análise.

Quanto ao módulo de análise de repositórios, foi implementado o acesso e análise repositórios acessíveis via NetBEUI, e CVS. Além disso foi implementado o suporte para *plug-ins* que realizem a análise e protecção dos ficheiros encontrados.

Nas consolas de administração foram implementadas as funcionalidades necessárias para a visualização dos resultados das análises efectuados à rede. Além disso foram implementadas funcionalidades para gestão de credenciais de aceso a repositórios, de forma a permitir que estas possam ser utilizadas pelo módulo de descoberta e análise de repositórios.

Na secção 4 será apresentada uma descrição mais detalhada dos resultados obtidos através desta implementação.

3.5 PROBLEMAS SUPERADOS

Nesta secção aborda desafios técnicos encontrados ao longo da definição da arquitectura e da implementação do sistema, indicando as soluções propostas.

3.5.1 ARQUITECTURA MODULAR

3.5.1.1 Problema

Devido ao facto de se pretender uma solução que no futuro venha a suportar diversos tipos de repositórios de dados e tendo em conta que se decidiu pela implementação inicial dos repositórios com comunicação via NetBEUI e repositórios CVS, surgiu a necessidade de criar uma solução que permitisse uma expansão de forma simples.

3.5.1.2 Solução

A forma encontrada para facilitar o desenvolvimento de módulos de suporte para outros tipos de repositórios, foi a criação de *interfaces* de forma a possibilitar a criação de *plug-ins* de análise de outros tipos de repositórios de dados.

Os *plug-ins* para análise de repositórios são responsáveis pelas tarefas definidas nas suas interfaces, tais como:

- Descoberta de repositórios de dados
- Análise ao conteúdo de um repositório de dados;
- Obtenção de um ficheiro específico de um repositório de dados;
- Substituição de um ficheiro específico num repositório de dados.

3.5.2 CONTA GUEST/CONVIDADO EM WINDOWS

3.5.2.1 Problema

Devido a falhas de segurança detectadas no sistema operativo Windows, a Microsoft decidiu colocar por omissão a conta Guest/Convidado inactiva. Cabendo aos administradores das respectivas máquinas a activação desta conta. Surgiu então a necessidade de contornar este problema uma vez que uma das user stories seleccionadas requeria a utilização deste tipo de conta para acesso a pastas partilhadas.

3.5.2.2 Solução

A abordagem escolhida passa pela utilização de uma credencial de domínio fornecida durante a instalação da aplicação. Ao utilizar uma credencial de domínio é permitido o acesso às pastas partilhadas publicamente no domínio da onde está inserido o servidor.

3.5.3 DESCOBERTA DE REPOSITÓRIOS CVS

3.5.3.1 Problema

Para saber verificar se existe um repositório CVS numa determinada máquina, é necessário saber o nome do mesmo e saber qual o protocolo através do qual este está acessível.

3.5.3.2 Solução

Uma vez que para verificar a existência de repositórios CVS é necessário o conhecimento prévio do nome do mesmo, foi decidida a realização de testes com os nomes de repositórios CVS mais comuns. Como tal optou-se por testar os nomes que vem por omissão nos diferentes instaladores de servidores CVS, ou que foram encontrados em algumas pesquisas realizadas na internet tais como: “myrepos”, ”cvs”, “prod”, “src”, “source”, “devel”, “test”, “cvsroot”, “cvsntroot”, “demo”, “cvs-repository”, “var/lib/cvsroot”, “/var/cvs”, “/home/cvsroot” e “/usr/local/cvs”.

4 Resultados

Nesta secção é feita a descrição dos resultados produzidos ao longo do projecto.

No âmbito do projecto foram desenvolvidas as tarefas avaliadas como mais prioritárias de forma a criar um mecanismo que permitisse a descoberta e protecção dos documentos partilhados no seio de uma organização. Desse trabalho resultou a implementação dos seguintes módulos:

- **Machine Discovery** Serviço que realiza a procura de máquinas na rede, segundo critérios definidos nas configurações da aplicação.
- **Repository Analyser** – Serviço de descoberta e análise de repositórios de dados.
- **Consolas de Administração** – Interface gráfica que permite a configuração da aplicação e visualização dos resultados obtidos pelos módulos de descoberta de máquinas e de análise de repositórios.

4.1 MACHINE DISCOVERY SERVICE

Este módulo efectua a consulta das configurações presentes na base de dados. Também realiza a pesquisa por máquinas activas na rede. Inicialmente verifica a existência de zonas de inclusão ou exclusão na base de dados, seguidamente calcula as zonas em que deve ser realizada a descoberta de máquinas, sendo as restantes zonas marcadas como indisponíveis para análise. O módulo foi implementado de acordo com as especificações de desenho detalhado anteriormente apresentadas (secção 3.3.3.3), recorrendo ao NMAP para a descoberta de máquinas activas.

Este módulo foi testado na rede interna da Critical Software de Coimbra que contempla cerca de 1000 IPs disponíveis, tendo demorado em média setenta e sete segundos (77s) a identificar em média duzentas e setenta e seis (276) máquinas, tal como pode ser verificado na Tabela 3.

Análise	Inicio de análise	Fim de análise	Tempo decorrido (segundos)	Máquinas encontradas
1	09:02:56	09:04:12	76,00	233
2	12:32:20	12:33:34	74,00	299
3	11:47:04	11:48:26	82,00	300
4	17:16:25	17:17:42	77,00	301
5	19:02:41	19:03:57	76,00	229
		Média (segundos)	77	276
		Desvio Padrão (segundos)	3	33

Tabela 3: Teste de performance à descoberta de máquinas

4.2 REPOSITORY ANALYSER

Este módulo é o responsável pela análise ao conteúdo partilhado nas máquinas disponíveis para análise. É através deste módulo que é feita a ela descoberta e análise de pastas partilhadas e de repositórios cvs na rede. Além disso é através deste módulo que são executados os plug-ins de análise e protecção de ficheiros.

Este módulo começa por verificar quais as máquinas que são analisáveis, ou seja, que não estão em zona de exclusão ou que foram marcadas pelo módulo de descoberta de máquinas como indisponíveis na última análise. Após a obtenção desta lista, efectua a procura de pastas que comuniquem via NetBEUI, recorrendo às diferentes credenciais de acesso disponíveis. Terminada a descoberta de repositórios inicia-se a análise do seu conteúdo e direitos de acesso. Caso existam plug-ins de análise e protecção de ficheiros serão executados e em alguns casos os ficheiros serão substituídos (como por exemplo no caso de o ficheiro ser cifrado pelo plug-in).

Terminada a análise às pastas partilhadas repete-se o processo para repositórios CVS.

De forma a testar este módulo em ambiente de produção fez-se uma análise aos servidores de apoio ao csSECURE, aos computadores dos elementos do projecto. Nos doze computadores analisados existiam quatro pastas partilhadas Windows e um repositório CVS, tal como pode ser verificado na

Tipo de repositórios	Número de repositórios	Número de pastas	Número de ficheiros	Tamanho total
CVS	1	4810	27917	4.71GB
SMB	5	9670	83213	175,43 GB

Tabela 4 Conteúdo analisado pelo módulo de análise de repositórios

Da análise efectuada resultaram

Análise	Inicio de análise	Fim de análise	Tempo decorrido(minutos)
1	10:29:53	12:44:34	30
2	18:15:37	20:11:09	135
3	14:15:49	16:22:03	116
4	21:54:30	23:53:15	126
5	11:03:52	13:20:21	119
		Média	126
		Desvio Padrão	9

4.3 CONSOLAS DE ADMINISTRAÇÃO

Este módulo é responsável pela configuração e apresentação dos resultados recolhidos pelos módulos de análise.

Deste módulo apenas foram implementados as funcionalidades de gestão de credenciais, de listagem da informação das máquinas encontradas e da apresentação do conteúdo descoberto nos repositórios analisados. Para tal foram implementadas 3 tabs que no futuro corresponderão à gestão de máquinas, gestão de credenciais, e gestão de repositórios.

Na Tab de gestão de credenciais, são permitidas operações de adição, remoção e a visualização da utilização das credenciais - tal como pode ser verificado na Figura 16. Para além das operações mencionadas, também é permitida a pesquisa e ordenação pelos diferentes campos, de forma a facilitar a sua consulta.

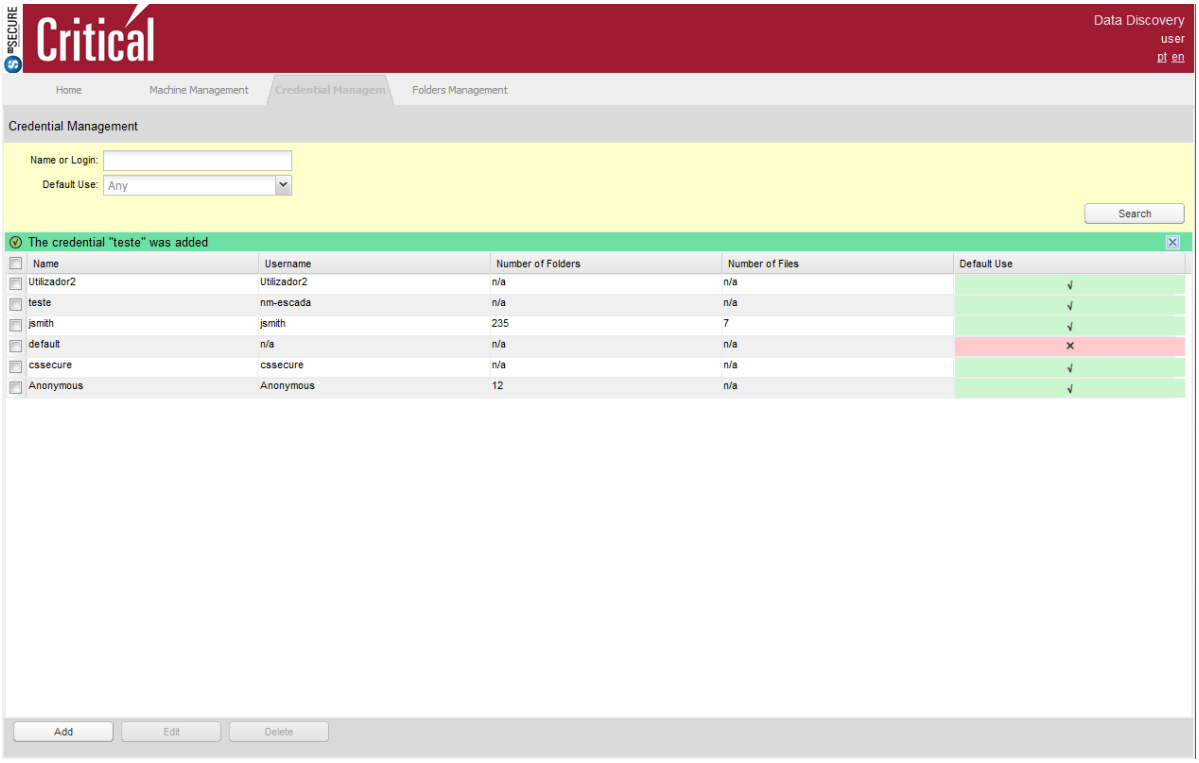

Figura 16 Tab de gestão de credenciais

Tal como no caso das credenciais, a apresentação dos resultados da análise às máquinas disponíveis na rede (Figura 17) também é apresentada numa grelha ordenável pelos diferentes campos apresentados.

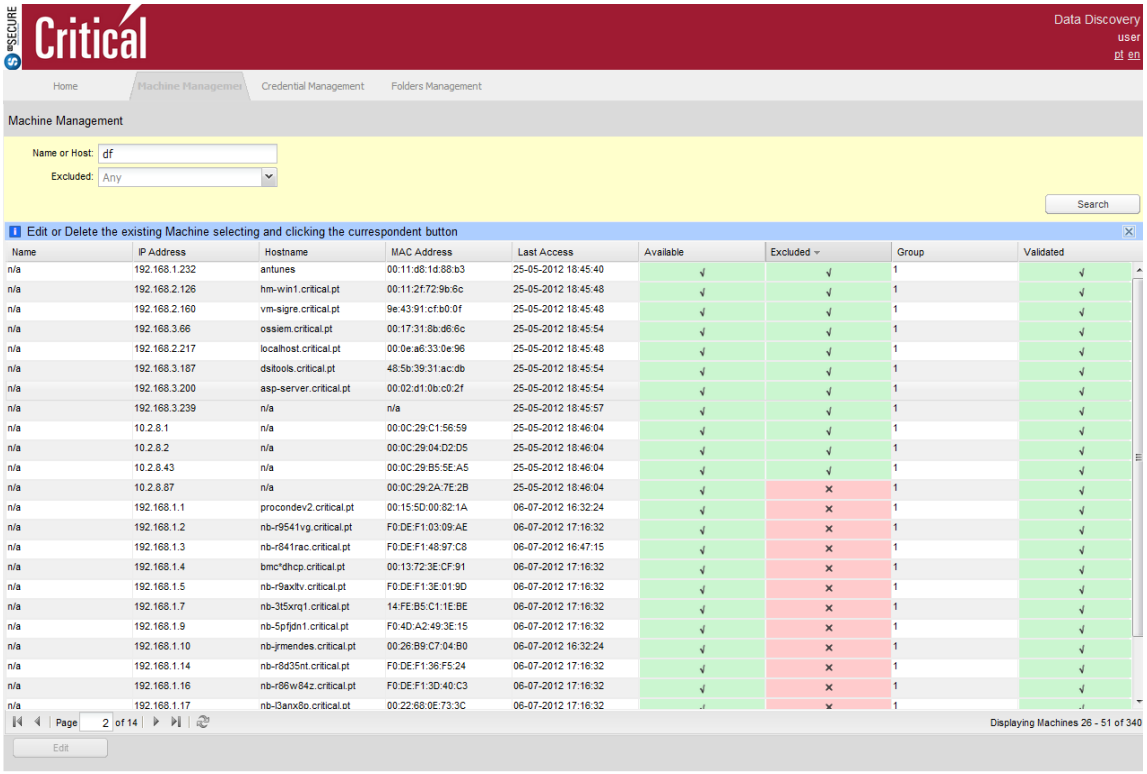

Figura 17 Tab de gestão de máquinas

Na *Tab* de gestão de repositórios apenas se encontra implementada a funcionalidade relacionada com a visualização dos repositórios e respectivo conteúdo. Para tal optou-se pelo desenvolvimento de uma vista ao estilo do explorador do Windows (Figura 18), uma vez que reduz o período de aprendizagem. Nesta *tab* para além da possibilidade de percorrer os repositórios encontrados também permite visualizar os detalhes relativos ao conteúdo dos mesmos. Detalhes esses que variam desde as permissões de acesso à informação sobre a credencial utilizada para aceder a esse conteúdo.

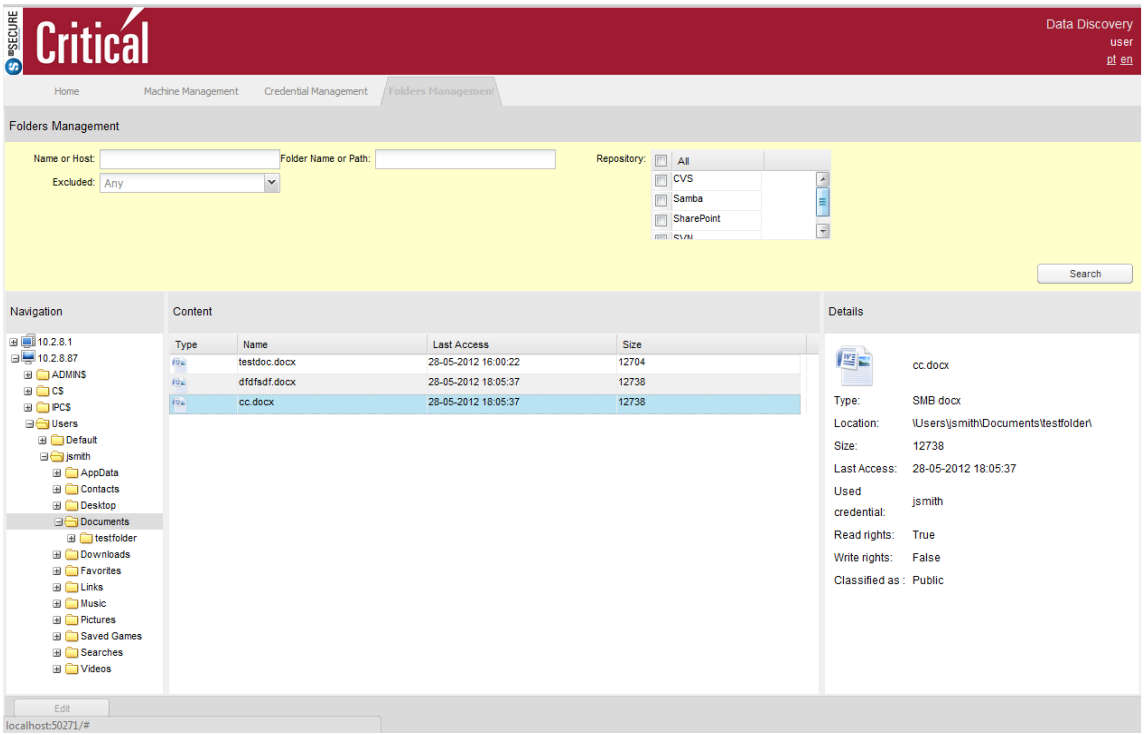

Figura 18 Tab de apresentação de conteúdo de repositórios

5 Conclusões

5.1 CONCLUSÃO

Ao longo das várias secções do documento foi analisado e exposto o trabalho realizado no âmbito da disciplina de Estágio do Mestrado em Engenharia informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Foram identificados objectivos, analisado o mercado, identificados os requisitos, delineada a arquitectura e, finalmente, apresentados alguns problemas e opções tomadas ao longo do estágio.

Concluído o projecto de estágio, fazendo uma retrospectiva do trabalho desenvolvido, pode-se afirmar que os objectivos foram maioritariamente cumpridos. Foi criada uma solução que permite fazer a descoberta de conteúdo partilhado numa rede. Disponibilizando-o a uma ferramenta externa para respectiva análise e protecção do seu conteúdo. Esta solução ainda possui algumas lacunas, nomeadamente ao nível da interface gráfica. Algumas dessas lacunas já se encontram identificadas priorizadas para uma futura implementação.

Relativamente ao conhecimento adquirido ao longo do estágio, podemos destacar a aprendizagem de novas tecnologias, que até então eram praticamente desconhecidas para o estagiário, podendo vir a ser útil em trabalhos futuros. Para além disso este estágio permitiu uma primeira experiência profissional num ambiente de produção, o que se traduziu num aumento da independência e autonomia durante a realização das tarefas, traduzindo-se numa melhor preparação para a nova etapa que se aproxima.

Esta experiência permitiu o conhecimento de uma realidade distinta da enfrentada num ambiente académico, onde na sua maioria os projectos estão bem estruturados e definidos. Durante o estágio foram verificadas algumas situações em que a solução não foi encontrada facilmente, levando a tomadas de decisão que originaram alterações no planeamento de forma a tentar não comprometer os objectivos definidos.

A nível pessoal, o estágio foi um desafio interessante, pois apesar de terem ocorrido alguns desvios e redefinições, se considerarmos os factores anteriormente mencionados podemos concluir que o balanço foi francamente positivo, principalmente se tivermos em conta a experiência no meio empresarial, onde foi possível aplicar o conhecimento adquirido ao longo do percurso académico.

5.2 TRABALHO FUTURO

Tal como pode ser verificado no *product backlog* (Anexo C [AD-3]) já existe trabalho planeado para o futuro, que consiste maioritariamente nas áreas seguintes:

- Suporte para outros tipos de repositórios de dados – Neste momento a ferramenta possui suporte para análise de apenas dois tipos de repositórios de dados (CVS e Pastas partilhadas via NetBEUI). No entanto, ainda existem alguns tipos de repositórios de dados cuja análise poderá ser interessante, como por exemplo FTP, NFS, SVN ou ainda AFP. Este tipo de suporte pode ser feito directamente na aplicação ou através da criação de plug-ins;

- Relatórios e notificações – Encontram-se planeadas várias funcionalidades relativamente à criação de relatórios e notificações de forma a facilitar o trabalho do utilizador deste software, essas funcionalidades podem ir desde a notificação da criação de novos repositórios, da

indisponibilidade de uma determinada máquina ou repositórios, criação de relatórios com gráficos com a disponibilidade das máquinas e repositórios, entre outros.

- *Logging d*e actividades – Seria proveitoso criar um registo com as actividades executadas, quer pela aplicação, quer pelo utilizador. Este registo serviria principalmente para realizar tarefas de auditoria às funções realizadas pelo utilizador do *software* e às funções realizadas pela aplicação (directamente ou pelos plug-ins);

- Interface gráfica – Esta é uma área que será muito importante para o futuro, uma vez que existem funcionalidades que já se encontram desenvolvidas mas que necessitam da correspondente implementação na interface gráfica para possam ser utilizadas. A falta de interface gráfica pode ser verificada por exemplo na gestão de restrições de análise ou ainda a adição manual de repositórios de dados.

- Além disso existe ainda a necessidade de implementação da novas interfaces gráfica para funcionalidades a serem desenvolvidas no futuro. Não esquecer que esta é uma área que deve ser tratada com especial cuidado, pois é a face visível da ferramenta.

Em suma, apesar de uma parte da ferramenta já se encontrar desenvolvida, ainda há muito trabalho que pode ser desenvolvido no âmbito desta ferramenta, de forma a tornar o trabalho desenvolvido num produto apelativo para o mercado.

6 Glossário

AFP (Apple Filing Protocol) – Protocolo de rede que permite a partilha de ficheiros incluído no Mac OS X.

Crawler – Aplicação que procura dados na rede de forma automática.

CVS (Concurrent Version System) – Sistema de controlo de versões muito usado para composição colaborativa de software e documentação.

Enterprise Search – Ferramentas de pesquisa vocacionadas para o ambiente empresarial. Estes sistemas analisam informação estruturada e não estruturada proveniente de sistemas de ficheiros, intranets, sistemas de gestão de documentos, etc.

FTP (*File Transfer Protocol)* – *Protocolo de partilha de arquivos na rede.*

Hostname – Nome atribuído a um dispositivo que serve para a sua identificação na rede.

NFS (**Network File System**) – Sistema de arquivos distribuídos desenvolvido pela Sun Microsystems

Product backlog – Documento que representa o conjunto pormenorizado de funcionalidades que se espera vir a incluir num produto.

Samba – Implementação do protocolo SMB para sistemas Unix.

SMB (Server Message Block) – Protocolo de rede utilizado para partilha de pastas, ficheiros, etc.

SVN (Apache Subversion) – Sistema de controlo de versões

User Story – Modo de formular um requisito através de uma ou mais frases de acordo com a linguagem de negócio do utilizador

XSD (XML schema) – Linguagem de validação de documentos xml

7 Referências

[AD-1] Anexo A - Estudo de Estado da Arte, Critical Software S.A., CSW-CSSECPRD-2011-RPT-04497-data-discovery-estado-da-arte

[AD-2] Anexo B - Especificação de casos teste, Critical Software S.A.

[AD-3] Anexo C – Planeamento e Metodologia de Desenvolvimento, Critical Software S.A.

[RD-1] MarketScope for Enterprise Search, Gartner (2010), http://www.gartner.com/technology/media-products/reprints/microsoft/vol14/article9/article9.html, Obtido em 10 de 10 de 2011

[RD-2] *MSF for Agile Software Development v5.0, Microsoft,* http://msdn.microsoft.com/en-us/library/dd380647.aspx, Obtido em 19 de 01 de 2012

[RD-3] *Forrester Wave™: Enterprise Search, Q2 2008, Forrester*

[RD-4] Open Source Enterprise Search Software, OpenSource-IT, http://www.opensource-it.com/open_source_enterprise_search_software, obtido em 3 do 10 de 2011