



Universidade do Porto  
Faculdade de Engenharia

**FEUP**



Sérgio Paulo da Silva Correia

# ENTERPRISE MANAGEMENT - PLATAFORMA DE GESTÃO DE SERVIÇOS PARTILHADOS

47.3)  
IC5202  
CORs

, 2005

**Faculdade de Engenharia da Universidade do Porto**  
**Licenciatura em Engenharia Informática e Computação**



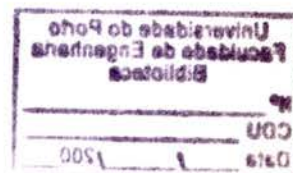
**Enterprise Management – Plataforma de Gestão de Serviços  
partilhados na Mainroad – Information Technology**

**Relatório do Estágio Curricular da LEIC <Ano lectivo>**

*Sérgio Paulo da Silva Correia*

Orientador na FEUP: Prof. Raul Oliveira  
Orientador na Mainroad: Eng. Pedro Leite

Setembro de 2005



004(047.3) LEIC/ETC 5202. 2005/COR 1

Universidade do Porto	
Faculdade de Engenharia	
Biblioteca	
Nº	81475
CDU	004.4(047.3)
Data	17/03/2006

*À minha mãe*



## Resumo

O presente relatório documenta o estágio realizado na Mainroad – Information Technology, no âmbito do Estágio Curricular da Licenciatura em Engenharia Informática e Computação (LEIC) da Faculdade de Engenharia da Universidade do Porto (FEUP).

A gestão de uma infra-estrutura complexa de sistemas informáticos, sem dispor de ferramentas de apoio, seria uma tarefa extremamente onerosa em termos de esforço de recursos humanos para ser economicamente recompensadora. Impõe-se por isso, a utilização de ferramentas adequadas para auxiliar na monitorização e gestão das infra-estruturas. O elevado nível de exigências dos clientes na prestação de serviços requereram uma reestruturação da plataforma de gestão utilizada.

A nova plataforma de gestão é constituída por um largo conjunto de ferramentas. Os módulos que a suportam são uma plataforma de Service Desk, uma plataforma de inventário, uma plataforma de monitorização, uma plataforma de geração e publicação de relatórios e um portal. O projecto de estágio tinha como objectivo implementar alguns dos componentes da nova plataforma de gestão.

O Service Desk é a plataforma responsável por gerir os problemas que surgem aos clientes da Mainroad. A plataforma de inventário tem como objectivo recolher os dados sobre os diversos componentes das infra-estruturas. A plataforma monitorização permite identificar os equipamentos das infra-estruturas e identificar os seus estados, prevenindo problemas e avarias. Esta plataforma avalia ainda o cumprimento dos serviços acordados com os clientes. A geração de relatórios e a sua respectiva publicação oferecem informações sobre os diversos componentes da plataforma de gestão. Por último, mas não menos importante, o portal possibilita o acesso a todas as ferramentas da plataforma. Esta aplicação disponibiliza ainda o acesso a todos os relatórios gerados.

Os requisitos funcionais e o desempenho inicialmente identificados para a plataforma de gestão foram cumpridos. Até ao momento, a plataforma implementada apoia mais de dez equipas de suporte na resolução de problemas e substitui a maioria dos processos manuais utilizados até à sua concepção.

## **Agradecimentos**

Apresento os meus agradecimentos a Pedro Emanuel Leite que me orientou e sempre se disponibilizou para me ajudar durante o estágio. Agradeço também a todas as pessoas da Mainroad que de várias formas me ajudaram a realizar o meu trabalho.

Agradeço ao professor Raul Teixeira Oliveira, que me orientou o estágio por parte da Faculdade.

Agradeço à minha família por todo o apoio que me deu durante o estágio.

Um agradecimento especial aos Panteras.

## Índice de Conteúdos

Introdução.....	1
1.1 Apresentação da Mainroad – Information Technology .....	1
1.2 O Projecto Plataforma de Gestão de Serviços Partilhados .....	2
1.3 Organização do Relatório.....	3
2 Plataforma de Gestão de Serviços Partilhados .....	4
2.1 Contexto do Projecto e Objectivos .....	4
2.2 Cronograma.....	6
3 Plataforma de Service Desk.....	7
3.1 Levantamento e Especificação de Requisitos .....	7
3.2 Implementação e Configuração.....	8
3.2.1 Base de Dados.....	8
3.2.2 Consola Administrativa.....	9
3.2.3 Perfis de Utilização .....	11
3.2.4 Importações .....	12
3.2.5 Interface .....	13
3.2.6 Integração com o Portal .....	14
3.3 Formações.....	14
4 Plataforma de Inventário .....	15
4.1 Funcionalidades .....	15
4.2 Arquitectura .....	16
4.3 Implementação e Configuração.....	17
4.3.1 Recolha de Inventário .....	17
4.3.2 Relatórios .....	18
4.4 Integrações.....	19
4.4.1 Portal.....	19
4.4.2 Service Desk.....	21
5 Plataforma de Monitorização .....	23
5.1 Network and System Management.....	23
5.1.1 Arquitectura.....	23
5.1.2 Implementação e Configuração .....	24
5.2 Database Management .....	26
5.2.1 Arquitectura.....	26
5.2.2 Implementação e Configuração .....	28
5.2.3 Interface Web.....	28
5.3 Exchange Management.....	28
5.4 SAP Management .....	28
6 Conclusões.....	29
6.1 Conclusões do projecto de estágio.....	29
6.2 Avaliação dos resultados do estágio .....	30
6.3 Perspectivas de trabalho futuro .....	30

Referências e Bibliografia .....	31
ANEXO A: Screenshots das aplicações .....	32
ANEXO B: Código.....	50
ANEXO C: Ficheiros gerados .....	72

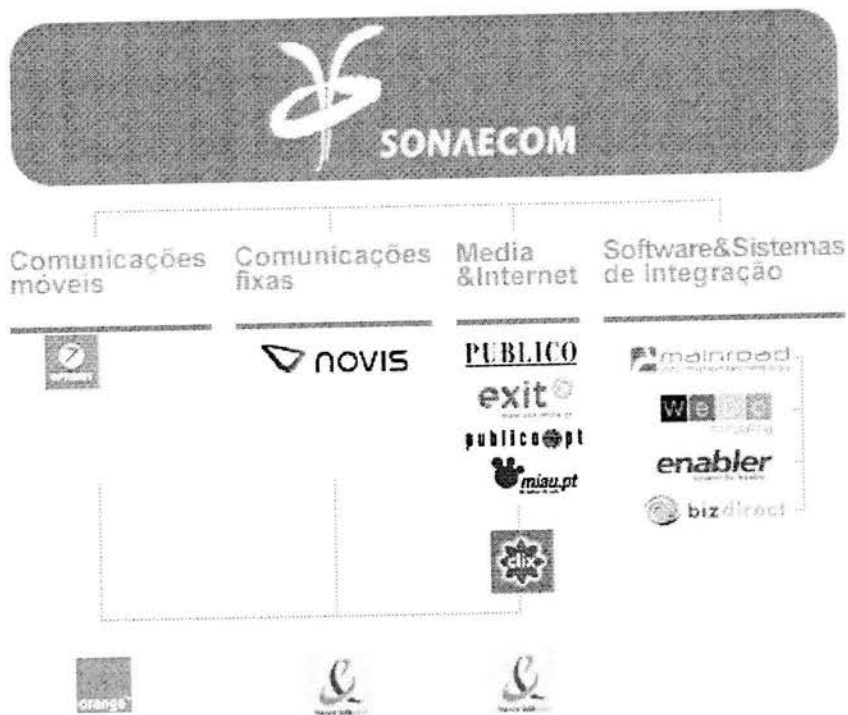


## Introdução

Neste capítulo é apresentada a empresa onde se realizou o estágio, contextualizados o projecto e o objectivo e descrita a organização do relatório.

### 1.1 Apresentação da Mainroad – Information Technology

A Mainroad é uma empresa inserida na área SSI (Software and Systems Integration) da Sonaecom. A Mainroad posiciona-se como um player em Tecnologias de Informação (IT) ao disponibilizar serviços em consultoria, implementação, gestão e suporte.



**Fig. 1 – Organograma da Sonaecom**

Fundada em 2003, a Mainroad é a única empresa no mercado nacional das IT que se dedica em exclusivo à prestação de serviços. Esta dedicação é sucedida através de equipas e recursos técnicos altamente especializados e certificados em soluções de referência junto dos principais fornecedores de hardware e software.

A Mainroad possui dois modernos Datacenters dispersos geograficamente com infra-estrutura especializada de alojamento para sistemas que necessitam de elevada performance e disponibilidade. A Mainroad oferece aos seus clientes um serviço de suporte contínuo, vinte e quatro horas por dia, sete dias por semana.

## 1.2 O Projecto Plataforma de Gestão de Serviços Partilhados

A gestão de uma infra-estrutura complexa de sistemas informáticos, sem dispor de ferramentas de apoio, seria uma tarefa extremamente onerosa em termos de esforço de recursos humanos para ser economicamente recompensadora. Impõe-se por isso, a utilização de ferramentas adequadas para auxiliar na monitorização e gestão das infra-estruturas. O elevado nível de exigências dos clientes na prestação de serviços requereram uma reestruturação da plataforma de gestão utilizada.

A nova plataforma de gestão é constituída por um largo conjunto de ferramentas novas e por outras existentes. Os módulos que a suportam são uma plataforma de Service Desk, uma plataforma de inventário, uma plataforma de monitorização, uma plataforma de geração e publicação de relatórios e um portal.

A plataforma de Service Desk veio substituir uma ferramenta antiga e limitada. A nova aplicação oferece um largo conjunto de funcionalidades inovadoras e adaptadas às necessidades das empresas ligadas às IT. A plataforma implementada é totalmente configurável, ajustando-se à realidade de cada empresa.

A plataforma de inventário veio complementar a plataforma utilizada. O conjunto de ferramentas desta plataforma obtém as informações dos equipamentos das infra-estruturas dos clientes da Mainroad. Os dados recolhidos correspondem ao hardware e software que os servidores e os equipamentos de rede possuem.

A plataforma de monitorização tem como objectivo avaliar o estado de todos os componentes da infra-estrutura, nomeadamente redes, sistemas, serviços e aplicações. Esta aplicação pretende prevenir falhas e avarias nos equipamentos, assegurando o seu correcto funcionamento. A plataforma de monitorização avalia ainda o cumprimento de serviço acordado com os clientes.

A geração de relatórios e a sua publicação estão ao cargo de um componente específico. Esta plataforma pretende disponibilizar informações sobre todos os componentes da plataforma de gestão. Os relatórios contêm informações sobre os registos do Service Desk, o inventário e sobre os componentes monitorizados.

O portal, por sua vez, tem como função possibilitar o acesso a todas as ferramentas da plataforma de gestão. O portal permite ainda o acesso aos relatórios gerados, devidamente organizados. Desta forma, o acesso a toda a informação é possível a partir de um único local.

### **1.3 Organização do Relatório**

O relatório é organizado em seis capítulos. Neste capítulo é feita a apresentação da empresa onde se realizou o estágio e do projecto proposto. No segundo capítulo é descrito o projecto, o seu contexto e os módulos que o envolvem. Os capítulos três, quatro e cinco destinam-se a explicar cada componente implementado da plataforma de gestão, nomeadamente o Service Desk, o inventário e a monitorização. No último capítulo são apresentadas as conclusões do trabalho realizado.

## 2 Plataforma de Gestão de Serviços Partilhados

A Mainroad decidiu renovar a sua plataforma de gestão. O elevado nível de exigência dos clientes na prestação de serviços, originou uma reestruturação da plataforma, substituindo componentes e acrescentando novas funcionalidades. O objectivo da nova plataforma de gestão é auxiliar as várias equipas da Mainroad no desempenho das suas funções, automatizando tarefas e processos.

Em conjunto com os seus parceiros tecnológicos, a Mainroad efectuou um estudo de mercado para identificar as soluções existentes para as necessidades dos seus clientes. Após analisar as propostas apresentadas, a Mainroad optou por utilizar a *Framework* Unicenter da Computer Associates (CA). O largo portefólio de produtos e a sua possível integração foram a razão principal desta adopção.

### 2.1 Contexto do Projecto e Objectivos

Uma empresa que presta serviços na área das Tecnologias da Informação tem de estar munida de ferramentas adequadas para poder oferecer um serviço de excelência aos seus clientes. O rápido avanço tecnológico obriga a actualização das infra-estruturas informáticas e dos processos utilizados para as gerir. A reestruturação da plataforma de gestão aparece para satisfazer às necessidades dos clientes, cumprindo o nível de serviço exigido e acordado.

A nova plataforma de gestão é constituída por um elevado número de componentes, nomeadamente por uma plataforma de Service Desk, uma plataforma de inventário, uma plataforma de monitorização, uma plataforma de níveis de serviços e por um portal.

A plataforma de Service Desk pretende assistir as equipas de suporte na resolução de problemas. Esta plataforma possui um largo conjunto de funcionalidades inovadoras e uma interface gráfica simples e intuitiva. O Service Desk é um dos componentes mais importantes da plataforma de gestão. O suporte contínuo que a Mainroad oferece aos seus clientes exige a estabilidade e a disponibilidade da ferramenta.

A plataforma de inventário pretende centralizar e armazenar todos os equipamentos das infra-estruturas dos clientes. Os processos de recolha e actualização tinham de ser automatizados, economizando esforços em termos de recursos humanos. Este módulo da plataforma de gestão é composto por três ferramentas, em que duas se destinam à arrecadação do inventário dos servidores e a outra para os equipamentos de rede.



A plataforma de monitorização pretende avaliar a disponibilidade e performance de todos os componentes das infra-estruturas. Esta plataforma é responsável por monitorizar redes, sistemas, serviços e aplicações. As ferramentas utilizadas para guarnecer este módulo são várias. Uma aplicação pretende avaliar a disponibilidade das redes e dos servidores. As restantes ferramentas encarregam-se de monitorizar a performance de bases de dados, servidores Exchange e servidores SAP.

A plataforma dos níveis de serviços tem como objectivo recolher métricas sobre o desempenho dos equipamentos monitorizados e analisar se os serviços contratados com os clientes estão a ser cumpridos.

Por último mas não menos importante, o portal é o componente que possibilita o acesso a todos os módulos da plataforma de gestão. O Portal é ainda responsável por gerar automaticamente vários relatórios e publicar os criados pelas restantes aplicações. Cada ferramenta da plataforma de gestão é integrada com o portal, divulgando os relatórios produzidos.

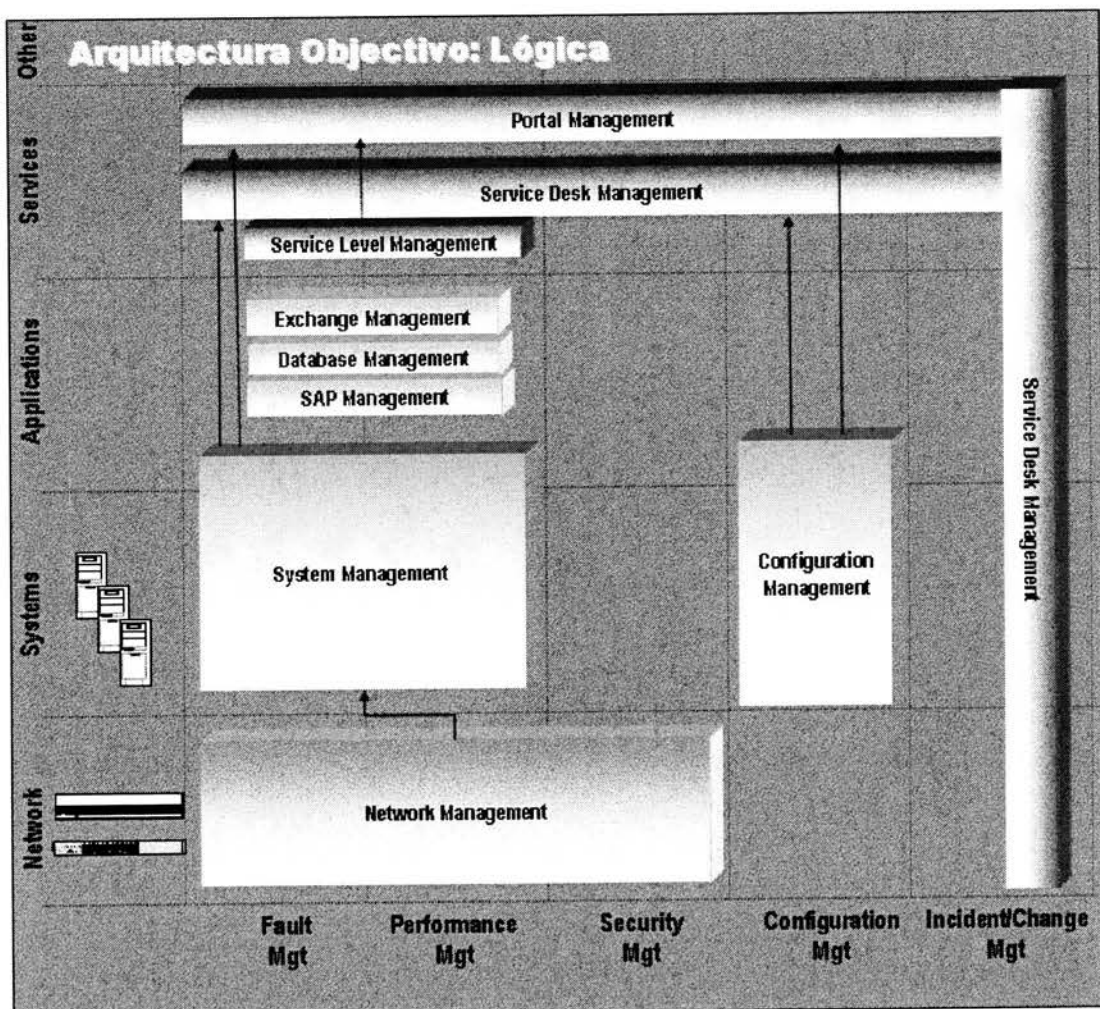


Fig. 2 – Arquitectura lógica da plataforma de gestão

## 2.2 Cronograma

A implementação da plataforma de gestão foi programada para estar concluída no fim do mês de Outubro. O tempo inicialmente estimado para a sua concepção foi de oito meses, correspondendo ao período de estágio acordado. A existência de um cronograma é fundamental, uma vez que permite enquadrar o trabalho desenvolvido no espaço temporal e verificar os avanços e atrasos na reestruturação da nova plataforma.

O Service Desk era primeira ferramenta a ser implementada e tinha seis semanas como tempo de desenvolvimento previsto. A segunda componente correspondia ao inventário. Estavam reservados três meses para recolher todo o inventário das infra-estruturas dos clientes e para automatizar os processos de recolha e de actualização. Pela sua natureza, a monitorização ocupa toda a duração do projecto. A monitorização das infra-estruturas é um processo trabalhoso e contínuo.

#	Tarefas/Mês	1	2	3	4	5	6	7	8
	<b>Macropiano - Objectivo</b>								
1	Uniformização de relatórios de serviço								
3	Cadastro de contratos de clientes								
6	Implementação de Service Desk								
2	Centralização do inventário								
5	Automatização de relatórios								
7	Implementação de visão de serviço								
8	Implementação de SLAs								
4	Consolidação da monitorização								

Fig. 3 – Cronograma

### **3 Plataforma de Service Desk**

A plataforma de Service Desk é um dos componentes mais importantes da plataforma de gestão. O suporte contínuo que a Mainroad presta aos seus clientes exige da plataforma de Service Desk estabilidade e disponibilidade.

A ferramenta utilizada para suportar esta plataforma foi o Unicenter ServicePlus Service Desk (USPSD). Esta aplicação é totalmente configurável, adaptando-se às necessidades de cada empresa. O USPSD possui inúmeras funcionalidades que apoiam as equipas de suporte na resolução de problemas. Devido à sua importância e complexidade, a implementação desta plataforma foi dividida em várias fases.

#### **3.1 Levantamento e Especificação de Requisitos**

A primeira fase do projecto pretendia identificar as funcionalidades que a aplicação deveria possuir. O desconhecimento das potencialidades do USPSD requereu uma leitura exaustiva dos seus manuais. A ferramenta também foi instalada numa máquina de testes. Esta instalação permitiu efectuar testes práticos sobre as configurações possíveis e verificar o comportamento da aplicação. Concluída a análise das funcionalidades da ferramenta, foi efectuado um levantamento de requisitos. A importância desta tarefa era elevada, uma vez que um levantamento deficiente iria prejudicar toda a implementação da plataforma.

O documento elaborado apresentava as principais funcionalidades do USPSD e pretendia obter diversas informações. Os principais dados requeridos eram os utilizadores, os grupos e os perfis de utilização; os tipos de registos, os campos que os constituem, os campos obrigatórios e a lista de valores para os campos que têm obrigatoriamente de assumir um valor de um determinado conjunto; as tipificações, os serviços, os equipamentos e os níveis de serviço. Pretendiam-se obter ainda outros dados técnicos como o servidor de e-mail e a conta a utilizar, os itens que deveriam formar o explorador de pastas e as opções da barra de menus.

Obtidas as informações pedidas, foi elaborado um documento a especificar os requisitos. Este documento pretendia comparar as funcionalidades pretendidas com as implementadas. A especificação de requisitos continha uma lista com todos os dados recolhidos no levantamento e caracterizava as opções inicialmente tomadas. O Service Desk tinha de utilizar a metodologia ITIL (Information Technology Library Infrastructure). A utilização das boas práticas definidas pelo governo britânico para a gestão e prestação de serviços IT fazia todo o

sentido, uma vez que a Mainroad se enquadra nessa área. O uso do inglês como idioma também foi definido antes da implementação.

### **3.2 Implementação e Configuração**

A ferramenta de Service Desk oferece, na sua instalação por defeito, um grande conjunto de funcionalidades. Contudo, a grande vantagem desta ferramenta é a sua configuração. Esta aplicação está preparada para ser modificada tanto ao nível das funcionalidades como ao nível da interface. A concepção inteligente do USPSD permite aos administradores efectuarem alterações sem modificar a configuração inicial da ferramenta.

Para assegurar uma implementação segura e funcional, o servidor de testes usado na análise da ferramenta foi reutilizado. As configurações da ferramenta na máquina de testes foram apagadas, com o intuito de testar a estabilidade das configurações finais, pretendidas para o USPSD. Após garantir que as operações efectuadas correspondiam ao que se pretendia, as mesmas eram efectuadas no servidor principal. A utilização de dois servidores com o USPSD instalado foi crucial para garantir a eficiência do Service Desk. Concluídas as instalações das ferramentas nos respectivos servidores, procederam-se as configurações.

#### **3.2.1 Base de Dados**

As primeiras configurações efectuadas foram na base de dados. O USPSD permite modificar as tabelas utilizadas, alterando e acrescentando campos a cada uma. A configuração da base de dados teve de ser efectuada no início, uma vez que cada alteração realizada a uma tabela interage com o seu conteúdo.

A ferramenta de Service Desk possui dois conceitos diferentes relacionados com base de dados, tabelas e objectos. As tabelas são utilizadas para armazenar os dados enquanto os objectos servem para os moldar e apresentar nas interfaces. Para modificar campos de uma tabela, o USPSD disponibiliza um directório específico. Nesse directório, tem de ser criado um ficheiro para cada tabela que se pretende alterar. Para modificar um campo existente, o nome do campo tem de ser referenciado, seguido da alteração pretendida. Para acrescentar novos, é necessário utilizar uma estrutura específica, evitando incompatibilidades com versões futuras do USPSD. A definição de objectos processa-se da mesma forma que as alterações nas tabelas. Existe um directório próprio onde têm de ser criados ficheiros para cada objecto. Nestes ficheiros é possível definir os campos obrigatórios, os campos que devem assumir um valor por defeito na criação de um novo registo e o respectivo valor e adicionar ao objecto novos campos, permitindo a sua utilização na interface gráfica.



Após criar os ficheiros para as tabelas e os objectos, é necessário executar algumas operações para que a base de dados assuma as novas configurações e seja reestruturada. Essas operações são específicas do Service Desk e não têm utilidade de ser referidas.

### 3.2.2 Consola Administrativa

O USPSD possui uma consola administrativa onde a maiorias das configurações relacionadas com o comportamento da aplicação podem ser efectuadas. As tarefas de configuração efectuadas nesta consola foram a integração com a Active Directory (AD) e a configuração da autenticação, do servidor de e-mail, da base de dados de conhecimento, do repositório de documentos, de questionários, de níveis de serviço, de notificações, de eventos e macros, de actividades, do explorador de pastas e das restrições.

- **Active Directory** – A integração com a AD foi um pouco complicada. A falta de informação sobre os valores a utilizar para a sua integração com o Service Desk foi a principal razão. Após identificar o servidor com a AD, foi utilizada uma conta para navegar na árvore de utilizadores. A correcta integração deste componente permitiu inserir todos os funcionários da Mainroad no USPSD.
- **Autenticação** – A ferramenta de Service Desk permite definir alguns parâmetros relacionados com a autenticação. O acesso a utilizadores anónimos não foi concedido. O processo de autenticação para utilizadores da Mainroad foi configurado para ser automático, utilizando para isso a conta do sistema operativo. No caso do utilizador não pertencer à Mainroad, um formulário de autenticação é apresentado.
- **E-mail** – A configuração do e-mail foi uma tarefa muito simples mas essencial. A configuração deste componente passou por identificar o servidor de e-mail e a conta a serem utilizados. Os protocolos utilizados para os e-mails foram o SMTP para enviar e o IMAP para receber. A configuração do e-mail permite ainda restringir a criação de registos aos utilizadores existentes na aplicação e obrigar a que o assunto do e-mail enviado esteja consistente com uma expressão regular. Estas duas opções não foram utilizadas, embora, a restrição da criação de registos a utilizadores existentes seja um caso a considerar quando à base de dados de clientes estiver completa. A importância do e-mail verifica-se uma vez que cerca de 85% dos registos abertos o usam e é o método adoptado para notificar os utilizadores.
- **Base de dados de conhecimento** – O USPSD possui uma base de dados de conhecimento. Uma base de dados de conhecimento tem como objectivo publicar artigos relacionados com diversas matérias. Esta base de dados pode ser utilizada

pelos analistas, para identificar soluções para problemas que já aconteceram, tal como pode ser acedida por utilizadores vulgares, permitindo-lhes descobrir a solução de um determinado problema, resolvendo-o. A base de dados de conhecimento não está a ser utilizada de momento.

- **Repositório de documentos** – Um repositório de documentos permite anexar documentos aos registos, armazenando os ficheiros num determinado directório. Foi configurado um único repositório de documentos, situado no servidor principal. Todos os e-mails com ficheiros em anexo também são enviados para o repositório, garantindo que nenhuma informação se perde.
- **Questionários** – O Service Desk permite enviar questionários aos utilizadores da ferramenta. Neste módulo é possível criar perguntas, determinar as respostas possíveis para cada uma e definir que perguntas podem ter mais de uma resposta. Embora nenhum questionário tenha sido configurado, o seu uso está normalmente associado à recolha de dados estatísticos sobre a satisfação dos clientes na resolução de registos. As respostas aos questionários são armazenadas e é possível gerar relatórios e gráficos sobre os resultados obtidos.
- **Níveis de serviço** – Os níveis de serviço correspondem ao tempo de resposta acordado entre o cliente e a Mainroad. O tempo de resposta é o período de tempo que passa desde que um registo é aberto até que um técnico inicia a sua resolução. Foram criados dez níveis de serviço, de forma a satisfazerem todos os serviços acordados.
- **Notificações** – As notificações correspondem ao envio de mensagens, por um determinado meio, para uma ou mais pessoas. Embora o USPSD possua uma aplicação própria para notificar, esta não foi utilizada, sendo o e-mail a única forma seleccionada para realizar essa tarefa. Foram definidos alguns processos para enviar um e-mail aos clientes sempre que um registo é aberto e fechado. Os clientes devem ainda receber uma mensagem sempre que for solicitada mais informação sobre um determinado problema. Os membros da equipa de suporte a que o registo está associado também são notificados sempre que um registo for aberto ou fechado e quando um registo é transferido para a área a que pertencem. As restantes notificações estão associadas a eventos e podem ser observadas mais à frente.
- **Eventos e Macros** – Um evento é um processo que se inicia automaticamente consoante um ou mais critérios. Uma macro é uma acção executada por um evento. Foram criadas macros para agregar campos de utilizadores. Esses campos

correspondem às pessoas que devem ser notificadas quando um evento acontecer. Os eventos criados, por sua vez, pretendem notificar aos 50%, aos 75% e aos 100% do tempo de reposta, determinados utilizadores, definidos nas macros. Enquanto o registo não for associado a nenhum técnico, o tempo decorrido continua a ser incrementado e os eventos permanecem activos.

- **Actividades** – O Service Desk possui seis tipos de actividades. As actividades pretendem efectuar uma alteração num registo e inserir um comentário sobre a mesma. As acções existentes permitem modificar o estado do registo, registar um telefonema, inserir um simples comentário, escrever a solução do problema, transferir o registo, escalar o mesmo e notificar manualmente. Para modificar o estado de um registo, é obrigatório recorrer à actividade correspondente. Esta exigência ocorre uma vez que o comentário inserido será a informação enviada ao cliente no fecho do registo.
- **Explorador de pastas** – O explorador de pastas é um dos principais componentes da ferramenta. Cada item do explorador de pastas corresponde a uma consulta, que quando seleccionada, devolve os registos que passaram nos critérios. Existem dois tipos diferentes de explorador de pastas. Um tipo é utilizado pelos técnicos normais e apenas contém consultas para os registos da sua área. O outro tipo destina-se a utilizadores privilegiados, que podem visualizar todos os registos. O explorador do segundo tipo é constituído por várias pastas, correspondendo cada uma a uma área. Esta organização permite uma melhor visualização dos registos. Os exploradores de pastas foram configurados para actualizarem automaticamente de dois em dois minutos.
- **Restrições** – O USPSD possui um componente que permite restringir o acesso a alguns dados e a algumas operações. As restrições podem ser associadas a perfis de utilização ou a utilizadores. A restrição principal limita o acesso dos técnicos aos registos associados à equipa que pertencem. Os técnicos de um determinado grupo apenas pode ver e editar registos associados à sua área.

### 3.2.3 Perfis de Utilização

Os perfis de utilização correspondem aos tipos de acesso dos utilizadores. Foram configurados seis tipos de acesso. Dois dos tipos destinam-se aos utilizadores que não resolvem registos, nomeadamente os funcionários e os clientes. Outros dois perfis correspondem aos dois níveis de técnicos que resolvem problemas. Os restantes perfis de utilização separam os dois tipos de administradores existentes.

- **Customer** – Os clientes apenas podem criar e ver os seus próprios registos. Após criar um registo, um cliente não pode alterar o seu estado, mas apenas ver e acrescentar informação.
- **Employee** – Os funcionários têm permissões idênticas aos clientes. A diferença entre estes perfis reside na autenticação, que é efectuada através do sistema operativo para os funcionários. Desta forma, a inserção do nome de utilizador e a palavra-chave não necessitam de ser digitadas.
- **AnalystLevel2** – Os analistas de segunda linha, para além das permissões dos funcionários, podem ver e modificar todos os registos associados ao grupo a que pertencem. Estes analistas são responsáveis por resolver os problemas de uma determinada área.
- **AnalystLevel1** – Os analistas de primeira linha possuem todas as permissões dos de segunda. Este tipo de analistas podem ver e modificar todos os registos, independentemente do grupo associado. Este tipo de utilizadores tem com função redireccionar os novos registos para a área correspondente. Os analistas podem ainda criar utilizadores e localizações.
- **AdministratorLevel2** – Os administradores de segunda linha, para além das permissões do perfil anterior, podem criar e modificar organizações, serviços e equipamentos. Este tipo de administrador desempenha funções ligadas ao ramo administrativo.
- **AdministratorLevel1** – Os administradores de primeira linha correspondem aos administradores da ferramenta. Este tipo de utilizador pode efectuar todas as operações e é o único com acesso à consola administrativa. Os administradores são responsáveis por assegurar a estabilidade do Service Desk.

### 3.2.4 Importações

A importação de dados foi uma das fases do processo de configuração do USPSD. Foram importados vários dados como utilizadores, grupos, membros dos grupos, organizações, localizações, tipificações e serviços. O Service Desk possui um módulo que permite a extracção e importação de dados, para um formato predefinido. Foram desenvolvidas aplicações para manipular dados, de forma a ficarem com a estrutura pretendida para a importação.

- **Utilizadores** – Os utilizadores da Mainroad foram importados utilizando a integração com a Active Directory. Também foram importados alguns clientes, identificados inicialmente. Esta importação foi efectuada recorrendo ao módulo do USPSD. Os dados importados, tanto para os utilizadores da Mainroad como para os clientes foram o nome, o nome de utilizador, o perfil de utilização, o e-mail, o número de telefone, o número de telemóvel e a empresa.
- **Grupos** – Cada grupo da aplicação corresponde a uma equipa de suporte. Como os grupos já tinham sido criados no servidor de testes, estes foram extraídos para um ficheiro e importados para o novo servidor.
- **Membros dos grupos** – Os membros dos grupos, tal como os grupos, foram extraídos do Service Desk de testes e importados no de produção. Nos grupos ainda foi definido, manualmente, o responsável de cada área.
- **Organizações** – As organizações correspondem às empresas às quais a Mainroad presta serviços. As organizações foram importadas de um ficheiro e podem ser associadas a utilizadores, serviços e equipamentos.
- **Localizações** – Tal como as organizações, as localizações também foram importadas de um ficheiro. As localizações são utilizadas para identificar um local e podem ser atribuídas a utilizadores, registos, organizações e equipamentos.
- **Tipificações** – As tipificações foram importadas de um ficheiro. As tipificações têm como objectivo, caracterizar, de uma forma geral, os registos. As tipificações importadas estão agrupadas em níveis, que podem ser aprofundados, para melhor descrever o registo. A tipificação determina ainda se um registo é um pedido ou um problema.
- **Serviços** – Os serviços existentes correspondem aos serviços que a Mainroad tem acordado com os clientes. Sempre que um cliente abre um registo, o serviço relativo ao tipo de pedido ou problema tem de ser associado. Este passo é necessário para garantir que o tempo de resposta acordado é cumprido, através dos eventos associados. Também serve para garantir que um cliente não está a pedir a resolução de um problema para o qual não tem contracto.

### 3.2.5 Interface

A ferramenta de Service Desk possui duas interfaces gráficas, a interface Java e a interface Web. A Mainroad decidiu que a única interface utilizada seria a interface Web, pelo que a



interface Java não necessitou de ser modificada. Para configurar a interface Web, o USPSD disponibiliza um conjunto de directórios para modificar os campos utilizados, as funções, o estilo adoptado e as imagens. Para garantir a integridade dos ficheiros originais, o Service Desk oferece uma estrutura de directórios idêntica aos originais, onde os ficheiros modificados devem ser colocados. Desta forma, a aplicação verifica se existe algum ficheiro nas pastas de alterações, prevalecendo sobre os ficheiros originais.

A maioria das alterações decorreu nos ficheiros relacionados com os campos que devem aparecer. Os campos desnecessários tinham de ser removidos, tal como os campos criados tinham de ser acrescentados. Para cada campo deve ser definido o seu tipo e o nome que deve aparecer na interface. A posição que cada campo tem de ocupar no formulário também tem de ser referida. Os formulários modificados foram o dos contactos, dos registos, das organizações, dos serviços e dos equipamentos.

As imagens da aplicação também foram modificadas. A interface foi programada para apresentar o logótipo da Mainroad em todos os formulários. As funções alteradas consistiram na remoção de alguns itens da barra de menus. Foram também alteradas algumas funcionalidades relativas a botões.

### **3.2.6 Integração com o Portal**

A ferramenta de Service Desk foi integrada com o portal. Esta integração teve de ser efectuada do lado do Portal e não teve impacto no USPSD. Verifique-se que o portal é o local onde todas as aplicações podem ser acedidas, tal como os relatórios gerados e publicados.

### **3.3 Formações**

Após o Service Desk ter as suas funcionalidades implementadas, foi desenvolvido o manual do utilizador. Este manual foi distribuído pelas equipas de suporte e possuía a distinção entre as funcionalidades apenas destinadas a um tipo de utilizadores. Estas referências eram necessárias uma vez que alguns perfis tinham mais privilégios que outros.

Foram agendadas várias formações para os analistas, duas para os de primeira linha e seis para os de segunda. Para evitar uma deslocação a Lisboa, foi realizada uma formação por vídeo-conferência, para os técnicos da capital. As formações eram constituídas por uma demonstração prática de trinta minutos. A seguir à demonstração, havia lugar ao esclarecimento de dúvidas.



## 4 Plataforma de Inventário

O inventário foi o segundo componente da plataforma de gestão implementado. A recolha de inventário tem como objectivo disponibilizar informações sobre as características dos vários componentes da infra-estrutura IT.

A reestruturação da plataforma de inventário trouxe duas novas ferramentas, manteve uma e eliminou outra. As novas ferramentas destinam à recolha de inventário nos servidores e a que se manteve corresponde a todos os equipamentos de rede das infra-estruturas. As ferramentas adquiridas foram o Unicenter Asset Management (UAM) e o Microsoft System Management Server (SMS). Estas duas aplicações têm objectivos idênticos, embora uma seja utilizada para a recolha de servidores com plataforma Windows e a outra para as restantes plataformas. A ferramenta para a recolha de inventário sobre os equipamentos de rede não foi modificada.

### 4.1 Funcionalidades

O UAM e o SMS são ferramentas concorrentes, com funcionalidades idênticas e objectivos comuns. Os interesses principais destas aplicações são o controlo de utilizadores, o controlo remoto, a distribuição de software, a monitorização de licenciamento e a geração de relatórios. Embora estas ferramentas apenas sejam utilizadas para recolher o inventário e gerar relatórios, a utilização das restantes funcionalidades está prevista para um futuro próximo.

- **Controlo de utilizadores** – O controlo de utilizadores é uma das funcionalidades das aplicações implementadas. Sempre que um utilizador autentica-se num dos servidores, é registado o seu nome. O número de vezes que o utilizador entrou na máquina é também é guardado. Este componente permite obter informações úteis sobre quem utiliza cada computador e assegurar que o uso indevido não acontece.
- **Controlo remoto** – O controlo remoto permite abrir uma sessão num servidor monitorizado. Os servidores com plataforma Windows permitem uma visualização gráfica enquanto que os restantes disponibilizam uma linha de comandos. É ainda possível definir os servidores que podem ou não ser acedidos.
- **Distribuição de Software** – A distribuição de software é um componente importante das ferramentas de inventário. É possível determinar, através de consultas, os servidores que possuem ou não determinado software, podendo o instalar remotamente. A distribuição de software possibilita ter servidores idênticos, com as mesmas aplicações instaladas e a mesma configuração.

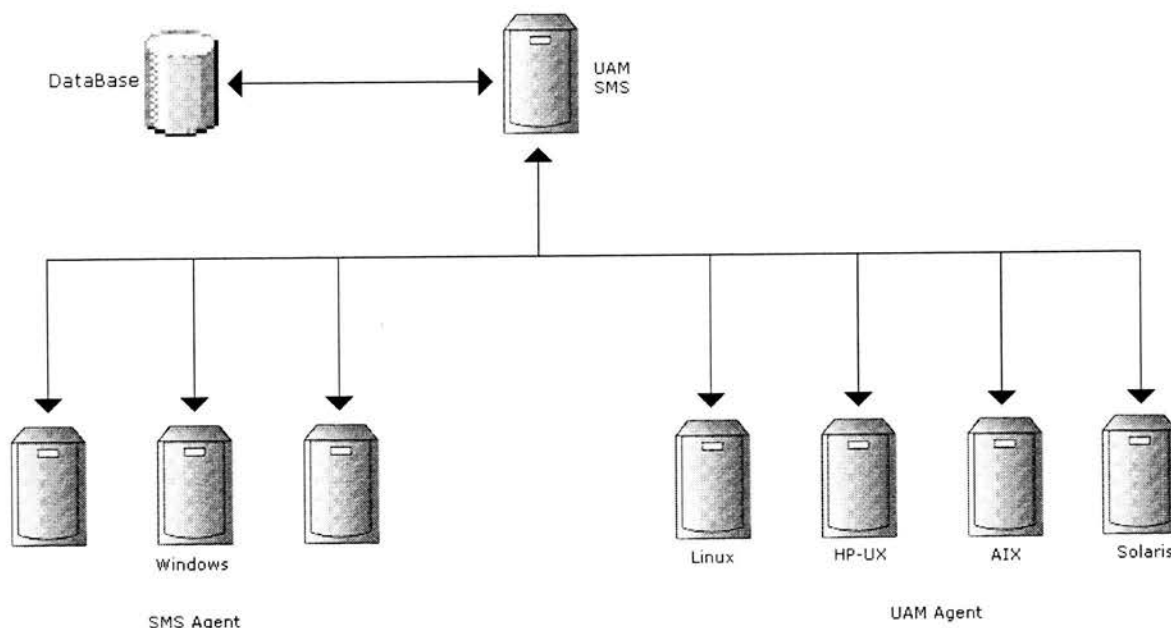
- **Monitorização de licenciamento** – A controlo de licenciamento permite verificar o uso de aplicações e determinar se o número de licenças adquiridas está adequado ao necessário. Uma ferramenta pode ter um número excessivo de licenças enquanto outras ser insuficiente. É possível gerar mensagens sempre que uma licença é ultrapassada e restringir o seu uso para esses casos.
- **Recolha de inventário** – A recolha de inventário é o único componente implementado. O objectivo da Mainroad era possuir uma base de dados com todas as informações sobre os servidores. A recolha de inventário será abordada mais à frente.
- **Geração de relatórios** – As ferramentas utilizadas possuem um módulo que permite gerar relatórios e aceder a todas as informações sobre as operações efectuadas. Foram gerados relatórios diversos relatórios sobre as diferentes categorias de inventário recolhidas.

## 4.2 Arquitectura

O UAM e o SMS possuem arquitecturas muito parecidas, embora as convenções utilizadas para os componentes sejam diferentes. O método de funcionamento baseia-se na instalação de uma agente, encarregue de obter várias informações e reporta-las para processamento. Os principais componentes do UAM são o motor, o sector, os agentes e a base de dados. O SMS, por sua vez é constituído por um site primário, um site secundário, agentes e uma base de dados.

- **Motor/Site primário** – O motor/site primário é o componente responsável por recolher e processar as informações enviadas pelos agentes aos sectores/site secundários. Estes componentes são os principais de cada aplicação e executam todas as operações de processamento.
- **Sector/Site secundário** – Estes componentes são os locais de armazenamento temporário dos dados recolhidos, até serem enviadas para a base de dados e serem processados. Embora seja possível possui mais do que um sector/site secundário, a arquitectura apenas possui um.
- **Agentes** – Os agentes são os processos responsáveis por recolher as informações dos servidores dos clientes. Cada máquina monitorizada possui um agente instalado e informações sobre o sector/site secundário para que deve reportar.
- **Base de dados** – A base de dados é o local onde todas as informações são armazenadas. A base de dados possui informação sobre todo o inventário dos

servidores monitorizados. A base de dados guarda o último valor diferente do actual e a data em que a modificação ocorreu.



**Fig. 4 – Arquitectura da plataforma de inventário**

### 4.3 Implementação e Configuração

A implementação das ferramentas de recolha de inventário não foi complicada. A instalação decorreu sem problemas e todos os componentes foram instalados com sucesso. A instalação dos agentes nos servidores a inventariar foi acompanhada com as equipas responsáveis por cada plataforma. Instalados os agentes, foram criados grupos para organizar as máquinas por clientes. Foram também gerados relatórios, também agrupados por cliente e separados pelas diversas categorias do inventário.

#### 4.3.1 Recolha de Inventário

A recolha de inventário é uma das funcionalidades que as ferramentas utilizadas disponibilizam. A recolha de inventário separa as informações adquiridas por categorias e disponibiliza informações sobre as configurações da rede, do sistema operativo, discos, dos dispositivos, dos sistemas e do software instalado.

- **Rede** – As informações sobre as configurações da rede obtidas dividem-se em duas partes, DNS e TCP/IP. O primeiro contém informações sobre as configurações do nome do servidor e do servidor de DNS utilizado. O segundo, por sua vez, disponibiliza informações sobre as placas de rede como os nomes das interfaces de rede, os endereços IP, os endereços MAC, a máscara e o domínio.

- **Sistema operativo** – Os dados recolhidos sobre o sistema operativo são a plataforma, a versão, o modelo, o fabricante. Existem ainda informações sobre as configurações aplicadas a cada servidor.
- **Disco** – As ferramentas utilizadas oferecem informações sobre os discos e as partições criadas para cada um. O tamanho total, o espaço livre e as unidades criadas são os principais dados recolhidos.
- **Dispositivo** – Os dispositivos encontrados correspondem a controladoras, placas de rede, placas PCI, portas USB e cartas de vídeo. São monitorizadas diversas informações relativamente ao modelo e fabricante de cada dispositivo encontrado.
- **Sistema** – As informações do sistema correspondem à versão do agente e aos dados monitorizados. A versão do agente e a data de execução pode ser visualizada no sistema.
- **Software** – As informações de software permitem recolher todas as aplicações instaladas nos servidores instalados. O método de recolha de software pode ser seleccionado.

#### 4.3.2 Relatórios

Os relatórios gerados foram agrupados por cliente, por sistema operativo e por categoria. Desta forma, os relatórios gerados podem ser acedidos consoante diversos critérios. Para obter relatórios actualizados, foram agendadas gerações automáticas. Foram definidos processos para correrem todas as semanas, gerando os relatórios e publicando-os.

**IP address overview - Auto History - 06/26/2002 11:38:03 AM**

Generated 06/26/2002 at 11:38:03 AM by ADMIN.  
Result contains 6 rows.

Area	Name	Network Adapter [1st]   Name	Network Adapter [1st]   Status	Network Adapter [1st]   IP Address	Network Adapter [1st]   IP Address (priv)
Area Name	Computer Name 1	[1] Novell NE2000 Adapter	OK	172.24.255.255	172.24.255.255
Area Name	Computer Name 2	3Com EtherLink III ISA (3C509/3C509b) in Legacy mode	OK	172.24.255.255	172.24.255.255
Area Name	Computer Name 3	[1] Novell NE2000 Adapter	OK	172.24.255.255	172.24.255.255
Area Name	Computer Name 4				
Area Name	Computer Name 5	3Com EtherLink III ISA (3C509/3C509b) in Legacy mode	OK	172.24.255.255	172.24.255.255
Area Name	Computer Name 6	3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)	OK	172.24.255.255	172.24.255.255

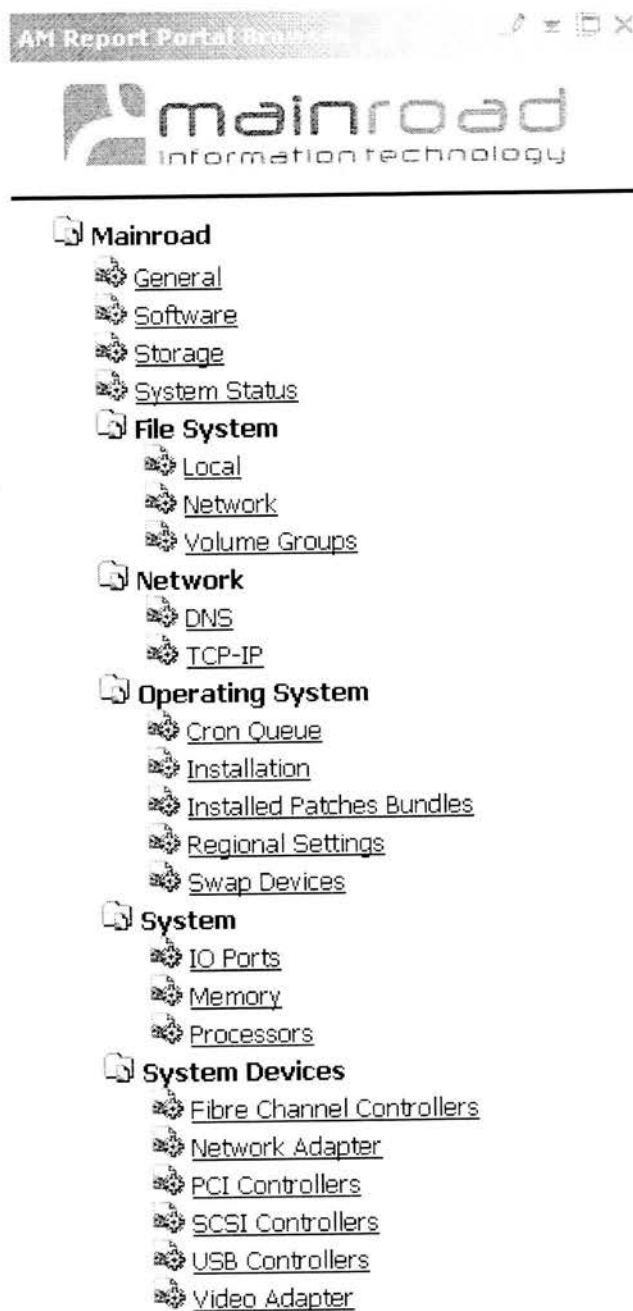
**Fig. 5 – Exemplo de um relatório**

#### 4.4 Integrações

As ferramentas de recolha de inventário foram integradas com o Portal e com o Service Desk. A *Framework* Unicenter já possui um módulo que permite a integração com o Portal. Contudo, a mesma não era possível para o SMS e tiveram de ser desenvolvidas aplicações para o poder integrar. O Service Desk ainda não possuía os módulos necessários para integrar com o UAM. As novas versões utilizadas ainda não tinham um módulo desenvolvido para efectuar a integração. Como a integração do SMS teria de ser desenvolvida, foram concebidas as aplicações para possibilitar as duas integrações.

##### 4.4.1 Portal

Os benefícios da integração do Portal com a plataforma de inventário foram vários. Um explorador de pastas era automaticamente criado, permitindo a navegação dos diversos relatórios. Como o SMS não possuía este componente, foi desenvolvida uma aplicação para gerar a mesma estrutura que o UAM, ou seja, o respectivo explorador de pastas.



**Fig. 6 – Explorador de pastas**

A integração beneficiava ainda de determinadas visualizações específicas como a construção de gráficos sobre as mais diversas categorias inventariadas. As visualizações criadas foram sobre os sistemas operativos e sobre os modelos de processadores utilizados. Os gráficos podem ser observados sobre diversas formas, entre as mais utilizadas, os gráfico de barras e a as tartes.

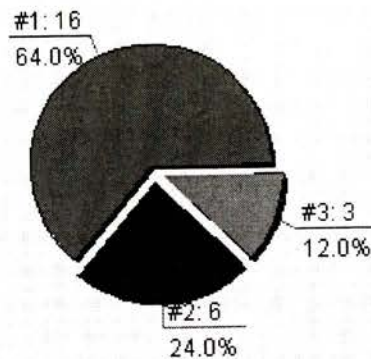


## Operating System Demographics

Chart


 Show scrollbars  Options

- 1: HP-UX
- 2: Linux
- 3: AIX

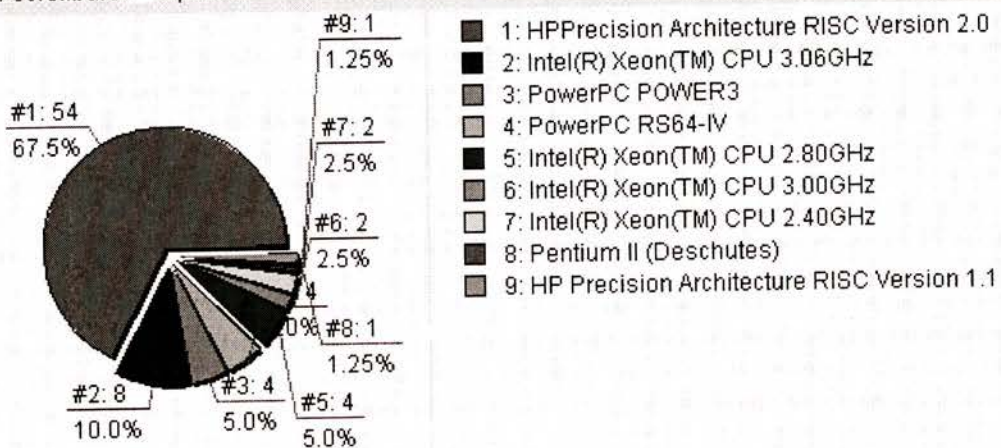


■ Cnt

Fig. 7 – Visualização das plataformas

## Processor Demographics

Chart


 Show scrollbars  Options


■ Cnt

Fig. 8 – Visualização dos processadores

## 4.4.2 Service Desk

Como foi referido, as versões utilizadas para a integração do UAM com o USPSD ainda não tinham sido desenvolvidas e a integração com o SMS tinha de ser implementada. Desta forma, foram criadas duas aplicações para integrar com cada uma das ferramentas. O objectivo destas integrações era actualizar o componente de inventário, existente no Service

Desk. A integração da ferramenta relativa aos equipamentos de rede também foi desenvolvida.

Antes de desenvolver as aplicações que iriam integrar a plataforma de Service Desk com a de inventário, outra operação foi realizada. Os dados administrativos dos diferentes tipos de equipamentos também tinham de ser importados. Foi desenvolvida uma aplicação para importar todos os equipamentos e os respectivos dados. As informações importadas correspondiam ao cliente, ao nome do contrato, às datas de início e fim de contrato, ao modelo e à marca do equipamento, ao local e a algumas observações existentes.

Terminada esta importação, ainda foram importados os servidores da ferramenta de inventário que iria ser descontinuada. Este processo dividiu-se em duas fases, nomeadamente à importação de servidores repetidos e à de novos. Esta divisão foi necessária uma vez que os servidores importados da base de dados administrativa, não deveriam ser novamente criados, mas sim actualizados. Contudo, as informações repetidas e diferentes encontradas para a nova importação, deveria prevalecer sobre os dados importados previamente. Para os novos servidores, a validação não era necessária, pelo evidente motivo de os servidores não existirem.

Concluídos os processos de importações dos dados administrativos e das ferramentas da plataforma que deixaram de ser utilizadas, procedeu-se ao desenvolvimento das integrações das novas plataformas. A integração das ferramentas de recolha de inventário com o Service Desk pretendia actualizar o inventário e automatizar o processo.

As aplicações ligam-se a cada uma das bases de dados e recolhem os dados que se desejam actualizar. Esses dados são a marca, o modelo, o número de série, o endereço IP das placas de rede e o respectivo endereço MAC, o sistema operativo, o número de processadores e a respectiva velocidade, a memória, o número de discos e o seu respectivo espaço. Para saber a evolução do inventário, é gerado um ficheiro com todas as alterações efectuadas e a respectiva data em que ocorreram. Antes de serem efectuadas quaisquer actualizações, é efectuada uma cópia de segurança da tabela dos equipamentos. Esta cópia assegura a integridade da tabela e permite restabelecer os dados em caso e algum erro ocorrer.

## 5 Plataforma de Monitorização

A plataforma de monitorização é o componente da plataforma de gestão responsável por monitorizar todos os componentes das infra-estruturas dos clientes da Mainroad. Esta plataforma é constituída por diversas ferramentas, com o objectivo de avaliar a disponibilidade e a performance de redes, sistemas, serviços e aplicações. A integração das ferramentas também tinha de ser assegurada, permitindo extrair o maior desempenho possível.

A extensiva duração do estágio não permitiu concluir a implementação desta plataforma na totalidade. A documentação apresentada sobre esta plataforma corresponde ao trabalho realizado até ao fim do mês de Setembro.

### 5.1 Network and System Management

A ferramenta responsável pela gestão de redes e sistemas é a principal da plataforma de monitorização. As restantes aplicações da plataforma de monitorização necessitam desta ferramenta para funcionarem e para serem integradas. A gestão de redes e sistemas pretende detectar falhas e os motivos da sua ocorrência bem como prevenir avarias.

A ferramenta utilizada para gerir as redes e os sistemas foi o Unicenter Network and System Management (UNSM). Esta aplicação é constituída por diversos módulos, devidamente integrados e com funcionalidades próprias. A complexidade desta ferramenta obriga a dividir a sua implementação em fases.

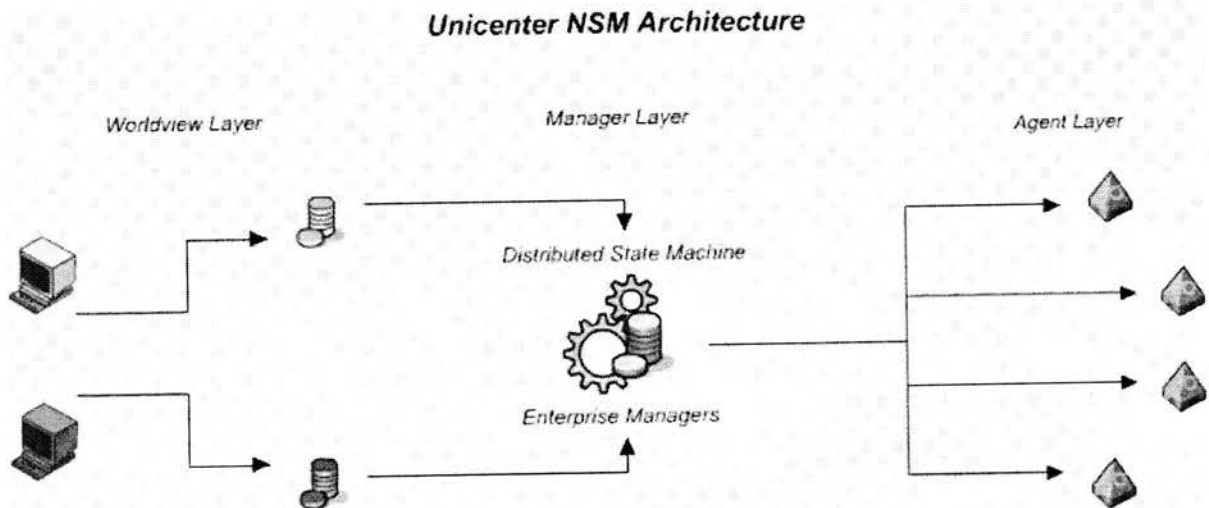
#### 5.1.1 Arquitectura

A arquitectura que era pretendida inicialmente para o UNSM não foi possível. Desejava-se utilizar a base de dados remota, o que não foi possível por diversas razões. A arquitectura foi reorganizada com base de dados local e é constituída por três camadas, WorldView, Manager e Agent.

- **WorldView** – A primeira camada do UNSM é responsável por apresentar os objectos que se estão a monitorizar. Esta camada é composta por três componentes, o COR (Common Object Repository), a Real World Interface e o WorldView Application. O COR é o coração desta camada e contém as classes e os objectos monitorizados. O COR corresponde à base de dados que armazena as informações das classes e respectivos objectos. A Real World Interface tem como objectivo comunicar com o COR e manipular os dados obtidos para os apresentar aos utilizadores. O WorldView

Application é o componente utilizado para pela maioria dos restantes componentes da aplicação para comunicar com o COR.

- **Manager** – A segunda camada serve de responsável por comunicar com o COR e com os agentes. Esta camada é constituída por dois componentes, o DSM (Distributed State Machine) e pelo Enterprise Manager (EM). O DSM é responsável por gerir os estados de todas as classes e objectos do UNSM. Este componente é o motor do processamento do status dos agentes, que podem ser Normal, Down, Warning ou Critical. O EM, por sua vez, é a função responsável por executar operações, automaticamente, consoante o tipo de estado dos agentes. Este componente permite definir que acções devem ser tomadas quando uma política é violada.
- **Agent** – A última camada da arquitectura do UNSM é o agente. Os agentes devem ser instalados nos servidores que se pretendem monitorizar e obtêm informações sobre o estado da máquina, reportando-o ao DSM. Com o agente instalado é possível obter informações sobre a carga do CPU, da memória utilizada, das placas de rede entre outros.



**Fig. 9 – Arquitectura do NSM**

### 5.1.2 Implementação e Configuração

A implementação do UNSM foi bastante problemática. Os motivos dos problemas encontrados estavam relacionados com a utilização da base de remota. Com base de dados local, a instalação não teve nenhum problema. A ferramenta de monitorização possui diversos componentes que permitem administrar correctamente a gestão das redes e servidores.

- **2D Map** – O 2D Map é a ferramenta utilizada para visualizar todas as infra-estruturas. O ambiente gráfico de duas dimensões apresenta as redes e os servidores monitorizados. Esta aplicação é utilizada para observar, de uma forma geral, os equipamentos da infra-estrutura com problemas. Também permite visualizar as Business Process View (BPV) e os domínios. O 2D Map permite ainda identificar os agentes instalados e possibilita o acesso a muitas ferramentas do UNSM.
- **3D Map** – Esta ferramenta é semelhante ao 2D Map. A única diferença é que o seu ambiente gráfico é a três dimensões, em vez de duas. Os recursos exigidos para funcionar levaram à não utilização desta ferramenta.
- **Business Process View** – Este componente permite agrupar equipamentos com propriedades idênticas. No ambiente da Mainroad, foram criados uma BPV para cada cliente, contendo cada uma, os equipamentos que lhes pertencem.
- **Domínios** – Os domínios são agregações dinâmicas de equipamentos, consoante alguns critérios. Foram criados dois domínios, um para os objectos com estado a Warning e outro com estado a Critical. Estas visualizações permitem agrupar todos os servidores com problemas.
- **NodeView** – O NodeView permite visualizar os diferentes componentes monitorizados. Através desta aplicação é possível identificar facilmente que objectos dos equipamentos estão com problemas. O NodeView é utilizado para identificar os problemas dos equipamentos de forma a poder encontrar a sua resolução.
- **Agent View** – Esta aplicação permite visualizar os estados dos diferentes objectos monitorizados, com um ambiente gráfico. O Agent View permite também configurar as políticas que determinarão os estados dos agentes. As políticas foram definidas consoante os critérios que a Mainroad utiliza para monitorização.
- **Poll Set** – Este componente permite determinar os portos que serão utilizados pela aplicação e pelos agentes para comunicar. Aqui foram definidos alguns parâmetros como a community string utilizada, o intervalo de tempo entre tentativas de comunicação e o número de tentativas até ocorrer um erro.
- **MIB Browser** – Esta aplicação permite efectuar testes de comunicação com os agentes. Tal como o nome indica, a Management Information Base (MIB) é utilizada para testar a conectividade entre o DSM e os agentes. Sempre que um agente instalado



não é reconhecido, esta ferramenta é utilizada para testar a comunicação entre as camadas do UNSM.

- **Enterprise Manager** – Esta aplicação permite configurar um largo conjunto de parâmetros. Aqui é possível criar barra de menus para integrar as restantes aplicações de monitorização e é possível definir as mensagens que devem ser geradas e as acções a ser tomadas quando um determinado evento ocorre. Esta consola é uma das principais da ferramenta. Quando um erro ocorre, um trap é enviado para esta consola. A mensagem recebida pode despoletar uma ou mais acções automaticamente. Os técnicos responsáveis pelos servidores possuem uma consola que permite verificar os problemas que acontecem e iniciar a sua resolução.
- **Discovery** – Esta aplicação permite descobrir equipamentos consoante alguns parâmetros. É possível pesquisar um equipamento ou uma rede, utilizando diversos protocolos de comunicação. Todas as infra-estruturas monitorizadas foram encontradas utilizando esta ferramenta.
- **Explorer** – O Explorer é uma ferramenta muito eficiente e permite realizar várias tarefas dos componentes anteriores. Esta aplicação utiliza uma interface Web e permite configurar toda a infra-estrutura monitorizada.

## 5.2 Database Management

A avaliação da performance das bases de dados era um dos objectivos da plataforma de monitorização. A ferramenta utilizada chama-se Unicenter DataBase Performance Management (UDPM) e é compatível com as marcas mais conhecidas. Contudo, como os clientes da Mainroad apenas utilizam bases de dados Microsoft SQL e Oracle, o UDPM apenas foi utilizado para elas.

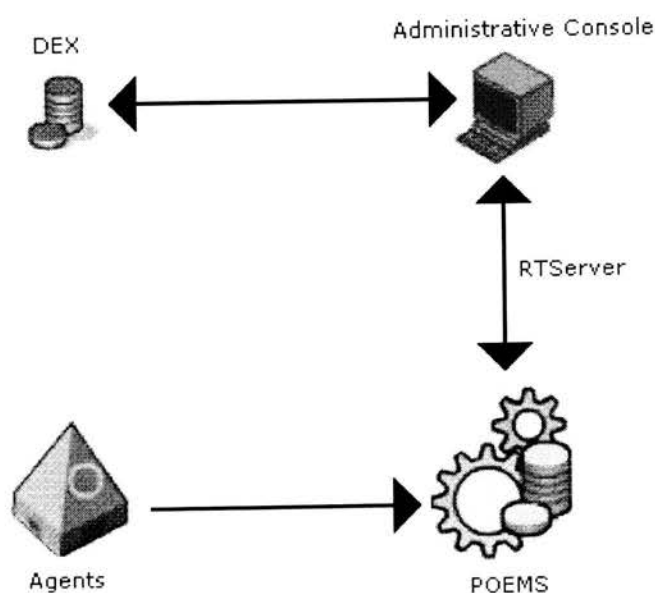
O UDPM recolhe informações sobre todos os processos relacionados com bases de dados. As principais informações monitorizadas são a carga de processamento, a memória utilizada, os bytes escritos e lidos em cada base de dados, os erros que ocorreram e os utilizadores que usam as bases dados. A partir dos dados obtidos é possível criar formulários, apresentando separadamente os gráficos pretendidos.

### 5.2.1 Arquitectura

A arquitectura do UDPM é constituída por quatro componentes principais, o DEX (Data Exchange Node), o RTServer, a consola administrativa, a interface Web, o POEMS e os agentes.



- **DEX** – O DEX é o repositório dos agentes e está associado a uma base de dados. Quando um agente é instalado, o DEX regista a sua existência com um registo na base de dados e armazena os dados que lhe são transmitidos.
- **RTServer** – O RTServer é o serviço responsável pelo processamento dos agentes. Numa configuração do UDPM apenas deve existir uma vez este serviço. Este serviço assegura o tratamento dos dados de forma a estes poderem ser apresentados.
- **Consola administrativa** – A consola administrativa permite aos utilizadores visualizar a informação manipulada pelo RTServer. Esta consola possibilita ainda configurar os agentes e determinar as informações que devem ser monitorizadas.
- **Interface Web** – A interface Web, tal como a consola administrativa, possibilita aos utilizadores visualizar a informação monitorizada pelos agentes. Esta interface possui ainda um módulo para gerar relatórios e para os integrar com o Portal.
- **POEMS** – O POEMS é o serviço responsável por comunicar com o DEX. Após os agentes obterem informação sobre a performance das bases de dados, é através deste serviço que o repositório é actualizado.
- **Agente** – O agente tem como objectivo recolher informação sobre a performance da base de dados e sobre o seu desempenho. O UDPM possui agentes para os mais variados tipos de base de dados como Microsoft SQL, Oracle, Informix, Sysbase entre outros.



**Fig. 10 – Arquitectura do UDPM**

### 5.2.2 Implementação e Configuração

A aplicação responsável pela monitorização das bases de dados foi instalada sem problemas. Foram instalados dois agentes de testes, um para base de dados SQL e outro para Oracle, permitindo observar os dados são recolhidos.

As configurações desta ferramenta ainda não foram definidas. Até ao momento, está a decorrer um processo de análise da informação recolhida, para identificar a que é importante e tem de continuar a ser monitorizada. Contudo, já são obtidos gráficos sobre as várias componentes monitorizadas.

### 5.2.3 Interface Web

O UDPM possui uma interface Web que permite visualizar e manipular os agentes de performance das bases de dados e de sistemas. Os agentes podem ser iniciados e parados, consoante o que se pretende efectuar.

O principal objectivo da interface Web é mostrar as métricas de performance recolhidas, sobre a forma de tabelas, gráficos ou texto. É possível criar formulários, agregando diferentes tipos de informações, consoante os desejos. Estes formulários serão criados no momento que a informação a ser monitorizada for identificada.

## 5.3 Exchange Management

A monitorização de servidores Exchange fazia parte da plataforma de monitorização. A ferramenta utilizada permite monitorizar todos os servidores Exchange e chama-se Unicenter Management for Microsoft Exchange (UMME).

As potencialidades da ferramenta e a recolha de métricas não foram possíveis. A versão do sistema operativo utilizado para os servidores com Exchange era muito recente para suportar o agente disponibilizado. Enquanto não for desenvolvido um agente compatível com a versão utilizada pelos clientes, esta implementação não será realizável.

## 5.4 SAP Management

A plataforma de monitorização pretendia avaliar a performance dos servidores SAP. A ferramenta utilizada chama-se Unicenter for SAP R/3 (USAP) e pretende recolher informações sobre o desempenho das aplicações SAP e suporta as versões mais comuns.

A arquitectura desta aplicação ainda não foi desenhada. Contudo, um agente de testes já foi instalado, para poder analisar os dados que este agente recolhe.

## 6 Conclusões

As conclusões obtidas na implementação da nova plataforma de gestão são muito optimistas. O trabalho desempenhado foi muito apreciado pelos elementos de chefia e demonstrou-se uma aposta gratificante e enriquecedora.

### 6.1 Conclusões do projecto de estágio

A integração na Mainroad foi muito simples. A existência de mais dois estagiários, a simpatia dos colaboradores e o bom ambiente ajudaram imenso na fase de adaptação. Os funcionários têm um bom conhecimento sobre as funções que desempenham e mostraram-se sempre disponíveis para ajudar a ultrapassar as dificuldades encontradas.

Os objectivos do projecto já estavam traçados no início do estágio. As tarefas a realizar e o tempo que cada uma deveria tomar já estavam delineados. Esta programação foi crucial para orientar o trabalho realizado e assegurar o progresso do seu desenvolvimento.

A implementação da nova plataforma de gestão demonstrou-se um projecto aliciante. O conhecimento da importância que esta plataforma tinha para a Mainroad e o facto de ela ser utilizada pela maioria dos seus funcionários contribuiu para motivar as pessoas envolvidas na sua concepção.

A plataforma de Service Desk era uma aposta de elevado risco. Como a Mainroad oferece um suporte contínuo aos seus clientes, a importância da estabilidade desta ferramenta era enorme. Com novas funcionalidades e utilizando a metodologia ITIL, o Service Desk podia trazer inicialmente um desajustamento nos métodos de trabalho utilizados. Através das formações realizadas, da interface atractiva e da simplicidade do funcionamento, as preocupações desvaneceram-se. Até ao momento, o Service Desk é utilizado por mais de dez equipas de suporte e revelou-se um aliado poderoso na resolução de problemas.

O inventário foi um componente que não apresentou grandes problemas. Graças à formação realizada, sobre uma das ferramentas utilizadas, a implementação desta plataforma não causou nenhum problema. As aplicações desenvolvidas para integrar com o Service Desk têm-se demonstrado fiáveis e estáveis. Os relatórios gerados permitem aos funcionários obterem facilmente toda a informação sobre determinado componente da infra-estrutura.

A monitorização é uma plataforma complexa e com vários objectivos. Até ao momento, mais de duzentos sistemas informáticos estão a ser monitorizados, garantindo o seu correcto funcionamento e prevendo falhas e avarias. A performance das bases de dados, dos servidores

SAP e dos servidores Exchange ainda não estão asseguradas, embora o seu funcionamento esteja previsto para breve.

## **6.2 Avaliação dos resultados do estágio**

Devido ao estágio ter duração de oito meses, o plano traçado para a implementação da nova plataforma de gestão ainda não foi cumprido. Contudo, todos os componentes que tinham de ser instalados já o foram, faltando apenas configurar os seus parâmetros e instalar os agentes correspondentes. Os objectivos estão a ser atingidos tanto ao nível pessoal como a nível profissional.

A nível pessoal, foi possível integrar numa equipa de trabalho com sucesso. O trabalho desenvolvido era muito interessante e muito motivador. A utilização de diversas tecnologias permitiu aprender novos conceitos e aprofundar os conhecidos. O conhecimento de novas pessoas, de diferentes empresas foi muito interessante, tal como a percepção do funcionamento do mercado das tecnologias da informação. A satisfação dos utilizadores das aplicações demonstrou-se uma experiência muito gratificante.

A nível profissional, a plataforma de gestão implementada está a cumprir os objectivos propostos e a ter muita utilidade no auxílio das equipas de suporte e na automatização de processos. A reestruturação da plataforma de gestão veio trazer inúmeros benefícios aos clientes e é uma mais valia para a Mainroad. As formações proporcionadas ajudaram imenso na utilização de ferramentas de gestão e na percepção dos mecanismos utilizados.

Ao longo do estágio, a estrutura da plataforma de gestão sofreu algumas modificações que dificultaram a sua evolução. As alterações efectuadas prejudicaram algum trabalho desenvolvido e atrasaram alguns processos. A maioria dos problemas que surgiram era relacionada com falhas das aplicações utilizadas ou requisitos desconhecidos.

## **6.3 Perspectivas de trabalho futuro**

A implementação da nova plataforma de gestão tem de ser concluída. Embora apenas faltem configurar alguns componentes, é necessário certificar que as diversas ferramentas interagem correctamente e a estabilidade da plataforma tem de ser assegurada. Os processos criados estão preparados para suportar o aparecimento de novos clientes, reduzindo o tempo dispensado em novas configurações.

A continuidade na empresa já está assegurada, e novos projectos já estão em vista.

## Referências e Bibliografia

itSMF – Gestão de Serviços IT, 2001

Office of Government Commerce (OGC) – ITIL, Service Support, TSO, 2000

Computer Associates – Unicenter ServicePlus Service Desk: Administration and Implementation

Computer Associates – Unicenter Desktop Management Foundations

Computer Associates – Unicenter Asset Management: Administration and Implementation

Microsoft – System Management Server: Concepts, Planning and Deployment

Computer Associates – Unicenter Network and System Management: Administration and Implementation

Computer Associates – Unicenter Database Performance Management: Administration and Implementation

Computer Associates – Unicenter Management for Microsoft Exchange: Administration and Implementation

Computer Associates – Unicenter Management for SAP R/3: Administration and Implementation

<http://www.itsmf.pt>, sítio com referências ao ITIL

<http://www.ital.co.uk>, sítio oficial do OGC

<http://ca.com>, sítio da Computer Associates

<http://hp.com>, sítio da Hewlett-Packard

<http://perl.com>, sítio com referências ao Perl

## ANEXO A: Screenshots das aplicações

Neste anexo encontram-se screenshots das diferentes aplicações implementadas. Por uma questão de confidencialidade, os nomes dos clientes, dos utilizadores, dos equipamentos, bem como outras informações, são omitidos.

### 1. Service Desk



Fig. 11 – Formulário de autenticação

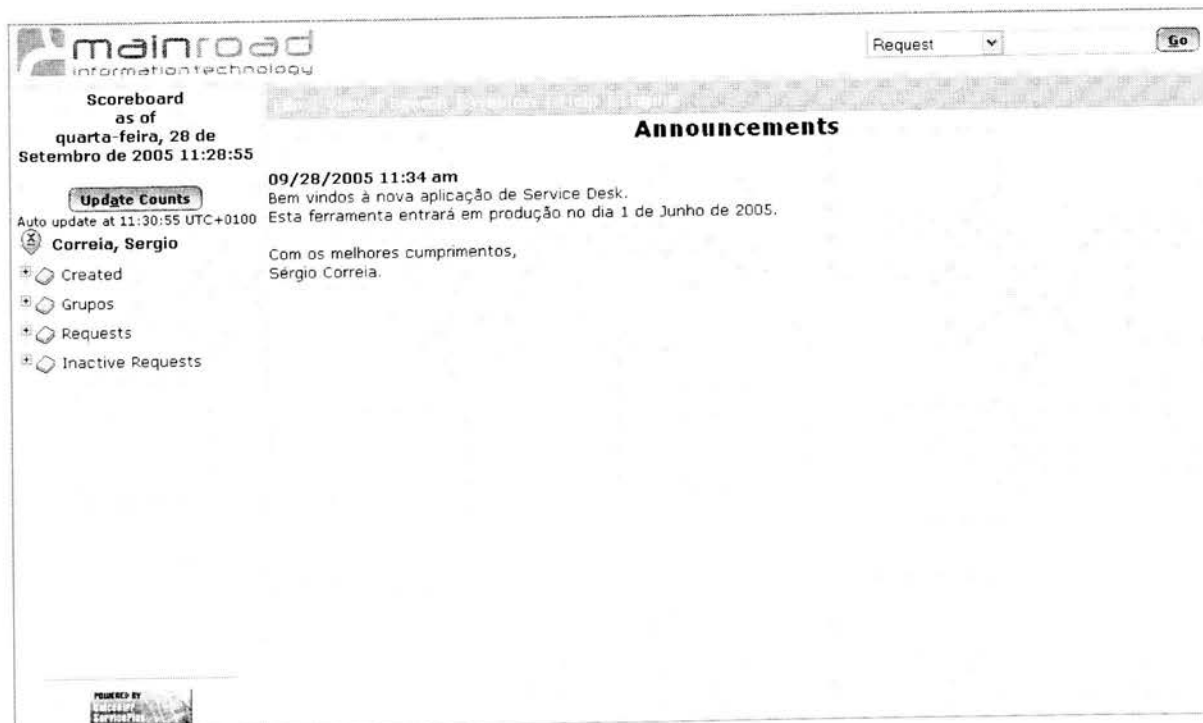



Fig. 12 – Formulário principal



**Scoreboard**  
as of  
**quarta-feira, 28 de**  
**Setembro de 2005 12:33:52**

**Update Counts**


Auto update at 12:35:52 UTC+0100

 **Correia, Sergio**

 Created

Open Requests (0)

Closed Requests (0)

 My Queue

My Group Requests (6)

Unassigned Requests (1)

My Requests (1)

Inactive Requests (38)

 Requests

1 - Emergency (0)

2 - Critical (0)

3 - Urgent (0)


4 - High (0)

5 - Medium (5)

6 - Low (1)

POWERED BY  
Unicenter  
ServicePlus

**Fig. 13 – Explorador de pastas**

 Request

**Status Change Request 55**

<b>Request Number</b> 55	<b>Request Summary</b> teste	<b>Time Stamp</b> 09/28/2005 12:06 pm	<b>Internal?</b> <input type="checkbox"/>
<b>Current Status</b> Cancelled	<b>New Status</b> Closed <input type="button" value="v"/>	<b>Date of Activity</b> 09/28/2005 12:06 pm	
<b>Time Spent</b> 00:20:00			
<b>User Description</b> This is a test to demonstrate a change status operation			

**Figura 14 – Formulário de uma actividade**

**55 Request Detail**





<b>Affected End User</b> Correia, Sergio	<b>Request Area</b> Por Definir	<b>Status</b> Cancelled	<b>Active?</b> NO
<b>Reported By</b> Correia, Sergio	<b>Configuration Item</b> Por Definir	<b>Priority</b> 6 - Low	<b>Location</b>
<b>Assignee</b>	<b>Group</b> SupportCentre	<b>External</b>	

<b>Summary</b> teste	<b>Total Activity Time</b> 00:00:19
-------------------------	--

<b>Description</b> teste
-----------------------------

<b>Open Date/Time</b> 06/01/2005 02:00 pm	<b>Last Modified</b> 06/09/2005 10:32 am	<b>Resolve Date/Time</b>	<b>Close Date/Time</b> 06/01/2005 03:00 pm
--	---	--------------------------	---

<b>1. Activities</b>	<b>2. Service Type</b>	<b>3. Parent / Child</b>	<b>4. Knowledge</b>
----------------------	------------------------	--------------------------	---------------------

<b>5. Attachments</b>	<b>6. Properties</b>
-----------------------	----------------------

**Search Logs****History**

Analyst	Date	Time Spent	Type	Summary
Correia, Sergio	06/09/2005 10:31 am	00:00:15	Update Status	'status' changed from 'Closed' to 'Cancelled'
Administrator	05/31/2005 08:48 pm	00:00:04	Close	'status' changed from 'Open' to 'Closed'
	05/31/2005 08:33 pm		Initial	create a new request/change/issue

**Fig. 15 – Formulário de um registo**

<input type="button" value="Show Filter"/> <input type="button" value="Search"/> <input type="button" value="Clear Filter"/> <input type="button" value="Edit in List"/>				
<b>37 found. Displaying 1 through 37.</b>				
<b>260</b>	Cancelled 06/02/2005 06:08 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Isto afinal ainda precisa de umas afinações.			
<b>247</b>	Cancelled 06/02/2005 05:12 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Ultimo teste			
<b>241</b>	Cancelled 06/02/2005 05:06 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Teste3. Teste real.			
<b>234</b>	Cancelled 06/02/2005 04:54 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Teste3. Teste real.			
<b>55</b>	Cancelled 06/01/2005 02:00 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	teste			
<b>53</b>	Cancelled 06/01/2005 02:00 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Segundo teste de abertura -----Original Message----- From:...			
<b>49</b>	Cancelled 06/01/2005 02:00 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee: End User: Correia, Sergio
Priority: 6 - Low	Este é um teste de abertura de email			
<b>111</b>	Cancelled 06/01/2005 02:00 pm	Mainroad-Serviços em tecnologias de informação, SA	SupportCentre	Assignee:

**Fig. 16 – Formulário de uma listagem de registos**

Correia, Sergio Analyst Detail - Unicenter ServicePlus Service Desk - Microsoft Internet Explorer

mainroad  
information technology

Request  Go

Close Window

### Correia, Sergio Analyst Detail

Last Name	First Name	Middle Name	Status
Correia	Sergio		Active
Password	System Login	Contact Type	Service Type
		Analyst	
Time Zone	Position	Data Partition	Access Type
Work Schedule	Available		
	No		

Phone Number	Extension	Cell Phone	Fax Number
Email Address	Pager Email Address		
sergio.correia@mainroad.pt			
Notifications	Low	Normal	High
Method	Email	Email	Email
Emergency			Email
Workshifts for Notification Methods			

Fig. 17 – Formulário de um utilizador

File View Search Reports Windows Help Logoff

Hide Filter Search Clear Filter

Last Name	First Name	Middle Name	Contact Type
			<empty>
Active	Email	System Login	Access Type
Active			<empty>
Location	Organization	Department	
Phone Number	Extension	Cell Phone	Less...
Additional Search Arguments			

5 contacts found. Displaying 1 through 5.

<< Top | < Prev | Next > | Bottom >> | List All

Name	Type	Email	Organization	Login	Phone Number
Correia, Sergio	Analyst	sergio.correia@mainroad.pt	Mainroad-Serviços em tecnologias de informação, SA		
	Analyst		Mainroad-Serviços em tecnologias de informação, SA		
	Analyst		Mainroad-Serviços em tecnologias de informação, SA		
	Manager		Mainroad-Serviços em tecnologias de informação, SA		
	Analyst		Mainroad-Serviços em tecnologias de informação, SA		

Fig. 18 – Formulário de uma listagem de utilizadores

**Configuration Item Detail**

Name	CI Class	Family	Active?
	Server	Computer	Active
Notify 75%	Notify 75%	Notify 75%	
Notify 100%	Notify 100%	Notify 100%	
Notes			

1. Inventory    2. Properties    3. Log    4. Service  
5. Location    6. Contacts    7. Organizations    8. Related CI's    9. Requests

CI Number	Contract	EID
Manufacturer	Model	Serial Number
HP	Proliant DL360 G3	Function
Acquisition Date	Expiration Date	Contract Start Date
		Contract End Date

Fig. 19 – Formulário de um equipamento

1. Inventory    2. Properties    3. Log    4. Service  
5. Location    6. Contacts    7. Organizations    8. Related CI's    9. Requests

**Computer Configuration Item Properties**

IP	MAC	Operating System
		Win 2003 SE
Physical Memory	Power Supplies	Processor Speed
1023 MB		4 * 3065
		Hard Disk Space
		2 * 48595 MB

Fig. 20 – Formulário das propriedades de um equipamento

Show Filter    Search    Clear Filter

3 configuration items found. Displaying 1 through 3.    << Top | < Prev | Next > | Bottom >> | List All

Name	Class	Organization	#Req	Status
	Server	Mainroad-Serviços em tecnologias de informação,SA	0	Active
	Server	Mainroad-Serviços em tecnologias de informação,SA	0	Active
	Server	Mainroad-Serviços em tecnologias de informação,SA	0	Active

Fig. 21 – Formulário de uma listagem de equipamentos

**Mainroad-Serviços em tecnologias de informação,SA Organization Detail**

Name	Group	Fiscal Number
Mainroad-Serviços em tecnologias de informação,SA	SonaCom	PT-506518108
Address	Location	Code
Rua Henrique Pousão, 432-2º	NorteShopping- Senhora da Hora	4461-901

Environment		
Name	Family	Class
	Computer	Server
	Computer	Server
	Computer	Server
	Computer	Server
	Computer	Server

Fig.22 – Formulário de uma organização

**mainroad**  
Information Technology

User by Phone

**Contact**

Correia, Sergio

1. Information  
2. Environment  
3. Request History

**Organization**  
Mainroad-Serviços em tecnologias de informação,SA

4. Information  
5. Environment  
6. Request History

**Contact Information for Correia, Sergio**

<b>Last Name</b>	<b>First Name</b>	<b>Middle Name</b>	<b>Status</b>
Correia	Sergio		Active
<b>System Login</b>	<b>Organization</b>	<b>Email Address</b>	
spcorreia	Mainroad-Serviços em tecnologias de informação,SA	sergio.correia@mainroad.pt	
<b>Phone Number</b>	<b>Extension</b>	<b>Cell Phone</b>	<b>Fax Number</b>
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
<b>Service Type</b>	<b>Data Partition</b>	<b>Time Zone</b>	<b>Access Type</b>
			[REDACTED]
<b>Location</b>	<b>Site</b>	<b>Contact Type</b>	
		Analyst	
<b>Address</b>			
Edificio NorteShopping Rua Henrique Pousão, 432, Matosinhos 4461 - 901 Sra PT			
<b>Contact Notes</b>			
Scratchpad			
<input type="button" value="New Request"/> <input type="button" value="Clear Scratchpad"/>			

Fig. 23 – Formulário do Profile Browser

**mainroad**  
Information Technology

Request

**Request Summary** As Of: 09/28/2005 12:10:24 pm

Request#	Customer	Assignee	Open Date	Status	Pri
4335	Correia, Sergio Summary: sdhsudbsi h dofughasgg		09/08/2005	Cancelled	6 - Low
4317	Correia, Sergio Summary: Teste de registo de logs para os registos violados.		09/08/2005	Cancelled	6 - Low
4314	Correia, Sergio Summary: teste de notificação de grupo.		09/08/2005	Cancelled	6 - Low
4313	Correia, Sergio Summary: Teste de notificação para os grupos.		09/08/2005	Cancelled	6 - Low
4312	Correia, Sergio Summary: Teste de registo de tempo de resolução de SLA.		09/08/2005	Cancelled	6 - Low
4284	Correia, Sergio Summary: Isto é um teste. Por favor não fechem o registo. Obrigado....		09/07/2005	Cancelled	6 - Low
4283	Correia, Sergio Summary: This is a test.		09/07/2005	Cancelled	6 - Low
4017	Correia, Sergio Summary: Não fechar este registo por favor! Obrigado, Sérgio Correi...		08/31/2005	Cancelled	6 - Low
4018	Correia, Sergio Summary: Não fechar este registo por favor! Obrigado, Sérgio Correi...		08/31/2005	Cancelled	6 - Low
4014	Correia, Sergio Summary: Não fechar este registo por favor! Obrigado, Sérgio Correi...		08/31/2005	Cancelled	6 - Low

Fig. 24 – Formulário de um relatório

## 2. Inventário

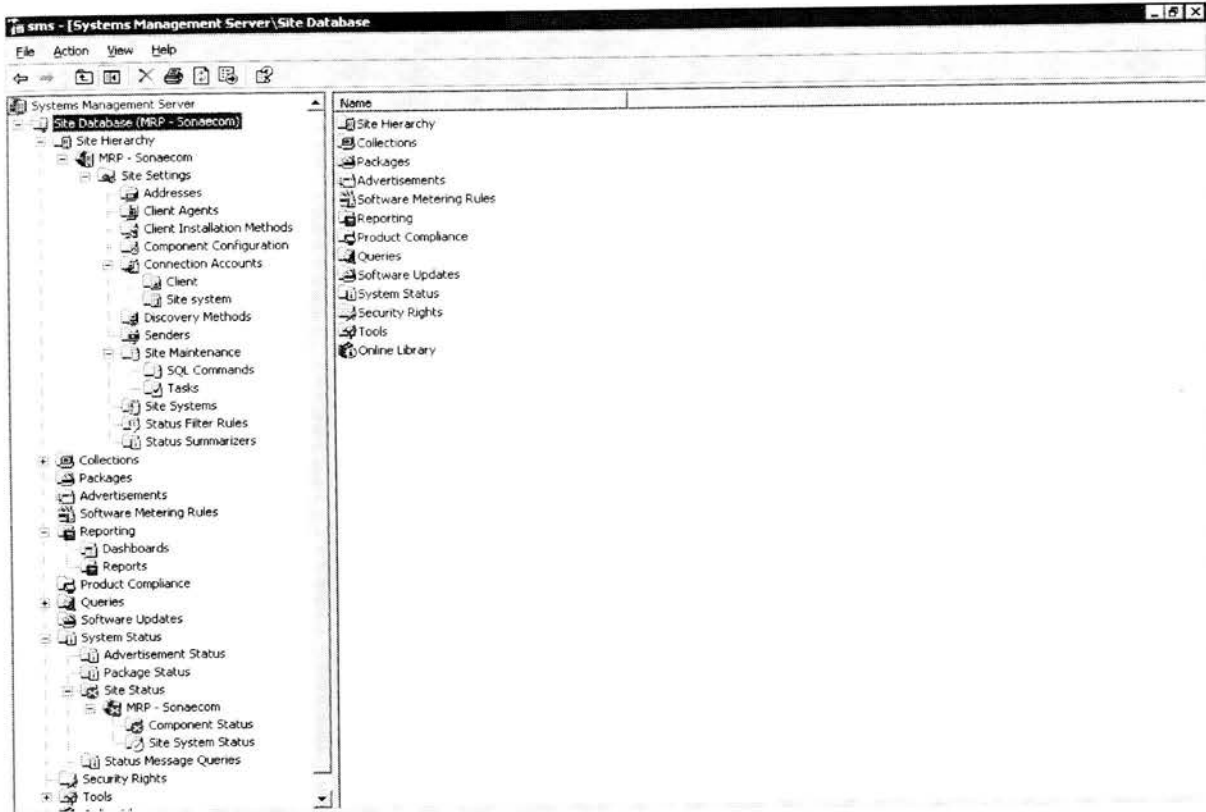


Fig. 25 – Consola administrativa do SMS

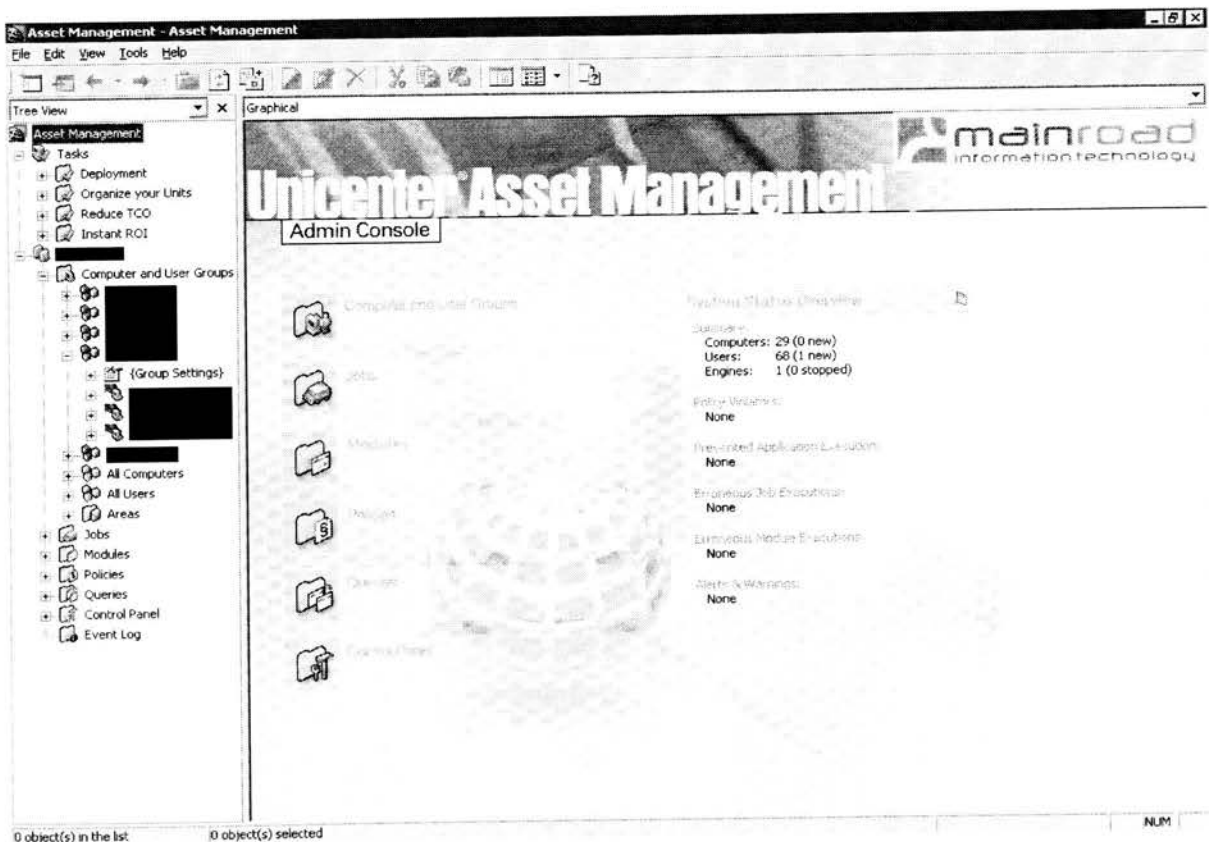


Fig. 26 – Consola administrativa do UAM



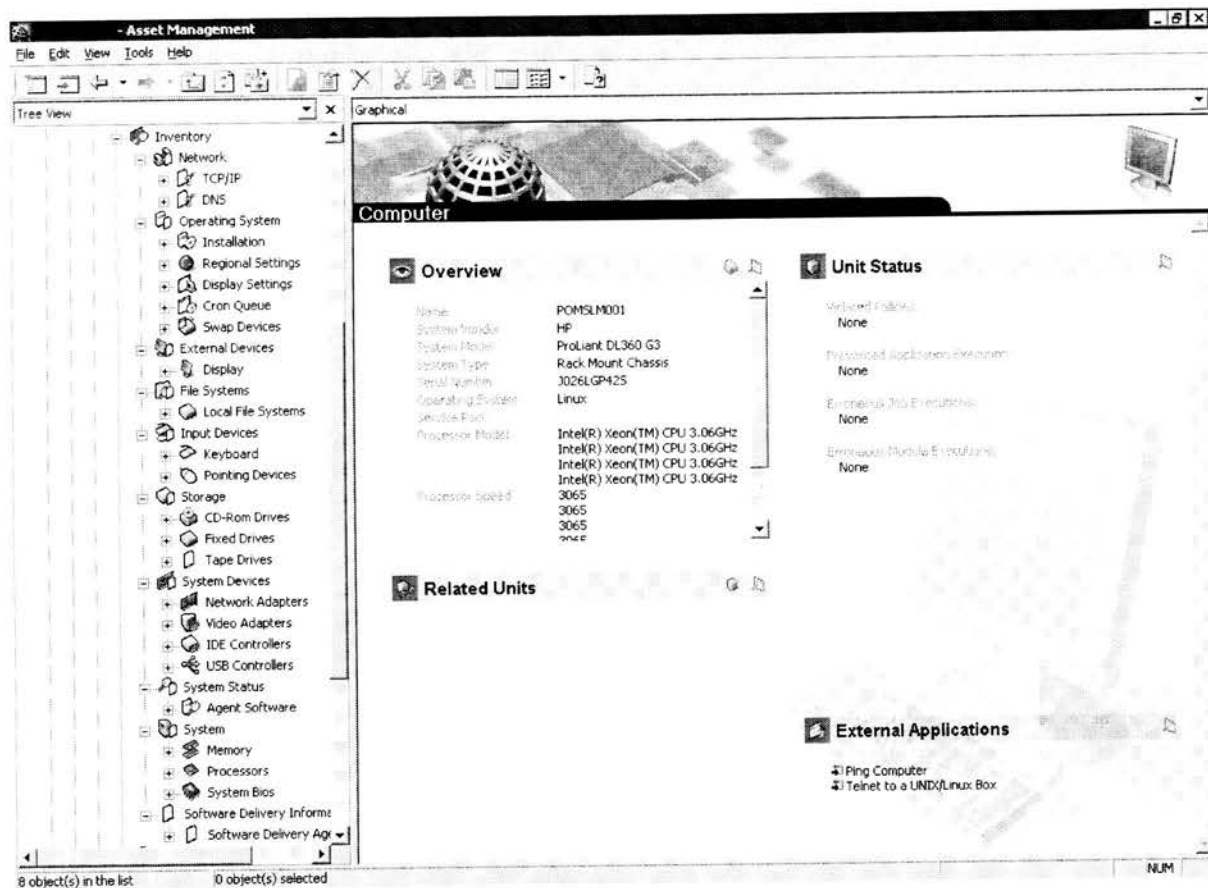


Fig. 27 – Inventário de um computador

**[REDACTED]** - 9/23/2005

Result contains 12 rows.

Name	System   Vendor	System   Model	System   Serial Number	System   Processor Count	System   Total Memory	Operating System   Operating System
[REDACTED]	Hewlett-Packard Company	9000/800/S16K-A	DEH423413L	7	16368 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/S16K-A	DEH423413L	7	16368 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/N4000-36	GBM402146Y	8	12288 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/L1000-44	USS411722F	2	2048 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/L1000-44	USS412451N	2	3072 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/813/D230	2008504543	1	512 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/898/K370	674623362	1	512 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/N4000-44	GBM39371XF	2	1536 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/898/K370	1608643332	6	2048 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/L3000-7x	DEH4418V8D	2	2048 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/820/D280	2000181215	2	2560 MB	HP-UX
[REDACTED]	Hewlett-Packard Company	9000/800/K370	1279364181	6	2048 MB	HP-UX

Fig. 28 – Relatório global de um cliente

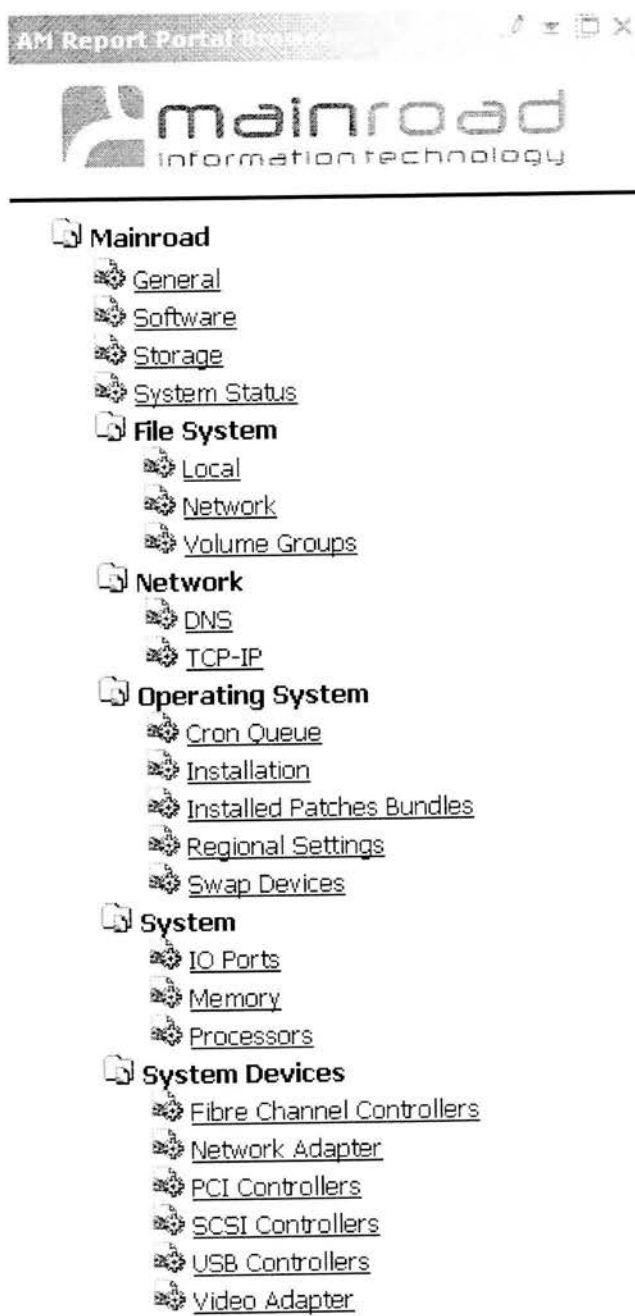


Fig. 29 – Explorador de pastas

### 3. Monitorização

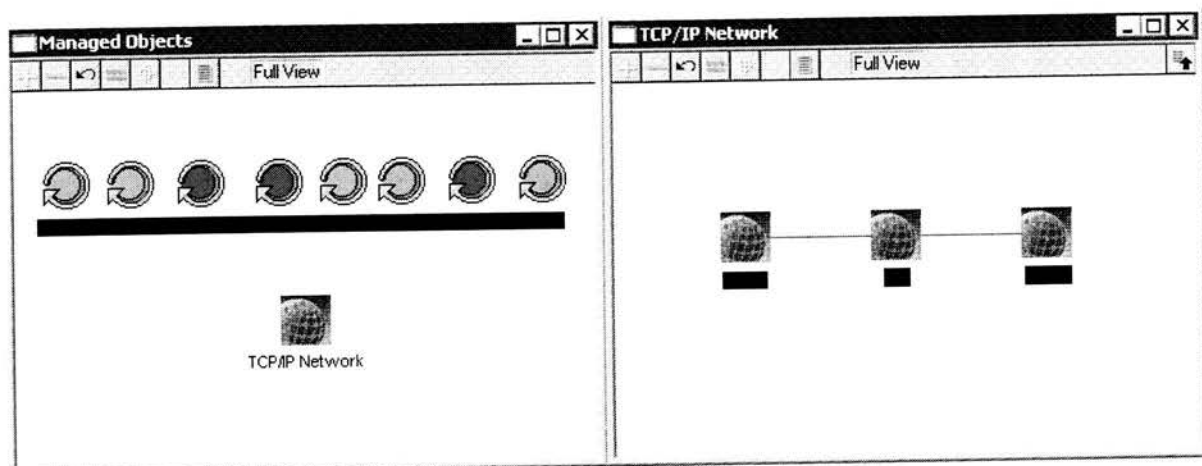


Fig. 30 – 2D Map – Visualização de BPV e objectos

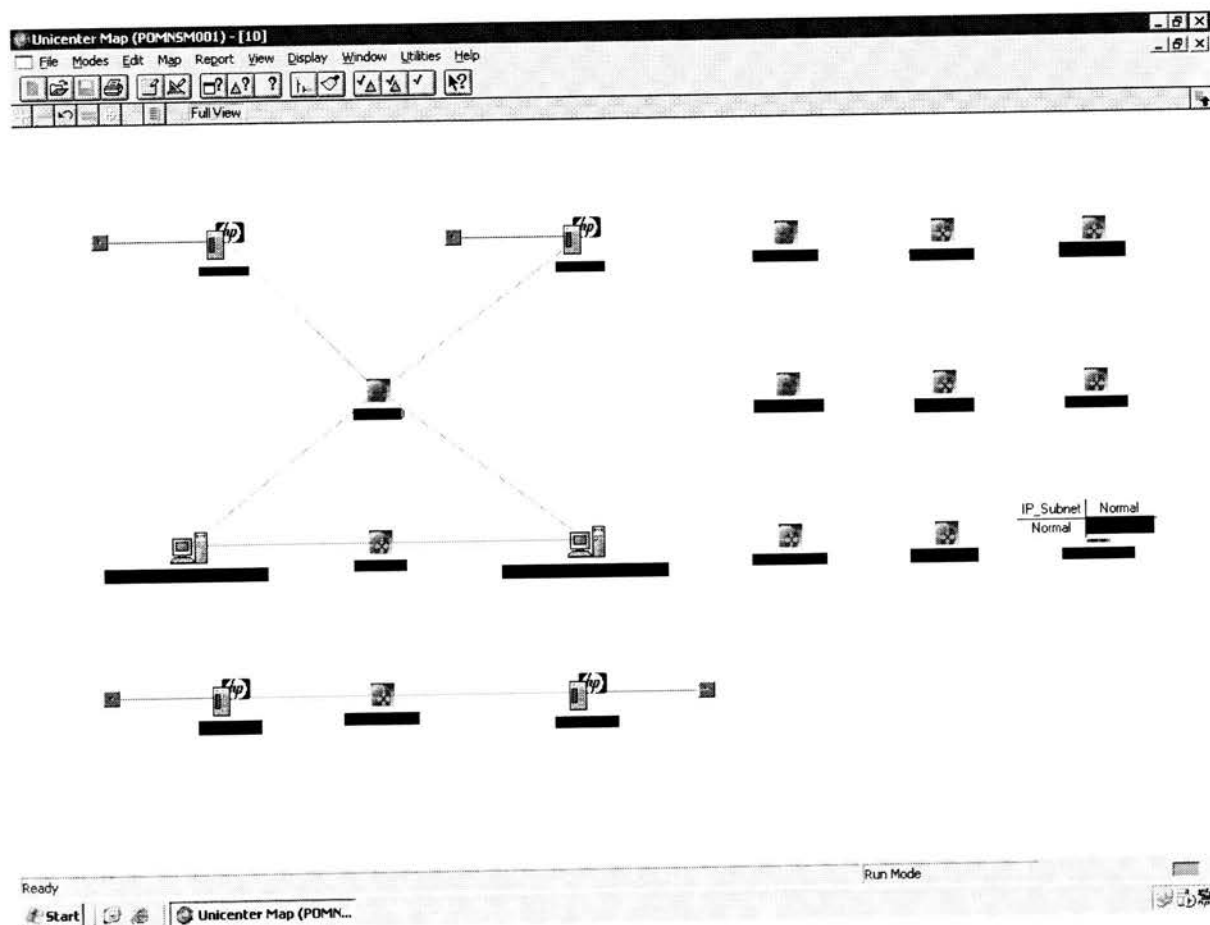


Fig. 31 – 2D Map – Visualização de redes

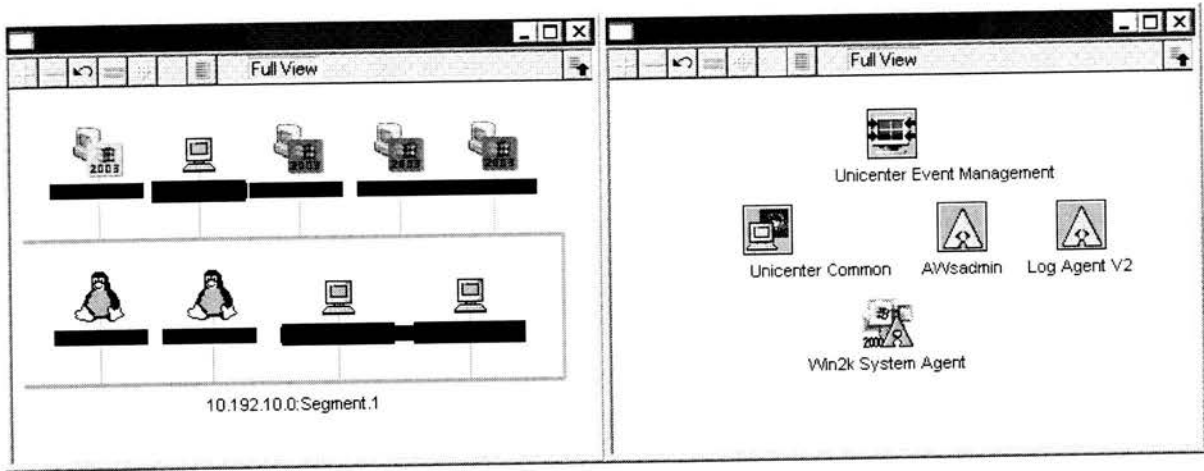


Fig. 32 – 2D Map – Visualização de equipamentos e agentes

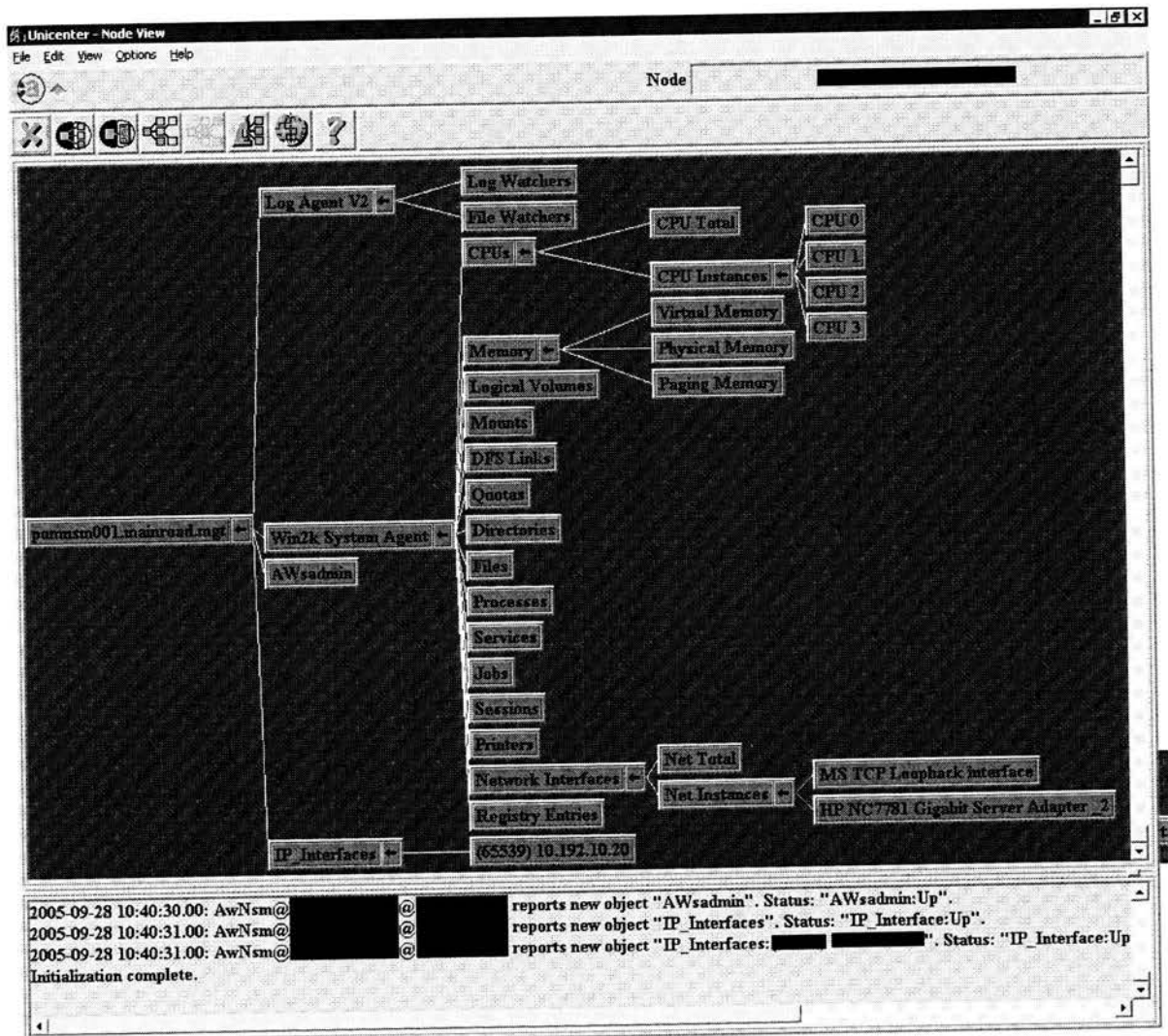


Fig. 33 – NodeView – Visualização de um nó

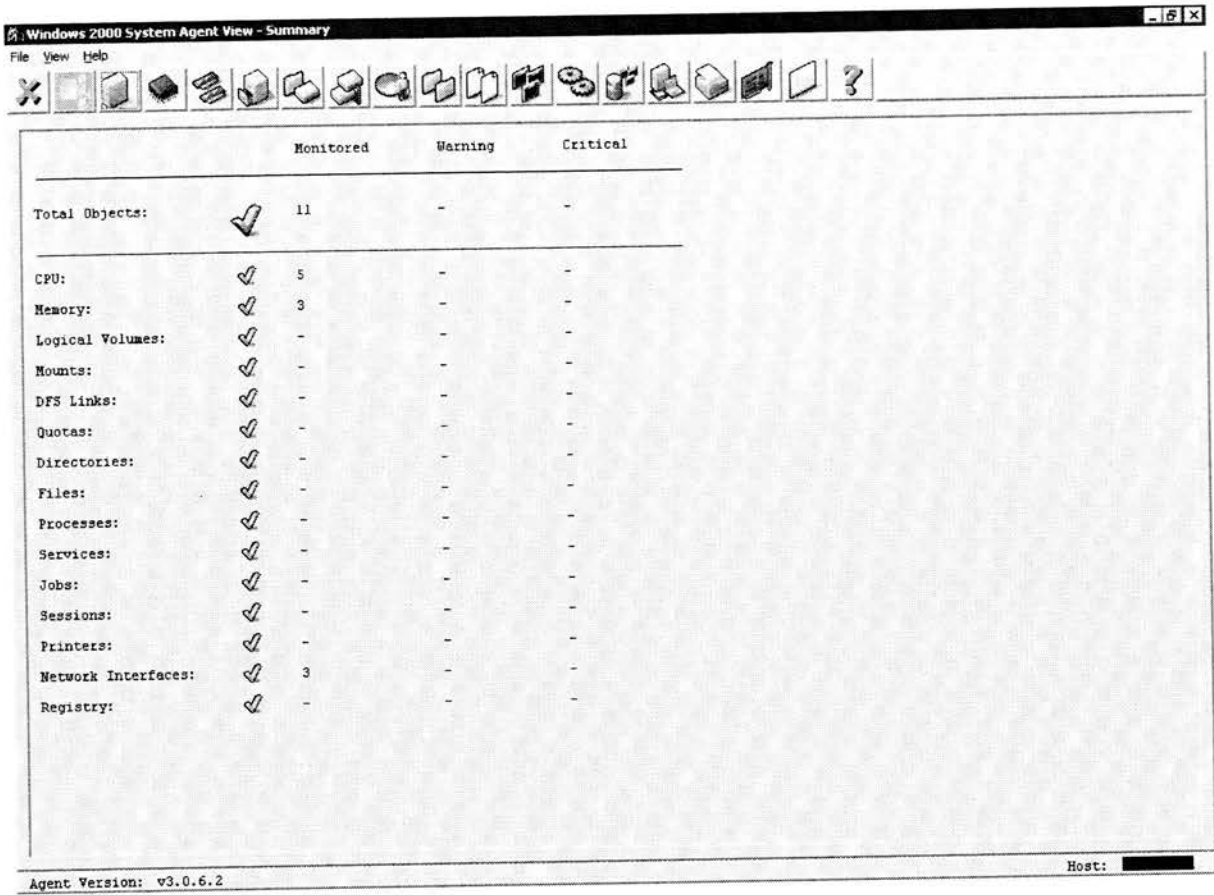


Fig. 34 – AgentView – Visualização sumariada

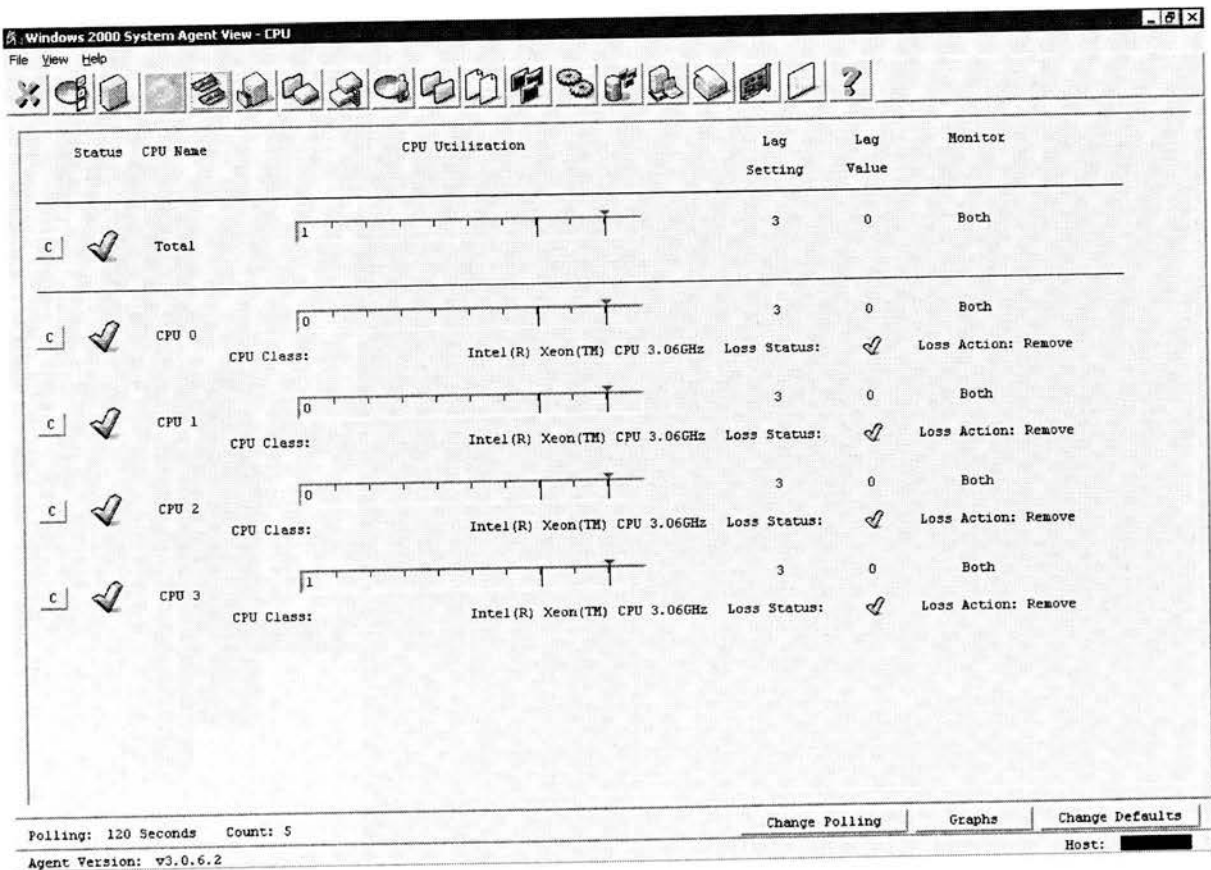


Fig. 35 – AgentView – Visualização da performance do CPU



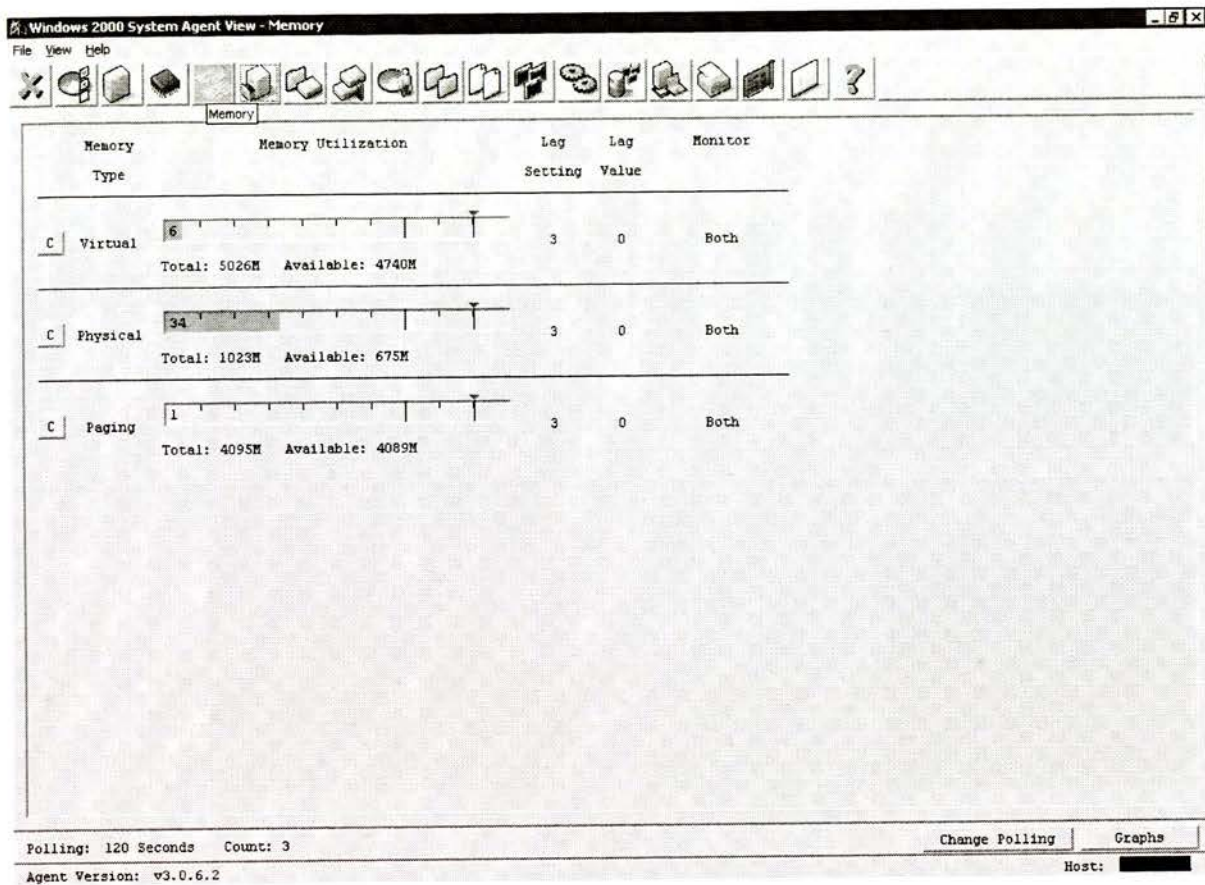


Fig. 36 – AgentView – Visualização da performance da memória

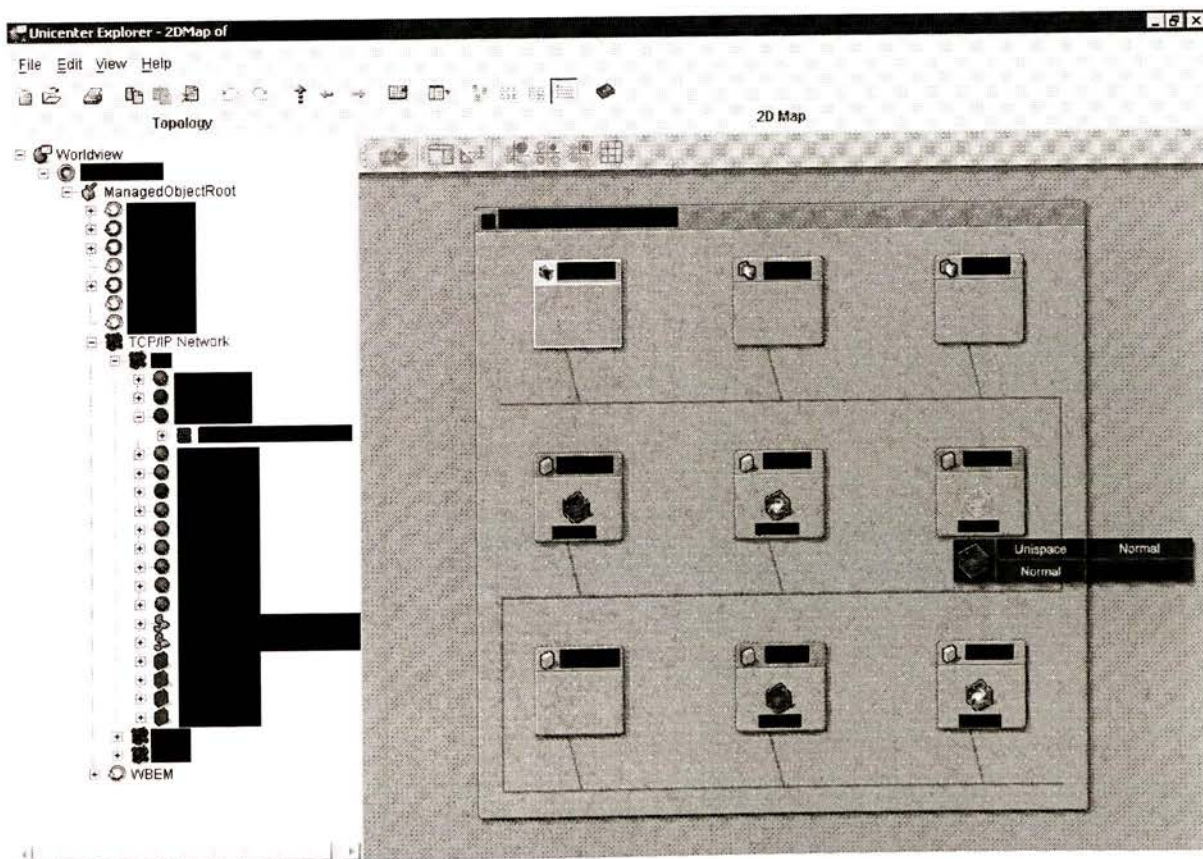


Fig. 37 – Explorer – Visualização de equipamentos



Console Log

Held Messages - Today's log

Row	Time Cr...	Node	User	Station	Message
	18:56:46		N/A		(1) Service Control Manager_7000_E: The Data Protector installation Service service failed

Log Messages - Today's log

Row	Time Cr...	Node	User	Station	Message
15577	14:27:27				%CATD_I_060, SNMPTRAP -c public
15578	14:27:27				%CATD_I_060, SNMPTRAP -c public
15579	14:27:28				%CATD_I_060, SNMPTRAP -c public
15580	14:27:29				%CATD_I_060, SNMPTRAP -c public
15581	14:27:30				%CATD_I_060, SNMPTRAP -c public
15582	14:27:31				%CATD_I_060, SNMPTRAP -c public
15583	14:27:31				%CATD_I_060, SNMPTRAP -c public
15584	14:27:33				%CATD_I_060, SNMPTRAP -c public
15585	14:27:34				Security_680_S: Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
15586	14:27:34				Security_552_S: Logon attempt using explicit credentials: Logged on user: User Name: [REDACTED]
15587	14:27:34				Security_540_S: Successful Network Logon: User Name: [REDACTED] Domain: [REDACTED]
15588	14:27:34				Security_576_S: Special privileges assigned to new logon: User Name: [REDACTED]
15589	14:27:34				Security_538_S: User Logoff: User Name: [REDACTED] Domain: [REDACTED]
15590	14:27:34				%CATD_I_060, SNMPTRAP -c public
15591	14:27:34				%CATD_I_060, SNMPTRAP -c public
15592	14:27:36				%CATD_I_060, SNMPTRAP -c public
15593	14:27:37				%CATD_I_060, SNMPTRAP -c public
15594	14:27:37				%CATD_I_060, SNMPTRAP -c public
15595	14:27:38				%CATD_I_060, SNMPTRAP -c public
15596	14:27:40				%CATD_I_060, SNMPTRAP -c public
15597	14:27:41				%CATD_I_060, SNMPTRAP -c public
15598	14:27:43				%CATD_I_060, SNMPTRAP -c public
15599	14:27:44				%CATD_I_060, SNMPTRAP -c public
15600	14:27:46				%CATD_I_060, SNMPTRAP -c public
15601	14:27:47				%CATD_I_060, SNMPTRAP -c public
15602	14:27:49				%CATD_I_060, SNMPTRAP -c public

Command:

Status: Running      Polling: Off      Sep 28, 2005 02:28:05 PM

Fig. 38 – Explorer – Consola de logs

Configuration

Monitoring

Configuration Hierarchy

- PerfMgmtMs2000
  - ms2000-default

Section

- Scan Types
  - Access Methods
  - Buffer Manager
  - Cache Manager
  - Databases
  - General Statistics
  - Individual Scan Types
  - Latches
  - Locks
  - Memory Manager
  - Replication Agents
  - Replication Distribution
  - Replication Logreader
  - Replication Merge
  - Replication Snapshot
  - SQL Statistics
- Events
  - bufhitratelow
  - deadlockshigh
  - genlimit
  - hitratelow
  - logusedpct
  - mssqlsrvdown
  - processwaiting
  - tablelockshigh
- Processes
  - Driver
  - Scan Pmon
  - Archive
  - Purge

Parameter Values

	PerfMgmtMs2000	ms2000-default
archive	on	on
compression	on	on
archive_retention	3	3

OK      Cancel      Apply      Help

Fig. 39 – Consola administrativa do UDPM

UDPM

Status	Name	Node	Type
Up	cb2-dbc271@COMP001	COMP001	cb2
Up	cb2-dbc271@COMP002	COMP002	cb2
Up	cb2-dbc272@COMP003	COMP003	cb2
Up	cb2-dbc272@COMP004	COMP004	cb2
Up	cb2-dbc21@COMP005	COMP005	cb2
Up	cb2-dbc2@COMP006	COMP006	cb2
Up	cb2-dbc2@COMP007	COMP007	cb2
Up	cb2-linuxp01@COMP008	COMP008	cb2
Up	ms2000-COMP008@COMP006	COMP006	ms2000
Down	ms7-COMP008@COMP008	COMP008	ms7
Up	ora-21x02@COMP009	COMP009	ora
Up	ora-SU10901@COMP010	COMP010	ora
Up	unix@COMP009	COMP009	unix
Up	unix@COMP010	COMP010	unix
Up	w2k@COMP005	COMP005	w2k
Up	w2k@COMP011	COMP011	w2k
Down	wt@COMP008	COMP008	wt
Up	wt@COMP010	COMP010	wt

COMP001:8080, showing 18 of 19 agents.

Fig. 40 – Agentes monitorizados

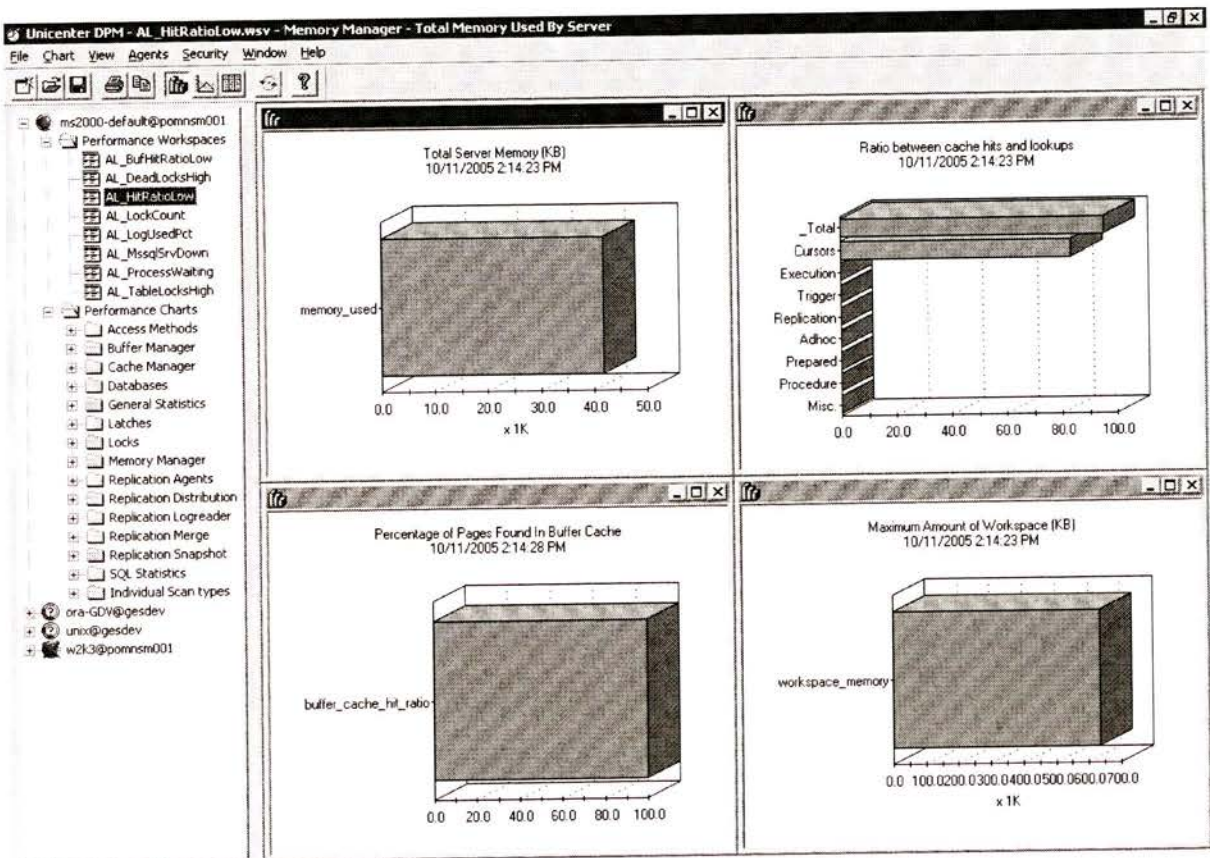


Fig. 41 – Visualização global da memória consumida



The screenshot shows the configuration interface for a monitoring view. On the left is a tree view of metrics and agents. The 'Details' tab is active, showing configuration for the 'AL\_CluExtMaxOut' view. The 'Chart' section has 'Strip' selected. The 'Agent' section shows 'Name: Default agent instance', 'Type: cra', and 'Node: Default agent node'. The 'Scan' section shows 'Scan name: Maximum Number of Extents', 'Scan Internal Name: clumax', 'Group Name: Cluster Extent Info', and 'Group Internal Name: clustgroup'. Below this is a 'Layout of the "AL\_CluExtMaxOut" view' section containing four charts: two horizontal bar charts for 'Maximum Number of Extents' and 'Allocated Extents', and two line charts for 'Maximum Number of Extents' and 'Allocated Extents' over time. Buttons for 'Open', 'Add chart', 'Del chart', 'Save layout', and 'Advisor' are visible.

Fig. 42 – Definição de um formulário

UDPM

Status	Name	Node	Type
Up	cb2-4b271@COMP001	COMP001	cb2
Up	cb2-4b271@COMP002	COMP002	cb2
Up	cb2-4b272@COMP003	COMP003	cb2
Up	cb2-4b272@COMP004	COMP004	cb2
Up	cb2-4b2c1@COMP005	COMP005	cb2
Up	cb2-4b2@COMP006	COMP006	cb2
Up	cb2-4b2@COMP007	COMP007	cb2
Up	cb2-4b2@COMP008	COMP008	cb2
Up	cb2-4b2@COMP009	COMP009	cb2
Up	cb2-4b2@COMP010	COMP010	cb2
Up	ms2000-COMP006@COMP006	COMP006	ms2000
Down	ms7-COMP008@COMP008	COMP008	ms7
Up	cra-41X92@COMP009	COMP009	cra
Up	cra-SU10901@COMP010	COMP010	cra
Up	unix@COMP009	COMP009	unix
Up	unix@COMP010	COMP010	unix
Up	w2k@COMP006	COMP006	w2k
Up	w2k@COMP011	COMP011	w2k
Down	wt@COMP008	COMP008	wt
Up	wt@COMP010	COMP010	wt

COMP001:8080, showing 18 of 18 agents.

Fig. 43 – Visualização dos agentes monitorizados

View: cpu (unix@COMP009)

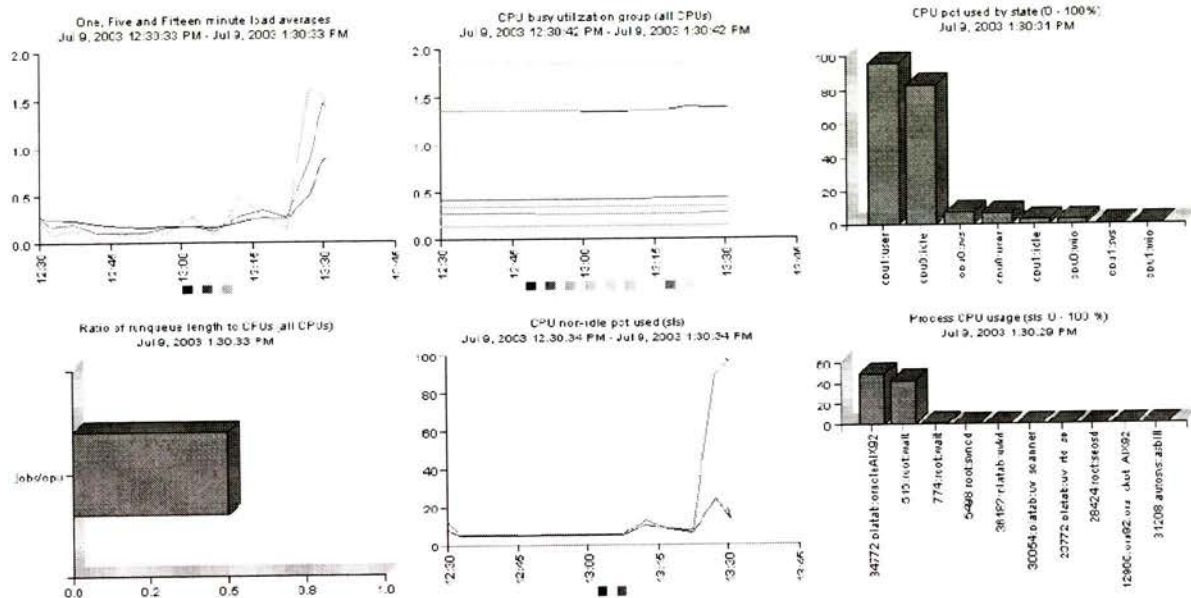


Fig. 44 – Visualização da memória consumida

Health Overview: ora-AIX92@COMP009

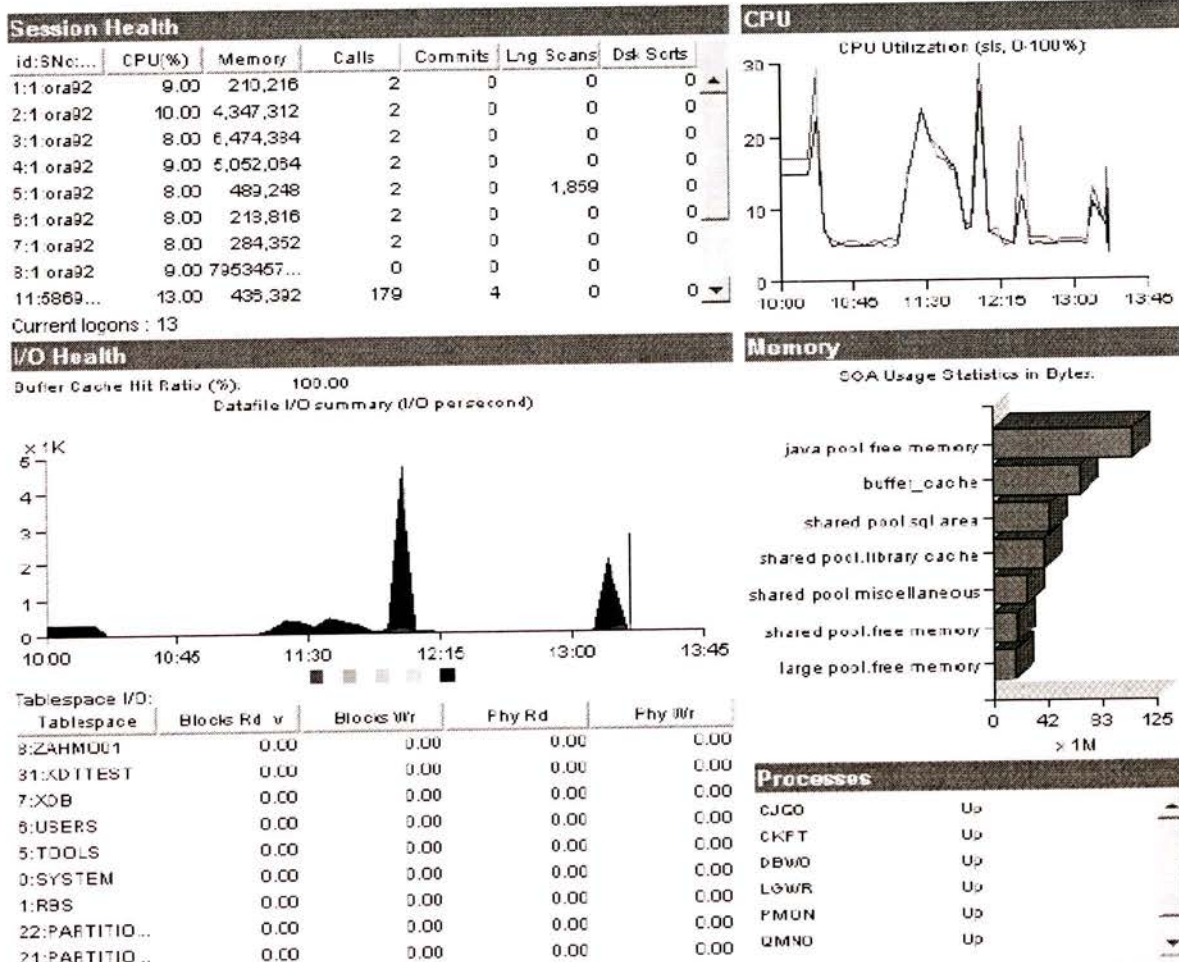


Fig. 45 – Visualização de um agente de sistema

The screenshot displays the SAP Administration Console interface. The left pane shows a tree view of system components, with 'Workload Analysis' selected. The right pane shows a table of workload details for instance (45) during the analysis period of 01/05/2005 14:59:18 to 01/05/2005 15:04:23.

**Workload Details Table:**

End T...	TCode	Program	Task	Screen	WP	User	Resp Ti
14:59:18		SAPMSSY1	R	3004	2	SAP*	
14:59:18		SAPMSSY1	R	3004	1	SAP*	
15:01:14		SAPMSSY1	R	3004	2	SAP*	
15:01:14		SAPMSSY1	R	3004	2	SAP*	
15:01:15		SAPMSSY1	R	3004	2	SAP*	
15:01:16		SAPMSSY1	R	3004	2	SAP*	
15:01:19		SAPMSSY1	R	3004	1	SAP*	
15:02:13		SAPMSSY1	R	3004	2	SAP*	
15:03:13		SAPMSSY1	R	3004	2	SAP*	
15:03:14		SAPMSSY1	R	3004	2	SAP*	
15:03:20		SAPMSSY1	R	3004	1	SAP*	
15:04:23		SAPMSSY1	R	3004	2	SAP*	

Below the table, there are buttons for 'Refresh', 'Settings', and 'Status and Triggers Configuration'. The bottom status bar shows 'Workload Details' and 'Standard'.

Fig. 46 – Consola administrativa do USAP



## ANEXO B: Código

Neste anexo encontra-se o código dos programas desenvolvidos. Por uma questão de confidencialidade, os nomes de utilizadores, palavras-chave, nomes de computadores, endereços IP, bem como outras informações, são omitidos.

### 1. Importação de inventário

#### file.pl – equipamentos do ficheiro

```
#open the files with the assets, models, manufacturers, type and the two types of
customers
open(FILE, "novo.txt");
@lines = <FILE>;
close(FILE);

open(FILE, "modelos.txt");
@models = <FILE>;
close(FILE);

open(FILE, "marcas.txt");
@mans = <FILE>;
close(FILE);

open(FILE, "tipos.txt");
@types = <FILE>;
close(FILE);

open(FILE, "clients.txt");
@clients = <FILE>;
close(FILE);

open(FILE, "properties.txt");
@properties = <FILE>;
close(FILE);

#remove the end of the line character
chomp foreach (@lines);
chomp foreach (@models);
chomp foreach (@mans);
chomp foreach (@types);
chomp foreach (@clients);
chomp foreach (@properties);

#convert lists into hashes
%model = @models;
%man = @mans;
%type = @types;
%client = @clients;
%property = @properties;

#define the start id
$i = 199;

#prints the table fields
print "TABLE Network_Resource\n";
```



```

print "id nr_serial_num nr_loc_id nr_room nr_cabinet nr_prim_search_key nr_grc_id
nr_family nr_mdl_id nr_mfr_id nr_org_id nr_maint_org_id nr_license_num
nr_wrty_end_dt nr_desc\n";

#prepare the information to be treated
foreach $line (@lines) {
    @values = split(/;/, $line);
    ++$i;

    if ($values[4] eq "Vazio") {
        $serial = "";
    }
    else {
        $serial = $values[4];
    }

    if ($values[8] eq "Norte Shopping") {
        $local = "1048";
    }
    else {
        $local = "1012";
    }

    $marca = $man{$values[13]};

    $mod = "$values[14];$values[13]";
    $modelo = $model{$mod};

    $tipo = $type{$values[12]};

    if ($tipo == 2818) {
        $fam = 3;
    }
    else {
        $fam = 0;
    }

    $cliente = lc($values[5]);
    $prop = lc($values[7]);

    $cli = $client{$cliente};
    $pro = $property{$prop};

    if ($values[3] =~ /-.-/) {
        @data = split(/-/, $values[3]);
        $date = "$data[1]/$data[0]/$data[2] 14:00:00";
    }
    else {
        $date = "";
    }

    if ($values[18] eq "Vazio") {
        $obs = "";
    }
    else {
        $obs = $values[18];
    }

    #print assets in the correct format of importation
    print "{ \"$i\" , \"$serial\" , \"$local\" , \"$values[9]\" , \"$values[10]\"
, \"$values[11]\" , \"$tipo\" , \"$fam\" , \"$modelo\" , \"$marca\" , \"$cli\" , \"$pro\"
, \"$values[6]\" , \"$date\" , \"$obs\" } \n";
}

```

**repetidos.pl – equipamentos repetidos da primeira aplicação**

```

#open the files with the assets, models, customers, sla, functions, rack, IP and MAC
address
open(FILE, 'repetidos.txt');
@lines = <FILE>;
close(FILE);

open(FILE, 'm.txt');
@modelos = <FILE>;
close(FILE);

open(FILE, 'orgs.txt');
@orgs = <FILE>;
close(FILE);

open(FILE, 'sla.txt');
@slas = <FILE>;
close(FILE);

open(FILE, 'func.txt'); #Ficheiro com as funcoes
@funcs = <FILE>;
close(FILE);

open(FILE, 'bastmap.txt'); #Ficheiro com os bastidores
@basts = <FILE>;
close(FILE);

open(FILE, 'ipmac.txt'); #Ficheiro com os macs e os ips
@ipmacs = <FILE>;
close(FILE);

#remove the end of the line character
chomp foreach(@modelos);
chomp foreach(@orgs);
chomp foreach(@slas);
chomp foreach(@funcs);
chomp foreach(@basts);
chomp foreach(@ipmacs);

#convert lists into hashes
%modelo = @modelos;
%org = @orgs;
%sla = @slas;
%func = @funcs;
%bast = @basts;
%ipmac = @ipmacs;

#prepare the information to be treated
foreach $line (@lines) {
  chomp $line;
  @values = split(/;/, $line);
  #push(@ids, $values[0]);

  $scnumber = "SC - $values[0]";

  if (exists($modelo{$values[0]})) {
    $modman = $modelo{$values[0]}
  }
  else {
    $modman = "1;1";
  }
}

```

```

if ($values[9] eq "Vazio") {
    $supp_date = "Vazio";
}
else {
    @date = split(/-/, $values[9]);
    $supp_date = "$date[1]/$date[0]/$date[2] 14:00:00";
}

if ($values[12] ne "Vazio") {
    $obs = $values[12];
}
else {
    $obs = "Vazio";
}

$client = $org{$values[3]};

if (exists($bast{$values[0]})) {
    $bas = $bast{$values[0]};
}
else {
    $bas = "Vazio";
}

if (exists($func{$values[0]})) {
    $fun = $func{$values[0]};
}
else {
    $fun = "Vazio";
}

if (exists($ipmac{$values[0]})) {
    $ip = $ipmac{$values[0]}
}
else {
    $ip = "Vazio;Vazio";
}

if ($values[6] eq "S") {
    $del = 0;
}
elseif ($values[6] eq "N") {
    $del = 1;
}
elseif ($values[6] eq "R") {
    $del = 1;
}
else {
    print "$line";<STDIN>;
}

$equip =
"$scnumber;2818;3;$modman;$supp_date;$del;$obs;$client;$values[1];$bas;$fun;$ip;";
$techs{$values[0]} = $equip;
}

#open the asset information
open(FILE, 'soc.txt');
@lines = <FILE>;
close(FILE);

#prepare the information to be treated
foreach $line (@lines) {

```

```

@values = split(/;/,$line);
if (exists($techs{$values[0]}) {
    $socs{$values[0]} = $line;
}
}

foreach $tech (keys %techs) {

    if (exists($socs{$tech})) {
        @values = split(/;/,$socs{$tech});

        $property = $org{$values[1]};
        $slas = $sla{$values[6]};

        if ($values[10] eq "Vazio") {
            $buy_date = "Vazio";
        }
        else {
            @date = split(/-/, $values[10]);
            $buy_date = "$date[1]/$date[0]/$date[2] 14:00:00";
        }

        if ($values[11] eq "Vazio") {
            $cont_date = "Vazio";
        }
        else {
            @date = split(/-/, $values[11]);
            $cont_date = "$date[1]/$date[0]/$date[2] 14:00:00";
        }

        $equip = $techs{$tech} .
"$property;$values[5];$slas;$buy_date;$cont_date";
    }
    else {
        $equip = $techs{$tech} . "Vazio;Vazio;Vazio;Vazio;Vazio";
    }

    print "$equip\n";
}

```

### diferentes.pl – equipamentos diferentes da primeira aplicação

```

#open the files with the assets, models, customers, sla, functions, rack, IP and MAC
address
open(FILE, 'diferentes.txt');
@lines = <FILE>;
close(FILE);

open(FILE, 'm.txt');
@modelos = <FILE>;
close(FILE);

open(FILE, 'orgs.txt');
@orgs = <FILE>;
close(FILE);

open(FILE, 'sla.txt');
@slas = <FILE>;
close(FILE);

open(FILE, 'func.txt');
@funcs = <FILE>;
close(FILE);

```

```

open(FILE, 'bastmap.txt');
@basts = <FILE>;
close(FILE);

open(FILE, 'ipmac.txt');
@ipmacs = <FILE>;
close(FILE);

#remove the end of the line character
chomp foreach(@modelos);
chomp foreach(@orgs);
chomp foreach(@slas);
chomp foreach(@funcs);
chomp foreach(@basts);
chomp foreach(@ipmacs);

#convert lists into hashes
%modelo = @modelos;
%org = @orgs;
%sla = @slas;
%func = @funcs;
%bast = @basts;
%ipmac = @ipmacs;

#prepare the information to be treated
foreach $line (@lines) {
    chomp $line;
    @values = split(/;/, $line);

    $scnumber = "SC - $values[0]";

    if (exists($modelo{$values[0]})) {
        $modman = $modelo{$values[0]};
    }
    else {
        $modman = "1;1";
    }

    if ($values[9] eq "Vazio") {
        $supp_date = "Vazio";
    }
    else {
        @date = split(/-/, $values[9]);
        $supp_date = "$date[1]/$date[0]/$date[2] 14:00:00";
    }

    if ($values[12] ne "Vazio") {
        $obs = $values[12];
    }
    else {
        $obs = "Vazio";
    }

    $client = $org{$values[3]};

    if (exists($bast{$values[0]})) {
        $bas = $bast{$values[0]};
    }
    else {
        $bas = "Vazio";
    }

    if (exists($func{$values[0]})) {

```



```

        $fun = $func{$values[0]};
    }
    else {
        $fun = "Vazio";
    }

    if (exists($ipmac{$values[0]})) {
        $ip = $ipmac{$values[0]}
    }
    else {
        $ip = "Vazio;Vazio";
    }

    if ($values[6] eq "S") {
        $del = 0;
    }
    elsif ($values[6] eq "N") {
        $del = 1;
    }
    elsif ($values[6] eq "R") {
        $del = 1;
    }
    else {
        print "$line";<STDIN>;
    }

    $equip =
"$scnumber;2818;3;$modman;$supp_date;$del;$obs;$client;$values[1];$bas;$fun;$ip;";
    $techs{$values[0]} = $equip;
}

#open the asset information
open(FILE, 'soc.txt');
@lines = <FILE>;
close(FILE);

#prepare the information to be treated
foreach $line (@lines) {
    @values = split(/;/,$line);
    if (exists($techs{$values[0]})) {
        $socs{$values[0]} = $line;
    }
}

foreach $tech (keys %techs) {
    if (exists($socs{$tech})) {
        @values = split(/;/,$socs{$tech});

        $property = $org{$values[1]};
        $slas = $sla{$values[6]};

        if ($values[10] eq "Vazio") {
            $buy_date = "Vazio";
        }
        else {
            @date = split(/-/, $values[10]);
            $buy_date = "$date[1]/$date[0]/$date[2] 14:00:00";
        }

        if ($values[11] eq "Vazio") {
            $cont_date = "Vazio";
        }
    }
}

```



```

    else {
        @date = split(/-/, $values[11]);
        $cont_date = "$date[1]/$date[0]/$date[2] 14:00:00";
    }

    $equip = $techs{$tech} .
"$property;$values[5];$slas;$buy_date;$cont_date";
    }
    else {
        $equip = $techs{$tech} . "Vazio;Vazio;Vazio;Vazio;Vazio";
    }

    print "$equip\n";
}

```

## segunda.pl – equipamentos da segunda aplicação

```

#open file with assets, types, manufacturers, models and customers
open(FILE, 'novo.txt');
@lines = <FILE>;
close(FILE);

open(FILE, 'tipos.txt');
@classes = <FILE>;
close(FILE);

open(FILE, 'marcas.txt');
@mans = <FILE>;
close(FILE);

open(FILE, 'modelos.txt');
@models = <FILE>;
close(FILE);

open(FILE, 'clientes.txt');
@clients = <FILE>;
close(FILE);

#remove the end of the line character
chomp foreach(@classes);
chomp foreach(@mans);
chomp foreach(@models);
chomp foreach(@clients);

#convert lists into hashes
$class = @classes;
$man = @mans;
$model = @models;
$client = @clients;

#prints the table fields
print "TABLE Network_Resource\n";
print "del nr_prim_search_key nr_nx_string3 nr_grc_id nr_mfr_id nr_mdl_id
nr_maint_org_id nr_org_id nr_desc nr_raw_id nr_serial_num nr_release nr_license_num
nr_service_type nr_wrty_end_dt nr_wrty_st_dt\n";

#prepare the information to be treated
foreach $line (@lines) {
    chomp $line;
    @values = split(/;/, $line);

    if ($values[40] eq "D") {
        $del = 1;
    }
}

```

```

else {
    $del = 0;
}

$name = uc($values[1]);
$nmgtid = $values[0];
$grc = $class{$values[2]};
$mfr = $man{$values[4]};
$mdl = $model{$values[5]};
$org = $client{$values[7]};
$desc = $values[9];
$ip = $values[17];
$serial = $values[20];
$release = $values[23];

#print assets in the correct format of importation
print "{ \"$del\", \"$name\", \"$nmgtid\", \"$grc\", \"$mfr\", \"$mdl\",
\"$org\", \"$org\", \"$desc\", \"$ip\", \"$serial\", \"$release\",
\"MRY04ENB00800\", \"24X7X2\", \"04/30/2008 18:00:00\", \"05/01/2005 09:00:00\"
}\n";
}

```

## 2. Actualização de inventário

### asset\_import.pl

```

#make backup of tables
system "pdm_extract Network_Resource > c:/AssetImport/backupnr.dat";
system "pdm_extract Computer_Extension > c:/AssetImport/backupcpex.dat";

#run the importation scripts
system("perl -W ./UAM/uam.pl");
system("perl -W ./SMS/sms.pl");

```

### uam.pl

```

#load ODBC module
use Win32::ODBC;

#define hash for map memory and hard disk space
%bytes = (0, "B", 1, "KB", 2, "MB", 3, "GB", 4, "TB");

#Get all the computers from UAM
$DSN = &dsnUAM;
$connection = &openConnection();
$SQL = "SELECT UNITID, NAME FROM UNIT WHERE TYPE = 1";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $units{$dataRow[0]} = $dataRow[1];
}
&closeConnection();

#Check if units is empty
if (!%units) {
    push(@logs, &date . " There are no servers in the database\n");
}

```

```
#Get all the IP from UAM
$connection = &openConnection();
$SQL = "SELECT DISTINCT UNITID, ITVAL, ITPRID FROM INFOTXT WHERE ITNAMEID = 72";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    push(@unitidip, $dataRow[0]);
    push(@ip, $dataRow[1]);
    push(@itpridip, $dataRow[2]);
}
closeConnection();
```

```
#Get all the MAC from UAM
$connection = &openConnection();
$SQL = "SELECT DISTINCT UNITID, ITVAL, ITPRID FROM INFOTXT WHERE ITNAMEID = 91";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    push(@unitidmac, $dataRow[0]);
    push(@mac, $dataRow[1]);
    push(@itpridmac, $dataRow[2]);
}
closeConnection();
```

```
#Get all the OS from UAM
$connection = &openConnection();
$SQL = "SELECT UNITID, ITVAL FROM INFOTXT WHERE ITNAMEID = 123";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    @dataRow = $connection->Data();
    $os{$dataRow[0]} = $dataRow[1];
}
closeConnection();
```

```
#Get all the memory from UAM
$connection = &openConnection();
$SQL = "SELECT UNITID, ITVAL FROM INFODBL WHERE ITNAMEID = 655";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    @dataRow = $connection->Data();
    $i = 0;
    $mem = $dataRow[1];
    if ($dataRow[1] =~ /\+(.+)/) {
        $exp = $1;
        while ($exp >= 4) {
            $mem /= 1024;
            $exp -= 4;
            $i++;
        }
    }
    $mem = int $mem;
    $memory{$dataRow[0]} = "$mem $bytes{$i}";
}
closeConnection();
```

```
#Get all the hard disk space from UAM
$connection = &openConnection();
```

```

$SQL = "SELECT UNITID, ITVAL FROM INFODBL WHERE ITNAMEID = 719";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    @dataRow = $connection->Data();
    $i = 0;
    $dis = $dataRow[1];
    if ($dataRow[1] =~ /\+(.+)/) {
        $exp = $1;
        while ($exp >= 4) {
            $dis /= 1024;
            $exp -= 4;
            $i++;
        }
    }
    $dis = int $dis;
    $disk{$dataRow[0]} = "$dis $bytes{$i}";
}
closeConnection();

#Get all the processor speed from UAM
$connection = &openConnection();
$SQL = "SELECT UNITID, ITVAL FROM INFOLNG WHERE ITNAMEID = 152 AND ITPRNID = 13";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    @dataRow = $connection->Data();
    if (exists($speed{$dataRow[0]}) and $speed{$dataRow[0]} ne $dataRow[1]) {
        push(@logs, &date . " The server $units{$dataRow[0]} has processors with
different speed\n");
    }
    $speed{$dataRow[0]} = $dataRow[1];
    $processor{$dataRow[0]}++;
}
closeConnection();

#Regroup IP and MAC address of same servers
$i = 0;
foreach $pridip (@itpridip) {
    $j = 0;
    foreach $pridmac (@itpridmac) {
        if ($pridip == $pridmac and $unitidip[$i] == $unitidmac[$j]) {
            if (exists($ips{$unitidip[$i]})) {
                $ips{$unitidip[$i]} .= "/-$ip[$i]";
            }
            else {
                $ips{$unitidip[$i]} = "$ip[$i]";
            }
        }

        if (exists($macs{$unitidip[$i]})) {
            $macs{$unitidip[$i]} .= "/-$mac[$j]";
        }
        else {
            $macs{$unitidip[$i]} = "$mac[$j]";
        }
    }
    $j++;
}
++$i;
}

```



```
#Prepares information to be treated
foreach $unit (keys %units) {
    next if (!exists($ips{$unit})); #remove computers without IP (clusters)
    $os{$unit} = "Vazio" if(!exists($os{$unit}));
    $memory{$unit} = "Vazio" if(!exists($memory{$unit}));
    $disk{$unit} = "Vazio" if(!exists($disk{$unit}));
    $speed{$unit} = "Vazio" if(!exists($speed{$unit}));
    $processor{$unit} = "Vazio" if(!exists($processor{$unit}));
    push(@data,
"$units{$unit};$unit;$ips{$unit};$macs{$unit};$os{$unit};$memory{$unit};$disk{$unit}
;$processor{$unit} * $speed{$unit}");
}

```

```
#Get all active servers from USVD
$DSN = &dsnUSVD;
$connection = &openConnection();
$SQL = "SELECT id, nr_prim_search_key, nr_raw_id, nr_mac_addr FROM AHD.net_res where
del = 0 and nr_family = 3";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $server = uc($dataRow[1]);
    if (exists($servers{$server})) {
        $servers{$server} = "REPEATED";
    }
    else {
        $servers{$server} = $dataRow[0];
        $dataRow[2] = "" if(!$dataRow[2]);
        $dataRow[3] = "" if(!$dataRow[3]);
        push(@nrdata, "{ \"$dataRow[0]\", \"$dataRow[2]\", \"$dataRow[3]\" }");
    }
}
&closeConnection();

```

```
#Get all servers data from USVD
$DSN = &dsnUSVD;
$connection = &openConnection();
$SQL = "SELECT id, ext_asset, cpex_operating_sys, cpex_physical_mem,
cpex_hard_disk_sp, cpex_processor_spd FROM AHD.cpex";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $cpex{$dataRow[1]} = $dataRow[0];
    $dataRow[2] = "" if(!$dataRow[2]);
    $dataRow[3] = "" if(!$dataRow[3]);
    $dataRow[4] = "" if(!$dataRow[4]);
    $dataRow[5] = "" if(!$dataRow[5]);
    push(@cpexdata, "{ \"$dataRow[0]\", \"$dataRow[2]\", \"$dataRow[3]\",
\"$dataRow[4]\", \"$dataRow[5]\" }");
}
&closeConnection();

```

```
#Variables that will check if new values need to be inserted
$c = $n = 0;

```

```
#net_res table construction on file nr.dat
open(FILE, '>c:/AssetImport/uam/nr.dat');
print FILE "TABLE Network_Resource\n";

```



```

print FILE "id nr_raw_id nr_mac_addr\n";
foreach $item (@data) {
    @values = split(/;/, $item);
    $name = uc($values[0]);
    if (exists($servers{$name})) {
        if ($servers{$name} eq "REPEATED") {
            push(@logs, &date . " The server $name wasn't updated because he is
repeated\n");
        }
        else {
            $code = "{ \"servers{$name}\", \"$values[2]\", \"$values[3]\" }";
            if (!grep(/\Q$code/, @nrdata)) {
                print FILE "$code\n";
                &assets;
                $n = 1;
            }
            $values[7] = "Vazio" if($values[7] eq "Vazio * Vazio"); #garante que
nao apareca Vazio * Vazio
            if (exists($cpex{$servers{$name}})) {
                push(@cpexs, "{ \"$cpex{$servers{$name}}\", \"$values[4]\",
\"$values[5]\", \"$values[6]\", \"$values[7]\" }");
            }
            else {
                push(@logs, &date . " The server $name isn't insert in CPEX\n");
            }
        }
    }
    else {
        push(@logs, &date . " The server $name doesn't exist in Service Desk\n");
    }
}
close(FILE);

#cpex table construction on file cpex.dat
open(FILE, '>c:/AssetImport/uam/cpex.dat');
print FILE "TABLE Computer_Extension\n";
print FILE "id cpex_operating_system cpex_physical_memory cpex_hard_disk_space
cpex_processor_speed\n";
foreach $cpex (@cpexs) {
    if (!grep(/\Q$cpex/, @cpexdata)) {
        print FILE "$cpex\n";
        &assets_detail;
        $c = 1;
    }
}
close(FILE);

#load asset and asset detail information
if ($n) {
    system "pdm_load -v -f c:/AssetImport/uam/nr.dat > c:/AssetImport/uam/nr.log
2>&1";
    push(@logs, &date . " Asset updated\n");
}
else {
    push(@logs, &date . " No assets were updated\n");
}
if ($c) {
    system "pdm_load -v -f c:/AssetImport/uam/cpex.dat >
c:/AssetImport/uam/cpex.log 2>&1";
    push(@logs, &date . " Asset detail updated\n");
}
else {

```

```

    push(@logs, &date . " No assets details were updated\n");
}

#update asset log file
if (@asset) {
    open(FILE, '>>c:/AssetImport/uam/asset.log');
    foreach $log (@asset) {
        print FILE $log;
    }
    close(FILE);
}

#update asset detail log file
if (@asset_detail) {
    open(FILE, '>>c:/AssetImport/uam/asset_detail.log');
    foreach $log (@asset_detail) {
        print FILE $log;
    }
    close(FILE);
}

#update log file
if (@logs) {
    open(FILE, '>>c:/AssetImport/uam/logs.log');
    foreach $log (@logs) {
        print FILE $log;
    }
    close(FILE);
}

#####

#define DSN to access UAM
sub dsnUAM {
    return "DSN=*****;UID=*****;PWD=*****;";
}

#define DSN to access USVD
sub dsnUSVD {
    return "DSN=*****;UID=*****;PWD=*****;";
}

#open a connection to the current DSN
sub openConnection {
    return new Win32::ODBC($DSN)
    or die "Could not open connection to DSN because of [$!]";
}

#close the connection
sub closeConnection {
    $connection->Close();
}

#run an sql statement
sub runStatement {

```

```

if ($connection->Sql($SQL)) {
    open(FILE, '>>c:/AssetImport/uam/logs.log');
    print FILE &date . " Could not execute the following statement: $SQL\n";
    print FILE &date . " Encountered this error: " . $connection->Error() .
"\n";
    close(FILE);
    $connection->Close();
    die;
}
}

#return the current date
sub date {
    @date = localtime(time);
    $second = $date[0];
    $minute = $date[1];
    $hour = $date[2];
    $day = $date[3];
    $month = $date[4] + 1;
    $year = $date[5] + 1900;
    $second = "0" . $second if($second < 10);
    $minute = "0" . $minute if($minute < 10);
    $hour = "0" . $hour if($hour < 10);
    $day = "0" . $day if($day < 10);
    $month = "0" . $month if($month < 10);
    $year = "0" . $year if($year < 10);
    return "$day-$month-$year $hour:$minute:$second";
}

#track assets in log file
sub assets {
    foreach $data (@nrdata) {
        @vals = split(/\"/,$data);
        next if($vals[1] != $servers{$name});
        @codes = split(/\"/,$code);
        push(@asset, &date . " The server $name has different IP\nLast: $vals[3] -
New: $codes[3]\n") if ($vals[3] ne $codes[3]);
        push(@asset, &date . " The server $name has different MAC address\nLast:
$vals[5] - New: $codes[5]\n") if ($vals[5] ne $codes[5]);
    }
}

#track assets details in log file
sub assets_detail {
    foreach $data (@cpexdata) {
        @vals = split(/\"/,$data);
        @codes = split(/\"/,$cpex);
        next if($vals[1] != $codes[1]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
OS\nLast: $vals[3] - New: $codes[3]\n") if ($vals[3] ne $codes[3]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
memory\nLast: $vals[5] - New: $codes[5]\n") if ($vals[5] ne $codes[5]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
hard disk\nLast: $vals[7] - New: $codes[7]\n") if ($vals[7] ne $codes[7]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
Processor\nLast: $vals[9] - New: $codes[9]\n") if ($vals[9] ne $codes[9]);
    }
}
}

```



```

#load ODBC module
use Win32::ODBC;

#Get all the computers from SMS
$DSN = &dsnSMS;
$connection = &openConnection();
$SQL = "SELECT MachineID, Name0 FROM System_DATA";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $units{$dataRow[0]} = $dataRow[1];
}
&closeConnection();

#Check if units is empty
if (!%units) {
    push(@logs, &date . " There are no servers in the database\n");
}

#Get all the IP from SMS
$connection = &openConnection();
$SQL = "SELECT MachineID, IPAddress0, MACAddress0 FROM Network_DATA";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    next if(!$dataRow[0] or !$dataRow[1] or !$dataRow[2]);
    if (exists($ips{$dataRow[0]})) {
        $ips{$dataRow[0]} .= "/-$dataRow[1]";
        $macs{$dataRow[0]} .= "/-$dataRow[2]";
    }
    else {
        $ips{$dataRow[0]} = $dataRow[1];
        $macs{$dataRow[0]} = $dataRow[2];
    }
}
closeConnection();

#Get all the operating system and memory from SMS
$connection = &openConnection();
$SQL = "SELECT MachineID, Caption0, TotalVisibleMemorySize0 FROM
Operating_System_DATA";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    if ($dataRow[1] =~ /\QMicrosoft(R) Windows(R) Server 2003, Standard Edition/) {
        $dataRow[1] = "Win 2003 SE";
    }
    $os{$dataRow[0]} = $dataRow[1];
    $memory{$dataRow[0]} = "$dataRow[2] MB";
}
&closeConnection();

#Get all the hard disk space from SMS
$connection = &openConnection();
$SQL = "SELECT MachineID, Size0 FROM Disk_DATA";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();

```

```

    if (exists($disk{$dataRow[0]}) {
        $disk{$dataRow[0]} .= "/-$dataRow[1] MB";
    }
    else {
        $disk{$dataRow[0]} = "$dataRow[1] MB";
    }
}
&closeConnection();

#Get all the Processor Speed from SMS
$connection = &openConnection();
$SQL = "SELECT MachineID, CurrentClockSpeed0 FROM Processor_DATA";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    @dataRow = $connection->Data();
    if (exists($speed{$dataRow[0]}) and $speed{$dataRow[0]} ne $dataRow[1]) {
        push(@logs, &date . " The server $units{$dataRow[0]} has processors with
different speed\n");
    }
    else {
        $speed{$dataRow[0]} = $dataRow[1];
    }
    $processor{$dataRow[0]}++;
}
closeConnection();

#Prepares information to be treated
foreach $unit (keys %units) {
    $os{$unit} = "Vazio" if(!exists($os{$unit}));
    $memory{$unit} = "Vazio" if(!exists($memory{$unit}));
    $disk{$unit} = "Vazio" if(!exists($disk{$unit}));
    $speed{$unit} = "Vazio" if(!exists($speed{$unit}));
    $processor{$unit} = "Vazio" if(!exists($processor{$unit}));
    push(@data,
"$units{$unit};$unit;$ips{$unit};$macs{$unit};$os{$unit};$memory{$unit};$disk{$unit};
;$processor{$unit} * $speed{$unit}");
}

#Get all active servers from USVD
$DSN = &dsnUSVD;
$connection = &openConnection();
$SQL = "SELECT id, nr_prim_search_key, nr_raw_id, nr_mac_addr FROM AHD.net_res where
del = 0 and nr_family = 3";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $server = uc($dataRow[1]);
    if (exists($servers{$server})) {
        $servers{$server} = "REPEATED";
    }
    else {
        $servers{$server} = $dataRow[0];
        $dataRow[2] = "" if(!$dataRow[2]);
        $dataRow[3] = "" if(!$dataRow[3]);
        push(@nrdata, "{ \" $dataRow[0]\", \" $dataRow[2]\", \" $dataRow[3]\" }");
    }
}
&closeConnection();

```



```

#Get all servers data from USVD
$connection = &openConnection();
$SQL = "SELECT id, ext_asset, cpex_operating_sys, cpex_physical_mem,
cpex_hard_disk_sp, cpex_processor_spd FROM AHD.cpex";
&runStatement();
while ($connection->FetchRow()) {
    @dataRow = $connection->Data();
    $cpex{$dataRow[1]} = $dataRow[0];
    $dataRow[2] = "" if(!$dataRow[2]);
    $dataRow[3] = "" if(!$dataRow[3]);
    $dataRow[4] = "" if(!$dataRow[4]);
    $dataRow[5] = "" if(!$dataRow[5]);
    push(@cpexdata, "{ \"$dataRow[0]\", \"$dataRow[2]\", \"$dataRow[3]\",
\"$dataRow[4]\", \"$dataRow[5]\" }");
}
&closeConnection();

#Variables that will check if new values need to be inserted
$c = $n = 0;

#net_res table construction on file nr.dat
open(FILE, '>c:/AssetImport/sms/nr.dat');
print FILE "TABLE Network_Resource\n";
print FILE "id nr_raw_id nr_mac_addr\n";
foreach $item (@data) {
    @values = split(/;/, $item);
    $name = uc($values[0]);
    if (exists($servers{$name})) {
        if ($servers{$name} eq "REPEATED") {
            push(@logs, &date . " The server $name wasn't updated because he is
repeated\n");
        }
        else {
            $code = "{ \"$servers{$name}\", \"$values[2]\", \"$values[3]\" }";
            if (!grep(/$code/, @nrdata)) {
                print FILE "$code\n";
                &assets;
                $n = 1;
            }
            $values[7] = "Vazio" if($values[7] eq "Vazio * Vazio"); #garante que
nao apareca Vazio * Vazio
            if (exists($cpex{$servers{$name}})) {
                push(@cpexs, "{ \"$cpex{$servers{$name}}\", \"$values[4]\",
\"$values[5]\", \"$values[6]\", \"$values[7]\" }");
            }
            else {
                push(@logs, &date . " The server $name isn't insert in CPEX\n");
            }
        }
    }
    else {
        push(@logs, &date . " The server $name doesn't exist in Service Desk\n");
    }
}
close(FILE);

#cpex table construction on file cpex.dat
open(FILE, '>c:/AssetImport/sms/cpex.dat');
print FILE "TABLE Computer_Extension\n";

```

```

print FILE "id cpex_operating_system cpex_physical_memory cpex_hard_disk_space
cpex_processor_speed\n";
foreach $cpex (@cpexs) {
    if (!grep(/\Q$cpex/,@cpexdata)) {
        print FILE "$cpex\n";
        &assets_detail;
        $c = 1;
    }
}
close(FILE);

#load asset and asset detail information
if ($n) {
    system "pdm_load -v -f c:/AssetImport/sms/nr.dat > c:/AssetImport/sms/nr.log
2>&1";
    push(@logs, &date . " One ore more assets were updated\n");
}
else {
    push(@logs, &date . " No assets were updated\n");
}
if ($c) {
    system "pdm_load -v -f c:/AssetImport/sms/cpex.dat >
c:/AssetImport/sms/cpex.log 2>&1";
    push(@logs, &date . " One or more assets details were updated\n");
}
else {
    push(@logs, &date . " No assets details were updated\n");
}

#update asset log file
if (@asset) {
    open(FILE, '>>c:/AssetImport/sms/asset.log');
    foreach $log (@asset) {
        print FILE $log;
    }
    close(FILE);
}

#update asset detail log file
if (@asset_detail) {
    open(FILE, '>>c:/AssetImport/sms/asset_detail.log');
    foreach $log (@asset_detail) {
        print FILE $log;
    }
    close(FILE);
}

#update log file
if (@logs) {
    open(FILE, '>>c:/AssetImport/sms/logs.log');
    foreach $log (@logs) {
        print FILE $log;
    }
    close(FILE);
}

```

```
#####
```

```

#define DSN to access SMS
sub dsnSMS {
    return "DSN=*****;UID=*****;PWD=*****;";
}

#define DSN to access USVD
sub dsnUSVD {
    return "DSN=*****;UID=*****;PWD=*****;";
}

#open a connection to the current DSN
sub openConnection {
    return new Win32::ODBC($DSN)
    or die "Could not open connection to DSN because of [$!]";
}

#close the connection
sub closeConnection {
    $connection->Close();
}

#run an sql statement
sub runStatement {
    if ($connection->Sql($SQL)) {
        open(FILE, '>>c:/AssetImport/sms/logs.log');
        print FILE &date . " Could not execute the following statement: $SQL\n";
        print FILE &date . " Encountered this error: " . $connection->Error() .
"\n";
        close(FILE);
        $connection->Close();
        die;
    }
}

#return the current date
sub date {
    @date = localtime(time);
    $second = $date[0];
    $minute = $date[1];
    $hour = $date[2];
    $day = $date[3];
    $month = $date[4] + 1;
    $year = $date[5] + 1900;
    $second = "0" . $second if($second < 10);
    $minute = "0" . $minute if($minute < 10);
    $hour = "0" . $hour if($hour < 10);
    $day = "0" . $day if($day < 10);
    $month = "0" . $month if($month < 10);
    $year = "0" . $year if($year < 10);
    return "$day-$month-$year $hour:$minute:$second";
}

#track assets in log file
sub assets {
    foreach $data (@nrdata) {
        @vals = split(/\//,$data);
    }
}

```



```

        next if($vals[1] != $servers{$name});
        @codes = split(/\"/, $code);
        push(@asset, &date . " The server $name has different IP\nLast: $vals[3] -
New: $codes[3]\n") if ($vals[3] ne $codes[3]);
        push(@asset, &date . " The server $name has different MAC address\nLast:
$vals[5] - New: $codes[5]\n") if ($vals[5] ne $codes[5]);
    }
}

#track assets details in log file
sub assets_detail {
    foreach $data (@cpexdata) {
        @vals = split(/\"/, $data);
        @codes = split(/\"/, $cpex);
        next if($vals[1] != $codes[1]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
OS\nLast: $vals[3] - New: $codes[3]\n") if ($vals[3] ne $codes[3]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
memory\nLast: $vals[5] - New: $codes[5]\n") if ($vals[5] ne $codes[5]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
hard disk\nLast: $vals[7] - New: $codes[7]\n") if ($vals[7] ne $codes[7]);
        push(@asset_detail, &date . " The details for cpex $codes[1] has different
Processor\nLast: $vals[9] - New: $codes[9]\n") if ($vals[9] ne $codes[9]);
    }
}

```

### 3. Gerador do explorador de pastas

#### explorer.pl – lista de ficheiros ou pastas com os relatórios

```

#print html start code
print "<body style=\"BACKGROUND-IMAGE: url(product_banner.jpg); BACKGROUND-REPEAT:
no-repeat\">";
print "<title>SMS Reports</title>";
print "<FONT face=Tahoma size=2>\n<BR><BR><BR>\n<STRONG>SMS Reports</STRONG><BR>";

#define variables
$current = "";
$increment = 6;
$spaces = 0;
&readdir("d:/Stuff/Import/lista/");

#function that generate explorer
sub readdir {
    local($dir) = @_ ;
    chdir($dir) or die "Cannot chdir to $dir\n";
    opendir(DIR, ".");
    local @items = readdir(DIR);
    closedir(DIR);

    foreach $item (@items) {
        next if ($item eq "." or $item eq ".." or $item eq "mainroad_list.html");

        if (-d $item) {
            for ($i = 0; $i < $spaces; $i++) {
                print "&nbsp;";
            }
            #print html code for folder
            print "<IMG SRC=\"rep_templates_folder.gif\"
BORDER=0>&nbsp;<b>$item</b><BR>\n";
            $spaces += $increment;
            $current .= $item."/";
            &readdir($item);
        }
    }
}

```

```

    }
    else {
        if ($item =~ /\.htm/) {
            $file = $item;
            $item =~ s/\.[a-z]*//g;
            for ($i = 0; $i < $spaces; $i++) {
                print "&nbsp;";
            }
            #print html code for file
            print "<IMG SRC=\"rep_template.gif\" BORDER=0>&nbsp;<a
href=\"\$current$file\" Target=\"AMReportOutput\">$item</a><BR>";
        }
    }

    chdir("..");
    $spaces -= $increment;
    $current =~ s/[a-zA-Z]*\///g;
}

#print html end code
print "\n</FONT></body>\n";

```



**ANEXO C: Ficheiros gerados**

Neste anexo encontra-se os conteúdos dos ficheiros gerados pela actualização do inventário. Por uma questão de confidencialidade, os nomes dos computadores, os endereços IP e MAC bem como outras informações, são omitidos.

**update.log – log das actualizações**

```
13-09-2005 23:00:03 Asset updated
13-09-2005 23:00:03 Asset detail updated
14-09-2005 23:00:03 Asset updated
14-09-2005 23:00:03 Asset detail updated
15-09-2005 23:00:03 Asset updated
15-09-2005 23:00:03 Asset detail updated
16-09-2005 23:00:04 Asset updated
16-09-2005 23:00:04 Asset detail updated
17-09-2005 23:00:03 No assets were updated
17-09-2005 23:00:03 No assets details were updated
18-09-2005 23:00:04 No assets were updated
18-09-2005 23:00:04 No assets details were updated
19-09-2005 23:00:03 No assets were updated
19-09-2005 23:00:03 No assets details were updated
20-09-2005 23:00:03 No assets were updated
20-09-2005 23:00:03 No assets details were updated
```

**asset.log – log da informação principal dos equipamentos**

```
13-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
13-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
13-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
13-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
13-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
13-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
14-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
14-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
14-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
14-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
15-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
15-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
16-09-2005 23:00:03 The server ***** has different IP
Last: ***** - New: *****
16-09-2005 23:00:03 The server ***** has different MAC address
Last: ***** - New: *****
```

**detail.log – log da informação detalhada de um equipamento**

```
13-09-2005 23:00:03 The details for server ***** has different OS
Last: ***** - New: *****
```

13-09-2005 23:00:03 The details for server \*\*\*\*\* has different memory  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
13-09-2005 23:00:03 The details for server \*\*\*\*\* has different hard disk  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
13-09-2005 23:00:03 The details for server \*\*\*\*\* has different Processor  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
14-09-2005 23:00:03 The details for server \*\*\*\*\* has different Processor  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
15-09-2005 23:00:03 The details for server \*\*\*\*\* has different OS  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
15-09-2005 23:00:03 The details for server \*\*\*\*\* has different memory  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
15-09-2005 23:00:03 The details for server \*\*\*\*\* has different hard disk  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
15-09-2005 23:00:03 The details for server \*\*\*\*\* has different Processor  
Last: \*\*\*\*\* - New: \*\*\*\*\*  
16-09-2005 23:00:03 The details for server \*\*\*\*\* has different Processor  
Last: \*\*\*\*\* - New: \*\*\*\*\*







FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000081475