



ESTGOH

**Escola Superior de Tecnologia e Gestão
de Oliveira do Hospital**

Instituto Politécnico de Coimbra

IT Service Management in the Organization

Relatório apresentado com vista à obtenção do grau de Mestre no âmbito da realização do ciclo de estudos de Mestrado em Informática Aplicada

Autor:

Duarte Tiago Ormonde Marote de Sousa

Orientador:

Professor Doutor Luís Veloso

Coorientador:

Professor Marco Veloso

Data: Dezembro 2015

Resumo

Atualmente, as organizações de média e grande dimensão possuem como principal objetivo apostar no seu crescimento e expansão, recorrendo por um lado à abertura de várias sucursais, sedes e *franchising* em vários pontos do mundo, e por outro à realização de parcerias.

Este crescimento tem como consequência um aumento do número de intervenientes no seu negócio. Deste modo será necessário proceder à criação de mais áreas, mais equipas, e por vezes recorrer à contratação de serviços a terceiros.

Neste contexto, é fundamental que exista um controlo de todos os fluxos de trabalho, de tarefas e de atividades, para que ocorra uma boa coordenação entre todos os elementos dessa entidade. Com este controlo e coordenação, será possível aumentar a produtividade, a eficiência e a reduzir-se os custos.

Tendo em conta estas necessidades, surgiu o *IT Service Management* (ITSM). Este consiste num conjunto de processos, atividades e boas práticas que envolve a área IT com o negócio. Possui como foco a gestão de serviços e entrega de informação relevante para o cliente, de forma a responder da melhor forma às suas necessidades e requisitos.

Neste documento é descrito em que consiste esta área de engenharia, e como é que a mesma contribui para o crescimento e maturidade de uma organização. Serão também analisados com algum detalhe algumas das plataformas que têm desempenhado um papel importante neste processo. Após esta descrição serão abordados as várias tarefas que foram desempenhadas no âmbito das plataformas de ITSM.

Abstract

Nowadays, the organizations of medium and high dimension have as main objective to invest on its growth and expansion. For that reason, in one hand they need to open several branches, headquarters and franchising in many locations around the world, and on the other hand they aim to establish partnerships.

The consequence of this growth is the increasing number of the participants in its business. This way, it is necessary to proceed with the creation of more areas, more teams, and sometimes to contract the services from third party companies.

In this context, to enable a good coordination between all elements of the entity it is fundamental that some kind of control exists over all the work flows, tasks and activities. With this control and coordination, it will be possible to increase the productivity, the efficiency and to reduce the overall costs.

The *IT Service Management* (ITSM) emerged having in consideration these needs. It consists in a set of processes, activities and best practices that involves the IT area with the business. It has as focus the service management and the delivery of relevant information to the client in order to support its needs and requisites in an efficient way.

In this document it will be described in what consists this engineering area and how it contributes to the growth and maturity of an organization. Some platforms with an important role on this process will be also analyzed with some detail. After this description, several tasks on the scope of the ITSM platforms that were performed while performing the present work will be described.

Índice

Resumo	i
Abstract	iii
Lista de Figuras	vii
Acrónimos.....	xi
1 Introdução	1
2 IT Service Management (ITSM)	3
3 Information Technology Infrastructure Library (ITIL)	5
3.1 Asset versus Configuration Item (CI)	6
3.2 Classificação de um ticket	7
3.3 Ciclo de vida de um ticket	8
4 Plataformas na área de ITSM.....	11
5 CA Service Desk Manager (CA SDM)	13
5.1 Arquiteturas de implementação do CA SDM	14
5.1.1 Arquitetura tradicional.....	14
5.1.2 Arquitetura avançada.....	15
5.2 Arquitetura lógica do CA SDM	16
5.2.1 Camada Física.....	16
5.2.2 Camada Lógica	16
5.2.3 Camada de Objeto.....	17
5.2.4 Camada do Cliente.....	17
5.3 Módulos Constituintes do CA SDM	18
5.3.1 Service Desk	21
5.3.2 Knowledge Base.....	27
5.3.3 Support Automation	27
5.3.4 Reporting.....	27
5.3.5 Configuration and Management Data Base (CMDB).....	27
5.3.6 Administration.....	28
6 CA Service Catalog (CA SC).....	29
6.1 Apresentação do CA Service Catalog	30
6.2 Elementos fundamentais do CA Service Catalog	32
6.2.1 Offerings.....	34
6.2.2 Option groups	36
6.2.3 Options.....	37
6.2.4 Forms	38
6.2.5 Report builders	40
7 CA Process Automation Manager (CA PAM).....	43
7.1 Apresentação do CA PAM	44
7.2 Exemplo de processo CA PAM	45
8 CA Business Intelligence (CA BI / BOXI).....	49
9 Integração das Ferramentas / Plataformas.....	53
10 Trabalho desenvolvido	55
10.1 CA Service Desk Manager (CA SDM)	55
10.1.1 Atividades de gestão da plataforma	55
10.1.2 Passagem de informação entre ambientes	57
10.1.3 Carregamentos massivos	60
10.1.4 Instalação de test fixes na plataforma	67
10.1.5 Criação de campos extra nos formulários	70
10.1.6 Criação de SPELL code.....	77
10.1.7 Monitorização do Tomcat.....	78
10.1.8 Monitorização e resolução de sessões bloqueadas	80
10.2 Desenvolvimento de aplicação cliente Java (WSCASDMClient)	82
10.3 CA Service Catalog (CA SC).....	88
10.3.1 Requisitos.....	88

10.3.2	Configurações implementadas	89
10.3.3	Gestão da plataforma	94
10.4	CA PAM e BOXI	95
11	Conclusão	97
12	Bibliografia	101

Lista de Figuras

Figura 1 - Organização e fluxo do SPOC. Retirado de (7).....	6
Figura 2 - Esquema sobre a diferença entre as linhas de resolução. Retirado de (7). ...	6
Figura 3 - Ciclo de vida de um <i>ticket</i> . Retirado de (7).....	8
Figura 4 - Relação entre o <i>access type</i> e a <i>role</i> (perfil). Retirado de (7).....	9
Figura 5 - Esquema representativo da Arquitetura Tradicional. Retirado de (14).	15
Figura 6 - Esquema representativo da Arquitetura Avançada. Adaptado de (14).	15
Figura 7 - Representação da Arquitetura Lógica do CA SDM. Retirado de (14).	16
Figura 8 - Ligação da plataforma CA SDM por linha de comando.	18
Figura 9 - Ligação da plataforma CA SDM pela ferramenta "Services".	18
Figura 10 - Ficheiro log da plataforma CA SDM.	19
Figura 11 - Página inicial da plataforma CA SDM.	19
Figura 12 - <i>Main Page</i> da plataforma CA SDM (secções).	20
Figura 13 - Opções de configuração da plataforma CA SDM.	21
Figura 14 - Ações que podem ser efetuadas no módulo " <i>Service Desk</i> ".	22
Figura 15 - Demonstração do processo de criação de um Incidente.	22
Figura 16 - Incidente após a sua criação.....	23
Figura 17 - Representação da secção da scoreboard.	24
Figura 18 - Demonstração do processo de criação de um Change Order.	24
Figura 19 - Visualização do <i>workflow</i> associado ao change order criado.	25
Figura 20 - Mudança de estados a nível de task.	25
Figura 21 - Demonstração do método de pesquisa de Incidentes.	26
Figura 22 - Diferença de resultados entre as tabelas <i>ca_contact</i> e <i>usp_contact</i>	28
Figura 23 - Relação entre a MDB e as diferentes áreas. Retirado de (18).....	28
Figura 24 - Esquema explicativo da necessidade do Service Catalog.....	29
Figura 25 - Detalhes do pedido de Service Catalog (secções).	32
Figura 26 - Esquema dos elementos do <i>Service Catalog</i>	33
Figura 27 - Esquema dos elementos do Service Catalog (aplicado a exemplo).	34
Figura 28 - Secção relativa à offering.....	34
Figura 29 - Secção relativa à offering (Definition).....	35
Figura 30 - Inserção/remoção de option groups da offering	36
Figura 31 - Secção referente às options groups.....	36
Figura 32 - Secção referente às options groups (Definition).....	37
Figura 33 - Secção referente à option.	38
Figura 34 - Secção referente ao form.....	39
Figura 35 - Secção do script do form.....	40
Figura 36 - Listagem das report builders criadas.....	41
Figura 37 - Construção de uma report builder.	41
Figura 38 - <i>Main Page</i> da plataforma ITPAM.	44
Figura 39 - Conteúdo da tab Libray.....	45
Figura 40 - Conteúdo da tab Operations.	45
Figura 41 - Exemplo de processo PAM.	46
Figura 42 - Código js para verificação do tipo de request.....	47
Figura 43 - Código de pós-execução do operador de login.	47
Figura 44 - Código de pós-execução do operador de criação do request.....	47
Figura 45 - Main page da plataforma CABI/BOXI.	49
Figura 46 - Criação de novo relatório CABI.	50
Figura 47 - Relatório CABI criado.	50
Figura 48 - Zona de configuração do relatório.....	50
Figura 49 - SQL gerado através do drag-and-drop.....	51
Figura 50 - <i>Output</i> do relatório.	51
Figura 51 - Relação entre as várias plataformas.	53
Figura 52 - Visualização de detalhes de um contacto.	58

Figura 53 - Procedimento para visualizar o código fonte de uma página.....	58
Figura 54 - Descrição da obtenção do objeto através do código fonte da página.	58
Figura 55 - Utilização do comando <code>bop_sinfo -f</code>	59
Figura 56 - Utilização do comando <code>bop_sinfo -dq</code>	59
Figura 57 - Utilização do comando <code>pdm_extract</code>	60
Figura 58 - Utilização do comando <code>pdm_userload</code>	60
Figura 59 - Visualização do ficheiro Excel a ser utilizado na importação massiva.	61
Figura 60 - Denominação das tabelas de conteúdos através do "Gestor de Nomes". .	61
Figura 61 - Descrição da atribuição de um nome a uma tabela.	62
Figura 62 - Formatação dos campos como texto.....	62
Figura 63 - Ficheiros que constituem a aplicação <i>CASDMSync</i>	62
Figura 64 - Ficheiro de configuração (acessos).....	62
Figura 65 - Ficheiro de configuração (indicação do WSDL e do ficheiro de Log).....	63
Figura 66 - Ficheiro de configuração (indicação da tabela e ODBC).	63
Figura 67 - Ficheiro de configuração (indicação do objeto).	63
Figura 68 - Ficheiro de configuração (mapeamento de campos).....	63
Figura 69 - Comando <code>bop_sinfo</code> na elaboração do mapeamento de campos.	64
Figura 70 - Ficheiro de configuração (modos de execução).	65
Figura 71 - Localização da aplicação que permite a criação de ODBC's.....	65
Figura 72 - Criação da ODBC.	66
Figura 73 - Como abrir uma linha de comandos na presente janela.....	66
Figura 74 - Execução da ferramenta <i>CASDMSync</i> em modo <code>false false</code>	66
Figura 75 - Execução da ferramenta <i>CASDMSync</i> em modo <code>true false</code>	67
Figura 76 - Visualização das IA's criadas por <i>CASDMSync</i>	67
Figura 77 - Componentes necessários para instalação de patch no CA-SDM.....	68
Figura 78 - Método de descompressão do <i>test fix</i>	68
Figura 79 - Test fix após descompressão.....	69
Figura 80 - Diretoria da aplicação <i>Applytpf</i>	69
Figura 81 - Menu de instalação do test fix.	69
Figura 82 - Seleção do ficheiro <code>.jcl</code>	70
Figura 83 - Instalação de <i>test fix</i>	70
Figura 84 - Descrição da obtenção do form que está a ser utilizado.....	70
Figura 85 - Indicação de como localizar o WSP.	71
Figura 86 - Descrição da localização do Schema Designer.....	71
Figura 87 - Schema Designer.	72
Figura 88 - Atribuição do nome e configuração da coluna a ser criada.	72
Figura 89 - Publish da alteração da schema.	73
Figura 90 - Comando <code>net stop pdm_daemon_manager</code> e <code>pdm_publish</code>	73
Figura 91 - <code>bop_sinfo</code> para validação da criação do novo campo.	74
Figura 92 - Processo para abertura do form.....	75
Figura 93 - Criação e mapeamento do novo campo do form.	76
Figura 94 - Confirmação do publish referente à alteração de form.	76
Figura 95 - Novo campo no form.....	76
Figura 96 - Localização da pasta que contem os spell codes.....	77
Figura 97 - Exemplo de ficheiro <code>.mod</code> (trigger).	77
Figura 98 - Exemplo de ficheiro <code>.spl</code> (definição da ação).	78
Figura 99 - Procedimento para se anexar ficheiro a ticket.....	78
Figura 100 - Erro que surge quando existem problemas no Tomcat.	79
Figura 101 - Erro no log quando existem problemas no Tomcat.	79
Figura 102 - Método de verificação dos estados dos processos Tomcat.....	79
Figura 103 - Listagem dos processos Tomcat utilizando o "Process Explorer"	80
Figura 104 - Método para efetuar restart aos processos Tomcat.	80
Figura 105 - Erro no log referente às sessões bloqueadas nos servidores.....	81
Figura 106 - Erro na criação de incidentes/reclamações nas sessões bloqueadas.	81
Figura 107 - Comando <code>pdm_webstat</code> para visualização das sessões ativas.	81
Figura 108 - Descrição de como retirar o servidor do balanceamento.....	82

Figura 109 - Esquema de uma integração entre plataformas SDM.	83
Figura 110 - Visualização da página WSDL da plataforma.....	84
Figura 111 - Localização do daemon para importação das funções do WSDL.	84
Figura 112 - Comando para a importação das funções do WSDL.	84
Figura 113 - Conteúdo da package resultante da execução do comando <i>wsimport</i>	85
Figura 114 - Ficheiros que constituem a package da aplicação cliente.	85
Figura 115 - Ficheiro de configuração da aplicação.	85
Figura 116 - Código da classe WSCAtest (ligação ao ws).....	86
Figura 117 - Função utilizada para se obter o SID para ligação ao ws.	86
Figura 118 - Função de criação de Request.....	87
Figura 119 - Execução do cliente.	87
Figura 120 - Resultado após execução do cliente (no CA SDM).	88
Figura 121 - Organização do projeto de SC do cliente.	89
Figura 122 - Associação de scripting aos eventos das componentes do form.	94

Acrónimos

AD	Active Directory
CA BI/BOXI	CA Business Intelligence
CA ITAM	CA IT Assets Management
CA NSM	CA Nimsoft
CA PAM	CA Process Automation Management
CA SC	CA Service Catalog
CA SDM, SD	CA Service Desk
CI	Configuration Item
CMDB	Configuration Management Data Base
BD	Base de Dados
BU	Business Unit
DBA	Data Base Administration
DEV	Ambiente de Desenvolvimento
FTP	File Transfer Protocol
IA	Incident Area
IE	Internet Explorer
IIS	Internet Information Server
ISO	International Organization of Standardization
IT, SI	Information Technology, Sistemas de Informação
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
JS	JavaScript
KB	Knowledge Base
KPI	Key Performer Indicators
PAM	Process Automation Management
ODBC	Open Data Base Connectivity
OOTB	Outside of the Box
PRD	Ambiente de Produção
QLD	Ambiente de Qualidade
RDBMS	Relational Data Base Management System
SC	Service Catalog
RFF	Request for Fulfillment
RFR	Request for Release
SGBD	Sistema Gestor de Base de Dados
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SPOC	Single Point of Contact

SQL	Structured Query Language
SSO	Single Sign On
STG	Ambiente de Staging
URL	Uniform Resource Locator
VM	Virtual Machine
XML	eXtensible Markup Language
WS	Web Service

1 Introdução

O presente trabalho tem como base o tema *IT Service Management* (ITSM) e a sua vital importância para uma organização.

A escolha deste tema deve-se maioritariamente a duas razões. A primeira prende-se ao facto de ter sido com a área de ITSM que foi iniciado e desenvolvido até ao momento o percurso a nível profissional. A segunda deve-se ao valor que esta área representa para organizações de média e grande dimensão.

Quanto maior for a dimensão de uma empresa, mais difícil será gerir todo o fluxo de atividades efetuadas por todos os intervenientes pertencentes a essa entidade, ou que estejam relacionados com a mesma. Tal poderá conduzir a uma falta de organização, de eficácia e ao aumento do tempo de resposta e de resolução face aos mais variados pedidos dos utilizadores, assim como dos possíveis obstáculos que poderão surgir.

É por isso fundamental que exista um controlo de todos os fluxos de trabalho, de tarefas e de atividades, para que então ocorra uma boa coordenação entre todos os elementos dessa entidade. Com este controlo e coordenação, será possível:

- Aumentar a produtividade, a eficiência e a eficácia;
- Melhorar a organização entre as áreas e as equipas envolvidas no negócio;
- Reduzir o tempo de resposta e de resolução;
- Reduzir os custos;
- Facilitar o processo de faturação.

Deste modo recorrendo a um conjunto de boas práticas, e a várias plataformas de ITSM, será possível satisfazer as necessidades e objetivos de cada cliente.

Neste relatório serão descritos os conceitos fundamentais de ITSM e as plataformas que neste âmbito são utilizadas, desempenhando papéis importantes relativamente à sustentabilidade desta área. Posteriormente, será efetuada uma apresentação das competências adquiridas neste ramo de engenharia, e serão descritas as atividades realizadas durante este período. Estas foram desenvolvidas no âmbito particular das seguintes plataformas ITSM: *CA Service Desk (SDM)*, *CA Service Catalog (SC)*, *CA Process Automation Manager (PAM)* e *CA Business Intelligence (BI /BOXI)*.

2 IT Service Management (ITSM)

O ITSM consiste num conjunto de processos, atividades e boas práticas que envolve a área IT com o negócio, tendo como grande foco a gestão de serviços e entrega de informação para o cliente, de forma a responder de forma satisfatória às suas necessidades e requisitos (2). Deste modo, o ITSM tem como principal objetivo a agilização de processos, consistindo estes na gestão de pedidos, de problemas e de incidentes referentes à organização.

Este tipo de gestão possui um grau elevado de complexidade, uma vez que é necessário existir uma categorização de todos os pedidos, incidentes e problemas, a construção de *workflows* caso a resolução tenha mais do que um interveniente, e a criação dos grupos responsáveis por essa resolução. É também importante definir períodos de tempo designados *Service Level Agreements* (SLA) para a capacidade de resposta e conclusão, assim como da aplicação de vários tipos de monitorização desses mesmos tempos. Com a gestão diária de serviço, poder-se-á garantir ao cliente (tendo em conta as metas que este pretenda atingir) uma entrega da informação referente ao grau de organização, da capacidade de resposta e resolução, entre outras métricas.

Para tal é necessário recorrer à utilização de plataformas e ferramentas que auxiliam no processo de gestão e agilização de processos. A nível técnico é importante dominar a instalação, configuração e personalização destas plataformas de forma a responder às necessidades do cliente, que vão variando ao longo do tempo. Assim como garantir a alta disponibilidade das mesmas (aproximadamente 24 horas por dia). Como estas plataformas estão sujeitas a alterações e melhorias contínuas é importante também que se garanta que as mesmas se encontrem devidamente atualizadas e funcionais (1).

Produto versus Serviço

Neste contexto há uma clara distinção entre produto e serviço. Recorrendo a um exemplo, se um cliente adquirir um computador por 1000€ estará a adquirir um produto que servirá para prestar um ou mais serviços como a capacidade de ligação à internet, impressão e digitalização documentos e envio/receção de *emails*. Para além do custo do portátil, seria também necessário adicionar a esse valor o custo de aquisição de uma impressora, ligação à internet, papel, tinteiros, entre outros.

Alternativamente, este cliente poderia optar por dirigir-se a um centro de cópias ou uma *lan house* nos quais encontraria os mesmos serviços mas não tendo de efetuar um pagamento tão elevado.

Do ponto de vista do centro de cópias ou da *lan house*, que possuem todos os equipamentos, quando estes apresentarem problemas, poderão contar com o suporte do fornecedor/fabricante, de manuais/guias auxiliares, e mesmo que não apresentem nenhum problema, é importante e necessário haver uma gestão constante a nível de produtos assim como de serviços. Neste exemplo teremos como produtos:

- Para cada computador:
 - Segurança (com a utilização de um antivírus, *firewall*, monitorização);
 - Memória;
 - Espaço no disco;
- Para cada impressora:
 - Níveis dos tinteiros;
 - Quantidade de papel restante por formato;
- *Routers*;
- Servidores.

Como serviços teremos

- Ligação à internet;
- Impressão;
- Cópias e Digitalização;
- *Emails*.

Relativamente aos serviços, é necessário deter o conhecimento de técnicas de gestão dos mesmos:

- Gestão de incidentes, pedidos e problemas;
- Requisições de serviços;
- Gestão de mudanças;
- Gestão de licenças;
- Gestão de configurações;
- Gestão financeira;
- Gestão de disponibilidade;
- Gestão de capacidade;
- Gestão de níveis de serviço;
- Outros (6).

3 Information Technology Infrastructure Library (ITIL)

Considerando uma empresa de grandes dimensões, em que o seu crescimento poderá resultante de várias fusões e aquisições, e que estará presente nas mais variadas vertentes do mercado, estaremos perante variados sistemas assim como um grande volume de dados e informação.

A gestão de uma empresa destas dimensões será difícil conduzindo a mesma a procurar constantemente uma agilização dos seus processos e operações, e por conseguinte, a seguir a abordagem *Information Technology Infrastructure Library* (ITIL).

Este consiste num conjunto de boas práticas e conhecimentos a serem aplicadas numa infraestrutura relativamente à sua gestão de serviços (ITSM). Ou seja, consiste em realizar uma gestão de modo a que se obtenham melhor desempenho, melhor capacidade de resposta e de solução de adversidades que possam vir a surgir. Desta forma, possibilita a realização de uma melhor gestão de incidentes, pedidos, problemas, entre outros (5).

No entanto, o ITIL não é prescritivo nem se trata de uma metodologia. Cada TI e empresa têm diferentes visões de negócio, assim como diferentes objetivos a serem atingidos, tendo o ITIL sido desenvolvido para se adaptar à realidade de qualquer que seja a empresa em questão (3 e 7).

A empresa irá seguir um processo denominado de *Single Point of Contact* (SPOC) descrito na seguinte figura em que:

- Todas as chamadas/pedidos são filtradas por um analista de nível 1;
- Cada analista de nível 1 tem a função de criar e resolver pedidos e incidentes;
- Os pedidos e incidentes (também chamados de *tickets*) que o analista de nível 1 não conseguir resolver, vai ter então que escalá-los para os analistas de nível 2;
- Quando os analistas de nível 2 resolverem esses *tickets*, voltam a transferi-los para os analistas de nível 1, de forma a confirmar com o utilizador que estes ficaram resolvidos para posteriormente serem fechados;
- Os analistas de nível 2 podem abrir os problemas ou pedidos de alteração que forem necessários para resolverem os incidentes (7).

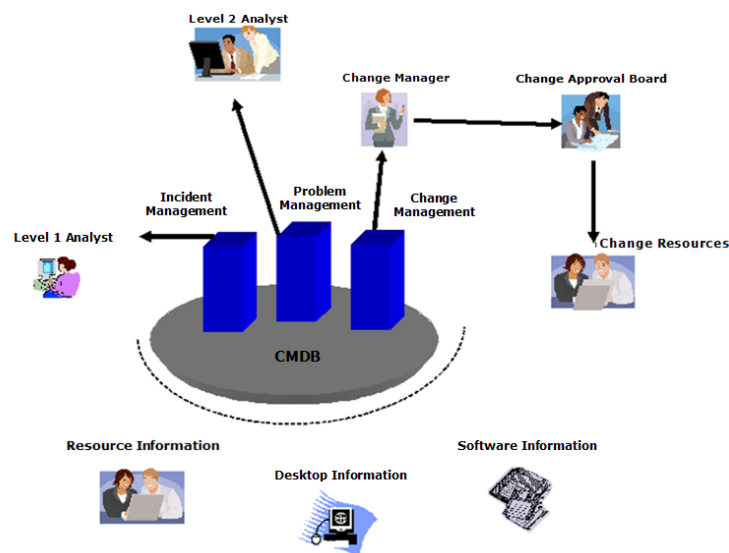


Figura 1 - Organização e fluxo do SPOC. Retirado de (7).

De notar que por vezes é necessário existirem mais do que 2 linhas de resolução. Na seguinte figura é apresentado um esquema relativo à diferença entre as linhas de resolução.

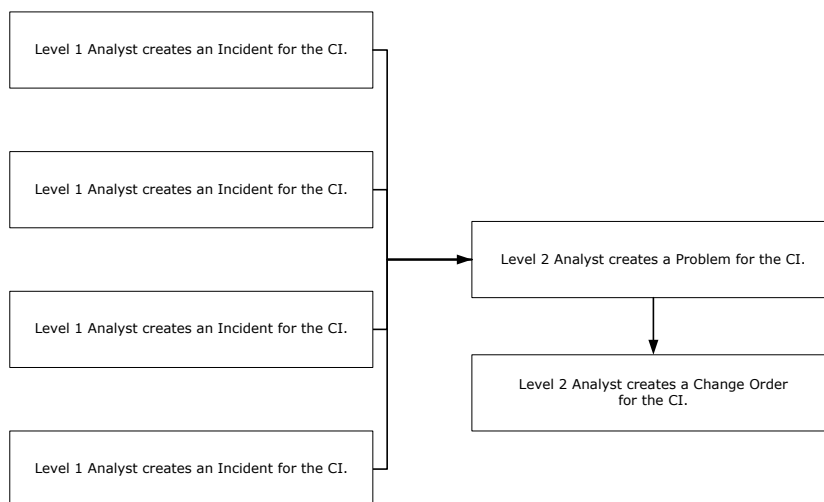


Figura 2 - Esquema sobre a diferença entre as linhas de resolução. Retirado de (7).

3.1 Asset versus Configuration Item (CI)

Estes são dois conceitos que apesar de similares são diferentes. Um asset é tudo o que representa um valor financeiro para a empresa. Deste modo um inventário será uma lista de assets.

Um *Configuration Item* (CI) representa o valor a nível da operação, ou seja, são elementos que são necessários para o bom funcionamento dos processos implementados na organização. Estes normalmente contêm e gerem a informação do

sector IT, assim como de todas as suas relações. Como exemplo de CI temos: servidores e as várias plataformas (e.g. CA SDM, Artemis, Maximos).

Assim sendo, um ticket está (normalmente) associado a um CI para que desta forma se possa fazer estudos estatísticos (e.g. quantas vezes apresentou problemas, quantas vezes teve que ser reparado ou sofrer algum tipo de intervenção) (4).

3.2 Classificação de um ticket

Um *ticket* poderá ser classificado de uma das seguintes formas:

- **Pedido (request):** Quando não tem impacto no negócio, ou seja, não é impeditivo nem prejudica o processo de atividade ou gestão do serviço. Normalmente é utilizado quando o *end user* (utilizador comum não técnico da organização) tem alguma dúvida ou solicitação.
- **Incidente (incident):** Quando prejudica o negócio ou serviço, apresentando alguma urgência na resolução (e.g quando um utilizador não tem acesso a uma determinada ferramenta, quando um *site* se encontra indisponível, ou quando a execução de um processo falha e é necessário a sua reposição).
- **Problema (problem):** Normalmente um problema é criado quando a gravidade do incidente é mais acentuada do que o previsto, e desta forma apresenta uma maior ameaça ao negócio ou processo de gestão de serviços, o que leva a que o tempo da sua resolução seja mais demorado.
- **Pedido de alteração (change order):** Um *change order* consiste num *request*, mas com um *workflow* associado. Por exemplo, a empresa acaba de contratar um novo funcionário, sendo desta forma necessário criar este funcionário no Active Directory (AD) (i.e. passar pela equipa de gestão de utilizadores), criar e configurar uma conta de *Exchange* (i.e. necessário passar pela equipa de gestão de conta de emails), facultar-lhe um portátil para poder desempenhar as suas funções (i.e. passar pela departamento de informática). Deste modo, para que o processo de entrada deste novo funcionário fique concluído é necessário passar por vários departamentos e equipas, sendo por isso necessário abrir um “*change order*”.
- **Issue:** Consiste num problema, ao qual após análise e diagnóstico, foi encontrada uma solução que foi posteriormente publicada. Desta forma, sempre que surgir um problema é necessário pesquisar por *issues* com descrições semelhantes para que se o possa resolver, de forma rápida e

eficiente, em vez de se realizar a análise e diagnóstico de um problema para o qual já existe uma solução publicada.

3.3 Ciclo de vida de um ticket

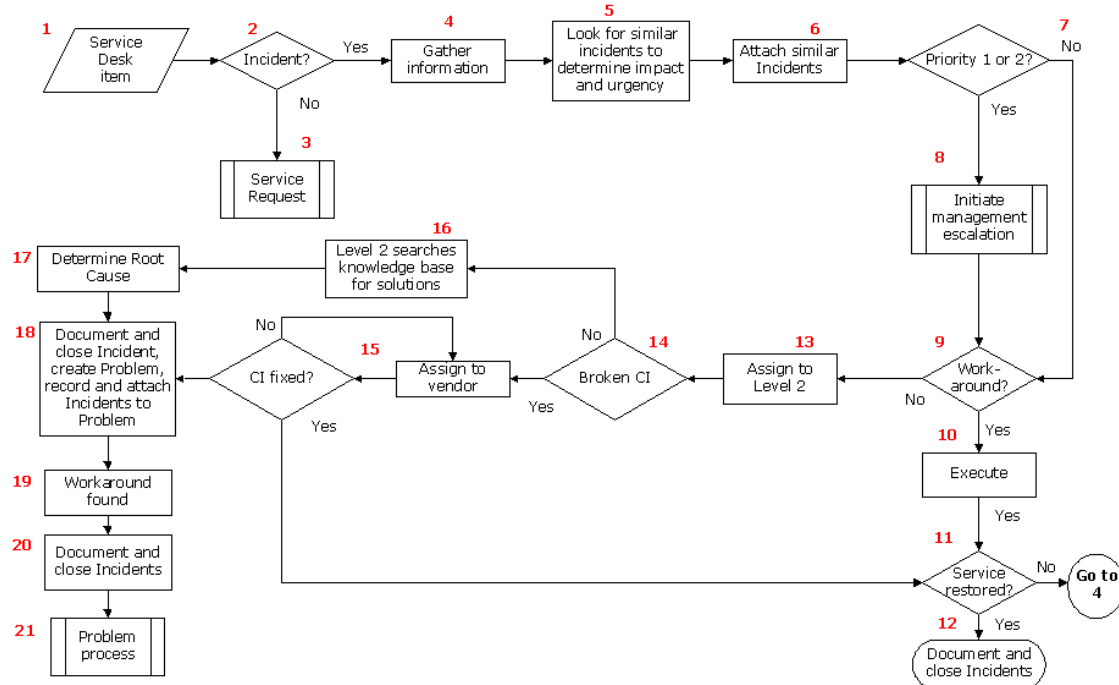


Figura 3 - Ciclo de vida de um *ticket*. Retirado de (7).

Neste esquema, podemos visualizar o ciclo de vida que um *ticket* usualmente possui:

1. Através de um *email*, um telefonema ou através de um interface *self-service* (como o *Service Catalog*), um pedido é enviado para o nível 1;
2. No nível 1, este pedido é analisado e posteriormente classificado como um pedido ou um incidente;
3. Tratando-se de um pedido, vai ser lidado através do processo de pedido de serviço;
4. Se for considerado um incidente, é necessário obter alguma informação sobre o mesmo;
5. De seguida o analista deve efetuar uma pesquisa de incidentes similares de forma a identificar-se o impacto e a urgência;
6. Se for encontrado algum incidente similar, este é anexado ao incidente;
7. A urgência e o impacto que foram determinados anteriormente vão ser utilizados para se determinar a prioridade;
8. Depois de determinada a prioridade, o analista começa então a trabalhar na resolução desse incidente;
9. Começa por procurar na *Knowledge Base* (KB) por documentos com soluções ou *workarounds* disponíveis;

10. Se o analista encontrar algum documento, vai executar essa solução/*workaround*,
11. Depois de implementada, é necessário validar se o problema ficou verdadeiramente resolvido;
12. É comunicado ao *end user* que o seu pedido foi resolvido, documenta-se o incidente e por fim colocamos o estado como resolvido. Após algum tempo o estado passará a fechado (se não houver uma reabertura por parte do *end user*);
13. Se não existir nenhum *workaround*, este incidente deverá ser escalado, ou seja, ser passado para o nível 2;
14. Se o problema estiver no *CI*, é necessário obter a informação da garantia e passar este incidente ao fabricante (alterando o estado do *CI* para "*In Repair*");
15. Se o problema for realmente o *CI*, quando o fabricante reparar ou substituí-lo, o incidente é resolvido e posteriormente fechado;
16. Se não se tratar de um problema no *CI*, os analistas têm que voltar a efetuar uma pesquisa na *KB* deste nível;
17. Se não for possível restaurar o serviço, é necessário encontrar a causa do problema;
18. Desta forma, um problema é criado e a referência desse problema é anexada ao incidente;
19. Quando um *workaround* for descoberto, é necessário documentar e colocá-lo na *KB*, passando todos os incidentes relacionados ao nível 1 onde esse *workaround* será aplicado.
20. Quando o problema for resolvido, o processo resume-se novamente ao do passo 19 (7, 10 e 12).

Access Type	Roles
IT Staff	Level 2 Analyst
Process Management	Level 2 Analyst
Service Desk Management	Level 1 Analyst Level 2 Analyst
Service Desk Staff	Level 1 Analyst

Figura 4 - Relação entre o *access type* e a *role* (perfil). Retirado de (7).

4 Plataformas na área de ITSM

Como mencionado anteriormente, para se efetuarem os mais variados tipos de gestão e atividades referentes à área do ITSM, é necessário recorrer-se à utilização de um conjunto de plataformas e ferramentas que nos auxiliam de forma organizada, simples e eficaz. Embora não seja obrigatório a utilização dessas ferramentas deverá seguir de preferência as práticas do ITIL.

Para tal, existem no mercado várias ferramentas de *ticketing*, *self-service*, gestão de *assets* e CI's, licenciamento de software, entre outros. Aquelas que mais se destacam a nível empresarial são as seguintes:

- **CA Technologies**
 - CA Service Desk (SDM);
 - CA Service Catalog (SC);
 - CA Process Automation Manager (PAM);
 - CA Business Intelligence (BI /BOXI).
- HP Helpdesk;
- BMC Remedy;
- Service Now.

Neste documento serão descritas as plataformas da empresa *CA Technologies*, pois foram as ferramentas efetivamente utilizadas neste trabalho.

Como em todas as ferramentas, as fornecidas pela CA apresentam vantagens e desvantagens relativamente às restantes. Infelizmente, é difícil mencioná-las uma vez que estas plataformas são proprietárias. Desta forma, não é possível fazer-se uma exploração aprofundada de forma a encontrar pontos fortes e fracos assim como da obtenção de informação sobre as mesmas, como por exemplo uma listagem das funções principais, guias, entre outros.

5 CA Service Desk Manager (CA SDM)

O CA Service Desk Manager é uma ferramenta/plataforma de “*ticketing*” que tem como objetivo funcionar como um *SPOC* entre os vários funcionários de uma ou mais empresas (entre funcionários “comuns” e funcionários das TI). Desta forma haverá uma agilização da comunicação e da gestão de processos, uma imposição de limites de tempo para resposta e resolução dos *tickets*, assim como de um registo de todas as atividades.

Como descrito anteriormente, para uma empresa de dimensões elevadas, que é constituída por vários setores, departamentos e com um elevado número de funcionários, é necessário implementar uma ferramenta que possibilite gerir o serviço de uma forma global.

Por exemplo, considere-se que está ser implementado nessa empresa um projeto em que vários departamentos estão envolvidos. Ou seja, a evolução do mesmo consiste na execução de várias tarefas, dependendo estas da receção de informação de tarefas anteriores.

Sem a utilização de uma ferramenta de *ticketing* como o *CA Service Desk Manager* (ou outra ferramenta para o efeito) é extremamente difícil controlar, de forma organizada, a evolução deste projeto. Não existiriam registos relativamente a quais as tarefas que são necessárias, quais as equipas/departamentos responsáveis por cada tarefa, o tempo estipulado e o tempo que efetivamente utilizado para a sua realização, um canal de comunicação entre os intervenientes, entre outros.

A utilização do *CA Service Desk Manager* permitirá o registo das diversas atividades e interações. Utilizando o cenário anteriormente mencionado, antes de se proceder à execução do projeto, seria definido um *workflow* que será composto por um conjunto de tarefas. Essas tarefas consistiriam na descrição do trabalho que seria expectável, qual o departamento/grupo responsável pelo mesmo, e definição do tempo limite para a execução do mesmo (SLA). Na situação desses departamentos precisarem de entrar em contato entre eles, poderiam proceder à abertura de *tickets*.

Assim, no decorrer da evolução desse projeto um departamento irá violar o *SLA*, se falhar ou levar mais tempo do que estava inicialmente estipulado a executar a tarefa que lhe estava atribuída, e não apresentar uma razão plausível para o sucedido

Este é um cenário que ocorre diariamente em grandes entidades/organizações, que recorrem à subcontratação de várias empresas para a prestação dos seus mais variados serviços, e assim sendo é muito importante haver este tipo de gestão e controlo.

Em seguida irá ser descrito mais aprofundadamente em que consiste o *CA Service Desk* a nível técnico.

5.1 Arquiteturas de implementação do CA SDM

Em todo o tipo de implementações do *CA Service Desk* (CA SDM) irá sempre verificar-se a utilização de uma e uma só *Configuration Management Data Base* (CMDB) (base dados), assim como de um servidor primário. Quanto aos servidores secundários, estes são opcionais (normalmente os secundários só contêm os processos de *web engine* e de *DOM server*).

Assim sendo, existem 2 tipos de arquitetura do CA SDM: a Arquitetura Tradicional e a Arquitetura Avançada.

5.1.1 Arquitetura tradicional

A arquitetura tradicional (também chamada de básica) é a arquitetura mais simples de ser implementada, sendo constituída por:

- 1 CMDB;
- 1 Servidor Primário;
- 1 ou mais Servidores Secundários.

Este tipo de arquitetura é normalmente utilizado por empresas de pequenas/médias dimensões.

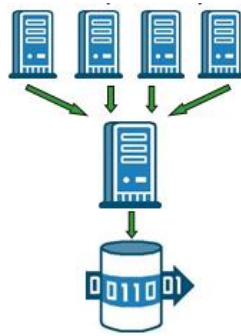


Figura 5 - Esquema representativo da Arquitetura Tradicional. Retirado de (14).

Os benefícios de utilizar servidores secundários serão os seguintes:

- ✓ Escalabilidade - as boas práticas recomendam adicionar-se um web engine a cada 200-400 utilizadores;
- ✓ Possibilidade de se ter vários *web servers*;
- ✓ Balanceamento de carga;
- ✓ *Single Sign On* (SSO);
- ✓ Integração com CA NSM (ferramentas de monitorização) ou outras (14).

5.1.2 Arquitetura avançada

A arquitetura avançada (também denominada de arquitetura de alta disponibilidade) é utilizada nas empresas de grande dimensão onde o principal objetivo é reduzir ao máximo o grau de indisponibilidade. Para tal, recorre-se a:

- 1 CMDB;
- 1 ou mais Servidores Standby;
- 1 Servidor Background (que vai servir de primário, ao qual os utilizadores não têm acesso direto);
- 1 ou mais Servidores Secundários (onde são realizados os acessos pelos utilizadores).

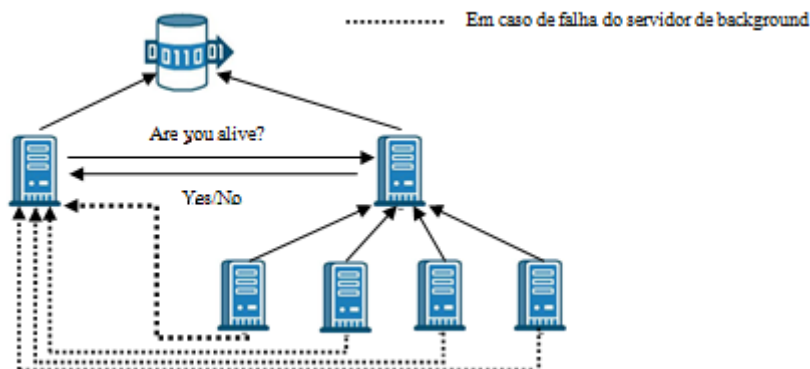


Figura 6 - Esquema representativo da Arquitetura Avançada. Adaptado de (14).

5.2 Arquitetura lógica do CA SDM

Relativamente à arquitetura lógica do SDM, esta é composta por quatro camadas, que se encontram representadas na seguinte figura.

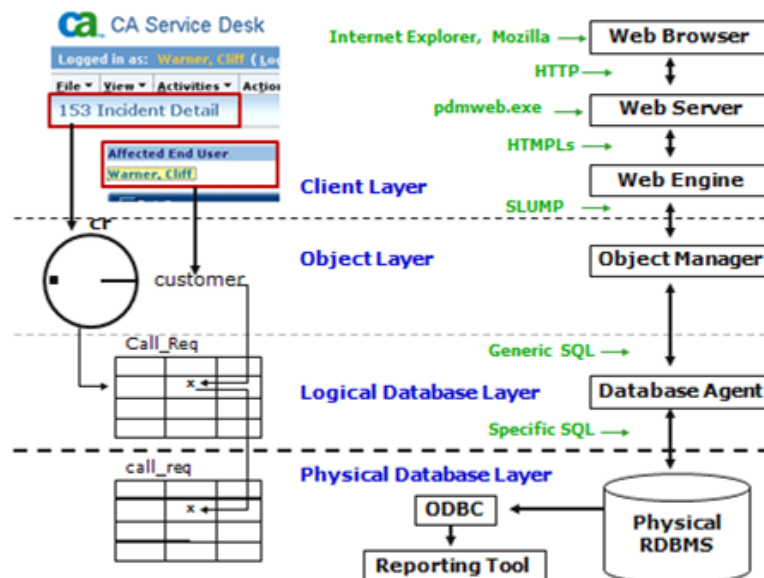


Figura 7 - Representação da Arquitetura Lógica do CA SDM. Retirado de (14).

5.2.1 Camada Física

Nesta camada podemos encontrar a CMDB, que consiste na base de dados responsável pelo armazenamento e gestão de todos os dados usados pelo *CA Service Desk* (assim como por outras ferramentas) (14).

5.2.2 Camada Lógica

Aqui irá encontrar-se um agente, usualmente denominado por *<nome da plataforma>_agent*, que tem como função estabelecer a ligação entre o produto e a base de dados. Ao receber da camada de objetos instruções de SQL genéricas irá convertê-las em instruções de SQL específicas. Para a execução desta tarefa, o agente tem de aderir à *Database Independent (DBI)*, constituída por um conjunto de tabelas e declarações de campos (denominado por *schema*) que vai ser utilizada para efeitos de mapeamento. Por exemplo, os *requests*, *incidents* e *problems* são guardados na DBI/tabela lógica *Call_Req*, que depois de traduzida para a camada física, vai consistir na tabela *call_req* na MDB.

A utilização de um agente/camada intermédia traz várias vantagens nomeadamente:

- permitir a troca do tipo da RDBMS (ex: SQL Server para Oracle). Para tal bastará proceder à instalação de um conjunto de agentes respetivos para que o CA Service Desk passe a utilizar esse novo tipo;
- em empresas de grande dimensão, as equipas de *Data Base Administration* (DBA) poderão não permitir um acesso direto à base de dados por ferramentas de manipulação de dados, como os SGBD's;
- a manipulação de dados é feita de forma mais segura. Por exemplo, se alguém decidisse fazer carregamentos em massa, seria necessário haver uma noção de quais os campos, tabelas, relações e as dependências que estavam envolvidas, porque caso contrário poderia resultar numa corrupção ou duplicação de registos (14).

5.2.3 Camada de Objeto

Nesta camada irá encontrar-se o *object manager*, normalmente representado pelo processo *domsrvr* ou *Domain Server*. Este permite-nos, interagir com cópias dos objetos que foram mencionados na camada lógica.

Por exemplo, quando um utilizador cria um novo incidente, o *object manager* irá criar um objeto do tipo *cr* na memória, e só após efetuar o “*save*” é que a informação é passada da memória para os atributos do verdadeiro objeto *cr* (na CMDB). O *object manager* também é responsável pela vertente da segurança do CA SDM. Ou seja, é ele que gere a atribuição de *access types* e *data partitions*, quando um utilizador se liga à plataforma.

É por esta razão que comandos como o *pdm_userload* (que irá descrito mais tarde neste relatório) só podem correr no servidor primário (14).

5.2.4 Camada do Cliente

Nesta camada irá encontrar-se o processo de *web engine* normalmente representado por um *web server* que usualmente é um Tomcat ou um IIS, e um *browser* (como o IE ou Firefox) (14).

5.3 Módulos Constituintes do CA SDM

Para a apresentação do CA SDM, optou-se por criar uma *Virtual Machine* (VM) e proceder à instalação do software. Desta forma será não só mais fácil demonstrar algumas das funcionalidades como também será mais seguro, uma vez que se irá lidar com informação que foi criada para esse efeito (demonstração).

Em primeiro lugar é necessário iniciar a plataforma que poderá ser realizada de duas formas:

- Linha de comandos, utilizando o comando:

```
net start pdm_daemon_manager
```

```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>net start pdm_daemon_manager_
```



```
C:\Users\Administrator>net start pdm_daemon_manager
The CA Service Desk Manager Server service is starting.....
The CA Service Desk Manager Server service was started successfully.
```

Figura 8 - Ligação da plataforma CA SDM por linha de comando.

- Através dos “Services”:

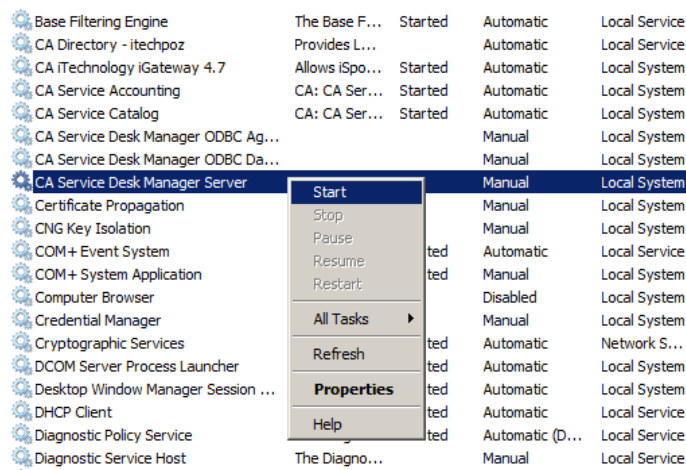


Figura 9 - Ligação da plataforma CA SDM pela ferramenta "Services".

No processo de levantamento da plataforma é necessário analisar os ficheiros de *logs* de forma a monitorizar possíveis avisos ou erros.

```

09/27 22:30:09.78 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory fu
09/27 22:30:09.89 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory sa
09/27 22:30:09.91 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory sa
09/27 22:30:10.15 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory sa
09/27 22:30:10.16 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory sa
09/27 22:30:10.34 WIN2008R2 web:wsp 2260 SIGNIFICANT sel_data_cache.c 1486 Factory pd
09/27 22:30:13.48 WIN2008R2 pdm_d_mgr 3200 SIGNIFICANT mgr_slump_if.c 1221 Startings
09/27 22:30:13.75 WIN2008R2 proctor_WIN2008R2 3236 SIGNIFICANT pdm_process.c 572 Process St
09/27 22:30:14.51 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 530 Select que
09/27 22:30:16.79 WIN2008R2 bpeid_nxd 5100 SIGNIFICANT bpebr_main.c 786 EBR::EID:
09/27 22:30:17.24 WIN2008R2 domsrvr 4268 SIGNIFICANT connmgr.c 2405 Connecting
09/27 22:30:17.25 WIN2008R2 bpeid_nxd 5100 SIGNIFICANT bpebr_main.c 1507 EBR::EID:
09/27 22:30:17.87 WIN2008R2 bpeid_nxd 5100 SIGNIFICANT EBRFileIndex.c 1846 Successful
09/27 22:30:17.87 WIN2008R2 bpeid_nxd 5100 SIGNIFICANT bpebr_main.c 1452 UNCLibrary
09/27 22:30:18.21 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 504 Select que
09/27 22:30:20.99 WIN2008R2 pdm_tomcat 4580 SIGNIFICANT pdm_tomcat.c 938 SERVICEDES
09/27 22:30:21.20 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 504 Select que
09/27 22:30:22.64 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 530 Select que
09/27 22:30:22.79 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 504 Select que
09/27 22:30:22.98 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 530 Select que
09/27 22:30:23.89 WIN2008R2 rep_daemon 4348 SIGNIFICANT RepDaemon.c 408 Repository
09/27 22:30:24.48 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 530 Select que
09/27 22:30:24.53 WIN2008R2 bpebr_nxd 5024 SIGNIFICANT kdfilter.c 673 KD filter
09/27 22:30:24.53 WIN2008R2 kt_daemon 4396 SIGNIFICANT KTDaemon.c 1479 Method ebr
09/27 22:30:24.62 WIN2008R2 bpvirtdb_srvr 3780 SIGNIFICANT vdbagent.c 522 Select que
09/27 22:30:25.26 WIN2008R2 boplogin 3764 SIGNIFICANT bplhndlr.c 2013 Restoring
09/27 22:30:25.26 WIN2008R2 boplogin 3764 SIGNIFICANT bplhndlr.c 2013 Restoring
09/27 22:30:25.26 WIN2008R2 boplogin 3764 SIGNIFICANT bplhndlr.c 1324 Boplogin in
09/27 22:30:26.37 WIN2008R2 bu_daemon 4372 SIGNIFICANT BURate.c 754 KI bubble
09/27 22:30:27.22 WIN2008R2 domsrvr 4268 SIGNIFICANT stc_mgr.c 1181 Transition
09/27 22:30:27.24 WIN2008R2 domsrvrwsp 4252 SIGNIFICANT stc_mgr.c 1181 Transition

```

Figura 10 - Ficheiro log da plataforma CA SDM.

Após a conclusão deste processo, dever-se-á abrir um browser e inserir o link:

<http://<hostname da máquina>/CAisd/pdmweb.exe>

De seguida deverá surgir uma janela semelhante à seguinte:

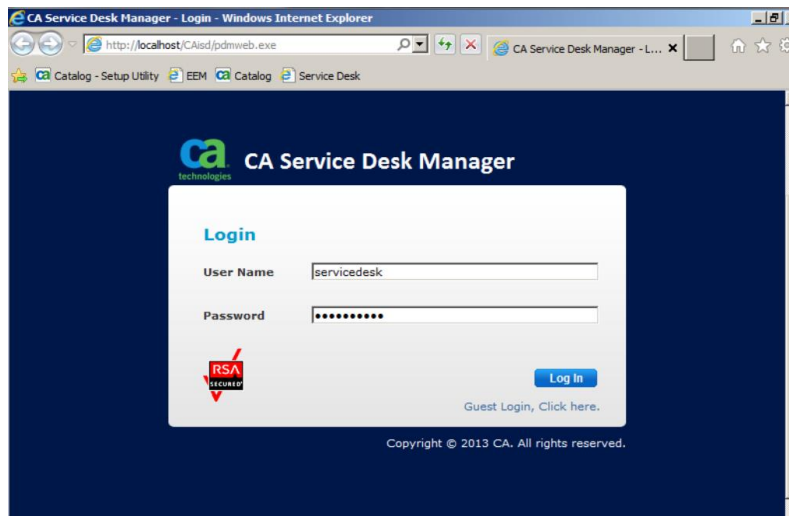


Figura 11 - Página inicial da plataforma CA SDM.

Após serem inseridas as credenciais (neste caso foi utilizado o *user* 'servicedesk', que é criado quando o software é instalado), e clicar em 'Log In', ir-se-á aceder à plataforma, que possui o aspeto exposto na seguinte figura.

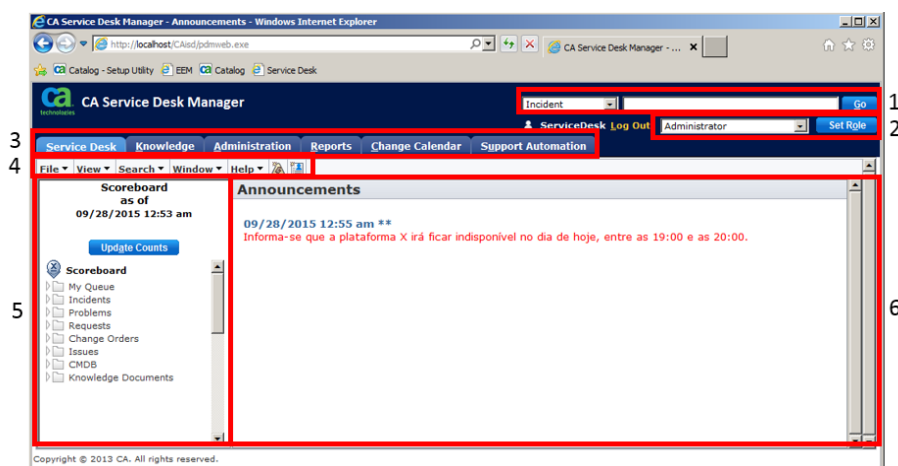


Figura 12 - Main Page da plataforma CA SDM (secções).

- 1) Zona onde se pode fazer uma pesquisa de *incidents*, *changes*, *request*, entre outros;
- 2) Local onde poderá escolher-se o perfil a ser utilizado na plataforma, tendo neste caso sido utilizado o perfil de *administrator*;
- 3) Diferentes *tabs* de operação da plataforma do CA SDM (as *tabs* podem variar consoante a forma da instalação, podendo por exemplo um cliente não estar interessado no módulo “*Support Automation*”);
- 4) A existência desta secção depende do módulo que está a ser utilizado. Neste caso do módulo de “*Service Desk*” obtêm-se os atalhos para a criação e pesquisa de *incidents*, *changes*, *problems*, *contacts* assim como algumas configurações de preferências da janela;
- 5) Esta secção vai variar consoante o módulo que esteja selecionado. No caso no módulo do “*Service Desk*” será apresentada uma listagem dos *tickets* que nos estão atribuídos (campo *assignee*) ou ao nosso grupo. A *scoreboard* pode ser personalizável, sendo possível criar novos nós e atribuir *queries* específicas. No módulo “*Administration*” será apresentado uma lista de todas as opções que podem ser configuradas da plataforma do CA SDM (8).

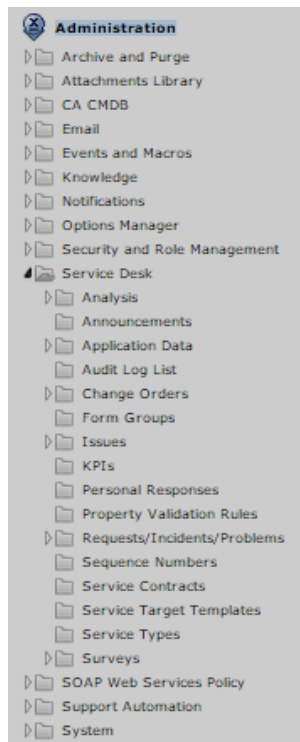


Figura 13 - Opções de configuração da plataforma CA SDM.

O *CA Service Desk Manager* é composto principalmente pelos seguintes módulos (13):

- **Service Desk;**
- **Knowledge;**
- **Support Automation;**
- **CMDB;**
- **Reporting** (basicamente faz um reencaminhamento para a plataforma de CABI BOXI);
- **Administration** (é aqui onde se procede a grande parte das configurações da plataforma).

5.3.1 Service Desk

Este é o módulo *core* da plataforma. Aqui é permitido (dependendo das configurações e permissões dadas):

- Criação e pesquisa de *requests, incidents, problems, change orders, issues;*
- Criação e pesquisa de *contacts, organizations, CI's, group, location, sites;*
- Personalização da *scoreboard;*

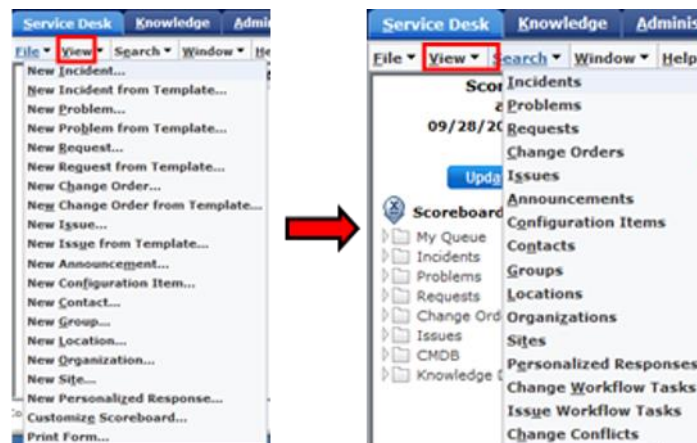


Figura 14 - Ações que podem ser efetuadas no módulo "Service Desk".

Exemplo de criação de incidente

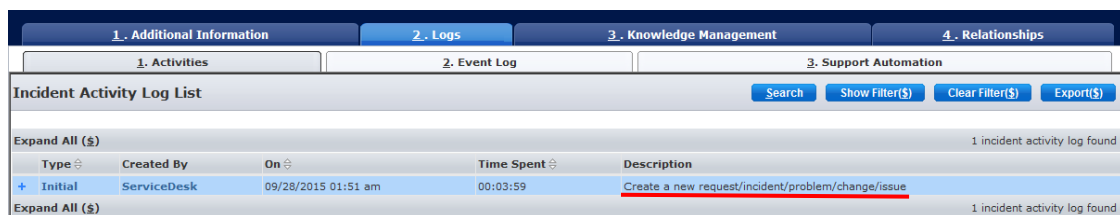
Se o utilizador depara com um problema ao abrir, por exemplo o Microsoft Word, e desconhece qual possa ser a solução, seria criado no CA SDM um *incidente*. Isto porque este problema apresenta algum impacto, dado impossibilitar o utilizador de efetuar as suas funções. Ao aceder à plataforma, deverá clicar-se em "File → New Incident". Surgirá então uma janela semelhante à presente na seguinte figura:

Figura 15 - Demonstração do processo de criação de um Incidente.

Após surgir a janela, poderá preencher-se os respetivos campos (dependendo das configurações e parametrizações que estejam implementadas) relativos a:

- **Requester e Affected End User** - para indicação de quem está a reportar o problema, assim como identificar quem o possui (o *end user* pode não ter acesso à plataforma);
- **Incident Area** - campo muito importante, pois é nele que está definido o grupo que irá ficar responsável pela sua resolução (ao preencher-se o campo da *IA*, o campo '*Group*' será auto preenchido). Normalmente é também associada à *IA* o *SLA*, que como visto anteriormente, é a definição do tempo máximo para que aquele pedido seja resolvido;
- **Urgency e Impact** - estes campos servem para definir uma urgência e um impacto. Os valores inseridos nestes campos vão ser utilizados para se calcular a prioridade de resolução (em que a prioridade=urgência x impacto). Caso estes campos não sejam preenchidos, existe uma prioridade dada por omissão (neste caso foi de nível 3).
- **Summary e Description** - Estes campos são usados simplesmente para se descrever em que consiste o incidente, de uma forma simples e breve.

Depois de preenchidos os campos realizar-se-á o “*Submit*” do incidente:



The screenshot shows a web interface for 'Incident Activity Log List'. At the top, there are four tabs: '1. Additional Information', '2. Logs', '3. Knowledge Management', and '4. Relationships'. Below these are three sub-tabs: '1. Activities', '2. Event Log', and '3. Support Automation'. The main content area has a search bar and buttons for 'Search', 'Show Filter(\$)', 'Clear Filter(\$)', and 'Export(\$)'. Below this is a table with columns: 'Type', 'Created By', 'On', 'Time Spent', and 'Description'. The table contains one entry: 'Initial' created by 'ServiceDesk' on '09/28/2015 01:51 am' with a time spent of '00:03:59'. The description is 'Create a new request/incident/problem/change/issue', which is underlined in red. There are 'Expand All (\$)' buttons on both sides of the table.

Type	Created By	On	Time Spent	Description
Initial	ServiceDesk	09/28/2015 01:51 am	00:03:59	Create a new request/incident/problem/change/issue

Figura 16 - Incidente após a sua criação.

Ao ser submetido, todos os elementos pertencentes à equipa de resolução (neste caso o grupo Suporte Informática) irão receber uma notificação (normalmente via *email*) a informar que chegou um novo incidente (9).

Para aceder a esse incidente bastará clicar no *link* que está contido no *email* direcionando para a plataforma CA SDM, ou então poderão fazê-lo acedendo à sua *scoreboard*, clicando por exemplo em “*My Groups* → *Incidents*”:

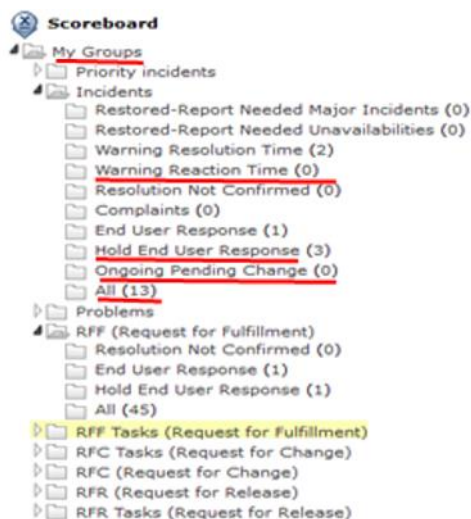


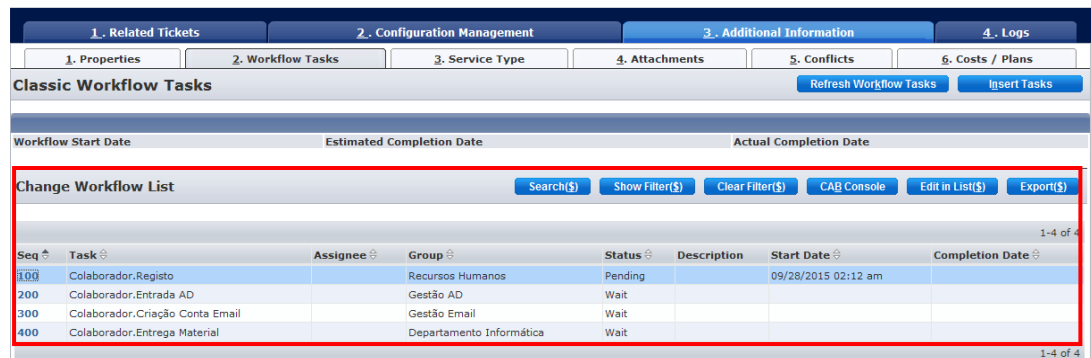
Figura 17 - Representação da secção da scoreboard.

Exemplo de criação de change order

Considerando-se o cenário da entrada de um novo colaborador numa empresa, deverá realizar-se a criação de um *change order*. Isto porque estamos perante um processo, ou seja, de um procedimento constituído por vários passos, onde poderão estar incluídos vários departamentos/equipas e atividades. Para tal deverá, na plataforma de CA SDM, clicar-se em “File → New Change Order”, surgindo então a seguinte janela, muito semelhante à janela de criação de “Incident”:

Figura 18 - Demonstração do processo de criação de um Change Order.

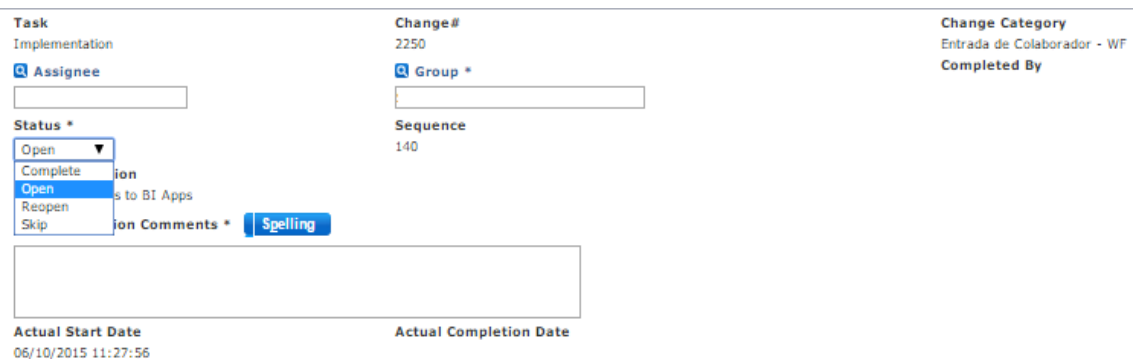
Após o preenchimento do formulário e de se proceder à sua submissão, o *ticket* será criado, mas desta vez poderemos encontrar uma diferença visualizada na seguinte figura.



Seq	Task	Assignee	Group	Status	Description	Start Date	Completion Date
100	Colaborador.Registo		Recursos Humanos	Pending		09/28/2015 02:12 am	
200	Colaborador.Entrada AD		Gestão AD	Wait			
300	Colaborador.Criação Conta Email		Gestão Email	Wait			
400	Colaborador.Entrega Material		Departamento Informática	Wait			

Figura 19 - Visualização do *workflow* associado ao change order criado.

Ao clicar-se na *tab* “Adicional Information” → “2. Workflow Tasks”, pode-se verificar que por norma existe um *workflow* associado a esse change (neste caso ao ter-se escolhido a categoria *Entrada.Colaborador*). Esse *workflow* é constituído pelo conjunto de *tasks* necessárias para que esse procedimento seja concluído de forma rápida, eficaz, e acima de tudo organizada. Isto porque poderão existir situações em que para se proceder à execução de uma determinada *task* seja necessária receber informação proveniente de *tasks* anteriores (9 e 11).



Task: Implementation
 Change#: 2250
 Change Category: Entrada de Colaborador - WF
 Completed By: []
 Assignee: []
 Group: []
 Status: Open (dropdown menu open with options: Open, Complete, Open, Reopen, Skip)
 Sequence: 140
 Spelling: []
 Actual Start Date: 06/10/2015 11:27:56
 Actual Completion Date: []

Figura 20 - Mudança de estados a nível de *task*.

Importante: As *Incident Areas*, *Categories*, *tasks*, estados de *tasks* e definição de transições são criadas e configuradas no módulo de “Administration” relativamente ao qual se irão apresentar alguns exemplos mais à frente.

Exemplo de pesquisa (e.g. incidentes)

A função de pesquisa, como em todas as aplicações, é de extrema importância. No âmbito desta plataforma é normalmente utilizada na procura de incidentes. Quando

estes surgem o analista poderá primeiro pesquisar por incidentes semelhantes que já tenham sido resolvidos (ou seja, já existe uma solução), apresentando assim uma maior capacidade de reação e resolução.

Esta funcionalidade é utilizada também para se visualizar os estados dos incidentes por categoria ou por grupo, quais os incidentes que se encontram num determinado estado, quantos *tickets* um determinado utilizador já abriu, entre outros.

Toda esta informação é de grande importância porque a junção destas pesquisas com a funcionalidade de *export* irá permitir a geração de um relatório (embora se utilize mais o CABI-BOXI para este efeito), com a extração de métricas que possam ser importantes para um coordenador de equipa ou para a empresa.

Por exemplo, verificando-se que existe um elevado número de *tickets* que estão abertos há mais de um determinado período de tempo (podendo mesmo existir violações de *SLA*) por uma certa equipa, seria importante a contratação de mais recursos para essa equipa, fornecer algum tipo de formação, entre outros. Desta forma, este tipo de informação pode revelar-se importante no âmbito de empresas de grande nível de maturidade (9).

Na seguinte figura pode-se visualizar um exemplo referente à pesquisa de todos os *tickets* abertos por um utilizador (e.g. Veloso, Marco):

The screenshot displays the 'Change Order Search' and 'Change Order List' interface. The search filters are set to Requester Name: Veloso, Marco. The list shows two tickets:

Change Order #	Summary	Priority	Category	Status	Assigned To	Change Type
+ 20	Mudar configuração de servidor	None	Change.IT.Server.Configuration	RFC		
+ 19	Entrada de novo colab	None	Colaborador.Entrada	RFC		

Figura 21 - Demonstração do método de pesquisa de Incidentes.

5.3.2 Knowledge Base

A *Knowledge Base* consiste num repositório documental que fortalece a ligação entre o departamento IT e o negócio, suportando-o com informação precisa, fidedigna e consistente, quer para os *end users*, quer para os analistas.

Por exemplo, um analista ao receber um incidente irá proceder à sua análise, elaborar um diagnóstico de avaliação das possíveis causas, podendo proceder à sua resolução ou então à criação de um *workaround*. Neste último caso poderá proceder à criação da documentação e publicá-la na *KB*.

Desta forma, caso este incidente volte a surgir no futuro (até mesmo para outro analista) bastará proceder a uma simples pesquisa (utilizando por exemplo *keywords*) pela descrição do problema, obtendo-se um documento que explica a resolução do mesmo.

5.3.3 Support Automation

Esta funcionalidade permite aos analistas resolverem incidentes e problemas de forma rápida e eficiente, fornecendo ferramentas e técnicas para identificar, diagnosticar e resolver as dificuldades antes que estas prejudiquem processos de negócio vitais. Normalmente é feito através da utilização de um *chat*, sessões remotas, fazendo uma gestão de incidentes e de problemas, recorrendo à utilização de comandos/scripts, de *screenshots*, *session logs*, entre outros.

5.3.4 Reporting

Este módulo recorre à utilização do *CA Business Intelligence / Business Objects XI* (CA BI / BOXI), que é uma plataforma que gera relatórios baseados em parâmetros previamente requeridos pelo cliente, normalmente utilizando *Key Performance Indicators* (KPI) de forma a ajudar o cliente na tomada de decisões de negócio. Mais à frente neste documento será descrito pormenorizadamente esta ferramenta.

5.3.5 Configuration and Management Data Base (CMDB)

A *Configuration and Management Data Base* (CMDB) é a base de dados mandatária que vai conter toda a informação utilizada pelo *CA Service Desk* (mas não só), como será descrito posteriormente neste relatório.

Normalmente as tabelas usam a seguinte convenção: *ca_ <table name>*. Quando o CA SDM possui informação complementar sobre os registos destas tabelas, esta vai ser guardada numa tabela obedecendo à convenção *usp_ <table name>*.

```
select *
from ca_contact
where userid like 'duarte'
```

contact_uid	middle_name	alias	last_name	first_name	pt_phone_number	email_address
0x8E48B9908C188D4EBB1E337686623CBF	Tiago	mestre	Sousa	Duarte	914125458	duartesousa@esomedomain.com

```
select *
from usp_contact
where contact_uid=0x8E48B9908C188D4EBB1E337686623CBF
```

contact_uid	c_acctyp_id	c_email_service	c_cm_id1	c_cm_id2	c_cm_id3	c_cm_id4	c_ws_id1
0x8E48B9908C188D4EBB1E337686623CBF	10002	NULL	1800	1800	1800	1800	NULL

Figura 22 - Diferença de resultados entre as tabelas *ca_contact* e *usp_contact*.

O CA SDM também possui as suas próprias tabelas na CMDb. Por exemplo, a informação relativa a *requests, incidents e problems* são guardadas na tabela *call_req*.

É importante que a equipa de *DBA* garanta que a *mdb* e todas as conexões ODBC não possam permissões de escrita direta. Estas deverão ser realizadas usando a camada de objetos/lógica, sendo apenas permitidas ações de leitura (para efeitos de consulta e/ou extração de relatórios) (18).

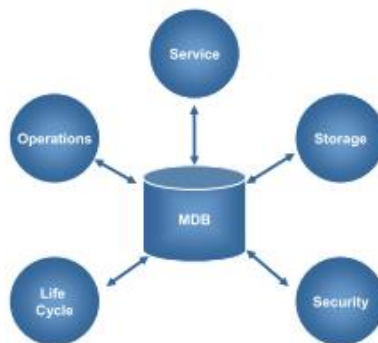


Figura 23 - Relação entre a *MDB* e as diferentes áreas. Retirado de (18).

5.3.6 Administration

Neste módulo será possível realizar grande parte das configurações da plataforma CA SDM:

- Criação de *IA's*, categorias de *change, tasks*, transição de estados, criação de *SLA's, workshifts*, contratos de serviços, *data partitions, access types* e *roles*;
- Configuração da arquitetura das plataformas;
- Criação de *Events* e *Macros*;
- Criação de métodos de notificação, *templates* e respetivas condições;
- Criação de *KPI's*;
- Configurações de alguns comportamentos OOTB da plataforma.

6 CA Service Catalog (CA SC)

A plataforma de *CA Service Catalog*, também denominada de “*self-service platform*”, foi uma ferramenta criada para oferecer alguma abstração ao utilizador relativamente à plataforma de CA SDM.

De notar que nem todos os utilizadores detêm conhecimentos informáticos, ou mesmo a experiência mínima, necessários para interagir com a plataforma de CA SDM. A sua má utilização pode conduzir à inconsistência de dados, a *tickets* mal categorizados, com conseqüente extração de métricas não correspondentes à realidade da organização.

Poderão surgir questões do tipo: deverá ser criado um incidente? Um *change order*? Que tipo de *change order* (RFF, RFR, RFC)? Que *IA* deverá ser utilizada? Que categoria de *change*? Quando o *ticket* está no estado x, o que significará? Todas estas questões embora pareçam simples, poderão dificultar o processo de amadurecimento.

Pelas razões que foram mencionadas, as organizações optam pela implementação do CA SC para que seja utilizada pelos *end users*, enquanto os analistas utilizarão o CA SDM.

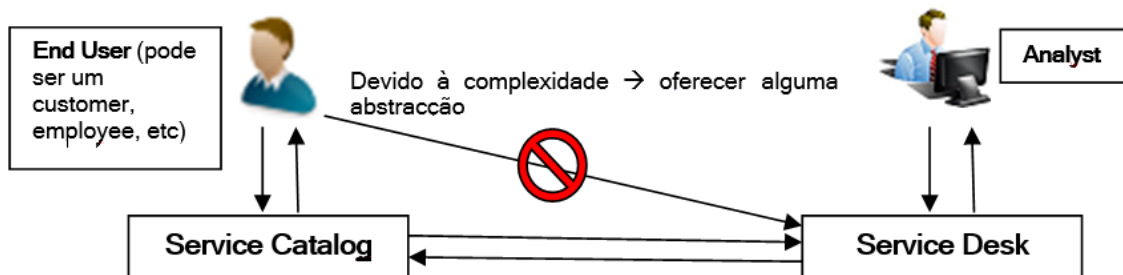


Figura 24 - Esquema explicativo da necessidade do Service Catalog.

Nota: É importante referir que neste tipo de abordagem, o SC não consiste numa plataforma independente, mas sim uma camada de abstração que vai “envolver” o SDM.

6.1 Apresentação do CA Service Catalog

O CA SC é composto fundamentalmente por dois grandes perfis:

- Perfil de administração;
- Perfil de utilizador;

No perfil de administração, irão ser efetuadas todas as configurações da plataforma, assim como a gestão de todos os serviços, opções e *forms*. Estes conceitos serão descritos com maior detalhe no capítulo seguinte.

No perfil de utilizador, irá ser apresentado um conjunto de serviços e opções, normalmente organizados por temas. Será neste capítulo que irá ser realizada uma breve apresentação deste perfil (19).

Ao entrar-se na plataforma, irá encontrar-se uma interface simples e intuitiva. Se um utilizador pretender efetuar um pedido relativamente aos acessos às caixas automáticas, poderá fazê-lo facilmente, clicando nas várias opções de forma a especificar o tema até encontrar a opção correta, como é apresentado na sequência de imagens seguintes:

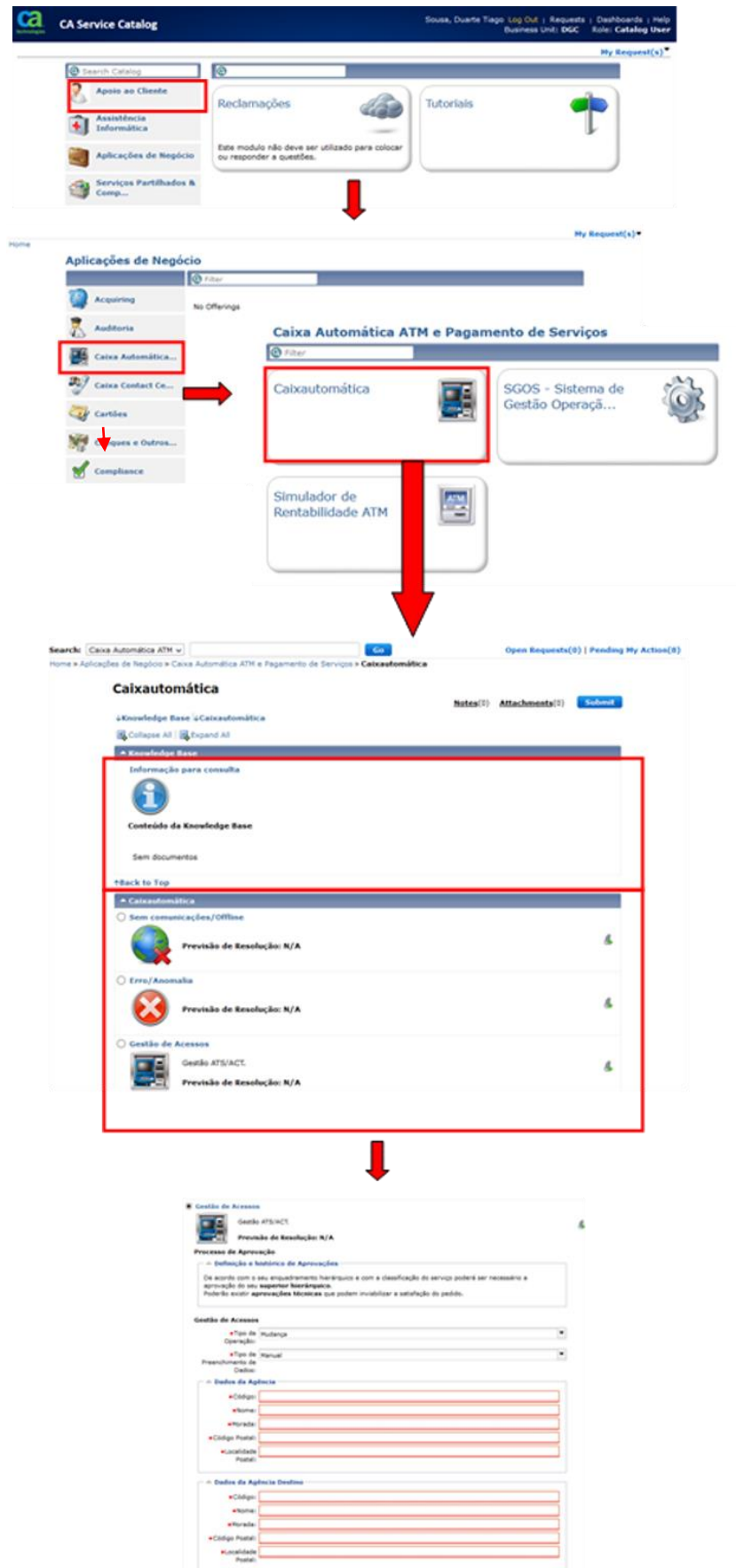


Figura 27 - Demonstração da navegação na árvore de serviços do Service Catalog.

Depois de preenchido o *form* será necessário fazer-se o *submit*. Após esta submissão é possível visualizar o estado do mesmo, ou então efetuar tarefas como editar e cancelar, entre outros. Para tal dever-se-á clicar em “*My Requests* → *Find Request*”. De seguida será apresentado uma listagem de todos os pedidos realizados pelo utilizador. Após selecionar o pedido que é pretendido, irá visualizar-se uma janela como a descrita na seguinte figura.

The screenshot displays the 'Requested Services' page in a Service Catalog. It features a table with columns for 'Service', 'Item Status', and 'Action'. Below the table, there is a section for 'Vidros' with a description: 'Reporte de anomalia ocorrida com os vidros (biombos, divisórias, janelas interiores e exteriores)'. A 'Previsão de Resolução: 5 dias' is also shown. To the right, a 'General Information' panel provides details such as Name, ID, Requested For, Date Created, Date Required, Requested By, Priority, Last Modified, and Current Status. At the bottom, a section titled 'Vidros' contains a form for 'Informação sobre a ocorrência', including a dropdown for 'Solicita-se' and a text area for 'Solicitação'.

Figura 25 - Detalhes do pedido de Service Catalog (secções).

Nesta página, o utilizador poderá visualizar os detalhes do seu pedido. Na secção 3 poderá visualizar o que foi preenchido no *form*. Na secção 2 poderá observar em que estado se encontra o seu pedido, neste caso o estado está em “*USD Request Opened*”, o que significa que o *ticket* já foi criado na plataforma de CA SDM. Na secção 1, existe a possibilidade de se anexarem ficheiros ao pedido, ou de se ler / escrever algumas notas, sendo estas utilizadas para comunicar com o analista (já o analista irá utilizar a atividade ‘*Log Comment*’ na plataforma de CA SDM para fazer essa comunicação).

6.2 Elementos fundamentais do CA Service Catalog

Como foi referido anteriormente, esta plataforma foi criada com o intuito de oferecer alguma abstração ao utilizador na ação de abertura de um *ticket*. Como se pode verificar nas imagens anteriores, aquele não necessitou de decidir da abertura de um incidente ou de um *change order*, assim como de ter selecionar a *IA* ou a categoria mais correta. Simplesmente apenas navegou numa simples árvore organizada por temas até encontrar a opção que mais se adequava ao que pretendia. Após selecionar essa opção, bastou preencher um formulário para descrever o problema, e por fim fazer o *submit*.

É importante ter em conta o significado de alguns elementos/conceitos que o SC utiliza, para que desta forma seja possível fazer uma configuração correta e otimizada da plataforma. Para tal o *Service Catalog* utiliza como base a seguinte estrutura:

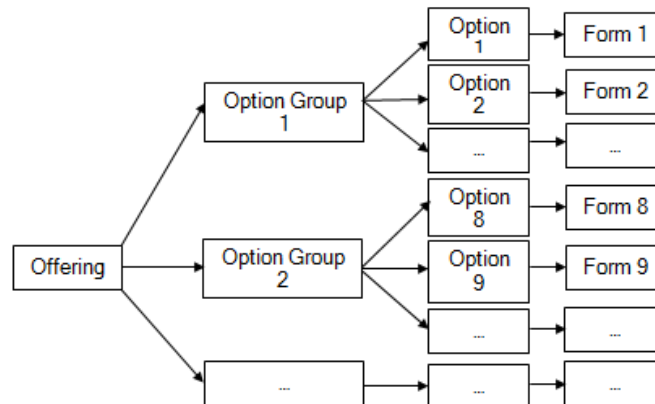


Figura 26 - Esquema dos elementos do *Service Catalog*.

Como se pode verificar através do esquema anterior, a plataforma trabalha com os elementos: *offerings*, *option group's*, *options* e *forms*.

Considere-se o seguinte cenário de forma a clarificar melhor a diferença entre os vários elementos mencionados: um utilizador encontra-se com problemas em abrir o Microsoft Excel e para tal vai utilizar a plataforma *Service Catalog*.

Para proceder à abertura de um *ticket* sobre esse problema, no ecrã inicial iria clicar em “Assistência Informática” (que consiste numa pasta que contem todas as *offerings* sobre esse tema), de seguida seleccionava a *offering* “Microsoft Office”, e após essa seleção iria-lhe surgir várias *options groups*, como por exemplo: “Microsoft Word”, “Microsoft Excel”, “Microsoft Power Point”, entre outros.

Por fim, ao clicar em “Microsoft Excel” aparecer-lhe-iam as várias *options*, como por exemplo: “Problema com Licenciamento”, “Instalação ou Remoção do Excel”, “Excel não inicia”, entre outros. Em *background*, cada uma destas *options* tem um “*External ID*” que irá corresponder a uma *IA* ou uma categoria de *Change*, que será utilizada para a criação do ticket na plataforma de CA SDM (19).

Após escolher-se a opção “Excel não inicia”, basta preencher o *form* e submetê-lo. A nível de estrutura teríamos algo como o esquema apresentado na seguinte figura.

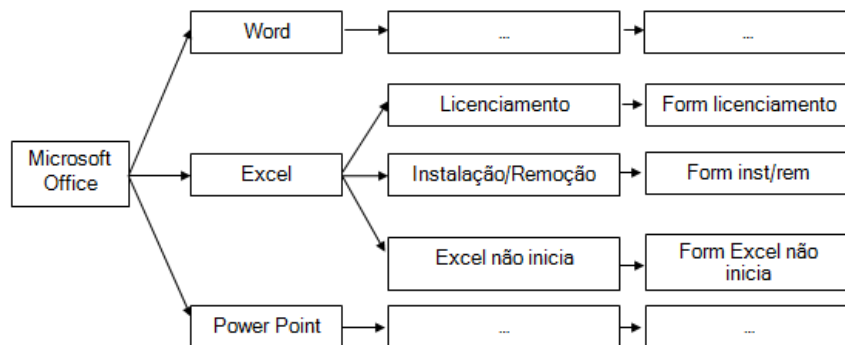


Figura 27 - Esquema dos elementos do Service Catalog (aplicado a exemplo).

Nota: Esta foi uma das várias formas de se poder organizar a árvore em offering, option group's e options. Visto que se tratam de conceitos com um significado um pouco abrangente a nível do negócio, consoante o seu entendimento e a necessidade do cliente existem várias maneiras de se proceder a esta organização.

6.2.1 Offerings

Como descrito anteriormente, uma offering consiste num serviço que é oferecido ao utilizador, contendo dentro dele uma listagem de várias opções de criação de incidentes ou *change orders*, sem que o utilizador tenha a necessidade de saber o tipo de *ticket* que está a ser aberto. Na plataforma de SC, para visualizarmos essas *offerings*, basta ir à tab “*Catalog (necessário perfil de administrador) → Service Offerings → Offerings*”. Na seguinte figura pode-se observar como é organizada uma árvore de *offerings*, que recorrendo à criação de pastas, permite o seu agrupamento por temas (19).

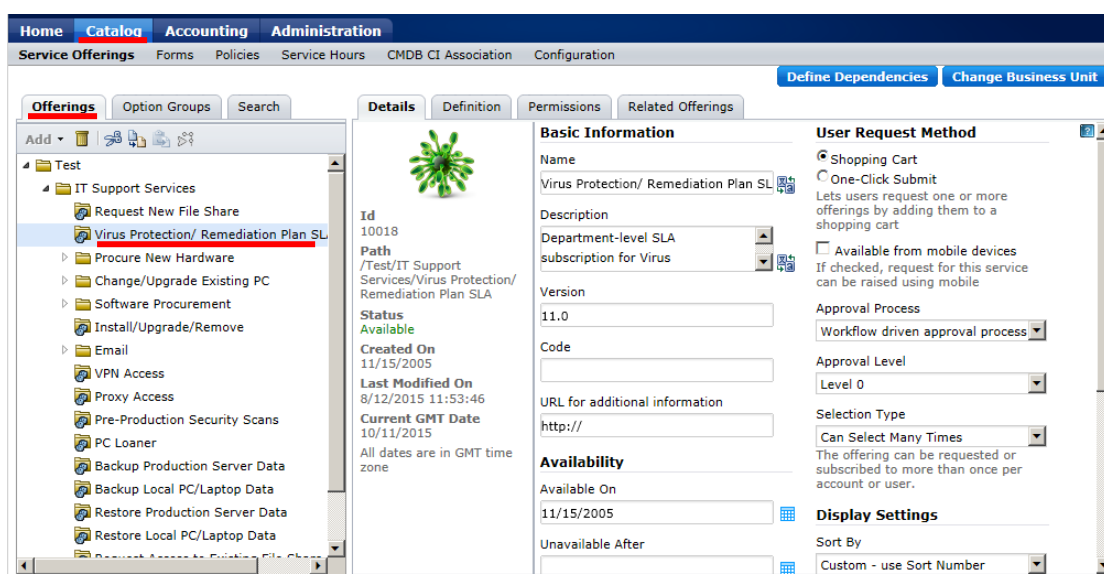


Figura 28 - Secção relativa à offering.

Para cada *offering* é permitido fazer:

- **Configuração de data de início de disponibilidade e uma data de fim** (caso se queira disponibilizar essa *offering* por um período de tempo);
- **Configurar o método de submissão**, podendo este ser através de “*shopping cart*” (cada pedido que for realizado pelo utilizador vai para o carrinho de compras e no fim poderá ser feito um “*check-out*” do que foi selecionado), ou então simplesmente através de um “*click and submit*” referente a cada opção, após preenchimento do *form*;
- **Método de aprovação**: pode ser efetuado através de *policy*, de *workflow* (caso exista mais do que um nível de aprovação), ou mesmo possibilitando submeter um pedido sem que este seja aprovado;
- **Tipo de selecção**: configurar se é permitido que esta *offering* seja requerida uma ou mais vezes;
- **Outras configurações...**

Ao clicar-se na *tab* “*Definition*” dessa *offering* será possível observar quais as *option groups*, assim como das *options* que estão associadas:

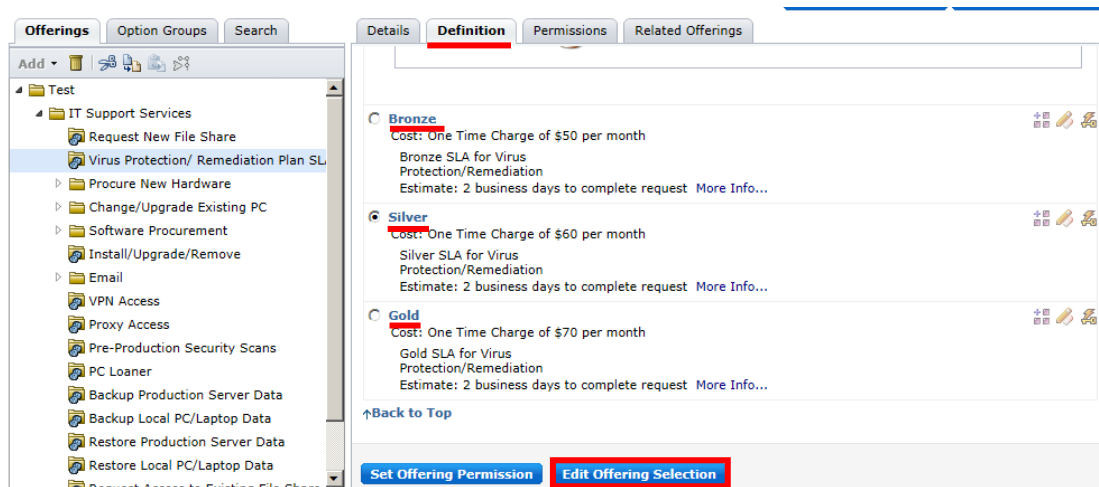


Figura 29 - Secção relativa à offering (Definition).

Caso se queira adicionar ou remover *options groups* a essa *offering*, bastará clicar em “*Edit Offering Selection*” e seleccionar qual ou quais se querará adicionar ou remover à *offering*.

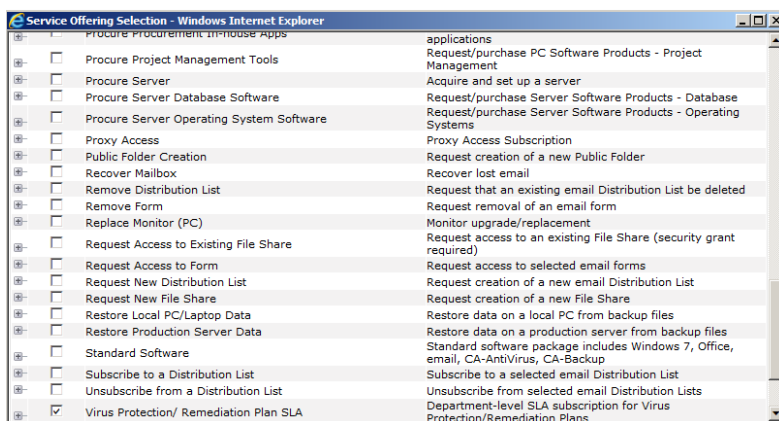


Figura 30 - Inserção/remoção de option groups da offering

6.2.2 Option groups

As *options groups* são aglomerados de *options* que são associados às *offerings*. Para visualizá-las na plataforma, deve-se ir, com perfil de administrador, a “Catalog → Service Offerings → Option Groups”:

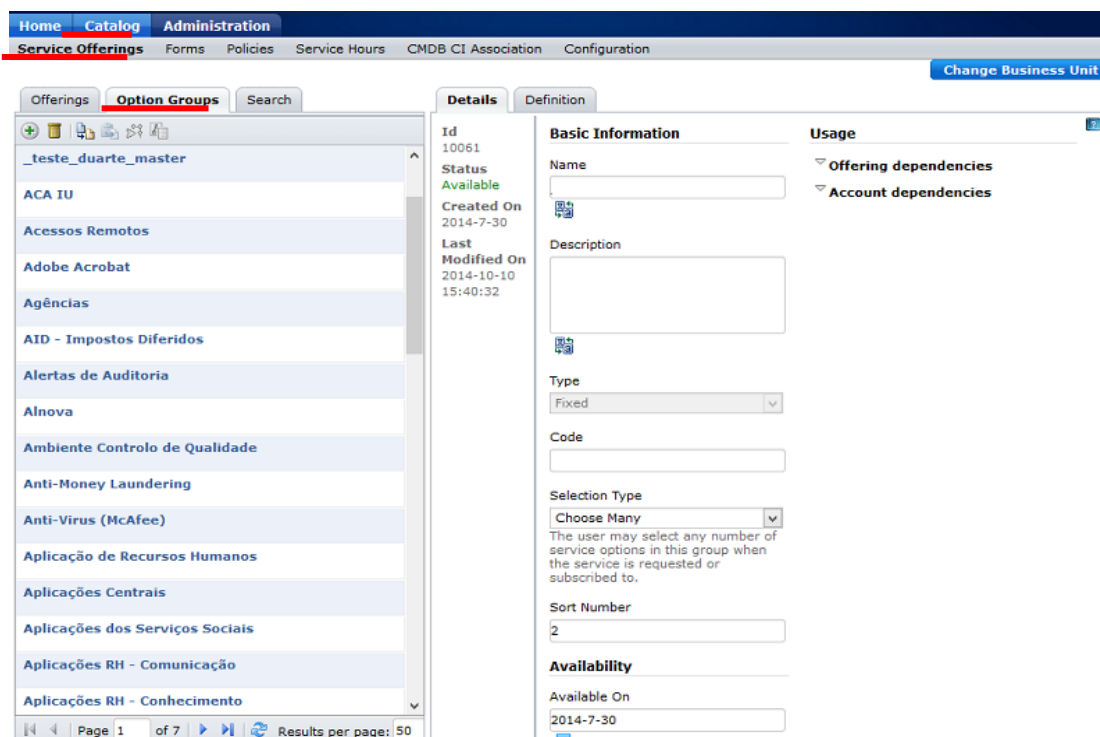


Figura 31 - Secção referente às options groups.

Depois de selecionada esta *tab*, é possível visualizar todas as *options groups* criadas. Para cada uma delas é possível: determinar um tipo de seleção (dentro da *offering*, escolher a possibilidade de permitir a seleção de uma ou mais *option groups*). Adicionalmente, à semelhança da *offering*, definir uma data de início de fim de disponibilidade (19).

Ao clicar na “Definition” da *option group* será possível visualizar quais as *options* associadas à mesma, assim como o conteúdo de cada uma delas:

The screenshot displays the 'Definition' configuration page for an option group. At the top, there are tabs for 'Details' and 'Definition', and buttons for 'Add Option' (highlighted in red) and 'Organize Options'. The main content is organized into two primary sections: 'Acessos Não Produção' and 'Acessos Produção'. Each section contains a brief description of the access type, a 'Processo de Aprovação' (Approval Process) section with a text box explaining the approval flow, and specific configuration fields. The 'Acessos Não Produção' section includes dropdown menus for 'Ambiente' and 'Operação'. The 'Acessos Produção' section includes input fields for 'User ID do Responsável' and 'Nome do Responsável', each with a 'Utilizador(es)' label, and a 'Tipo de' dropdown at the bottom.

Figura 32 - Secção referente às options groups (Definition).

Caso se pretenda adicionar mais uma *option*, deverá clicar-se em “Add Option”.

6.2.3 Options

Após clicar-se na opção “Add Option”, irá surgir uma janela semelhante à presente na seguinte figura (19).

Basic Information

Name: Mouse

Description: Corded 2 button mouse with PS2 connector

Attachment Mandatory
Note: This option will take effect only when the Catalog configuration 'Allow Attachments at Service Option Level' is activated.

Reservation

Reservation Service Option

Reservation Display Text

Operation: Reserve (selected), Extend, Return, Modify

Reservation Ready Status: Backordered

Reservation Failure Status: Backordered

Reservation System

Form

Form: Mouse Form (selected), Add

CA BSI Contract

Add

Categorization

Category: Hardware

Category Class: IT

Category Subclass: Storage

Keywords

External ID: CHGSC98457

Track as an Asset

Information Service Option Only

Cost & Pricing

Rate: Cost, Add

Adjustment: Add

Usage Based Price: Add

Additional Elements

Text: Estimate: 1 week to compl... (selected), (No Text), Add

Numeric: Add

Figura 33 - Secção referente à option.

Nesta página poderá então ser definido a opção dando-lhe: um nome, descrição, categoria, classe, um *external id* (que como referido anteriormente, é através deste campo que o CA SDM distingue se vai criar um incidente ou um *change*, assim como a *IA* ou categoria correta), e por fim atribuir a essa opção o respetivo *form*.

6.2.4 Forms

Os *forms* são os elementos que vão ser associados a cada opção da *offering*. Estes são criados e utilizados com o objetivo de se obter do utilizador a maior informação possível, para que desta forma se consiga compreender no que consiste o seu pedido ou problema. Para visualizar os *forms* deverá ir-se, com perfil de *administrator*, a “*Catalog* → *Forms*”:

The screenshot shows the 'Administration' section of a software interface. The left sidebar contains a tree view under 'Components' with 'Forms' expanded. The main area displays a form titled 'Dados do Responsável' with several fields: 'User ID do Responsável', 'Nome do Responsável', 'Tipo de Operação', 'Nome Completo', 'Empresa', 'Orgão de Estrutura', 'Órgão de Estrutura', 'User ID', and 'Acesso a dados de ambiente'. There are also checkboxes for 'Remover todos os acessos', 'TSO', and 'Ambiente Controlo de Qualidade'. The right sidebar shows a table of properties for the form.

Name	Value
_id	form1
Business Unit	GrupoCGD
CSS Class	
Created	2014-11-26 17:52:15
Created By	Sousa, Duarte Tiag...
Form Type	request
Label Align	right
Label Width	110
Modified By	Sousa, Duarte Tiag...
Modified On	2015-7-9 18:12:49
Name	form1
onLoad	ca_fd.js.initFields();
onSubmit	ca_fd.js.validateFiel...

Figura 34 - Secção referente ao form.

Após a criação de um elemento deste tipo, bastará expandir a pasta “*Components*”. Nesta pasta irão encontrar-se as várias possibilidades de componentes de um *form*, como por exemplo: *text boxes*, *check boxes*, *labels*, *text areas*, *select boxes*, *dual lists*, *slider*. Para se construir um *form*, será necessário arrastar as componentes desejadas para dentro do elemento que foi criado, atribuindo-lhes os valores que se pretender (19).

Adicionalmente também é possível criar funções em *javascript*, recorrendo a rotinas do produto, para se implementar comportamentos dinâmicos e validações. Estas são associadas aos diferentes eventos da componente em questão (e.g. *onLoad*, *onSubmit*, *onClick*). Para tal bastará clicar no botão “*Script*” descrito na seguinte figura.

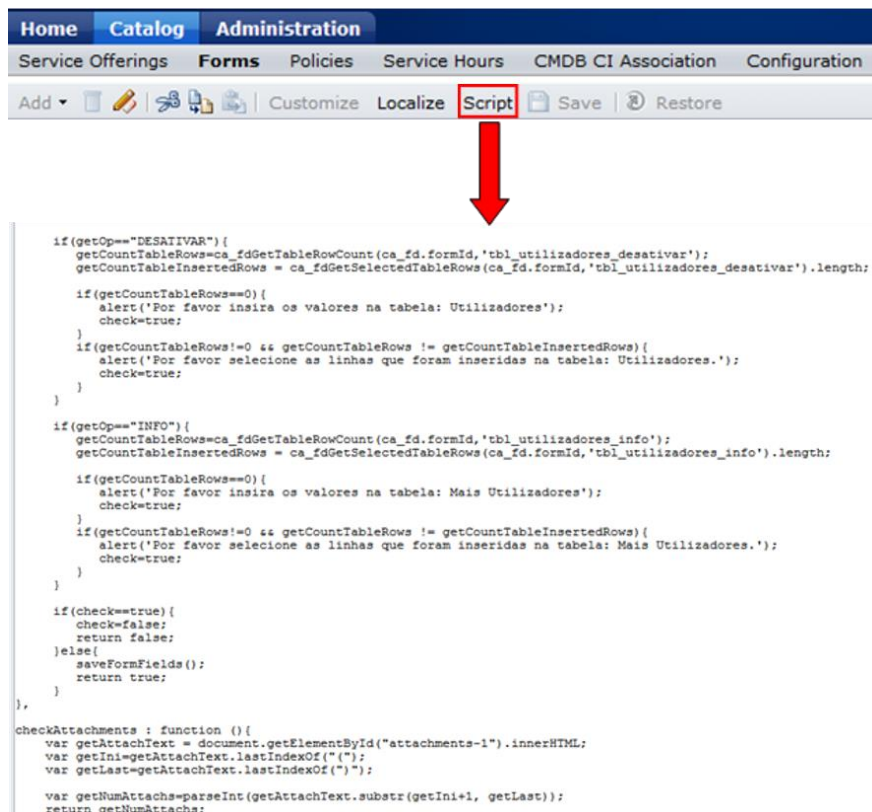


Figura 35 - Secção do script do form.

6.2.5 Report builders

Quando no *form* existem componentes como *select boxes* e *dual lists*, necessitando estas de ser preenchidas com valores, é necessário por vezes obter esses valores à CMDB (por exemplo se existir uma *select box* denominada de contactos, onde que o objetivo é seleccionar um dos contactos que estão na tabela *ca_contact*), é então necessário recorrer-se às *report builders*, que consistem na criação de um data object que nos devolve o *output* necessário recorrendo a uma *query* à CMDB. Para tal basta ir, com perfil de *administrator*, a “Administration → Report Builder” (18 e 19):

The screenshot shows the 'Report Builder' interface. At the top, there are navigation tabs: Home, Catalog, and Administration. Under Administration, there are sub-tabs: Business Units, Users, Report Builder (selected), Dashboard Builder, Events-Rules-Actions, Tools, and Configuration. A 'Menu' sidebar on the left lists 'Data Objects', 'Data Views', and 'Layouts'. The main area is titled 'Data Objects' and has buttons for 'InfoView', 'Create Data Object', and 'Refresh'. Below this, there are buttons for 'Set Folders', 'Set Permissions', and 'Delete Items'. The interface shows a 'Folder View | List View' toggle. A table lists various data objects with columns for Name, Comment, Last Updated, and Actions.

<input type="checkbox"/>	Name	Comment	Last Updated	Actions
<input type="checkbox"/>	Default			
<input type="checkbox"/>	Catalog			
<input type="checkbox"/>	CGD			
<input type="checkbox"/>	CGD View Reports			
<input type="checkbox"/>	Data Mediation			
<input type="checkbox"/>	GrupoCGD			
<input type="checkbox"/>	GetSC_MyOpenRequests		2015-2-18 12:37:26	
<input type="checkbox"/>	GetSDM_Approval_Rule		2014-10-30 12:17:01	
<input type="checkbox"/>	GetSDM_Artemis_Skill_Externo		2014-11-4 14:45:36	
<input type="checkbox"/>	GetSDM_Artemis_Skill_Interno		2014-11-4 14:46:02	
<input type="checkbox"/>	GetSDM_Catalogo_TipoRecurso		2014-9-30 08:46:48	
<input type="checkbox"/>	GetSDM_CatalogoAmbiente		2014-9-30 08:46:56	
<input type="checkbox"/>	GetSDM_CatalogoAmbiente3		2014-9-30 08:47:10	
<input type="checkbox"/>	GetSDM_CatalogoTipoOperacao		2014-9-30 08:46:16	
<input type="checkbox"/>	GetSDM_CatalogoTipoOperacao2		2014-9-30 08:47:27	
<input type="checkbox"/>	GetSDM_ChgCtg		2015-9-29 16:23:47	
<input type="checkbox"/>	GetSDM_ChgCtgPrp2		2015-9-23 16:54:36	
<input type="checkbox"/>	GetSDM_Ctg		2015-9-29 16:09:43	
<input type="checkbox"/>	GetSDM_CtgPrp2		2015-9-16 14:11:29	
<input type="checkbox"/>	GetSDM_Equip		2014-8-18 14:15:16	
<input type="checkbox"/>	GetSDM_Error		2014-6-30 15:08:17	
<input type="checkbox"/>	GetSDM_Grupos		2015-3-26 17:27:31	
<input type="checkbox"/>	GetSDM_HoldEndUserPED		2014-9-11 08:58:52	

Figura 36 - Listagem das report builders criadas.

De seguida será apresentada uma listagem de todas as *data objects* criadas e, caso queiramos criar uma nova, bastará clicar em “*Create Data Object*”:

The screenshot shows the configuration screen for a data object named 'GetSDM_Kbdoc'. The 'Action' is set to 'Permissions'. Under 'Data Object Properties', there is a 'Show Advanced...' checkbox. The 'Type' is set to 'Query'. The 'ID' is 'GetSDM-Kbdoc'. The 'Database' is 'mdb', the 'Table' is 'skeletons', and the 'Fields' are 'kbid,title,summary'. The 'Query' text area contains the following SQL: `select kb.id as kbid, kb.title as title, kb.summary as summary, ci.resource_name from skeletons kb LEFT OUTER JOIN ca_owned_resource ci ON kb.sd_asset_id = ci.own_resource_uid where ci.resource_name like N'%%%'ci_name%%'`. There are buttons for 'Manage Variables...' and 'Create Variable...'. A 'Query Builder' tool is also visible on the right.

Figura 37 - Construção de uma report builder.

Após criada a *data object* bastará adicionar o seu ID ao parâmetro *Plug-in* da *select box*.

7 CA Process Automation Manager (CA PAM)

O *Process Automation Manager* (PAM) consiste numa plataforma utilizada como intermediária na comunicação com outras plataformas. Por exemplo, quando se descreveu anteriormente a necessidade de implementação da plataforma de *Service Catalog*, e que esta comunicava com o CA SDM, na realidade esta comunicação existe devido ao PAM. Este tem como função receber os *outputs* das diferentes plataformas, podendo manipular o seu conteúdo e enviar para a plataforma destino.

Para possibilitar esta comunicação, o PAM recorre ao *Simple Object Access Protocol* (SOAP), através do qual irá receber as informações em envelopes (*soap envelopes*), proceder à leitura do seu conteúdo (ficheiro xml), e em seguida criar um novo envelope com a nova informação a ser enviada para a plataforma destino, através de *web services*.

Tendo em conta o cenário da comunicação entre o SC e o SDM, esta é realizada da seguinte forma:

1. Quanto o utilizador cria um pedido no catálogo, vai despoletar um evento que utiliza uma macro. Estes eventos e macros são criados na plataforma de catálogo e são configurados para esse efeito. Por sua vez essa macro vai invocar um processo PAM (irá ser descrito um exemplo de seguida), passando para o mesmo, através de um *soap envelope* via *web services*, a informação que foi inserida no *form*, assim como do *external id* que é fundamental para o CA SDM distinguir se é para ser criado um *change*, um incidente, entre outros;
2. Ao chegar ao PAM, quando este recebe a indicação de qual o processo que deve ser utilizado, vai proceder à leitura do envelope recebido e irá adaptar a informação contida ao seu processo. Por exemplo, na sua leitura poderá ter que retirar a informação e separá-la por variáveis de processo.
3. Um processo PAM consiste num *workflow* constituído por vários operadores, em que cada um deles tem uma determinada função. Desta forma, para este cenário deverá no respetivo processo encontrar-se um:
 - ✓ Operador responsável pela leitura de envelope recebido;
 - ✓ Operador para a passagem do conteúdo recebido a variáveis de ambiente;
 - ✓ Operador responsável por fazer *login* à plataforma de CA SDM;

- ✓ Operador para criação do incidente ou *change* na plataforma CA SDM, passando o conteúdo das variáveis de processo;
 - ✓ Operador responsável por fazer *logout* da plataforma de CA SDM.
4. Ao ser criado esse *ticket* na plataforma de CA SDM, será necessário fazer uma sincronização de estados do mesmo. Ao fazer essa alteração haverá novamente a invocação de processos PAM para esse efeito através de eventos e macros do SDM;
 5. E assim sucessivamente, tudo se sumarizando na invocação de processos PAM para que se possibilite esta comunicação entre as duas plataformas.

7.1 Apresentação do CA PAM

Do que foi descrito anteriormente poderemos concluir que o PAM tem elevada importância. Deste modo, iremos de seguida analisar em que consiste esta plataforma. Ao fazermos *login* na aplicação iremos visualizar uma página como a descrita na seguinte figura.



Figura 38 - Main Page da plataforma ITPAM.

Esta página será a que será apresentada após a entrada na aplicação. Na secção 1 estarão presentes as *tabs* do produto, sendo apenas mencionadas as que são mais importantes no presente contexto:

- **Tab Library.** aqui poder-se-ão encontrar todos os processos criados;

Nome	Tipo	Proprietário	Data de modificação	Reservado po
Processes	Pasta	padams	18/06/2014 19:19:31	
Change Management old	Processo	pamadmin	06/11/2014 23:43:56	
Create Change Order old	Processo	pamadmin	06/11/2014 23:43:55	
Create Request old	Processo	pamadmin	06/11/2014 23:43:52	pamadmin
Create Request to Modify KD old	Processo	pamadmin	18/06/2014 19:19:21	
Get Contact Info for User ID old	Processo	pamadmin	06/11/2014 23:43:59	
Get Object Properties old	Processo	pamadmin	06/11/2014 23:43:55	
Get Supervisor old	Processo	pamadmin	06/11/2014 23:43:59	
Order PC old	Processo	pamadmin	06/11/2014 23:43:51	
Problem Analysis old	Processo	pamadmin	06/11/2014 23:43:56	
Transfer Ticket old	Processo	pamadmin	06/11/2014 23:43:54	
Update Staus & Comment old	Processo	pamadmin	06/11/2014 23:43:49	

Figura 39 - Conteúdo da tab Library.

- **Tab Creator:** zona na qual se podem criar os processos;
- **Tab Operations:** aqui será apresentada uma listagem de todos os processos que foram executados ou estão a ser executados, mostrando o estado dos mesmos.

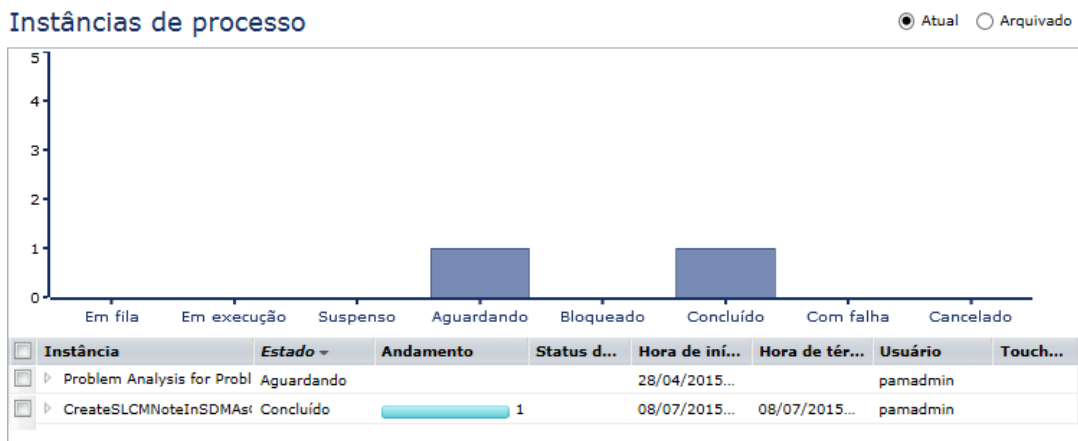


Figura 40 - Conteúdo da tab Operations.

Na secção 2 é disponibilizado um conjunto de ações básicas como criar um novo processo, visualização de tutoriais e de fóruns, entre outros. Na secção 3 será apresentada uma versão simplificada da *tab Operations*. Por fim, na secção 4 serão apresentados todos os processos que se encontram a correr naquele momento.

7.2 Exemplo de processo CA PAM

Na seguinte figura é apresentado um exemplo do processo de criação de um *request* na plataforma de CA SDM.

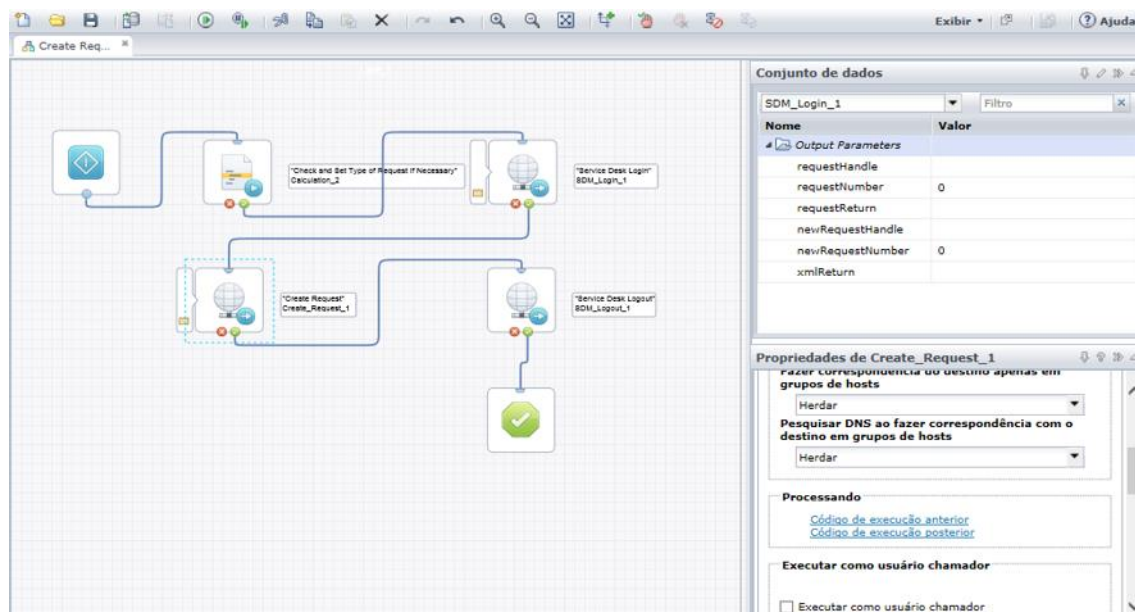


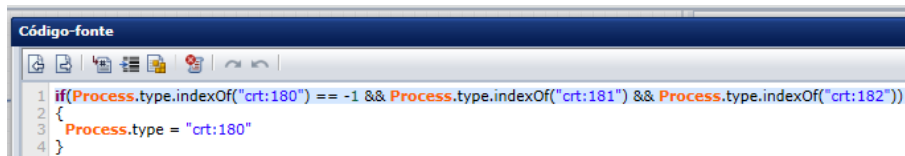
Figura 41 - Exemplo de processo PAM.

Como descrito anteriormente, um processo PAM consiste num conjunto de operadores, cada um com a sua própria função. Normalmente, grande parte dos operadores *core* já vêm criados no produto, mas existe a possibilidade de se criarem os nossos próprios operadores.

Um processo PAM consiste num *workflow*, e como se pode verificar neste caso, o processo começa sempre com o operador *start* (operador azul) visualizando-se de seguida uma sequência de operadores. De notar que alguns destes operadores necessitam do *output* de outros operadores para que dessa forma consigam executar sem problemas a sua tarefa. Dependendo do operador que está a ser utilizado, é possível fazer com que esse operador execute funções de javascript, assim como da implementação de pré e pós condições de execução.

Neste exemplo básico (criação de um *request* na plataforma CA SDM) podem-se observar os seguintes operadores:

- **Operador Start (azul):** é sempre necessário para dar início a um processo;
- **Operador "Check and Set Type of Request If Necessary":** que verifica o tipo de *request* a ser criado, atribuindo o *ID* desse tipo de *request* a uma variável de processo (executando um pequeno *javascript*):



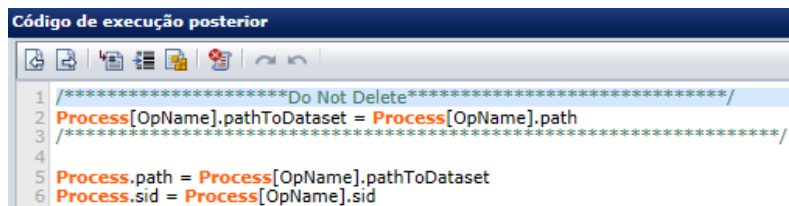
```

Código-fonte
1 if(Process.type.indexOf("crt:180") == -1 && Process.type.indexOf("crt:181") && Process.type.indexOf("crt:182"))
2 {
3   Process.type = "crt:180"
4 }

```

Figura 42 - Código js para verificação do tipo de request.

- **Operador "Service Desk Login":** este operador obtém a partir de um *dataset* a informação que necessita para se ligar à plataforma de SDM, gerando assim um *SID* através de pós execução:



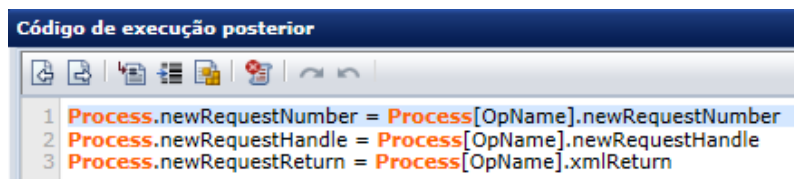
```

Código de execução posterior
1 /*****Do Not Delete*****/
2 Process[OpName].pathToDataset = Process[OpName].path
3 /*****Do Not Delete*****/
4
5 Process.path = Process[OpName].pathToDataset
6 Process.sid = Process[OpName].sid

```

Figura 43 - Código de pós-execução do operador de login.

- **Operador "Create Request":** este será o operador mais importante, pois é o responsável por criar o *ticket* propriamente dito na plataforma CA SDM através do que foi recebido do envelope SOAP (informação que se encontra em xml):



```

Código de execução posterior
1 Process.newRequestNumber = Process[OpName].newRequestNumber
2 Process.newRequestHandle = Process[OpName].newRequestHandle
3 Process.newRequestReturn = Process[OpName].xmlReturn

```

Figura 44 - Código de pós-execução do operador de criação do request.

- **Operador "Service Desk Logout":** à semelhança do operador do *login*, este operador será o responsável pelo *logout* da plataforma CA SDM, após criação do *request*.

8 CA Business Intelligence (CA BI / BOXI)

Esta plataforma tem como principal objetivo a extração de informação da CMDB sob forma de relatórios. A construção destes relatórios é realizada através da utilização de componentes que foram previamente configurados para esse efeito. Desta forma, consoante as métricas requeridas pelo cliente, coordenador de equipa ou mesmo de um diretor, bastará selecionar os respetivos elementos para realizar a criação do relatório.

Estes relatórios apresentam uma grande importância pois será através deles que existirá um controlo a nível do estado e qualidade de serviço implementado. Por exemplo, se os coordenadores de uma equipa pretenderem avaliar o rendimento de um determinado elemento da equipa, é necessário efetuar um pedido à equipa de BOXI para que seja feito um relatório que nos informe quais os *tickets* que estão associados àquele membro, assim como se estes são um incidente, um *change*, o seu estado, entre outros.

Em seguida será descrito como o relatório do exemplo anterior seria construído. Ao ser realizado o *login* na plataforma CABI/BOXI, será apresentada uma janela semelhante à presente na seguinte figura.

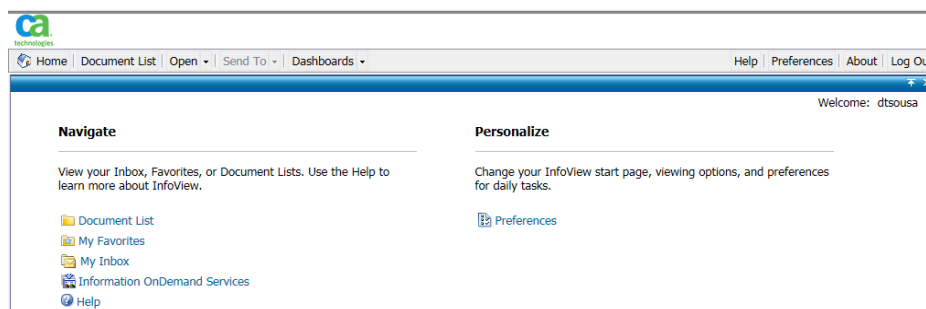


Figura 45 - Main page da plataforma CABI/BOXI.

De seguida, irá clicar-se em “*Document List* → *Public Folders* → *DEMO* → *New* → *Web Intelligence Document*”, e dar-se como nome a esse relatório “*dtsousa_myTickets*”:

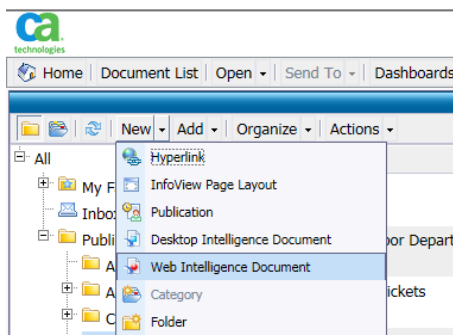


Figura 46 - Criação de novo relatório CABI.

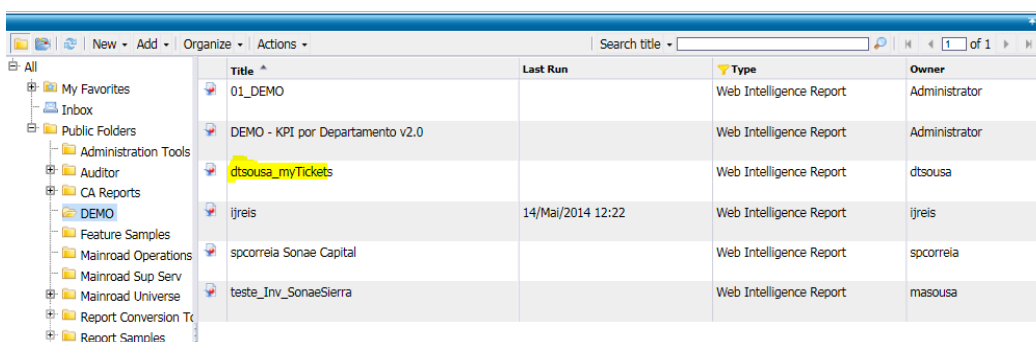


Figura 47 - Relatório CABI criado.

Posteriormente bastará clicar duas vezes no relatório criado e proceder à sua criação. Na seguinte figura será apresentada a página que irá surgir após se ter clicado no relatório.

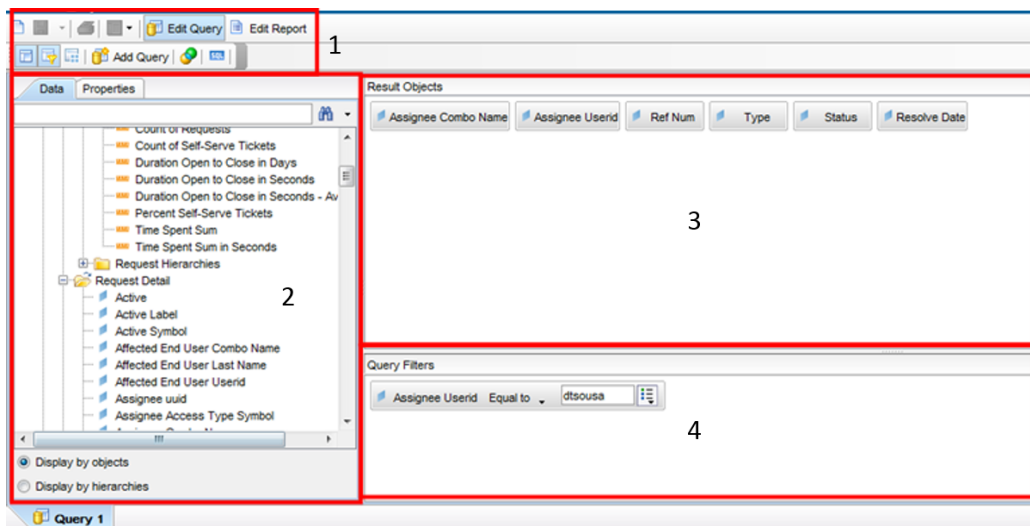


Figura 48 - Zona de configuração do relatório.

Na secção 2 encontram-se todos os elementos que constituem o seu universo. Estes elementos são os atributos dos campos da CMDB, como o campo *assignee*, data, estado, tipo do *ticket*, entre outros.

A secção 3 representa a zona para a qual se irão arrastar os campos que se pretende observar no relatório (por outras palavras será o “select” do relatório). Neste exemplo pretendem-se visualizar os campos: nome do *assignee*, *user id*, a referência do *ticket*, tipo, o estado e por fim a data de resolução.

A secção 4 irá ser a zona onde se colocará quais os campos que se pretende filtrar. Neste exemplo pretende-se ter uma listagem de todos os *tickets* que estão sob a responsabilidade do membro com o userid “dtsousa”.

Caso se deseje personalizar o relatório e tal não seja possível através do *drag-and-drop*, bastará ir à secção 1 e clicar no botão “SQL”:

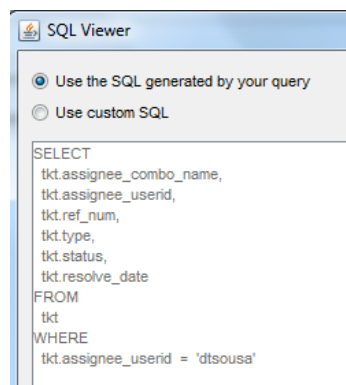


Figura 49 - SQL gerado através do drag-and-drop.

Após clicar no botão mencionado, irá surgir uma nova janela contendo a *query* SQL que foi gerada após o processo de *drag-and-drop*. Caso se queira editar essa *query*, bastará seleccionar o botão “Use custom SQL” e proceder à alteração da mesma. Por fim, restará clicar no botão “Refresh Data” para se visualizar o *output*.

Assignee Combo Name	Assignee Userid	Ref Num	Type	Resolve Date	Status
Sousa, Duarte Tiago	dtsousa	594915	R	05-02-2014 03:09:10 PM	CL
Sousa, Duarte Tiago	dtsousa	595500	R	05-05-2014 12:49:14 PM	CL
Sousa, Duarte Tiago	dtsousa	595519	R	05-05-2014 04:13:52 PM	CL
Sousa, Duarte Tiago	dtsousa	596221	R	05-07-2014 11:19:06 AM	CL
Sousa, Duarte Tiago	dtsousa	597704	R	05-14-2014 04:17:46 PM	CL
Sousa, Duarte Tiago	dtsousa	597928	R	05-28-2014 12:16:16 PM	CL
Sousa, Duarte Tiago	dtsousa	598133	R	05-14-2014 10:46:06 AM	CL
Sousa, Duarte Tiago	dtsousa	598324	R	05-14-2014 05:46:42 PM	CL

Figura 50 - Output do relatório.

No lado esquerdo poderão visualizar-se quais os elementos do universo que foram utilizados na construção deste relatório, e no lado direito será apresentada uma listagem de todos os *tickets* que estão/estavam sob responsabilidade do *user* “dtsousa”. No presente exemplo, a análise do relatório permite concluir que este não está a ter problemas no desempenho das suas funções. Tal conclusão resulta de todos os *tickets* que estão ou estavam sob a sua responsabilidade já se encontrarem no estado *closed*, ou seja, já foram resolvidos corretamente.

A plataforma CA BI/BOXI permite também fazer um agendamento de relatórios. Deste modo, poder-se-á instruir a plataforma a criar este relatório de x em x tempo (e.g. semanalmente, mensalmente), podendo enviar esse relatório em pdf, excel ou xml para vários *emails*, para um repositório FTP, entre outros.

9 Integração das Ferramentas / Plataformas

Após ter-se descrito cada plataforma como uma entidade individual, poder-se-á então através do esquema presente na seguinte figura, ter uma visão global como todas elas interagem entre si.

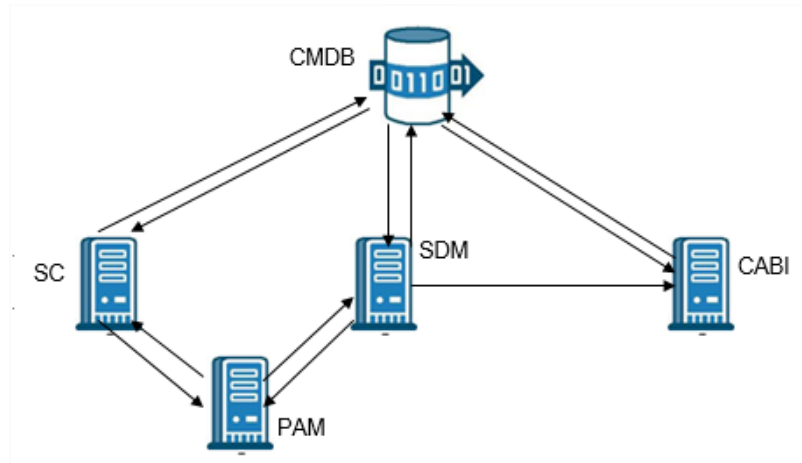


Figura 51 - Relação entre as várias plataformas.

10 Trabalho desenvolvido

Neste capítulo será apresentado e demonstrado o trabalho efetuado. As várias tarefas/atividades realizadas serão separadas pelas diferentes plataformas que foram utilizadas.

10.1 CA Service Desk Manager (CA SDM)

O trabalho realizado nesta plataforma é em grande parte a nível de suporte (interno e para os vários departamentos) e de gestão da plataforma. Ou seja, foi efetuado no âmbito de uma equipa responsável por satisfazer todos os *tickets* referentes a grande parte das ferramentas ITSM implementadas no cliente. De seguida irá ser descrito com maior detalhe algumas dessas tarefas.

10.1.1 Atividades de gestão da plataforma

Relativamente às atividades de gestão desta plataforma, existem atualmente no cliente quatro ambientes da plataforma de CA SDM:

- **DEV (desenvolvimento)** - Neste ambiente são efetuados todos os desenvolvimentos numa primeira fase. Desta forma, se algo não correr como planeado não ocorrerá grande impacto, uma vez que esta não é a plataforma que é realmente utilizada por todos os utilizadores.
- **STG (*staging*)** - Esta é a plataforma onde serão efetuadas as migrações pela primeira vez, assim como a instalação de test fixes. Neste ambiente o objetivo consiste em, após a implementação de (grandes) alterações, proceder à execução de testes exaustivos de forma a se encontrar erros ou funcionalidades que não estejam a funcionar como deveriam. Por exemplo, em integrações com outras plataformas como o SAP, MAXIMUS ou outras plataformas de CA SDM externas. Posteriormente, corrigir todos esses erros de forma a trazer de volta a plataforma à estabilidade.
- **QLD (qualidade)** - este ambiente irá ser uma cópia integral do ambiente de PRD, ou seja, tudo o que é inserido, alterado ou removido neste ambiente, tem de ser realizado primeiro em QLD de forma a evitar possíveis problemas que possam levar à indisponibilidade da plataforma ou mesmo à inconsistência dos dados inseridos.

- **PRD (produção)** - este é o ambiente “oficial” que é utilizado por todos os analistas, *end users* e *customers*. A nível de infraestrutura, o ambiente produtivo é composto por:
 - ✓ 1 Servidor Standby;
 - ✓ 1 Servidor Background (com função de primário);
 - ✓ 4 Servidores aplicativos (cada um deles tem uma capacidade que ronda os 200 utilizadores);
 - ✓ 1 Servidor aplicativo que é responsável pelos *attachments*;
 - ✓ 1 Servidor aplicativo para os clientes de Espanha;
 - ✓ 1 Servidor para acessos externos, fora da rede do cliente.

O ambiente de produção do cliente possui aproximadamente:

- 50 000 *contacts*;
- 500 *groups*;
- 2500 *change categories*;
- 3300 *incident areas (IA's)*;
- 2800 *task's*;
- 9900 *locations*;
- 11300 *organizations*;
- 91500 CI 's;

Desta forma, com a ajuda da equipa, é realizada uma gestão de todos os pontos anteriormente mencionados, desde a:

- Criação, alteração e desativação de *contacts, groups, locations e organizations*;
- Criação e configuração de *IA's, change categories, task's, SLA's, workshift's*;
- Associações entre os vários elementos (e.g. associações de *contacts a groups*, de *change categories a workflow's* e por sua vez às *task's* que o constituem, de *SLA's a IA's*);
- Criação de *access type's e roles* e atribuição dos mesmos aos contactos;
- Suporte interno para os utilizadores de Portugal e Espanha (resolução de *tickets*);
- Criação de relatórios via SQL;
- Importações massivas na plataforma;
- Customizações (alteração de *forms*);
- Instalação de *test-fixes* de forma a corrigir problemas reportados;
- Monitorização a nível da disponibilidade e performance da plataforma;
- Passagem de informações/registos/dados entre os vários ambientes;

- Migrações da plataforma (upgrade de versões);
- Criação de SPELL CODE (consistem na criação de *triggers* para despoletar certas ações, de forma dinâmica);

10.1.2 Passagem de informação entre ambientes

Como referido anteriormente, quando se pretende passar dados para o ambiente de PRD (ou entre outros ambientes) é necessário fazê-lo primeiro no ambiente de QLD. De notar que isto não se trata de uma obrigação, depende muito do nível de maturidade e rigor do cliente. Isto porque certos clientes querem que esse ambiente seja uma cópia exata do ambiente de PRD, para que desta forma se evite ao máximo a ocorrência de problemas ou que leve a uma inconsistência de dados.

Será equívoco assumir que introduzindo exatamente o mesmo em PRD e em QLD, estar-se-ia a manter os dados completamente iguais. Por exemplo:

- ao se inserir a informação em ambiente QLD e após se fazer o “*Save*”, esse registo iria ser criado na base de dados, por exemplo, com o *ID:123*;
- Ao se inserir em ambiente PRD exatamente a mesma informação e após o “*Save*”, esse ID na base de dados iria ser diferente (por exemplo, *ID: 456*).

Esta diferença relativa aos *ID's* poderá ter um efeito desastroso porque existem vários processos que utilizam os *ID's* dos registos.

Uma das formas de se efetuar aquela passagem poderia ser através da base de dados, mas como foi referido anteriormente, o acesso à mesma é restrito, estando normalmente sobre a responsabilidade da equipa de *DBA*.

É precisamente aqui que desempenham o seu papel a camada de objeto e lógica, descritas anteriormente. Estas camadas permitem-nos fazer *queries* aos objetos da plataforma, não existindo desta forma um contacto direto com a BD. Para fazer essas consultas, basta utilizarmos o comando: *pdm_extract*.

No capítulo 6.3.5. CMDB foi apresentado um pequeno exemplo de uma consulta feita na BD sobre um contacto da plataforma. Coloca-se a questão de como realizá-lo através da camada de objeto/lógica e de qual o objeto que se deve utilizar relativo à tabela *ca/ups_contact*.

Para obter objeto responsável pelos contactos da plataforma, bastará realizar os seguintes passos:

1. Na plataforma de SDM, efetuar uma pesquisa de contactos e seleccionar qualquer um dos registos listados:

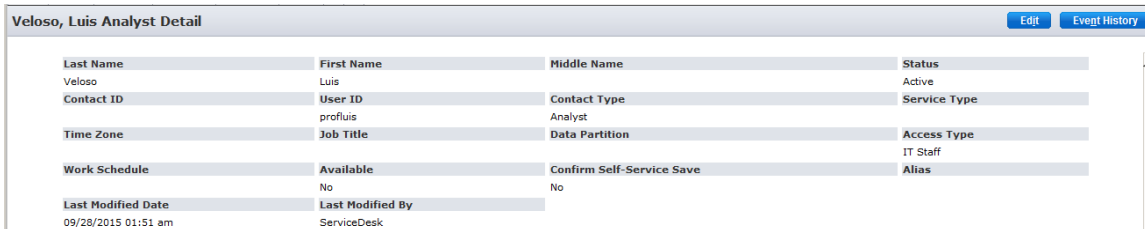


Figura 52 - Visualização de detalhes de um contacto.

2. Clicar em *CTRL+lado direito do rato* na página e escolher a opção 'View Source' (e.g. utilizando o Internet Explorer):

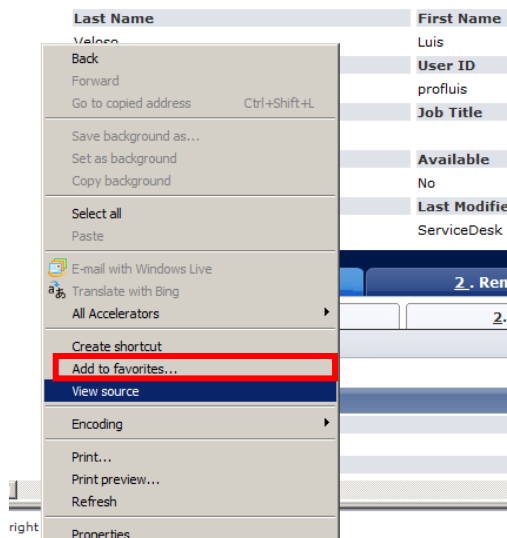


Figura 53 - Procedimento para visualizar o código fonte de uma página.

3. Depois de surgir a página com o código fonte, realizar-se-á uma pesquisa (*CTRL+F*) e procurar-se-á a *string* "factory". Será obtida informação semelhante à apresentada na seguinte figura.

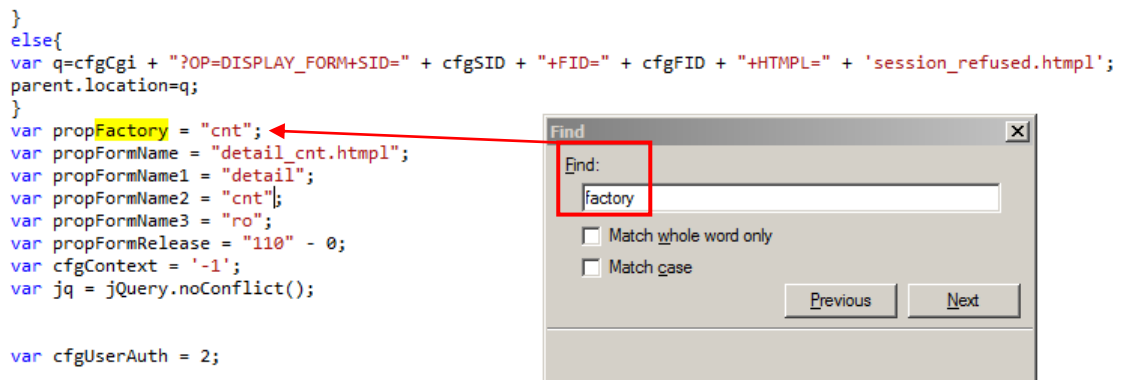


Figura 54 - Descrição da obtenção do objeto através do código fonte da página.

Após se efetuar este passo é possível concluir que o objeto relativo aos contatos é o *cnt*.

4. Pode-se utilizar um comando muito interessante (*bop_sinfo*) para se ter mesmo a certeza que é este o objeto pretendido, como é demonstrado na seguinte figura.

```
C:\Users\Administrator>bop_sinfo -f cnt
Factory cnt
  Rel Attr      = id
  Common Name   = combo_name
  Func Group    = contact
  Display Name  = Contact
  Display Group = event_tree_node stored_query macro act_log object_notify_ac
t log notify_rule notify_msg_tpl
  REST Operations = CREATE READ UPDATE
  Schema Name    = ca_contact
  Producers     = cnt usp_contact
```

Figura 55 - Utilização do comando *bop_sinfo -f*.

Como se pode verificar, o objeto *cnt* corresponde às tabelas *ca/usp_contact* da BD. Pretendendo-se conhecer quais os atributos que compõem esse objeto, deverá usar-se o *bop_sinfo* da forma descrita na seguinte figura.

```
C:\Users\Administrator>bop_sinfo -dq cnt
Factory cnt < ca_contact, usp_contact >
Attributes:
  id <contact_uid>          UUID UNIQUE
  producer_id              LOCAL STRING(20)
  persistent_id            LOCAL STRING(60)
  supervisor_contact_uid   SREL -> cnt.id
  last_mod_dt last_update_date DATE
  creation_date            DATE
  last_name                STRING(100) REQUIRED SKEY
  first_name               STRING(100) SKEY
  middle_name              STRING(100) SKEY
  alias                    STRING(30)
  delete_flag_inactive     SREL -> actbool.enum REQUIRED
  version_number           INTEGER
  last_mod_by last_update_user STRING(64)
  creation_user            STRING(64)
  exclude_registration    INTEGER
  delete_time              DATE
  userid                   STRING(100)
  phone_number pri_phone_number STRING(40)
  alt_phone alt_phone_number STRING(40)
  mobile_phone mobile_phone_number STRING(40)
  type contact_type       SREL -> ctp.id
  billing_code cost_center SREL -> cost_ctr.id
  location location_uid   SREL -> loc.id
  dept department         SREL -> dept.id
  organization organization_uid SREL -> org.id
  admin_org admin_organization_uid SREL -> org.id
  company company_uid     SREL -> ca_cmpny.id
  notes comment<comments> STRING(255)
  fax_phone fax_number    STRING(40)
  beeper_phone pager_number STRING(40)
  email_address            STRING(120)
  pager_email_address     STRING(120)
  room_location            STRING(30)
  contact_num alternate_identifier STRING(30)
  position job_title      SREL -> position.id
```

Figura 56 - Utilização do comando *bop_sinfo -dq*.

Importante: este comando é extremamente relevante no âmbito da construção de ficheiros de configuração com o intuito de efetuar carregamentos massivos, que irá ser descrito mais tarde neste documento.

5. Após conhecer qual o objeto a ser trabalhado, recorre-se à utilização do comando *pdm_extract*. Para se passar um contacto do ambiente de QLD para o de PRD, tem-se numa primeira fase que se extrair essa informação. Para tal

bastará utilizar o comando citado, seguindo-se da *flag -f* e de uma *query* SQL como descrito a seguinte figura.

```

Administrator: Command Prompt
C:\Users\Administrator>pdm_extract -f "select * from ent where userid='duarte'"
TABLE ca_contact
ent
  admin_organization_uuid alias alt_phone_number alternate_identifier conn
ent
  company_uuid contact_type cost_center creation_date creation_user delete
_time department email_address exclude_registration fax_number first_name floo
r_location
  id inactive job_function job_title last_name last_update_date last_updat
e_user
  location_uuid nail_stop middle_name mobile_phone_number organization_uid
d
  pager_email_address pager_number pri_phone_number room_location supervis
or_contact_uuid
  tenant tenant_group userid version number
  "473D8A1AE78F5541A2840C28DAE807D0" "teste" "43434" "duarte" "" ""
  "2307" "" "08/12/2015 15:03:49" "ServiceDesk" "" ""
  "duartesousa@sonedonain.com" "" "123456789" "Duarte" ""
  "8E4BB998C188D4EBB1E337686623CBF" "0" "" "702" "Sousa"
  "09/30/2015 00:27:25" "ServiceDesk" "35170E3EFB189E4DA022C25C7B458E1AB"
  "Tiago" "" "473D8A1AE78F5541A2840C28DAE807D0" "343434" "34343"
  "914125458" "" "" "" "" "duarte" "2"
ca_contact
rows:1
C:\Users\Administrator>

```

Query

Atributos que serão devolvidos

Output da query

Figura 57 - Utilização do comando pdm_extract.

- Depois de efetuada a consulta, colocar-se-á o mesmo *output* num ficheiro. Ou seja, executar-se-á o mesmo comando acrescentando > teste.txt (sem aspas). Depois de criado esse ficheiro, bastará passá-lo para o servidor de PRD.

Em seguida usa-se uma linha de comandos para correr o comando *pdm_userLoad*:

```

C:\Users\Administrator\Desktop>pdm_userload -f teste.txt
command = C:\PROGRAMAS\CA\SERVICIT\bin\adload teste.txt
C:\Users\Administrator\Desktop>

```

Figura 58 - Utilização do comando pdm_userload.

Após este passo o registo é criado no ambiente de PRD, de forma exatamente igual ao que foi criado em QLD.

Importante: Este procedimento pode ser usado também nas alterações e mesmo nas remoções de registos na plataforma CA SDM.

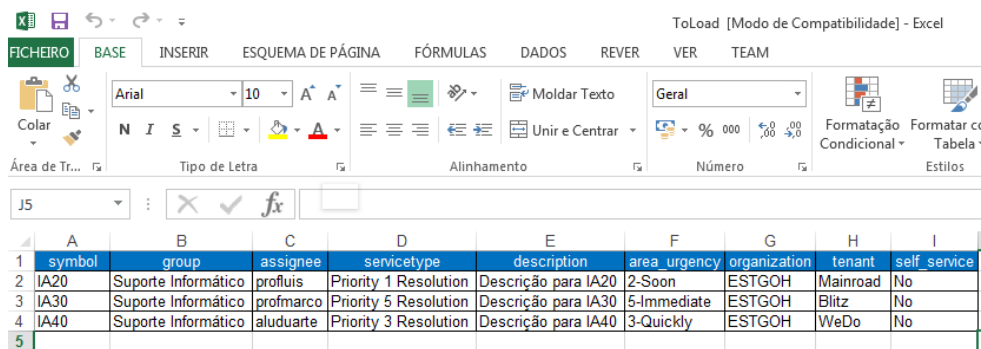
10.1.3 Carregamentos massivos

Por vezes são recebidos pedidos para se inserir, alterar ou desativar na plataforma (indiretamente na CMDB), um elevado número de dados, podendo esse número chegar às centenas ou mesmo aos milhares. Esses registos normalmente são referentes à criação/alteração/desativação de *IA's*, *change categories*, *task's*, *CI's*, alteração de grupos em *tickets*, entre outros.

Desta forma, devido ao elevado número, não seria prático fazer estas ações de forma individual, sendo por isso usada uma ferramenta/aplicação denominada “CASDMSync”. Esta aplicação foi desenvolvida em .NET e recorre à utilização de um ficheiro de configuração para proceder à leitura de um ficheiro excel (que contém toda a informação a ser importada), enviando-a para a plataforma utilizando *web services*.

Em seguida, será apresentado um pequeno exemplo referente à criação de *incident áreas*. Em primeiro lugar é necessário analisar o ficheiro excel, pois é necessário efetuar algumas configurações no mesmo, caso contrário impossibilitará a importação.

Como se pode visualizar na seguinte figura, trata-se de um ficheiro relativamente simples, onde o cabeçalho possui a referência dos campos que vão ser preenchidos na criação da IA, e por baixo o conteúdo desses vários campos.



	A	B	C	D	E	F	G	H	I
1	symbol	group	assignee	servicetype	description	area_urgency	organization	tenant	self_service
2	IA20	Suporte Informático	profuis	Priority 1 Resolution	Descrição para IA20	2-Soon	ESTGOH	Mainroad	No
3	IA30	Suporte Informático	profmarco	Priority 5 Resolution	Descrição para IA30	5-Immediate	ESTGOH	Blitz	No
4	IA40	Suporte Informático	alduarte	Priority 3 Resolution	Descrição para IA40	3-Quickly	ESTGOH	WeDo	No
5									

Figura 59 - Visualização do ficheiro Excel a ser utilizado na importação massiva.

É necessário ter em conta três principais aspetos relativamente a este ficheiro:

- O ficheiro tem que ser gravado como ‘*Microsoft Excel 97-2003 Worksheet*’;
- Dar um nome à tabela de conteúdos recorrendo a “*Fórmulas → Gestor de Nomes*”:

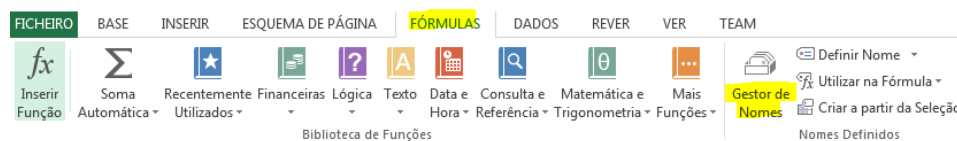


Figura 60 - Denominação das tabelas de conteúdos através do "Gestor de Nomes".

De seguida bastará dar um nome (neste caso foi dado o nome **IA**) e selecionar toda a tabela, incluindo os cabeçalhos:

symbol	group	assignee	servicetype	description	area_urgency	organization	tenant	self_service
IA20	Suporte Informático	profuis	Priority 1 Resolution	Descrição para IA20	2-Soon	ESTGOH	Mainroad	No
IA30	Suporte Informático	profmarco	Priority 5 Resolution	Descrição para IA30	5-Immediate	ESTGOH	Blitz	No
IA40	Suporte Informático	alduarte	Priority 3 Resolution	Descrição para IA40	3-Quickly	ESTGOH	WeDo	No

Editar Nome

Nome:

Âmbito:

Comentário:

Refere-se a:

Figura 61 - Descrição da atribuição de um nome a uma tabela.

- Todos os campos necessitam de estar formatados como texto:

B	C	D	E	F	G	H	I
group	assignee	servicetype	description	area_urgency	organization	tenant	self_service
Suporte Informático	profuis	Priority 1 Resolution	Descrição para IA20	2-Soon	ESTGOH	Mainroad	No
Suporte Informático	profmarco	Priority 5 Resolution	Descrição para IA30	5-Immediate	ESTGOH	Blitz	No
Suporte Informático	alduarte	Priority 3 Resolution	Descrição para IA40	3-Quickly	ESTGOH	WeDo	No

Figura 62 - Formatação dos campos como texto

Após efetuado este passo, o ficheiro deverá ser gravado e passado para dentro da pasta da ferramenta *CASDMSync*:

Log	2015-10-09 01:08	File folder	
CASDMSync4	2011-11-24 10:12	Application	170 KB
CASDMSync4.exe	2015-10-09 01:53	XML Configuration File	4 KB
ToLoad.xls	2015-10-09 01:54	XLS File	307 KB

Figura 63 - Ficheiros que constituem a aplicação CASDMSync.

De seguida procede-se à abertura do ficheiro de configuração (*CASDMSync4.exe*, ficheiro xml) e procede-se à sua configuração. Este ficheiro é complexo, pois é nele que vão estar todas as configurações: desde a ligação ao *web service*, os mapeamentos dos campos, o modo de execução e *debug*, entre outros. Seguem-se alguns excertos desse ficheiro e respetiva explicação:

1. Utilizador de acesso aos *ws's*:

```
<add key="CASDM.Username" value="servicedesk" />
<add key="CASDM.Password" value="!" />
```

Figura 64 - Ficheiro de configuração (acessos).

Aqui será indicado qual o *user* e *password* a serem utilizados para se efetuar o *login* ao *web service*.

2. Indicação do *URL* do WSDL e da localização do ficheiro de log:

```
<add key="CASDM.Url" value="http://192.168.220.129:8080/axis/services/USD_R11_WebService?wsdl"/>
<!-- Log file -->
<add key="CASDM.LogFile" value="C:/Users/Administrator/Desktop/CASDMsync/Log/Results.log" />
```

Figura 65 - Ficheiro de configuração (indicação do WSDL e do ficheiro de Log).

Referência ao *web service* e da localização dos *logs*, que será importante ser analisado caso surja algum erro.

3. Referência à tabela de conteúdos e da ODBC:

```
<add key="ODBC.TableName" value="IA" />
<add key="ODBC.ConnectionString" value="DSN=USDSync2" />
```

Figura 66 - Ficheiro de configuração (indicação da tabela e ODBC).

Aqui terá de se referenciar a tabela que foi criada no excel, através do gestor de nomes (foi criado com o nome **IA**). É necessário atenção ao valor **DSN=USDSync2** que será mencionado mais tarde no contexto do ODBC.

4. Referência ao objeto que será alvo da importação massiva:

```
<add key="CASDM.Object" value="pcat" />
```

Figura 67 - Ficheiro de configuração (indicação do objeto).

A identificação do objeto que deve ser utilizado foi realizada no capítulo “11.1.2. Passagem de informação entre ambientes”.

5. Mapeamento dos campos:

```
<add key="ODBC.Fields" value=
"symbol;group;assignee;servicetype:description;area_urgency:organization:tenant:self_service" />
<add key="CASDM.Attributes" value=
"sym;group;assignee;service_type:description;category_urgency:organization:tenant:ss_include"/>
<add key="CASDM.Derefs" value=";grp.last_name;agt.userid;sdsc.sym;;urg.sym;org.name;tenant.name;bool.sym"/>
<!-- Insert any string in the correct position if numeric. Use 1 by example -->
<add key="CASDM.DerefNumeric" value=":::" />
<add key="CASDM.DerefWhere" value=":::" />
<add key="CASDM.Keys" value="sym:::" />
```

Figura 68 - Ficheiro de configuração (mapeamento de campos).

Esta constituirá a parte mais difícil do processo, pois implica um uso abusivo do comando *bop_sinfo*, que anteriormente foi referido ser um comando de extrema importância. Este excerto será de seguida analisado por partes:

- ✓ **ODBC.Fields** → Referência aos cabeçalhos do ficheiro excel, separados por “;”

symbol	group	assignee	servicetype	description	area_urgency	organization	tenant	self_service
--------	-------	----------	-------------	-------------	--------------	--------------	--------	--------------

- ✓ **CASDM.Attributes** → É aqui necessário recorrer-se ao comando *bop_sinfo* para se poder referenciar os campos. Embora estes sejam parecidos aos cabeçalhos do excel, não se tratam dos mesmos. Estes campos são relativos aos campos da camada de objeto. Como já se viu anteriormente, está-se a trabalhar com o objeto “*pcat*”, e para tal é importante conhecer por que atributos é constituído este objeto. Desta forma, basta utilizar-se-á o comando:

bop_sinfo -dq pcat

```
C:\Users\Administrator>bop_sinfo -dq pcat
Factory pcat < Prob_Category >
Attributes:
id                INTEGER UNIQUE
producer_id       LOCAL STRING(20)
persistent_id     persid  STRING(30)
delete_flag       del    SREL -> actbool.enum REQUIRED
sym              STRING(1000) REQUIRED SKEY
last_mod_dt       DATE
last_mod_by       SREL -> cnt.id TENANCY_UNRESTRICTED
description       STRING(500)
organization      SREL -> org.id
assignee          SREL -> agt.id SERVICE_PROVIDER_ELIGIBLE
group             group_id SREL -> grp.id SERVICE_PROVIDER_ELIGIBLE
properties        BREL <- cr_prptpl.owning_area <owning_area = ?>
survey            SREL -> svy_tpl.id
auto_assign       INTEGER
service_type      SREL -> sdsc.code
category_urgency SREL -> urg.enum
owning_contract   SREL -> svc_contract.id SERVICE_PROVIDER_ELIGIBLE
cr_flag           INTEGER
in_flag           INTEGER
pr_flag           INTEGER
suggest_knowledge INTEGER
assignable_ci_attr STRING(60)
service_grps      BREL <- lrel_svc_grps_svc_pcat.pcat <LREL cnt> <pcat = ?>
service_locs      BREL <- lrel_svc_locs_svc_pcat.pcat <LREL loc> <pcat = ?>
service_schedules BREL <- lrel_svc_schedules_pcat_svc.pcat <LREL wrkshft> <pcat = ?>
ss_include        SREL -> hoal.enum REQUIRED
ss_sym            STRING(128)
flow_flag         INTEGER
```

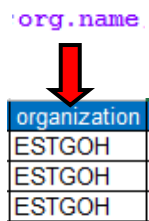
Figura 69 - Comando *bop_sinfo* na elaboração do mapeamento de campos.

Com o *output* deste comando obtêm-se então os campos que são necessários (marcados de amarelo) para o *CASDM.Attributes*. De notar que estes campos terão de estar pela mesma ordem que no *ODBC.Fields* para que o mapeamento seja feito corretamente.

- ✓ **CASDMDerefs** → Como se pode verificar, vários dos campos consistem em SREL's. Um SREL significa que a informação desse campo não se encontra na mesma tabela do objeto, tratando-se sim de uma referência para outra tabela, por exemplo:

organization SREL -> **org.id**

O campo *organization*, apesar de ser um campo utilizado pelo *objeto pcat* é simplesmente uma referência para outro objeto/tabela, que neste caso é o *objeto org*. Poderemos verificar também que essa referência é feita normalmente utilizando o *ID*, mas não é obrigatório. Aliás verifica-se que no ficheiro de configuração, optou-se por utilizar o campo *org.name*:



- ✓ **CASDMKEYS** → Aqui é necessário referenciar-se qual o atributo diferenciador de/para cada inserção, da importação, que neste caso foi utilizado *sym*.

6. Modos de execução:

```
<add key="CASDM.AllowCreate" value="false"/>
<add key="CASDM.AllowUpdate" value="false"/>
```

Figura 70 - Ficheiro de configuração (modos de execução).

Quando não há a certeza que todos os passos descritos até agora estão 100% corretos/mapeados, aconselha-se a correr a ferramenta com estes 2 modos a *false*.

Após finalizado o processo de criação do ficheiro de configuração, é necessário criar o ODBC. Para tal ter-se-á que ir à diretoria “*C:\Windows\SysWOW64*” e abrir o executável “*odbcad32*”:

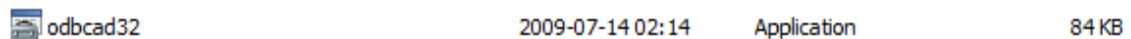


Figura 71 - Localização da aplicação que permite a criação de ODBC's.

De seguida, clica-se em “*System DSN*” e cria-se o ODBC, escolhendo-se a opção “*Driver do Microsoft Excel (*.xls)*”. Seleciona-se o ficheiro que vai ser usado para a importação (neste caso foi o *toLoad.xls*), e por fim atribui-se um nome ao ODBC, que como indicado anteriormente, foi dado o nome *USDSync2* (de notar que poderia ter sido utilizado um outro nome).

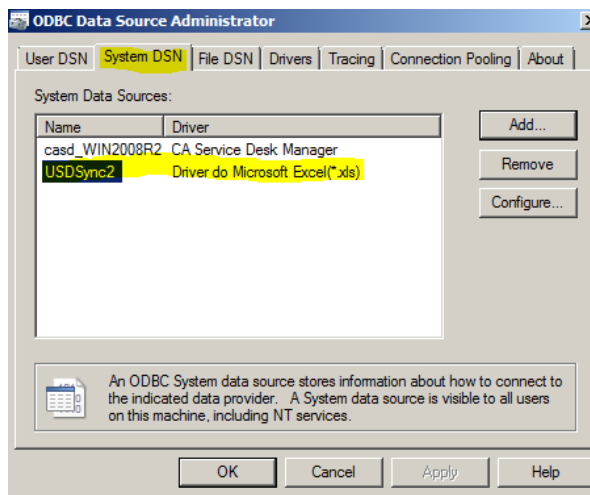


Figura 72 - Criação da ODBC.

Por fim, bastará executar a aplicação, sendo para tal necessário abrir a pasta de ferramentas, e abrir uma linha de comandos como descrito na seguinte figura:

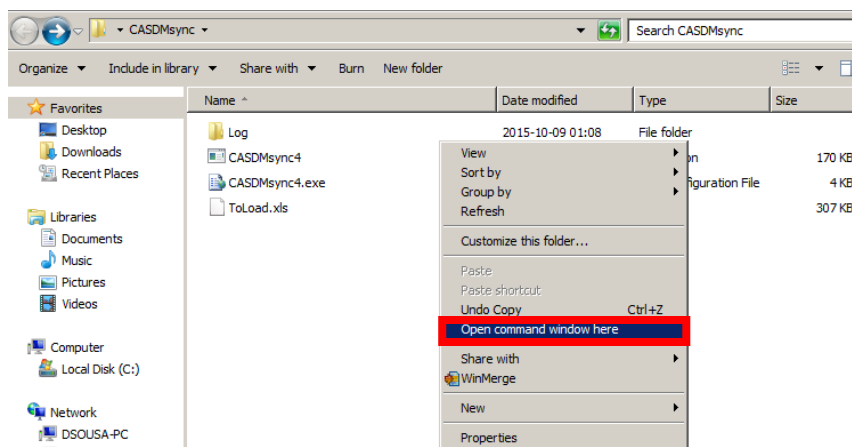


Figura 73 - Como abrir uma linha de comandos na presente janela.

Em seguida executa-se o ficheiro “CASDMSync4.exe”:

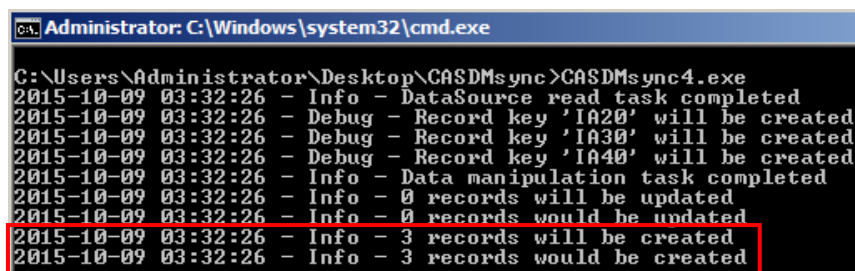


Figura 74 - Execução da ferramenta CASDMSync em modo false false.

Como se colocou o modo “AllowCreate” e “AllowUpdate” a false, a ferramenta irá fazer simplesmente uma verificação de consistência dos dados, a ligação ao ws e uma validação do mapeamento dos campos. Como se pode verificar tudo decorreu

corretamente pois não surgiu nenhum erro. Desta forma, como se trata de uma inserção, coloca-se o parâmetro “*AllowCreate*” a true e volta-se a correr a ferramenta:

```
C:\Users\Administrator\Desktop\CASDMSync>CASDMSync4.exe
2015-10-09 03:35:52 - Info - DataSource read task completed
2015-10-09 03:35:52 - Debug - Record key 'IA20' will be created
2015-10-09 03:35:52 - Debug - Record key 'IA30' will be created
2015-10-09 03:35:52 - Debug - Record key 'IA40' will be created
2015-10-09 03:35:52 - Info - Data manipulation task completed
2015-10-09 03:35:52 - Info - 0 records will be updated
2015-10-09 03:35:52 - Info - 0 records would be updated
2015-10-09 03:35:52 - Info - 3 records will be created
2015-10-09 03:35:53 - Info - 3 records were created
```

Figura 75 - Execução da ferramenta *CASDMSync* em modo true false.

Finalmente as *incident areas* serão criadas no CA SDM.

+ IA20	Mainroad	Veloso, Luis	NO		Active	YES	YES
+ IA30	Blitz	Veloso, Marco	NO		Active	YES	YES
+ IA40	WeDo	Sousa, Duarte	NO		Active	YES	YES
+ Networks			YES	Networks	Active	YES	YES
+ Printer			YES	Printer	Active	YES	YES
+ Software			YES	Software	Active	YES	YES
+ Software.Environment			YES	Software.Environment	Active	YES	YES



IA20 Request/Incident/Problem Area Detail		
Tenant: Mainroad		
Symbol	Organization	Record Status
IA20	ESTGOH	Active
Group	Assignee	Service Type
Suporte Informático	Veloso, Luis	Priority 1 Resolution
Survey	Service Contract	Area Urgency
		2-Soon
Self-Service Include	Self-Service Symbol	
No		
Description		
Descrição para IA20		
Last Modified Date	Last Modified By	
10/09/2015 03:35 am	ServiceDesk	
This category is valid for the following:		
Requests	Incidents	Problems
Yes	Yes	Yes

Figura 76 - Visualização das IA's criadas por *CASDMSync*.

10.1.4 Instalação de test fixes na plataforma

Por vezes, após o processo de migração ou *upgrade* da plataforma CA SDM, poderão ser encontrados problemas. Estes podem ser a nível comportamental (e.g. se um botão deixa de funcionar), ou a nível de performance (e.g. SDM demorar muito a reiniciar, o Tomcat estar repetidamente a terminar, entre outros). Quando essas

situações ocorrem, tem-se numa primeira abordagem de verificar se é possível solucionar o problema, ou implementar um *workaround*.

Caso não seja possível resolver o problema, será necessário entrar em contacto com o suporte da CA. Será essencial descrever no que consiste exatamente o problema e fornecer todas as informações necessárias (como a versão da plataforma, todos os *patches* instalados até ao momento, entre outros) para que o consigam reproduzir.

Após este processo, o suporte da CA enviará um *test fix* para solucionar o problema. No procedimento da sua instalação serão necessários três componentes:

- **Applyptf** (irá possibilitar a instalação do *test fix* na plataforma);
- **Test Fix** recebido pelo suporte da CA (é composto por um conjunto de ficheiros que vão substituir, se necessário, os que se encontram atualmente na plataforma);
- **Cazipxp** (permite fazer a descompressão do *test fix*).

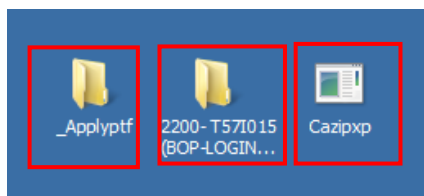





Figura 77 - Componentes necessários para instalação de patch no CA-SDM.

O processo de instalação do *test fix* consistirá nos seguintes passos:

1. Por precaução, realizar um *backup* da pasta do *Service Desk*;
2. Abrir a pasta do *test fix* onde irá ser encontrado um ficheiro com a extensão *.caz*:

 T57I015.caz	2014-04-10 14:37	CAZ File	1,148 KB
---	------------------	----------	----------

3. Copiar o "Cazipxp" para o interior da pasta do *test fix*.

 Cazipxp	2011-02-03 11:11	Application	56 KB
 T57I015.caz	2014-04-10 14:37	CAZ File	1,148 KB

4. Abrir uma linha de comando e executar o *cazipxp* com a opção -u e o nome do *test fix .caz*.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\2200- T57I015 (BOP-LOGIN PROCESS MAY HANG)\New folder\Cazipxp.exe -u T57I015.caz
```

Figura 78 - Método de descompressão do *test fix*.

5. De seguida poderá verificar-se a existência de novos ficheiros após o ato de descompressão:






 boplgin	2014-04-10 11:44	Application	1,912 KB
 Cazipxp	2011-02-03 11:11	Application	56 KB
 T57I015.caz	2014-04-10 14:37	CAZ File	1,148 KB
 T57I015.jcl	2014-04-10 14:37	JCL File	1 KB
 T57I015	2014-04-10 14:37	Text Document	4 KB

Figura 79 - Test fix após descompressão.

6. Abrir a ferramenta “*Applyptf*”:







 applyptf.aix	2010-03-18 08:58	AIX File	423 KB
 applyptf.cfg	2010-01-07 07:11	CFG File	3 KB
 applyptf.esx	2010-03-22 13:08	ESX File	104 KB
 APPLYPTF	2010-05-25 09:29	Application	520 KB
 applyptf.hp	2010-03-18 09:03	HP File	148 KB
 APPLYPTF	2010-06-17 09:44	Configuration settings	75 KB

Figura 80 - Diretoria da aplicação Applyptf.

Escolher a opção “*Apply PTF to local or remote nodes*” e de seguida em “Next”:

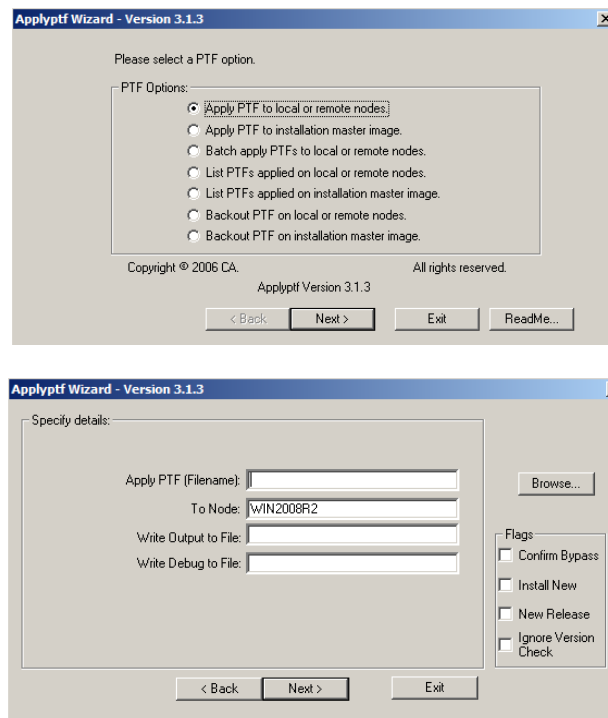


Figura 81 - Menu de instalação do test fix.

No campo “*To Node*” terá de se colocar o *hostname* da máquina. Em seguida selecionar-se-á o ficheiro de instalação do *test fixe*, para tal, basta ir à diretoria onde este foi descomprimido e escolher o ficheiro com a extensão *.jcl* → “*Open* → *Next*” e o *test fix* ficará instalado:

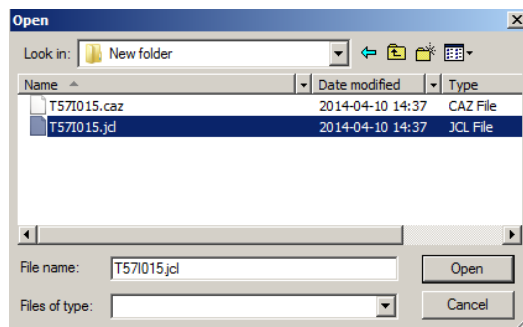


Figura 82 - Seleção do ficheiro .jcl.

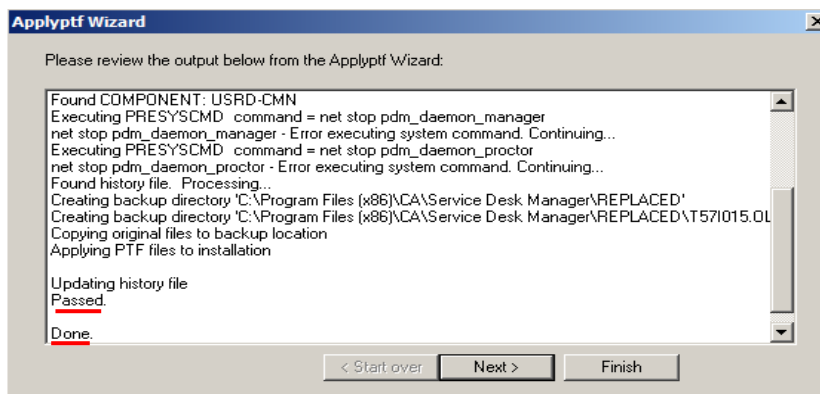


Figura 83 - Instalação de *test fix*.

10.1.5 Criação de campos extra nos formulários

Por vezes os campos que existem por omissão no produto (e.g. *form* de criação de incidentes, de contactos) não são suficientes para satisfazer as necessidades do cliente e por essa razão é necessário criar novos campos.

Por exemplo, para se criar um campo extra no *form* dos incidentes, dever-se-á efetuar o procedimento a seguir descrito:

1. Em primeiro lugar terá de se conhecer o *form* a ser alterado. Como foi referido anteriormente, a única informação que se possui é a necessidade de alterar o *form* de criação de incidente. Para tal, na plataforma CA-SDM clicar-se-á na opção de criação de novo incidente, e em seguida, clicar-se-á com o lado direito do rato em cima de qualquer zona em branco da página. Após esse procedimento irá surgir o seguinte *pop-up*:

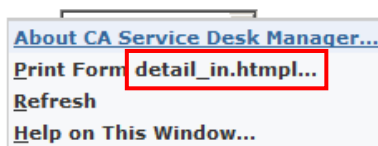


Figura 84 - Descrição da obtenção do form que está a ser utilizado.

2. Deste modo ficará a saber-se que o nome do *form* que nos é apresentado é o “*detail_in.html*”;
3. Em seguida é necessário entrar no servidor primário. Ao entrar neste servidor irá utilizar-se uma ferramenta disponibilizada pelo produto denominada “WSP” (*Web Screen Painter*). Para tal bastará clicar no “*menu Start → My Programs → CA → Service Desk Manager → Web Screen Painter*”:



Figura 85 - Indicação de como localizar o WSP.

4. Após ser aberta a ferramenta WSP, é importante ter a noção que não se trata apenas de criar o novo campo no formulário. É necessário criar uma coluna nova na tabela do objeto *cr* (*Call_Req*) na CMDB, para que haja um mapeamento entre o campo do *form* e o respetivo campo na CMDB. Para tal, no WSP ter-se-á de clicar no botão “*Schema Designer*” (15):

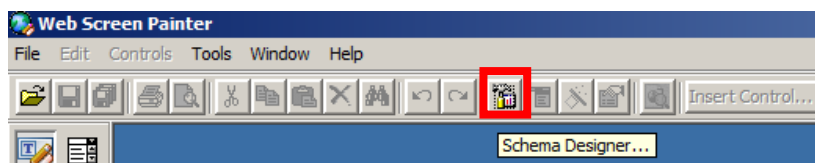


Figura 86 - Descrição da localização do Schema Designer.

De seguida irá surgir a seguinte janela:

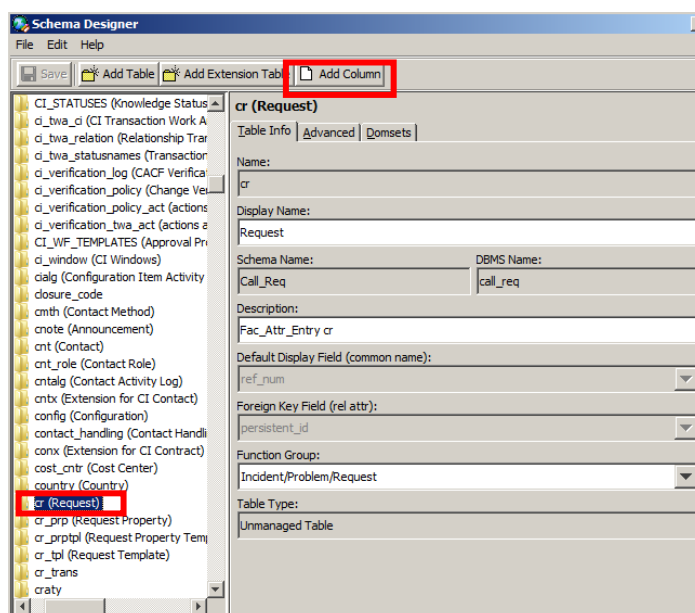


Figura 87 - Schema Designer.

5. Nesta janela, terá que se escolher qual o objeto em que se quer criar uma nova coluna (que como visto anteriormente será o objeto *cr*). De seguida bastará clicar em “Add Column”:

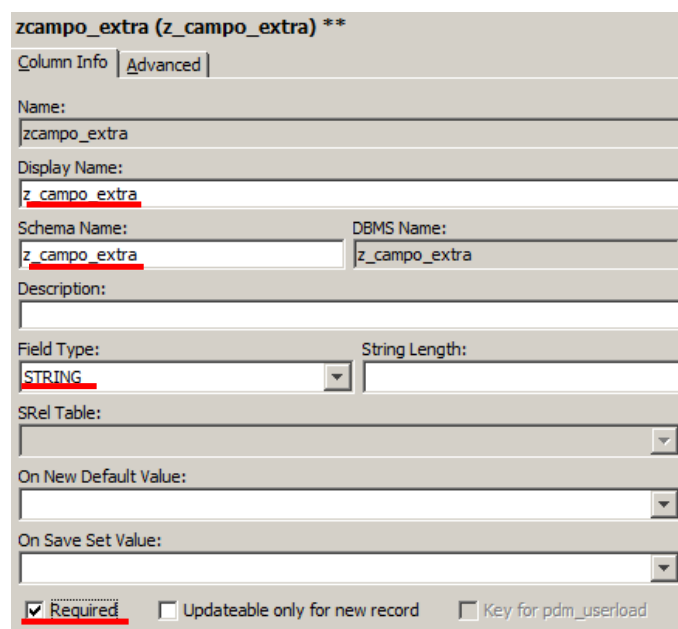
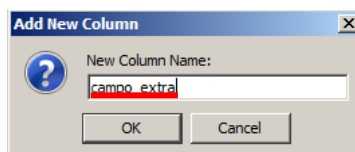


Figura 88 - Atribuição do nome e configuração da coluna a ser criada.

Depois de atribuído o nome à nova coluna é sempre associado o prefixo *z*. Isto porque se trata de uma nova coluna (ou seja, uma coluna que não

vem com o produto). De seguida é necessário proceder à sua configuração, especificando-se qual o tipo de campo (*string*, *boolean*, *int*...) e se consiste num campo obrigatório ou não.

6. Posteriormente é necessário fazer um “Save” e de seguida é necessário clicar na opção “Publish”:

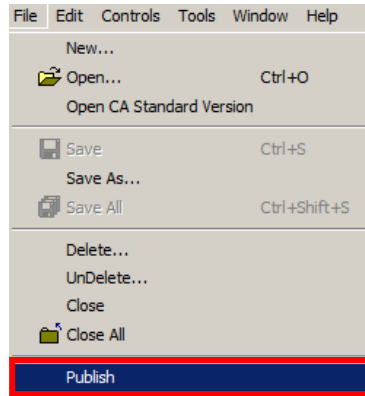


Figura 89 - Publish da alteração da schema.

7. Após ter-se feito o “Publish” da alteração é necessário desligar o CA SDM recorrendo ao comando: `net stop pdm_daemon_manager`, ou através dos “Services”. Quando este passo estiver concluído teremos que correr o comando `pdm_publish` para que desta forma a alteração seja realmente implementada na CMDB e na *Schema*.

```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>net stop pdm_daemon_manager_

Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>pdm_status
The Daemons are not running.

C:\Users\Administrator>pdm_publish
Pdm_publish - Implement DBMS schema changes from Web Screen Painter
Verifying files output from Web Screen Painter...
Verifying CA Service Desk Manager server is not running...

Merging schema files...
Schema merge successful

Building data dictionary...
Data dictionary build successful

Adding new columns to existing tables...
Creating column Call_Req.z_campo_extra
1 column(s) added successfully

Building data dictionary...
Data dictionary build successful

Building tblobj.cfg file...
Build tblobj.cfg file successful

Normal end of pdm_publish
C:\Users\Administrator>
```

Figura 90 - Comando `net stop pdm_daemon_manager` e `pdm_publish`.

Quando este passo estiver efetuado, bastará voltar a levantar o CA SDM através do comando `net start pdm_daemon_manager`, ou através dos “Services”. Para se confirmar que o novo campo foi realmente criado, usar-se-á o comando: `bop_sinfo -dq cr` e procurar-se-á pelo campo em questão:

```
sap_sysid          STRING(3)
sap_systyp         STRING(12)
sap_userstatus     STRING(40)
sap_xnum           STRING(32)
zcampo_extra z_campo_extra STRING(1) REQUIRED
```

Figura 91 - bop_sinfo para validação da criação do novo campo.

8. Quando a plataforma CA SDM estiver levantada, é necessário voltar novamente ao WSP para que desta vez seja criado o novo campo no formulário. Para tal, clica-se em “Open”.

Como se poderá visualizar na seguinte figura, após a abertura do *form* (*detail_in.html*), surgiu uma página que demonstra a estrutura desse form. Ou seja, todos os campos que o constituem (secção 1).

No lado esquerdo (secção 2) encontra-se um painel com as várias componentes que podem ser adicionadas a um *form*. Desta forma bastará selecionar a componente desejada (neste caso usou-se uma *text box*) e arrastá-la para a zona do *form*.

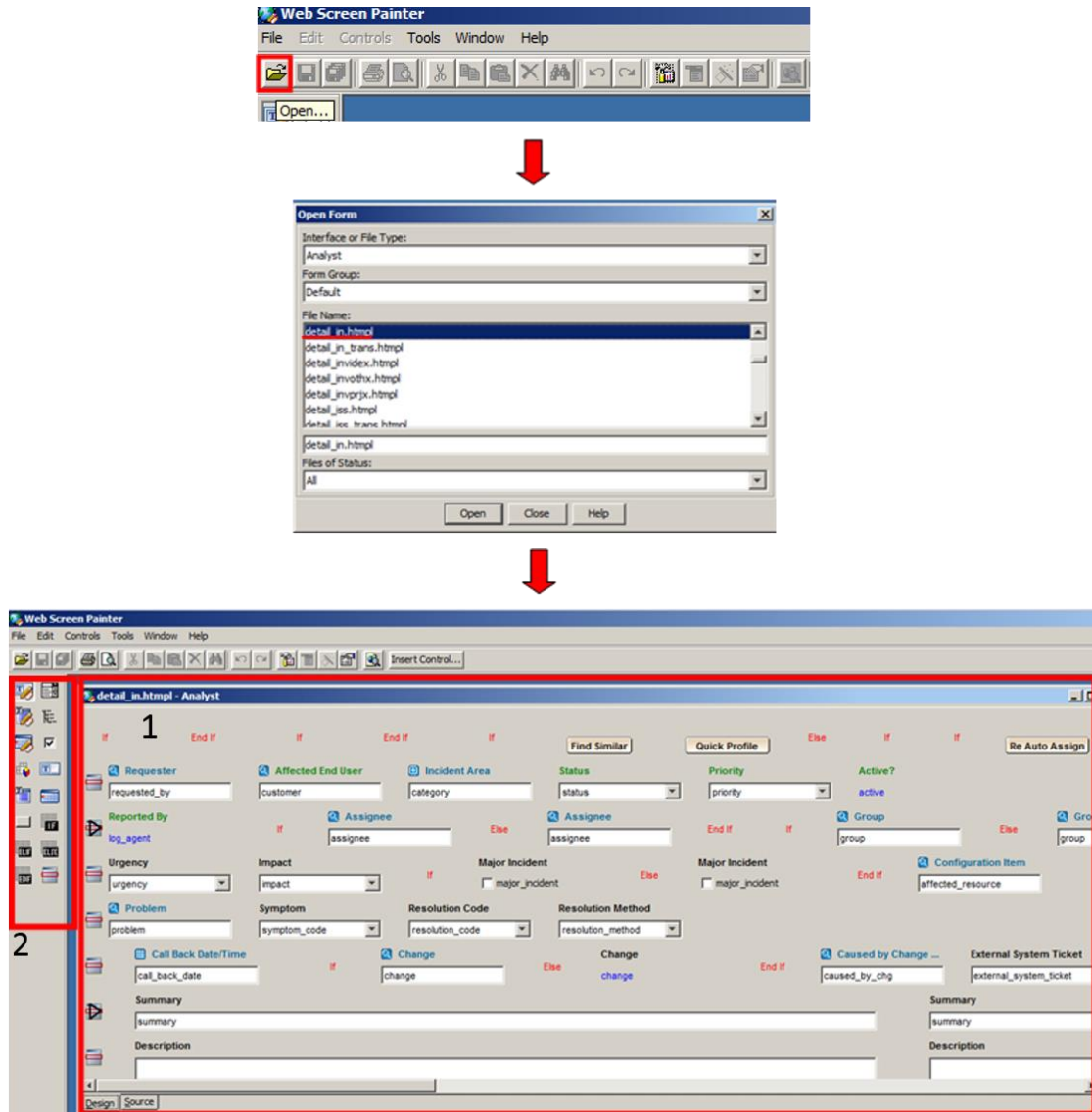


Figura 92 - Processo para abertura do form.

9. Depois de se arrastar o componente para dentro do *form*, bastará clicar e proceder à sua configuração (neste caso só foi atribuído um nome, “*Campo Extra*”).

Em seguida realizar-se-á o mapeamento com o novo campo criado no início deste processo, bastando para tal clicar duas vezes no campo “*Attribute*” e seleccionar então o campo em questão.

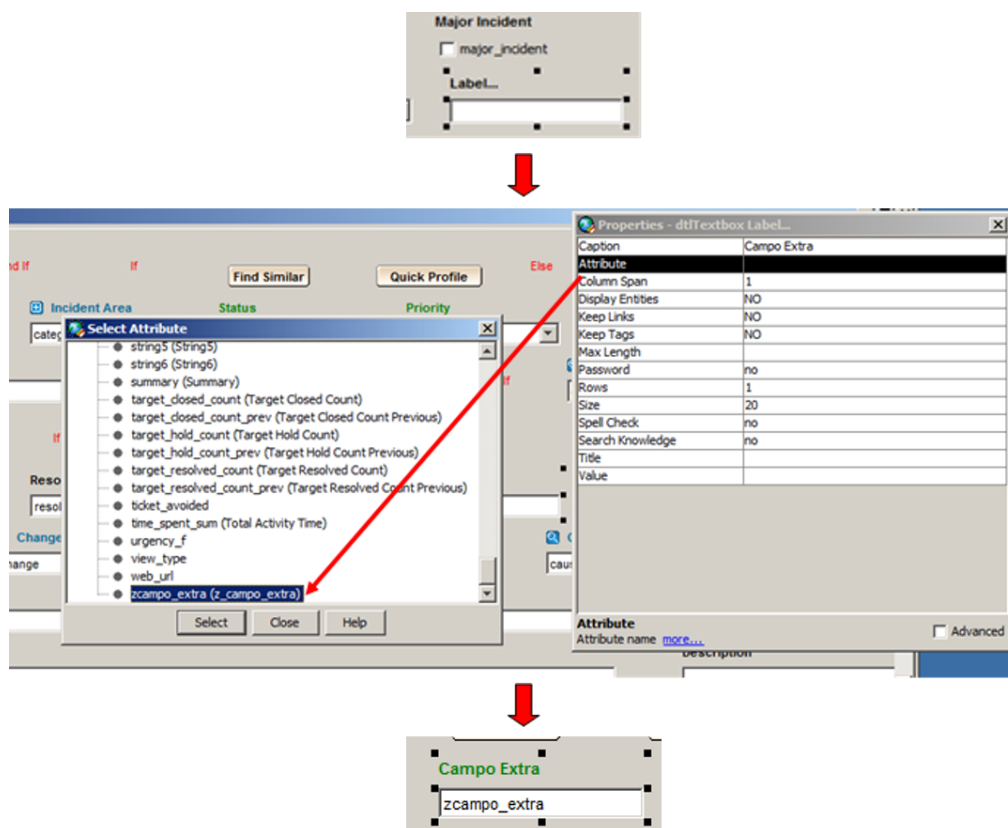


Figura 93 - Criação e mapeamento do novo campo do form.

- Depois do mapeamento estar feito bastará realizar novamente um “Save” e “Publish”:

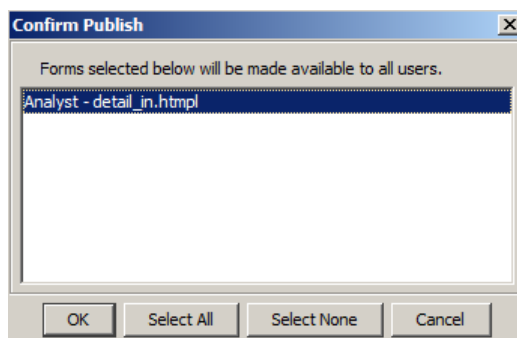


Figura 94 - Confirmação do publish referente à alteração de form.

- Por fim, é necessário validar se o campo foi realmente criado no *form*:

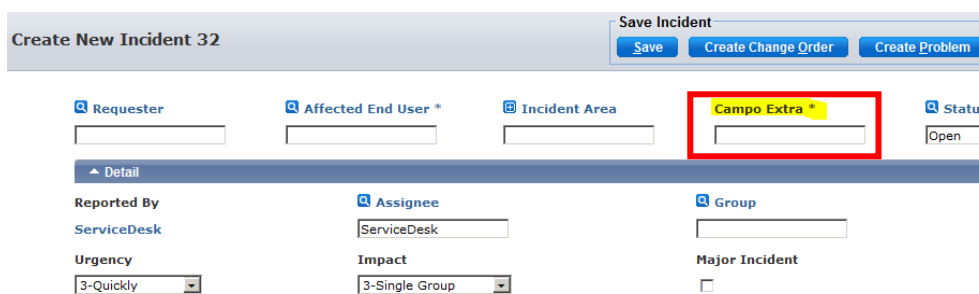


Figura 95 - Novo campo no form.


```

Module: z_cr_rootcause.spl
Author: dtsousa
Created: 2014-07-14

Set root_cause to required when the Request status is updated from 'Resolved'
for the user role 'L2 Analyst MS'.

Version 1.00

cr::z_cr_rootcause(...) {
    string log_user, role_analyst;
    role_analyst = "L2 Analyst MS";

    log_user = cnt::get_login_user_role_name();

    if (log_user == role_analyst)
    {
        if (is_null(rootcause))
        {
            set_error(1);
            set_return_data("Para poder resolver o registo é necessário preencher o campo Root Cause.");
            return;
        }
    }
}
    
```

Figura 98 - Exemplo de ficheiro .spl (definição da ação).

Este Spell é invocado sempre que há alteração de estado do *ticket* para “Resolved”. Depois de invocado, é efetuada uma verificação da *role* que está a ser utilizada para se realizar a alteração de estado. Caso esta alteração seja feita por um utilizador com a *role* “L2 Analyst MS” vai obrigar este a preencher o campo “Root Cause”.

10.1.7 Monitorização do Tomcat

O Tomcat consiste num servidor web java, que na plataforma de CA SDM tem como função mais importante fazer uma gestão dos ficheiros que estão anexados aos *tickets*, ou seja, é através dele que é possível fazer o *download/upload* de ficheiros.

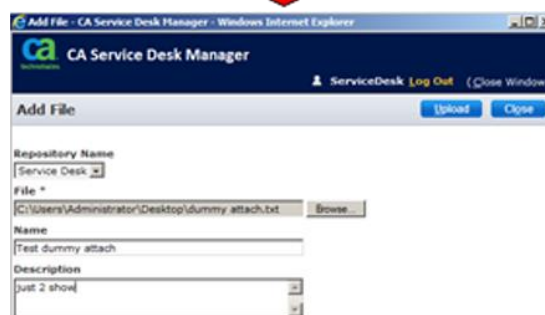
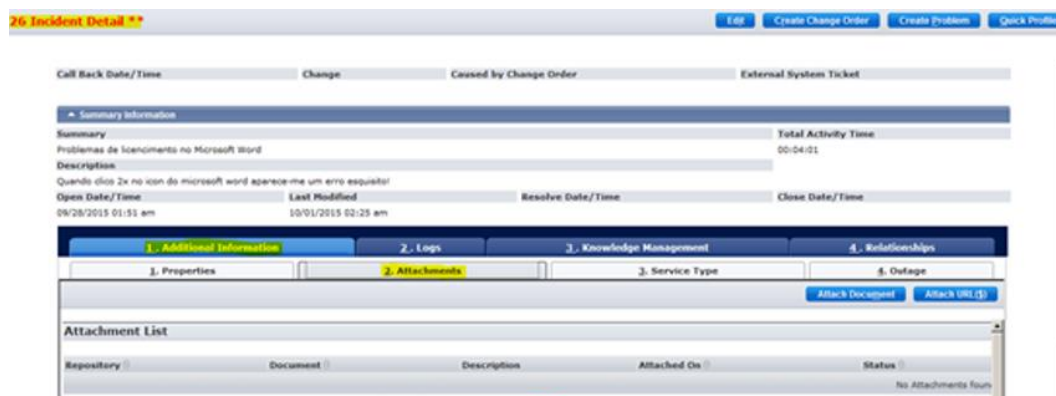


Figura 99 - Procedimento para se anexar ficheiro a ticket.

Por vezes a função de *download/upload* de ficheiros falha e é importante ser-se rápido a detetar o problema assim como de se proceder à sua resolução.

Normalmente problemas deste tipo são detetados através de duas formas:

- Através de um incidente por parte do utilizador a reportar que estão a visualizar uma janela deste tipo ao clicar no botão de *download* de *attachment*,



Figura 100 - Erro que surge quando existem problemas no Tomcat.

- Através de um ficheiro específico de *log* relativo ao Tomcat, identificando-se através de erros do tipo descrito na seguinte figura.

```

10/01 02:32:20.75 WIN2008R2 web:local 3452 SIGNIFICANT session.c 10282 Session 2036426650:0x03FCC798 login by analyst ServiceDesk (cnt:92A0659CCF76FA4C
10/01 02:34:06.09 WIN2008R2 tty_nxd 4180 SIGNIFICANT tty_nxd.c 631 TTY Statistics - Queue requests(11) : projections set(0)
10/01 02:35:02.16 WIN2008R2 boplogin 1992 SIGNIFICANT bplaccess.c 5267 AHD04012:Security id () not found or has expired
10/01 02:35:02.16 WIN2008R2 rep_daemon 1296 ERROR DomWrap.c 928 Failed to complete reply method: 'MTH::validate_bopid_done' ,BOP Name:'Attacher
10/01 02:35:02.18 WIN2008R2 web:local 3452 ERROR session.c 3258 Form Id missing from cgi
10/01 02:35:02.18 WIN2008R2 web:local 3452 SIGNIFICANT session.c 3540 URI from session 2036426650 POSTDATA: Error_Msg=AHD04012:Security%20id%20()%20nc
10/01 02:35:02.71 WIN2008R2 web:local 3452 ERROR session.c 3258 Form Id missing from cgi
10/01 02:35:02.71 WIN2008R2 web:local 3452 SIGNIFICANT session.c 3540 URI from session 2036426650 POSTDATA: Error_Msg=AHD04012:Security%20id%20()%20nc
10/01 02:35:02.72 WIN2008R2 web:local 3452 ERROR session.c 3258 Form Id missing from cgi
10/01 02:35:02.72 WIN2008R2 web:local 3452 SIGNIFICANT session.c 3540 URI from session 2036426650 POSTDATA: Error_Msg=AHD04012:Security%20id%20()%20nc
10/01 02:35:02.77 WIN2008R2 web:local 3452 ERROR session.c 3258 Form Id missing from cgi
10/01 02:35:02.78 WIN2008R2 web:local 3452 SIGNIFICANT session.c 3540 URI from session 2036426650 POSTDATA: Error_Msg=AHD04012:Security%20id%20()%20nc
10/01 02:35:41.89 WIN2008R2 pdm_tomcat 4276 SIGNIFICANT pdm_tomcat.c 1388 Stopping SERVICEDESK Tomcat javaw process (pid=4512): "C:\Program Files (x86)\CA

```

Figura 101 - Erro no log quando existem problemas no Tomcat.

Após se tomar conhecimento deste problema, procede-se da seguinte forma:

1. Aceder, de forma individual à plataforma de cada servidor aplicacional. Isto porque existe balanceamento de carga, sendo necessário simular o problema em todos eles de forma a encontrar o servidor ou servidores que se encontram com problemas:
2. Depois de encontrado o servidor que manifesta o problema, é necessário aceder-lhe e ir à linha de comandos executar o seguinte comando:

```
pdm_status | findstr tomcat
```

```

C:\Users\Administrator>pdm_status | findstr tomcat
CA SD Tomcat (pdm_tomcat_nxd) Running WIN2008R2 4276 Thu Oct 01
:24:06
Visualizer (pdm_tomcat_nxd) Running WIN2008R2 4256 Thu Oct 01
:24:06

```

Figura 102 - Método de verificação dos estados dos processos Tomcat.

Caso o estado esteja diferente de “*Running*” significa que o processo terminou e que necessita de ser restaurado (irá mostrar-se brevemente como o fazer).

Por vezes acontece que ao executar o comando anteriormente mencionado, os processos apresentam o seu estado a “*Running*” mas o problema persiste. Neste caso será necessário proceder a uma análise dos processos a correr na máquina:

sq_agent_nxd.exe	1816	5,224 K	12,328 K	18:37:53 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\sq_agent_nxd.exe	
spel_srvr.exe	3504	0.16	49,372 K	55,276 K	18:37:53 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\spel_srvr.exe
reo_daemon.exe	3216		3,136 K	7,140 K	18:38:19 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\reo_daemon.exe
pdm_tomcat_nxd.exe	3472		2,568 K	6,924 K	18:38:25 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_tomcat_nxd.exe
pdm_tomcat_nxd.exe	4448		2,572 K	6,944 K	18:38:26 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_tomcat_nxd.exe
pdm_tomcat_nxd.exe	5048		2,596 K	6,944 K	18:38:27 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_tomcat_nxd.exe
pdm_tomcat_nxd.exe	4824		2,580 K	6,892 K	18:38:43 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_tomcat_nxd.exe
pdm_text_nxd.exe	3076		0,000 K	11,232 K	18:38:13 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_text_nxd.exe
pdm_rfbroker_nxd.exe	4860		2,368 K	6,204 K	18:38:04 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_rfbroker_nxd.exe
pdm_proctor_nxd.exe	5372		3,084 K	6,612 K	18:37:40 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_proctor_nxd.exe
pdm_mail_nxd.exe	6068		5,620 K	10,828 K	18:37:52 2015-09-...	D:\Program Files (x86)\CA\Service Desk Manager\bin\pdm_mail_nxd.exe

Figura 103 - Listagem dos processos Tomcat utilizando o "Process Explorer" .

Ao ser analisada a lista de processos, há que ter especial atenção ao processo “*pdm_tomcat_nxd.exe*”. Como referido anteriormente, o Tomcat é um servidor web java, mas ao analisar cada um destes 4 processos de Tomcat, pode-se verificar não existir um processo *javaw.exe* a correr sob cada um deles (note-se que cada processo de Tomcat já se encontra expandido). Com a falta do processo java a correr, estes processos Tomcat ficaram órfãos, daí aparecerem com o estado “*Running*” quando se fez o *pdm_status*.

- Depois de serem analisadas as possíveis causas deste problema, irá então proceder-se à sua solução. Para tal é necessário realizar um restauro do processo executando os comandos presentes na seguinte figura.

```

Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ >pdm_tomcat_nxd -c STOP

C:\Users\ .pdm_tomcat_nxd -c START

C:\Users\ >_

```

Figura 104 - Método para efetuar restart aos processos Tomcat.

Importante: Ao fazer stop é importante validar na lista de processos que os relativos ao Tomcat morreram mesmo.

10.1.8 Monitorização e resolução de sessões bloqueadas

A arquitetura utilizada pelo cliente é a da alta disponibilidade. Deste modo, são utilizados quatro servidores aplicativos com distribuição de carga, tendo cada um

deles uma capacidade aproximada de 200 utilizadores. Por vezes, um ou mais dos quatro servidores poderão ficar sobrecarregados conduzindo ao bloqueio das sessões. Normalmente este problema é detetado através das seguintes abordagens:

- Erros nos *logs* da plataforma do CA SDM:

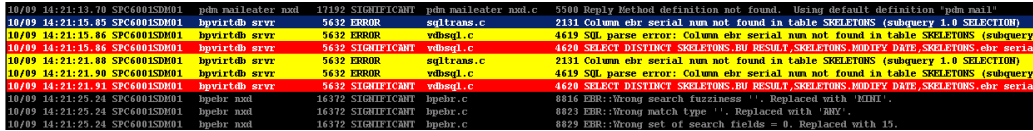


Figura 105 - Erro no log referente às sessões bloqueadas nos servidores.

- Quando é reportado que o utilizador não consegue criar incidentes e/ou reclamações (este clica mas nada acontece):



Figura 106 - Erro na criação de incidentes/reclamações nas sessões bloqueadas.

Inicialmente quando esta situação é reportada ou detetada é necessário proceder à semelhança de quando existem problemas nos anexos. Ou seja, será necessário entrar individualmente na plataforma CA SDM dos servidores aplicativos e verificar qual ou quais deles se encontram bloqueados (basta tentar criar um incidente). Depois de detetados quais os servidores que não estão a funcionar corretamente, bastará aceder remotamente e correr o comando *pdm_webstat* para se conhecer quantos utilizadores se encontram ligados àquele servidor:

```
=====  
Report from Webengine: web:local  
=====  
Cumulative sessions so far = 9182  
Most sessions at a time   = 232  
Currently active sessions = 136
```

Figura 107 - Comando *pdm_webstat* para visualização das sessões ativas.

Em seguida há que retirar esse servidor do balanceador. Para tal é necessário desativar a opção *'listen'* no IIS de forma que esse servidor não seja considerado no balanceamento. Deste modo evitar-se-á que os utilizadores o utilizem enquanto está ser efetuada uma pequena intervenção.

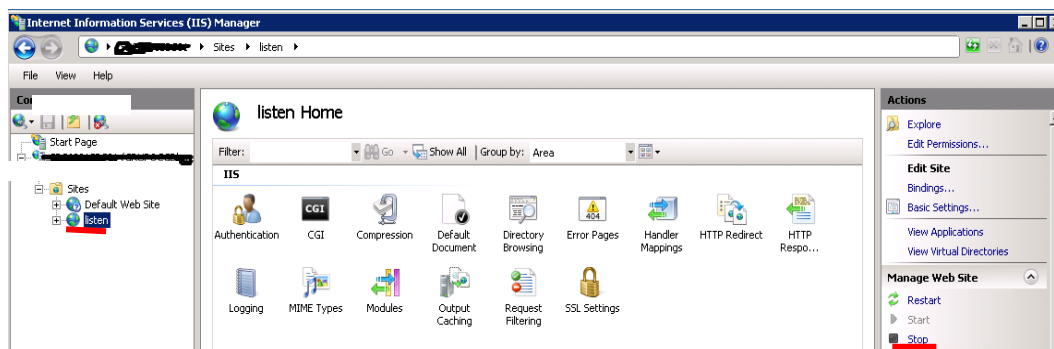


Figura 108 - Descrição de como retirar o servidor do balanceamento.

Após retirar o servidor do balanceamento é necessário aguardar de modo a que número de ligações ao servidor decresça. Após esse tempo de espera, bastará fazer uma *restart* à plataforma que poderá ser feito das seguintes formas:

- Através do comando: `net start pdm_daemon_manager`;
- No caso em que se pretenda avisar os utilizadores, pode-se utilizar o comando: `pdm_server_control -q <tempo em segundos> -s <hostname da máquina>`.

10.2 Desenvolvimento de aplicação cliente Java (WSCASDMClient)

Por vezes é necessário que dois ou mais CA SDM diferentes comuniquem entre eles. Para que tal aconteça é necessário fazer-se uma integração. Esta irá consistir no processo de transferência de informação de um CA SDM para outro recorrendo a *web services*.

Por exemplo, quando um incidente é criado, é despoletado um evento e macro (que são previamente criados para esse efeito), passando a informação (e.g. a descrição, sumário, grupo) para variáveis de ambiente. Após esta passagem é invocado um cliente que vai proceder à leitura dessas variáveis, e consoante as regras de negócio que estiverem implementadas, envia-las para o CA SDM destino recorrendo a *web services* (16 e 17).

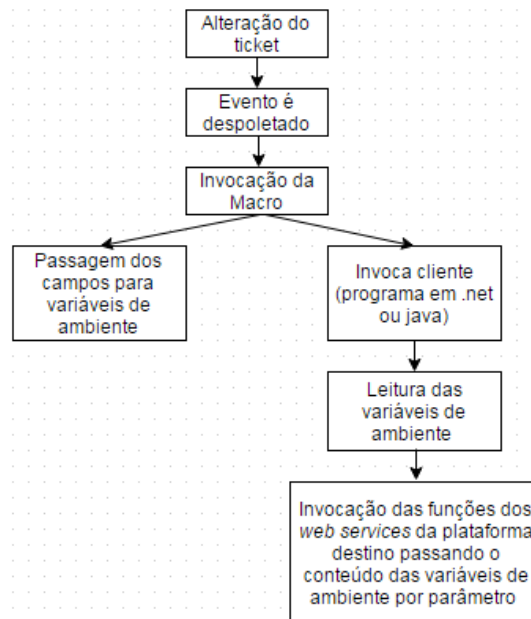


Figura 109 - Esquema de uma integração entre plataformas SDM.

Algum tempo após ter sido iniciado o trabalho nesta área, surgiu um projeto de integração entre várias plataformas de CA SDM. Um dos requisitos da integração será a existência de um cliente que interaja com os *web services* do produto. No caso particular da empresa onde foi desenvolvido o presente trabalho, o cliente utilizado para enviar os dados para os *web services* foi construído em .NET. No entanto, nem todos os servidores dos clientes que serviam a plataforma corriam Windows. Deste modo, noutros servidores como será exemplo os servidores *Unix*, não seria possível correr o cliente desenvolvido em .NET.

Foi por isso necessário desenvolver um cliente recorrendo à tecnologia java. Este deveria ter a capacidade de efetuar uma pequena seleção de ações que são realizadas no CA SDM:

- Criação de *log comment*,
- Criação de *quick ticket (incident)*;
- Criação de *request*,
- Atualização de *ticket*,
- *Do select* e *do query* (estas duas ações diferem apenas no tamanho do *output*) (16 e 17).

Após efetuada alguma pesquisa e exploração, foi possível desenvolver um cliente que foi designado "*WSCASDMClient*". Este cliente invoca funções que estão definidas e publicadas pelo *web service* do produto, através da *Web Services Description*

Language (WSDL), passando por parâmetro os vários valores dependendo da ação que se pretendia efetuar. Por exemplo, no caso da ação de adicionar um *log comment*, é importante passar a informação de qual o *ticket* em que se pretende adicionar o comentário, assim como o conteúdo do mesmo.



Figura 110 - Visualização da página WSDL da plataforma.

Em seguida serão descritos de forma resumida os passos que foram necessários para a construção deste cliente:

1. Em primeiro lugar foi necessário proceder à importação das funções que se encontravam publicadas no WSDL. Para tal foi necessário aceder à pasta *C:\ProgramFiles\Java\jdk1.8.0_05\bin*.

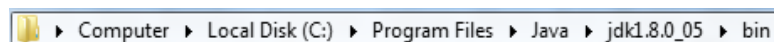


Figura 111 - Localização do daemon para importação das funções do WSDL.

Nesta foi executado o seguinte comando:

```
wsimport -keep -verbose http://<hostname máquina que tem o WSDL>:8080/axis/services/USD_R11_WebService?wsdl
```

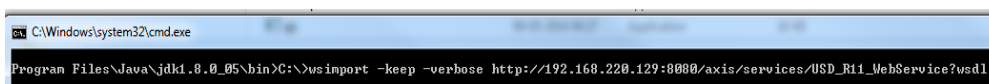


Figura 112 - Comando para a importação das funções do WSDL.

Depois de finalizado o processo, será criada uma pasta contendo todas as funções (ficheiros .java) disponibilizadas pelo WSDL.

- De seguida, no NetBeans (IDE que foi escolhido para se construir o cliente), criou-se uma package contendo todas as funções que foram retiradas através do comando anteriormente referido.

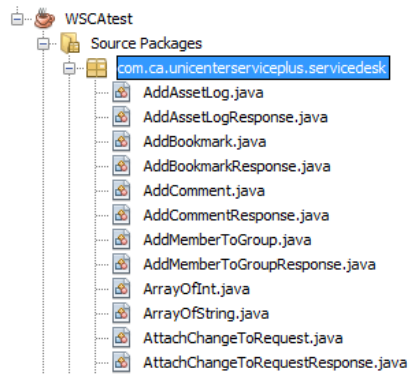


Figura 113 - Conteúdo da package resultante da execução do comando *wsimport*.

- Criou-se também outra package que irá conter a definição do cliente propriamente dito:

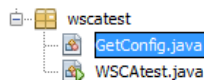


Figura 114 - Ficheiros que constituem a package da aplicação cliente.

Esta package é constituída por 2 ficheiros:

- *GetConfig.java*
- *WSCAtest.java*

A classe *GetConfig* tem como responsabilidade ler um ficheiro .xml de configuração de ligação ao *web service* destino. Optou-se pela criação de um ficheiro de configuração, pois como o projeto de integração engloba a interação entre mais do que 2 servidores, considerou-se que seria mais fácil realizar a alteração dos dados necessários para se efetuar essa ligação num ficheiro à parte em vez de se alterar diretamente no código. Este ficheiro basicamente contém a informação referente ao *username*, *password* e URL do WSDL:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <Usd.Username>servicedesk</Usd.Username>
  <Usd.Password> </Usd.Password>
  <Usd.Url>http://192.168.220.129:8080/axis/services/USD_R11_WebService?wsdl</Usd.Url>
</configuration>
```

Figura 115 - Ficheiro de configuração da aplicação.

Por sua vez a classe *WSCAtest* irá ser a responsável por:

- Invocar a leitura do ficheiro de configuração;
- Preparar e efetuar a ligação ao *web service* utilizando os objetos/funções que foram importados do WSDL utilizando *Simple Object Access Protocol (SOAP)*:

```
private URL url;
private final USDWebService service;
private final USDWebServiceSoap soapService;
private int sid;

...

public WSCAtest() {
    try {
        this.url = new URL(info.get_Url());
    } catch (MalformedURLException ex) {
        System.out.println(ex);
    }
    service = new USDWebService(url);
    soapService = service.getUSDWebServiceSoap();
}
```

Figura 116 - Código da classe WSCAtest (ligação ao ws).

- Quando se efetua a ligação a um *web service*, é necessário gerar-se um SID (*session ID*), desta forma foi também necessário criar a função para este efeito:

```
//Getting the SID
public int get_SID() {
    sid = soapService.login(info.get_Username(), info.get_Password());
    return sid;
}
```

Figura 117 - Função utilizada para se obter o SID para ligação ao ws.

- Chamar as funções que foram criadas para realizar as ações previamente requeridas.

4. Segue-se a definição de uma das ações requeridas (criação de *Request*):

```

//Creating new request - Working
public void create_Request() {
    List<String> r_list = new ArrayList<>();

    r_list.add("customer");
    r_list.add("cnt:96FF1D388A5BE24883221B30C544F954");

    r_list.add("summary");
    r_list.add("Request by Duarte Sousa and his amazing client");

    r_list.add("description");
    r_list.add("Olá, este foi um request criado pelo cliente WSCASDMDApp.");

    ArrayOfString attrVals = new ArrayOfString();
    attrVals.getString().addAll(r_list); //important!

    soapService.createRequest(sid,
        "",
        attrVals,
        emptyarray,
        "",
        emptyarray,
        emptyholder,
        emptyholder,
        emptyholder);
}

```

Figura 118 - Função de criação de Request.

Como se pode verificar, na função de criar *Request* foi necessário desenvolver uma lista. Nessa lista é obrigatório inserir os valores em pares. Assim, em primeiro lugar é necessário referir qual o campo do CA SDM em que se pretende inserir algo, e em seguida indicar o seu conteúdo. No exemplo apresentado, quando o *Request* for criado irá ter o *customer/affected end user* com aquele ID, o sumário "Request by Duarte Sousa and his amazing cliente" e a descrição: "Olá, este foi um request criado pelo cliente WSCASDMDApp."

Em seguida foi necessário passar o conteúdo dessa lista para o interior de um *ArrayOfString* através de um cast. Esta conversão é necessária pelo facto da função que vai ser invocada no *web service* só aceitar como parâmetro este tipo de estrutura. Na seguinte figura é apresentado o que acontece quando o cliente é executado:

```

run:
Starting client...
Generating session id...
Session id generated!Ready to perform action on CA CDM
Task done...
Client ended...
BUILD SUCCESSFUL (total time: 5 seconds)

```

Figura 119 - Execução do cliente.

No CA SDM, fazer uma pesquisa por todas os incidentes ativos:

Request #	Tenant	Summary	Priority	Category	Status	Assigned To	Projected Violation
+ 104	WeDo	Request by Duarte Sousa and his amazing client	None		Open	ServiceDesk	
+ 102	WeDo	Request by Duarte Sousa	None		Open	ServiceDesk	
+ 101	WeDo	Request by Duarte Sousa	None		Open	ServiceDesk	
+ 100	WeDo	Request by Duarte Sousa	None		Open	ServiceDesk	
+ 53	WeDo	Request by User2	None		Open	ServiceDesk	
+ 33	Blitz	This is Test 1.	3		Open		
+ 28	Mainroad	This is Test 3.	None		Open		
+ 27	Mainroad	This is Test 2.	2		Open		
+ 26	Mainroad	This is Test 1.	3		Open		

Expand All (\$) 1-9 of 9



104 Request Detail
[Edit](#) [Create Change Order](#) [Create Incident](#) [Quick Profile](#)

Tenant: WeDo

Requester	Affected End User	Request Area	Status	Priority
	User2		Open	None

Detail

Reported By	Assignee	Group	Configuration Item
ServiceDesk	ServiceDesk		
Severity	Urgency	Impact	Active?
		None	YES
Charge Back ID	Call Back Date/Time	Resolution Code	Resolution Method
Change	Caused by Change Order	External System Ticket	

Summary Information

Summary	Total Activity Time		
Request by Duarte Sousa and his amazing client	00:00:00		
Description			
Oii, este foi um request criado pelo cliente WSCASDMap.			
Open Date/Time	Last Modified	Resolve Date/Time	Close Date/Time
10/08/2015 02:44 am	10/08/2015 02:44 am		

Figura 120 - Resultado após execução do cliente (no CA SDM).

10.3 CA Service Catalog (CA SC)

O SC foi a segunda plataforma que no âmbito do presente trabalho foi necessário aprofundar. De notar que o primeiro projeto no cliente foi realizado nesta plataforma.

10.3.1 Requisitos

Neste caso foi requerido pelo cliente a existência de um único ponto de contacto de fácil comunicação entre os *end users* e os analistas (que têm como principal missão responder às suas solicitações). Anteriormente, duas ferramentas eram utilizadas para satisfazer estas necessidades, mas apresentavam algumas inconsistências, e não eram tão intuitivas quanto o que se pretendia. Por este motivo optou-se pelo *Service Catalog*.

Nesta plataforma seriam então apresentados todos os serviços a ser solicitados pelos vários departamentos/*business units* pertencentes ao cliente. Essas solicitações necessitariam de aprovação hierárquica (aprovação feita pelo manager da pessoa que fez a solicitação), e/ou técnica (aprovação por parte da equipa responsável por resolver essa solicitação).

10.3.2 Configurações implementadas

Após analisados todos os requisitos, decidiu-se organizar a plataforma da seguinte forma: Pastas Gerais, Áreas de Negócio, Serviços, Opções e Informações de Esclarecimento, nas quais:

- Cada pasta geral é constituída por áreas de negócio;
- Cada área de negócio é constituída por serviços;
- Cada serviço é constituído por opções de submissão, opções informativas e informações de esclarecimento.

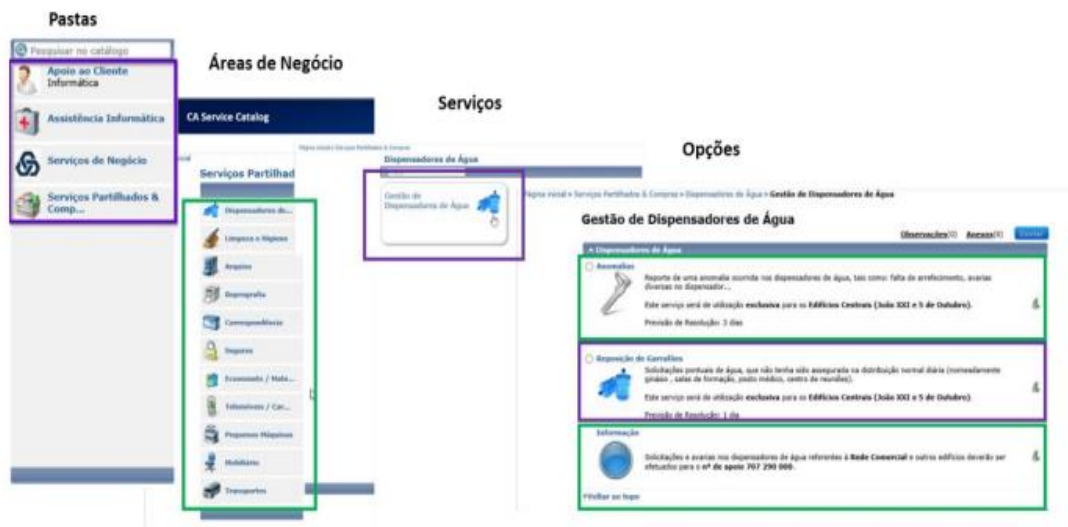


Figura 121 - Organização do projeto de SC do cliente.

Assim sendo, o trabalho desenvolvido nesta plataforma pode resumir-se no que se apresenta na seguinte tabela:

Criação e configuração	
Service Catalog	Service Desk
7 Pastas gerais	≈ 500 <i>Task's</i>
85 Áreas de negócio	≈ 370 <i>Change Orders</i>
≈ 260 Serviços	≈ 380 <i>Incident Areas</i>
≈ 760 Opções/forms	<i>Workflows, Status, Behaviors...</i>

Numa primeira fase do projeto, decidiu-se escolher somente duas direções (*business units*) de forma a criar-se um piloto.

Para tal, foi feito um levantamento de todos os serviços que constituíam a ferramenta que estava a ser utilizada anteriormente à implementação do *Service Catalog*. Com o auxílio dos coordenadores dessas duas direções, criaram-se documentos em que o

objetivo de cada um deles seria caracterizar o serviço a ser configurado na nova plataforma. Cada um desses documentos continha dados como:

- A informação que se considerasse necessária e relevante para a resolução da solicitação, a qual seria preenchida pelo *end user*. Por outras palavras seria a forma como os *forms* deveriam ser construídos, como por exemplo o tipo de componente a ser utilizado (e.g. *texto fields*, *check boxes*, *droplists*), a sua nomenclatura, a obrigatoriedade de preenchimento, validações necessárias e comportamento dinâmicos (recorrendo a *scripting*);
- A IA (caso se tratasse de um incidente) ou *Change Category* (se tratasse de um pedido) que serviria como categorização dessa solicitação;
- O grupo responsável pela resolução dessa solicitação, caso fosse um incidente;
- Se a solicitação fosse um pedido, o documento continha a definição do *workflow* a ser implementado. Ou seja, quais as *tasks* necessárias, e respetivos grupos, que seriam necessários para a resolução da mesma.
- SLA's a serem implementados para as diferentes solicitações.

Após receção destes documentos, procedeu-se então à configuração dos serviços na plataforma. Para tal, tendo em conta que estes serviços teriam de ser disponibilizados para os vários departamentos do cliente, onde a disponibilidade do serviço poderia variar entre ambientes, decidiu-se criar uma BU com o nome do cliente. É nesta BU que são criados pela primeira vez todos os serviços, grupos de opções, opções e respetivas associações. Ou seja, é aqui que se encontra o aglomerado de todos os serviços que podem ser utilizados. Em seguida criaram-se várias BU's em que cada uma delas representa o departamento do cliente. Desta forma, consoante os requisitos de disponibilização de serviço, passou-se por herança os serviços para esta nova BU.

Após ter-se dado por concluída a implementação deste piloto em ambiente de QLD, iniciou-se um período de testes. Nestes foram incluídos os vários utilizadores desses departamentos, com o objetivo de se detetarem problemas (e.g. erros nos *forms*, níveis de aprovações incorretos, erros nos *workflows*), e deste modo se proceder à sua resolução e à implementação de melhorias.

Com a conclusão da fase de testes e normalização, passou-se as mesmas configurações da plataforma para o ambiente de PRD. Iniciou-se então um processo de *rollout* cujo objetivo foi segregar para todos os departamentos do cliente, cujo número se aproxima atualmente dos 50.

Relativamente à evolução da implementação deste projeto, a tarefa mais complexa realizada foi a criação dos *forms* para cada opção do serviço. Isto porque é necessário recorrer-se a *scripting* de forma a se implementarem comportamentos dinâmicos e de validações. Em seguida é apresentado um exemplo de *scripting* criado para um form:

```
{
iniFields : function(){
  if (_.request.status > 100 || _.request.status == 2 || _.request.status == 4){
    var getspfvalue = ca_fdGetTextFieldValue(ca_fd.formId,
'_txtf_get_spf_value');
    ca_fdShowField(ca_fd.formId, 'sel_normal');

    if(getspfvalue == 'normal'){
      ca_fdHideField(ca_fd.formId, '_spf_urgente_Azul');
      ca_fdHideField(ca_fd.formId, '_spf_qtd_normal_registro');
      ca_fdShowField(ca_fd.formId, '_spf_qtd_normal');
    }
    if(getspfvalue == 'normal_registro'){
      ca_fdHideField(ca_fd.formId, '_spf_urgente_Azul');
      ca_fdShowField(ca_fd.formId, '_spf_qtd_normal_registro');
      ca_fdHideField(ca_fd.formId, '_spf_qtd_normal');
    }
    if(getspfvalue == 'urgente_azul'){
      ca_fdShowField(ca_fd.formId, '_spf_urgente_Azul');
      ca_fdHideField(ca_fd.formId, '_spf_qtd_normal_registro');
      ca_fdHideField(ca_fd.formId, '_spf_qtd_normal');
    }
  }else{
    ca_fdHideField(ca_fd.formId, 'sel_normal');
    ca_fdHideField(ca_fd.formId, '_spf_urgente_Azul');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal_registro');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal');
  }
}
makeReadOnly('sel_tipo_envio');
makeReadOnly('sel_normal');
},

viewMap: function() {
  var img_url = '';
  img_url = 'FileStore/custom/documents/Zonas_2015_TD_PT_Final_DESB.pdf';
}
```

```
openWinReport(img_url, 438, 1154);
},
showOnTipoEnvio : function(){
var getTipologia =
ca_fdGetSelectedOptionValues(ca_fd.formId,'sel_tipo_envio')[0].toUpperCase();
  if(getTipologia == "NORMAL"){
    ca_fdShowField(ca_fd.formId, 'sel_normal');
    ca_fdHideField(ca_fd.formId, '_spf_urgente_Azul');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal_registro');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal');
  }else{
    ca_fdHideField(ca_fd.formId, 'sel_normal');
    ca_fdShowField(ca_fd.formId, '_spf_urgente_Azul');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal_registro');
    ca_fdHideField(ca_fd.formId, '_spf_qtd_normal');
    ca_fdSetTextFieldValue(ca_fd.formId, '_txtf_get_spf_value', 'urgente_azul');
  }
},
submit : function(){
  saveFormFields();
  return true;
},
writeSPFValue : function (spinner){
  var getspnval = ca_fdGetTextFieldValue(ca_fd.formId, spinner);

  ca_fdSetTextFieldValue(ca_fd.formId, 'txtf_getqtd', getspnval);
},
validateFields : function(){
  var getOp =
ca_fdGetSelectedOptionValues(ca_fd.formId,'sel_tipo_op')[0].toUpperCase();
  var check = false;
  if(getOp == 'CRIAR'){
    var getCountTableRows=ca_fdGetTableRowCount(ca_fd.formId,'tbl_utilizadores');
    var getCountTableRowsSelected =
ca_fdGetSelectedTableRows(ca_fd.formId,'tbl_utilizadores').length;
    if(getCountTableRows == 0){
      alert('Por favor, insira o(s) registro(s) na tabela: Mais Utilizadores.');
```

```
      check = true
    }
    if(getCountTableRows!=0 && (getCountTableRows != getCountTableRowsSelected)){
      alert('Por favor, selecione o(s) registro(s) inserido(s) na tabela: Mais
Utilizadores.');
```

```
      check = true
    }
  }
}
```

```
    }  
  }  
  checkAttachments : function () {  
    var getAttachText = document.getElementById("attachments-1").innerHTML;  
    var getIni=getAttachText.lastIndexOf("(");  
    var getLast=getAttachText.lastIndexOf(")");  
  
    var getNumAttachs=parseInt(getAttachText.substr(getIni+1, getLast));  
  
    if(getNumAttachs==0){  
      alert('Por favor. anexo o ficheiro com a informação complementar.');      return false;  
    }else{  
      saveFormFields();  
      return true;  
    }  
  },  
  validateDate : function(insertedDate){  
    var getTodayDate = (new Date).getTime();  
    var getInsertedDate=ca_fdGetDateFieldValueInMillis(ca_fd.formId, insertedDate);  
  
    var getTodayDay = parseInt(new Date(getTodayDate).getDate());  
    var getInsertedDay = parseInt(new Date(getInsertedDate).getDate());  
  
    if(getInsertedDate != '' && $.request.status == -1 && (getInsertedDate >  
getTodayDate || getTodayDay == getInsertedDay)){  
      alert('Tem que inserir uma data anterior à de hoje.');      return false;  
      //ca_fdSetDateFieldValue(ca_fd.formId, insertedDate, null);  
    }  
  }  
}
```

Estas funções são depois associadas aos eventos que existem para cada componente, como se pode observar nas seguintes imagens:

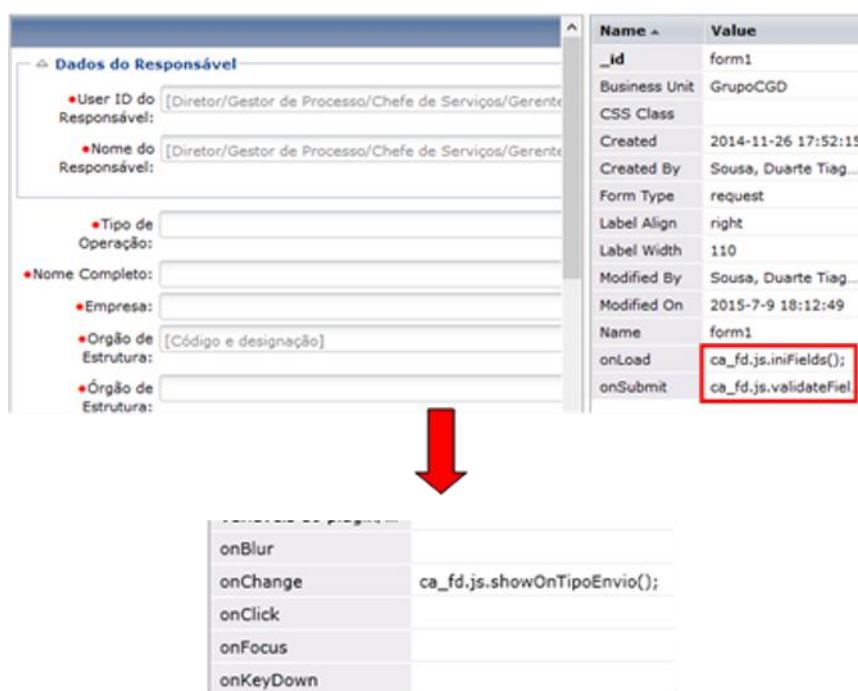


Figura 122 - Associação de scripting aos eventos das componentes do form.

10.3.3 Gestão da plataforma

Como descrito no capítulo anterior, aproximadamente 50 departamentos do cliente utilizam a plataforma de *Service Catalog* sendo por isso necessário fazer uma gestão da mesma.

Esta gestão consiste:

- **Suporte ao cliente:** quando este apresenta dúvidas de utilização referentes à plataforma ou reporta inconformidades da mesma;
- **Disponibilidade:** é importante garantir que a plataforma cumpra o nível de disponibilidade estipulada no contrato;
- **Monitorização:** Devido ao ponto anterior é necessário fazer-se uma monitorização ao nível dos processos que são fundamentais para que a plataforma se mantenha disponível e funcional;
- **Resolução de erros e implementação de melhorias:** Apesar de se dar por concluída a fase de implementação, por vezes encontram-se erros (e.g. *forms*, *scripting*, níveis de aprovações incorretos, erros nos *workflows*, processos PAM que falharam...). Além disso podem surgir sugestões de melhorias, quer do nosso lado, quer do lado do cliente. Quando se trata de um problema que seja inerente ao próprio produto e não de configuração, é necessário contactar o suporte do fabricante (CA) para que seja providenciado um *patch/test fix*,

- **Garantir que a plataforma se mantém na versão mais recente:** à semelhança de outros produtos de software, novas versões e *service packs* deste software saem ao longo do tempo, sendo desta forma importante realizar-se os *upgrades* da plataforma e garantir que estas se mantêm funcionais.

10.4 CA PAM e BOXI

As plataformas de PAM e de BOXI são aquelas onde se teve menor contacto durante a realização deste trabalho, dado que existem outros elementos da equipa responsáveis pelas mesmas. No entanto foram no âmbito destas realizadas várias tarefas.

A nível de PAM foram realizadas as seguintes tarefas:

- Diariamente verificar se ocorreu alguma falha na execução dos processos PAM. Caso isso aconteça será necessário descobrir a sua causa e tentar proceder à sua resolução. Caso não seja possível, o problema será reportado aos colegas responsáveis;
- Aplicação de melhorias nos processos, como por exemplo a alteração de expressões regulares e de parâmetros;
- Passagem de processos entre ambientes.

A nível de BOXI foram efetuadas as seguintes tarefas:

- Criação de *users* para acesso à plataforma;
- Criação e execução de relatórios de pequena/média complexidade;
- Configuração da periodicidade da execução de relatórios.

11 Conclusão

Com a elaboração deste relatório foi possível consolidar os conhecimentos adquiridos ao longo da realização do trabalho. Nomeadamente, foi necessário proceder à criação de uma VM, instalar e configurar as plataformas necessárias, assim como efetuar a simulação de todos os procedimentos que foram apresentados ao longo deste documento.

Relativamente aos resultados e objetivos, estes foram cumpridos de acordo com as prioridades da equipa. Com a entrega de relatórios periódicos e com a gestão de pedidos, incidentes e problemas verifica-se uma melhoria significativa na agilização de processos no cliente, o que possibilita o seu crescimento e maturidade. É importante notar que todo este processo é efetuado de uma forma contínua. Ou seja, os objetivos e prioridades estão a ser constantemente alterados.

A nível de *Service Desk*:

- A taxa de resolução de *tickets* sem violação de SLA's que são direcionados para a nossa equipa aproxima-se dos 98%;
- Os problemas que nos surgem diariamente relativamente a esta plataforma (e.g. níveis de *performance*, incoerências, falhas) encontram-se a ser resolvidos com celeridade;
- Recentemente foi realizado o *upgrade* da plataforma para a versão 14.1;
- Foi também colocada em PRD uma integração com a plataforma de ITSM de um parceiro do cliente, na qual estivemos envolvidos na fase de testes;
- São feitas regularmente alterações na plataforma para que esta satisfaça as regras de negócio do cliente.

A nível de *Service Catalog*:

- A plataforma continua em fase de disponibilização pelos vários departamentos do cliente, o que tem levado também a um alargamento do seu âmbito. Este interesse é indicador que o cliente está satisfeito, e desta forma pode-se afirmar que o principal objetivo do projeto de implementação foi cumprido;
- Os problemas reportados nesta plataforma estão a ser resolvidos de forma célere assim como a implementação de melhorias.

Os objetivos e prioridades desta área estão em constante alteração. Recentemente um dos departamentos pretendeu implementar a norma ISO 27001 (referente à gestão de licenciamento de software instalado nos diferentes *assets*). Deste modo, encontra-se atualmente a decorrer a aprendizagem de uma nova plataforma de ITSM denominada de CA ITAM (*IT Assets Management*), para que esta seja implementada e configurada de forma a satisfazer as necessidades do cliente.

Relativamente ao trabalho a ser realizado no futuro, este consistirá na continuação da gestão a nível de pedidos, incidentes e problemas de forma a haver uma maior agilização de processos, e conseqüente aumento do grau de maturidade no cliente. Adicionalmente continuará a ser efetuada a configuração e gestão de plataformas ITSM para a satisfação das necessidades do cliente.

A implementação de melhorias também faz parte do trabalho a ser efetuado futuramente, pois estas são identificadas e reportadas diariamente à nossa equipa com o intuito de melhorar a qualidade do serviço que está a ser prestado ao cliente. De notar que estas ferramentas ao serem instaladas não ficam imediatamente funcionais. Ou seja, carecem de um conjunto de alterações e configurações de forma a se adaptarem e satisfazerem a visão e os requisitos de negócio do cliente.

Por outro lado, existem inevitavelmente problemas associados aos próprios produtos (os denominados “*bugs*”). Por exemplo, a nível de *Service Desk*, no processo de atualização da versão, nas instalações de *cumulative patches* e de *service packs* é regular ocorrerem erros. Isto porque este procedimento realiza alterações na estrutura da CMDB assim como de registos e ficheiros (e.g. de configuração, *forms*). Devido a estas ações, algumas funcionalidades da plataforma poderão ser afetadas, assim como as integrações com outras ferramentas.

Por vezes não nos é possível resolver esses problemas e para tal é necessário contactar o suporte do fabricante (CA) para que seja criado um *test fix*. No entanto, apesar da instalação desses *test fixes* resolverem habitualmente um problema, será normal acabarem por surgir outros. Desta forma, temos um contacto constante e direto com o suporte para que sejam implementadas melhorias de forma contínua com o objetivo de se evitar o conjunto de situações que foram anteriormente mencionadas.

A nível de *Service Catalog* também se verificarão melhorias no futuro. Aquela com maior impacto é a referente à maneira que o *form* é apresentado após a sua submissão. Por exemplo, em certos *forms* utiliza-se *scripting* para se aplicar comportamentos dinâmicos, como mostrar e esconder certos campos consoante as opções que foram selecionadas. O problema surge após submissão do pedido/*form* uma vez que todos os campos ficarão visíveis, enquanto que o comportamento expetável seria serem só visíveis os campos que foram efetivamente preenchidos. Esta situação conduz a que os *end users* e analistas possam ficar confusos com a informação que é recebida.

De forma a solucionar aquele problema, foi desenvolvida uma função que será invocada sempre que seja carregada a página referente ao *form* submetido, e que irá verificar quais os campos que estão efetivamente preenchidos. Caso não o estejam, forçará que estes permaneçam no estado *hidden*. Esta função já se encontra construída e aplicada para alguns dos *forms*, sendo agora necessário aplica-la a todos os restantes (aproximadamente 400).

12 Bibliografia

- (1) <https://mainroad.pt> (*Online - reencaminhando na atualidade para o endereço <http://www.nossistemas.pt>, acessado a 02-05-2015*)
- (2) <http://searchcio.techtarget.com/definition/ITSM> (*Online - Atualizado, acessado a 21-08-2015*)
- (3) <http://www.italtraining.com/what-is-itol.asp> (*Online - Atualizado, acessado a 21-08-2015*)
- (4) <http://www.rightstar.com/what-is-the-difference-between-an-asset-and-a-configuration-item-ci/> (*Online - Atualizado, acessado a 10-09-2015*)
- (5) <http://posi.ist.utl.pt/cepei/ITIL%20-%20OSA.htm> (*Online - Atualizado, acessado a 21-08-2015*)
- (6) <https://www.quora.com/What-are-the-key-differences-between-a-product-and-a-service> (*Online - Atualizado, acessado a 19-12-2015*)
- (7) CA Service Desk Manager - Analyst Operation 200 - 00_Introduction
- (8) CA Service Desk Manager - Analyst Operation 200 - 01_Explore the CA Service Desk Web Interface
- (9) CA Service Desk Manager - Analyst Operation 200 - 02_Create and Search for Tickets
- (10) CA Service Desk Manager - Analyst Operation 200 - 03_Update Incidents, Problems, and Requests
- (11) CA Service Desk Manager - Analyst Operation 200 - 04_Edit and Monitor Change Orders
- (12) CA Service Desk Manager - Analyst Operation 200 - 05_Resolve and Close IN, PR, RE, and CO
- (13) CA Service Desk Manager - Administration 200 - 00_Introduction
- (14) CA Service Desk Manager - Administration 200 - 01_Administer CA Service Desk_V01
- (15) CA Service Desk Manager - Service Desk Modification 300
- (16) CA_SDM_Tech_Ref_ENU
- (17) CA_Workflow_API_Ref_Guide_ENU
- (18) CA MDB r1.5 Overview
- (19) CA Service Catalog - Svc_Cat_Admin_ENU