Mestrado em Engenharia Informática
Dissertação/Estágio
Relatório de Estágio

csSECURE- Business Intelligence

Pedro Ricardo do Rosário Silva de Almeida
prsa@student.dei.uc.pt

Orientadores:
Prof. Dr. Bruno Cabral, DEI, FCTUC
Engº Sérgio Cruz, Watchful Software

Data: 3 de Julho de 2013

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

Resumo

Cada vez mais a informação das organizações encontra-se armazenada em formato digital, sendo grande parte desta informação confidencial. A existência de fugas de informação tem um impacto grande nas organizações, tanto a nível financeiro como a nível de competitividade e protecção da Propriedade Intelectual. Assim, torna-se essencial conseguir proteger a informação.

Além de medidas preventivas para que não existam fugas de informação, é necessário ser também proactivo e tentar detectar em tempo real eventuais tentativas, mitigando o mais possível eventuais danos.

O objectivo deste estágio é retirar informação útil das actividades dos utilizadores de um sistema de protecção de informação, através dos dados que estas actividades geram e que são passíveis de ser analisados. Assim, pretende-se criar uma Consola de monitorização destinada ao Auditor do sistema, dotando-o de capacidade para detectar comportamentos dos utilizadores que possam eventualmente representar uma ameaça para a protecção da informação da sua organização. Esta consola será uma solução de *OLAP* assente numa *Data Warehouse* a ser desenvolvida no estágio.

Palavras-Chave

Confidencialidade, Informação visual, Análise de dados, Inteligência no Negócio, Monitorização, Protecção Informação, Segurança

Agradecimentos

Gostaria de agradecer nesta fase de final do meu percurso académico a todos os amigos e colegas com quem partilhei horas de dedicação e trabalho.

Ao Professor Bruno Cabral, pela constante disponibilidade, apoio e acompanhamento em todos os períodos do meu estágio.

À equipa da Watchful Software, pela integração que me proporcionaram. Pelo companheirismo e o bom ambiente constante.

Ao Engº Sérgio Cruz, pelo acompanhamento e disponibilidade.

Ao Engº Bernardo Patrão, pelo interesse demonstrado no sucesso do meu estágio e pelo tempo disponibilizado em vários momentos importantes.

Ao Daniel, por ter sido um guia incomparável e uma presença constante. Pela paciência, apoio e sabedoria transmitida. Pelas palavras certas nas horas certas. Pela amizade.

À minha família, e em especial aos meus pais por terem estado sempre comigo.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

BD	Base de Dados
CSO	*Chief Security Officer*
DWH	*Data Warehouse*
ERM	*Enterprise Rights Management*
LM	*Log Management*
OLAP	*Online analytical processing*
SIEM	*Security Information and Event Management*
GUID	*Globally unique identifier*
MDX	*Multidimensional eXpressions*

Capítulo 1 - Introdução

O objectivo deste relatório é apresentar o trabalho desenvolvido na disciplina de "Estágio/Dissertação", do ano lectivo 2012/2013, pelo aluno Pedro Ricardo do Rosário Silva de Almeida, integrado na empresa Watchful Software. A Watchful Software é uma empresa *spin off* da Critical Software, que foi criada aquando do início do estágio, sendo que o produto sobre o qual o estágio inseriu - csSECURE - foi renomeado para RightsWATCH.

Neste estágio foi proposto desenvolver uma Consola de Monitorização destinada ao Auditor de um sistema de protecção de informação, retirando informação útil das actividades dos utilizadores do sistema. Esta Consola irá dotar o Auditor de capacidade para detectar comportamentos dos utilizadores que possam eventualmente representar uma ameaça para a protecção da informação da sua organização.

Neste capítulo é apresentado o contexto do estágio, os seus objectivos e uma descrição da documentação produzida e da estrutura do presente relatório.

1.1. Enquadramento

O estágio descrito neste relatório é uma unidade curricular do Mestrado em Engenharia Informática do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. A entidade que acolheu o estágio foi a empresa ITGrow [1], que é detida pela Critical Software e pelo Banco BPI. Assim, o estágio decorreu integrado num projecto da Critical Software, de nome csSECURE.

Pouco antes do início do estágio, o projecto csSECURE deu aso à criação de uma empresa *spin-off* da Critical Software, de nome Watchful Software [2]. Foi no seio da Watchful Software, incidindo no produto anteriormente conhecido como csSECURE e agora como RightsWATCH [3] que o estágio decorreu.

O estágio foi realizado sob a orientação do Professor Bruno Cabral, professor do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, e do Engenheiro Sérgio Cruz da Watchful Software.

1.1.1. RightsWATCH

A aplicação na qual o presente estágio se integrou, de nome RightsWATCH, é uma solução de protecção de informação centrada nos dados.

A abordagem mais comum das empresas com vista a proteger a sua informação é a protecção das suas fronteiras, ou seja, utilizar firewalls para controlar e impedir eventuais ataques externos à organização. Esta solução não é a mais eficaz, pois a maioria das fugas de informação acontecem do interior para o exterior, ou seja, com a informação a ser levada da própria empresa para o exterior. São diversos os motivos que levam a esta situação, seja a fuga propositada ou involuntária. Um eventual engano no destinatário de um email contendo informação confidencial ou mesmo um colaborador que seja dispensado e queira causar dano na empresa são duas situações em que podem ocorrer fugas de informação. Nestes casos, a protecção das fronteiras da empresa não é suficiente para impedir a fuga de informação.

O RightsWATCH surge neste contexto de protecção de informação, mais especificamente, da informação não estruturada (emails e documentos). Este sistema é uma solução *Enterprise*

Rights Management (ERM), que assenta no conceito de segurança multinível, ou seja, há uma estrutura hierárquica de classificação de informação. Deste modo, no seio de uma organização que utilize o RightsWATCH, a informação não estruturada é classificada aquando da sua criação com um dos níveis de confidencialidade definidos na política de segurança existente.

Para além da classificação da informação, também os utilizadores da aplicação (regra geral os colaboradores da empresa) são credenciados com acesso a um ou mais níveis de confidencialidade, tendo associado a este acesso alguns direitos sobre a informação (como abertura, edição, impressão, cópia, resposta, reencaminhamento, entre outros). Esta gestão dos direitos é feita de forma detalhada e independente, sendo possível assim partilhar informação dentro de uma organização sem permitir que existam acções indesejadas que possam levar à fuga de informação.

A Figura 1 ilustra um exemplo de classificação ao enviar um e-mail, sendo possível ver que o utilizador está associado a 2 empresas, sendo que na empresa *"Watchful"* tem 1 âmbito de informação associado, e neste tem 2 níveis de classificação (*"Public"* e *"Internal"*).

O e-mail a ser redigido na imagem está classificado como *Watchful-WSW-Internal*.

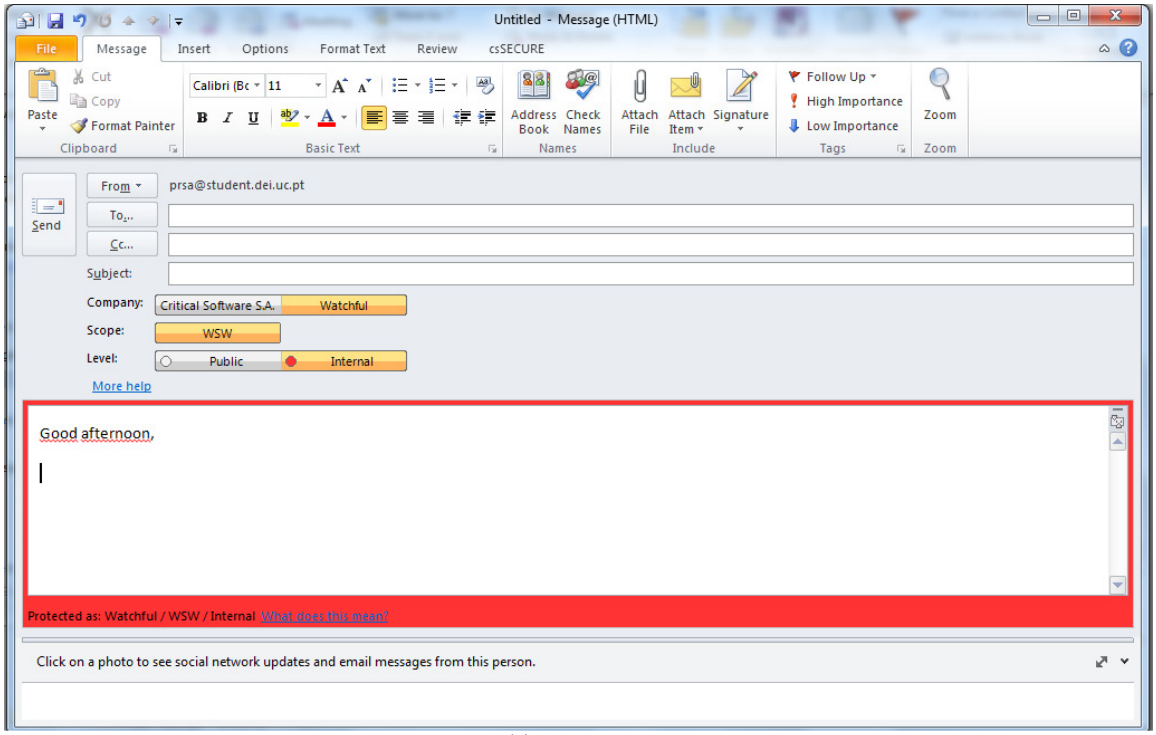


Figura 1: Interface RightsWATCH de classificação de *email* no Outlook

Toda a informação classificada torna-se inacessível às pessoas de fora da organização e também a todas as pessoas da organização que, apesar de utilizarem o RightsWATCH, não tenham credenciais para aceder a um determinado nível de confidencialidade de informação.

Existem 3 perfís de utilização do RightsWATCH: os utilizadores comuns, que criam e classificam essa informação e que acessam a informação protegida; os administradores, que definem as politicas de segurança e atribuem as credenciais aos utilizadores e o Auditor, que se certifica que não existem comportamentos da parte dos utilizadores e/ou administradores que possibilitem a fuga de informação.

Sendo o RightsWATCH direccionado para o uso empresarial, o seu desenvolvimento é focado nas aplicações de produtividade mais comuns: Microsoft Outlook, Word, Excel e Powerpoint e dispositivos móveis (iOS, BlackBerry, WindowsPhone, e Android).

O RightsWATCH é ainda dotado de um sistema de *logging* que regista todos os eventos de utilização no dia a dia, como por exemplo todos os acessos ou tentativas de acesso a informação, reenvios, alterações de classificação ou alterações da credenciação de um determinado utilizador. A informação do *logging* é colocada ao dispor do Auditor para consulta, com o objectivo de ser útil na detecção de eventuais comportamentos mal intencionados.

1.2. Objectivos

À data de início do estágio, a única ferramenta de monitorização das acções dos utilizadores do RightsWATCH passível de ser utilizada pelo Auditor era uma ferramenta de pesquisa directa na base de dados de actividade. Como se pode ver na Figura 2, a informação apresentada ao Auditor não é mais do que uma listagem de eventos, não relacionados e de difícil consulta.

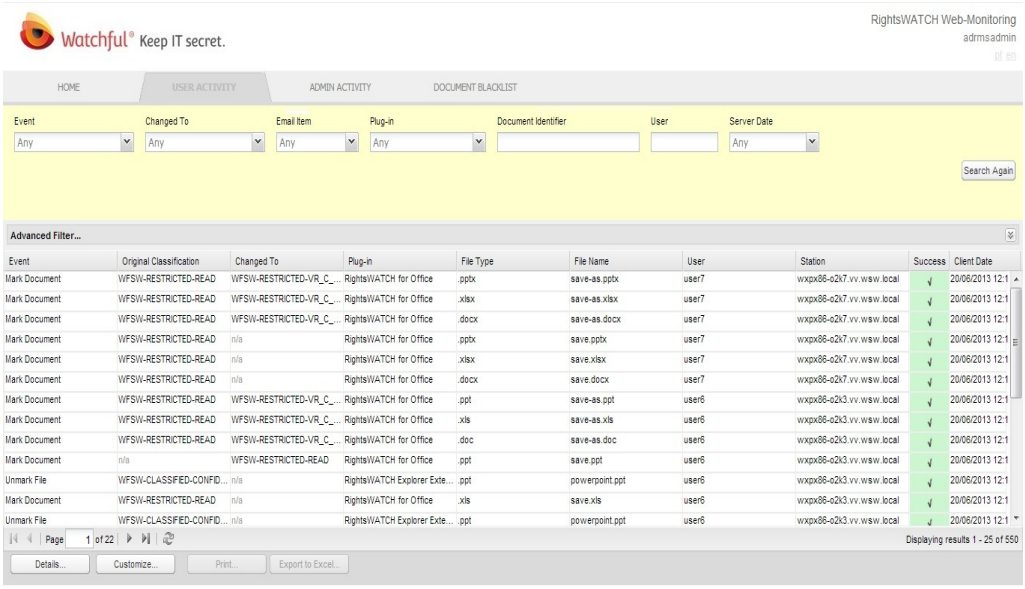


Figura 2: Listagem de eventos actual do RightsWATCH

É possível aplicar um conjunto de filtros às pesquisas, mas ainda assim torna-se impossível de detectar padrões de comportamento ou de percepcionar o estado do sistema e eventuais comportamentos não desejáveis.

Existe uma grande quantidade de informação, bastante valiosa, mas à qual não é dada verdadeira utilização, apesar do potencial enorme que esta traz consigo no que toca à monitorização do sistema, ou seja, à pro-actividade na detecção de fugas de informação.

Pretende-se com o estágio:

- Projectar uma *Data Warehouse* que sirva como base ao desenvolvimento de capacidades de monitorização para o produto *RightsWATCH;*
- Desenvolver uma Consola de Monitorização do sistema que permita ao Auditor perceber o estado geral do sistema;
- Permitir ao Auditor monitorizar o cliclo de vida dos documentos;

- Permitir ao Auditor monitorizar as actividades dos utilizadores do sistema;

A área deste estágio é a Inteligência no Negócio, que é definida pelo *Data Warehouse Institute* [4] como "Os processos, tecnologias e ferramentas necessárias para transformar dados em informação, informação em conhecimento e conhecimento em planos que guiam acções de negócio rentáveis.".

1.3. Estrutura do relatório

No Capítulo 1 - Introdução descreve-se brevemente o estágio, o seu âmbito e objectivo.

No Capítulo 2 - Planeamento do estágio apresenta-se a equipa de trabalho do RightsWATCH e o planeamento definido e seguido no estágio.

No Capítulo 3 - Análise de mercado descreve-se um estudo de mercado realizado pelo estagiário, no âmbito de soluções existentes que se assemelham à solução desenvolvida no estágio.

No Capítulo 4 - Análise de requisitos apresentam-se os requisitos levantados, descrevendo os processos seguidos pelo estagiário nesta fase e apresentando a prioritização para implementação dos requisitos levantados.

No Capítulo 5 - Arquitectura da Consola de Monitorização e escolha de tecnologias descrevem-se as tecnologias escolhidas pelo estagiário para o desenvolvimento do produto bem como a arquitectura da Consola de monitorização, com destaque para a integração da Consola de Monitorização desenvolvida com a arquitectura já existente, bem como para a arquitectura da aplicação desenvolvida (diagramas de classes).

No Capítulo 6 - Arquitectura da Data Warehouse e processo *ETL* apresenta-se o processo seguido para a implementação da Data Warehouse do RightsWATCH, bem como a descrição do processo de ETL seguido.

No Capítulo 7 - Implementação descreve-se o processo seguido na implementação, de acordo com a metodologia ágil *SCRUM* em vigor na Watchful Software.

No Capítulo 8 - Testes apresenta-se a especificação de testes e respectivos resultados relativos ao produto desenvolvido.

No Capítulo 9 - Conclusões faz-se um balanço do trabalho desenvolvido no estágio e das competências ganhas, sendo tecidos algumas considerações relativas ao trabalho futuro a desenvolver no âmbito da monitorização do RightsWATCH.

Capítulo 2 - Planeamento do estágio

Esta secção descreve o método de trabalho da Watchful Software, que foi seguido pelo estagiário.

Apresenta-se também o planeamento do estágio e as tarefas desenvolvidas nas diferentes fases do mesmo.

2.1 Gestão da Equipa

A metodologia que actualmente é utilizada pela Watchful Software para o desenvolvimento do RightsWATCH é o *SCRUM*, uma metodologia ágil que funciona por *sprints* (sendo uma *sprint* um ciclo de desenvolvimento), que no caso da Watchful Software são regra geral de duas semanas.

Para o trabalho do primeiro semestre do presente estágio, foi decidido que o estagiário iria seguir a metodologia sequencial *Waterfall*, por ser mais adequada ao tipo de trabalho a desenvolver, sendo que no 2º semestre o estagiário foi integrado no desenvolvimento da equipa em *SCRUM* para a fase de implementação. No entanto, a familiarização com esta metodologia de desenvolvimento começou no primeiro semestre, com a presença do estagiário em momentos importantes da equipa, como as *Daily Scrum Meetings, Sprint review* e *Sprint Retrospective.*

Regista-se também o facto de todos os anexos produzidos no âmbito do estágio estarem escritos em inglês, sendo esta uma prática na Watchful Software, derivada do facto de existirem colaboradores de diversas nacionalidades na empresa. Assim, alguns termos presentes no relatório estarão também em inglês, pois são termos comuns utilizados pela equipa.

A equipa de SCRUM do RightsWATCH é composta pelos seguintes elementos:

- *Product Owner:* Pessoa que representa a parte interessada. É ele quem define as prioridades de acordo com as oportunidades de negócio que surgem.
- *Scrum Master:* Elemento que assegura a continuidade das boas práticas de SCRUM. Lidera a equipa sendo um facilitador que desbloqueia eventuais impedimentos garantindo que a equipa se encontra exclusivamente focada no desenvolvimento.
- *Team:* Seis pessoas, que analisam, planeiam, implementam e testam as funcionalidades do produto.

Já na fase de implementação, o estagiário foi plenamente integrado no desenvolvimento com a equipa, cumprindo todas as sessões previstas pela metodologia *SCRUM:*

- *Daily Scrum Meetings*: Realizadas diariamente no início da manhã e com a duração de 15 minutos, têm como objectivo cada elemento da equipa responder a três questões:
  - *O que esteve a fazer no dia anterior?*
  - *O que irá fazer no dia actual?*
  - *Existe qualquer tipo de impedimento à normal realização do trabalho?*
- O estagiário participou nas reuniões, deixando a equipa a par do seu trabalho e ao mesmo tempo ficando a par do trabalho realizado pela equipa de desenvolvimento.
- *Sprint Planning*: Reunião que marca o início da *sprint*. As *User Stories* que fazem parte da Sprint são partidas em pequenas tarefas, cada uma com duração máxima prevista

de 2 dias (16 horas). Esta estimativa é calculada com base numa média pesada de 3 valores, descritos de seguida, prática que tem o nome de *Three Point Estimation*:
    - *Worst case:* duração prevista em horas que demorará a completar a tarefa no pior caso possível;
    - *Best case*: duração prevista em horas que demorará a completar a tarefa no melhor caso possível;
    - *Most likely*: duração prevista em horas que demorará a completar a tarefa no caso mais provável;
- *Grooming*: Sessão onde é feita a estimativa em *story points* (definição em 7.1 Resumo das *Sprints*) para as *User Stories* definidas e onde se clarificam os objectivos de cada uma para que a equipa esteja alinhada relativamente às *User Stories*.
- *Sprint Review*: Reunião de final de *sprint* onde são apresentados os resultados conseguidos ao *Product Owner*, que por sua vez avalia o desempenho da equipa.
- *Sprint Retrospective*: Reunião que decorre após a *Sprint Review* onde é analisado o que correu melhor na *sprint* que termina (*well-done)* e o que necessita de ser melhorado para as *sprints* seguintes (*to improve*), bem como factores que afectaram a *sprint* mas que não estão relacionados directamente com o funcionamento da metodologia ou da equipa *(rants)*.

Durante o estágio ocorreram variadas reuniões entre o estagiário e o orientador interno sempre que estas se revelavam importantes, como aconteceu para a revisão do material produzido e para validação do trabalho efectuado em cada uma das metas definidas. Também ocorreram reuniões entre o estagiário, o orientador interno e orientador externo, Professor Bruno Cabral, a um ritmo mensal, para acompanhamento do trabalho realizado.

2.2. Planeamento do estágio

O estágio teve início no dia 24 de Setembro de 2012 e terminou no dia 8 de Julho de 2013.

As tarefas foram planeadas inicialmente, sendo definida uma *milestone* interna para todas (designada por "Entrega" no planeamento abaixo), regra geral marcada para uma semana antes do prazo destinado para o término da tarefa, que visava rever o trabalho realizado junto do orientador, a fim de fazer eventuais alterações atempadamente.

A carga horária alocada para o estágio no 1º semestre foi de 60% do tempo total, ou seja, 24 horas semanais, sendo que durante o período correspondente à época de exames esta carga foi aumentada para 80%, ou seja, 32 horas semanais. No 2º semestre a carga horária foi de 100%, ou seja, 40 horas semanais

Na Figura 3 pode ser consultado o diagrama de *Gantt* com a sequência das tarefas realizadas durante o estágio.

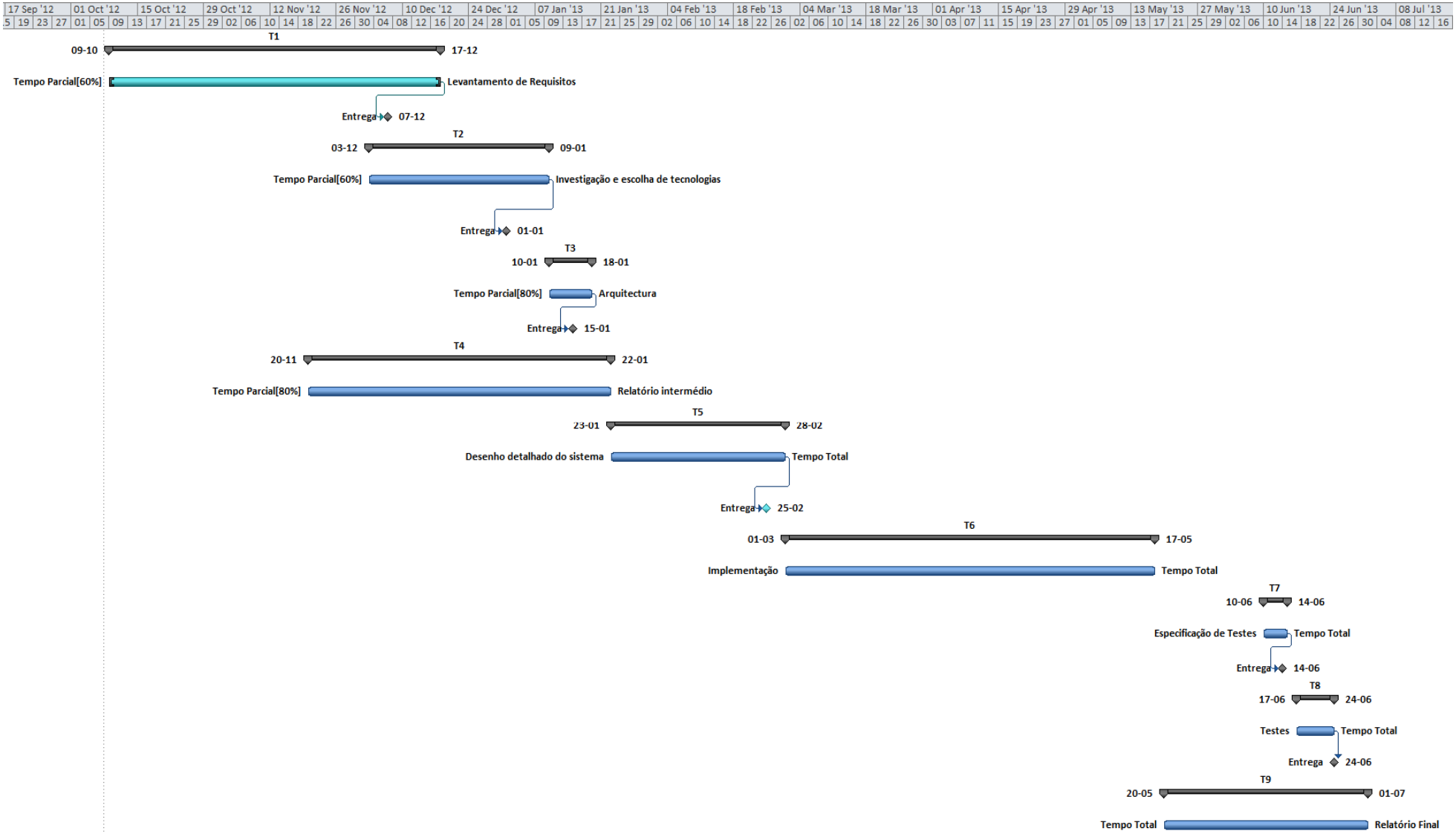


Figura 3: Diagrama de *Gantt* de planeamento do estágio

Na tabela abaixo encontra-se a descrição das tarefas apresentadas no diagrama de *Gantt*, bem como a duração das mesmas em dias úteis de trabalho, sendo 1 dia equivalente a 8 horas.

Tarefa	Descrição	Duração
T1 – Levantamento de requisitos	Elaboração de um conjunto de tarefas directamente relacionadas com os requisitos do sistema: • Elaboração de questionário de funcionalidades de uma solução de monitorização (Anexo [1] Questionário para levantamento de requisitos, WSW-2013-RPT-00004-information-protection-system-survey); , visando a sua distribuição por um conjunto de pessoas que desempenham papéis semelhantes ao de Auditor nas suas empresas. • Estudo de mercado, descrito no Capítulo 3 - Análise de mercado) e formalizado no Anexo [2] Estudo de mercado, WSW-2013-RPT-00003-rightswatch-monitoring-lm-siem-market-analysis; • Levantamento de requisitos, descrito no Capítulo 4 - Análise de requisitos) e formalizado no documento de especificação de requisitos (Anexo [3] Levantamento de requisitos da Consola de Monitorização do RightsWATCH, WSW-2012-SRS-00008-rightswatch-monitoring-srs);	30
T2 – Investigação e escolha de tecnologias	Escolha das tecnologias e ferramentas a utilizar e pesquisa sobre cada uma delas de forma a perceber as potencialidades e adequação ao trabalho a desenvolver. Este trabalho é descrito no Capítulo 5 - Arquitectura da Consola de Monitorização e escolha de tecnologias). Foi elaborada uma pesquisa mais detalhada sobre ferramentas *Extract, Transform and Load* (*ETL*), formalizada num documento interno (Anexo [4] Estudo de ferramentas de *ETL*, WSW-2013-RPT-00005-dw-technology-research).	16.8
T3 – Arquitectura	Desenho da arquitectura de alto nível do sistema a desenvolver, descrita no Capítulo 5 - Arquitectura da Consola de Monitorização e escolha de tecnologias).	5.6
T4 – Relatório intermédio	Redacção do relatório intermédio de estágio e preparação da apresentação intermédia do estágio.	36.8
T5 – Desenho detalhado do sistema	Desenho do sistema a implementar, tendo por base a arquitectura definida na tarefa T3 – Arquitectura, no primeiro semestre, que foi detalhado num documento interno, disponível no anexo [5] Desenho detalhado do sistema.	27
T6 – Implementação	Implementação da DWH bem como dos requisitos propostos, em forma de User Stories. Inclui o desenvolvimento de todo o processo de *ETL* relativo à DWH proposta no desenho de arquitectura detalhada do	56

	sistema.	
T7 – Especificação de testes	Especificação de testes para validar o sistema desenvolvido na tarefa anterior.	5
T8 – Testes	Execução dos testes desenhados na tarefa anterior e corrigir eventuais *bugs* encontrados.	6
T9 – Relatório Final	Redacção do relatório final do estágio e preparação da apresentação final de estágio.	54

Tabela 1: Descrição das tarefas realizadas

Capítulo 3 - Análise de mercado

De forma a conhecer as potencialidades de um sistema de monitorização, foi desenvolvido um estudo tecnológico de mercado em duas vertentes:

- Componente de monitorização de Sistemas de Protecção de Informação de empresas concorrentes no mercado da Watchful Software;
- Soluções *Security Information and Event Management (SIEM) / Log Management (LM);*

Neste estudo descrevem-se as soluções encontradas bem como os critérios de comparação utilizados.

Este estudo foi importante no contexto do estágio por vários motivos:

- Foi o primeiro contacto do estagiário com sistemas semelhantes ao que se pretende desenvolver;
- Permitiu uma ambientação ao contexto de monitorização e gestão de eventos;
- Recolheu-se informação sobre o que é feito em termos de monitorização no segmento de mercado do RightsWATCH;
- Percepção das funcionalidades mais comuns em sistemas de monitorização;

Todos estes factores se mostraram relevantes e tiveram implicação directa no levantamento de requisitos, fazendo assim com que a Consola de Monitorização do RightsWATCH cubra as funcionalidades mais importantes e interessantes de entre as estudadas, e ainda outras funcionalidades que não foram identificadas nestas soluções.

Este estudo foi formalizado num documento interno (Anexo [2] Estudo de mercado, WSW-2013-RPT-00003-rightswatch-monitoring-lm-siem-market-analysis) e é resumido nas secções seguintes.

3.1. Soluções de monitorização dos concorrentes de mercado do RightsWATCH

Esta pesquisa foi realizada com base em empresas que se encontravam identificadas internamente como sendo concorrentes do produto RightsWATCH e teve como objectivo perceber as potencialidades das ferramentas de monitorização destes produtos, ficando assim a conhecer o que os concorrentes oferecem em termos de monitorização dos seus sistemas.

Estavam identificadas internamente, pela Watchful Software, seis empresas do mesmo segmento de mercado. Destas empresas, três não ofereciam capacidades de monitorização na sua solução, incidindo assim a pesquisa nas restantes três empresas/soluções, a saber:

- ***Titus*:** empresa que desenvolve uma solução de segurança e protecção de informação com o mesmo nome. Para a monitorização do sistema utilizam *software* externo de nome *Splunk App*, desenvolvido pela *Splunk Inc.*
- ***GigaTrust*:** empresa que desenvolve *software* de protecção de conteúdo digital, oferecendo algumas funcionalidades de monitorização.
- ***Fasoo*:** empresa que desenvolve uma solução de protecção de informação de nome *Fasoo Enterprise DRM*. Para acrescentar capacidades de monitorização a esta solução a *Fasoo* desenvolveu o *Fasoo Usage Tracer*.

Houve alguma dificuldade na recolha de informação sobre os produtos analisados, pois a informação disponibilizada pelas empresas é muito pouco detalhada, como é normal dada a competição existente. Estas empresas foram contactadas pelo estagiário afim de recolher informação mais detalhada, tendo sido tentado também a marcação de demonstração destes produtos. Estas tentativas revelaram-se infrutíferas, dado que se trata de empresas sensíveis à fuga de informação e roubo de propriedade intelectual. Com base nas funcionalidades encontradas, foram definidos critérios que permitem comparar as soluções, fornecendo uma visão de alto nível das capacidades de monitorização dos produtos analisados.

3.1.1. Matriz comparativa

As funcionalidades que foram definidas como critério para a elaboração da matriz de comparação foram as seguintes:

- ***Search***: Possibilidade de executar pesquisas nas actividades/eventos.
- ***Dashboard***: Existência de um painel com informação visual acerca do sistema.
- ***Customizable Dashboard***: Possibilidade de personalização do *Dashboard*.
- ***Alerts***: Existência de alertas que notificam o utilizador no caso de ocorrência.
- ***Reports***: Possibilidade de criação de relatórios, que visam dar uma visão do sistema num determinado período de tempo, baseado em métricas definidas.
- ***Level of Risk***: Atribuição de um grau de risco a cada alerta. O objectivo deste grau de risco é quantificar o perigo que cada alerta definido representa e prioritizar os alertas existentes no sistema.
- ***Suspicious List***: Existência de uma lista de ficheiros ou utilizadores suspeitos. A informação relativa à monitorização destes ficheiros/utilizadores é mostrada ao utilizador numa secção própria.
- ***Information Tracking***: Possibilidade de percorrer o ciclo de vida de um determinado documento ou outro tipo de ficheiro encriptado (termos abreviados para o genérico *informação*), de um modo visual, dando a possibilidade ao utilizador de ver todos os eventos associados a essa informação.

A Figura 4 mostra os diferentes critérios aplicados às capacidades de monitorização dos produtos de cada uma das empresas identificadas, bem como as funcionalidades que se desejam ver implementadas no RightsWATCH.

	Companies			
	RightsWATCH	Titus	GigaTrust	Fasoo Usage Tracer
Search	✓	✓	✓	✓
Dashboard	✓	✓	✓	✓
Customizable Dashboard	✓	✓		
Alerts	✓	✓	✓	✓
Reports	✓	✓	✓	✓
Level of Risk	✓	✓		✓
Suspicious List				✓
Information Tracking	✓			
		Uses Splunk		

Figura 4: Matriz comparativa de soluções de monitorização dos concorrentes de mercado

3.2. Soluções SIEM / LM

Neste âmbito foi realizada uma pesquisa sobre de soluções SIEM/LM. Inicialmente a pesquisa incidiu sobre soluções de LM, que são destinadas à recolha de eventos de vários tipos e em alguns casos à sua análise. Ao pesquisar estas soluções apareceu um novo conceito, e com este um novo tipo de soluções: SIEM. O objectivo destas aplicações é a criação de relatórios e alertas em tempo real baseando-se em ficheiros de log que contêm a informação.

No entanto, existem soluções identificadas como LM com capacidades de análise de dados, criação de relatórios e alertas (que na teoria são funcionalidades SIEM) bem como soluções identificadas como SIEM com capacidades de recolha e armazenamento de eventos (que na teoria são funcionalidades LM).

Assim, foram consideradas para este estudo as duas categorias de produtos sem distinção por tipo, pois as funcionalidades destas aplicações são bastante semelhantes e na maioria das vezes cruzam-se.

Nesta secção são analisadas e comparadas 8 soluções comerciais de SIEM/LM que estão disponíveis no mercado e 4 soluções *open source.*

As soluções comerciais identificadas foram as seguintes:

- ***LogRythm SIEM 2.0:*** Desenvolvido pela empresa *LogRythm*, é uma das soluções mais completas de entre as estudadas. É dotada de um sistema de nome *Advanced Intelligence Engine*, que correlaciona informação recolhida por aplicações de diferentes tipos.
- ***Sentinel LM:*** Desenvolvido pela empresa *NetIQ*, sendo capaz de recolher armazenar e analisar eventos, faz destaque a um sistema de nome *Anomaly Detection*, que possibilita a detecção de eventos que fujam a padrões estabelecidos pelo utilizador, com base em padrões existentes no histórico de eventos armazenados.
- ***Splunk:*** Desenvolvido pela *Splunk*, é um produto utilizado por uma empresa concorrente do RightsWATCH – Titus. É, a par do *LogRythm SIEM 2.0* a solução estudada que abrange o maior número de funcionalidades.
- ***HP ArcSight Logger:*** Desenvolvido pela *Hewllet-Packard*, tendo muito pouca informação disponível, mas que é mencionada como tendo capacidade de recolha e análise de informação, com capacidade de exportar relatórios e criação de alertas, entre outros.
- ***Log Logic:*** Desenvolvido pela *TIBCO*, é anunciado como o produto que consegue armazenar o maior número de eventos por segundo: cerca de 250 mil.
- ***QRadar SIEM:*** Desenvolvido pela *Q1 Labs*.
- ***Nagios XI:*** Desenvolvido pela *Nagios Enterprise*, é uma ferramenta que prima pela personalização dos seus *dashboards*, sendo possível criar vários écrans com *dashboards* diferentes.
- ***RSA Envision:*** Desenvolvido pela EMC, permite que se definam padrões de utilização normais de utilização, sendo o utilizador notificado se ocorrer algum padrão que não se enquadre nos definidos.

As soluções *open source* identificadas foram as seguintes:

- ***GrayLog2:*** Ferramenta que utiliza um servidor *Elastic Search* para armazenar e pesquisar informação. Possibilita a criação de *dashboards* simples.
- ***Logstash:*** Ferramenta que também utiliza *Elastic Search*, oferecendo possibilidade de exportar os eventos para um servidor *GrayLog2* ou *mongoDB*. É possível pesquisar os eventos utilizando expressões regulares, através da biblioteca *Grok*.
- ***NXLog:*** Ferramenta que suporta recolha de logs através de várias fontes como *Syslog*, *Windows Event Log* ou *GrayLog2*. O objectivo desta solução é a recolha de logs, disponibilizando-os depois para que possam ser armazenados e analisados por outra solução.
- ***Cyberoam iView:*** Ferramenta *open source* mais completa entre as analisadas, possibilita a recolha e análise de logs, oferecendo funcionalidades de *dashboard* e relatórios.

3.2.1. Matriz comparativa

Os critérios de comparação foram agrupados por conjuntos de funcionalidades:

- ***Logging:*** Conjunto de funcionalidades relacionadas com *logs* de eventos;
- ***Searching:*** Funcionalidade relacionada com a pesquisa de eventos por filtros;
- ***Monitoring:*** Funcionalidades relacionadas com a monitorização de eventos relacionados com os utilizadores, ficheiros ou máquinas.
- ***Dashboards:*** Funcionalidades relacionadas com o *dashboard* da ferramenta.
- ***Alerts:*** Funcionalidades relacionadas com os alertas.
- ***Reporting:*** Funcionalidades relacionadas com os relatórios.
- ***Information Tracking:*** Funcionalidades relacionadas com a monitorização e visualização do ciclo de vida da informação do sistema.
- ***Others:*** Conjunto de funcionalidades que não se encaixavam nos grupos definidos, como por exemplo a existência de acções automáticas executadas pela ferramenta e a correlação de dados de diversas fontes.

A Figura 5 e Figura 6 mostram os diferentes critérios aplicados às capacidades de monitorização dos produtos identificados, bem como as funcionalidades desejadas para o RightsWATCH.

		Comercial Solutions								
		RightsWATCH	LogRythm SIEM	Sentinel LM	Splunk	ArcSight Logger	Log Logic	Qradar SIEM	Nagios Xi	RSA Envision
Logging	Log Collecting	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Log Analysis	✓	✓	✓	✓	✓	✓	✓	✓	✓
Searching	Filtered Searching	✓	✓	✓	✓	✓	✓	✓		✓
Monitoring	File Monitoring	✓	✓		✓				✓	
	Users Monitoring	✓	✓		✓					
	Host Monitoring	✓	✓	✓	✓	✓	✓	✓	✓	✓
Others	Event/Data Correlation	✓	✓	✓	✓			✓		✓
	Pattern Recognition		✓	✓						
	Automatic Actions	✓	✓		✓					✓
Dashboards	Dashboard	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Customizable Dashboard	✓	✓		✓		✓	✓	✓	
	Drill-Down & Drill-Back	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Save Dashboard Layouts	✓	✓		✓					
Alerts	Real-Time Alerts	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Alerts Prioritization/Level of risk	✓	✓		✓					✓
Reporting	Reporting	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Turn search into report	✓		✓	✓		✓			
	Export Compliance Reports		✓	✓			✓	✓	✓	✓
	Scheduled Reports	✓	✓	✓				✓		✓
Information Tracking	Information events	✓	✓		✓				✓	
	Information visual tracking	✓								
					Used by Titus					

Figura 5: Matriz de comparação de soluções comerciais SIEM/LM

		Comercial Solution	OpenSource			
		RightsWATCH Monitoring	GrayLog2	LogStash	NXLog	Cyberoam iView
Logging	Log Collecting	✓		✓	✓	✓
	Log Analysis	✓	✓	✓	✓	✓
Searching	Filtered Searching	✓	✓	✓		✓
Monitoring	File Monitoring	✓				
	Users Monitoring	✓				
	Host Monitoring	✓				✓
Others	Event/Data Correlation	✓			✓	
	Pattern Recognition				✓	
	Automatic Actions	✓				
Dashboards	Dashboard	✓	✓			✓
	Customizable Dashboard	✓				
	Drill-Down & Drill-Back	✓				
	Save Dashboard Layouts	✓				
Alerts	Real-Time Alerts	✓				
	Alerts Prioritization/Level of risk	✓				
Reporting	Reporting	✓				✓
	Turn search into report	✓				
	Export Compliance Reports					✓
	Scheduled Reports	✓				
Information Tracking	Information events	✓				
	Information visual tracking	✓				

Figura 6: Matriz de comparação de soluções *open source* SIEM/LM

3.3. Conclusões do estudo

A decisão da Watchful Software de desenvolver a sua própria solução é justificada pelos seguintes pontos:

- Possibilidade de desenho de *interface* de acordo com o já existente nas consolas de monitorização;
- Inexistência de possibilidade de monitorização do ciclo de vida da informação nas aplicações estudadas;
- Facilidade em personalizar o tipo de métricas e informação a retirar da utilização do RightsWATCH;

O resultado deste estudo teve implicações directas nos requisitos da aplicação a desenvolver no presente estágio.

Uma funcionalidade que não existe nas ferramentas identificadas e que é considerada fulcral pela Watchful Software é a monitorização do ciclo de vida de informação encriptada com o RightsWATCH. Assim, foram identificados pelo estagiário os requisitos necessários à implementação desta ferramenta durante o estágio.

Foram também retirados deste estudo ideias válidas, que se tornaram requisitos da aplicação a desenvolver, dos quais se destacam:

- Existência de vários *layouts* de *dashboard*;
- Possibilidade de converter uma pesquisa num relatório;
- Agendamento periódico de relatórios;

Capítulo 4 - Análise de requisitos

Este capítulo apresenta os requisitos do projecto a desenvolver no estágio.

Foram identificados requisitos de dois tipos:

- **Funcionais:** Requisitos que descrevem o que o sistema deve fazer;
- **Não Funcionais**: Requisitos que caracterizam e descrevem qualidades do sistema, e não as suas funcionalidades.

Antes de ser realizado o levantamento de requisitos, foram percorridas algumas fases com vista a melhor perceber as funcionalidades mais importantes de um sistema de monitorização:

- **Análise de mercado:** Foi realizado um estudo de mercado, que é descrito no Capítulo 3 - Análise de mercado) do presente relatório.
- **Reunião com *Chief Security Officer* (CSO) Critical Software:** Foi agendada uma reunião com o *CSO* da Critical Software, ou seja, o Auditor do sistema RightsWATCH da empresa. Nesta reunião, que decorreu no dia 22 de Outubro de 2012, foram abordados alguns tópicos de importância para o estágio, tendo sido alvo de maior enfoque as métricas que podem existir num sistema de monitorização.
- **Questionário:** Com base nos dois passos anteriores, foi criado um questionário [5] que foi distribuído por uma rede de contactos da Watchful Software, sendo enviado para um total de aproximadamente 75 destinatários, *CSOs* de empresas ou detentores de um cargo equivalente. O questionário incidiu sobre as secções de *Dashboard* e Alertas e visou essencialmente perceber quais as funcionalidades mais interessantes para os questionados. Foi dado um prazo de resposta de 15 dias após o envio do questionário, a fim de coincidir com o início do levantamento de requisitos, sendo que o número de respostas foi muito reduzido. Dado ser um número pequeno para validação de resultados, decidiu ignorar-se essas respostas e os resultados que pudessem advir dos questionários.

4.1. Estrutura Modular

Para facilitar a identificação de requisitos, foi decidido agrupar os requisitos em secções, sendo cada secção relacionada com um tipo específico de requisitos. Este conceito vai de encontro ao praticado na Consola de Administração do RightsWATCH, que também segue um modelo modular. Esta encontrava-se já implementada à data do estágio e é utilizada para definir as configurações do RightsWATCH.

Os conjuntos de funcionalidades identificados para a Consola de Monitorização deram aso à criação das secções de requisitos, que por sua vez serão as secções da Consola de Monitorização. Assim, as secções identificadas são:

- ***Home***: Secção principal da Consola;
- ***Dashboard:*** *Dashboard* com informação da utilização do RightsWATCH;
- ***Alerts:*** Alertas do sistema;
- ***Reports:*** Relatórios de informação da utilização do RightsWATCH;
- ***Information Tracking:*** Monitorização do ciclo de vida de informação encriptada com o RightsWATCH;
- ***User Activity:*** Monitorização dos eventos dos utilizadores do sistema;

- ***Admin Activity:*** Monitorização dos eventos dos administradores do sistema;
- ***Users:*** Informação acerca dos utilizadores existentes no sistema e os seus perfis de utilização;
- ***Roles:*** Permissões dos perfis de utilização existentes no sistema e que podem ser atribuídos aos utilizadores;
- ***Classifications:*** Classificações de informação existentes no sistema;

4.2. Casos de Uso

Para ajudar à identificação de requisitos foram criados diagramas de Casos de Uso para todas as secções do sistema a desenvolver, estando os mesmos disponíveis para consulta no anexo relativo ao levantamento de requisitos ([3] Levantamento de requisitos da Consola de monitorização do RightsWATCH, WSW-2012-SRS-00008-rightswatch-monitoring-srs).

Na presente subsecção apresentam-se, a título exemplificativo, os Casos de Uso relativos à secção de *Dashboard*.

O actor que irá realizar acções nesta secção é o seguinte:

- **Auditor**: Utiliza directamente a Consola de Monitorização;

4.2.1. Secção *Dashboard*

A Figura 7 diz respeito aos casos de uso da secção *dashboard*. As principais interacções com o sistema nesta secção são a criação de métricas e vistas de *dashboard*, personalização de vistas de *dashboard* e a possibilidade de o Auditor criar as suas próprias métricas, como por exemplo os utilizadores com mais eventos realizados no sistema ou a quantidade de tentativas negadas de leitura de ficheiros encriptados. É ainda possível ao Auditor verificar o nível de risco atribuído à organização, baseado no número de alertas existentes e ocorrências de cada um deles.

O Auditor pode exportar informação existente no *dashboard*, sendo que tal é possível de ser feito para um ficheiro de imagem ou de *MS Excel* podendo também imprimir directamente esta informação.

É possível ao Auditor seleccionar informação presente no *dashboard*, como por exemplo:

- Um gráfico, para personalizar a informação relacionada (janela de tempo, por exemplo);
- Um evento de um gráfico, para visualizar informação relacionada somente com esse evento;
- O nível de risco da empresa, para personalizar a informação relacionada (janela de tempo, por exemplo);

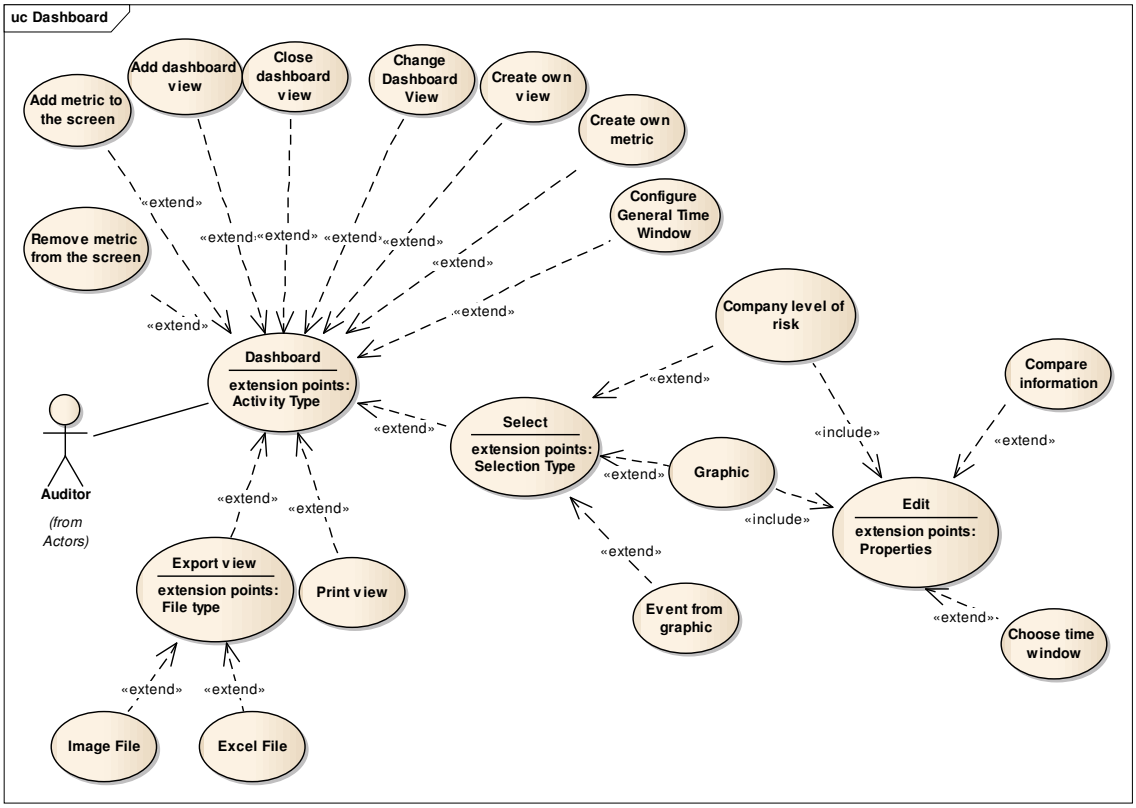


Figura 7: Casos de uso secção *dashboard*

4.3. Protótipos

Nesta secção é apresentado o protótipo respeitante à secção *Dashboard* da Consola de Monitorização. Para ajudar à identificação de requisitos foram criados protótipos dos ecrãs da aplicação, que podem ser consultados no anexo [3] Levantamento de requisitos da Consola de monitorização do RightsWATCH, WSW-2012-SRS-00008-rightswatch-monitoring-srs .

De seguida apresenta-se um exemplo de uma das secções mais relevantes, a secção de *Dashboard*.

4.3.1. Secção *Dashboard*

Na Figura 8 está desenhado o protótipo da secção *Dashboard*, onde o Auditor pode ter uma visão gráfica e geral sobre o sistema bem como executar várias acções.

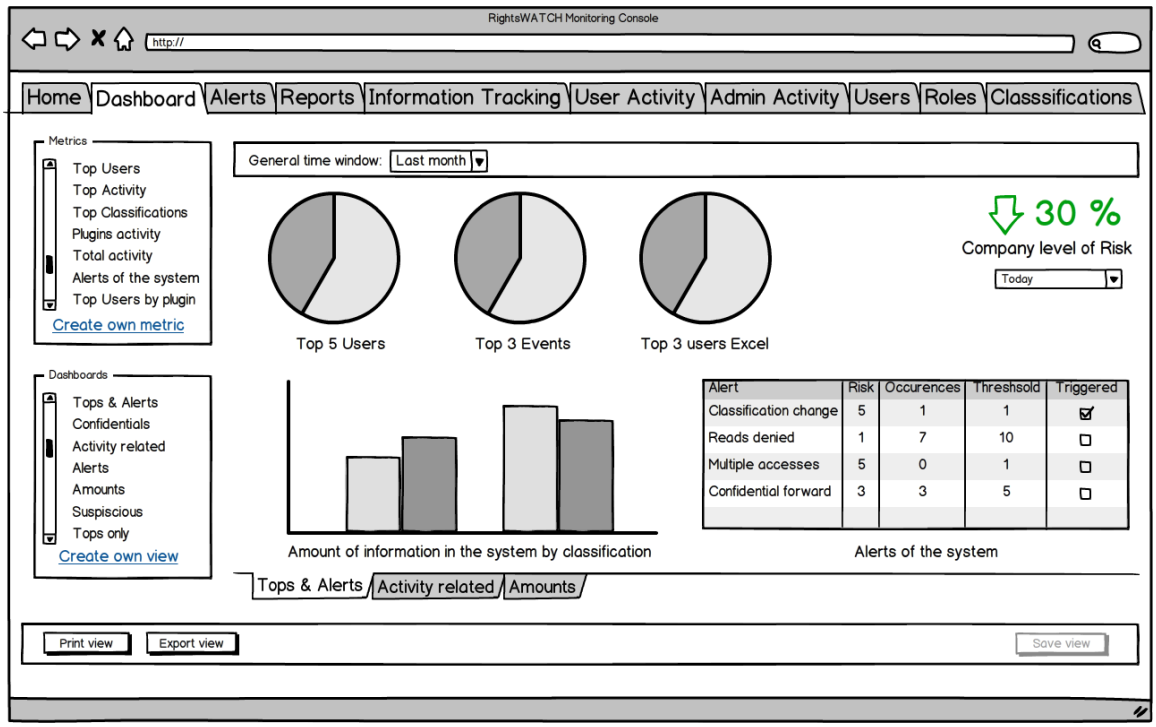


Figura 8: Protótipo da secção *Dashboard*

O *Dashboard* é totalmente personalizável, podendo o Auditor escolher a quantidade de métricas que deseja ver no ecrã, a dimensão de cada um dos gráficos e ainda o número de vistas de *Dashboard*, podendo assim ter o *Dashboard* configurado da maneira que lhe pareça mais adequada.

Na figura acima exemplifica-se a secção com recurso a métricas como os 5 utilizadores mais activos (com mais eventos realizados) no sistema, os 3 tipos de eventos mais realizados no sistema ou os 3 utilizadores que realizam mais eventos no *Microsoft Excel*.

É possível a qualquer altura alterar uma vista, podendo posteriormente a mesma ser gravada.

A partir deste ecrã é possível chegar a qualquer outra secção da Consola, através de somente um clique, utilizando os separadores disponíveis no cimo do ecrã.

4.4. Requisitos Funcionais

Nesta secção são apresentados, a título exemplificativo, os requisitos funcionais identificados no levantamento de requisitos, respeitantes à secção de *Dashboard* da Consola de Monitorização.

A descrição completa dos requisitos pode ser consultada no Anexo [3] Levantamento de requisitos da Consola de Monitorização do RightsWATCH, WSW-2012-SRS-00008-rightswatch-monitoring-srs .

A Figura 9 mostra uma visão geral dos requisitos funcionais do sistema, em que cada grupo de requisitos diz respeito a uma secção da Consola de Monitorização, de acordo com o descrito na secção 4.1 Estrutura Modular do presente relatório.

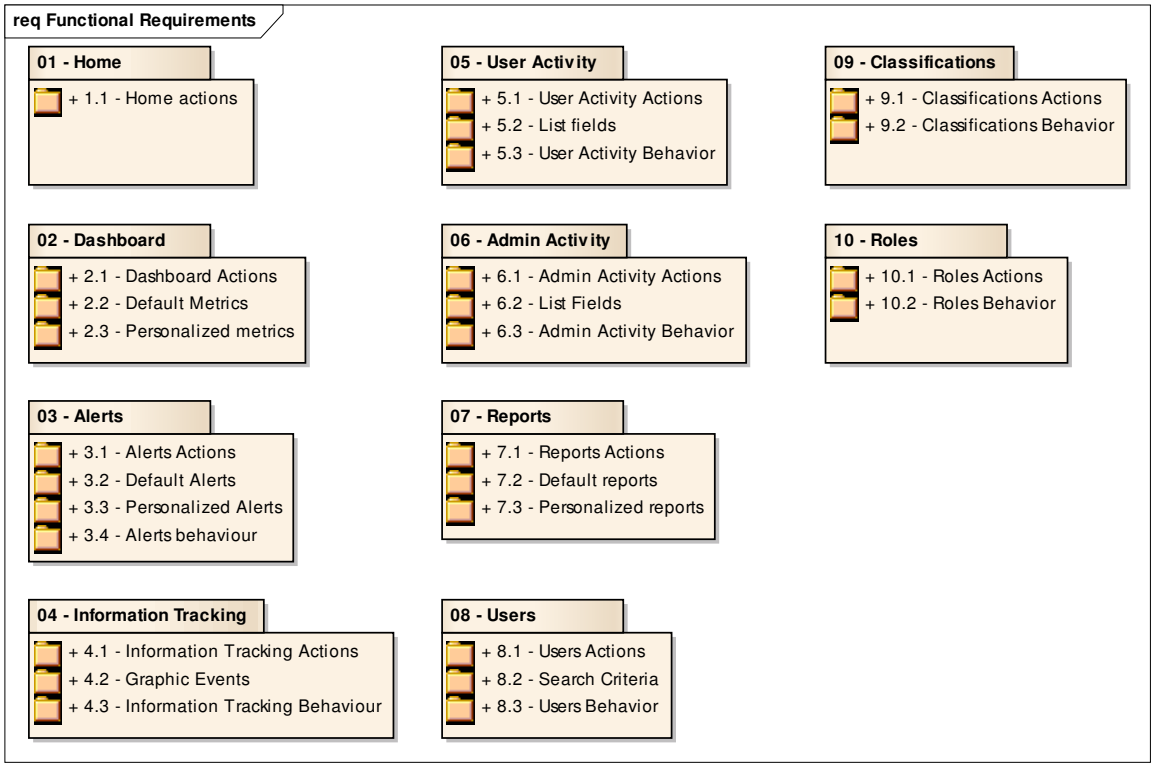


Figura 9: Visão geral dos Requisitos Funcionais

4.4.1. Secção *Dashboard*

Na Figura 10 encontram-se os requisitos da secção *Dashboard* da Consola, cujo objectivo é dar uma visão geral do sistema ao Auditor de um modo visual.

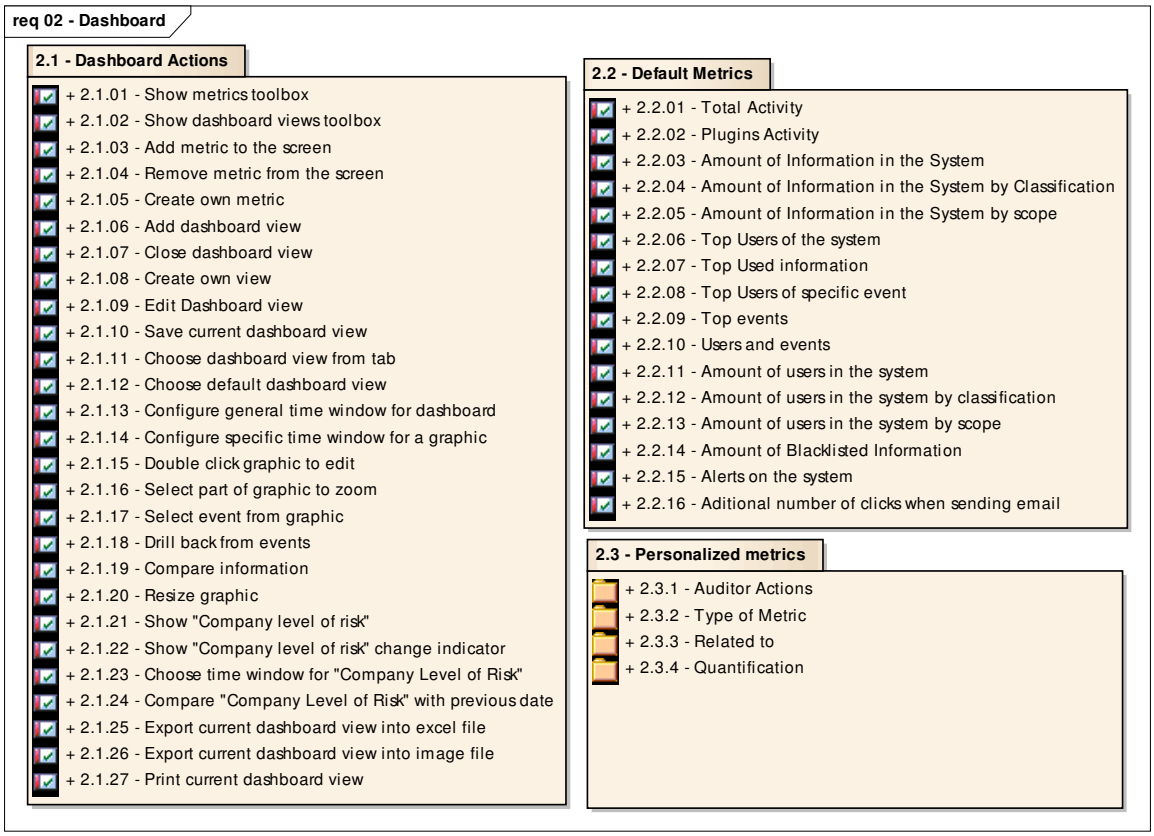


Figura 10: Requisitos Funcionais secção *Dashboard*

Estes requisitos encontram-se divididos em 3 grupo, que são descritos de seguida.

2.1 - Dashboard Actions: Grupo de requisitos que dizem respeito às acções que o Auditor pode realizar nesta secção da Consola. Destacam-se destes requisitos os seguintes:

- ***2.1.03 – Add Metrics to the screen:*** O sistema deve permitir que o Auditor adicione informação (métricas) ao ecrã actual, a partir de uma caixa de ferramentas com as métricas existentes no sistema. As métricas que se encontram nesta caixa de ferramentas são as métricas por omissão do sistema (secção 2.2 – *Default Metrics*) e as métricas criadas pelo Auditor (secção 2.3 – *Personalized Metrics*);
- ***2.1.05 – Create Own Metric:*** O sistema deve permitir que o Auditor crie uma métrica personalizada que mais tarde possa ser adicionada ao *dashboard*. A criação de métricas personalizadas pelo Auditor encontra-se especificada na secção *2.3 – Personalized Metrics*.
- ***2.1.17 – Select Event from Graphic:*** O sistema deve permitir que o Auditor seleccione um evento especifico do sistema para detalhar a sua pesquisa. (Esta acção é conhecida em Business Intelligence como *drill down)*. Esta acção pode ser revertida, sendo tal comportamento especificado no requisito *2.1.18 – Drill back from events*.
- ***2.1.21 – Show Company Level of Risk:*** O sistema deve mostrar o valor de "Nível de risco da empresa". O sistema deve calcular este valor com base no número de alarmes definidos pelo Auditor que tenham pelo menos uma ocorrência de eventos, tendo por base a fórmula seguinte:

- $Individual\ alert\ risk = \frac{level\ of\ alert}{\max level\ of\ alert}\ x \frac{number\ of\ occurrences}{threshold}$
- ***2.1.25 – Export current dashboard view into excel file:*** O sistema deve permitir que o Auditor exporte a vista actual do *dashboard* para um ficheiro de MS Excel.

2.2 - Default Metrics: Grupo de requisitos que dizem respeito às métricas que o sistema contém por omissão e que podem ser utilizadas pelo Auditor. Destacam-se destes requisitos os seguintes:

- ***2.2.03 - Amount of information in the system:*** O sistema deve permitir que o Auditor escolha como métrica o número de ficheiros/informação classificados existentes no sistema.
- ***2.2.07 – Top Used Information:*** O sistema deve permitir que o Auditor escolha como métrica um gráfico que mostre a informação mais utilizada no sistema. O sistema deve permitir que o Auditor escolha o X da variável TopX.
- ***2.2.10 – Users and Events:*** O sistema deve permitir que o Auditor escolha como métrica uma tabela que relacione o número de utilizadores existentes no sistema com o número de acções de cada um destes.

4.5. Requisitos Não Funcionais

Nesta secção são apresentados os requisitos não funcionais identificados no levantamento de requisitos.

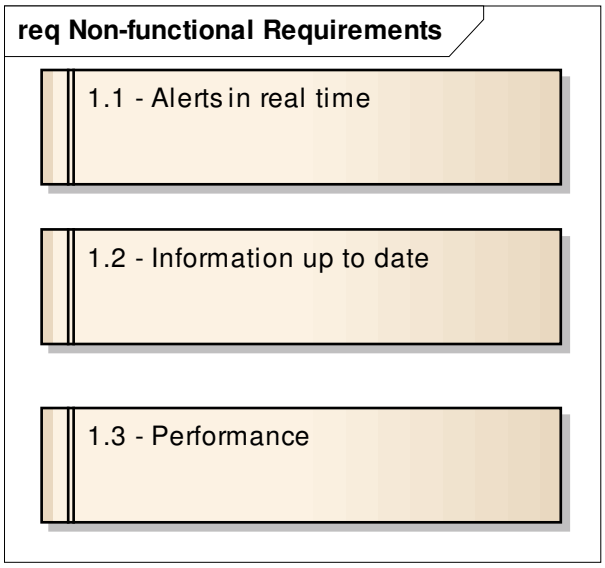


Figura 11: Requisitos não funcionais da Consola de Monitorização

1.1 **Alertas em tempo real:** O sistema deverá disparar os alertas em tempo real (ou seja, imediatamente assim que aconteça algum evento considerado crítico pelo Auditor).

1.2 **Informação actualizada:** A informação disponibilizada pelo sistema deverá estar sempre actual de acordo com o período de carregamento periódico definido.

1.3 **Performance:** A Consola de Monitorização não deverá interferir com o normal uso e performance da aplicação RightsWATCH.

Capítulo 5 - Arquitectura da Consola de Monitorização e escolha de tecnologias

O desenvolvimento da Consola de Monitorização requer o uso de diversas tecnologias e ferramentas, que foram escolhidas com base nos Requisitos e Arquitectura definida. Neste capítulo é apresentada a arquitectura de alto nível e descritas as tecnologias e ferramentas nas quais a Consola de Monitorização do RightsWATCH irá assentar para satisfazer os requisitos propostos.

5.1 Arquitectura de alto nível

Nesta secção é descrita a arquitectura de alto nível do RightsWATCH bem como da Consola de Monitorização implementada, numa perspectiva de integração com o RightsWATCH.

5.1.1 RightsWATCH

Na Figura 12 é possível observar as camadas da arquitectura de alto nível do RightsWATCH.

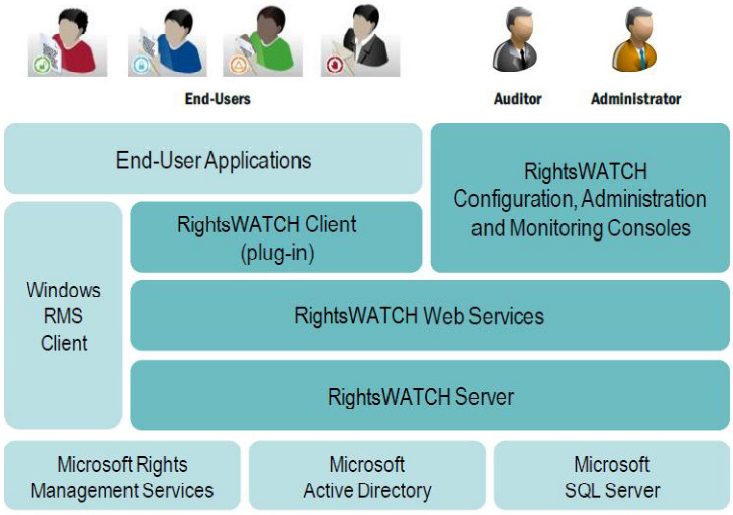


Figura 12: Arquitectura de alto nível do RightsWATCH [6]

O *Microsoft Active Directory* é um serviço de directório da Microsoft que gere as contas dos utilizadores que estão no domínio da rede local. A definição da política de segurança bem como a credenciação e gestão dos direitos dos utilizadores existentes é feita com recurso ao *Microsoft Rights Management Services*. As bases de dados que contêm a informação sobre a política de segurança definida e o registo de eventos do sistema assenta em *Microsoft SQL Server*. Os utilizadores do RightsWATCH utilizam os plug-ins de aplicações existentes, comunicando com o servidor através de um *Web Service*, sendo assim no RightsWATCH *Server* validados os perfis e concedido ou negado o acesso à informação.

5.1.2 Consola de Monitorização do RightsWATCH

Nesta secção é apresentada a arquitectura de alto nível da Consola de Monitorização a desenvolver no presente estágio. A arquitectura da Figura 13 integra também o RightsWATCH e as Consolas de Administração já existentes, encontrando-se contornado a azul o sistema que irá ser acrescentado como trabalho deste estágio.

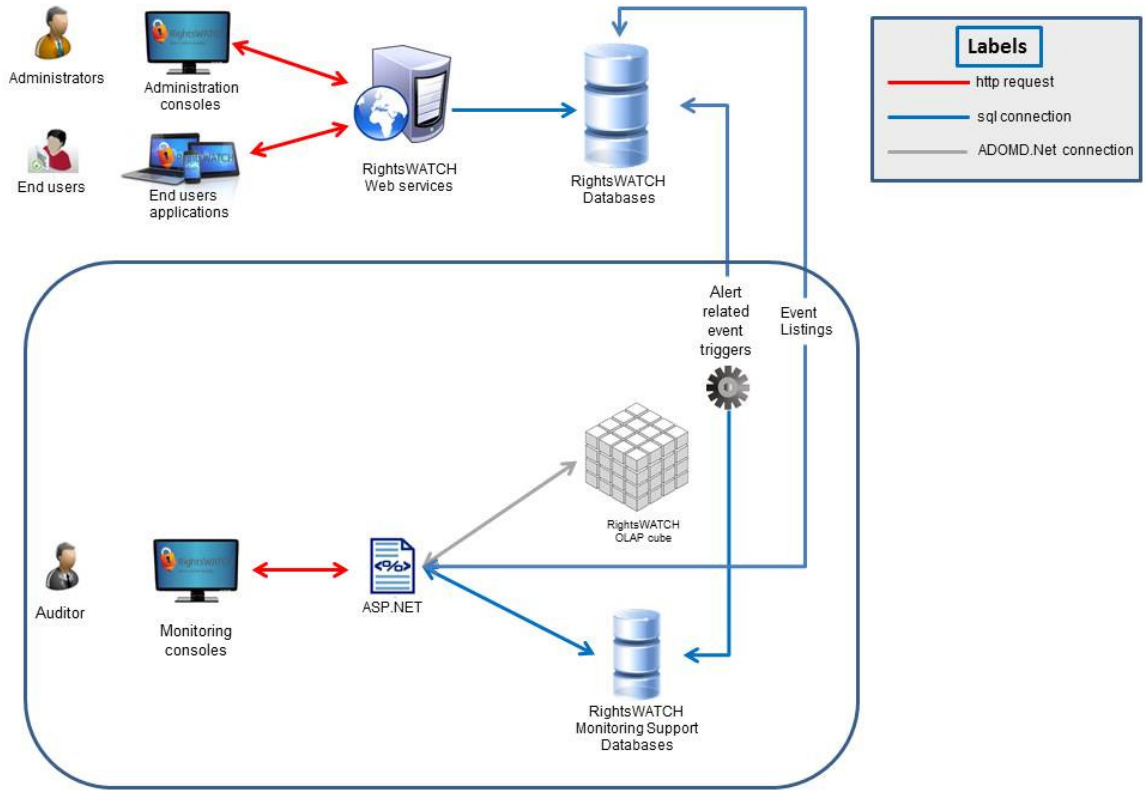


Figura 13: Arquitectura alto nível do sistema

Destacam-se nesta arquitectura algumas decisões técnicas que foi necessário tomar e que são explicadas e justificadas de seguida:

- **Alertas em tempo real:** Uma das limitações das DWH é a sua actualização, que não pode ser realizada em tempo real (ou seja, não reflecte o estado do sistema em cada exacto instante, havendo sempre um período temporal não reflectido na informação da mesma). Assim, houve necessidade de encontrar uma alternativa para o factor tempo real dos alertas. Os dados da DWH não poderiam ser utilizados nesta componente do sistema, pois desse modo o Auditor somente seria notificado da ocorrência dos alertas algum tempo depois de estes aconteceram (de acordo com a periodicidade definida para os carregamentos da DWH), o que não vai de encontro à funcionalidade pretendida dos alertas. Deste modo decidiu-se implementar um *trigger* que, na ocorrência de um evento que se relacione com um evento dos alertas definidos no sistema, colocará esse evento numa BD que será criada para o efeito e se encontra representada na figura 34 *(RightsWATCH Alerts Database*). Deste modo garante-se que os alertas serão executados em tempo real, havendo ainda um histórico dos eventos que dispararam um determinado alerta, comportamento que também se encontra especificado nos requisitos do sistema.
- **Listagem de eventos:** A consola de monitorização existente à data de começo do estágio consistia na secção de Listagem de eventos, pelo que se decidiu manter o mesmo esquema já implementando e assim os dados serão carregados directamente da BD, visto tratar-se somente de uma operação simples, tendo assim o Auditor acesso a todos os eventos em tempo real.
- **Conexão ao cubo *OLAP*:** A informação disponibilizada na secção de *Dashboard* não é retirada directamente da DWH, mas sim de uma estrutura criada de nome *OLAP Cube*, tal como é explicado na subsecção 6.4.4 Processamento do cubo

OLAP. Esta ligação é feita recorrendo a uma biblioteca existente na tecnologia *.NET* da Microsoft, através de uma linguagem de *querying* multidimensional de nome *Multidimensional eXpressions (MDX)* [7]*;*

5.2 Estrutura de Dados

O conceito de Inteligência no Negócio está a ganhar expressão, com as empresas a procurarem cada vez mais perceber ao máximo o que se passa dentro da sua organização, tentando ter a melhor e máxima informação útil disponível.

Ao criar um sistema como o pretendido neste estágio, o primeiro passo e mais importante é a estrutura de dados a ser utilizada. Existindo actualmente uma Base de Dados (BD) Microsoft SQL, que regista os eventos de utilização do RightsWATCH, poder-se-ia pensar em utilizar esta BD e desenvolver a Consola de Monitorização utilizando directamente os dados existentes na BD. No entanto, esta opção apresentaria um conjunto elevado de riscos [8]:

- Os dados nas BD, regra geral, não estão padronizados, e pode acontecer haver campos difíceis de decifrar ou utilização de formatos estranhos;
- As tabelas das BD estão estruturadas para performance e optimização da entrada de dados, tornando-se difícil utilizar estes dados para análises;
- A prioridade tem que ser dada à inserção de dados na BD. Utilizando a BD para análise de dados, a performance destas funções iria ser comprometida o que se reflectiria no software cliente;
- Existe o risco de corrupção de dados de transacção para BD;

Assim, optou-se por utilizar uma *Data Warehouse* (DWH), prática comum no desenvolvimento de aplicações de Inteligência no Negócio. Os eventos continuarão a ser inseridos na BD existente, sendo que esta informação será carregada periodicamente para a DWH. Para o carregamento, os dados passam por um processo conhecido como *ETL*, que é realizado com recurso a uma ferramenta, sendo este processo abordado na secção 5.3 Ferramenta *ETL*. A informação será carregada da BD do RightsWATCH, transformada e inserida na DWH, para assim ser apresentada na Consola de Monitorização.

Existem várias vantagens inerentes a esta abordagem, sendo as mais importantes [8] [9]:

- As DWH são optimizadas para facilitar as consultas e análises aos dados e permitem aceder a um número elevado de registos em cada acesso;
- Uma DWH é desenhada para estar preparada para uma alteração contínua dos seus dados. Quando se acrescentam dados a uma DWH, estes dados são relacionados com os dados já existentes sem que haja qualquer corrupção dos mesmos.
- Com uma DWH preenchida, é possível criar um sistema de BI com um conjunto de funcionalidades relacionadas com análise de dados, que se pretendem desenvolver neste estágio.

5.3 Ferramenta *ETL* e *OLAP*

Para o processo de *ETL*, o uso de uma ferramenta é um factor fundamental no sucesso de um projecto de *Data Warehousing*, pois este processo é o maior consumidor de tempo e a parte mais extensa em todo o projecto.

Existem algumas vantagens inerentes ao uso destas ferramentas, de acordo com *Ralph Kimball* [9]:

- **Fluxo Visual:** Tornam possível seguir visualmente o fluxo da transferência dos dados da fonte para o destino, bem como efectuar a depuração em tempo real durante o processo;
- **Lógica avançada nas transformações:** Possuem transformações e algoritmos avançados, que seriam complicados de desenhar manualmente;
- **Performance de *ETL* elevada:** Implementar manualmente um sistema para manusear grandes volumes de dados com elevada performance é uma tarefa complicada, sendo necessária uma grande experiência para o conseguir. A utilização destas ferramentas elimina este factor.

Com o crescimento do conceito de Inteligência no Negócio, as opções para ferramentas deste tipo vão também elas ganhando expressão no mercado, havendo um número bastante razoável de soluções disponíveis.

Para o objectivo do estágio foi feita uma pesquisa de soluções neste mercado (anexo [4] Estudo de ferramentas de *ETL*, WSW-2013-RPT-00005-dw-technology-research), a fim de escolher a ferramenta que melhor se adequasse aos objectivos do estágio, tendo em conta alguns constrangimentos, que serão detalhados adiante. Foram pesquisadas 5 ferramentas, das quais uma é comercial. Todas as ferramentas comerciais disponibilizam versões *open source*, mas que não serviam para os propósitos do projecto dado o número diminuto de funcionalidades.

Foram identificadas as seguintes ferramentas:

- ***Microsoft Business Intelligence Development Studio (BIDS):*** Serviços disponibilizados pela Microsoft nos pacotes SQL Server 2005, 2008 e 2012 a partir da edição *Standard*, nos quais se inclui uma ferramenta de *ETL* para DWH (*SQL Server Integration Services - SSIS*) e também uma ferramenta de análise *OLAP* (*SQL Server Analysis Services - SSAS*). O *SSIS* permite a transformação de dados de várias fontes (Microsoft ou não), como por exemplo *SQL Server*, *Oracle, Teradata,* ficheiros de texto, entre outros. É referido que a ferramenta *ETL* do *SSIS* é uma das mais rápidas do mercado a mover quantidades elevadas de dados.
- ***Pentaho Data Integration (PDI):*** A empresa *Pentaho* disponibiliza duas *packages* : *Data Integration* e *Business Analytics*. A *package* da *Pentaho* que inclui os serviços de *ETL* é a primeira – *PDI :* possibilita a extracção e transformação de informação das fontes, disponibilizando a informação pronta a ser analisada. Inclui ligação a várias fontes de dados, como por exemplo *MySQL, SQL Server, Oracle, NoSQL, SAP,* entre outras. A arquitectura da *Pentaho* inclui processamento em paralelo para acelerar a transformação e carregamento de dados. O revendedor da *Pentaho* em Portugal contactou o estagiário, após um pedido de plano de preços, sendo que foi feita uma estimativa de 9 mil euros para a utilização do *PDI*, e ainda foram abordadas eventuais complicações para vender um produto que utilizasse *PDI* para fora de um determinado conjunto de países. Estes pontos fizeram com que esta ferramenta não fosse a escolhida para o desenvolvimento do projecto de estágio.
- ***Talend Data Integration:*** *Package* da empresa *Talend* para extrair, transformar e carregar dados para uma DWH. Alguns exemplos de fontes de dados suportadas por esta ferramenta são *Microsoft SQL Server, MySQL, NoSQL,* entre outras. É referido a existência de um número elevado de conectores (450) para aplicar transformações de dados. Existe ainda uma versão *open source*, mas que perdeu para a *Pentaho Kettle*

Community Edition em funcionalidades e também em termos de documentação disponível.

- ***CloverETL:*** Produto da empresa *Clover* que contém um número elevado de funcionalidades de *ETL*, com um especial enfoque na produtividade e performance do processo *ETL*. Foi agendada com a empresa *Clover* uma demonstração via *Web* do produto, onde se percebeu que o *CloverETL* é uma ferramenta extremamente poderosa para *ETL*, mas cujo orçamento para aquisição de licença se revelou alto: 25 mil euros.
- ***Pentaho Data Integration Community Edition (Kettle):*** Produto com as mesmas funcionalidades de *ETL* do produto comercial *Pentaho Data Integration* (descrito acima), excluindo o suporte técnico profissional que este disponibiliza. Assente no verdadeiro conceito de *open source*, há bastante informação de suporte na *wiki* do projecto, havendo regularmente novas versões do produto (a última versão, *Kettle 4.4.0* foi disponibilizada no dia 29 de Novembro de 2012).

Foram tidos em linha de conta alguns factores aquando da escolha da ferramenta, nomeadamente:

- **Necessidades vs Funcionalidades:** Quais as necessidades em termos de *ETL* (por exemplo ao nível de transformações necessárias) para a Consola de Monitorização e quais as funcionalidades oferecidas (por exemplo a existência de linguagem de scripting);
- **Integração:** Ter em conta as fontes e destino dos dados e estudar o suporte dado por cada aplicação;
- **Custo:** Ter em conta o custo de cada uma das soluções estudadas, pois além do impacto que teriam para a *Watchful Software*, este custo iria ser reflectido no valor de venda do *RightsWATCH*;
- **Parcerias:** A *Watchful Software* é *Silver Partner* da Microsoft*;*
- **Estudos de mercado:** Foram lidos alguns artigos e estudos, sendo dado enfâse a um estudo da conhecida *Gartner* [10].

Após estudo dos pontos acima e discussão dos mesmos com o *Technical Manager* do projecto optou-se por utilizar o Microsoft BIDS. Os factores estudados não tiveram todos o mesmo peso na decisão, tendo contríbuido decisivamente para esta escolha o facto da *Watchful Software* trabalhar com base em infrastruturas *Microsoft,* aliado ao facto de todos os clientes da Watchful Software utilizarem a versão *Standard* do SQL Server. Nesta versão incluem-se as ferramentas necessárias para o processo de *ETL*, tornando-se possível utilizá-las sem encarecer o preço do RightsWATCH com licenças extras.

5.4 Framework de desenvolvimento

A tendência actual do mercado é o desenvolvimento de aplicações *Web*. No caso do RightsWATCH, as Consolas de administração existentes seguem esse modelo aplicacional, pelo que tendo por base estas premissas, decidiu-se que a Consola de Monitorização também seria uma aplicação *Web*.

Assim, foi elaborada uma pesquisa de *frameworks* de desenvolvimento *Web*, com um especial enfoque nas capacidades de *User Interface*, visto que a Consola de Monitorização assenta no conceito de visualização da informação, sendo portanto a componente de visualização parte fundamental do sistema.

Foram identificados os seguintes *frameworks*:

- ***SproutCore:*** *Framework* baseado em *Ruby*;
- ***ZK Framework:*** *Framework* com versão *open source* e baseado em *Java*, que é utilizada por empresas de renome como a *Sony, Sun, Adobe* e *IBM*. No entanto, a versão *open source* não suporta algumas funcionalidades fundamentais para o objectivo do estágio, como a componente gráfica, bastante rudimentar neste aspecto;
- ***Sencha EXT JS:*** *Framework* baseado em *Javascript* e que é actualmente utilizada pela Watchful Software no desenvolvimento das Consolas de administração. Tem como um dos pontos fortes e imagem de marca uma componente de *User Interface* bastante evoluída e com imensas funcionalidades;
- ***Qooxdoo:*** *Framework* baseado em *Javascript;*

Havendo já na Watchful Software a experiência positiva com a utilização do *Sencha EXT JS*, esta foi a ferramenta que foi investigada mais a fundo para ter a certeza que cobriria os requisitos necessários para a Consola de Monitorização.

A Framework *Sencha EXT JS* é caracterizada por:

- Suportar os *browsers* mais utilizados (Google Chrome, Safari, Internet Explorer, etc.);
- Disponibilizar vários *widgets* visuais (grelhas, listas, menus,etc.);
- Disponibilização de um conjunto de métodos e classes que agilizam o desenvolvimento de interfaces gráficas;
- Possibilidade de utilizar *frameworks* para testes de código como o *Jasmine* ou o *Siesta*, esta última optimizada especificamente para o *Sencha EXT JS* para efeitos de testes unitários;
- Suporte, treino e documentação online;

Assim, a escolha acabou por incidir no *Sencha EXT JS* [11], tendo por base diversos factores:

- O *Sencha EXT JS* já é utilizado na Watchful Software, nas Consolas de administração do RightsWATCH;
- Desnecessário adquirir outra *framework*, com custos associados;
- Facilitação da gestão e reutilização de código;
- Consistência no aspecto gráfico das Consolas de administração e monitorização;
- Forte componente gráfica, facto a que não é alheio a combinação de esforços da empresa que desenvolve a *framework* com o projecto *Raphael,* uma biblioteca *Javascript* para simplificação do manuseamento de gráficos na *Web*;

Com vista ao desenvolvimento do diagrama de *Information Tracking* teve que ser escolhida uma *framework* que permitisse desenhar o tipo de gráfico pretendido. O Sencha EXT JS apesar das funcionalidades gráficas, não suporta o desenho deste tipo de diagramas, pelo que foram estudadas duas alternativas:

- **JointJS:** Framework de JavaScript *open source* que permite a criação de vários tipos de diagrama bem como a possibilidade de implementar interacções sobre os elementos que compõem os diagramas. Para implementar os diagramas, esta framework faz uso da framework *Raphael*.
- **D3JS:** Framework de JavaScript para manipulação de documentos de dados. Permite a criação de gráficos a partir de ficheiros com informação (JSON, XML,…). Suporta vários tipos de gráficos, sendo que uma possibilidade para o pretendido na secção de *Information Tracking* era o *"TreeGraph"*.

Tendo em conta os requisitos relacionados com o tempo dos eventos, era necessário a utilização de um diagrama onde a posição dos eventos pudesse ser controlada. Assim, a escolha recaiu na JointJS, visto que utilizando gráficos da D3JS, as posições dos mesmos teriam que ser fixas, impossibilitando o espaçamento entre eventos de acordo com o período temporal dos mesmos.

5.5 Arquitectura do Sistema

Nesta secção é apresentado o diagrama de classes da Consola de Monitorização do RightsWATCH, sendo descritas somente as classes e métodos relacionados com a secção de *Dashboard*.

É possível consultar o documento completo de desenho detalhado do Sistema no anexo [5] Desenho detalhado do sistema, com a especificação completa das Consolas de Monitorização, incluindo todas as classes.

A Figura 14 mostra o diagrama da arquitectura da aplicação, seguindo a lógica já implementada pela Watchful Software – modelo *n-tier*, composto pela Camada de Interface *(Interface Layer)* , Camada de acesso a dados *(Data Access Layer)* e Camada de Lógica*(Business Logic Layer)*.

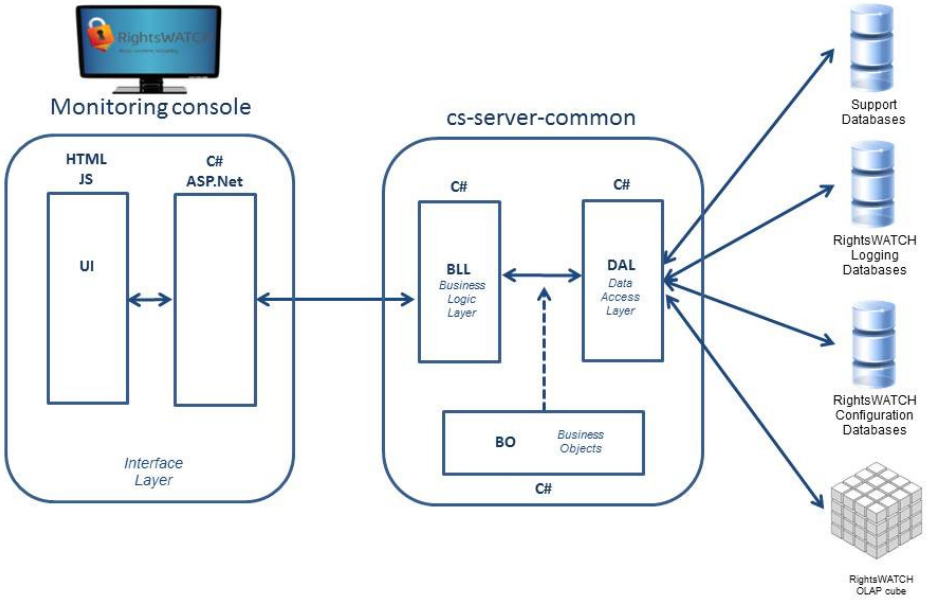


Figura 14: Arquitectura da aplicação

A lógica da aplicação e o acesso a dados é feito nos componentes comuns da aplicação *(cs-server-common)*, estando divididos em:

- ***Data Access Layer (DAL):*** Contém a implementação do código responsável pelo acesso a dados, tando das BDs do RightsWATCH como da DWH;
- ***Business Logic Layer (BLL):*** Contém a implementação da lógica da aplicação, incluindo pedidos à camada de dados e entrega dos mesmos à camada de Interface;
- ***Business Objects (BO):*** Contém as classes tipo da aplicação.

Na Figura 15 estão especificadas as classes existentes em cada uma das componentes descritas acima.

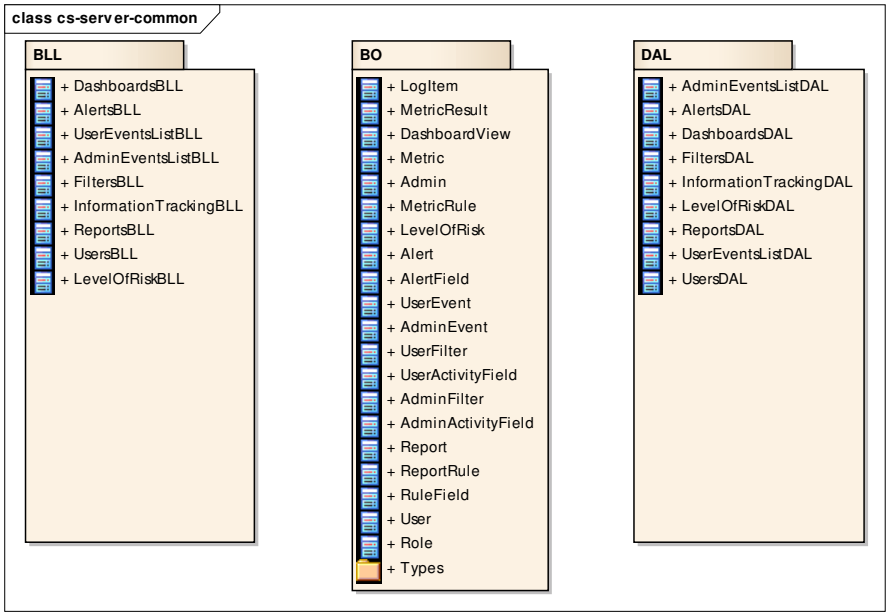


Figura 15: Classes cs-server-common

5.5.1 Business Logic Layer

Na Figura 16 encontram-se as classes e respectivas funções que compõem a *BLL* da Consola de Monitorização. A especificação completa das classes de *BLL* encontra-se o anexo [5] Desenho detalhado do sistema.

Destacam-se as classes das secções implementadas:

- ***DashboardBLL:*** Inclui as funções relacionadas com a lógica de negócio das funcionalidades da secção de *Dashboard*, incluindo os métodos para listas as métricas existentes, executar métricas ou filtrar as métricas por período temporal
- ***InformationTrackingBLL:*** Inclui as funções relacionadas com a lógica de negócio das funcionalidades da secção de *Information Tracking*, incluindo o método para pesquisar o ciclo de vida completo de uma informação;

Figura 16: Classes e funções da *BLL* da Consola de Monitorização

5.5.2 Business Objects

Na Figura 17 encontram-se os BO definidos que foram utilizados na implementação da Consola de Monitorização. A especificação completa dos *BOs* encontra-se o anexo [5] Desenho detalhado do sistema.

Destacam-se os BO das secções implementadas:

- ***Metric:*** Define uma métrica existente na secção de *Dashboard*, que é composta por um identificador, o nome e as regras que a definem.
- ***MetricRule:*** Define uma regra que, em conjunto com outra(s) definirá uma métrica da secção de *Dashboard*.

- ***MetricResult:*** Define o resultado da execução de uma métrica, sendo composta por um objecto do tipo *CellSet*, contendo o resultado da execução de uma *query* multidimensional.
- ***LogItem:*** Define um evento feito com o RightsWATCH.

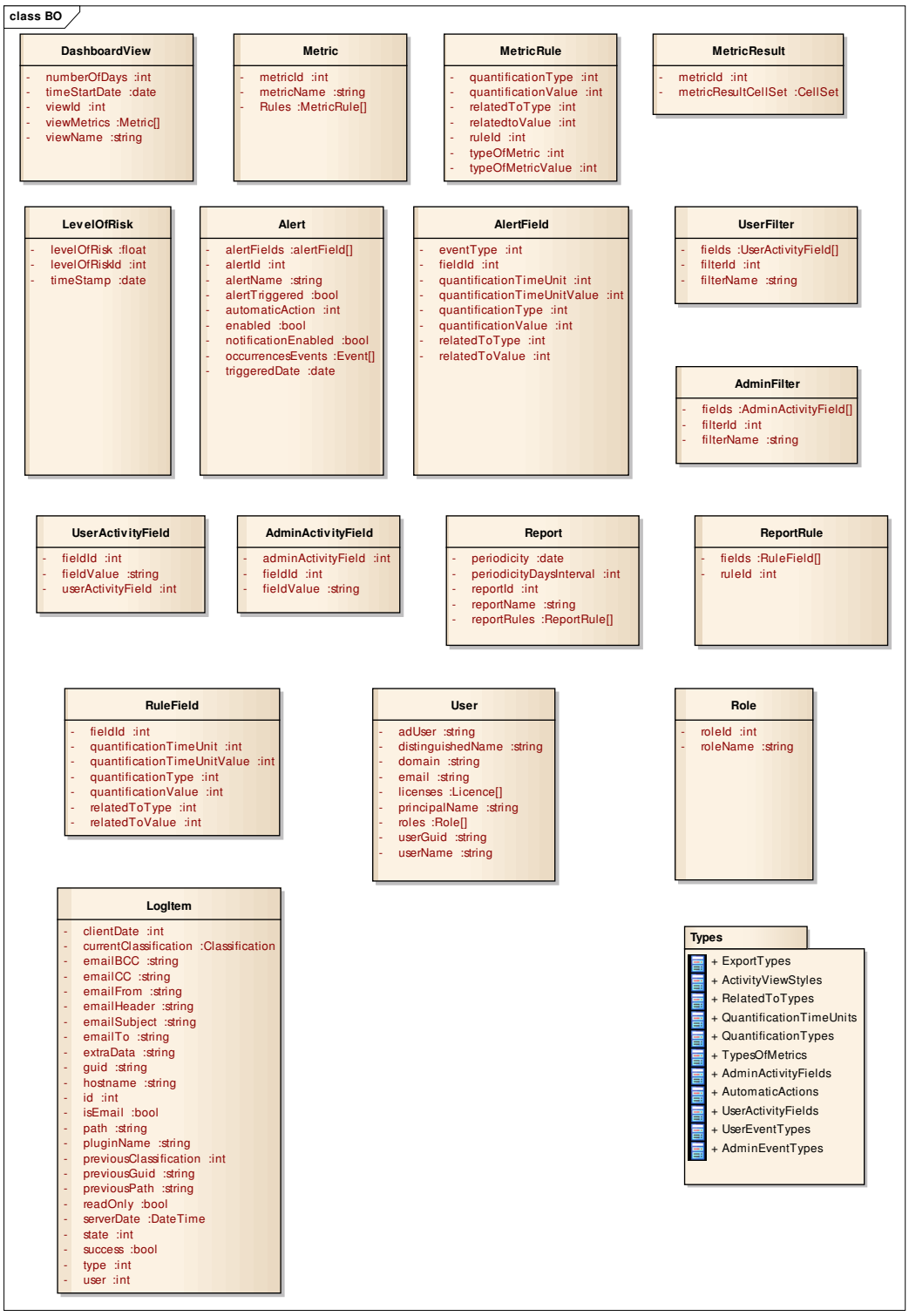


Figura 17: *BOs* da Consola de Monitorização

5.5.3 Data Access Layer

Na Figura 18 encontram-se as classes e respectivas funções que compõem a *DAL* da Consola de Monitorização. A especificação completa das classes de *DAL* encontra-se o anexo [5] Desenho detalhado do sistema.

Destacam-se as classes das secções implementadas:

- ***DashboardDAL:*** Inclui as funções relacionadas com o acesso a dados na secção de *Dashboard*, incluindo as ligações às BD que contêm as métricas e ao cubo *OLAP* para execução das mesmas.
- ***InformationTrackingDAL:*** Inclui as funções relacionadas com o acesso a dados para a secção de *Information Tracking*, incluindo as ligações à BD que contém os eventos realizados com o RightsWATCH.

Figura 18: Classes e funções da *DAL* da Consola de Monitorização

Capítulo 6 - Arquitectura da Data Warehouse e processo *ETL*

Neste capítulo é descrita a Arquitectura da DWH, incluindo o modelo multidimensional escolhido, as fontes de dados utilizadas, o processo de *ETL* e a criação e processamento do cubo *OLAP*. Esta informação encontra-se de forma mais detalhada no anexo [5] Desenho detalhado do sistema.

6.1 Desenho detalhado da Data Warehouse

Nesta secção é apresentado o desenho detalhado da DWH, incluindo as fontes de dados utilizadas, o modelo multidimensional da DWH, as transformações do processo *ETL* e o cubo *OLAP*.

6.1.1 Fontes de Dados

Para o carregamento da DWH são utilizadas as bases de dados SQL do RightsWATCH.

De seguida são apresentadas as BD e tabelas utilizadas bem como uma breve descrição dos campos da tabela de registo dos eventos realizados pelos utilizadores no RightsWATCH.

6.1.1.1 RightsWATCH_logging

Nesta BD encontram-se as tabelas relativas ao registo de eventos dos utilizadores.

Esta é a fonte de dados utilizada para o carregamento das tabelas de facto relativas à utilização do sistema e também de algumas dimensões, sendo dado especial utilização à tabela *LogItems*.

De seguida apresenta-se o diagrama desta BD.

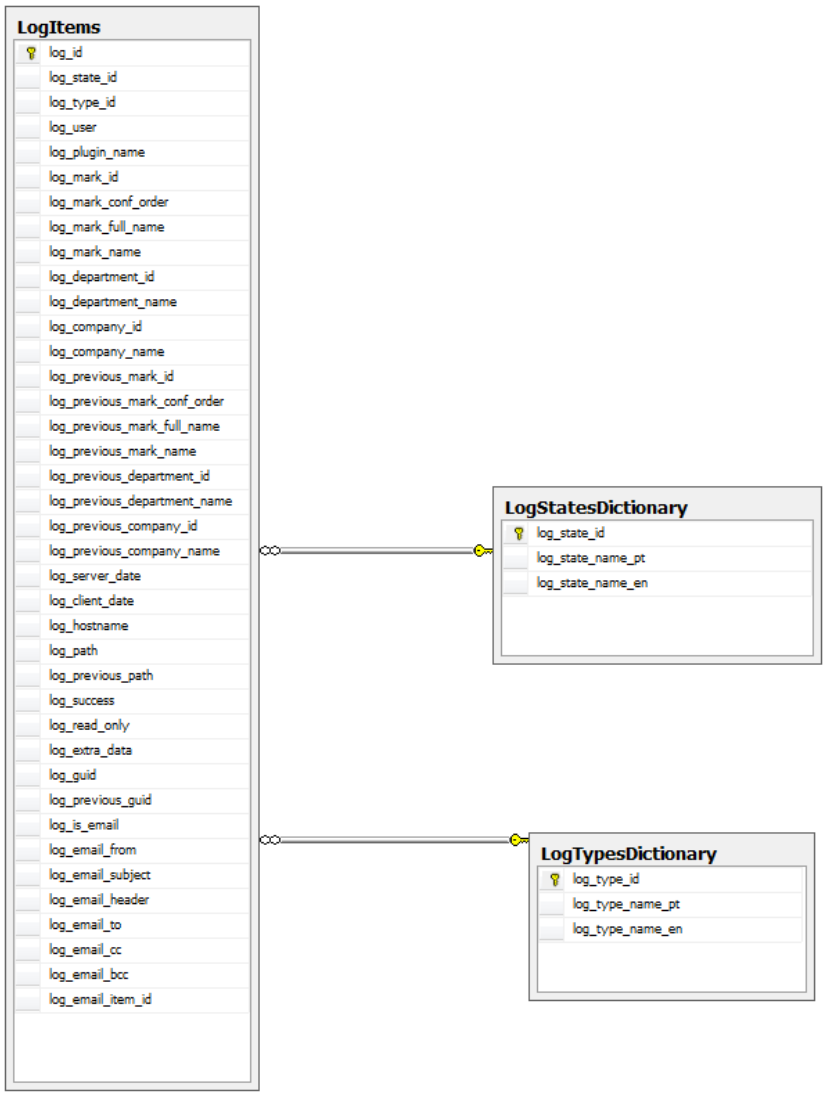


Figura 19: Diagrama da BD RightsWatch_Logging

De seguida é feita uma descrição das tabelas de configuração do RightsWATCH utilizadas para o carregamento da DWH.

Tabela	**Descrição**
LogItems	Tabela que contém os eventos realizados por utilizadores do RightsWATCH.
LogStatesDictionary	Tabela que contém todos os estados dos eventos realizados pelos utilizadores do RightsWATCH (*online, offline,..)*
LogTypesDictionary	Tabela que contém os tipos de eventos definidos no sistema fonte.

Tabela 2: Descrição das tabelas *logging* do RightsWATCH

A tabela *LogItems* tem em cada entrada um evento realizado no RightsWATCH, contendo cada um dos campos informação relativa com o evento. A tabela seguinte descreve os campos mais relevantes desta tabela:

LogItems	
Atributo	**Descrição**
Log_type_id	Tipo de evento realizado, de acordo com o definido na tabela *LogTypesDictionary*.
Log_user	Utilizador que realizou o evento.
Log_mark_conf_order	Nível de confidencialidade da classificação relacionada com a informação sobre a qual o evento foi realizado.
Log_server_date	Data no servidor em que o evento foi realizado.
Log_success	Regista com valor 0 ou 1 o insucesso ou sucesso da operação, respectivamente.
Log_guid	Identificador único da informação relacionada com o evento realizado.
Log_previous_guid	No caso de uma alteração do identificador único da informação, regista o identificador anterior.

Tabela 3: Descrição da tabela *LogItems*

6.1.1.2 RightsWATCH_Admin

Nesta BD encontram-se as tabelas relativas à configuração da política de privacidade do RightsWATCH.

Esta é a fonte de dados utilizada para o carregamento de maioria das dimensões da DWH.

De seguida apresenta-se o diagrama desta BD.

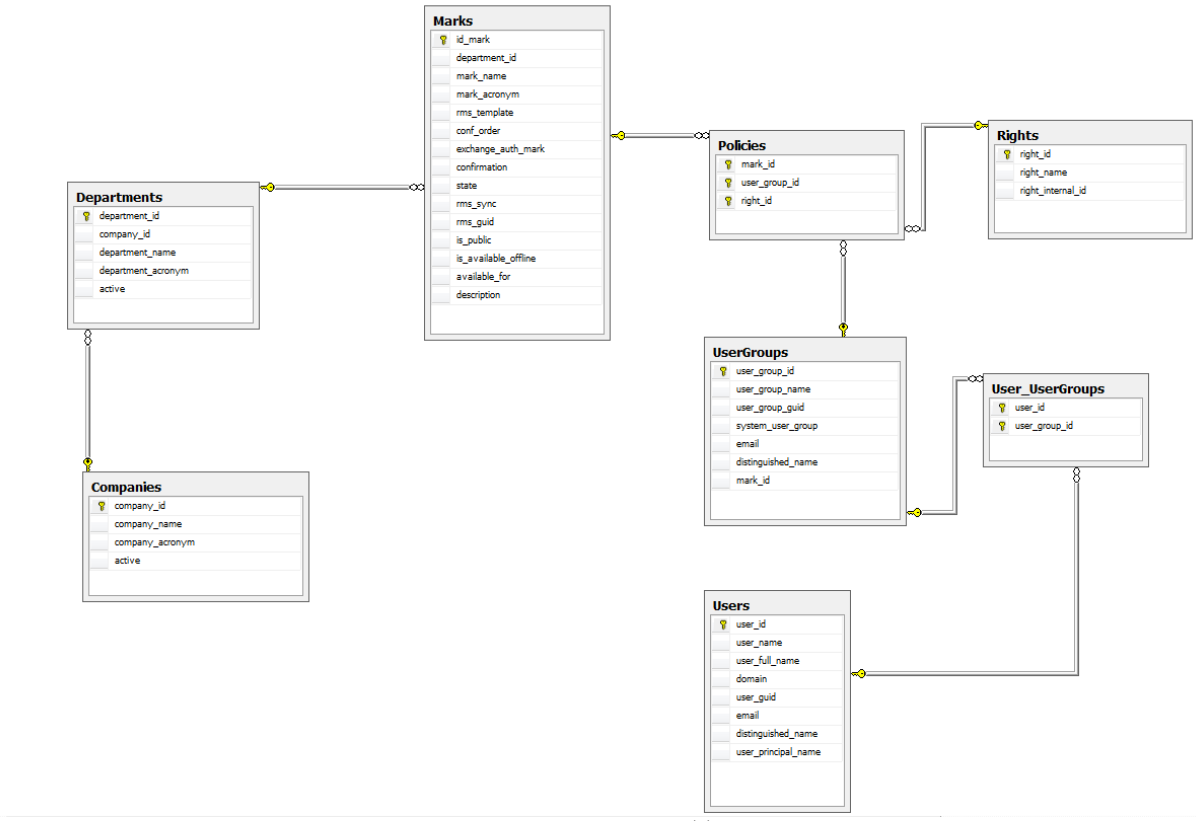


Figura 20: Diagrama da BD RightsWATCH_Admin

De seguida é feita uma descrição das tabelas de configuração do RightsWATCH utilizadas para o carregamento da DWH.

Tabela	**Descrição**
Companies	Tabela que contém as empresas existentes no RightsWATCH.
Departments	Tabela que contém os âmbitos existentes no RightsWATCH.
Marks	Tabela que contém todas os níveis de classificação de informação existentes no RightsWATCH.
Users	Tabela que contém todos os utilizadores existentes no RightsWATCH.
UserGroups	Tabela que contém todos os perfis *(roles)* existentes no RightsWATCH.
User_UserGroups	Tabela que contém a associação dos perfis a cada utilizador do RightsWATCH.

Tabela 4: Descrição das tabelas de configuração utilizadas para carregamento da DWH

6.1.2 Modelo de dados multidimensional

O modelo de dados escolhido para a DWH é o tradicional Esquema em Estrela, que é composto por tabelas de Facto e tabelas de Dimensão.

As tabelas de Facto guardam os factos ocorridos (ou seja, os acontecimentos) e são utilizadas para medir os requisitos de negócio identificados, sendo que geralmente são

tabelas compostas por chaves estrangeiras (que servem para ligar às tabelas de dimensão) e por atributos numéricos e aditivos. No caso do RightsWATCH, as tabelas de Facto somente têm chaves externas, sendo por isso chamadas de "Tabelas de Facto sem factos", em que cada entrada é somente composta pelo conjunto de chaves das dimensões que caracterizam cada evento. Esta é uma prática comum em DWHs que servem para marcar acontecimentos/registar eventos.

As tabelas de Dimensão caracterizam um determinado requisito de negócio, sendo cada uma das tabelas de dimensão composta por um ou mais atributos. Uma dimensão é definida pela sua chave primária que ligará à tabela de Factos e por atributos que a caracterizam.

Num esquema em estrela tradicional, a relação entre Factos e Dimensões é N:1 , ou seja múltiplas entradas na tabela de factos ligam a uma e só uma entrada na tabela de dimensão, enquanto que uma entrada na tabela de dimensão pode ligar a mais do que uma entrada na tabela de factos.

A presente secção descreve o esquema em estrela utilizado para a implementação da DWH do RightsWATCH. Foram identificadas 3 tabelas de Facto (a azul, na Figura 21) e 10 tabelas de Dimensão (a amarelo, na Figura 21).

Podemos ver na Figura 21 o esquema multidimensional implementado em alto nível, sendo depois este descrito nas subsecções seguintes. O diagrama físico do esquema multidimensional da DWH pode ser consultado no anexo [5] Desenho detalhado do sistema.

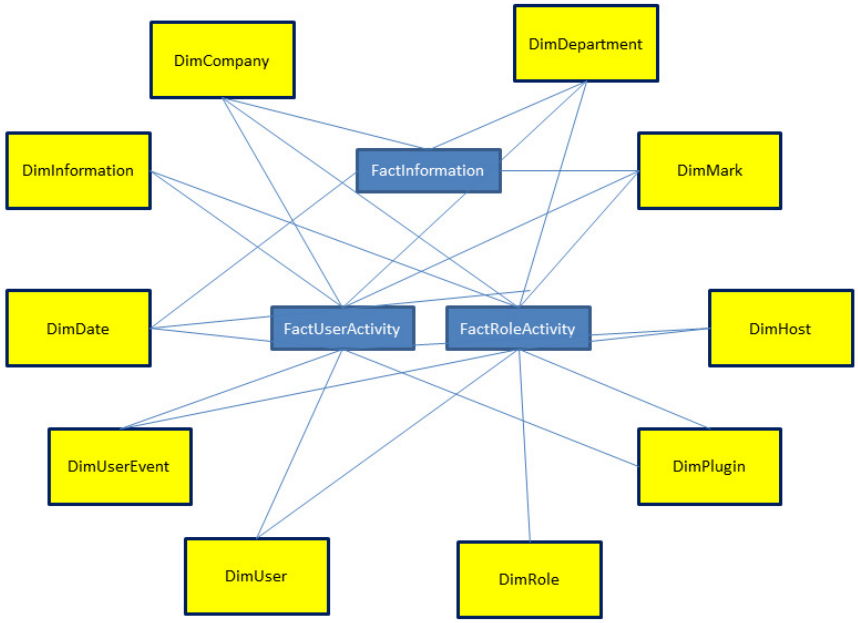


Figura 21: Esquema em estrela DWH RightsWATCH

6.1.2.1 Tabelas de Facto

As necessidades identificadas, que serviram para o levantamento de requisitos da Consola de Monitorização, levaram à criação de 3 tabelas de facto, cada uma com um propósito específico em termos de monitorização.

Um dos requisitos era a contagem de eventos realizados por perfil de utilizador, sendo que cada utilizador tem um ou mais perfis associados, onde cada perfil é o conjunto dos seus direitos no RightsWATCH. Sendo a relação entre tabelas de facto e dimensão N:1, como explicado em cima, tornava-se impossível ligar a dimensão *DimRole* (identificada no diagrama anterior) à tabela de facto *FactUserActivity*, pois um utilizador ao ter mais do que 1 perfil, não iria ser respeitada a relação de N:1 entre a tabela *FactUserActivity* e a tabela *DimRole*.

Para ultrapassar e solucionar este problema foram identificadas 3 alternativas [12]:

- ***Tabela de ligação (Bridge Table):*** Criação de uma tabela intermédia entre a tabela *FactUserActivity* e a *DimRole*, que permitisse manter todas as ligações de N:1 entre as tabelas. Foi identificado como ponto negativo um aumento na performance das queries bem como a sua complexidade.
- **Dimensão com todas as combinações de perfis:** Na tabela *DimRole* ter todas as combinações possíveis de perfis. Sendo que o número de perfis existente numa configuração do RightsWATCH não é fixo, o número de entradas (e respectivo tamanho da tabela) iria ser extremamente elevado. Segundo o artigo citado em [12] , para combinar 40 *roles* seriam necessários aproximadamente 10 TB de espaço físico no disco para a tabela *DimRole*.
- **Nova tabela de factos com granularidade reduzida:** Esta opção visa a criação de uma nova tabela de factos que respeite a granularidade mais fina da dimensão em causa, ou seja, criar uma tabela de factos em que cada entrada diga respeito a somente um, implicando a replicação de cada evento tantas vezes quantos os perfis do utilizador que o executou. Assim facilita-se as *queries* e obtém-se melhor *performance* pois evita operações *join* adicionais.

Após consideração das soluções possíveis, optou-se pela 3ª, visto que leva à simplificação das *queries* com uma *performance* elevada. Assim, foi criada uma tabela de factos respectiva aos eventos por *role: FactRoleActivity*.

As 3 tabelas de factos criadas foram as seguintes:

- *FactUserActivity:* Tabela que tem por objectivo registar os eventos realizados pelos utilizadores, tendo granularidade de um evento realizado por utilizador em cada linha da tabela;
- *FactRoleActivity:* Tabela que tal como a anterior regista os eventos realizados, mas tendo como granularidade um evento realizado por um determinado perfil em cada linha da tabela.
- *FactInformation:* Tabela que controla a quantidade de informação encriptada com o RightsWATCH, tendo a granularidade de uma informação encriptada por cada linha da tabela;

De seguida são descritas as três tabelas de factos e os respectivos atributos.

FactUserActivity	
Atributo	**Descrição**
User_activity_date_key	Chave que identifica a data em que o evento ocorreu.
Company_key	Chave que identifica a empresa associada à classificação do documento sobre o qual o evento foi realizado.
Department_key	Chave que identifica o âmbito associado à classificação do documento sobre o qual o evento foi realizado.
Mark_key	Chave que identifica o nível de classificação associado à classificação do documento sobre o qual o evento foi realizado.
Plugin_key	Chave que identifica o *plugin* utilizado para realizar o evento.
User_key	Chave que identifica o utilizador que realizou o evento.
User_event_key	Chave que identifica o tipo de evento realizado.
Localhost_key	Chave que identifica a máquina onde o evento foi realizado.
Information_key	Chave que identifica a informação sobre a qual o evento foi realizado.

Tabela 5: Descrição da tabela de facto FactUserActivity

FactRoleActivity	
Atributo	**Descrição**
User_activity_date_key	Chave que identifica a data em que o evento ocorreu.
Company_key	Chave que identifica a empresa associada à classificação do documento sobre o qual o evento foi realizado.
Department_key	Chave que identifica o âmbito associado à classificação do documento sobre o qual o evento foi realizado.
Mark_key	Chave que identifica o nível de classificação associado à classificação do documento sobre o qual o evento foi realizado.
Plugin_key	Chave que identifica o *plugin* utilizado para realizar o evento.
User_key	Chave que identifica o utilizador que realizou o evento.
User_event_key	Chave que identifica o tipo de evento realizado.
Localhost_key	Chave que identifica a máquina onde o evento foi realizado.
Information_key	Chave que identifica a informação sobre a qual o evento foi realizado.
Role_key	Chave que identifica o perfil *(role)* que realizou a acção.

Tabela 6: Descrição da tabela de Factos FactRoleActivity

FactInformation	
Atributo	**Descrição**
User_activity_date_key	Chave que identifica a data em que a informação/documento foi criada.
Company_key	Chave que identifica a empresa associada à classificação do documento criado.
Department_key	Chave que identifica o âmbito associado à classificação do documento criado.
Mark_key	Chave que identifica o nível de classificação associado à classificação do documento criado.
Information_key	Chave que identifica a informação criada.

Tabela 7: Descrição da tabela de Factos FactInformation

6.1.2.2 Tabelas de Dimensão

Para caracterizar as tabelas de Facto identificadas, foram criadas 10 tabelas de Dimensão, tendo como pressuposto que as mesmas servem para caracterizar uma interacção de um utilizador com o RightsWATCH. As tabelas identificadas foram as seguintes:

- *DimInformation:* Dimensão que contém todas as informações que são encriptadas com o RightsWATCH.
- *DimCompany:* Dimensão que contém todas as empresas que existem na configuração do RightsWATCH.
- *DimDepartment:* Dimensão que contém todos os âmbitos que existem na configuração do RightsWATCH.
- *DimMark:* Dimensão que contém todos os níveis de classificação que existem na configuração do RightsWATCH.
- *DimHost:* Dimensão que contém todos os *hosts* com operações realizadas no RightsWATCH.
- *DimPlugin:* Dimensão que contém todos os *plugins* com os quais é possível utilizar o RightsWATCH.
- *DimRole:* Dimensão que contém todos os perfis que existem na configuração do RightsWATCH.
- *DimUser:* Dimensão que contém todos os utilizadores que existem na configuração do RightsWATCH.
- *DimUserEvent:* Dimensão que contém todos os tipos de eventos passíveis de serem realizados no RightsWATCH.
- *DimDate:* Dimensão de tempo, que contém as datas com a granularidade horária, ou seja, cada dia terá 24 entradas nesta tabela.

De seguida são descritas as três tabelas de dimensão e os respectivos atributos.

DimInformation	
Atributo	**Descrição**
Information_key	Chave primária da informação na tabela de dimensão.
Information_guid	Identificador único da informação.
Owner_user	Utilizador que criou a informação.
Information_type	Tipo do documento/informação.

Tabela 8: Descrição da tabela de dimensão *DimInformation*

DimCompany	
Atributo	**Descrição**
Company_key	Chave primária da empresa na tabela de dimensão.
Company_name	Nome da empresa.
Company_business_key	Chave primária da empresa na tabela fonte, utilizada para fazer o *matching* entre a empresa associada ao evento e o seu valor na dimensão.

Tabela 9: Descrição da tabela de dimensão *DimCompany*

DimDepartment	
Atributo	**Descrição**
Department_key	Chave primária do âmbito na tabela de dimensão.
Department_name	Nome do âmbito.
Department_business_key	Chave primária do âmbito na tabela fonte, utilizada para fazer o *matching* entre o âmbito associado ao evento e o seu valor na dimensão.

Tabela 10: Descrição da tabela de dimensão *DimDepartment*

DimMark	
Atributo	**Descrição**
Mark_key	Chave primária do nível de classificação na tabela de dimensão.
Mark_name	Nome do nível de classificação.
Mark_is_public	*Boolean* que identifica se o nível é público.
Mark_confidentiality_order	O nível de confidencialidade do nível.
Mark_business_key	Chave primária do nível de classificação na tabela fonte, utilizada para fazer o *matching* entre o nível associada ao evento e o seu valor na dimensão.

Tabela 11: Descrição da tabela de dimensão *DimMark*

DimHost	
Atributo	**Descrição**
Host_key	Chave primária da máquina na tabela de dimensão.
Host_name	Nome/endereço da máquina.

Tabela 12: Descrição da tabela de dimensão *DimHost*

DimPlugin	
Atributo	**Descrição**
Plugin_key	Chave primária do *plugin* na tabela de dimensão.
Plugin_name	Nome do *plugin*.
Plugin_business_key	Chave do *plugin* no sistema fonte, utilizada para fazer o *matching* entre o *plugin* associado ao evento e o seu valor na dimensão.

Tabela 13: Descrição da tabela de dimensão *DimPlugin*

DimRole	
Atributo	**Descrição**
Role_key	Chave primária do perfil na tabela de dimensão.
Role_name	Nome do perfil.
Role_guid	Identificador único do perfil.
Role_business_key	Chave primária do perfil na tabela fonte, utilizada para fazer o *matching* entre o perfil associado ao evento e o seu valor na dimensão.

Tabela 14: Descrição da tabela de dimensão *DimRole*

DimUser	
Atributo	**Descrição**
User_key	Chave primária do utilizador na tabela de dimensão.
User_name	Nome do utilizador.
User_GUID	Identificador único do utilizador na *Active Directory*.
User_login	*Login* do user no domínio da empresa.
User_business_key	Chave primária do utilizador na tabela fonte, utilizada para fazer o *matching* entre o utilizador que efectuou o evento e o seu valor na dimensão.

Tabela 15: Descrição da tabela de dimensão *DimUser*

DimUserEvent	
Atributo	**Descrição**
User_event_key	Chave primária do tipo de evento na tabela de dimensão.
User_event_type	Tipo do evento.

Tabela 16: Descrição da tabela de dimensão *DimUserEvent*

DimDate	
Atributo	**Descrição**
User_activity_date_key	Chave primária da data na tabela de dimensão.
Date	Data no formato " YYYY-MM-DD".
Day_number	O número do dia do mês.
Day_of_week	O nome do dia da semana.
Month_name	O nome do mês.
Hour	A hora do dia.
Is_weekend	*Boolean* que indica se a data é fim de semana.
Is_leap_year	*Boolean* que indica se a data é de um ano bissexto.
Year	Ano.
Week	O número da semana do ano.
Timestamp	*Timestamp* no formato "YYYYMMDDHHmmSS".

Tabela 17: Descrição da tabela de dimensão DimDate

6.2 Mapa de transformações do processo de *ETL*

A presente secção descreve o processo de *ETL* para as dimensões e factos que compõem a DWH da Consola de Monitorização do RightsWATCH, incluindo as necessárias transformações.

Este processo foi realizado utilizando a ferramenta *Microsoft SSIS*, sendo que cada tabela de dimensão e facto tem uma *package* associada. Em cada uma destas packages são utilizados dois mecanismos para efectuar o processo de *ETL*:

- ***Control Flow:*** Define o fluxo das tarefas a serem executadas, permitindo o controlo lógico da *package*. É um processo mais alto nível, sendo que o processo de *ETL* é definido num *Data Flow* (descrito em baixo) que é especificado no *Control Flow*.
- ***Data Flow:*** Define o fluxo de dados desde uma fonte até um destino, com as transformações a serem aqui aplicadas antes dos dados chegarem à fonte.

6.2.1 Tarefas de *Control Flow* utilizadas

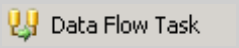
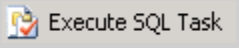
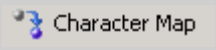
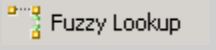
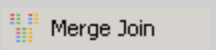
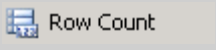
Tarefa	Descrição
	Permite movimentar dados entre fonte e destino, permitindo a transformação, limpeza e alteração de dados no processo. É a tarefa que possibilita a execução do processo de *ETL*.
	Permite executar *queries* de *SQL*. Utilizada para o carregamento da dimensão *DimUserEvent* e também para leitura e escrita de dados nas tabelas de suporte de carregamento periódico.

Tabela 18: Tarefas de *Control Flow* utilizadas

6.2.2 Transformações utilizadas

Para levar a cabo o carregamento da DWH, foram utilizados um conjunto de transformações disponíveis no *SSIS*, sendo estas descritas na presente subsecção.

Transformação	Descrição
	Aplica funções de tratamento de caracteres, como por exemplo conversões para minúsculas ou maiúsculas. Utilizado para realizar limpezas a *strings*, como por exemplo o nome dos utilizadores.
	Avalia um determinado conjunto de entrada através de várias condições, reencaminhando os dados para diferentes *outputs* conforme o resultado da expressão. Usado por exemplo para verificar os tipos de eventos.
	Cria colunas novas após aplicar expressões de entre um conjunto disponível às colunas de entrada. Utilizado para fazer conversões de dados, verificar a existência caracteres especiais em *strings*, e para fazer validação de tipos de eventos (por exemplo avaliar condições afim de verificar se um determinado evento é uma subida ou descida do nível de confidencialidade de uma determinada informação.
	Semelhante à transformação de *Lookup*, mas permite fazer comparações com caracteres especiais. Utilizado para encontrar referências das máquinas na tabela *DimHosts*.
	Permite efectuar pesquisas utilizando um *join* que pesquisa referências de um determinado valor num conjunto de dados (por exemplo uma tabela de BD). Utilizado para encontrar as referências dos valores da tabela de facto nas dimensões.
	Permite combinar dois conjuntos de dados de entrada distintos, tendo como resultado um conjunto único de dados. Utilizado para repetir os eventos tantas vezes quantos os números de perfis dos utilizadores, através da combinação da tabela de eventos com a tabela *Users_UserGroups* que contém os perfis de cada utilizador.
	Permite contar o número de linhas que passam no *data flow*. É utilizado para armazenar o número de linhas carregadas tendo em vista o carregamento periódico da DWH e a informação que existe nas BD de suporte de carregamentos.

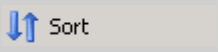
	Permite a utilização de código personalizado. É utilizado para criar um conjunto de dados de entrada temporário contendo valores de variáveis (como por exemplo o *ID* e *timestamp* lidos no carregamento) afim de escrever estes valores nas BD de suporte de carregamentos.
	Permite ordenar um conjunto de dados de modo ascendente ou descendente. Utilizado para ordenar os dois conjuntos de dados antes da transformação de *Merge Join* no processo de *ETL* da tabela *FactRoleActivity*.
	Permite combinar múltiplos *inputs* no mesmo fluxo. Utilizado após transformações de *Conditional Split* para voltar a ter somente um fluxo único.

Tabela 19: Transformações utilizadas

6.2.3 Carregamento das dimensões

Nesta sub-secção é descrito o carregamento das dimensões da DWH da Consola de Monitorização do RightsWATCH, incluindo imagens que descrevem o *Control Flow* e *Data Flow* de cada um destes carregamentos e as transformações aplicadas.

6.2.3.1 *DimInformation*

Como referido anteriormente, não existe uma tabela fonte contendo todos os documentos/informação encriptada com o RightsWATCH, pelo que se tornou necessário controlar quais os *GUID* já carregados para a dimensão, a fim de evitar repetições na dimensão de informação.

Para preencher os campos da dimensão é necessário pesquisar qual o utilizador que criou a informação e também de que tipo de documento se trata (email, *excel, powerpoint, word, image,…).*

Na Figura 22 é visível o *Control Flow* de carregamento da dimensão *DimInformation*, sendo que na tarefa *"DimInformation Populate"* está definido o *Data Flow* (Figura 23) onde é realizado o processo de *ETL*. Este carregamento necessita de tarefas adicionais no *Control Flow* devido à necessidade de controlar quais os documentos que já existem na dimensão *DimInformation.*

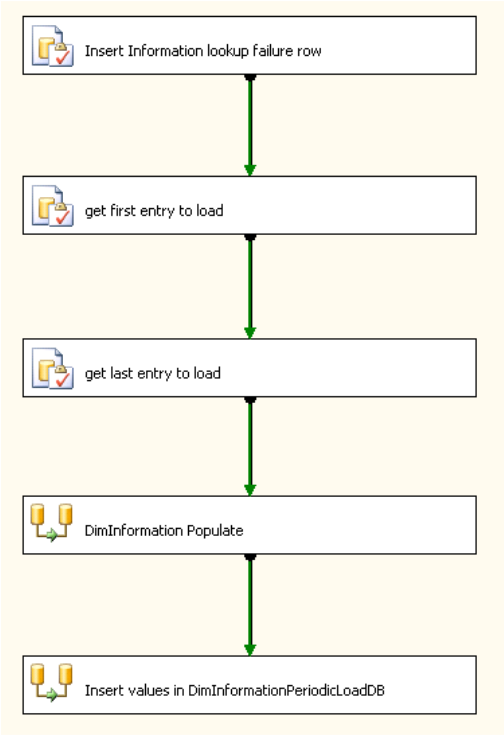


Figura 22: ***DimInformation Control Flow***

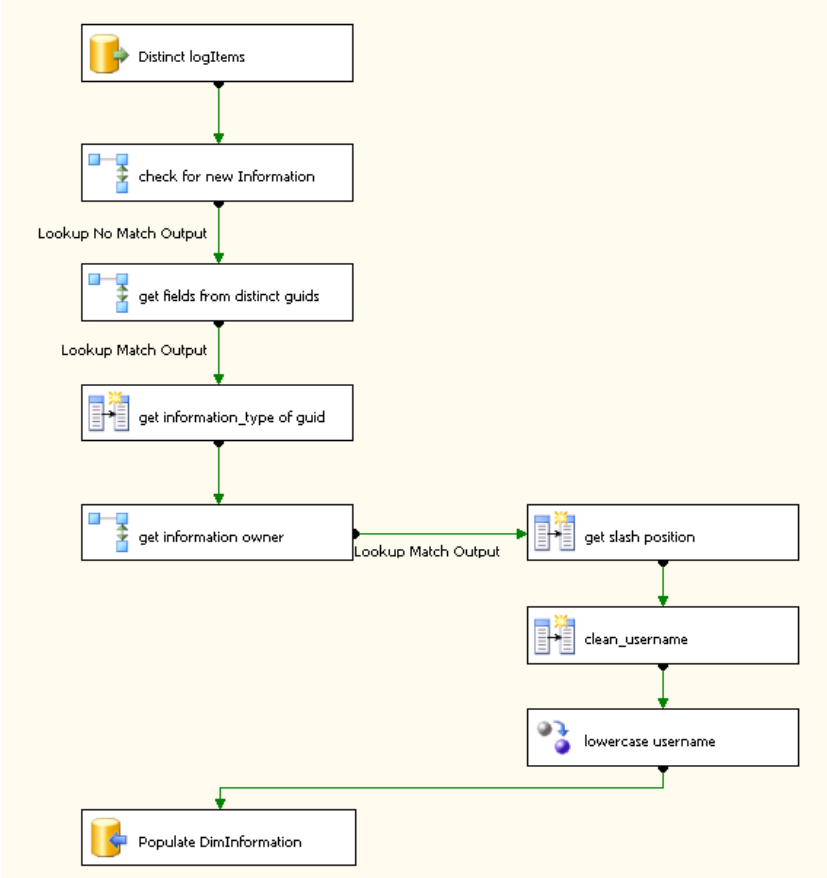


Figura 23: ***DimInformation Data Flow***

6.2.3.2 DimCompany, DimDepartment, DimMark, DimPlugin, DimRole, DimUser

As dimensões citadas têm uma tabela fonte, pelo que o seu esquema de carregamento é em tudo semelhante. Somente é necessário verificar se existem entradas novas na tabela fonte, e em caso afirmativo, carregar essas entradas para a dimensão.

A Figura 24 mostra o *Data Flow* para a dimensão *DimCompany,* sendo que exemplifica o que se passa nas outras dimensões mencionadas, visto o esquema de carregamento ser, como já referido, semelhante.

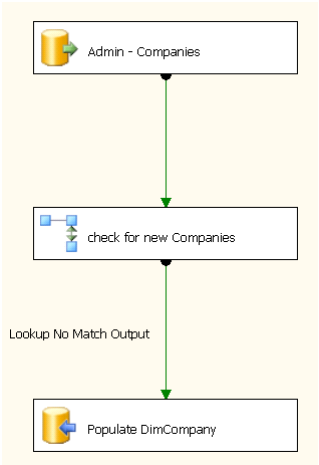


Figura 24: *DimCompany Data Flow*

6.2.3.3 DimHost

Como referido anteriormente, não existe uma tabela fonte contendo todos as máquinas que utilizam o RightsWATCH, pelo que se tornou necessário controlar quais as que já se encontravam na dimensão a cada carregamento, afim de evitar repetições na dimensão.

Na Figura 25 é visível o *Control Flow* de carregamento da dimensão *DimHost*, sendo que na tarefa *"DimLocalhost Populate"* está definido o *Data Flow* onde é realizado o processo de *ETL*. Este carregamento necessita de tarefas adicionais no *Control Flow* devido à necessidade de controlar quais as máquinas que já existem na dimensão *DimHost*.

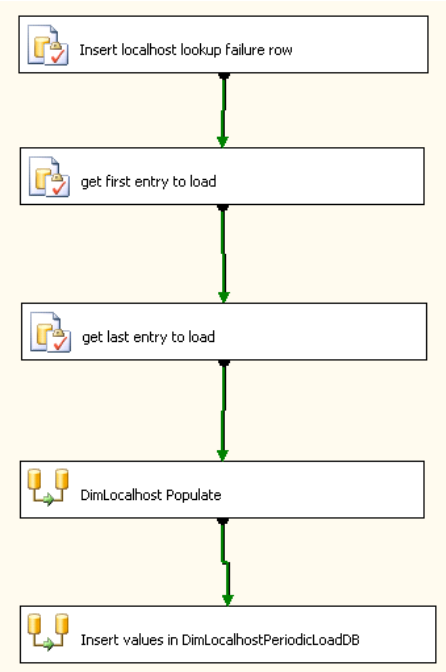


Figura 25: *DimHost Control Flow*

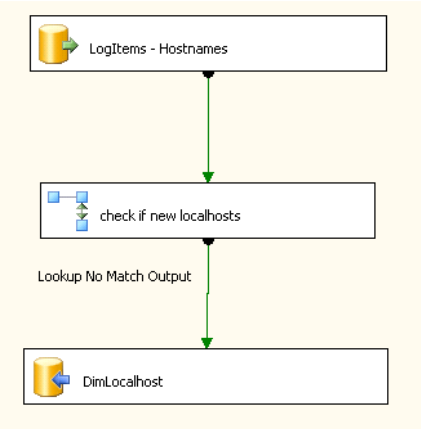


Figura 26: *DimHost Data Flow*

6.2.3.4 DimUserEvent

A tabela fonte de tipos de eventos do RightsWATCH está bastante generalista, pelo que se optou por especificar novos eventos para utilizar na DWH. Assim, estes eventos são inseridos na dimensão respectiva somente uma vez, recorrendo a uma *query SQL*.

6.2.3.5 *DimDate*

Tal como no exemplo acima, esta dimensão é carregada somente uma vez com todas as datas necessárias, sendo para tal executado uma *query T-SQL*.

6.2.4 Carregamento dos factos

Nesta sub-secção é descrito o carregamento das tabelas de facto da DWH da Consola de Monitorização do RightsWATCH, incluindo imagens que descrevem o *Control Flow* e *Data Flow* de cada um destes carregamentos e as transformações aplicadas.

Para melhor compreensão, será apresentado um diagrama alto nível do *Data Flow* de cada uma das tabelas de Facto, sendo explicado em detalhe as partes mais importantes do fluxo.

Encontra-se no anexo [5] Desenho detalhado do sistema, WSW-2013-SAS-00013-rightswatch-monitoring-architecture a explicação completa do processo de *ETL* para as Tabelas de Facto.

6.2.4.1 FactUserActivity

Para efectuar o carregamento da tabela de factos de eventos dos utilizadores, é necessário tratar a informação de cada linha da tabela *LogItems* afim de encontrar a referência para cada uma das dimensões da tabela de factos nos dados de entrada.

No fim de todas as chaves serem encontradas é necessário validar que todas os valores encontraram equivalência nas dimensões, e caso contrário inserir o valor "-1" na chave de dimensão, que iria apontar para uma entrada definida em todas as dimensões cujo valor é "Indefinido". Este procedimento é util para visualizar se há algum problema com o registo de eventos do RightsWATCH, possibilitando perceber quais os valores que não encontraram equivalência nas dimensões e que como tal estarão com problemas no registo dos dados na tabela *LogItems*.

Na Figura 27 encontra-se o *Control Flow* do processo de ETL para esta tabela de facto, sendo que na Figura 28 encontra-se o diagrama de alto nível do *Data Flow* (Tarefa *"Load UserActivity facts")*, sendo que as transformações marcadas com a cor azul serão explicadas em detalhe nas figuras seguintes, visto tratar-se de transformações mais complexas e do facto do diagrama no seu todo ser demasiado grande para ser inserido no presente relatório.

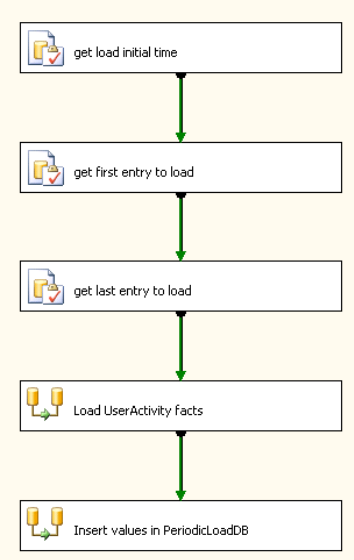


Figura 27: *FactUserActivity Control Flow*

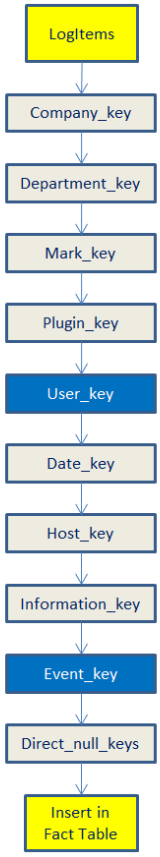


Figura 28: *FactUserActivity Data Flow*

Para o carregamento da chave *User_key* da dimensão *User* foi necessário efectuar algumas transformações, nomeadamente de limpeza de dados. Isto deve-se ao facto do RightsWATCH registar o nome do utilizador que fez a operação ao invés da sua chave na tabela de utilizadores, sendo que o nome do utilizador pode receber diferentes valores conforme o plugin utilizado. Por exemplo, um evento realizado por um utilizador "UtilizadorXPTO" do domínio "XYZ" poderia ter como registo no campo utilizador um dos seguintes valores:

- XYZ\UtilizadorXPTO
- UtilizadorXPTO@XYZ.COM
- UtilizadorXPTO

O valor que irá fazer o *matching* na tabela de dimensão é somente "UtilizadorXPTO", pelo que se torna necessário efectuar uma limpeza de caracteres não necessários. Este procedimento é mostrado na figura seguinte, onde se encontra a transformação utilizada pra tal.

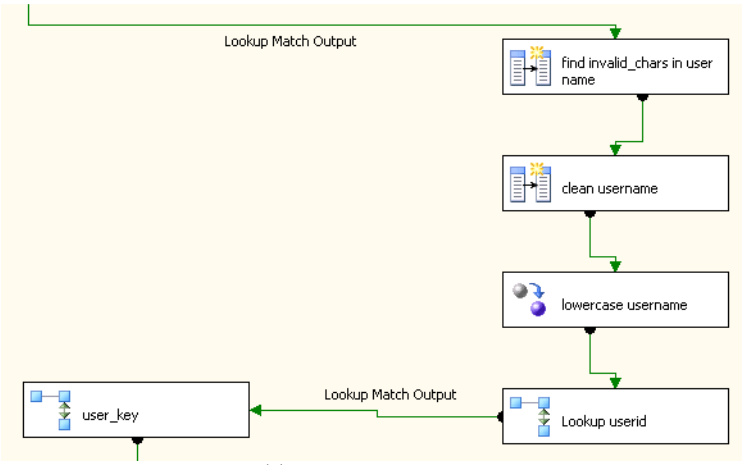


Figura 29: Transformações para *User_Key*

Para o carregamento da chave de *User_event_key* da dimensão *User_event* foi necessário efectuar transformações para converter a acção no tipo de evento respectivo, tal como demonstra a Figura 30.

Estas transformações prendem-se maioritariamente com a necessidade de classificar um tipo de evento "Marcação de documento" na tabela fonte com os vários tipos de Eventos relacionados com a classificação: "Subida do nível de confidencialidade", "Nível de confidencialidade mantido" ou "Descida do nível de confidencialidade.". Outro aspecto tido em conta foi o facto de um evento poder ser realizado com sucesso ou não, sendo esse facto explícito nos eventos existentes na DWH e não nos eventos definidos na tabela fonte, pelo que se torna também em algo necessário de validar.

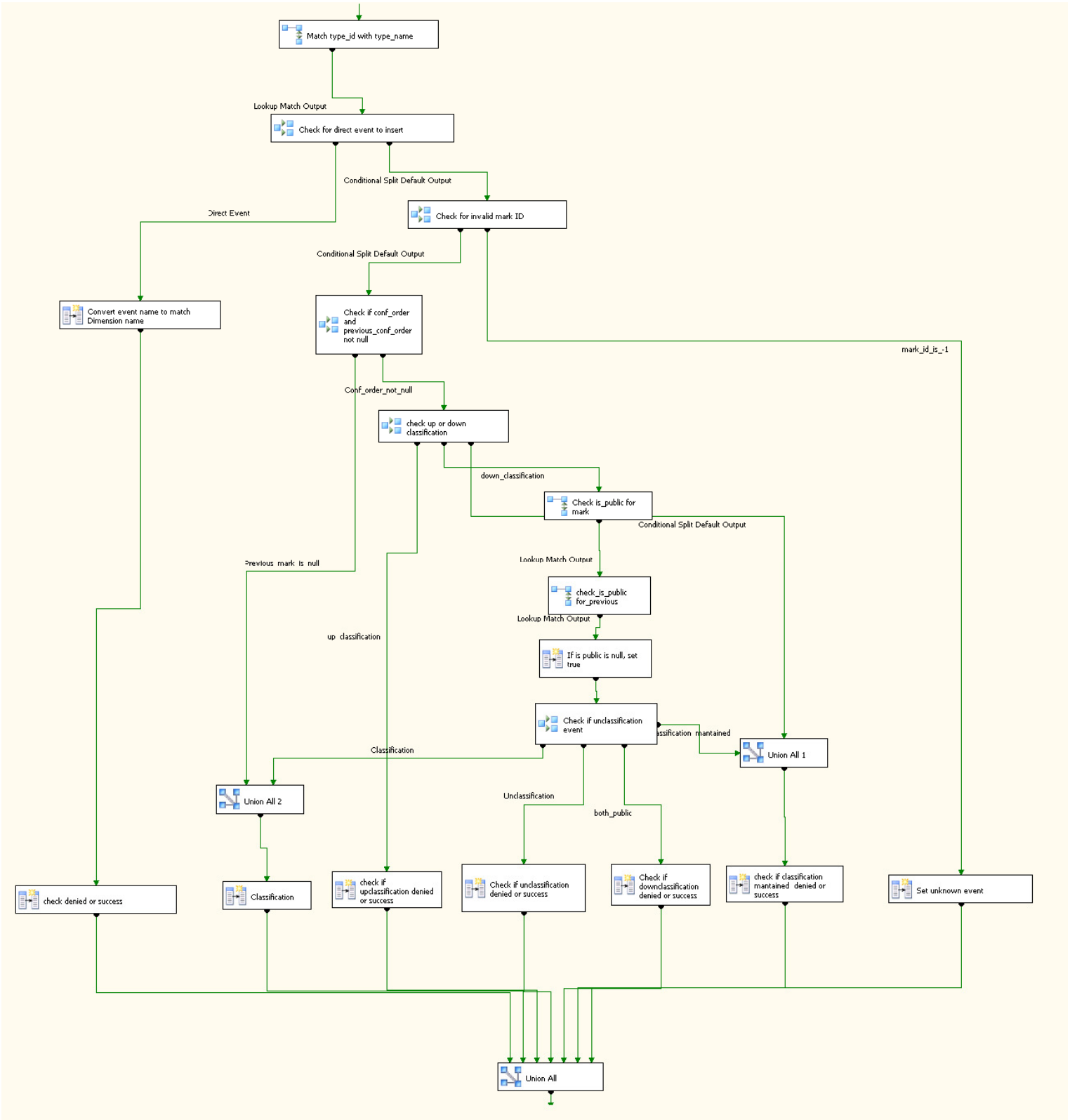


Figura 30: Transformações para *User_event_key*

6.2.4.2 FactRoleActivity

O *Control Flow* e as transformações aplicadas *(Data Flow)* no carregamento da tabela de factos *FactRoleActivity* são em tudo semelhante às explicadas para a tabela *FactUserActivity*. Há uma excepção, relacionada com o *matching* para a *Role_Key* no *Data Flow*, identificado com a cor vermelha no diagrama de *Data Flow* de carregamento da tabela *FactRoleActivity* representado na figura seguinte.

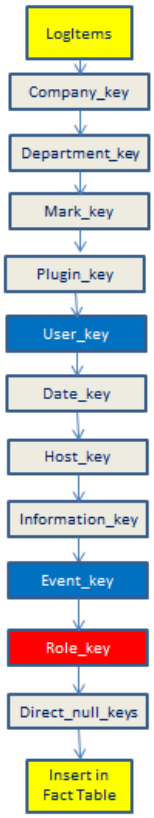


Figura 31: *FactRoleActivity Data Flow*

Para efectuar o *matching* da *Role_key* com o *User* respectivo e replicar o evento tantas vezes quantas o número de roles do *User* (um evento por cada um dos roles) foi necessário utilizar a transformação *Merge Join*, explicada na Tabela 19.

A Figura 32 mostra as transformações aplicadas relativas ao *matching* da *Role_key*.

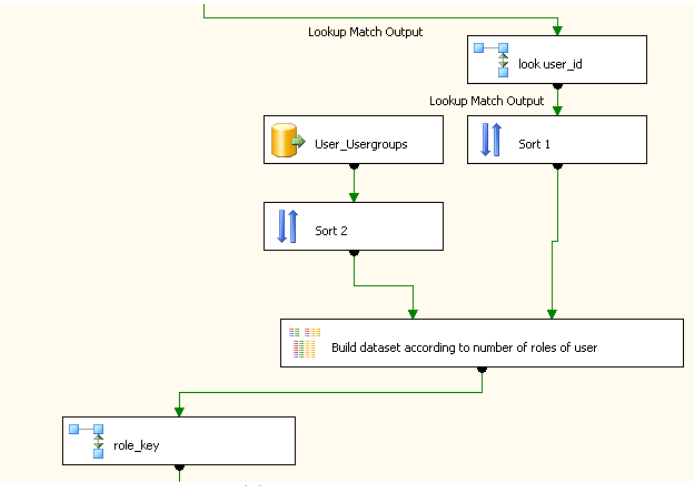


Figura 32: Transformações para *Role_key*

6.2.4.3 FactInformation

Para a tabela *FactInformation*, o esquema do *Control Flow* é semelhante aos anteriores, havendo naturais diferenças no *Data Flow* onde são aplicadas as necessárias transformações aos dados de entrada.

Podemos verificar na Figura 33 que são recolhidas sucessivamente as chaves para as referências dos dados na dimensão *DimInformation*, havendo depois uma transformação responsável por verificar se cada informação encriptada já existe na tabela de factos. Assim, somente se inserem novas informações, pois se se inserisse a mesma informação duas vezes nesta tabela, não se iria ter uma conta exacta de quantos documentos foram efectivamente encriptados recorrendo ao RightsWATCH, ou seja, somente são inseridos documentos com *GUID* diferente.

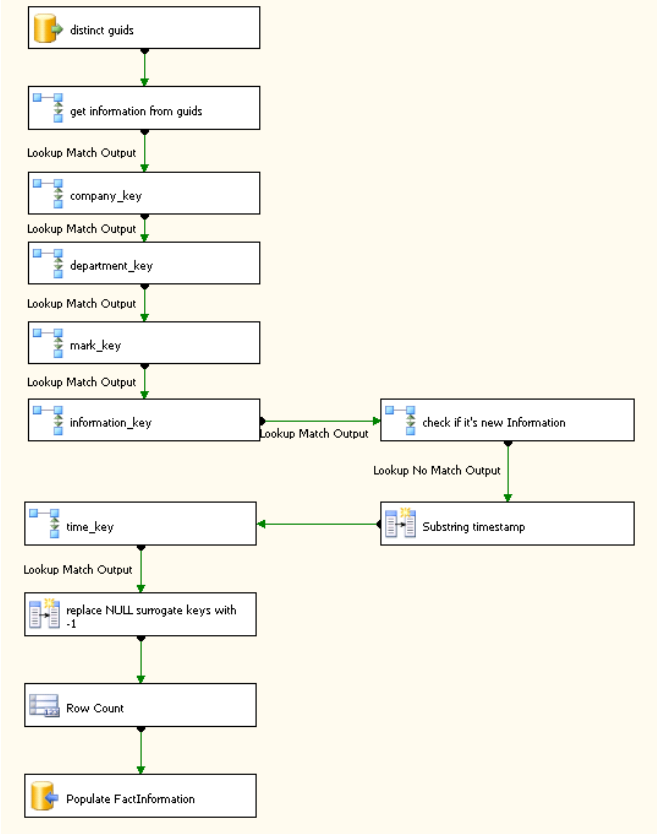


Figura 33: *FactInformation Data Flow*

6.3 Periodicidade do processo ETL

Após a conclusão do desenvolvimento da DWH e dos respectivos automatismos para proceder ao carregamento periódico da mesma, decidiu-se colocar a mesma no ambiente de produção da Critical Software, cuja BD inclui eventos realizados desde 2009.

Tendo em conta a integração num ambiente real, foi necessário chegar a um valor de intervalo do carregamento periódico da DWH. Assim, foram realizados testes com diferentes períodos de carregamento (em dias de semana), e verificados alguns valores, tanto para o carregamento de dados para a área de estágio como para o tempo total de carregamento.

Importa salientar as configurações da máquina responsável pelo processo de ETL, visto estas terem um impacto decisivo na performance do ETL:

- **Processador:** Intel Xeon CPU E7-L8867 @ 2.13GHz
- **Memória RAM:** 2GB

Um aumento no valor da memória RAM reflectir-se-ia num aumento da performance, sendo que este facto pode levar a que os tempos medidos na Critical Software não se verifiquem nas empresas que utilizam o RightsWATCH, pois os valores dependerão da capacidade da máquina onde o processo ETL será realizado.

O resultado do estudo realizado com os carregamentos na Critical Software encontra-se na tabela seguinte:

Carregamento *StagingArea*				
Período de intervalo	Número de carregamentos em 24 horas	Número total de linhas carregado	Média de linhas por carregamento	Tempo médio de carregamento
30 minutos	34	2245	66	3.5 segundos
1 hora	18	2584	144	3.5 segundos
2 horas	9	2243	249	4.54 segundos
3 horas	6	2136	356	4.31 segundos

Tabela 20: Métricas de carregamento de informação para a área de estágio

Carregamento total (*StagingArea + Dimensões + Factos)*	
Período de intervalo	Tempo médio de carregamento total
30 minutos	2 minutos e 41 segundos
1 hora	2 minutos e 55 segundos
2 horas	2 minutos e 31 segundos
3 horas	2 minutos e 43 segundos

Tabela 21: Tempo de carregamento total da DWH para diferentes períodos de carregamento

Percebeu-se após medição dos tempos apresentados, que não valeria a pena o intervalo de carregamento ser mais estendido do que o período de 4 horas, visto que para os intervalos estudados os tempos de carregamento totais se relevaram muito semelhantes.

A escolha para a utilização na Critical Software recaiu sobre o período de 2 em 2 horas, após balanceamento entre o período de carregamento da DWH (poucas diferenças apresentou como referido em cima) e também o facto de, após cada um dos carregamentos periódicos, existir uma quebra de performance momentânea no *Dashboard* durante o período de processamento do cubo *OLAP*, que é feito em cerca de 1 minuto. Durante este período, ao tentar arrastar uma métrica para o ecrã, o tempo de resposta é compreensivelmente mais lento do que no normal funcionamento, mas nunca mais do que 30 segundos a 1 minuto (tempo máximo de processamento do cubo).

6.4 Fluxo do processo de *ETL*

Tendo em conta restrições ao nível de performance que impedem a DWH de ser actualizada em tempo real, a extracção, transformação e carregamento dos dados para a DWH têm que ser feitos periodicamente. Com vista à normal performance do sistema, o processo ETL é realizado periodicamente, num valor que tem que ser ajustado para a realidade de cada empresa que utilize o RightsWATCH, pois dependerá da quantidade de utilizadores bem como das necessidades da própria empresa.

Este carregamento é levado a cabo por uma tarefa criada no SQL Server Management Studio através da definição de um Job no SQL Server Agent, sendo aqui definido o tempo de repetição desta tarefa. No caso de estudo este estágio (Critical Software) o processo de ETL foi realizado de 2 em 2 horas no período entre as 08h e as 24h, como demonstra o diagrama seguinte.

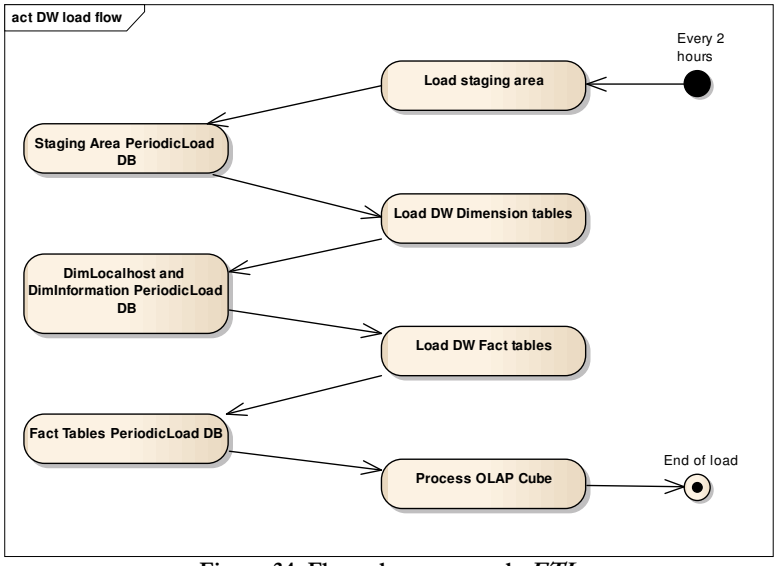


Figura 34: Fluxo do processo de *ETL*

Um carregamento completo é composto por etapas sequenciais, sendo a ordem a seguinte:

- Carregamento da área de estágio;
- Carregamento das dimensões;
- Carregamento das tabelas de factos;
- Processamento do cubo OLAP;

No intermédio destas etapas existem actualizações das tabelas de suporte de carregamento periódico dos Factos e Dimensões.

Estes passos são descritos nas subsecções seguintes.

6.4.1 Área de estágio

Antes do processo de *ETL*, toda a informação nova existente desde o último carregamento é copiada para uma área de estágio. Este procedimento permite que a tabela fonte seja liberta rapidamente, aplicando posteriormente todas as transformações sobre a informação existente na área de estágio. Após cada um destes carregamentos é registada informação sobre o carregamento numa tabela de suporte (*StagingAreaPeriodicLoadDB)* afim de controlar os dados já carregados, evitando assim o carregamento de informação repetida.

A Tabela 22 descreve a BD utilizada para controlar os carregamentos periódicos efectuados da BD de produção para a área de estágio.

StagingAreaPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Rows_loaded	Integer	Quantidade de linhas carregadas num carregamento.
Begin_of_load_time	Datetime	Data e hora do início do carregamento.
End_of_load_time	Datetime	Data e hora do fim do carregamento.
Last_entry_loaded_from_view	Integer	O *log_id* do ultimo evento carregado.
Last_entry_loaded_timestamp	Text	*Timestamp* do ultimo evento carregado.

Tabela 22: Descrição da BD *StagingAreaPeriodicLoadDB*

Cada novo carregamento da BD de produção do RightsWATCH para a área de estágio é iniciado utilizando o conjunto *Last_entry_loaded_from_view* juntamente com o *Last_entry_loaded_timestamp* para identificar a posição a partir da qual a informação ainda não foi carregada para ser processada para a DWH.

6.4.2 Carregamento periódico das dimensões

As dimensões são também elas carregadas periodicamente, sendo este processo feito tendo como fonte de dados as tabelas de configuração do *RightsWATCH*, excepto para duas das Dimensões:

- **DimHost:** Não existe nenhuma tabela com os *hosts* existentes em cada empresa que utilize o RightsWATCH, pelo que é necessário verificar a cada carregamento periódico quais os novos *hosts* existentes na tabela fonte *LogItems* relativamente aos existentes na tabela *DimLocalhost*, carregando somente estes novos *hosts* encontrados. É registado numa tabela de suporte (*DimLocalhostPeriodicLoadDB)* qual a última entrada e *timestamp* onde foi verificado a existência de novos *hosts*, utilizando esta informação para pesquisa somente em novas entradas na tabela *LogItems*.
- **DimInformation:** Tal como na situação anterior, também não existe nenhuma fonte de dados com a Informação que é encriptada com o *RightsWATCH*. Como tal, é necessário seguir o mesmo processo descrito em cima: verificar a cada carregamento periódico quais os novos *GUIDS* existentes na tabela fonte *LogItems* relativamente aos existentes na tabela *DimInformation*, carregando somente informação relativa a estes novos *GUIDS* encontrados. É registado numa tabela de suporte (*DimInformationPeriodicLoadDB)* qual a última entrada e *timestamp* onde foi verificado a existência de novos *GUIDS*, utilizando esta informação para pesquisa somente em novas entradas na tabela *LogItems*.

A Tabela 23 e Tabela 24 descrevem as duas BD acima mencionadas, sendo que também neste caso se utiliza o conjunto *Last_entry_loaded_from_view* em conjunto com o *Last_entry_loaded_timestamp* para identificar a posição a partir da qual a informação ainda não foi carregada para ser processada para as dimensões da DWH.

DimHostPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Date_of_load	Datetime	Data e hora do início do carregamento.
Last_entry_loaded_timestamp	Text	*Timestamp* do ultimo evento carregado.
Last_logitems_entry_loaded	Integer	O *log_id* do ultimo evento carregado.

Tabela 23: Descrição da BD *DimHostPeriodicLoadDB*

DimInformationPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Date_of_load	Datetime	Data e hora do início do carregamento.
Last_entry_loaded_timestamp	Text	Timestamp do ultimo evento carregado.
Last_logitems_entry_loaded	Integer	O log_id do ultimo evento carregado.

Tabela 24: Descrição da BD DimInformationPeriodicLoadDB

6.4.3 Carregamento periódico das tabelas de Factos

As tabelas de Factos são carregadas após os dados que se encontram na área de estágio sofrerem as transformações necessárias.

Para que não haja factos repetidos a serem carregados, são utilizadas três tabelas na base de dados de suporte: *FactUserActivityPeriodicLoadDB*, *FactRoleActivityPeriodicLoadDB* e *FactInformationPeriodicLoadDB,* que são descritas respectivamente na Tabela 25, Tabela 26 e Tabela 7.

FactUserActivityPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Rows_loaded	Integer	Quantidade de linhas carregadas num carregamento.
Begin_of_load_time	Datetime	Data e hora do início do carregamento.
End_of_load_time	Datetime	Data e hora do fim do carregamento.
Last_entry_loaded_from_view	Integer	O *log_id* do ultimo evento carregado.
Last_entry_loaded_timestamp	Text	*Timestamp* do ultimo evento carregado.

Tabela 25: Descrição da BD *FactUserActivityPeriodicLoadDB*

FactRoleActivityPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Rows_loaded	Integer	Quantidade de linhas carregadas num carregamento.
Begin_of_load_time	Datetime	Data e hora do início do carregamento.
End_of_load_time	Datetime	Data e hora do fim do carregamento.
Last_entry_loaded_from_view	Integer	O *log_id* do ultimo evento carregado.
Last_entry_loaded_timestamp	Text	*Timestamp* do ultimo evento carregado.

Tabela 26: Descrição da BD *FactRoleActivityPeriodicLoadDB*

FactInformationPeriodicLoadDB		
Atributo	**Tipo de dados**	**Descrição**
Load_id	Integer	Identificador do carregamento.
Rows_loaded	Integer	Quantidade de linhas carregadas num carregamento.
Begin_of_load_time	Datetime	Data e hora do início do carregamento.
End_of_load_time	Datetime	Data e hora do fim do carregamento.
Last_entry_loaded_from_view	Integer	O *log_id* do ultimo evento carregado.
Last_entry_loaded_timestamp	Text	*Timestamp* do ultimo evento carregado.

Tabela 27: Descrição da BD *FactInformationPeriodicLoadDB*

6.4.4 Processamento do cubo *OLAP*

Para possibilitar a análise *OLAP* dos dados, a *Microsoft* tem o já mencionado *Microsoft SSAS*, que permite a criação de um cubo multidimensional, que agrega e indexa a informação de modo a que a consulta dos dados tenha uma performance elevada, permitindo o cruzamento de todas as dimensões na procura da informação desejada.

A informação que é inserida na DWH não é directamente utilizada pela Consola de Monitorização. Ao invés, ela é carregada para o Cubo *OLAP* – que é na verdade uma estrutura física - sendo sobre esta informação que as consultas são feitas.

Este cubo utiliza o modo *Multidimensional OLAP* (*MOLAP*), que se traduz nas seguintes vantagens:

- A informação é comprimida, pelo que não há uma alocação elevada de espaço;
- A informação é indexada para maximizar a performance;
- Evita a necessidade de conexão à DWH pela aplicação, facto que iria influenciar negativamente os carregamentos periódicos. Assim, as *queries* são feitas contra o cubo, que possui a informação necessária sem necessidade de ligar à DWH, o que permite que mesmo estando a acontecer inserção de dados na DWH a performance das *queries* não seja afectada, dado serem estruturas independentes neste aspecto.
- De entre os modos disponíveis *(Relational OLAP, Multidimensional OLAP, Hybrid OLAP)*, *MOLAP* é o mais comummente utilizado e o que apresenta melhor performance.

É necessário efectuar a criação do cubo, sendo que este processo é explicado em detalhe no anexo [5] Desenho detalhado do sistema, e resumido de seguida.

Os passos a seguir para a criação do cubo *OLAP* são os seguintes:

1. Definir fonte de dados (DWH);
2. Criar uma vista sobre a DWH;
3. Criar o cubo, especificando as tabelas de dimensão e de factos da DWH;
4. Especificar os atributos definidos nas dimensões que se pretendem utilizar para a criação de métricas e efectuar análise *OLAP*;
5. Criar uma hierarquia temporal, que consiste em relacionar os atributos da dimensão *DimDate* de modo sequencial do menor período de tempo para o maior;
6. Acrescentar a propriedade de *Time Intelligence*, processo este que consiste em definir quais dos atributos da dimensão *DimDate* que dizem respeito a que período temporal (por exemplo definir qual dos atributos diz respeito às horas, às semanas, ao dia, …) para permitir a utilização de funções temporais nas *queries* MDX, que são usadas por exemplo na filtragem das métricas por período temporal;

Após a conclusão deste processo, é possível utilizar o *Microsoft SSAS* para análise de dados, ou configurar uma conexão no *Microsoft Excel* para o cubo definido, permitindo assim fazer a mesma análise *OLAP* utilizando uma ferramenta de produtividade bastante comum como é o caso do *MS Excel*. Finalmente, após a configuração do cubo passa a ser possível a conexão da Consola de Monitorização ao cubo, através da qual são retiradas as métricas, recorrendo às *queries MDX*.

Capítulo 7 - Implementação

O presente capítulo descreve o processo de implementação seguido pelo estagiário, apresentando também algumas métricas relativas ao código da aplicação desenvolvida e o resultado final do produto.

7.1 Resumo das *Sprints*

A implementação decorreu em *sprints* da metodologia ágil *SCRUM*.

No anexo [6] Backlog Consola de Monitorização RightsWATCH, 2013-RPT-00018-monitoring-console-backlog encontram-se detalhadas as *sprints* realizadas pelo estagiário, sendo nesta secção apresentado um resumo das mesmas.

Os requisitos identificados no levantamento de requisitos foram traduzidos em *User Stories*, sendo uma *User Story*, uma frase que traduz em linguagem "tradicional" as necessidades de um determinado utilizador do sistema a ser desenvolvido. Usualmente seguem o formato:

As an <user> I want to <goal/desire> so that I <benefit to take>;

Importa também esclarecer alguns conceitos subjacentes à definição de *User Story:*

- ***Epic:*** Representa um conjunto de *Themes*, pelo que pode ser considerado um conjunto de várias *User Stories*.
- ***Theme:*** Representa também um conjunto de *User Stories*, mas a um nível menor, podendo ser considerado uma *User Story* grande.
- ***Story Points:*** Representam uma escala de complexidade de implementação de uma determinada *User Story*. Usualmente, está ligado ao tempo que levará a completar uma *User Story*. Na *Watchful Software* é utilizada uma sequência de *Fibonacci* para estimar as *User Stories*, sendo portanto a escala utilizada 1-3-5-8-13. Esta sequência foi escolhida para evitar debates na atribuição de *story points* com pequena diferença entre si (por exemplo debater entre 8 ou 9 *story points*). Foi escolhida uma *User Story*, como base, e todas as outras são estimadas em comparação com essa *User Story*. Usualmente é escolhida uma *User Story* de dificuldade mínima para servir de comparação. Se se estimar uma *User Story* em mais do que 13 *Story Points*, então deverá ser partida em duas ou mais *User Stories*.

A Tabela 28 contém as *User Stories* implementadas durante o estágio, bem como a sua estimativa em *Story Points*. A *User Story* de referência encontra-se marcada a ***negrito e itálico.***

RightsWATCH Monitoring Console Backlog			
Epic	**Theme**	**User Story**	**Story Points**
	Home Section	As an Auditor I want to view a list of all the Monitoring console sections so that I can choose the section to open.	2
		As an Auditor I want to open a Monitoring console section from the listed sections of the	1

		Home Section so that I can access all the monitoring sections.	
Dashboard	Data Warehouse	As an Auditor I want to be able to access and analyze the information related with RightsWATCH usage so that I can have useful information related to the system.	13
		As an Auditor I want to be able to view useful information in the RightsWATCH monitoring console so that I can view the state of the system at each moment.	5
		As an Auditor I want to retrieve useful information from the events produced by the RightsWATCH users so that I can monitor the system usage.	8
		As an Auditor I want the system to periodically load RightsWATCH usage data so that the information is up to date.	5
	Visual Interface	As an Auditor I want to view the default metrics listed in the metrics toolbox so that I can know all the default metrics that exist in the section.	5
		As an Auditor I want to drag one metric to the screen so that the correspondent bars graphic with the metric result appears.	8
		As an Auditor I want to drag one metric to the screen so that the correspondent pie graphic with the metric result appears.	8
		As an Auditor I want to drag all the metrics that exist to the screen so that the correspondent graphics with the result appear.	5
		As an Auditor I Want to change the Time Window of the Dashboard to default values (last day, week, month, year) so that the metric graphic reflects this change.	5
Information Tracking	Search Information	As an Auditor I want to search specific Information so that I can view the respective events list ordered by date.	3
	Visual Interface	As an Auditor I want to scroll the graphic horizontally so that I can move ahead and backward in the time.	5
		As an Auditor I want to view the ID, type and name of the Information that I'm currently tracking so that I can know which Information is being tracked.	2
		As an Auditor I want o to change between list and graphic visualization so that I can track the information lifecycle in two different manners.	3
		As an Auditor I want to view the lifecycle of specific	13

		information in a graphic manner so that I can track the information lifecycle and respective events.	

Tabela 28: Conjunto de *User Stories* implementadas no estágio

No anexo [6] Backlog Consola de Monitorização RightsWATCH encontra-se a descrição completa e detalhada das *Sprints* de implementação realizadas.

A Tabela 29 resume as 4 *Sprints* realizadas na fase de implementação, de acordo com as seguintes métricas:

- **Velocity:** Soma dos *Story Points* conseguidos na *Sprint*;
- **Achievement:** Percentagem dos *Story Points* implementados na *Sprint*, tendo em conta as estimativas feitas no *Planning*.

Assim, é possível perceber que para a *Sprint#1*, os 18 *Story Points* conseguidos dizem respeito a 95% do total dos *Story Points* previstos de implementar.

Sprint	Velocity	Achievement
#1	18	95%
#2	13	100%
#3	31	94%
#4	29	100%

Tabela 29: Resumo das *Sprints* de implementação

7.2 Métricas de código desenvolvido

A tabela seguinte apresenta o número de linhas de código desenvolvidas.

Não servindo este valor como medidor de complexidade ou qualidade do código, serve como nota do trabalho realizado em termos de código, salientando-se que grande parte do trabalho do estágio foi desenvolvido em termos de *Data Warehousing* e não sendo reflectido nestes valores.

O código relativo aos *common-components* foi implementado em *C#*, sendo que o código *Web* diz respeito a implementação em *Javascript* e *C#*.

	Nº de classes implementadas	Nº de linhas de código
Common-components	6	366
Monitoring Console (Web)	-	3005

Tabela 30: Métricas de código

7.3 Imagens das secções implementadas

Na Figura 35 encontra-se uma imagem do *Dashboard* implementado no decorrer do estágio, enquanto na Figura 36 apresenta-se a secção de *Information Tracking*, também implementada no decorrer do estágio.

É possível observar na secção de *Dashboard* algumas das funcionalidades implementadas, como a possibilidade de escolher as métricas a mostrar, o tipo de gráfico de cada métrica, ou o período temporal das métricas.

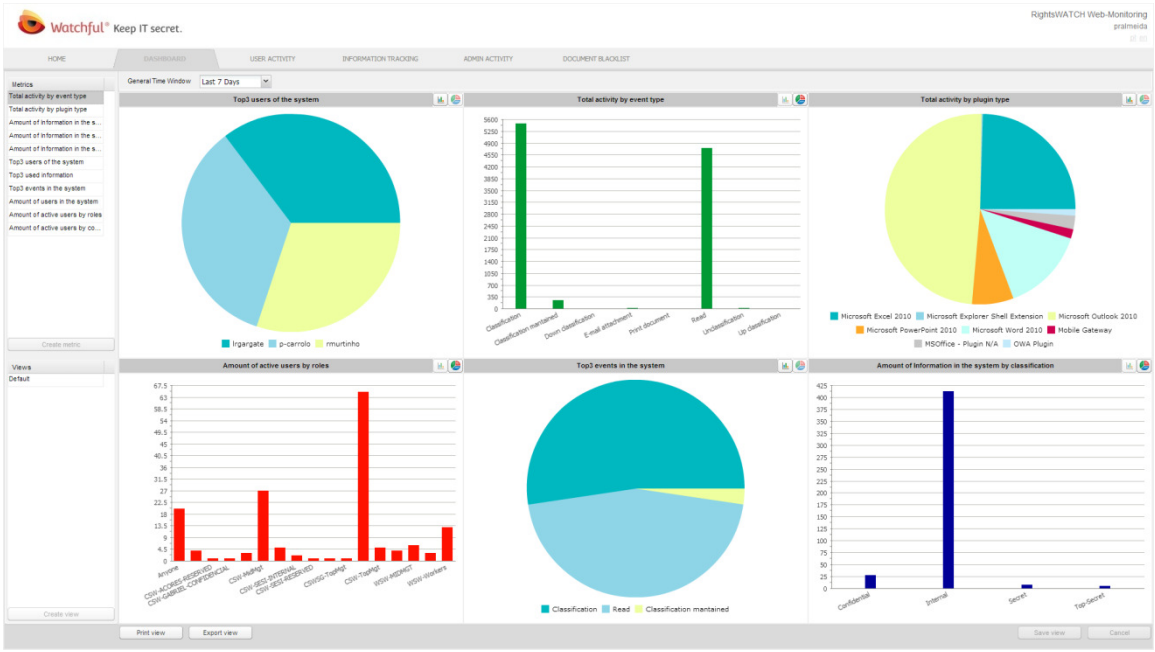


Figura 35: ***Dashboard RightsWATCH Monitoring Console***

Na secção de *Information Tracking* destaca-se a possibilidade de alterar o modo de visualização (diagrama/gráfico ou listagem de eventos), sendo que na imagem é possível verificar um diagrama de monitorização de um documento com a respectiva régua temporal na parte inferior do ecrã que permite ao Auditor ter uma noção rápida sobre a data em que cada evento foi realizado.

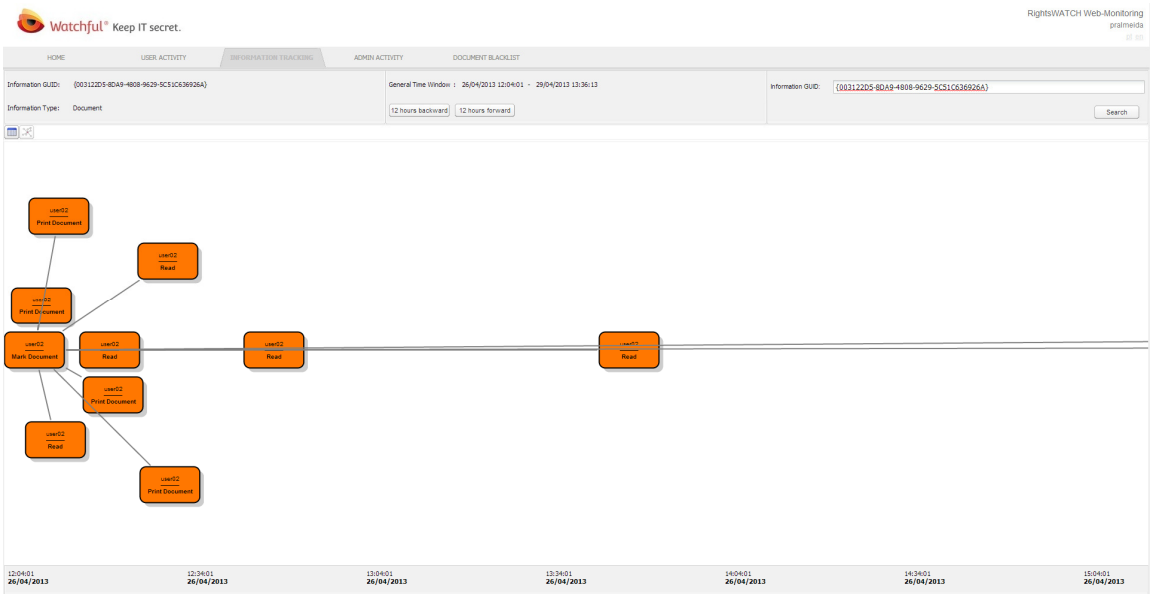


Figura 36*: Information Tracking RightsWATCH Monitoring Console*

Capítulo 8 - Testes

Para validação dos requisitos implementados, o estagiário desenhou um conjunto de testes de aceitação, que se encontram detalhados no anexo [7] Especificação de Testes Consola de Monitorização RightsWATCH.

A Tabela 31 detalha a ligação entre os requisitos e os casos de teste usados para validação sobre a forma de uma matriz de rastreabilidade, apresentando também o resultado da execução dos testes.

		Requisitos																							
		1.1.1 - List console sections	1.1.2 - Choose console sections	2.1.01 - Show metrics toolbox	2.1.03 - Add metric to the screen	2.1.04 - Remove metric from the screen	2.1.13 - Configure general time window for dashboard	2.1.15 - Edit graphic	2.2.01 - Total Activity	2.2.02 - Plugins Activity metric	2.2.03 - Amount of information in the system	2.2.04 - Amount of information in the system by classification	2.2.05 - Amount of information in the system by scope	2.2.06 - Top users of the system	2.2.07 - Top used information	2.2.09 - Top events	2.2.11 - Amount of users in the system	2.2.12 - Amount of users in the system by scope	4.1.01 - Search for specific information	4.1.02 - Show information ID	4.1.03 - Show information plugin	4.3.1 - Time oriented graphic	PASSOU
Testes	RSW-TCS-CSW-001 – Sections listings	X																						✓
	RSW-TCS-CSW-002 – Open Dashboard section from Home button		X																					✓
	RSW-TCS-CSW-003 – Open Dashboard section from tab		X																					✓
	RSW-TCS-CSW-004 – Open Information Tracking from Home button		X																					✓
	RSW-TCS-CSW-005 – Open Information Tracking from tab		X																					✓
	RSW-TCS-CSW-006 – List metrics in toolbox			X																				✓
	RSW-TCS-CSW-007 – Draggable metrics				X																			✓
	RSW-TCS-CSW-008 – Drop metric in invalid zone				X																			✓
	RSW-TCS-CSW-009 – Drop metric in valid zone				X																			✓
	RSW-TCS-CSW-010 – Change metric graphic from bar to pie							X																✓
	RSW-TCS-CSW-010 – Change metric graphic from pie to bar							X																✓
	RSW-TCS-CSW-012 – Remove metric from screen					X																		✓
	RSW-TCS-CSW-013 – Last day as general time window						X																	✓
	RSW-TCS-CSW-014 – Last 7 days as general time window						X																	✓
	RSW-TCS-CSW-015 – Last 30 days as general time window						X																	✓
	RSW-TCS-CSW-016 – Last 365 days as general time window						X																	✓
	RSW-TCS-CSW-017 – Events by type metric								X															✓
	RSW-TCS-CSW-018 – Events by plugin metric									X														✓
	RSW-TCS-CSW-019 – Amount of Information metric										X													✓
	RSW-TCS-CSW-020 – Amount of information by level metric											X												✓
	RSW-TCS-CSW-021 – Amount of information by department metric												X											✓
	RSW-TCS-CSW-022 – Top3 users with more events metric													X										✓
	RSW-TCS-CSW-023 – Top3 Information with more associated events metric														X									✓
	RSW-TCS-CSW-024 – Top3 event types performed metric															X								✓
	RSW-TCS-CSW-025 – Amount of active users in the system metric																X							✓
	RSW-TCS-CSW-026 – Amount of active users by department metric																	X						✓
	RSW-TCS-CSW-027 – Add metric to drop zone with metric				X																			✓
	RSW-TCS-CSW-028 – Information Tracking by GUID in grid mode																			X				✓
	RSW-TCS-CSW-028 – Information Tracking by GUID in graphic mode																			X				✓
	RSW-TCS-CSW-030 – View Information GUID being searched																				X			✓
	RSW-TCS-CSW-031 – View file type being searched																					X		✓
	RSW-TCS-CSW-032 – Flow diagram ordered by time																						X	✓

Tabela 31: Matriz de rastreabilidade

Capítulo 9 - Conclusões

Após o término do estágio, é possível fazer um balanço do trabalho realizado e nos passos seguintes a realizar no âmbito da monitorização do RightsWATCH.

9.1 Balanço do estágio

Os objectivos do estágio, apresentados na subsecção 1.2 (Objectivos) foram cumpridos, tendo por base a DWH projectada e correctamente implementada. Assim, as funcionalidades implementadas no estágio tornam possível ao Auditor:

- Perceber o estado geral do sistema através de métricas definidas;
- Monitorizar o ciclo de vida da informação encriptada com o RightsWATCH;
- Monitorizar as actividades dos utilizadores, tanto através da consola desenvolvida como de ferramentas *OLAP (SQL Server Analysis Services, MS Excel);*

É de salientar neste ponto a oportunidade dada ao estagiário, da parte do Prof. Bruno Cabral, de apresentar o trabalho realizado no estágio na aula final da disciplina de Inteligência no Negócio do Mestrado em Engenharia Informática do DEI-FCTUC, factor que atesta o sucesso do mesmo.

À data de escrita do relatório, o *Dashboard* do RightsWATCH encontra-se em pleno funcionamento, com toda a arquitectura de suporte (DWH) a funcionar em pleno, sendo esta uma das partes mais desafiantes e importantes do estágio que agora termina.

Relativamente ao estágio, existem pontos a salientar:

- O estágio proposto foi desde o dia 1 encarado pelo estagiário como uma oportunidade única, pela possibilidade dada de criar uma aplicação desde o seu início, passando por todas as fases do desenvolvimento de software, bem como pela liberdade dada ao estagiário em todas as decisões necessárias;
- O estágio estava relacionado com uma área da qual o estagiário não tinha conhecimento prévio, pelo que foram desenvolvidas competências muito importantes de auto-aprendizagem e capacidade de iniciativa;
- O contacto com a realidade empresarial e o respectivo acompanhamento, tanto da parte do DEI como como da parte da Watchful Software, foram factores fundamentais no sucesso do estágio e do trabalho realizado;
- O estagiário desenvolveu competências de trabalho em equipa, de trabalho metódico e de aprendizagem de processos de desenvolvimento de Software, nomeadamento dos processos seguidos pela Critical Software, uma empresa reconhecida pela qualidade dos mesmos, sendo uma empresa CMMI 5.

9.2 Trabalho futuro

Neste momento está montada a base para as capacidades de *Dashboard* do RightsWATCH crescerem, nomeadamente no que toca à possibilidade de criação de métricas e vistas personalizadas.

A DWH está implementada e servirá de suporte às capacidades de *Reporting* do RightsWATCH, pelo que este estágio se torna importante no crescimento futuro da monitorização do RightsWATCH.

Existem ainda as funcionalidades de *Alerting* para desenvolver, completando assim as funcionalidades identificadas pelo estagiário no levantamento de requisitos realizado.

Referências

[1] "ITGrow," [Online]. Available: http://www.itgrow.pt/.

[2] "Watchful Software," [Online]. Available: http://www.watchfulsoftware.com.

[3] "RightsWATCH," [Online]. Available: http://www.watchfulsoftware.com/en/products/rightswatch.

[4] "The Data Warehouse Institute," [Online]. Available: http://tdwi.org/.

[5] Watchful Software, "WSW-2013-RPT-00004-information-protection-system-survey".

[6] Critical Software, S.A., "CSW-CSSECPRD-2010-PRS-01174".

[7] Microsoft, "Multidimensional eXpressions," Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/library/ms145506(v=sql.90).aspx.

[8] J. Guerra, "Why You Need a Data Warehouse," Andrews Consulting Group, 2011.

[9] R. Kimball, The Data Warehouse Lifecycle Toolkit, Wiley.

[10] Gartner, "Magic Quadrant for Data Integration Tools," 2010.

[11] "Sencha Ext JS," [Online]. Available: http://www.sencha.com/products/extjs.

[12] W. C. M. E. E. I-Y. Song, "An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling".

[13] Watchful Software, "WSW-2012-SRS-00008-rightswatch-monitoring-srs," 2012.

[14] "Kettle Project: Pentaho Data Integration," [Online]. Available: http://kettle.pentaho.com/.

Anexos

NOTA: devido ao estatuto de confidencialidade dos anexos, estes não serão submetidos na plataforma, nem impressos. No entanto, estes documentos estão disponíveis e poderão ser consultados pelo júri nos CDs entregues juntamente com o relatório final de estágio.

[1] Questionário para levantamento de requisitos, WSW-2013-RPT-00004-information-protection-system-survey

[2] Estudo de mercado, WSW-2013-RPT-00003-rightswatch-monitoring-lm-siem-market-analysis

[3] Levantamento de requisitos da Consola de monitorização do RightsWATCH, WSW-2012-SRS-00008-rightswatch-monitoring-srs

[4] Estudo de ferramentas de ETL, WSW-2013-RPT-00005-dw-technology-research

[5] Desenho detalhado do sistema, WSW-2013-SAS-00013-rightswatch-monitoring-architecture

[6] Backlog Consola de Monitorização RightsWATCH, 2013-RPT-00018-monitoring-console-backlog

[7] Especificação de Testes Consola de Monitorização RightsWATCH, WSW-2013-TCS-00026-rightswatch-monitoring-test-specification