Mário João Gonçalves Antunes

Administração centralizada de grupos de sistemas UNIX

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2001

Mário João Gonçalves Antunes

Administração centralizada de grupos de sistemas UNIX

Tese submetida à Faculdade de Ciências da Universidade do Porto para obtenção do grau de Mestre em Informática, área de especialização em Sistemas e Redes

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2001

à Berta José

Agradecimentos

Gostaria de agradecer a algumas pessoas que me apoiaram durante o desenvolvimento deste trabalho e elaboração da respectiva tese de dissertação.

Em primeiro lugar, um agradecimento especial aos orientadores deste trabalho, o Professor Doutor Fernando Silva e o Dr. Manuel Eduardo Correia, pelo apoio e disponibilidade que demonstraram, mantendo o nível de motivação elevado. Agradeço igualmente pela orientação científica e pedagógica que forneceram, a qual constituiu uma grande ajuda durante todas as fases de desenvolvimento do trabalho.

De seguida, agradeço à administração da empresa *Novis Telecomunicações, S.A.* e ao seu Departamento de Planeamento e Desenvolvimento de TI, DPDTI, a disponibilidade concedida para a frequência da componente lectiva deste Mestrado.

Uma palavra de agradecimento a todos os familiares, amigos ou apenas conhecidos que, à sua maneira, demonstraram o seu apoio e incentivo, em especial aos meus pais e irmã pelo apoio que sempre demonstraram.

Por fim, e não menos importante, à Berta José, pelo seu apoio incondicional a todos os projectos pessoais e profissionais onde me envolvo, a que este não foi excepção. Agradeço-lhe pelas várias sugestões e correcções que efectuou ao documento, bem como pelo apoio, paciência e compreensão nas muitas noites e fins de semana despendidos no desenvolvimento deste trabalho.

A todos vós, um imenso obrigado.

Resumo

A composição das redes de computadores UNIX nas organizações é cada vez mais heterogénea, contendo máquinas de várias arquitecturas e de diferentes famílias UNIX, com características de configuração e gestão próprias, dispersas geograficamente por vários edifícios. Desta forma, a execução de uma tarefa em várias máquinas da rede implica que o administrador domine vários sistemas operativos UNIX e conheça profundamente todas as máquinas, tornando o seu trabalho naturalmente mais complexo e repetitivo, podendo conduzir ao aparecimento de erros na configuração.

Uma solução mais eficiente baseia-se na centralização das tarefas de administração e de configuração das máquinas da rede, bem como no seu agrupamento em conjuntos lógicos de acordo com regras definidas pelo administrador, como a localização geográfica ou a funcionalidade. Nesta solução cada grupo tem uma configuração que se associará a todas as máquinas do grupo.

O MetaWebmin é um sistema de administração que implementa a solução descrita. Permite a gestão de grupos de sistemas e entre eles a configuração de recursos da rede numa perspectiva Cliente/Servidor, podendo cada um dos componentes pertencer a grupos diferentes. A gestão da configuração da rede efectua-se pelo preenchimento de formulários e a posterior centralização num repositório implementado numa base de dados MySQL. Essa configuração pode corresponder aos utilizadores, grupos UNIX e recursos da rede, como o NFS e o DNS. A actualização local dos ficheiros de configuração nos sistemas garante-se por pedidos HTTP do MetaWebmin aos servidores Webmin em execução. Estes recebem no pedido os parâmetros necessários à actualização dos ficheiros de configuração envolvidos. A arquitectura do MetaWebmin divide-se em três camadas; a *interface* cliente, o servidor MetaWebmin juntamente com a base de dados e os servidores Webmin em execução nos sistemas.

Para solucionar os problemas de consistência e integridade dos dados entre a configuração central e a local de cada sistema, o MetaWebmin disponibiliza um módulo de análise de consistência e actualização do sistema com a configuração central. No sentido de minimizar situações de inconsistência derivadas de erros nos pedidos do MetaWebmin aos servidores Webmin, existe um módulo de tratamento de falhas e correcção automática de pedidos em erro.

O MetaWebmin foi desenvolvido em *Perl*, é modular, usa protocolos *standard* da *Internet* e é de fácil instalação e utilização.

Abstract

Managing a computing environment consisting of hundreds of systems from multiple vendors and hundreds of users dispersed over several buildings is a complex task. The UNIX networks are heterogeneous, composed by a variety of machines with different UNIX operating systems. Each UNIX operating system has its own characteristics of configuration and management, which means that system administrators' tasks can be very hard and complex. The repetitive tasks executed on the machines may increase configuration errors on the configuration files.

A more efficient solution is to implement centralized administration tasks and system configuration files. The main idea is to group the systems logically according to some rules defined by the system administrator, such as the geographical site or machines activity on the network. With this solution, each group has a network configuration that will be distributed by its activated machines. The network configuration may consist of UNIX users and groups, as well as NFS and DNS resources. This solution allows a request to be made once and executed simultaneously on several systems, according to the central configuration, eliminating the redundancy of the tasks and data duplication.

The thesis concerns the design and implementation of an administration system, MetaWebmin, which centralizes the administration of tasks and configuration of a heterogeneous UNIX network. It supports grouping of entities so that they can be managed as if they were just one entity. The systems and network configuration is managed by filling forms and is centralized in a central data repository implemented on a relational database in MySQL. The MetaWebmin makes HTTP requests to the Webmin servers running on the designated systems, which updates the local configuration files according to the values of the request. The MetaWebmin is a three-tier system administration. Three layers compose its architecture: the system administration interface, the MetaWebmin server together with the central database and the Webmin servers. MetaWebmin has a module, which verifies data integrity and, if necessary, updates the local configuration of a system to the correspondent in the central data repository. To prevent data inconsistency derived from HTTP requests with errors, MetaWebmin detects these fails and allows automatic or manual recovery. MetaWebmin is a modular application, developed in *Perl*, which uses standard Internet protocols and it is easy to use and install.

Índice Geral

Índice de Figuras

Capítulo 1

Introdução

A administração de sistemas UNIX, na sua forma mais geral, baseia-se na gestão e configuração de serviços numa máquina. Esta tarefa requer paciência, compreensão, conhecimento e experiência. À semelhança de um serviço de urgência de um hospital, é necessário, por vezes, intervir em várias frentes, afim de resolver o problema colocado no momento. É importante saber trabalhar com os normalmente limitados recursos disponíveis, ser inventivo e recorrer ao improviso numa situação de crise, assim como conhecer bem a forma como os computadores trabalham nas mais variadas situações. Estes factores implicam que o administrador de sistemas tenha grande capacidade de compreensão das situações com que se vai deparando e de aprendizagem constante de novos temas no âmbito da computação em geral e da administração de sistemas e redes em particular.

Para o desempenho da sua actividade o administrador recorre à edição de ficheiros de configuração dos diversos serviços ou a sistemas de administração fornecidos para o efeito com o sistema operativo. Estas duas formas de administrar poderão ser locais ou através de acesso remoto à máquina. Por máquina entende-se um sistema ligado à rede. Um sistema de administração corresponde a uma aplicação que permite administrar um sistema, ou máquina, da rede.

Os sistemas operativos UNIX têm algumas características de desenvolvimento próprias, nomeadamente aspectos relacionados com o nome e formato de alguns ficheiros de configuração, as directorias onde residem e a sintaxe de alguns comandos. Este facto implica que numa rede heterogénea de sistemas UNIX, o administrador deva ter um vasto conhecimento das diversas famílias de sistemas operativos UNIX e das suas especificidades. O Webmin é um sistema de administração existente para plataformas UNIX que pretende dar resposta a este tipo de problemas. Trata-se de uma aplicação disponível para a gestão e administração da grande maioria de sistemas operativos UNIX, usando os mesmos programas e *interface* gráfica. Todavia, a administração através deste utilitário continua a ser local à máquina.

Actualmente as redes de computadores nas organizações são cada vez maiores e mais heterogéneas, constituídas por vários sistemas de famílias UNIX, o que torna o trabalho do administrador mais complexo. Esses sistemas têm, normalmente, necessidades de configuração de serviços muitos parecidas, de acordo com a sua função ou localização geográfica na rede. Por exemplo, as máquinas de um laboratório numa Universidade partilham os mesmos recursos de NFS, têm a mesma configuração de DNS e poderão ser acedidas pelos mesmos utilizadores. Este facto implica uma repetição de tarefas quando o administrador de sistemas pretende configurar um recurso em várias máquinas da rede.

A tarefa de administrar torna-se ainda mais complexa quando os sistemas estão geograficamente dispersos, em que a presença do administrador nos vários locais se torna inviável. Para resolver este problema identificam-se duas possíveis soluções:

- Descentralizar as tarefas de administração por vários administradores de sistemas, cada um responsável por um conjunto de máquinas da rede.
- Centralizar as tarefas de administração numa unidade funcional da organização, acedendo remotamente a cada sistema e efectuando administração local em cada um.

Todavia, a descentralização das tarefas de administração origina alguns problemas de sincronismo entre os vários administradores. A falta de documentação das acções efectuadas bem como a implementação de práticas mais pessoais de administrar, poderão levar a um desconhecimento da rede por parte de todos e à recuperação de um sistema ou da rede mais morosa e susceptível de erros. Por outro lado, a centralização de tarefas implica que o conhecimento da rede e da sua configuração resida apenas num conjunto restrito de pessoas que têm necessariamente a mesma visão de toda a organização.

Nas duas soluções apresentadas existe um problema comum para o administrador de sistemas; a repetição de tarefas que tem de executar. Por exemplo, a criação de um novo utilizador para acesso a um conjunto de máquinas geograficamente distantes. Em ambos os casos a solução consiste em repetir a operação de criação do utilizador em todas as máquinas. Em nenhuma das soluções apresentadas é possível agrupar logicamente as máquinas e garantir que o pedido de criação do utilizador se efectue, de uma só vez, em todas as máquinas seleccionadas.

No sentido de resolver os problemas apontados anteriormente, uma solução mais eficiente baseia-se em:

1. Centralizar as tarefas e pedidos de administração.
2. Centralizar a configuração das máquinas da rede.
3. Agrupar logicamente as máquinas e serviços da rede, segundo um padrão comum, como a função que desempenham na rede ou a localização geográfica.

Uma arquitectura que inclua estas três características constituirá uma mais valia para o administrador de sistemas e para a organização, na medida em que:

1. Evitará a repetição de tarefas e consequente ocorrência de erros na configuração, devido à **centralização dos pedidos** às máquinas da rede.

2. Permitirá o conhecimento da rede por todos os administradores através da **centralização da sua configuração**.

3. Possibilitará a **organização lógica** da rede em grupos e a gestão simultânea das máquinas que os constituem.

1.1 Motivação

A motivação para este trabalho assenta na análise e desenvolvimento de uma arquitectura que implemente uma solução para os problemas apresentados anteriormente. Dessa forma, descrevem-se os aspectos que estiveram na base da motivação para este trabalho:

1. Centralizar a configuração dos sistemas da rede numa base de dados relacional de um sistema de administração, possibilitando a centralização da configuração da rede num único ponto, evitando-se assim a sua duplicação.

2. Centralizar os pedidos de administração num único ponto, permitindo a execução de um pedido em várias máquinas.

3. Agrupar os sistemas e serviços da rede, abstraindo o administrador de aspectos relacionados com características, como a família UNIX, a localização e funcionalidade dos sistema.

4. Evitar a repetição de tarefas pelo administrador, garantindo a automatização das mesmas e a consequente diminuição do tempo gasto em cada uma. O agrupamento das máquinas permitirá efectuar o mesmo pedido a um conjunto de máquinas e não apenas a uma isoladamente.

5. Permitir a gestão de uma rede de sistemas UNIX heterogénea, abstraindo ao administrador as características específicas de cada sistema operativo UNIX.

1.2 Objectivos

O trabalho desenvolvido visa apresentar um modelo de administração centralizada de uma rede heterogénea de sistemas UNIX. A validação do modelo consiste na apresentação dos módulos de gestão de recursos de rede e serviços de administração, desenvolvidos para o efeito. Os objectivos que se pretendem alcançar com este trabalho são:

- Garantir a gestão dos sistemas independentemente da plataforma UNIX.
- Centralizar num SGBD relacional a informação sobre a configuração da rede.
- Centralizar os pedidos de administração aos sistemas da rede num único ponto de acesso.
- Desenvolver uma arquitectura portável e robusta, implementável em qualquer plataforma UNIX.
- Permitir o desenvolvimento e adição de novos módulos sem alterar a sua arquitectura e núcleo aplicacional.
- Disponibilizar mecanismos de verificação de consistência entre as configurações local e central existentes em cada momento. Este é um aspecto particularmente importante quando se define uma estratégia de centralização, atendendo a que a administração local e directa dos sistemas pode existir e é de difícil controlo.
- Disponibilizar mecanismos de correcção de falhas que permitam recuperar, ou mesmo resolver, situações de erro ocorridas nos pedidos efectuados pelo sistema central de administração às outras máquinas da rede.

Pela extensão do trabalho e áreas interligadas envolvidas, houve necessidade de definir correctamente o âmbito, retirando alguns aspectos de desenvolvimento igualmente importantes. Considera-se fora do âmbito deste trabalho:

- O desenvolvimento de uma *interface* gráfica sofisticada. Os formulários e páginas desenvolvidos são simples e funcionais, não incluindo características sofisticadas de grafismo e estética.
- A gestão de servidores aplicacionais.

1.3 Estrutura da tese

Este texto está estruturado da seguinte forma:

O **Capítulo 2** descreve a administração de sistemas UNIX e redes em geral. Este capítulo aborda alguns conceitos essenciais de administração de sistemas e alguns utilitários existentes, destacando-se o Webmin, o SAT e o GeNUAdmin.

No **Capítulo 3** apresentam-se as características e arquitectura do sistema de administração desenvolvido. Descrevem-se ainda os pressupostos e os módulos desenvolvidos no âmbito deste trabalho.

O **Capítulo 4** descreve em detalhe a implementação dos vários módulos e as opções de desenvolvimento tomadas. Descrevem-se os três principais módulos de ligação do sistema de administração com o exterior; base de dados, sistemas da rede e *interface* do cliente. Neste capítulo descrevem-se ainda os processos de instalação da aplicação e os modelos Entidade-Associação e Relacional.

O **Capítulo 5** descreve os principais processos envolvidos na administração de sistemas, utilizando o sistema de administração desenvolvido. São contempladas algumas das principais acções de administração do dia-a-dia, como por exemplo, a criação de um novo utilizador na rede ou a partilha de recursos NFS.

O **Capítulo 6** apresenta as conclusões ao trabalho desenvolvido e à investigação a ele associado, bem como os principais melhoramentos que poderão constituir desenvolvimentos futuros nesta área.

Capítulo 2

Administração de sistemas UNIX

Neste capítulo apresentam-se os conceitos mais relevantes de administração de sistemas e redes. Descrevem-se ainda alguns dos sistemas de administração actuais que mais se assemelham, em termos de objectivos, ao trabalho proposto.

2.1 Definir administração de sistemas

Por administração de sistemas entende-se o acto de gerir, de forma eficiente, os recursos de um computador. Estes recursos são habitualmente partilhados numa arquitectura multi-utilizador, onde se incluem discos, memória e CPU.

A administração de sistemas não pode ser dissociada da de redes. A forma mais comum de trabalhar com um computador é ligando-o a uma rede local e, normalmente, a uma rede mundial de computadores; a *Internet*. A tarefa do administrador de sistemas é também extensível à componente de gestão e configuração dos serviços de rede disponibilizados aos utilizadores.

Para uniformizar a administração de sistemas e redes deverão definir-se boas e más políticas e práticas. Por exemplo, todas as acções efectuadas no dia-a-dia têm sempre uma justificação e podem ser aproveitadas em situações futuras.

Em [19] defende-se a necessidade de cultivar um modo de pensar em administração de sistemas, que inclua os seguintes princípios básicos:

- Independência ou auto-suficiência.
- Sistematização de tarefas.
- Visão altruísta do sistema, onde os utilizadores são prioritários, primeiro colectivamente e depois individualmente.

- Visão fatalista da ocorrência de erros inevitáveis, afim de alcançar controlo firme sobre o sistema.

Algumas práticas contraproducentes deverão igualmente evitar-se. Por exemplo:

- Acreditar que existe uma resposta para qualquer problema.
- Sentir frustração e aborrecimento quando as coisas não correm do modo esperado.
- Esperar que cada problema tenha sempre um princípio, meio e fim. Alguns, pelo seu carácter crónico, não seguirão esta regra.

Ainda segundo [19] deverão encorajar-se as seguintes práticas:

- Procurar respostas em manuais e *newsgroups*.
- Utilizar mecanismos de detecção controlada de falhas.
- Não menosprezar os utilizadores que afirmam ter um problema no seu computador.
- Documentar as experiências de modo a ter registos da resolução de um determinado problema.
- Assumir a responsabilidade nas acções efectuadas, nomeadamente os acidentes que ocorrem e que deverão ser corrigidos pelo administrador de sistemas.
- Investir na prática da língua inglesa. Esta língua é, por excelência, a que é utilizada na documentação em administração de sistemas. A grande maioria dos artigos, manuais e livros sobre esta temática estão escritos em inglês.

Alguns dos principais desafios da administração de sistemas, segundo [19], consistem em:

- Desenhar a rede, física e logicamente.
- Instalar e configurar as máquinas na rede, que poderão ser posteriormente actualizadas com novas versões de programas e sistema operativo.
- Decidir que serviços utilizar na rede de acordo com as necessidades.
- Planear e implementar mecanismos de segurança adequados.
- Disponibilizar um ambiente de trabalho confortável para os utilizadores.
- Desenvolver formas de detectar e corrigir erros e problemas que possam ocorrer. Para o efeito, poderão definir-se mecanismos de análise de *logs* e activar-se outros de auditoria à actividade do sistema.

- Recolher informação pertinente sobre as temáticas de administração de sistemas e redes e garantir mecanismos de consulta e pesquisa rápidos e eficientes.

Em [30] e [19] encontra-se uma descrição das principais funções desempenhadas pelos administradores de sistemas. Estas funções são sintetizadas nos seguintes pontos:

- **Instalação do sistema.** Inicialização e configuração do sistema e dos periféricos. Configuração inicial do *Kernel*. Instalação de *software* aplicacional e de sistema. Definição de acessos ao sistema.
- **Paragem e arranque do sistema.** Definição de *scripts* de paragem e arranque automático das aplicações.
- **Gestão de discos.** Gestão física e lógica dos discos disponíveis. Planeamento da distribuição dos discos pelas aplicações e utilizadores. Definição de grupos de discos para atribuir às aplicações. Análise e implementação de políticas de *mirroring* e *striping*.
- **Gestão de *filesystem*.** Monitorização do espaço ocupado pelos *filesystems*. Atribuição de espaço e implementação de mecanismos de limpeza dos sistemas de ficheiros mais críticos. Definição e gestão de quotas em disco. Partilha remota de ficheiros. Gestão de sistemas de ficheiros de rede.
- **Gestão das áreas de *swap* e *dump*.** Dimensionamento, implementação e monitorização das áreas de *swap*.
- **Definição de políticas de *backup* e *restore*.** Implementação de políticas de *backup* dos dados. Definição do tipo de suporte magnético a utilizar. *Backups* totais *versus* incrementais. Estratégias de *Backups Offsite*. Definição de *scripts* de *backup*. Verificação de *logs* de *backups*.
- **Políticas de elevada disponibilidade.** Definição de arquitecturas de elevada disponibilidade e de recuperação de falhas. Simulação de emergências.
- **Gestão de impressoras e de filas de impressão.** Definição de filas de impressão associadas às impressoras e gestão da actividade do *spooler*.
- **Gestão de serviços de rede TCP/IP.** Configuração e gestão dos serviços de rede necessários aos utilizadores na execução do seu trabalho, destacando-se, NFS, DNS, *Sendmail*, FTP, Telnet e Rlogin.
- **Gestão de segurança do sistema.** Planeamento da segurança do sistema. Gestão de acessos às aplicações e periféricos. Planeamento da segurança da rede.

Na administração de sistemas deverá ainda considerar-se o senso comum e o bom senso. É importante ouvir conselhos de várias fontes, nomeadamente de outros administradores de sistemas, sobre a forma como analisar os erros numa determinada situação e como os ultrapassar.

2.2 Administração por manipulação de ficheiros

A configuração dos serviços de rede e de gestão de um sistema está em ficheiros de texto. Por exemplo, os dados de um utilizador guardam-se nos ficheiros `/etc/passwd`, `/etc/shadow` e `/etc/group`. Este tipo de administração consiste na edição e manipulação directa dos ficheiros de configuração de um serviço. Esta prática de administração utiliza-se normalmente quando:

- Os utilitários existentes para auxiliar a administração de sistemas não contemplam determinadas operações.
- Alguns serviços são mais rapidamente configuráveis através da edição dos ficheiros de configuração do que através dos utilitários gráficos disponíveis. Por exemplo, a configuração de DNS através da edição do ficheiro `/etc/resolv.conf`.
- Alguns administradores mais experientes preferem a edição directa dos ficheiros de configuração.

2.3 Administração através de utilitários

Felizmente para os administradores em geral, e para os menos experientes em particular, os construtores de sistemas operativos disponibilizam sistemas de administração através de aplicações, habitualmente gráficas, para realizar algumas das tarefas descritas na secção 2.1. Embora permitam a gestão e configuração de sistemas remotos, o objectivo consiste quase sempre em configurar ou administrar apenas um sistema isolado de cada vez.

Não existe um *standard* no desenvolvimento destas aplicações, sendo que, cada construtor desenvolve a sua de acordo com as características específicas de cada sistema operativo UNIX.

Estes utilitários têm a preocupação de automatizar e simplificar as tarefas de administração e gestão do sistema. Quanto à primeira preocupação, torna-se possível devido à definição de subtarefas numa mesma tarefa. Na segunda, o uso de *interfaces* gráficas amigáveis orientadas à tarefa auxiliam o administrador em tarefas que, sem o auxílio destes utilitários se tornariam bastante difíceis de executar.

Das operações disponibilizadas por estes utilitários destacam-se:

- gestão de utilizadores e grupos UNIX;
- configuração de *interfaces* de rede e tabelas de encaminhamento;
- visualização de *logs* do sistema;

- configuração de sequências de *shutdown* e *startup* do sistema;
- gestão de recursos de disco;
- gestão de periféricos, nomeadamente impressoras e *modems*.
- configuração e gestão de serviços de rede, como o NFS, DNS, DHCP, entre outros.

O âmbito de actuação destes utilitários é grande, cobrindo as principais operações de administração e gestão de um sistema. Todavia, existem tarefas que pela sua especificidade são difíceis de automatizar e simplificar. São ainda desenvolvidos por cada construtor utilitários para a configuração de determinados serviços específicos. Por exemplo, o utilitário *netconf* nas distribuições de *Linux* para a configuração da componente de rede do sistema.

A análise e estudo de alguns utilitários permite identificar as seguintes características favoráveis para o administrador:

- Configurar e administrar, normalmente de uma forma gráfica, os serviços de sistema e rede. Actualmente estes utilitários existem para ambientes gráficos e acedidos por um *browser*.
- Permitir a gestão simultânea da configuração de um sistema por vários administradores, garantindo a integridade dos dados através de mecanismos de *locking* aos ficheiros de configuração.
- Garantir controlos de acesso e níveis de segurança na sua utilização.
- Permitir a configuração de um determinado serviço por pessoas que não tenham muita experiência do sistema operativo. As *interfaces* são, normalmente, simples e intuitivas para o administrador, minimizando a ocorrência de erros.
- Auxiliar a configuração de serviços complexos, onde a edição de ficheiros é uma tarefa difícil, morosa e susceptível de erros. Por exemplo, a configuração do serviço de *Sendmail*.

Contudo, estes utilitários de administração apresentam também alguns inconvenientes:

- Não cobrem todas as actividades de administração de um sistema, embora assegurem as mais importantes e de rotina.
- Não existe um *standard* no desenvolvimento destes utilitários. Numa rede heterogénea de sistemas UNIX, o administrador terá, por vezes, que conhecer bem vários utilitários de diferentes plataformas UNIX.

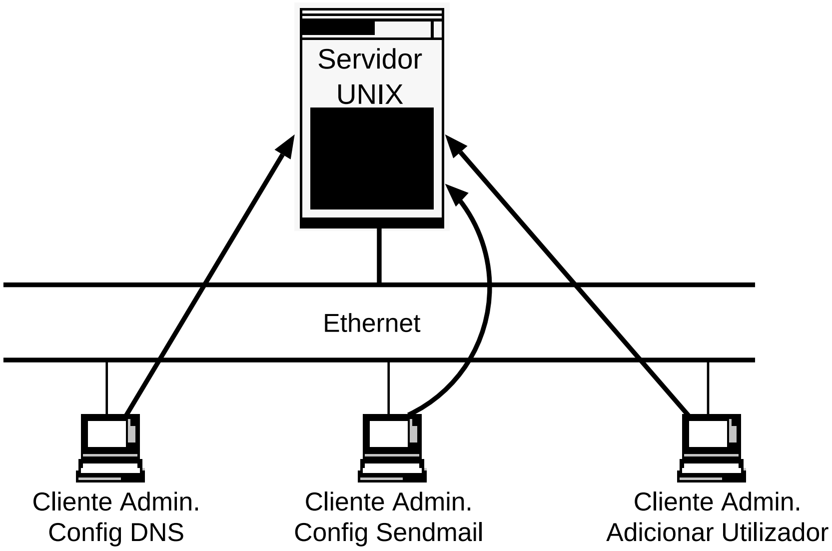

Figura 2.1: Administração de uma máquina usando um sistema de administração

A figura 2.1 ilustra a administração de um sistema UNIX por vários administradores, através de um utilitário de administração, estando cada um daqueles a configurar um serviço.

Estes sistemas de administração existem, normalmente, para ambientes gráficos e não gráficos. Todavia, existem ainda outros utilitários para configuração de alguns serviços pontuais. Um exemplo é a criação de utilizadores através do utilitário `adduser` em sistemas *Linux*.

O alargamento das redes de computadores e a necessidade de efectuar tarefas de gestão, administração e suporte remoto originou o aparecimento da administração remota. Esta forma de administrar consiste em efectuar uma determinada tarefa remotamente num sistema, usando uma infra-estrutura de comunicações. Para tal, o administrador deverá efectuar *login* ao sistema pretendido e ter privilégios para administrar os serviços.

A administração remota efectua-se através de:

- Edição dos ficheiros de configuração dos serviços.
- Utilização de utilitários de administração no sistema remoto.
- Utilização de comandos de execução remota. Por exemplo, através do comando `"rsh sistema1 <comando>"` executa o `comando` em `sistema1`.

Contudo, esta perspectiva de administração permite apenas configurar localmente um sistema de cada vez e de uma forma isolada. Além disso, a utilização de comandos remotos como o `rsh` implica que a rede se baseie em normas de acesso e autenticação bem definidas, caso contrário, poderão suceder situações de quebra de segurança e intrusões a sistemas.

2.4 Sistemas de administração

Actualmente, nas organizações encontramos redes onde existem vários tipos de sistemas operativos UNIX, nomeadamente Linux, AIX, HP-UX, entre outros. A necessidade de sistematizar e centralizar as tarefas de administração levou ao aparecimento e desenvolvimento constante de sistemas de administração que garantam independência relativamente ao sistema operativo UNIX. Todavia, esta necessidade encontra alguns problemas técnicos inerentes às características de implementação do próprio sistema operativo UNIX. Por exemplo:

- Nomenclatura e sintaxe de alguns comandos é diferente. Por exemplo, as opções disponibilizadas pelo comando `find`.
- Directorias onde se encontram os ficheiros de configuração é diferente. Por exemplo, a estrutura da directoria `/etc` onde se encontram os ficheiros de configuração de várias aplicações.
- Formato dos ficheiros de configuração. Por exemplo, a estrutura dos ficheiros `/etc/exports` e `/etc/resolv.conf`.

Quanto à centralização, consideram-se duas abordagens distintas:

- **Centralização de tarefas.** O administrador acede remotamente a vários sistemas, a partir de um ponto central, e executa a configuração de um serviço. Os ficheiros de configuração dos serviços residem localmente em cada sistema.
- **Centralização da configuração.** A configuração de todos os sistemas da rede está centralizada num único ponto de acesso, onde o administrador selecciona os sistemas onde pretende reflectir uma determinada configuração. Os pedidos efectuam-se remotamente a cada sistema, actualizando os seus ficheiros de configuração locais.

Nesta secção destacam-se três sistemas de administração estudados; o Webmin, o SAT e o GeNUAdmin. Relativamente ao Webmin [46], este apenas permite centralizar as tarefas de administração, permitindo o acesso remoto a vários sistemas. Quanto ao SAT [36] e ao GeNUAdmin [20] permitem também a centralização da configuração dos sistemas da rede e dos pedidos a executar em cada um. Os três sistemas de administração apresentados serviram, pelas suas características, de motivação para o trabalho desenvolvido.

2.4.1 Webmin

O Webmin [46] é um sistema de administração que permite a gestão de um sistema UNIX local ou remotamente, sempre de uma forma isolada. Fornece uma *interface* baseada na Web e acessível por um *browser* através do protocolo HTTP [33] [11] [32] [12]. A configuração dos serviços faz-se com base no preenchimento de formulários específicos. A actualização dos ficheiros de configuração efectua-se imediatamente após a submissão do formulário. O processo é rápido e consistente, sendo validados todos os valores inseridos no formulário antes da actualização dos ficheiros de configuração. Este sistema de administração existe actualmente para uma grande variedade de sistemas operativos UNIX. É de fácil instalação, configuração e utilização.

2.4.1.1 Características gerais

O desenvolvimento do Webmin nas plataformas UNIX em que está disponível contempla as seguintes características principais:

- **Abstracção relativamente à plataforma UNIX e serviços de rede.**

  É possível devido ao trabalho de generalização da ferramenta e especialização dos serviços em cada um dos sistemas operativos UNIX. Na prática, utilizam-se rotinas de configuração específicas do Webmin para cada um dos serviços, numa determinada plataforma UNIX.

- **Modularização dos serviços disponibilizados ao administrador de sistemas.**

  Cada módulo corresponde à gestão de um serviço numa determinada plataforma. É possível desenvolver novos módulos de administração e gestão de serviços para uma determinada plataforma UNIX sem alterar o núcleo aplicacional, conforme se pode verificar em [29].

- **Robustez de implementação.**

  Por recurso a uma linguagem de *scripting* portável e robusta, o *Perl*, sendo possível desenvolver os módulos da aplicação independentes da plataforma de sistema operativo UNIX.

- **Licenciamento.**

  Está disponível através de uma licença BSD, ou seja, pode ser distribuído e modificado para aplicações comerciais e não comerciais.

- **Custo.**

  O Webmin é desenvolvido em *Open Source* [4] e está disponível gratuitamente em [46] para vários sistemas operativos UNIX.

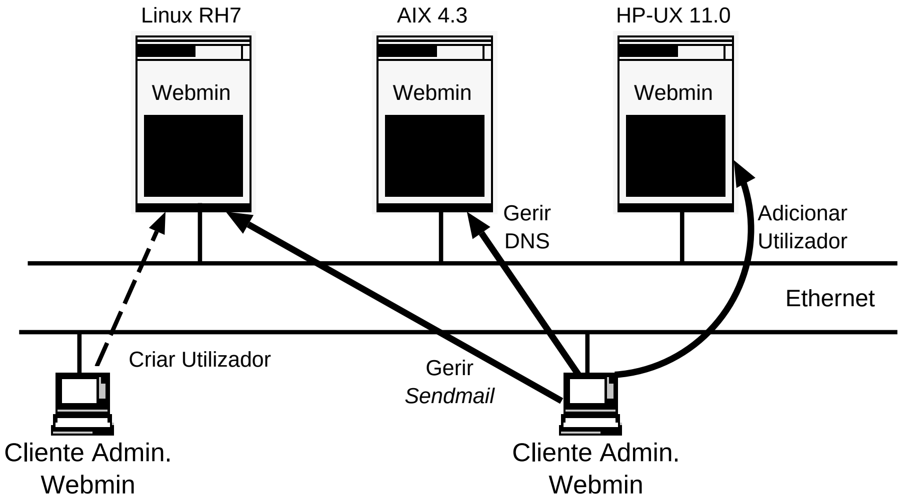

Figura 2.2: Administração usando o Webmin

- **Acesso remoto.**

  Permite a gestão remota de vários servidores, usando a mesma *interface* gráfica, a partir de um ponto de acesso à rede.

- **Segurança.**

  Permite definir vários perfis de administrador de sistemas. Quando configurado correctamente, o Webmin permite utilizar o protocolo SSL [1] [7] para uma comunicação segura em HTTP.

A figura 2.2 ilustra a administração de vários sistemas, na perspectiva do administrador, utilizando o Webmin.

Apesar das vantagens de utilização do Webmin, podem apontar-se os seguintes pontos fracos:

- Não cobre todas as áreas de administração, existindo alguns serviços que não estão contemplados para algumas plataformas UNIX. Por exemplo, a configuração do cliente DNS correspondente ao ficheiro `/etc/resolv.conf`.
- Não permite a centralização da configuração dos sistemas da rede, sendo as alterações efectuadas directamente nos ficheiros de configuração de cada sistema.
- Não suporta a organização dos sistemas em grupos e posterior gestão centralizada das respectivas configurações.

[1] *Secure Socket Layer*

As principais funcionalidades disponíveis resumem-se assim a:

- Gerir localmente o sistema onde se encontra instalado, destacando-se:
  - utilizadores e grupos UNIX;
  - *exports* de NFS;
  - sistemas de ficheiros locais e de rede;
  - serviços de rede e servidores aplicacionais, destacando-se, FTP, DNS, DHCP, *Sendmail* e *Apache Webserver*;
  - componentes de *hardware*, por exemplo, impressoras;
- Gerir a actividade do Webmin, destacando-se:
  - utilizadores e níveis de acesso aos módulos do Webmin;
  - configuração dos módulos;
  - adição de novos módulos entretanto desenvolvidos;
  - adição de novos servidores existentes na rede e com possibilidade de serem geridos remotamente.
  - *logs* de actividade do servidor;
- Disponibilizar, através de um *browser*, o acesso a formulários para gestão dos serviços de administração.
- Processar os dados recebidos do formulário através do módulo associado.
- Enviar resposta ao cliente com o resultado das acções efectuadas.

Alguma documentação sobre os módulos desenvolvidos, bem como a contribuição com novas ideias e desenvolvimentos pode consultar-se em [46]. Em [43] disponibiliza-se um manual de instalação e utilização do Webmin. Em [46] e [29] apresenta-se alguma documentação sobre o desenvolvimento de novos módulos, nomeadamente a estrutura e funções genéricas já desenvolvidas.

2.4.1.2 Organização

O Webmin está organizado em módulos. Cada módulo tem uma estrutura de directorias e ficheiros de configuração idêntica. Supondo que se instala a aplicação em `/usr/local/`, a estrutura de directorias é a seguinte:

- `/usr/local/webmin-083/`
  - `miniserv.pl`
    Servidor Webmin, desenvolvido em *Perl*, responsável pelo processamento dos pedidos dos clientes.

  - `web-lib.pl`
    Funções genéricas em *Perl*, desenvolvidas para o Webmin e utilizadas nos programas.
  - `*.pl`
    Scripts em *Perl* para instalação e configuração do servidor Webmin.
  - `*.cgi`
    Programas que processam os pedidos ao servidor Webmin, através de CGIs desenvolvidas em *Perl*.
  - `mime.types`
    MimeTypes aceites pelo servidor nas sessões HTTP do Webmin.
  - `config-<PlatUNIX>`
    Variáveis de ambiente utilizadas pelo servidor Webmin em cada plataforma UNIX.

- `/etc/webmin/`
  - Directoria para cada um dos serviços geridos pelo servidor. Em cada directoria reside o ficheiro `config` com a definição de variáveis de ambiente utilizadas pelo módulo.
  - *Scripts* de paragem e arranque do servidor.

Cada módulo tem igualmente uma estrutura de ficheiros e directorias bem definida e idêntica. Por exemplo, o módulo de gestão de utilizadores, instalado na directoria `/usr/local/webmin-083/useradmin/`, tem a seguinte estrutura:

- `./help/*`
  Ficheiros de *help* para as opções disponibilizadas no módulo.
- `./images/*`
  Imagens diversas.
- `./lang/*`
  Ficheiros de tradução da *interface* gráfica para as diversas línguas suportadas pelo Webmin.
- `config-<OS>`
  Define, para cada sistema operativo, as características específicas de configuração do serviço.
- `config.info.<LANG>`
  Ficheiros com tradução para as línguas suportadas pelo Webmin, dos valores das variáveis usadas nos programas.

- `cron-lib.pl`

  Diversas funções em *Perl* para a configuração e gestão do serviço.

- `index.cgi`

  Ficheiro correspondente à página principal do módulo.

- `*.cgi`

  Scripts em *Perl* que efectuam as acções pedidas pelo cliente ao servidor Webmin.

Cada *script* em *Perl* do Webmin, que processa uma determinada acção, tem a seguinte estrutura:

1. `require './user-lib.pl'`

   Inclui as funções em *Perl* desenvolvidas para o módulo de gestão de utilizadores.

2. `&ReadParse`

   Executa a função em *Perl* denominada `ReadParse`. Esta função, disponível no ficheiro `web-lib.pl`, analisa os valores incluídos no formulário enviado ao servidor Webmin e efectua a respectiva análise sintáctica (*parsing*).

3. `Código`

   Código em *Perl* que implementa um pedido de administração definido.

2.4.1.3 Processamento dos pedidos

O servidor Webmin inicializa-se pelo *script* `/usr/local/webmin-083/miniserv.pl`. Por defeito, o servidor atende os pedidos no porto TCP 10000, sendo este parâmetro configurável na fase de instalação. O funcionamento do servidor Webmin consiste em:

1. Ler e validar os ficheiros de configuração do servidor, onde são definidos diversos parâmetros gerais, nomeadamente, plataforma de UNIX, língua a utilizar e definição de diversas variáveis de ambiente.

2. Abrir um *socket* principal do servidor e escutar os pedidos no porto TCP 10000.

3. Abrir um *socket* UDP para permitir ligações UDP a outros servidores Webmin. É desta forma que um servidor Webmin acede remotamente a outros que já estejam previamente configurados. No entanto, o protocolo usado entre estes servidores Webmin e o cliente é o HTTP.

4. Inserir em `/var/webmin/miniserv.pid` o número do processo de sistema atribuído [2] e iniciar limpeza de *logs*.

[2] Correspondente ao PID (*Process ID*)

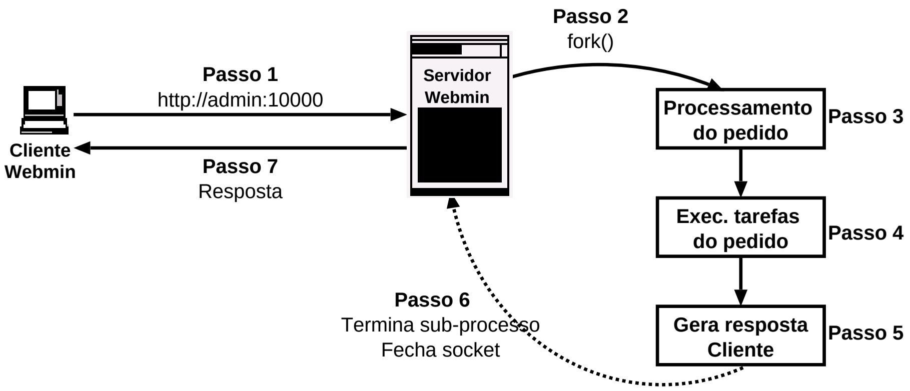

Figura 2.3: Processamento de pedidos no Webmin

5. Iniciar ciclo principal de atendimento de pedidos:
   - Aceitar pedido do cliente.
   - Criar um subprocesso para atender o pedido.
   - Testar a possibilidade de utilização do protocolo SSL.
   - Evocar a função `handle_request` que trata o pedido do cliente e evoca os correspondentes *scripts* CGI.
   - Terminar subprocesso e correspondente *socket*.
   - Terminar o *socket* principal quando o servidor pára.

A figura 2.3 ilustra o processamento de um pedido HTTP ao servidor Webmin. O cliente efectua um pedido HTTP ao servidor Webmin. Este cria um subprocesso para processar as acções relativas ao pedido efectuado e gerar a página de resposta ao cliente. Após concluídas as acções associadas ao pedido, o servidor termina o subprocesso e envia a página de resposta ao cliente que efectuou o pedido.

A figura 2.4 ilustra a gestão remota efectuada pelo Webmin. Na figura, o servidor A tem configurado o acesso remoto aos servidores B e C. O administrador de sistemas estabelece uma ligação ao servidor A e posteriormente gere os outros dois, utilizando os seus servidores Webmin. Os servidores B e C podem ser geridos independentemente deste método, podendo o administrador aceder directamente ao servidor Webmin de cada um deles.

2.4.2 SAT

O SAT, *System Administration Tool*, [36], é um sistema de administração que utiliza uma base de dados distribuída onde guarda a informação sobre as máquinas da rede e utilizadores. A base de dados disponibiliza processos de réplica de dados, controlo de

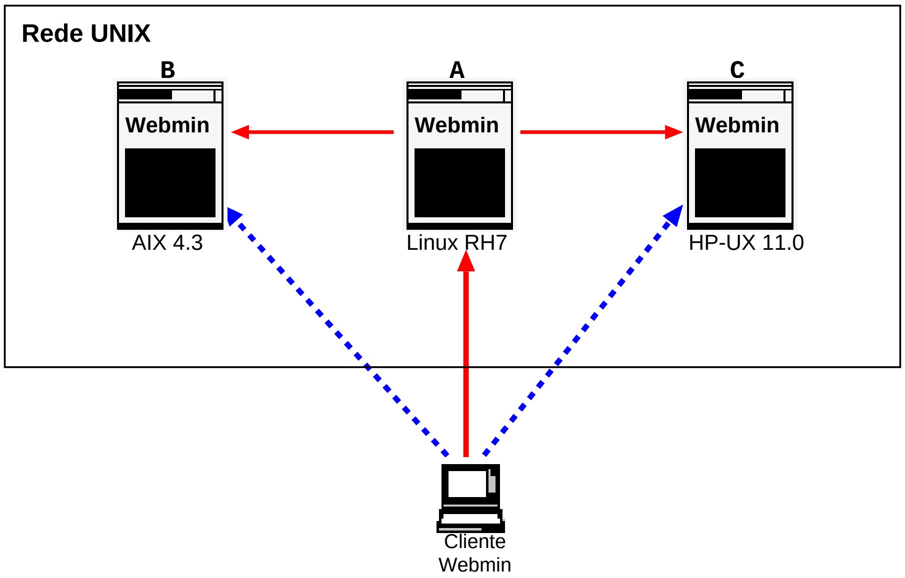

Figura 2.4: Administração de uma rede UNIX usando o Webmin

acessos e mecanismos de *locking*. Pode consultar-se directamente através de programas do SAT, que podem ser *scripts* escritos em SAL [3] ou *Perl*.

Sempre que o administrador altera dados na base de dados, são executados programas em cada sistema da rede que actualizam os ficheiros de configuração, baseados em:

- Informação inserida no SAT.
- Arquitectura UNIX de cada sistema.
- Requisitos específicos de cada sistema.

A informação centralizada no SAT, relativa aos ficheiros de configuração de cada sistema, pode ser de dois tipos:

1. **Específica da máquina**, relativa a configurações específicas de cada sistema, por exemplo, os utilizadores autorizados.
2. **Específica do sistema operativo**, definindo as características de configuração de um serviço num determinado sistema operativo UNIX. Por exemplo, a configuração do ficheiro de *exports* de NFS nas várias plataformas UNIX envolvidas.

A base de dados do SAT é constituída por vários ficheiros de texto em disco, correspondentes às relações existentes. As relações podem ser **estáticas** ou **dinâmicas**.

[3] System Administration Language, cujo interpretador se disponibiliza com o SAT. Esta linguagem combina as características de DCL (VMS), *sh* de UNIX e linguagens de consulta em bases de dados relacionais.

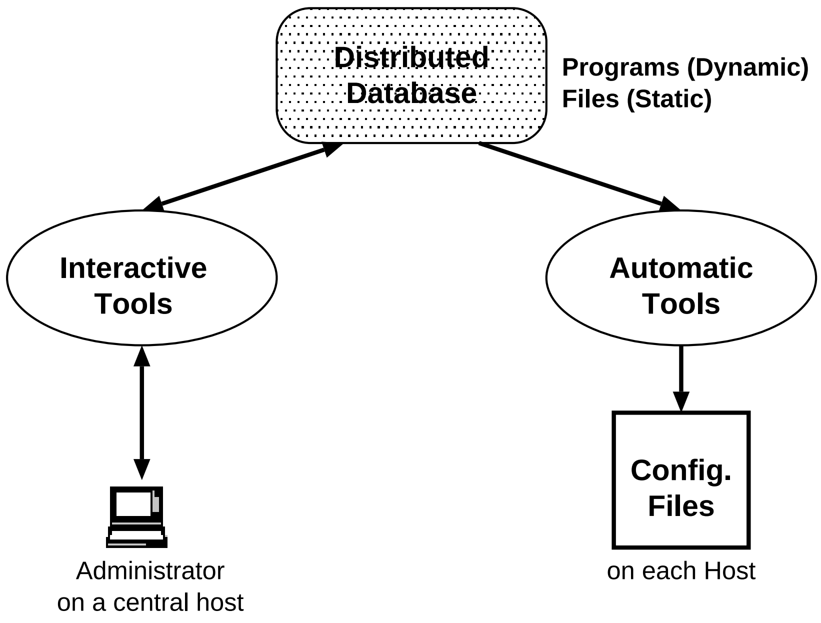

Figura 2.5: Arquitectura do SAT

As primeiras referem-se à informação de cada sistema, gerida directamente pelo administrador, enquanto as segundas são calculadas dinamicamente, através de dados resultantes do *output* de um determinado comando. Por exemplo, alguma informação do *output* do comando `ps -ef` poderá, de acordo com o critério do administrador, ser inserida dinamicamente numa tabela do SAT com a actividade do sistema.

Após a alteração de uma relação estática no SAT efectuam-se as seguintes acções:

1. Distribuição por todos os sistemas, através do comando `rsh`, da tabela correspondente à relação.

2. Execução, em cada sistema, de um programa que constrói adequadamente os ficheiros de configuração associados à relação alterada, tendo em conta aspectos relativos à família UNIX e identificação do sistema.

A figura 2.5 [36] ilustra a arquitectura do SAT e os mecanismos utilizados para a configuração de um serviço num sistema.

A implementação do SAT reforça o cuidado relativamente à integridade dos dados no sistema central, a sua distribuição pelos sistemas da rede e mecanismos de *locking*. É um utilitário modular, podendo adicionar-se mais funcionalidades sem alterar o seu núcleo. Em [47] descreve-se a migração do SAT para *Perl*, reforçando a independência do utilitário relativamente à plataforma UNIX utilizada. Alguma informação mais detalhada sobre o SAT poderá consultar-se em [34] e [35].

2.4.3 GeNUAdmin

O GeNUAdmin [20] [27] é um sistema de administração centralizada que permite controlar a configuração e administração de máquinas UNIX de diferentes plataformas a partir de um ponto central de gestão. O GeNUAdmin guarda a informação relevante de cada sistema da rede num repositório central, sob a forma de ficheiros de texto. Os ficheiros de configuração e métodos de instalação, designados de *targets* na arquitectura do GeNUAdmin, são gerados para várias plataformas UNIX. Antes de gerar os ficheiros de configuração ou efectuar uma actualização dos sistemas, efectuam-se diversos testes de consistência dos dados, garantindo-se assim consistência entre o sistema central e a configuração local daqueles. O sistema está escrito em *Perl* e constrói dinamicamente *scripts* em *Bourne Shell* para auxiliar as tarefas de configuração remota. Os pedidos de administração remotos correspondem a comandos `rsh` às máquinas da rede.

A actualização da configuração de um sistema através do GeNUAdmin engloba as seguintes etapas:

1. Construir a configuração do sistema (ficheiros de *passwords*, impressoras, entre outros) no ambiente do GeNUAdmin.
2. Consultar a base de dados e analisar a consistência dos dados existentes com os relativos à configuração local do sistema.
3. Criar a configuração do sistema numa directoria local, sem modificar a configuração actual.
4. Instalar os *targets*:
    - seleccionar os *targets* e os sistemas a instalar;
    - efectuar uma instalação completa de todos os *targets* em todos os sistemas.

Estas etapas podem executar-se separadamente. No entanto, normalmente são efectuadas de uma só vez, associadas a um comando na aplicação GeNUAdmin.

Os dados guardados na base de dados do GeNUAdmin correspondem a:

hosts.db Máquinas existentes na rede.

disks.db *Filesystems* de rede.

user.db Utilizadores da rede.

printer.db Impressoras disponíveis.

links.db Lista de *soft links* existentes no sistema.

net.db Descrição da topologia da rede.

Algumas das funcionalidades disponibilizadas pelo GeNUAdmin são:

- **Gestão de utilizadores.** Disponibiliza módulos que permitem a manipulação dos dados relativos aos utilizadores e gestão de níveis de acesso. Garante ainda a consistência dos dados inseridos na base de dados.

- **Configuração dos *targets* nos sistemas da rede.** A configuração de um determinado serviço num sistema implica activar o respectivo *target*. Alguns dos *targets* definidos são:

  Passwd

  Etchosts

  Dotrhosts

  Exports

A definição da configuração de um determinado serviço pode ser dependente ou independente do sistema operativo. A implementação da primeira efectua-se no núcleo principal do GeNUAdmin. A segunda representa-se em ficheiros de configuração, para cada tipo de sistema operativo e *target* a configurar. Estes ficheiros são carregados, desde que existam, na execução de um qualquer módulo de configuração. Se não estiver definido um método para um determinado sistema operativo, todos os *targets* nessas máquinas são ignorados. É ainda possível definir um método por defeito para esses casos.

Este sistema de administração é disponibilizado gratuitamente, estando o seu desenvolvimento a cargo da empresa GeNUA mbH [4] [27], integrado num conjunto de soluções de administração de sistemas e redes.

2.4.4 Outros

O LDAP, *Lightweight Directory Access Protocol* [2] [22], é um protocolo que permite a pesquisa e localização de recursos numa rede de computadores, que poderá ser uma *Intranet* ou a *Internet*. Corresponde a uma versão *lightweight* do protocolo DAP, *Directory Access Protocol*, parte do X500 que constitui um *standard* de serviços de directoria numa rede. A "leveza" do LDAP corresponde ao pouco código envolvido nas primeiras versões, uma vez que não incluíam funcionalidades de segurança e controlo de acessos. Alguns serviços, como o DNS, *Domain Name Service*, que disponibiliza um sistema de directoria de nomes de máquinas e endereços IP, podem configurar-se através deste protocolo. A informação relacionada com utilizadores pode igualmente aceder-se centralmente utilizando o LDAP.

[4] *Gesellschaft für Netzwerk - und UNIX-Administration.*

Inicialmente designado de *Yellow Pages*, o NIS, *Network Information Services* [9], é um serviço Cliente/Servidor que permite gerir centralmente vários ficheiros de configuração, como o `/etc/passwd` ou `/etc/shadow`. O NIS disponibiliza mecanismos de cópia dos ficheiros alterados centralmente para os clientes do serviço. Existem alguns procedimentos que poderão ser disponibilizados centralmente pelo servidor de NIS. Por exemplo, a autenticação de um utilizador num sistema. O utilizador efectua *login* e autentica-se num servidor NIS da rede. Se o seu acesso é autorizado, o serviço *automount* efectua o *mount*, através de NFS, da sua directoria de *login* e concede acesso ao sistema.

No âmbito dos sistemas de administração comerciais destacam-se o SMIT, *System Management Interface Tool* [24] [25], para o sistema operativo AIX da IBM e o SAM, *System Administration Manager* [30], da HP, para o HP-UX. São sistemas de administração proprietários de cada um dos sistemas operativos dos construtores IBM e HP, os quais implementam as suas características específicas.

A arquitectura SP [26] da IBM é escalável, constituída por vários nós, ligados em *cluster* através de uma rede interna de alta velocidade, partilhando recursos e efectuando processamento paralelo. Os nós do SP têm instalado o sistema operativo AIX e são geridos centralmente através de um sistema de administração centralizado proprietário da IBM; o PSSP, *Parallel System Support Programs* [23]. Esta camada de *software* disponibiliza aplicações gráficas de instalação, operação, gestão e administração centralizada dos nós do SP. Através da já referida aplicação é ainda possível evocar programas em paralelo nos vários nós do *cluster* SP.

Outros sistemas de administração centralizada são o *Igor* [31], *Athena* [21], *Gash* [10], *SaTool* [44] e o *Environment Molecular Sciences Laboratory - Unix Host Administration - EMSL-UHA* [45]. As funcionalidades destes sistemas variam relativamente aos analisados. Alguns dão enfoque a um determinado tipo de funcionalidades (e.g. gestão de utilizadores) descuidando outras consideradas importantes na gestão de sistemas e redes, como a gestão de recursos de NFS.

2.5 Sumário

O estudo dos sistemas de administração permite concluir que, de uma forma geral, existem características comuns e desejáveis no âmbito da administração de sistemas, destacando-se:

- Independência relativamente à plataforma UNIX.

- Centralização da informação de configuração dos sistemas da rede, nomeadamente, gestão de utilizadores e respectivos grupos UNIX, bem como de alguns recursos de rede, como sejam, impressoras e sistemas de ficheiros da rede.

- Desenvolvimento modular permitindo adicionar novos módulos sem alterar o núcleo da aplicação.
- Mecanismos que garantam a integridade entre a informação existente no repositório central e a configuração local de cada sistema.
- Diferentes níveis de acesso à gestão do servidor central de administração.

O tipo de repositório central utilizado varia entre a utilização de uma base de dados através de ficheiros de texto e a de um SGBD [5], normalmente relacional. O processo de difusão das operações de administração entre o sistema central e os restantes sistemas da rede é igualmente diferente, variando entre a utilização de:

- Comandos remotos do tipo `rsh`, como no GeNUAdmin e SAT.
- Protocolo HTTP, no caso do Webmin.

Todavia, nenhum dos sistemas de administração estudados permite a definição de grupos de sistemas, associando-lhes uma configuração de serviços da rede. Este conceito aumentará a flexibilidade do administrador da rede, permitindo a actualização de vários sistemas em simultâneo de acordo com o grupo a que pertencem, evitando tarefas repetitivas e a consequente ocorrência de erros.

Dessa forma, um modelo de administração mais completo consistirá em considerar as características analisadas anteriormente nos sistemas de administração estudados e incluir ainda a possibilidade de:

- Agrupar as máquinas em grupos de acordo com regras estabelecidas pelo administrador de sistemas, o que permitirá que tenham todas a mesma configuração. Dessas regras poderá constar a funcionalidade ou localização física das máquinas da rede.
- Possibilitar a gestão de recursos inter-grupais, onde as componentes cliente e servidor pertençam a grupos de sistemas distintos.
- Integrar o maior número de tarefas numa mesma operação de administração, eliminando passos intermédios e repetitivos.

[5] Sistema Gestor de Base de Dados

Capítulo 3

MetaWebmin

O MetaWebmin é um sistema de administração que centraliza as tarefas de administração e a configuração das máquinas UNIX existentes na rede. As alterações aos ficheiros de configuração locais efectuam-se através do servidor Webmin instalado nos sistemas da rede. O repositório central de dados é uma base de dados em MySQL [38], acedido e gerido pelo MetaWebmin. Esses mesmos dados referem-se aos sistemas da rede e respectivos grupos, utilizadores e respectivos grupos UNIX e configuração de serviços de DNS e NFS. Baseia-se em protocolos *standard* da *Internet*, acessível através de um *browser*, onde é possível, por exemplo, consultar a base de dados ou preencher os formulários para a introdução dos dados relativos à configuração dos serviços de rede. É constituído por diversos *scripts* em *Perl*, usando o método CGI[1] [16] [13] para processamento dos pedidos entre o MetaWebmin e os servidores Webmin.

Este capítulo apresenta as características do MetaWebmin, a sua arquitectura, os pressupostos considerados no desenvolvimento e uma breve descrição dos módulos desenvolvidos.

3.1 Características

Dos sistemas de administração analisados no capítulo anterior identificaram-se os pontos fortes, as características mais distintivas e o respectivo modo de implementação, a considerar no desenvolvimento do MetaWebmin. Assim sendo, destacam-se as seguintes características do MetaWebmin:

- **Centraliza a configuração** dos serviços de rede, nomeadamente utilizadores, grupos UNIX, recursos de DNS e NFS. Estes dados são armazenados numa base de dados gerida pelo MetaWebmin.

[1] *Common Gateway Interface*

- **Centraliza os pedidos** de administração aos sistemas da rede no servidor MetaWebmin.

- **Gere grupos de sistemas** constituídos por um ou mais sistemas. A cada grupo associam-se utilizadores e grupos UNIX, bem como uma configuração de recursos de DNS e NFS. Os sistemas pertencentes a cada grupo de sistemas são actualizados com a configuração do grupo a que pertencem. Todas as alterações à configuração de um grupo, por exemplo, a adição de um novo utilizador, implica a alteração imediata nos sistemas desse grupo.

- **Configura e gere recursos de rede inter-grupais**, numa lógica de arquitectura Cliente/Servidor. Na configuração de recursos de rede Cliente/Servidor, por exemplo o NFS, é possível incluir as componentes cliente e servidor em grupos de sistemas diferentes.

- **Garante a consistência** entre os dados armazenados na base de dados do MetaWebmin e os ficheiros de configuração dos sistemas da rede. A situação ideal consiste em que todas as operações de administração se efectuem a partir do MetaWebmin. No entanto, qualquer utilizador com privilégios pode aceder a um sistema gerido pelo MetaWebmin e, por exemplo, adicionar directamente um novo utilizador ou alterar o ficheiro `/etc/resolv.conf`. Esta situação originará inconsistências entre a configuração local do sistema e a central existente no MetaWebmin. Para solucionar este problema o MetaWebmin disponibiliza mecanismos de análise de inconsistências e posterior actualização da configuração local com a existente na sua base de dados.

- **Diminui a probabilidade de erros e a repetição de tarefas de administração**. Para tal, o MetaWebmin permite o agrupamento de várias tarefas numa mesma acção, valida a sua consistência e executa-as simultaneamente em vários sistemas. Numa rede heterogénea de média ou grande dimensão esta característica pode significar uma grande economia de tempo e diminuição de erros.

- **Modular**, possibilitando a adição de novos módulos do MetaWebmin sem alterar a estrutura da aplicação.

- **Utiliza protocolos de comunicação abertos e *standard***, à semelhança do Webmin que utiliza protocolos *standards* da *Internet* como o HTTP.

- **Garante a independência da arquitectura** do MetaWebmin relativamente à plataforma UNIX. A utilização da linguagem *Perl* permite, devido à sua portabilidade, a instalação do MetaWebmin em qualquer plataforma UNIX. Quanto à independência da gestão dos sistemas relativamente à plataforma UNIX, garante-se pelos servidores Webmin neles instalado. Havendo independência relativamente à plataforma UNIX do servidor central e dos sistemas da rede, a implementação de uma arquitectura de gestão centralizada de uma rede heterogénea de sistemas UNIX fica mais simplificada.

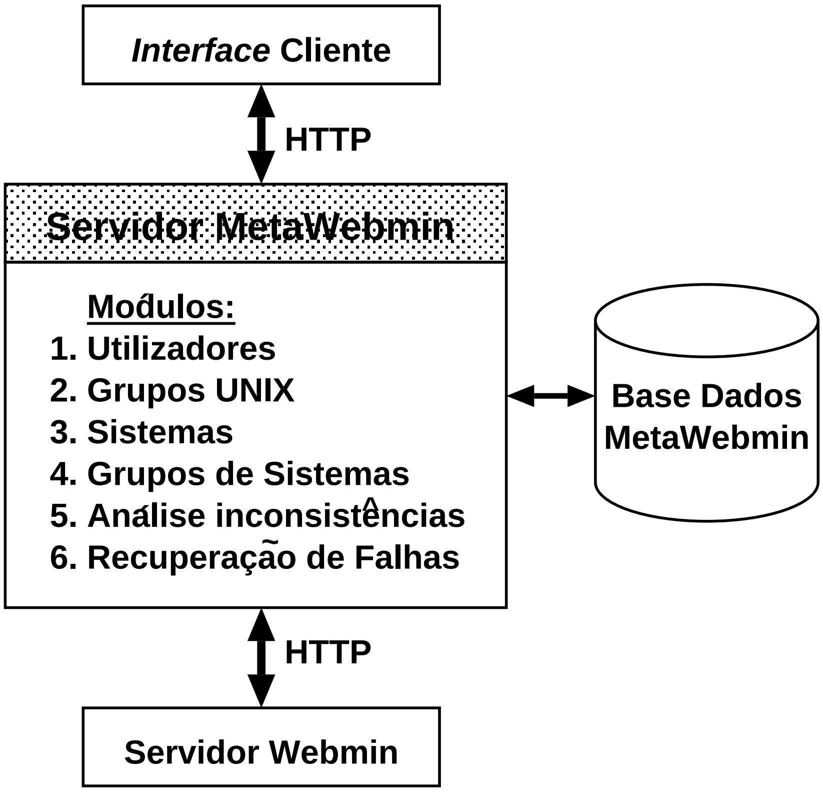

Figura 3.1: Arquitectura do MetaWebmin

- **Permite a gestão de *Backoffice***, utilizando uma *interface* de gestão disponibilizada pelo SGBD do MySQL.

3.2 Arquitectura

A arquitectura do MetaWebmin apresenta-se em três camadas, conforme ilustra a figura 3.1. O servidor MetaWebmin e a base de dados formam a camada central que comunica com a *interface* cliente e os servidores Webmin.

Na arquitectura do MetaWebmin identificam-se os seguintes componentes:

Servidor MetaWebmin. Constitui o elemento centralizador da actividade de administração da rede. Mantém activo um servidor HTTP que disponibiliza uma *interface* de administração baseada em formulários, processa os dados recebidos e, sob a forma de CGI, executa as alterações pedidas nos sistemas envolvidos. Esta componente está estruturada por módulos, dos quais se destacam:

- Gestão de utilizadores e grupos UNIX
- Gestão de sistemas

- Gestão de grupos de sistemas
- Gestão de recursos NFS e DNS
- Analisador de consistência
- Recuperador de falhas

***Interface* Cliente** Constitui o ponto de acesso, via *browser*, do administrador ao MetaWebmin e, consequentemente, à gestão da rede.

Base de Dados Armazena os dados inseridos nos formulários e disponibiliza mecanismos de consulta. Acede-se através dos programas do MetaWebmin ou por uma *interface* de *backoffice* disponibilizada pelo SGBD.

Servidores Webmin Existem em execução nos sistemas geridos pelo MetaWebmin e garantem a alteração local dos ficheiros de configuração, conforme os pedidos de administração efectuados.

A *interface* cliente do MetaWebmin é o único ponto de administração da rede, possibilitando ao administrador a gestão dos dados existentes na base de dados e a evocação dos pedidos de administração aos diversos sistemas.

A figura 3.2 ilustra a arquitectura do MetaWebmin na gestão de sistemas UNIX organizados em grupos. Através da gestão de perfis de administração e utilização de mecanismos de *locking* será possível ter vários administradores a gerir simultaneamente a rede através do MetaWebmin.

Analisando a figura 3.2 identificam-se dois tipos de acções evocadas ao MetaWebmin:

1. Associadas a um grupo de sistemas. No exemplo da figura, a criação do utilizador ''`abc`'' no grupo de sistemas `LAB1_UNIX`, constituído por três sistemas UNIX.
2. Associadas a um sistema UNIX em particular. Por exemplo, quando o administrador pretende exportar a directoria `/home` do sistema `G1_2`.

A figura 3.3 ilustra a sequência de passos para a criação de um utilizador no grupo de sistemas `LAB1_UNIX`, apresentados de seguida:

1. Administrador efectua ligação ao servidor MetaWebmin. Preenche o formulário com os dados do utilizador.
2. MetaWebmin valida os parâmetros constantes do pedido.
3. MetaWebmin actualiza a base de dados.
4. MetaWebmin identifica sistemas associados ao grupo `LAB1_UNIX`.
5. MetaWebmin efectua pedido HTTP aos sistemas do grupo `LAB1_UNIX`.

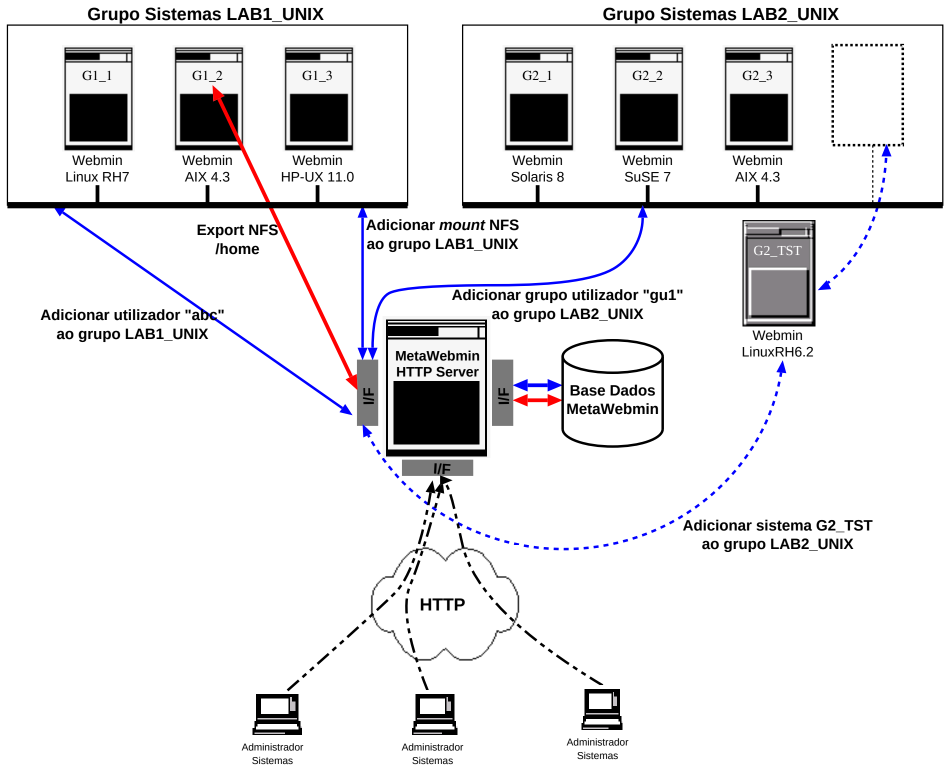

Figura 3.2: Administração usando o MetaWebmin

6. Servidores Webmin efectuam as acções associadas à criação do utilizador.
7. Servidores Webmin enviam resposta sobre o sucesso da operação ao servidor MetaWebmin.
8. MetaWebmin analisa o resultado das respostas ao pedido e insere o evento correspondente na base de dados.
9. MetaWebmin notifica o administrador do resultado da operação.

3.3 Pressupostos

Na arquitectura do MetaWebmin consideraram-se alguns pressupostos ao nível dos seus componentes. Nos sistemas geridos pelo MetaWebmin deverá garantir-se que:

1. O sistema operativo UNIX, na versão suportada pelo Webmin, está devidamente instalado.
2. A *interface* de ligação à rede está devidamente configurada.

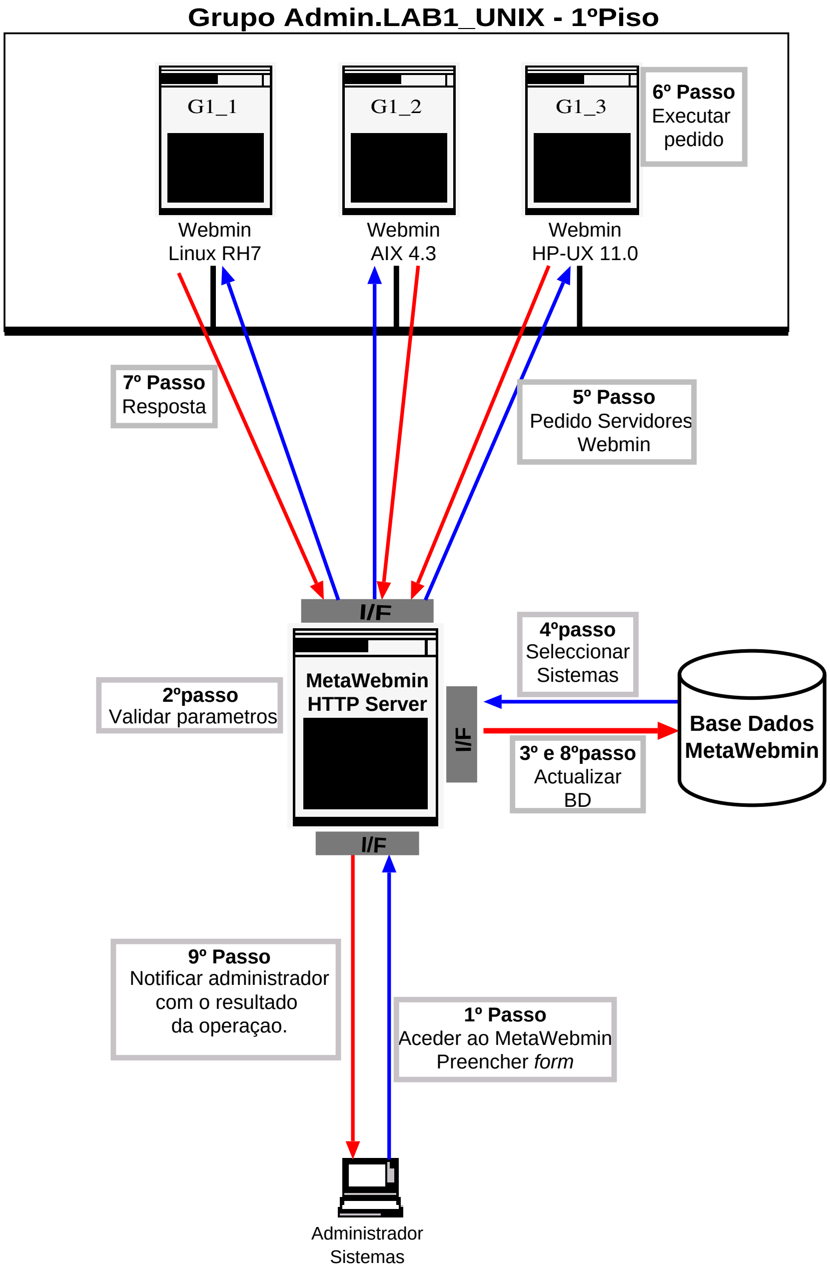

Figura 3.3: Processamento de um pedido no MetaWebmin

3. O servidor Webmin está devidamente instalado e configurado, na sua versão 0.83, para o sistema operativo e versão em causa.
4. A administração da rede é centralizada e efectuada, preferencialmente, a partir do servidor MetaWebmin.

No sistema onde será instalado o MetaWebmin pressupõe-se ainda que:

- Todos os componentes necessários ao funcionamento do MetaWebmin estão instalados, como sejam:
  - SGBD MySql.
  - *Software* do MetaWebmin, incluindo os programas e o servidor HTTP.
  - Módulos de *Perl* utilizados nos programas do MetaWebmin, descritos em pormenor na secção 4.2 do capítulo seguinte.
- O sistema está devidamente ligado e configurado na rede.

Os sistemas Webmin e MetaWebmin baseiam-se no preenchimento de formulários e manipulação de tabelas em HTML. A sua comunicação efectua-se através de pedidos HTTP, cuja estrutura varia entre cada versão do servidor Webmin. Uma vez que o MetaWebmin foi desenvolvido com base na estrutura da versão 0.83 do Webmin, será apenas e só esta que deverá ser instalada nos sistemas da rede. A adaptação a novas versões implica alterar no MetaWebmin os pedidos HTTP aos servidores Webmin, bem como os procedimentos de análise das tabelas do Webmin, nomeadamente utilizadores e grupos UNIX.

3.4 Módulos desenvolvidos

O MetaWebmin está organizado em módulos. Cada módulo disponibiliza operações de manipulação da base de dados e execução dos pedidos nos sistemas da rede. A gestão da base de dados do MetaWebmin efectua-se através de formulários acedidos por um *browser*, cujos campos reflectem a informação necessária para a actualização da configuração local dos sistemas da rede. Nesta secção descrevem-se os módulos desenvolvidos para o MetaWebmin.

3.4.1 Gestão de utilizadores e grupos UNIX

Este módulo é responsável pela gestão de utilizadores e grupos UNIX. Um utilizador tem sempre um grupo UNIX primário e pode ter zero ou mais grupos UNIX secundários. Cada grupo UNIX tem um nome e um GID[2] e associa-se a um intervalo de

[2] Número identificativo do Grupo UNIX.

UIDs[3] disponíveis, a atribuir a cada utilizador que o escolha como grupo primário. A definição dos intervalos de UIDs para o grupo é definida pelo administrador de sistemas aquando da sua criação.

3.4.2 Gestão de sistemas

Um sistema identifica-se pelo seu nome na rede TCP/IP e pertence a um domínio dessa mesma rede. Por exemplo, o sistema `lab1` no domínio `ncc.up.pt` identifica-se por `lab1.ncc.up.pt`. Outra informação guardada na base de dados refere-se ao servidor Webmin, nomeadamente o porto TCP/IP que lhe está associado e o *username* e *password* de administração. As acções associadas a um sistema são:

- **Activar**

  Esta acção associa o sistema a um grupo de sistemas, efectuando os pedidos de administração ao seu servidor Webmin, de acordo com a configuração existente na base de dados para esse grupo.

- **Desactivar**

  Esta opção elimina a associação do sistema ao grupo a que pertence, mantendo os seus dados na base de dados. Se um sistema, por motivo de avaria, é retirado da rede, há necessidade de o desactivar temporariamente do grupo a que pertence, activando-o quando estiver operacional. Um sistema apenas pode pertencer, num determinado momento, a um grupo de sistemas. Para transferir um sistema de um grupo para outro, aquele deverá desactivar-se no grupo original e activar-se no novo.

- **Actualizar**

  Se um sistema está activo num grupo, o administrador pode ainda analisar as inconsistências na configuração local e, caso pretenda, actualizá-la com a existente na base de dados do MetaWebmin.

3.4.3 Gestão de grupos de sistemas

Um grupo de sistemas, na sua forma mais geral, tem vários sistemas que poderão corresponder a diferentes plataformas UNIX. Estes sistemas são configurados com a configuração do seu grupo, existente na base de dados do MetaWebmin, relativa aos serviços NFS e DNS, bem como utilizadores e respectivos grupos UNIX. Um grupo de sistemas caracteriza-se por:

- Identificação básica constituída por um identificador e um nome.

[3] Número identificativo do utilizador.

- Configuração das directorias a que as suas máquinas terão acesso através do serviço NFS.

- Configuração do ficheiro `/etc/resolv.conf`.

- Utilizadores e grupos UNIX criados nas suas máquinas.

A caracterização de um grupo de sistemas é fundamental no MetaWebmin, na medida em que determina o modo como todas as máquinas que lhe estejam associadas são configuradas.

3.4.4 Gestão de recursos de rede

Nesta fase de desenvolvimento consideraram-se dois recursos de rede; as directorias exportadas por NFS e a configuração de DNS, existente no ficheiro `/etc/resolv.conf`.

Relativamente aos recursos de NFS define-se a directoria de cada sistema do MetaWebmin exportada para a rede com determinadas permissões, actualizando em cada máquina o ficheiro `/etc/exports`.

Quanto ao recurso DNS, guarda-se no MetaWebmin a informação necessária à construção do ficheiro `/etc/resolv.conf` e que consiste no domínio e nos servidores utilizados na gestão dos nomes e endereços TCP/IP.

3.4.5 Recuperação de falhas

Um aspecto particularmente importante consiste na recuperação de erros existentes na ligação entre o MetaWebmin e os agentes Webmin dos respectivos sistemas. Existem dois tipos de controlo de falhas:

- Falhas ocorridas durante o processamento de um pedido. Estas falhas correspondem a erros ocorridos no processamento de um determinado pedido a um ou mais sistemas envolvidos. No final do processamento, o administrador tem a possibilidade de repetir os pedidos resultantes em erro, após ter identificado a causa e rectificado o problema. Para alguns tipos de erros permite-se a sua correcção automática nos sistemas.

- Recuperação global de falhas. Todos os eventos são inseridos numa tabela global de operações. O administrador pode em qualquer altura consultar a tabela e seleccionar um conjunto de pedidos que originaram em erro e iniciar o respectivo processo de recuperação.

3.4.6 Análise de inconsistências

Atendendo a que poderá haver gestão local nas máquinas, bem como problemas de ligação entre o MetaWebmin e os servidores Webmin daquelas, poderão ocorrer inconsistências entre a configuração dos sistemas e a centralizada no MetaWebmin. O administrador poderá, em qualquer momento, analisar as inconsistências existentes e, caso o pretenda, actualizar a configuração da máquina com a do grupo de sistemas a que pertence, existente na base de dados do MetaWebmin. Esta opção apenas está disponível em máquinas activas num grupo.

Capítulo 4

Implementação do MetaWebmin

Este capítulo descreve a estrutura de directorias e processo de instalação do MetaWebmin, os detalhes da implementação dos respectivos módulos e os modelos ER e relacional utilizados.

4.1 Estrutura de directorias

Os vários módulos do MetaWebmin estão desenvolvidos na linguagem *Perl*, na sua versão 5.6 [18] [41] [5]. O MetaWebmin instala-se habitualmente na directoria `/work/META/`, que tem a seguinte estrutura:

- `INSTALL/`

  Directoria dos ficheiros e programas necessários à instalação do MetaWebmin.

- `cgi-bin/`

  Directoria referente aos programas em *Perl* que implementam as CGIs utilizadas no processamento dos vários pedidos.

- `cgi-bin/log/`

  Directoria onde se colocam os ficheiros de *log* gerados pelo MetaWebmin e consultados através de um editor de texto.

- `cgi-bin/conf/`

  Directoria que contém o ficheiro `metawebmin.conf` de configuração do ambiente do MetaWebmin, nomeadamente variáveis e descrição de alguns conteúdos das páginas.

- `index.html`

  Página inicial de acesso ao MetaWebmin com as opções de navegação nos módulos existentes.

O ficheiro de configuração `cgi-bin/conf/metawebmin.conf` contém a definição de:

- Mensagens de erro tratadas pelo MetaWebmin.
- Variáveis utilizadas nos módulos do MetaWebmin, por exemplo, o porto TCP/IP do Webmin e os limites mínimo e máximo para a definição de utilizadores num grupo UNIX.
- Título e outra informação adicional das páginas do MetaWebmin.

A estrutura deste ficheiro é tabular, sendo os valores separados por ":" e acedidos pelo primeiro campo. Por exemplo:

```
hostactual_rep:Relatório de Diferenças Webmin - MetaWebmin
hostportowm:10000
```

onde `hostactual_rep` corresponde ao nome da variável (neste caso uma mensagem de erro) e o texto a seguir a ":" é o comentário explicativo da variável. Quanto ao segundo exemplo, corresponde à definição da variável `hostportowm` do porto Webmin nos sistemas, que é 10000.

Em novos desenvolvimentos deve utilizar-se este ficheiro para colocar toda a informação variável relativa a cada módulo. Da mesma forma, se se pretender alterar, por exemplo, a designação associada a uma mensagem de erro, dever-se-á fazê-lo aqui, mantendo intacto o código do MetaWebmin.

4.2 Instalação

Para instalar o MetaWebmin utiliza-se o *script* `install.pl`, existente na directoria `/work/META/INSTALL/`. Este *script* assegura as seguintes acções:

- Valida a existência dos componentes necessários à instalação e utilização do MetaWebmin, como sejam, SGBD e módulos de *Perl*.
- Copia para a directoria `/work/META/` a estrutura de subdirectorias e programas do MetaWebmin.
- Configura as variáveis mais importantes do MetaWebmin e constantes do ficheiro `/work/META/cgi-bin/conf/metawebmin.conf`.
- Configura os parâmetros de autenticação no servidor *Apache* do MetaWebmin, existentes no ficheiro `/etc/httpd/conf/httpd.conf`.

Antes de executar este *script* deverá garantir-se a instalação dos seguintes componentes:

1. *Perl 5.6*
2. SGBD MySQL [38]
   - MySQL-3.23.37-1
   - MySQL-client-3.23.37-1
   - MySQL-shared-3.23.37-1
   - MySQL-devel-3.23.37-1
3. Servidor Apache WebServer [3]
   - apache-1.3.14-3
4. Módulos de *Perl* [28]
   - CGI.pm
   - DBI-1.14 [1]
   - Digest-MD5-2.12
   - HTML-Format-1.23
   - HTML-Parser-3.23
   - HTML-Tagset-3.03
   - HTML-Tree-3.11
   - MIME-Base64-2.11
   - Msql-Mysql-modules-1.2215
   - URI-1.09
   - libnet-1.0703
   - libwww-perl-5.48

   O processo mais fácil e rápido de instalar estes módulos de *Perl* consiste na instalação remota, através da *Internet*, usando o CPAN [28]. Para o efeito, executar `perl -MCPAN módulo`. Através deste método instalam-se os módulos pretendidos e, caso existam, as respectivas dependências.

Nos sistemas geridos pelo MetaWebmin deverão executar-se as seguintes tarefas:

1. Instalar o servidor Webmin na versão 0.83 para a plataforma UNIX respectiva, conforme descrito em [46].

2. Copiar o *script* `create_dns_client.cgi` existente na directoria de instalação do MetaWebmin para a directoria `$WEBMIN_DIR/dnsadmin/` do Webmin, sendo a variável `$WEBMIN_DIR` o caminho onde está instalado o servidor Webmin. Este programa cria o ficheiro `/etc/resolv.conf` de acordo com os valores inseridos no formulário de criação de DNS no MetaWebmin.

3. Copiar o *script* `rec_autom_user.cgi` existente na directoria de instalação do MetaWebmin para a directoria `$WEBMIN_DIR/useradmin/` do Webmin. Este programa é executado quando o administrador pretende recuperar automaticamente a criação de um utilizador que não foi bem sucedida. Todas as situações de erro na criação de um utilizador, que possam ser recuperadas automaticamente, deverão incluir-se neste programa.

4.3 Modelo de dados

O modelo lógico de dados utilizado é o relacional. Neste modelo representam-se as entidades e relações necessárias à gestão dos dados do MetaWebmin relativos a utilizadores, grupos UNIX, sistemas, grupos de sistemas e recursos de NFS e DNS.

4.3.1 Entidades e associações

Para representar a informação necessária ao MetaWebmin identificam-se as seguintes entidades do modelo lógico de dados:

Utilizador Utilizadores geridos pelo MetaWebmin.

GrupoUNIX Grupos UNIX, primários e secundários a que pertencem os utilizadores geridos pelo MetaWebmin.

Sistema Sistemas geridos pelo MetaWebmin, identificados por um nome num domínio.

GrupoSistema Grupo de sistemas, identificado por um nome, a que será associada a configuração de recursos DNS e NFS, utilizadores e grupos UNIX.

RecursoNFS Directorias exportadas por sistemas geridos pelo MetaWebmin, com determinadas permissões de acesso, através do protocolo NFS.

RecursoDNS Configuração do cliente de DNS que permitirá criar o ficheiro `/etc/resolv.conf`.

Entre estas entidades identificam-se os seguintes relacionamentos:

Pertence (Utilizador, GrupoUNIX) Um utilizador ”pertence” sempre a um grupo UNIX primário e, opcionalmente, a vários grupos secundários.

Existe (Utilizador, GrupoSistema) Um utilizador ”existe” em zero ou mais grupos de sistemas. As alterações efectuadas a um utilizador reflectem-se em todos os sistemas dos grupos de sistemas a que pertence.

Tem (Sistema, GrupoSistema) Um sistema ”pertence”, em cada momento, apenas a um grupo de sistemas. Um grupo de sistemas poderá ter associados vários sistemas. Um sistema pode estar criado no MetaWebmin mas não estar associado a qualquer grupo.

Configura (RecursoNFS, GrupoSistema) Associa a configuração de um ou mais recursos de NFS a um grupo de sistemas.

Configura (RecursoDNS, GrupoSistema) Associa a configuração de uma, e apenas uma, definição do cliente de DNS a um grupo de sistemas.

A figura 4.1 ilustra o modelo Entidade-Associação [37] relativo ao MetaWebmin. Quanto à participação das entidades nos relacionamentos, pode ser total ou parcial. As participações parciais estão representadas a traço fino e as totais a traço mais grosso. A seta em traço grosso da entidade **Sistema** para o relacionamento **Tem** indica que cada sistema pertence (ou está associado) a exactamente um e apenas um grupo de sistemas. Para simplificar a figura não se apresentam os atributos das entidades e associações.

4.3.2 Modelo relacional

O modelo de dados definido na secção anterior mapeia-se facilmente num modelo relacional. A seguir descrevem-se as tabelas correspondentes, indicando-se os atributos e atributos chave de cada uma.

- Tabela UTIL (Utilizadores)
  - Atributo chave: *login* do utilizador.
  - Outros atributos: UID, nome do utilizador, *home directory*, informação relativa à *password* e *shell* utilizada.
- Tabela GUNIX (Grupos de Utilizador)
  - Atributo chave: Nome do grupo UNIX.

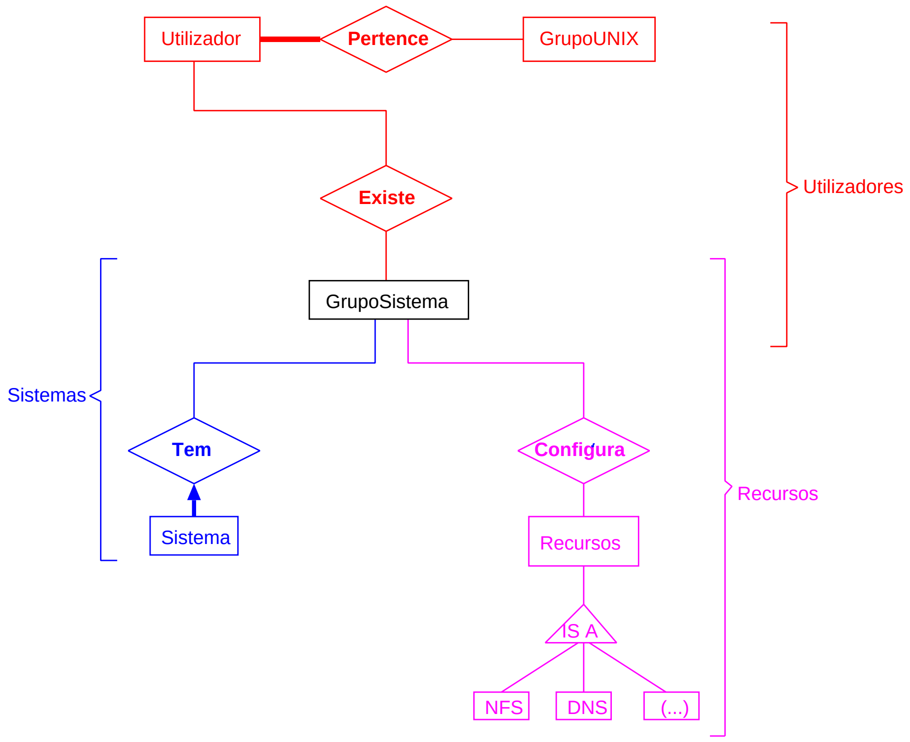

Figura 4.1: Modelo Entidade-Associação do MetaWebmin

  - Outros atributos: GID e intervalo de UIDs para a criação de utilizadores nesse grupo.

- Tabela SIS (Sistemas)
  - Atributos chave: Identificador do sistema na rede e respectivo domínio.
  - Outros atributos: Designação do sistema para melhor identificação e informação sobre o acesso ao servidor Webmin (*username* e *password*).

- Tabela GSIS (Grupo Sistemas)
  - Atributo chave: Identificador do grupo de sistemas.
  - Outros atributos: Designação completa e localização do grupo de sistemas.

- Tabela SRV_NFS (Recursos de NFS)
  - Atributo chave: Número sequencial que identifica o recurso de NFS.
  - Outros atributos: Sistema da rede que disponibiliza o recurso, directoria exportada e respectivas permissões.

- Tabela SRV_DNS (Recursos de DNS)
  - Atributo chave: Número sequencial que identifica o recurso de DNS.
  - Outros atributos: Domínio e servidores DNS a pesquisar.

- Tabela UTIL_GUNIX
  - Atributos chave: Identificadores do utilizador e do grupo UNIX.
- Tabela UTIL_GSIS
  - Atributos chave: Identificadores do utilizador e do grupo de sistemas.
- Tabela GSIS_SIS
  - Atributos chave: Identificadores do grupo de sistemas e do sistema (nome e domínio).
- Tabela NFS_GSIS
  - Atributos chave: Identificadores do grupo de sistemas e do recurso de NFS.
- Tabela DNS_GSIS
  - Atributos chave: Identificadores do grupo de sistemas e do recurso de DNS.

A figura 4.2 ilustra as várias tabelas do modelo relacional e as dependências entre as tabelas através das chaves externas.

4.4 Criação da base de dados

O SGBD MySQL [38] está disponível gratuitamente na *Internet*. A versão instalada no MetaWebmin é a `3.23.37-1`. Relativamente a outros SGBD, o MySQL tem algumas limitações, por exemplo:

- Não permite o tratamento de *constraints*.
- Não permite a definição de *triggers*.
- Não permite o uso de questões complexas.
- Não permite a definição de *views*.

Contudo, no caso concreto do MetaWebmin, e nesta fase de desenvolvimento, o MySQL serve perfeitamente os requisitos pretendidos de um SGBD.

No ficheiro `/work/META/INSTALL/def_tabelas.sql` definem-se as instruções de SQL para criação das tabelas do modelo de dados relacional do MetaWebmin.

A inicialização da base de dados efectua-se através dos seguintes programas existentes em `/work/META/INSTALL/`:

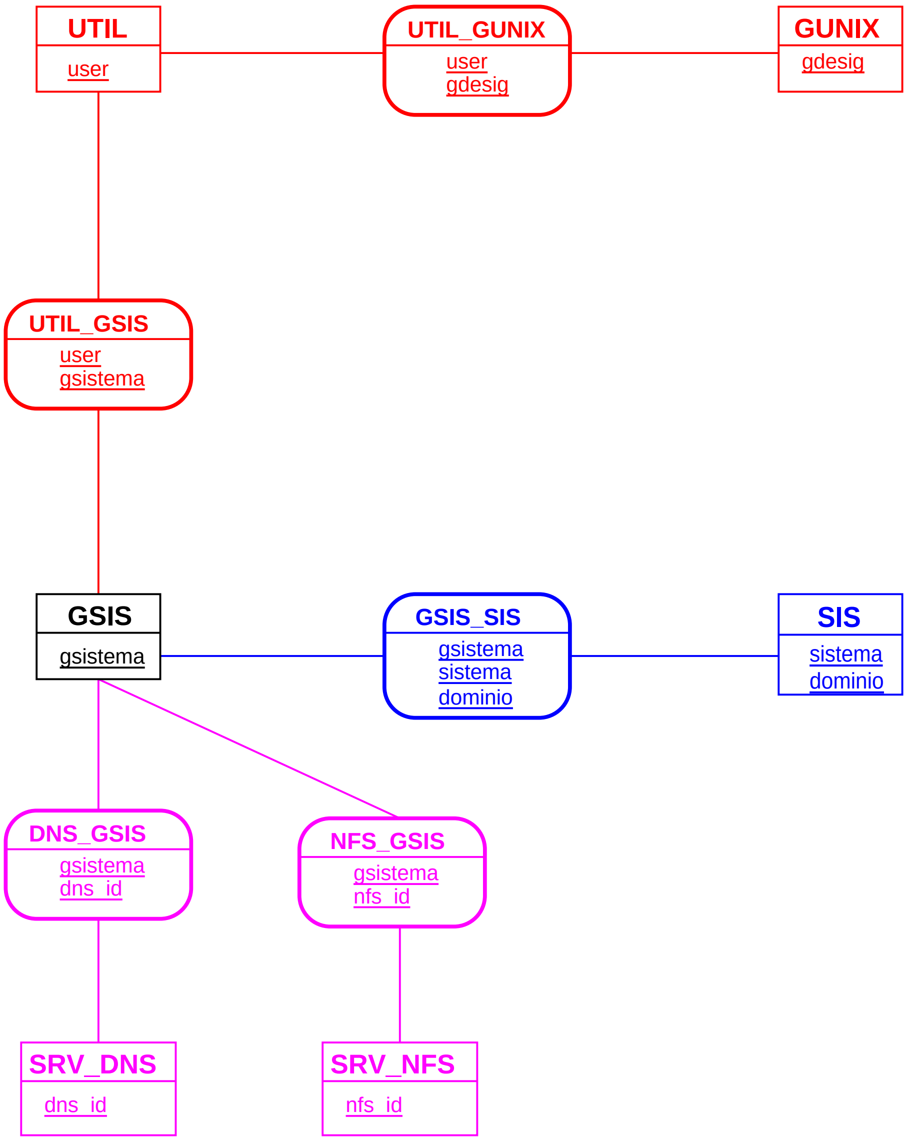

Figura 4.2: Modelo de dados relacional do MetaWebmin

- `cria_tabelas.pl`

  Com base no ficheiro `/work/META/INSTALL/def_tabelas.sql`, cria a base de dados `mwebmin` no MySQL e as tabelas do MetaWebmin.

- `apaga_tabelas.pl`

  Remove, do MySQL, as tabelas do MetaWebmin e a base de dados `mwebmin`.

Na figura 4.2 apresenta-se o modelo relacional de dados do MetaWebmin, onde se ilustram as tabelas descritas anteriormente.

4.5 *Interface* com a base de dados

A comunicação do MetaWebmin com a base de dados faz-se através de uma *interface* definida pelo módulo DBI [1] de *Perl* com os *drivers* do SGBD MySQL.

4.5.1 Módulo DBI

O DBI é um módulo independente da base de dados que disponibiliza um conjunto consistente de rotinas de acordo com o SGBD designado na sua utilização. A vantagem de utilizar o módulo DBI de *Perl* consiste em ter apenas instalado um *driver* DBI, evitando assim uma multiplicidade de *interfaces* dos vários SGBDs.

Conforme [1], a arquitectura do DBI divide-se em dois grupos principais de *software*: a API do DBI e os *drivers* para os SGBDs. O primeiro consiste na *interface* de programação e nos métodos de acesso aos *drivers* específicos da base de dados. No segundo implementam-se os vários *drivers* para os diferentes tipos de base de dados suportados, disponibilizando as operações de manipulação das tabelas, conforme se ilustra na figura 4.3.

Quando se efectua uma operação de manipulação numa base de dados, evoca-se um método DBI indicando o nome do *driver* referente ao SGBD. O DBI passa como parâmetro o método para o *driver* especificado para execução da operação a ele associado. Assim, o módulo DBI não implementa operações específicas para um determinado SGBD, mas sim genéricas passadas para o driver do SGBD pretendido.

Num *script* em *Perl* a passagem de um método para um *driver* de MySQL implica a execução dessa operação na base de dados e, consequentemente, a passagem do resultado para variáveis do *script*. Toda a informação trocada entre o DBI e os seus *drivers* utiliza tipos de dados *standard* do *Perl*, isolando assim o módulo DBI do conhecimento do SGBD utilizado. A figura 4.4 ilustra esse mecanismo.

[1] *DataBase Independent*

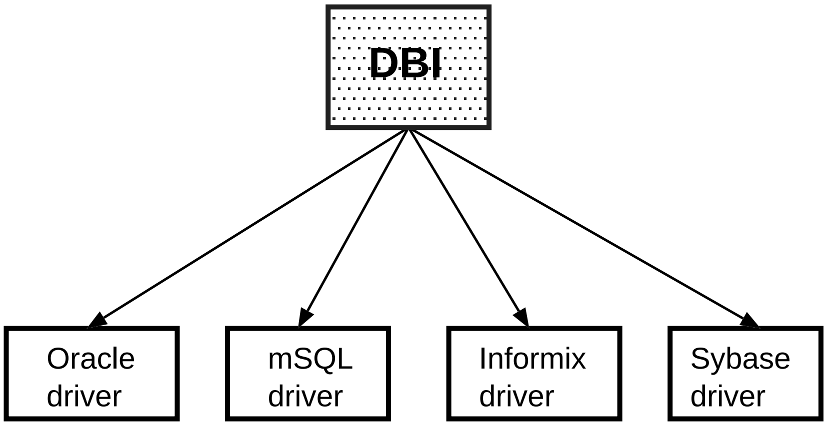

Figura 4.3: Arquitectura do módulo DBI de *Perl*

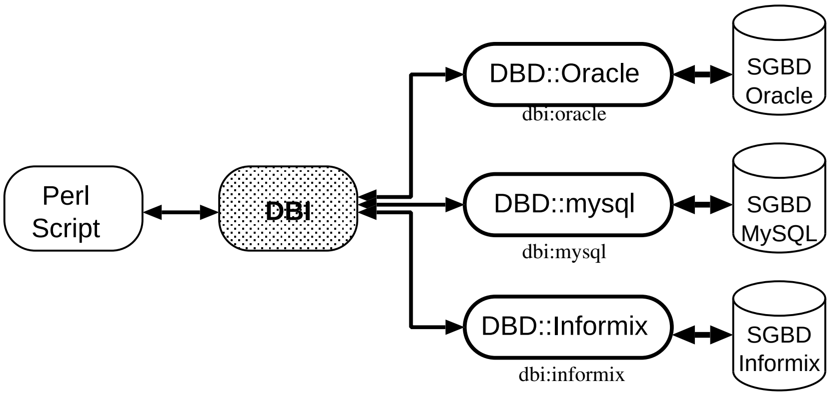

Figura 4.4: Processamento de um método do módulo DBI

4.5.2 Exemplo utilizando o *Perl* DBI

No ficheiro `/work/META/cgi-bin/metafunc-lib.pl` do MetaWebmin disponibilizam-se várias funções utilizadas nos programas do MetaWebmin. Entre essas encontram-se as de consulta, inserção, remoção, alteração e listagem dos registos existentes nas tabelas da base de dados.

Como exemplo, analisemos a rotina seguinte que insere os dados do sistema na tabela `SIS` da base de dados `mwebmin`. Os dados são aceites e enviados para a base de dados após o preenchimento do formulário disponibilizado pelo MetaWebmin para o efeito.

```
#!/usr/bin/perl -w                                                  (1)

use DBI;                                                            (2)

my $dbh = DBI->connect( 'DBI:mysql:mwebmin' , 'username',
                        'password' );                               (3)

my $sth = "insert into SIS (id, desig, dominio, wmport,
                  wmuname, wmpwd, com) values ('01', 'mwm',
                  'ncc.up.pt', '10000', 'admin', 'admin',
                  'Desenv')";                                       (4)

my $in  = $dbh->prepare($sth);                                      (5)

$in ->execute;                                                      (6)

$dbh->disconnect();                                                 (7)

exit;                                                               (8)
```

Em (1) evoca-se o interpretador de *Perl* com a opção para mostrar os avisos (opção -w) durante a execução do programa.

Em (2) o programa localiza e disponibiliza o módulo DBI, sendo os *drivers* de cada base de dados carregados conforme vão sendo utilizados no programa.

A definição do tipo de base de dados a utilizar e respectivo *driver* efectua-se em (3). Para tal cria-se uma ligação (*handler*) para a instância da base de dados do MetaWebmin, denominada `mwebmin`, que corresponde a uma base de dados MySQL denominada `mwebmin`, acessível com o *username* e *password*.

Em (4) define-se a variável `$sth` com a instrução SQL a enviar para o *driver* de MySQL. Nos programas do MetaWebmin a construção desta variável é dinâmica, ou seja, a tabela, o nome dos campos e respectivos valores são passados por parâmetro para uma rotina genérica de inserção. As operações de remoção, alteração e consulta de dados nas tabelas usam um processo semelhante ao descrito.

O passo (5) corresponde à formatação da instrução de SQL definida em `$sth`, na base de dados referenciada por `$dbh`.

A execução da operação de SQL na base de dados faz-se em (6). Em (7) remove-se a ligação à base de dados referenciada pela variável `$dbh`. O fim do programa ocorre em (8).

O módulo DBI dispõe ainda de métodos para tratamento de erros. Após a execução da operação de SQL podem evocar-se os métodos `err`, `errstr` e `state` para saber o estado actual de uma ligação à base de dados, *driver*, base de dados ou instrução.

Um aspecto importante na gestão da base de dados consiste na garantia de integridade da mesma. Na medida em que o SGBD utilizado não garante a validação de *constraints*, os programas que inserem, apagam ou alteram registos validam todos os dados manipulados antes da respectiva operação nas tabelas.

4.6 Desenvolvimento

Esta secção descreve as opções que foram tomadas no desenvolvimento dos módulos do MetaWebmin.

4.6.1 Gestão de utilizadores

A atribuição do UID do utilizador é dinâmica, de acordo com o intervalo do grupo UNIX primário seleccionado. O administrador pode alterar o UID de um utilizador, reflectindo essas alterações em todos os grupos de sistemas onde esteja criado. Todavia, verifica-se sempre se o UID é válido no grupo UNIX primário seleccionado. A alteração do grupo UNIX primário de um utilizador implica a sua validação no intervalo de UIDs do novo grupo.

4.6.2 Gestão de grupos UNIX

Cada grupo UNIX tem uma designação e um GID que o identifica no MetaWebmin e em todos os sistemas onde é criado. O GID do grupo deverá ser superior ao valor atribuído à variável `gbasegid` do MetaWebmin. O valor definido por defeito é 550, evitando colisões com os GIDs dos grupos UNIX reservados a utilizadores de gestão e de aplicações eventualmente instaladas. Define-se ainda um intervalo mínimo e máximo para a criação dos utilizadores dentro de um grupo, de acordo com as necessidades de administração.

As seguintes regras garantem a consistência da informação relativa a grupos UNIX na base de dados.

1. Os intervalos deverão pertencer ao domínio definido pelos valores das variáveis `gminuid` e `gmaxuid` do MetaWebmin. Os valores definidos por defeito são 1000 e 65534 respectivamente, evitando colisões com utilizadores de sistema e de aplicações instaladas, normalmente criados abaixo do valor 1000.

2. Os intervalos definidos têm obrigatoriamente que ser disjuntos, ou seja, não poderão ser definidos subintervalos, nem os limites superior ou inferior poderão interferir com os de intervalos de grupos UNIX já existentes.

3. A variável do MetaWebmin `gfolga` define a folga de UIDs existente entre cada grupo UNIX. Este valor permitirá aumentar de forma contígua o número de UIDs disponíveis num grupo UNIX.

4. Os intervalos têm de ter, pelo menos, 5 UIDs.

A gestão do espaço de UIDs deverá efectuar-se com bom senso por parte do administrador de sistemas. É importante haver um planeamento prévio por forma a garantir uma distribuição próxima da realidade.

Uma das operações que o administrador não consegue efectuar é criar intervalos de UIDs não contíguos num mesmo grupo UNIX, permitindo a folga definida apenas uma extensão contígua. Todavia, o administrador pode prescindir da folga e criar aí um novo grupo.

Face a este problema, a solução ideal consiste em permitir que o administrador crie vários intervalos não contíguos de UIDs para o mesmo grupo. As alterações a efectuar para implementar esta solução consistem em:

- Alterar a definição da tabela de grupos UNIX, passando a ter como campos chave a designação do grupo e o identificador do intervalo. Esta alteração implica a revisão de todos os programas onde seja utilizada a tabela de grupos UNIX.

- Modificar as *interfaces* de criação e alteração do grupo UNIX, por forma a incluir a gestão de intervalos.

4.6.3 Gestão de sistemas

Um sistema apenas pode estar associado, num determinado momento, a um grupo de sistemas previamente criado no MetaWebmin. O processo de activação de um sistema inicia-se apenas se o sistema responde a um pedido de `ping` e o seu servidor Webmin está operacional. Este processo consiste em copiar para o sistema toda a informação do MetaWebmin relativa à configuração do grupo de sistema a que está a ser activado. O processo de activação não apaga configurações existentes, nomeadamente relativas a outros grupos de sistemas a que o sistema tenha pertencido. Por esse facto, é altamente recomendável, após activar um sistema num grupo, verificar possíveis inconsistências e, caso necessário, actualizá-lo com a informação do MetaWebmin.

Desactivar um sistema consiste apenas em eliminar a sua associação ao grupo onde está activo.

Actualizar um sistema consiste em listar as diferenças existentes entre os ficheiros de configuração locais e os dados do MetaWebmin e, caso pretenda, o administrador pode efectuar a actualização do sistema. O método para a análise de inconsistência nos utilizadores e grupos UNIX consiste em:

- Verificar grupos UNIX que existam no sistema em actualização e possuam um GID que conste na base de dados do MetaWebmin. A definição da variável `gbasegid` é de extrema importância, na medida em que corresponde ao valor a partir do qual serão identificados os GIDs válidos. Poderá ainda suceder que seja criado localmente um grupo UNIX com um GID dentro dos valores geridos pelo MetaWebmin. Esta situação originará uma inconsistência entre a configuração local dos grupos UNIX e a existente no MetaWebmin.
- Verificar utilizadores que existam no sistema em análise, possuam um UID pertencente a um intervalo dos grupos UNIX geridos pelo MetaWebmin e não existam na base de dados central. Todos os utilizadores pertencentes a grupos UNIX não geridos pelo MetaWebmin, mesmo se criados localmente, serão considerados grupos UNIX de sistema. Aqui igualmente se destaca a necessidade de planear a distribuição dos UIDs pelos grupos UNIX criados.

Em todos os desenvolvimentos efectuados houve a preocupação de validar o maior número de possíveis situações de erro nas *interfaces* do MetaWebmin. Foi assim possível evitar que muitas situações de erro ocorressem no servidor Webmin. Por exemplo, no Webmin a *home directory* deve iniciar-se sempre por "/". Essa validação efectua-se nos formulários de criação e alteração do utilizador, evitando-se o pedido ao Webmin e a esperada resposta com uma situação de erro.

4.7 *Interface* com o Webmin

Os pedidos HTTP do MetaWebmin aos servidores Webmin dos sistemas da rede efectuam-se através do módulo de LWP, *Library for WWW in Perl* [17] [8]. Este módulo de *Perl* disponibiliza uma API que permite interagir com serviços da Web, por exemplo, FTP [2] e HTTP. A sua estrutura é orientada a objectos. Cada classe disponibiliza um conjunto de métodos para interagir com os atributos dos objectos. Algumas das funcionalidades do LWP são:

- Pedir (GET) um documento a um servidor Web usando o seu URL.
- Colocar (POST) um documento num servidor Web. Por exemplo, através da submissão de um formulário.

[2] *File Transfer Protocol*

- Copiar um documento da Web apenas se corresponder a uma versão mais recente da existente no servidor local.

- Efectuar *parse* de um documento HTML, podendo assim seleccionar *links* ou outra informação de interesse.

- Garantir operações de redireccionamento, autenticação, *proxy*, *cache* e utilização de *cookies*.

Nos pedidos HTTP entre o MetaWebmin e os servidores Webmin utiliza-se a classe de objectos `HTTP::Request` do LWP e o método *POST*.

Um exemplo consiste na preparação e execução do pedido HTTP no sistema `lab1`, correspondendo à criação de um novo grupo UNIX.

```
# Pedido HTTP ao servidor Webmin do sistema lab1.ncc.up.pt.

$str ="http://lab1.ncc.up.pt:10000/useradmin/save_group.cgi?
       group=gu1\&gid=555\&passmode=0\&encpass=\&pass=\&members=";

# Criar um novo objecto LWP.

my $agent    = LWP::UserAgent->new;

# Criar um novo objecto da classe HTTP::Request, do tipo POST,
# com o argumento $str

my $req      = HTTP::Request->new(POST => $str);

# Definir os parâmetros de autorização ao servidor Webmin.

$req->authorization_basic('admin', 'admin');

# Definir o content-type do pedido.

$req->content_type('application/x-www-form-encoded');

# Executar o pedido com as definições efectuadas em $req.
# Em $res guarda-se o resultado do pedido.

$res = $agent->request($req);

# Imprimir o resultado do pedido.

print $res->content;
```

Este exemplo apenas considera o pedido a um sistema. O MetaWebmin utiliza uma rotina genérica denominada `pedido`, a qual é evocada para a construção de um pedido HTTP a vários grupos de sistemas. No exemplo seguinte utiliza-se essa mesma rotina para criar um grupo UNIX em vários sistemas da rede.

```
# Neste ponto do programa já foi actualizada a base de dados
# e seleccionados os sistemas envolvidos no pedido.
# O array @hosts contém o nome dos sistemas.

foreach my $j ( @hosts )
{
       # O hash array %host guarda os dados de cada sistema.
       %host = &consulta_host ($j);

       # A rotina &ping_webmin verifica se o servidor Webmin
       # do sistema está activo.

       $rc = &ping_webmin($host{'hid'}, $host{'hdominio'},
                          $host{'hwmport'}, $host{'husername'},
                          $host{'hpasswd'}
                         );

       # A rotina &num_u_g valida se o utilizador já existe.

       if ($rc == 0)  { @erros="Webmin Down."; $ok=0; }
       if ($rc == 1)  { $num = &num_u_g($host{'hid'}, 'group',
                                        $valores{'gdesig'} );
                      }

       # A variável $str define o pedido ao servidor Webmin.

       $str = "http://$host{'hid'}.$host{'hdominio'}:
               $host{'hwmport'}/useradmin/save_group.cgi
               ?group=\$valores{'gdesig'}
               \&gid=$valores{'gid'}\&passmode=0\&encpass=
               \&pass=\&members=";

       if ( $num eq "" && $rc == 1 )
       {
             # Executar pedido ao servidor Webmin.
             # Validar resultado do pedido.

             $res = &pedido( $str,'POST', $host{'husername'},
                                 $host{'hpasswd'});
             @erros = &testa_pedido_webmin ($res);
       }
```

```
            # Caso exista, imprime erro do pedido formatado e
            # actualiza log de eventos.

            if ($#erros>-1) {&imprime_erro($erros[0], $j); $ok=0;}
            &log_webmin_sessao($erros[0], $j, $str, 'GUNIX',
                               'Adicionar',$valores{'gdesig'},$ok);
    }
```

A API do LWP disponibiliza ainda métodos que permitem validar situações de erro que ocorram na execução de um pedido HTTP.

4.8 Tratamento de falhas

Um aspecto importante do MetaWebmin consiste na possibilidade de recuperar pedidos efectuados aos servidores Webmin que não foram concretizados com sucesso. As situações em que o sucesso de um pedido do MetaWebmin a um servidor Webmin pode estar comprometido são:

- Sistema não responde ao comando `ping`.
- Servidor Webmin do sistema não responde.
- Erro na execução de um pedido no servidor Webmin.

Considerando o exemplo de criação de um utilizador num grupo de sistemas, constituído por 35 máquinas, a possibilidade de um dos pedidos não ser concluído com sucesso é grande. No entanto, havendo sucesso em pelo menos um, este não poderá deixar de ser actualizado pelo erro ocorrido nos restantes 34 sistemas. O método de processamento de um pedido e tratamento de falhas consiste em:

1. Actualizar sempre a base de dados do MetaWebmin, independentemente da ocorrência ou não de erro nos pedidos efectuados. De notar que um utilizador pode associar-se a um grupo de sistemas que, naquele momento, ainda não tenha associado nenhum sistema.
2. Efectuar o pedido a todos os sistemas envolvidos.
3. Registar o resultado do pedido.
4. Disponibilizar ao administrador de sistemas mecanismos de recuperação de falhas. Essa recuperação poderá ser imediata ou efectuada mais tarde. Existe ainda a possibilidade de aceitar a sugestão de recuperação automática, disponível para alguns erros.

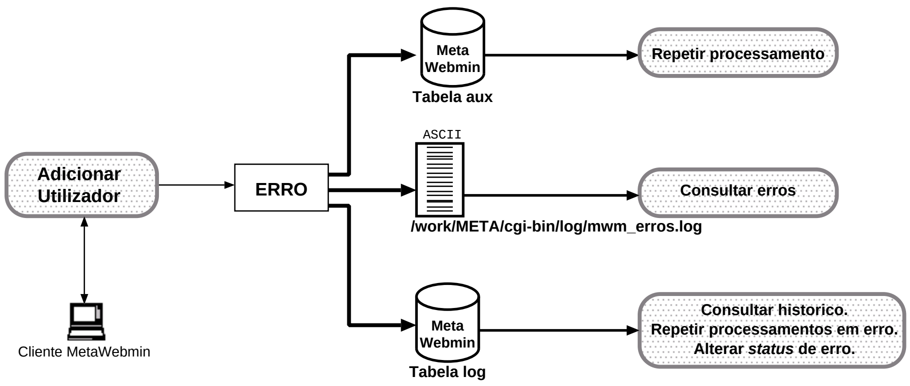

Figura 4.5: Registo de eventos no MetaWebmin

Quando ocorre uma situação de erro o administrador é sempre notificado e o evento registado em três repositórios diferentes com finalidades igualmente distintas, conforme se ilustra na figura 4.5. O registo dos eventos faz-se:

1. Numa tabela auxiliar da base de dados, criada no início de cada processamento, para recuperação imediata do pedido, se este terminar numa situação de erro.

2. Num ficheiro de texto, onde são armazenados todos os erros ocorridos no MetaWebmin.

3. Numa tabela de *log* da base de dados, onde são inseridos todos os eventos ocorridos. O MetaWebmin disponibiliza uma *interface* para gestão desta tabela, onde é possível recuperar pedidos em erro ou consultar o histórico.

Sempre que possível, associa-se à mensagem de erro uma sugestão para resolução do problema. Essa sugestão pode consistir numa descrição de uma acção correctiva ou, quando disponível, na execução de um *script* gerado dinamicamente no sistema onde ocorreu o erro.

O administrador tem ao seu dispor várias formas de tratar os erros, ilustradas na figura 4.6:

1. Corrigir a situação de erro manualmente e repetir de imediato o pedido.

2. Aceitar a sugestão do MetaWebmin e executar o *script* associado ao erro para a sua correcção.

3. Diferir a correcção, preferindo efectuar uma análise cuidada dos pedidos em erro e repeti-los posteriormente.

Para a gestão da tabela de erros diferidos existe uma *interface* que permite:

- Seleccionar um ou mais pedidos em erro para execução.

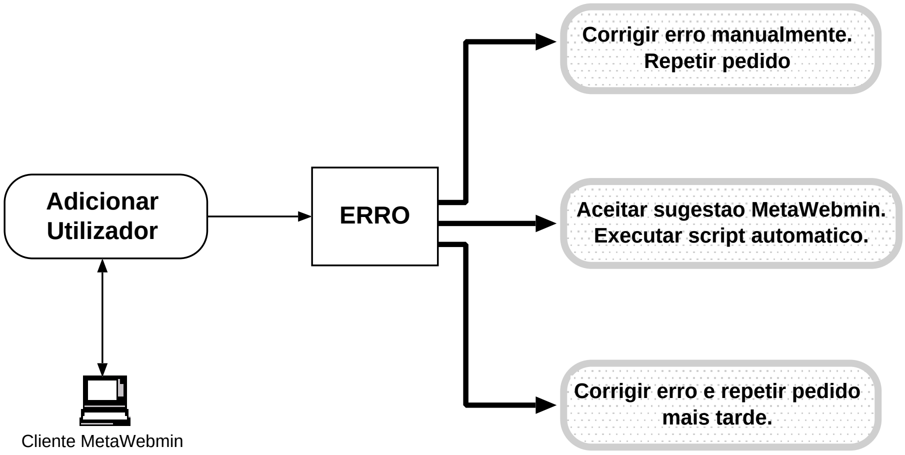

Figura 4.6: Formas de tratamento de erros do MetaWebmin

- Executar a correcção automática sugerida pelo MetaWebmin.
- Alterar o *status* de um determinado erro, ignorando-o e passando-o para resolvido.

Na tabela global de erros, o *status* de um registo pode ser:

- **OK**, sempre que a execução de um determinado pedido é concluída com sucesso;
- **ERRO**, quando o pedido originou uma situação de erro que ainda não foi corrigida;
- **OK-IGNORAR**, se o administrador assume que o erro está resolvido e força a alteração do *status*;

Na figura 4.7 apresenta-se o ciclo de vida de um pedido e processamento dos seus possíveis erros. Apresentam-se ainda os diferentes códigos de *status* e respectivo valor no ciclo de vida do pedido. O registo das alterações do *status* de um pedido efectua-se nas formas existentes para o efeito ilustradas na figura 4.5.

4.9 Análise de inconsistências

Nas operações de gestão de sistemas poderão ocorrer inconsistências entre a configuração existente nos ficheiros locais e a centralizada na base de dados do MetaWebmin. São duas as situações que poderão originar tal facto:

1. Administração local e directa em sistemas geridos pelo MetaWebmin por um utilizador com privilégios. Por exemplo, se um utilizador aceder a um sistema com uma conta privilegiada e adicionar um novo utilizador.

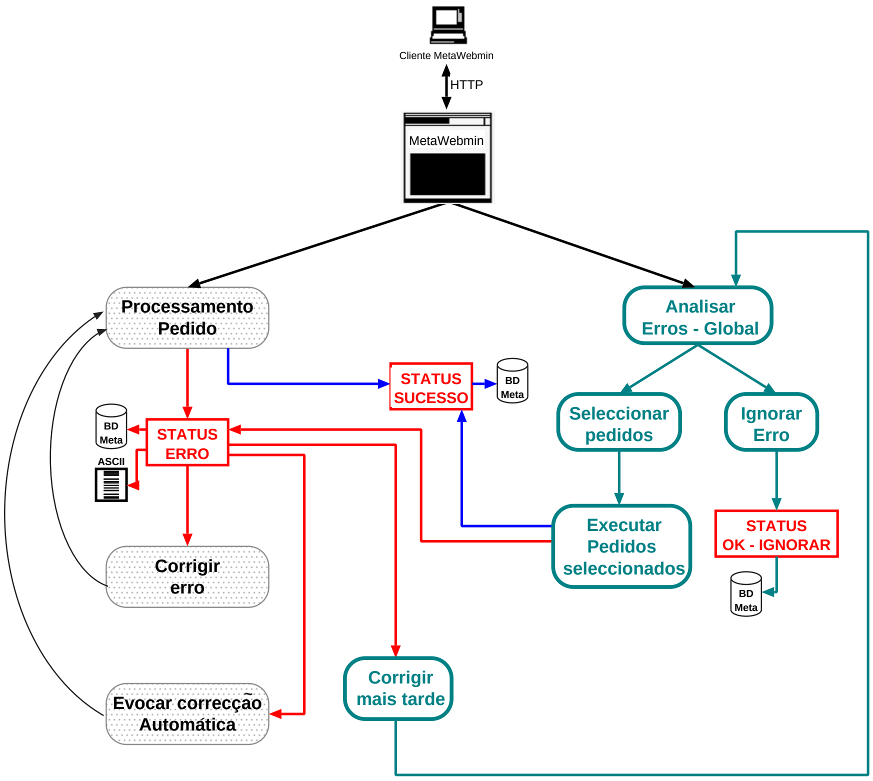

Figura 4.7: Tratamento global de erros do MetaWebmin

2. Pedido efectuado pelo MetaWebmin a um grupo de sistemas que não foi concluído com sucesso nalguns dos seus sistemas. Por exemplo, se os seus servidores Webmin estiverem inactivos.

A gestão centralizada da rede é vantajosa se existirem mecanismos que garantam que a configuração de cada sistema está consistente com a centralizada no MetaWebmin. Ou seja, o administrador pode, em qualquer momento, validar a configuração local de um sistema com a existente no MetaWebmin, referente ao grupo de sistemas em que está activo naquele momento.

A figura 4.8 ilustra os processos de detecção de inconsistências e actualização da configuração do sistema local.

No exemplo apresentado analisam-se as diferenças entre a configuração local do sistema `s_1`, activo no grupo `LAB1`, e a configuração do MetaWebmin para este grupo de sistemas. Após essa análise actualiza-se a configuração local do sistema `s_1` com as diferenças detectadas. O resultado da análise de diferenças é de dois tipos, com utilizações distintas:

- Um relatório com as diferenças, para análise do administrador.
- Uma estrutura de dados que servirá de *input* ao processo de actualização do sistema com a configuração existente na base de dados do MetaWebmin.

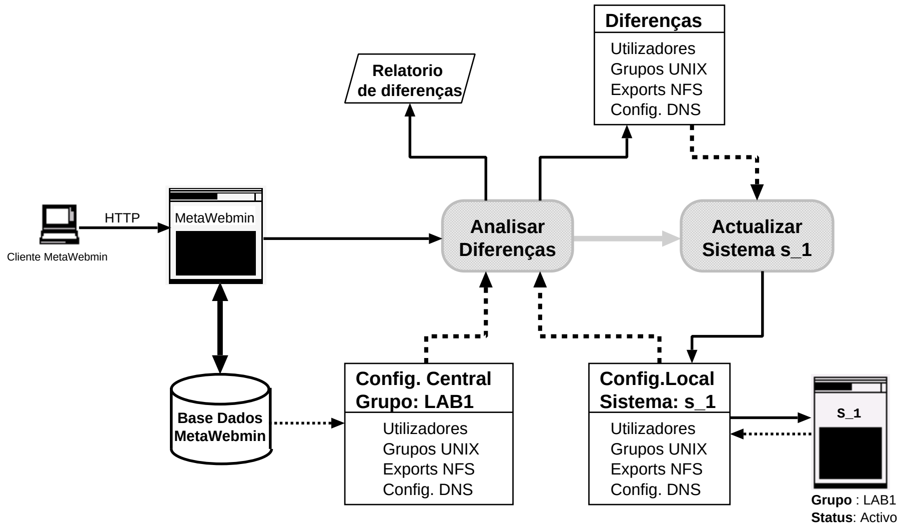

Figura 4.8: Análise de inconsistências usando o MetaWebmin

Para a análise da configuração local do sistema efectua-se o *parse* das tabelas em HTML produzidas pelo Webmin com a informação relativa, por exemplo, aos utilizadores e grupos UNIX. A figura 4.9 ilustra um extracto da tabela de utilizadores activos no sistema.

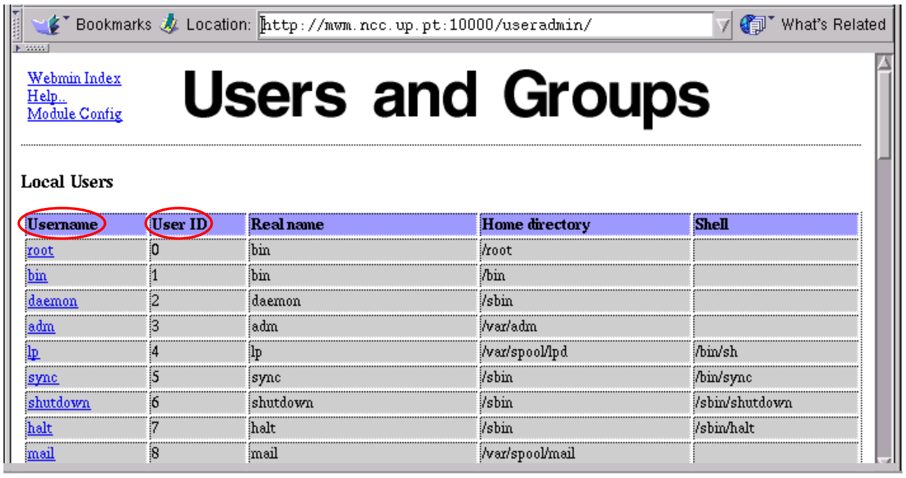

Figura 4.9: Tabela de utilizadores activos num sistema gerada pelo Webmin

Nem todos os campos existentes nesta tabela interessam para validar a consistência da configuração. No exemplo apresentado apenas serão considerados os campos *Username* e *User ID*. Esta tabela é gerada dinamicamente e corresponde a um documento em HTML cujo extracto se apresenta na figura 4.10. Nesta figura assinalam-se com um circulo as *tags* utilizadas para representar os campos do nome e UID do utilizador, na tabela de utilizadores do Webmin.

Uma análise cuidada permite definir uma estrutura em árvore da tabela de utilizadores gerada pelo Webmin. A figura 4.11 ilustra essa estrutura, onde se identificam os

```
<!doctype html public "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<meta http-equiv="Content-Type" content="text/html; Charset=iso-8859-1">
<head>
<title>Users and Groups</title>
<SCRIPT LANGUAGE="JavaScript">

defaultStatus="admin logged into Webmin 0.83 on mypc.teste.pt (Redhat Linux 7

</SCRIPT>
</head>
<body bgcolor=#ffffff link=#0000ee vlink=#0000ee text=#000000  >
<table width=100%><tr>
<td width=15% valign=top align=left><a href='/?cat=system'>Webmin Index</a><b
<a onClick='window.open("/help.cgi/useradmin/intro", "help", "toolbar=no,menul
<a href="/config.cgi?useradmin">Module Config</a><br>
</td>
<td align=center width=70%><img src=/images/letters/85.gif alt="U" align=bott
<td width=15% valign=top align=right></td></tr></table>
<hr>
<h3>Local Users</h3>
<table width=100% border>
<tr bgcolor=#9999ff> <td><b>Username</b></td> <td><b>User ID</b></td> <td><b>
<tr bgcolor=#cccccc> <td><a href='edit_user.cgi?num=0'>root</a></td> <td>0</t
<tr bgcolor=#cccccc> <td><a href='edit_user.cgi?num=1'>bin</a></td> <td>1</td
<tr bgcolor=#cccccc> <td><a href='edit_user.cgi?num=2'>daemon</a></td> <td>2</
<tr bgcolor=#cccccc> <td><a href='edit_user.cgi?num=3'>adm</a></td> <td>3</td
<tr bgcolor=#cccccc> <td><a href='edit_user.cgi?num=4'>lp</a></td> <td>4</td>
```

Figura 4.10: Extracto do código HTML da tabela de utilizadores

diversos ramos, correspondentes às *tags*, e os nós terminais aos seus valores.

A metodologia de análise de inconsistências para os utilizadores é a seguinte:

1. *Parse* da tabela de utilizadores gerada pelo Webmin para o *array* A.
2. Listar os utilizadores registados na base de dados do grupo de sistemas a que pertence o sistema em análise, para o *array* B.
3. Calcular os limites mínimos e máximos dos grupos UNIX administrados pelo MetaWebmin.
4. Identificar utilizadores que existem no MetaWebmin e não existem na configuração local. Se o resultado for diferente do conjunto vazio, então o sistema está inconsistente.
5. Para os utilizadores que existem na configuração local e não existem no MetaWebmin verificar se o UID está incluído nalgum dos seus intervalos:
   (a) Se for o caso, o sistema está inconsistente e o utilizador é incluído na lista de utilizadores a remover na configuração local.
   (b) Caso contrário, o utilizador considera-se de sistema ou associado a uma aplicação instalada localmente.
6. Se o conjunto de utilizadores a apagar ou adicionar à configuração local é diferente de vazio, então o sistema está inconsistente. Nesse caso, ocorrerá uma das três possíveis acções:

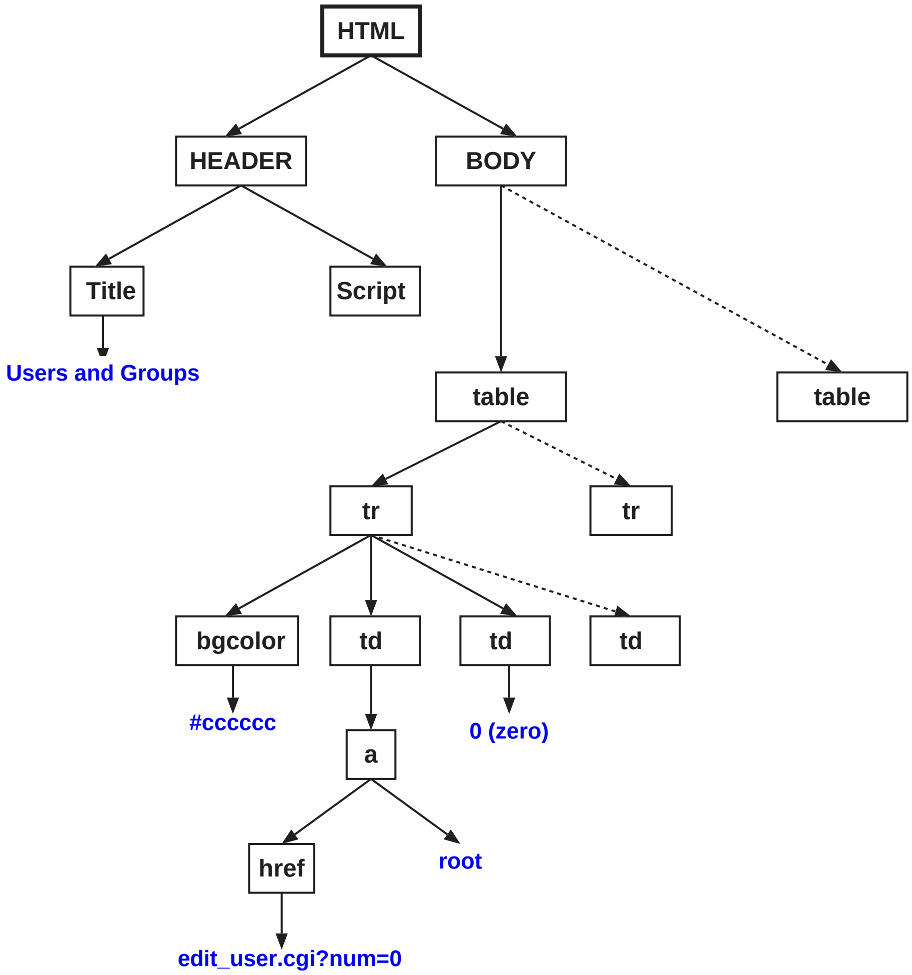

Figura 4.11: Estrutura em árvore do extracto da tabela de utilizadores

(a) Foi apenas pedido o relatório de inconsistências e o processo termina.

(b) A opção escolhida foi a actualização mas o administrador não pretende actualizar já o sistema e termina o processo.

(c) A opção escolhida foi a actualização e o administrador pretende actualizar o sistema e continuar o processo até ao final.

Este processo poderá seguir-se noutros módulos onde o Webmin disponibiliza a informação da configuração local do sistema sob a forma de tabelas HTML. Por exemplo, nos grupos UNIX e *exports* NFS.

Para efectuar o *parse* das tabelas em HTML produzidas pelo Webmin utiliza-se o módulo `HTML::TreeBuilder` [40] [14] [39] do *Perl*. É um módulo especificamente utilizado para percorrer um documento em HTML e construir uma árvore de *parse*, onde os ramos correspondem às *tags* de HTML e os nós terminais ao seu conteúdo. Existem três referências principais para percorrer a tabela criada pelo `HTML::TreeBuilder`:

- `_tag`, correspondente a um elemento definido por uma `tag`;
- `_parent`, define o elemento pai de um outro;
- `_content`, refere-se ao conteúdo de um determinado elemento, por exemplo, os nós terminais da árvore;

O processo de construção da árvore de *parse* é tão complexo quanto mais complexo for o documento em HTML. Por isso, é importante uma análise detalhada do documento e uma definição concreta da informação que se lhe pretende extrair. No caso concreto dos utilizadores, o documento gerado tem três tabelas, sendo duas delas referentes aos utilizadores e grupos UNIX respectivamente. Na tabela de utilizadores existem duas colunas particularmente importantes e que irão ser extraídas do documento; o nome do utilizador e o seu UID.

Considerando o sistema `lab1.ncc.up.pt`, a construção da árvore de *parse* para a tabela de utilizadores faz-se executando `http://lab1.ncc.up.pt:10000/useradmin/`, através do módulo LWP, ao servidor Webmin. Os passos para gerar uma árvore de *parse*, com base no documento HTML gerado pelo Webmin, são os seguintes:

1. Definir o pedido ao Webmin que criará a tabela de utilizadores.

   ```
   $str = "http://$host{'hid'}.$host{'hdom'}:$host{'hport'}/useradmin/";
   ```

2. Guardar o resultado do pedido correspondente à tabela de utilizadores do sistema.

   ```
   $res = &pedido($str, 'POST', $host{'husername'}, $host{'hpasswd'});
   ```

3. Instanciar um objecto do tipo `HTML::TreeBuilder`.

   ```
   $html_tree = HTML::TreeBuilder->new;
   ```

4. Evocar o método `parse` para construir a árvore referente à tabela.

   ```
   $html_tree = HTML::TreeBuilder->parse($res);
   ```

5. Libertar a variável utilizada para o processamento.

   ```
   $html_tree = HTML::TreeBuilder->eof;
   ```

Após a construção da árvore de *parse*, extrai-se o conteúdo de cada uma das tabelas para duas variáveis distintas. Para tal, recorre-se ao método `look_down` do módulo `HTML::TreeBuilder`.

```
my $table_users  = ($html_tree->look_down('_tag', 'table')) [1];
my $table_grupos = ($html_tree->look_down('_tag', 'table')) [2];
```

O método `look_down` é um dos mais importantes do módulo `HTML::TreeBuilder`. Permite procurar numa sub-árvore, com início num ponto especificado, os elementos que satisfazem um dado critério. O critério pode ser simples, como procurar todas as *tags* do tipo `H1` ou mais complicado, podendo incluir várias pesquisas dentro da mesma chamada `look_down`.

Por exemplo, a seguinte instrução atribui a `$h1` o primeiro elemento da árvore cuja *tag* é `H1`.

```
$h1 = $html_tree->look_down('_tag', 'h1');
```

O MetaWebmin antes de efectuar os pedidos aos servidores Webmin valida a sua viabilidade. Ou seja, se um pedido implica a criação de um utilizador num sistema, esse utilizador apenas é criado se não existir. Da mesma forma, o utilizador apenas é removido se existir. Este procedimento evita que sejam efectuados pedidos inconsequentes que aumentariam os registos de erros, bem como as inconsistências entre o MetaWebmin e os servidores Webmin. A sua implementação consiste em percorrer com o método `look_down` a tabela do Webmin correspondente à configuração local, procurando um determinado objecto. Por exemplo, a pesquisa do nome de um utilizador ou grupo UNIX num sistema (ambas referentes à posição zero das respectivas tabelas) poderia implementar-se da seguinte forma:

```
foreach my $link ( $table->look_down ('_tag', 'tr',
        sub \{not $_[0] -> look_down('_tag', 'b') }))
{
        $f1 = ($link -> look_down('_tag', 'td') ) [0];
        push ( @x, \$f1->as\_text);
}
```

Neste exemplo evoca-se o método `look_down` para pesquisar o utilizador numa sub-árvore iniciada por uma *tag* `TR`, cujos ramos não sejam uma *tag* `B`. O array `@x` guarda o conteúdo dos elementos da primeira posição das linhas seleccionadas pelo `look_down`, correspondente à posição zero do *array*. A utilização de chamadas recursivas ao método `look_down` para visitar sub-árvores permite refinar o mais possível o critério pretendido, eliminando elementos do documento HTML que não interessam para o processamento em causa. Por exemplo, neste caso eliminam-se as linhas do cabeçalho que têm o seu conteúdo a *bold*, identificadas pela utilização da *tag* `B`.

4.10 Segurança

A segurança é um aspecto importante na utilização de sistemas de administração centralizados. O MetaWebmin analisa esta questão segundo dois pontos de vista:

1. Acesso ao MetaWebmin, possibilitando a definição e gestão de vários perfis de utilização e níveis de acesso. Por exemplo, a existência de administradores para cada um dos serviços suportados, e outros para a gestão de utilizadores. Existirão ainda utilizadores com privilégios totais a todos os serviços. Além disso, poderão ainda ser criados perfis de consulta de informação no MetaWebmin, não permitindo a actualização de dados.

2. Comunicação segura entre o MetaWebmin e os sistemas UNIX da rede. As versões disponíveis do Webmin suportam o protocolo SSL, garantindo que os pedidos HTTP se efectuam através de um *socket* seguro. Embora este processo envolva um atraso no envio da informação, o mesmo garante a encriptação dos dados trocados com o servidor Webmin.

Na instalação do Webmin sugere-se ao administrador a utilização do SSL para ligações seguras com o servidor. É necessário garantir a instalação da biblioteca *OpenSSL* e do módulo de *Perl* `Net::SSLeay` nos sistemas da rede. Para tal deverá efectuar-se:

1. *Download* e instalação da biblioteca *OpenSSL* a partir de *http://www.openssl.org*.

2. *Download* e instalação do módulo de *Perl* `Net::SSLeay` disponível, por exemplo, em *http://www.cpan.org*.

Saliente-se ainda a possibilidade de activar o módulo SSL mesmo depois do Webmin já estar instalado. Para tal, basta aceder ao menu de configuração do Webmin e activar a opção relativa ao SSL.

A criação de utilizadores no MetaWebmin efectua-se ao nível do servidor *Apache*. Para o efeito deverão incluir-se as seguintes directivas no ficheiro de configuração `/etc/httpd/conf/httpd.conf` do servidor *Apache*:

```
<Directory />
     Options FollowSymLinks
     AllowOverride None
     AuthUserFile /etc/httpd/conf/passwd
     AuthType Basic
     AuthName membership
     require valid-user
</Directory>
```

O ficheiro de texto `/etc/httpd/conf/passwd` contém os pares *username* e *password* de acesso à directoria "/" do servidor *Apache*. No caso concreto do MetaWebmin, a directoria "/" corresponde à directoria `/work/META/` onde reside a página `index.html`.

A inserção de novos acessos efectua-se através do utilitário `/usr/bin/htpasswd`. Por exemplo, para adicionar o utilizador `nfsadmin` utiliza-se o comando `#htpasswd passwd nfsadmin`. Nesta fase de implementação do MetaWebmin apenas se definiu um utilizador, denominado `admin`, com privilégios totais de administração.

4.11 Manutenção do MetaWebmin

O MetaWebmin, como qualquer outra aplicação, necessita de manutenção. No entanto, se já existirem procedimentos de administração genéricos, o administrador deverá incluir o MetaWebmin nos mesmos. Algumas das tarefas que se deverão garantir são:

- *Backup* periódico da base de dados do MetaWebmin. No caso de corrupção ou destruição da base de dados deverá haver um *backup* recente para reposição.
- Procedimento de manutenção do ficheiro de erros e da tabela de eventos da base de dados do MetaWebmin. Uma sugestão consiste na execução de um *backup* diário e imediata inicialização do ficheiro e tabela. Deverá garantir-se que, durante esse período, não existem pedidos do MetaWebmin aos servidores Webmin.

Todos os aspectos não apresentados correspondem às práticas e políticas normais e de rotina da administração de uma aplicação.

Capítulo 5

Administração usando o MetaWebmin

Este capítulo não tem como objectivo a descrição pormenorizada de todas as funcionalidades desenvolvidas e não pretende ser um manual de utilização completo. Contudo, descrevem-se e ilustram-se algumas das principais tarefas efectuadas através do MetaWebmin. Essas tarefas são:

- Autenticação no servidor MetaWebmin.
- Adicionar um novo grupo de sistemas.
- Adicionar um novo sistema.
- Adicionar um utilizador e de um grupo UNIX.
- Alterar a *password* de um utilizador.
- Analisar inconsistências.
- Recuperar erros.

5.1 Acesso ao MetaWebmin

O primeiro passo consiste em aceder ao servidor MetaWebmin. Para tal, basta aceder ao porto TCP/IP onde o servidor *Apache* está em execução. Por exemplo, considerando que o MetaWebmin está instalado no sistema `mwm.ncc.up.pt` e o servidor *Apache* escuta pedidos no porto TCP/IP 80, o administrador deverá abrir um *browser* e aceder ao URL `http://mwm.ncc.up.pt:80`. Com este acesso, aparecerá a janela de autenticação para acesso ao servidor HTTP *Apache* e, consequentemente, ao MetaWebmin, conforme se ilustra na figura 5.1.

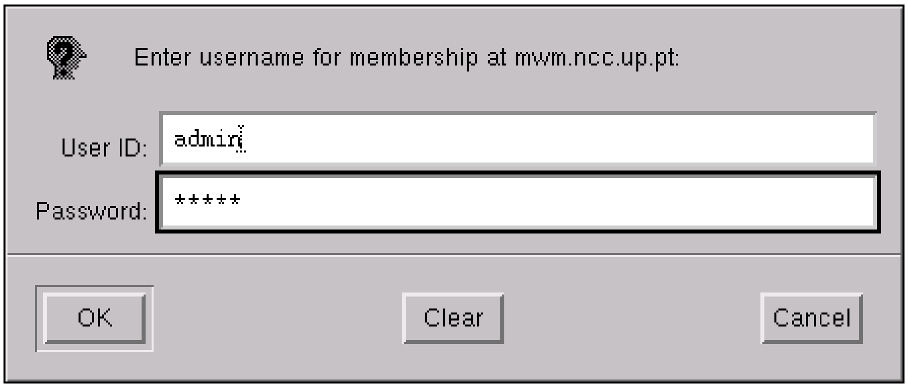

Figura 5.1: Autenticação no servidor HTTP do MetaWebmin

Se o acesso ao servidor HTTP for concedido, o utilizador acederá à página inicial do MetaWebmin. Terá então acesso a um conjunto de opções de navegação dentro do utilitário, bem como aos formulários de gestão dos serviços na rede.

5.2 Adicionar grupo de sistemas

O grupo de sistemas é um dos módulos principais do servidor MetaWebmin. Pode criar-se em duas situações distintas:

1. Através da opção `"Adicionar"` no módulo de gestão de grupos de sistemas.
2. Na activação de um sistema num grupo de sistemas.

Na página principal do módulo de gestão de grupos de sistemas apresentam-se os grupos já existentes na base de dados, conforme ilustra a figura 5.2.

Através da opção `"Novo Grupo Sistemas"` inicia-se a caracterização do novo grupo e a sua inserção no MetaWebmin, preenchendo o formulário de identificação da figura 5.3. No exemplo, o grupo em causa é o `gs4`, relativo ao Mestrado de Inteligência Artificial (IA).

Após aceitar os dados inseridos no formulário, o processo de caracterização do grupo continua com a configuração dos recursos de NFS (ver figura 5.4). Neste formulário apresentam-se todos os recursos de NFS existentes na rede, nomeadamente o sistema que presta o serviço, o nome da directoria exportada e as respectivas permissões. Se o pretender, o administrador pode ainda adicionar novos recursos de NFS através da opção `"Novo NFS"`. Através de uma *checkbox* o administrador escolhe os recursos pretendidos para fazer parte da configuração deste grupo de sistemas. Neste exemplo seleccionaram-se as directorias `/work/docMestrado/SDIST/` e `/work/scratch/` dos sistemas `lab1` e `lab2` respectivamente.

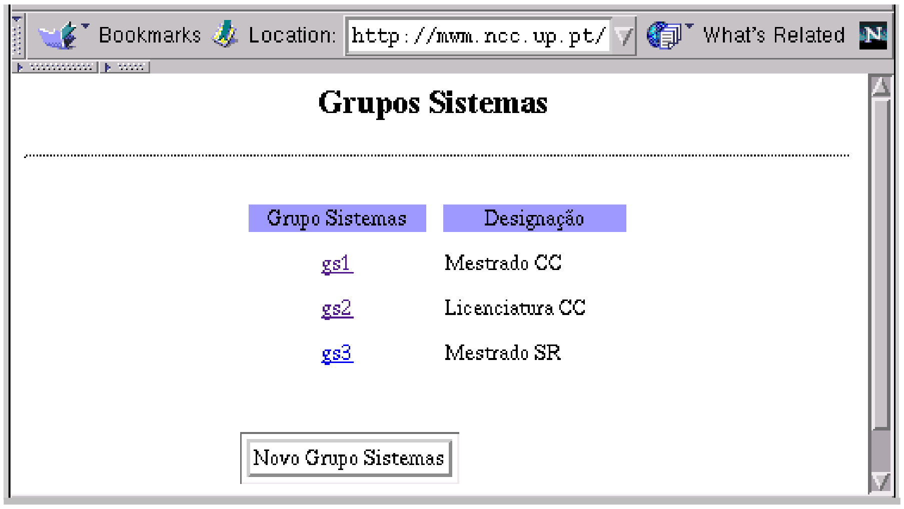

Figura 5.2: Página inicial do módulo de gestão de grupos de sistemas

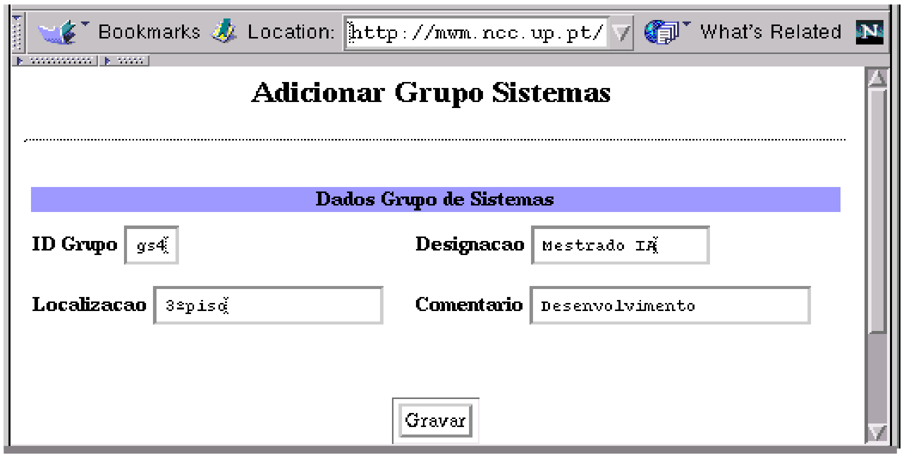

Figura 5.3: Caracterização de grupo de sistemas - formulário de identificação

Na configuração de DNS o processo é semelhante ao do NFS. A figura 5.5 ilustra as diversas opções de configuração do ficheiro `/etc/resolv.conf` para o grupo de sistemas em causa. Se pretender configurar o recurso DNS num grupo, o administrador apenas poderá escolher uma das configurações apresentadas. À semelhança do NFS é igualmente possível adicionar uma nova configuração utilizando a opção `"Novo DNS"`.

No exemplo apresentado, o ficheiro `/etc/resolv.conf` dos sistemas do grupo `gs4` terá a seguinte estrutura:

```
domain ncc.up.pt
search alfa.ncc.up.pt
search beta.ncc.up.pt
```

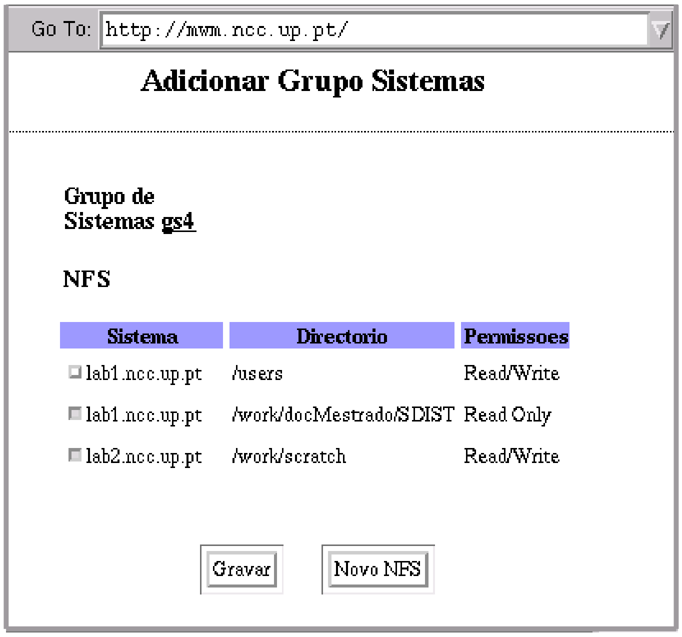

Figura 5.4: Caracterização de grupo de sistemas - recursos de NFS

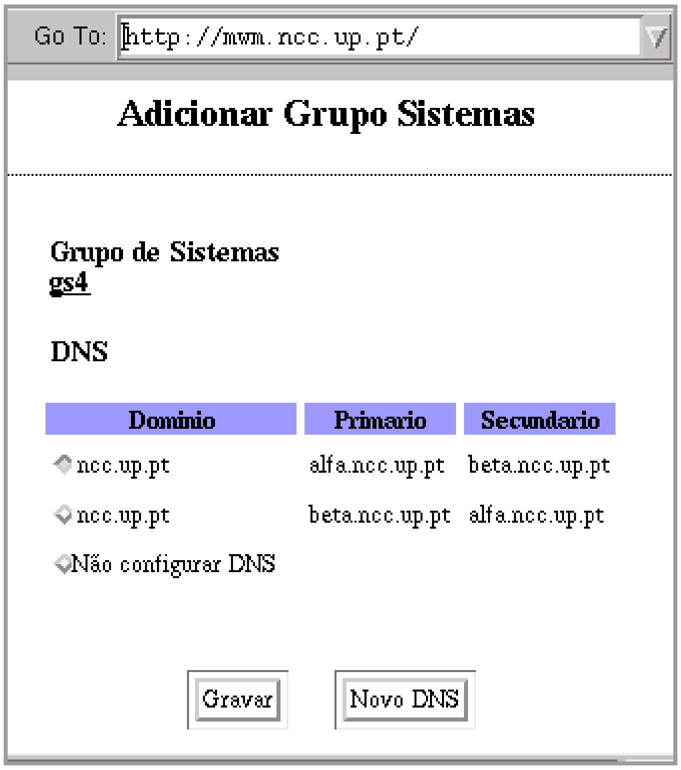

Figura 5.5: Caracterização de grupo de sistemas - recursos de DNS

Após o preenchimento dos formulários, as tabelas envolvidas são actualizadas, nomeadamente a `GSIS`, a `NFS_GSIS` e a `DNS_GSIS`. A alteração à configuração de um grupo de sistemas efectua-se seguindo os *links* correspondentes a cada grupo, na página inicial deste módulo, ilustrada na figura 5.2. A página de alteração disponibiliza ainda a remoção do grupo e a listagem detalhada da sua constituição.

5.3 Adicionar sistema

Os sistemas geridos pelo MetaWebmin têm, obrigatoriamente que existir na base de dados, apesar de poderem estar permanentemente desactivados de um grupo. Para adicionar um sistema à rede preenche-se o formulário apresentado na figura 5.6. No exemplo apresentado adiciona-se o sistema `lab3` ao MetaWebmin.

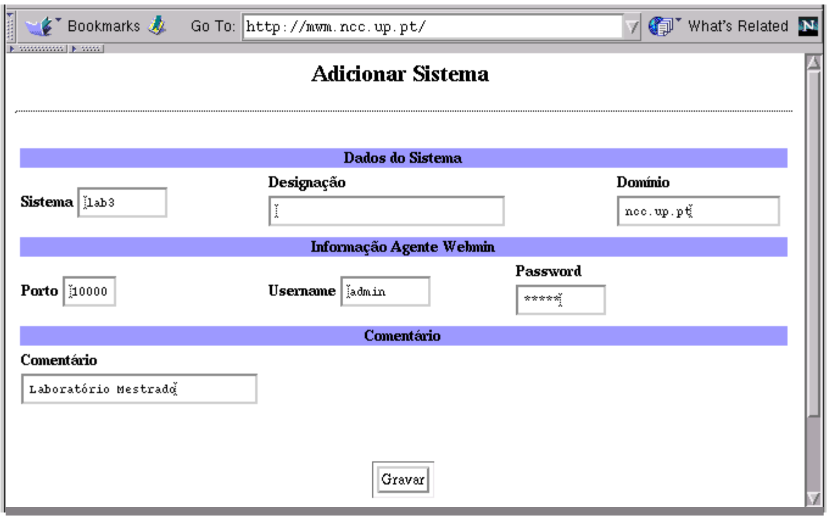

Figura 5.6: Formulário para adicionar novo sistema ao MetaWebmin

Após a inserção na base de dados, o administrador poderá activar de imediato este sistema num grupo, por exemplo, no `gs4` criado anteriormente. Esta operação ilustra-se na figura 5.7.

Activar o sistema `lab3` consiste em descarregar-lhe toda a configuração relativa a utilizadores, grupos UNIX, NFS e ao ficheiro `/etc/resolv.conf` do grupo de sistemas `gs4`. A partir deste momento o sistema está activo no grupo, o que significa que todas as alterações à configuração do grupo `gs4` reflectir-se-ão no sistema `lab3`. Por exemplo, a inserção de um utilizador no grupo de sistemas `gs4`, implica a sua criação no sistema `lab3`. Após activar o sistema `lab3`, a página inicial deste módulo toma o aspecto da figura 5.8.

Sempre que o sistema está activo num grupo, o administrador pode ainda desactivá-lo ou actualizar a informação local com a existente no MetaWebmin.

Figura 5.7: Formulário para activar um sistema num grupo de sistemas

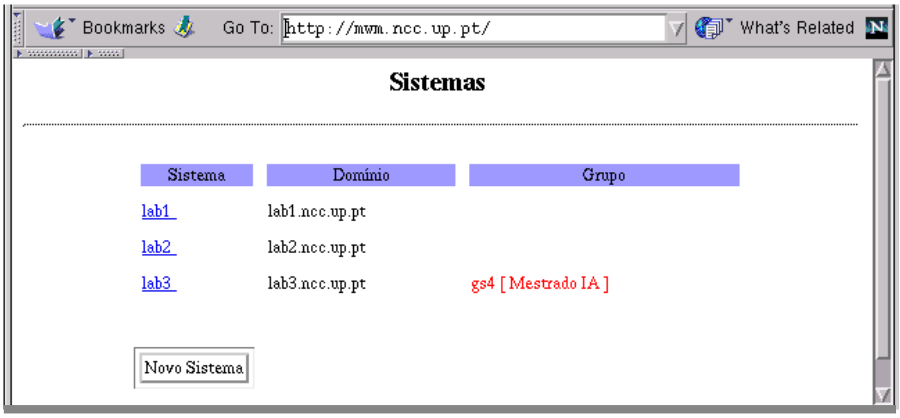

Figura 5.8: Página inicial do módulo de gestão de sistemas

5.4 Adicionar utilizador

Um utilizador gerido pelo MetaWebmin tem obrigatoriamente que existir na base de dados. Para aceder a um sistema da rede tem ainda de estar associado ao grupo desse sistema. A gestão de utilizadores e respectivos grupos UNIX faz-se pela mesma página do MetaWebmin, ilustrada pela figura 5.9.

Nesta página o administrador edita os dados de um determinado utilizador ou grupo UNIX. As opções `"Novo Utilizador"` e `"Novo Grupo UNIX"` acedem às páginas de caracterização de um novo utilizador e grupo UNIX respectivamente. A figura 5.10 ilustra o formulário de inserção de um novo utilizador no MetaWebmin.

Após aceitar os dados do formulário, o MetaWebmin atribui automaticamente um UID ao utilizador de acordo com o grupo UNIX primário seleccionado. Se o administrador pretender associar este utilizador a um grupo UNIX ou de sistemas que não existam no MetaWebmin, poderá adicioná-los sem sair da página, através das opções disponibilizadas para o efeito. Se o pretender, o administrador pode criar o utilizador num grupo de sistemas que não tenha qualquer sistema associado. Todavia, o utilizador é imediatamente criado aquando da associação de um sistema a esse

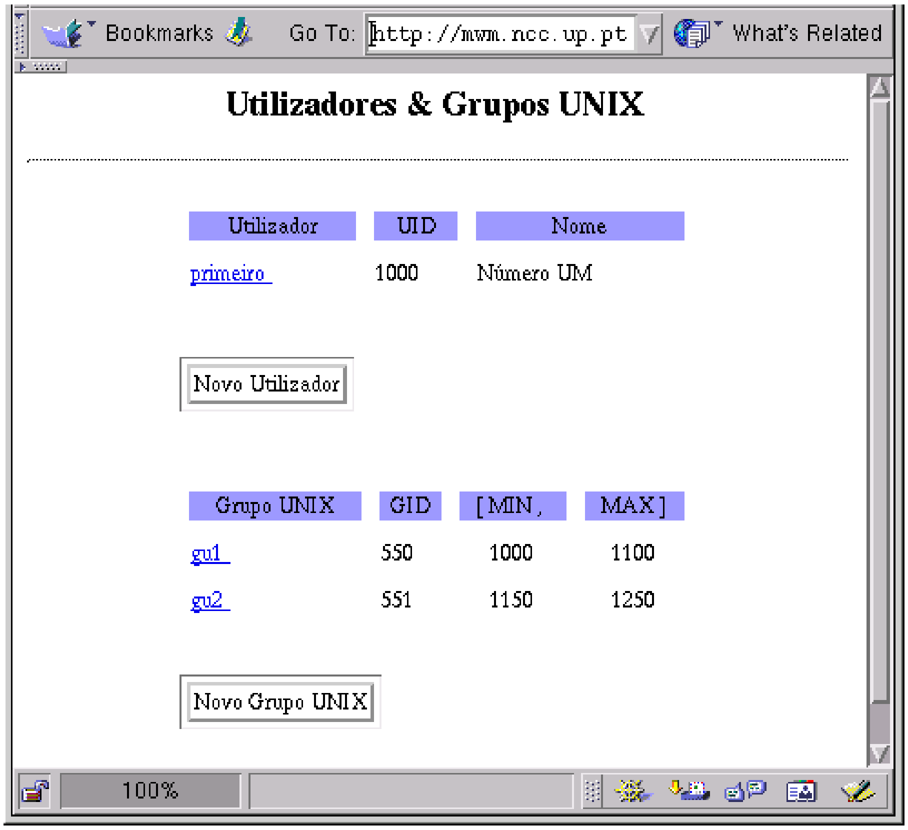

Figura 5.9: Página inicial de módulo de gestão de utilizadores e grupos UNIX

grupo de sistemas. Se os grupos UNIX seleccionados na criação de um utilizador não existirem, o MetaWebmin cria-os automaticamente nos sistemas envolvidos nos grupos de sistemas seleccionados.

A criação de um utilizador é uma das tarefas onde é notória a vantagem da utilização do MetaWebmin comparativamente com outras formas de administração. Analisemos o seguinte exemplo:

- Pretende-se criar o utilizador ''`abc`'' em trinta sistemas da rede.
- O utilizador deverá ter como grupo UNIX primário o `gu_1` e como secundários `gu_2` e `gu_3`.

Para satisfazer o pedido consideram-se duas abordagens distintas:

1. Efectuar *login* remoto a cada um dos sistemas envolvidos na operação e, utilizando um utilitário para o efeito, criar o utilizador. Poderá utilizar um comando de execução remota de comandos, por exemplo, o `rsh`.
2. Definir grupos de sistemas e criar o utilizador no MetaWebmin. Para tal, deverá preencher o formulário apresentado na figura 5.10 que, após aceitação dos valores, cria o utilizador nas trinta máquinas.

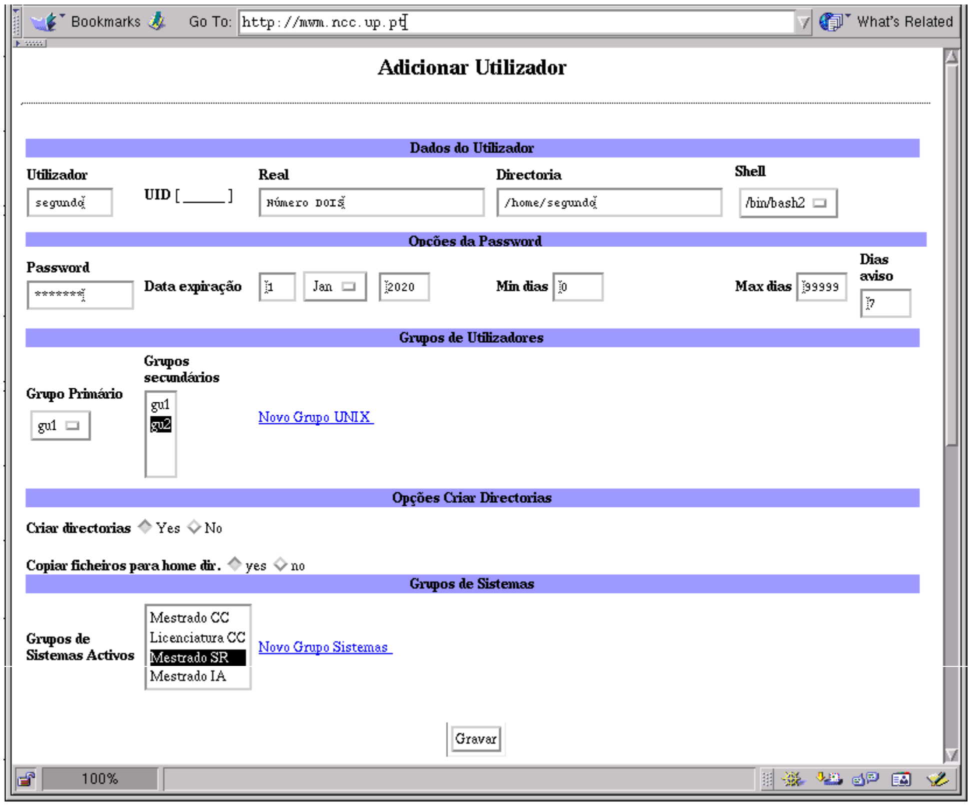

Figura 5.10: Formulário para adicionar um utilizador no MetaWebmin

Na segunda abordagem apontam-se os seguintes benefícios imediatos para o administrador:

1. Evita erros na introdução dos dados do utilizador aquando da sua criação em cada sistema. Os dados do utilizador ''`abc`'' introduzem-se uma única vez e guardados centralmente na base de dados do MetaWebmin. Atendendo a que a rede é heterogénea, na primeira abordagem o administrador pode deparar-se com diferentes plataformas UNIX. Essa situação não se verificará na segunda abordagem, na medida em que o servidor Webmin executa a operação de acordo com a plataforma UNIX onde está instalado e configurado.

2. Diminui o tempo gasto na criação do utilizador nas máquinas envolvidas na operação. Nesta abordagem eliminam-se, para cada sistema, os tempos de efectuar *login* e construir o comando (ou comandos) para a criação do utilizador. Esse tempo será substituído pelo do preenchimento único do formulário de introdução do utilizador no MetaWebmin. O tempo envolvido na execução do comando local de criação do utilizador é sensivelmente o mesmo nas duas abordagens.

3. Encadeia operações de administração colaterais à criação do utilizador. Na primeira abordagem, antes de criar o utilizador, o administrador tem que verificar se os grupos UNIX `gu_1`, `gu_2` e `gu_3` existem em cada uma das máquinas. Se

não existirem terá que os criar previamente, aumentando o número de operações a executar em cada sistema, o tempo gasto e a probabilidade de erros. Na segunda abordagem, o MetaWebmin valida a existência dos grupos UNIX e, caso não existam nas máquinas, serão previamente criados, evitando assim passos intermédios antes da criação do utilizador.

O tempo gasto diminui com o aumento do número de sistemas, tarefas intermédias necessárias para realizar uma determinada operação e respectivo grau de complexidade. Outros exemplos poderão encontrar-se nos restantes módulos, como sejam, a gestão de grupos UNIX e de serviços inter-grupais como é o caso do NFS.

5.5 Adicionar grupo UNIX

A página de criação de um grupo UNIX acede-se através da página inicial deste módulo, ilustrada na figura 5.9 ou nas opções disponíveis para o efeito nos formulários de inserção e alteração de um utilizador. O grupo UNIX caracteriza-se por um nome, um GID e um intervalo para atribuição de UIDs. O valor sugerido para o GID corresponde ao GID de valor mais elevado, existente na base de dados, mais uma unidade. Os limites sugeridos para os intervalos de UIDs baseiam-se nos valores das variáveis do MetaWebmin `gfolga` e `gincruid`, definindo esta última o número de UIDs a atribuir a cada grupo. A figura 5.11 ilustra o formulário de criação do grupo UNIX `gu3` nos grupos de sistemas `Mestrado SR` e `Mestrado IA`.

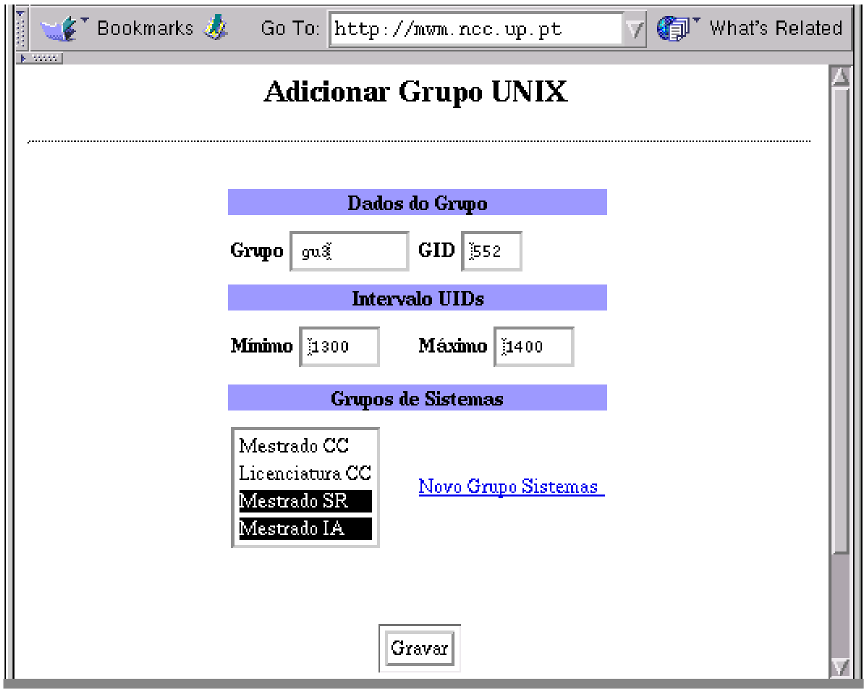

Figura 5.11: Formulário para adicionar um grupo UNIX no MetaWebmin

Na criação e alteração de um grupo UNIX valida-se ainda se os conjuntos definidos são disjuntos.

5.6 Alterar *password* de utilizador

A alteração da *password* de um utilizador efectua-se no formulário de alteração dos dados do utilizador. O administrador altera o campo da *password* com o novo valor e, após aceitar as alterações, o MetaWebmin executa as seguintes acções:

1. Altera o valor da *password* na base de dados do MetaWebmin, na tabela dos dados do utilizador.

2. Executa um pedido de actualização da *password* a todos os sistemas que pertençam aos grupos de sistemas onde o utilizador está criado no MetaWebmin.

5.7 Analisar inconsistências

Após os trabalhos de configuração do MetaWebmin e consequente activação do sistema `lab3` no grupo `Mestrado IA`, um outro utilizador que não o administrador do MetaWebmin, adicionou localmente ao sistema `lab3` o utilizador ''`segundo`''. Este tipo de acções provoca diferenças entre a configuração local de cada sistema e a centralizada no MetaWebmin. A figura 5.12 ilustra o relatório de análise de inconsistências na configuração local do sistema `lab3`.

Conforme se pode verificar, o utilizador ''`segundo`'' existe na configuração local do sistema e não na do MetaWebmin.

A análise de inconsistências deverá ser extensível a todos os serviços adicionados posteriormente ao MetaWebmin. Todavia, para simplificar o relatório, apenas se apresenta a análise de inconsistências dos utilizadores e grupos UNIX.

No sentido de automatizar e sistematizar a tarefa de análise de inconsistências, o administrador poderá optar por incluir nas rotinas diárias de gestão da rede a geração em *batch* de relatórios de diferenças para cada sistema. O administrador terá assim um conhecimento efectivo das alterações efectuadas localmente em cada sistema, podendo escalonar as respectivas actualizações.

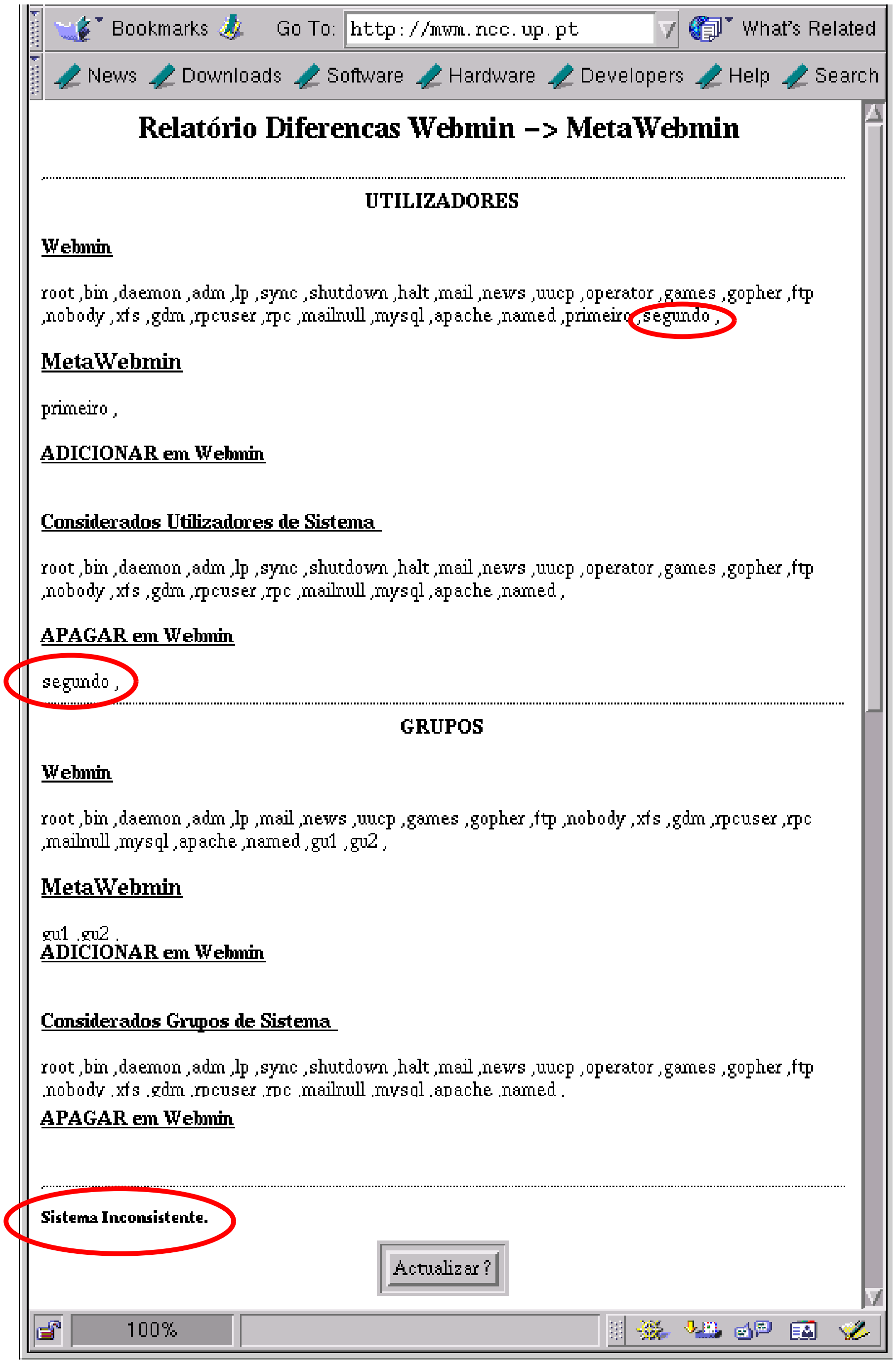

Figura 5.12: Relatório de análise de inconsistências

5.8 Recuperar falhas

O tratamento de erros pode efectuar-se logo após a execução de um pedido ou posteriormente. Após a ocorrência de um erro e respectivo registo, o administrador tem ao seu dispor duas possibilidades imediatas de correcção:

1. **Manual**, acedendo a cada sistema onde ocorreu o erro, corrigindo-o e repetindo novamente o pedido.
2. **Automática**, aceitando o processo de correcção sugerido pelo MetaWebmin e a repetição do pedido. Esta funcionalidade está disponível apenas em alguns tipos de erro.

A figura 5.13 apresenta o resultado da adição do utilizador ''`segundo`'' no grupo de sistemas `Mestrado IA`, do qual o sistema `lab3` faz parte, seleccionando a *home directory* `/home/primeiro/`. O Webmin não permite a criação de um utilizador numa directoria que já seja a *home directory* de outro. Neste caso, a directoria `/home/primeiro/` é a *home directory* do utilizador ''`primeiro`''.

O MetaWebmin disponibiliza ao administrador a possibilidade de corrigir manualmente o erro, apresentando sempre que possível uma sugestão. Possibilita uma nova execução do pedido e a alteração dos dados do utilizador. Neste caso, a alteração da *home directory*. Como se trata de um erro onde a correcção automática está disponível, o administrador pode aceitar a sugestão do MetaWebmin no sistema `lab3`.

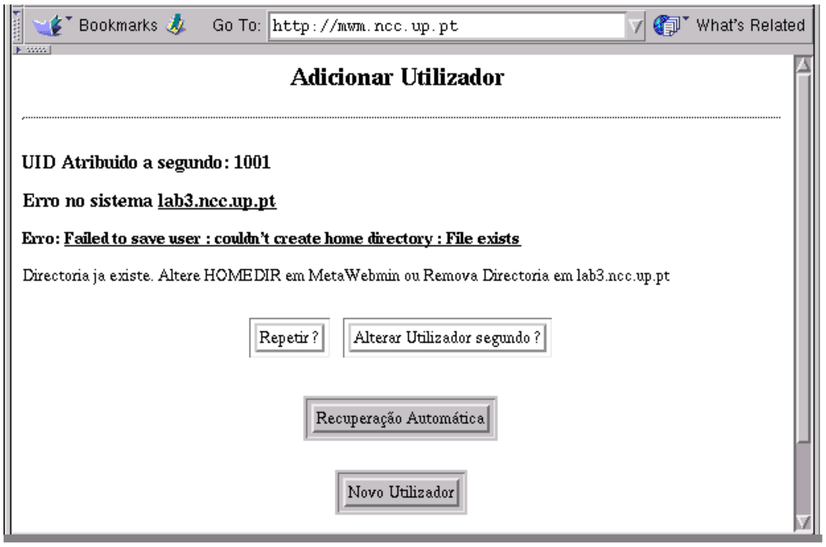

Figura 5.13: Exemplo de uma falha no pedido de criação de um utilizador no Webmin

Se não pretender a correcção imediata do erro e preferir uma análise e acção mais elaboradas, o administrador pode optar pela gestão global de erros, conforme se ilustra na figura 5.14.

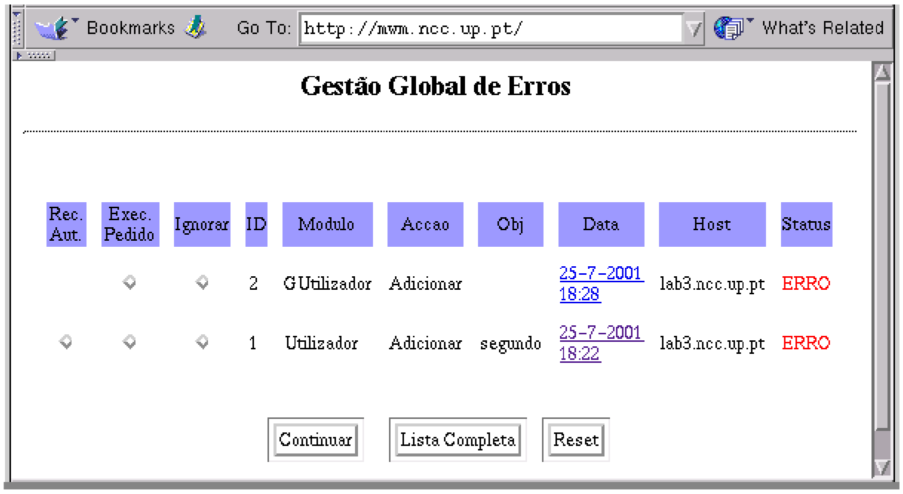

Figura 5.14: Gestão global de erros

A identificação dos eventos corresponde a um número sequencial. Em cada evento regista-se a data e hora em que ocorreu, qual o objecto em causa, o módulo, o sistema e o *status* actual. Apenas é possível efectuar uma acção em eventos cujo valor do *status* seja **ERRO**. As acções possíveis são:

1. Evocar a recuperação automática, se estiver disponível, ou seja, se existir uma *checkbox* para o erro.

2. Executar o pedido HTTP em erro, pressupondo que o administrador já corrigiu manualmente o erro e há condições para repetir o pedido.

3. Ignorar o erro, alterando o seu *status* para **OK - IGNORAR**.

Para aceder à listagem do histórico de todos os eventos existentes na base de dados utiliza-se a opção ''`Lista Completa`''.

Seguindo o *link* do campo da data de ocorrência é possível aceder ao detalhe do erro. A informação adicional apresentada corresponde ao pedido completo de HTTP em erro.

Capítulo 6

Conclusões

Inicialmente, aquando do estudo desta arquitectura, existia pouca informação sobre como e em quanto tempo seria possível implementar o modelo apresentado. Neste momento é possível afirmar que o desafio proposto, e aceite à partida, sem hesitações e consciente das dificuldades, foi ganho.

O desenvolvimento do MetaWebmin permitiu a análise e estudo de vários sistemas de administração centralizada para arquitecturas UNIX. Nessa análise identificaram-se as características de funcionamento desejáveis num utilitário de administração centralizado e implementadas no MetaWebmin. Todavia, em todos os utilitários analisados, não foi encontrado um que permitisse organizar os sistemas segundo determinadas características ou funções. Ou seja, abstraísse o administrador dos sistemas existentes da rede, nomeadamente da sua localização e actividade que desempenham. O MetaWebmin introduz esse conceito através da gestão de grupos de sistemas UNIX. Associado a este conceito implementa a gestão inter-grupal de recursos da já referida rede. Estas duas características do MetaWebmin marcam a diferença relativamente aos utilitários analisados, dando uma visão diferente ao administrador sobre a forma de gerir a rede e organizar os seus sistemas.

O trabalho desenvolvido pretende, acima de tudo, apresentar e validar um modelo de administração de sistemas e redes. Desse ponto de vista, conclui-se que:

- A utilização do servidor Webmin nos sistemas remotos e o MetaWebmin a coordenar os pedidos do administrador foi bem sucedida. O Webmin é um utilitário bastante estável e a versão utilizada existe para um vasto número de plataformas UNIX.

- A tecnologia utilizada nos diversos componentes da arquitectura constituiu, de um modo geral, uma aposta ganha.

- A centralização da configuração e dos pedidos no MetaWebmin permite um melhor conhecimento da rede e da sua configuração. Pode ainda ser um factor importante na prevenção de erros na configuração de novas máquinas na rede.

Em organizações de média e grande dimensão, onde a rede está geograficamente dispersa, este conceito poderá contribuir para a centralização das competências técnicas em administração de sistemas e redes.

O elemento diferenciador do MetaWebmin dos outros sistemas analisados consiste na possibilidade de gerir grupos de sistemas heterogéneos, agrupando as máquinas de acordo com parâmetros comuns e, entre elas, configurar recursos de rede numa perspectiva Cliente/Servidor. Assim sendo, conclui-se que a implementação destes dois conceitos diferenciadores:

- Permite organizar logicamente as máquinas na rede. Do ponto de vista do administrador torna-se mais fácil conhecer por exemplo, o enquadramento funcional de uma máquina na rede, que tipo de configuração de um determinado serviço lhe deve ser definida ou quais os utilizadores que devem ser criados.
- Facilita o trabalho de configuração de recursos da rede, nas suas componentes cliente e servidor. Permite uma maior flexibilidade na configuração desses serviços, podendo as máquinas de cada componente pertencer a grupos de sistemas diferentes.
- Diminui a possibilidade de ocorrerem erros na configuração de uma máquina. Os dados são introduzidos e guardados num único ponto da rede, evitando a sua duplicação e consequente incoerência. Assegura-se ainda que os pedidos HTTP às diversas máquinas são uniformes e coerentes com a informação guardada centralmente no MetaWebmin.
- Diminui o tempo gasto nalgumas tarefas de configuração de um sistema na rede, em grande parte devido à eliminação de tarefas repetitivas.
- Encadeia, sempre que possível, tarefas colaterais, evitando a execução separada de tarefas intermédias. Por exemplo, se a criação de um utilizador implicar a criação prévia de um grupo UNIX, o MetaWebmin assegura essa sequência e encadeamento de tarefas.

O trabalho desenvolvido contempla apenas a administração de alguns serviços, não sendo por isso um produto acabado. Todavia, conclui-se que os objectivos inicialmente propostos para este trabalho foram alcançados e o modelo e conceitos apresentados justificam o seu alargamento a outros serviços e recursos de administração de sistemas e redes.

Durante o desenvolvimento do MetaWebmin surgiram diversos assuntos relacionados com esta temática que foram alvo de investigação e análise. Alguns são apresentados de seguida e constituem ideias para um trabalho futuro no MetaWebmin.

- **Converter os dados do Webmin para XML [1] [15] [42].**
A linguagem XML é a sucessora do HTML. O XML disponibiliza uma análise semântica dos documentos *Web*, legível tanto pelo ser humano como pela máquina. O MetaWebmin pressupõe que os servidores Webmin dos sistemas da rede estejam na sua versão 0.83. O formato de comunicação HTTP entre o MetaWebmin e os servidores Webmin corresponde apenas a esta versão de Webmin. Ou seja, se for instalada uma versão diferente [2], a arquitectura do MetaWebmin apresentada já não funciona. Torna-se assim numa limitação importante na arquitectura do MetaWebmin, podendo minimizar-se pela tradução do Webmin para XML e alteração do MetaWebmin para interpretar XML.

- **Utilizar o protocolo SOAP [6] [42].**
Utilização de um ambiente descentralizado e distribuído para efectuar os pedidos HTTP entre o MetaWebmin e os servidores Webmin.

- **Incluir serviços de instalação remota de máquinas.**
Na arquitectura actual pressupõe-se que as máquinas para serem geridas pelo MetaWebmin, tenham previamente configurado um sistema operativo UNIX e correspondente versão de Webmin. O objectivo consiste em incluir nas funcionalidades do MetaWebmin processos de instalação remota de máquinas, necessitando apenas de ter uma ligação física à rede. Será assim possível instalar integralmente uma máquina, incluindo o sistema operativo.

- **Incluir a administração de servidores aplicacionais.**
Uma funcionalidade bastante útil para o MetaWebmin consiste em incluir a centralização da configuração de servidores, por exemplo, a configuração dos servidores de *mail* e de DNS da rede. Trata-se de uma funcionalidade importante se pensarmos, por exemplo, que um destes servidor pode avariar e há necessidade de replicar a sua configuração e a do serviço que executa noutra máquina.

- **Utilizar um SGBD mais robusto.**
O SGBD utilizado nesta fase foi o MySQL. Trata-se de um sistema muito simples e, consequentemente, limitado nas opções que disponibiliza ao nível de utilização e administração. A proposta consiste em utilizar um SGBD mais evoluído que resolva as lacunas do MySQL. Duas alternativas são o Oracle ou o Postgres.

Pelo exposto, é consciente afirmar que há, de facto, matéria de investigação e desenvolvimento futuro para o MetaWebmin. Pretende-se dar continuidade ao trabalho efectuado até ao momento, com o objectivo central de simplificar cada vez mais a função de administração de sistemas UNIX e redes.

[1] *XML: eXtended Markup Language*

[2] Anterior ou posterior. Actualmente o Webmin já existe na versão 0.87, desde 26 de Junho de 2001

Bibliografia

[1] Descartes A. and Bunce T. *Programming the Perl DBI.* O'Reilly & Associates, 1st edition, 2000.

[2] Reed A. *Implementing Directory Services (Enterprise Computing).* McGraw-Hill, book and cdrom edition, 2000.

[3] Laurie B. and Laurie P. *Apache: The Definitive Guide.* O'Reilly & Associates, 2nd edition, February 1999.

[4] DiBona C., Ockman S., and Stone M. *Open Sources : Voices from the Open Source Revolution.* O'Reilly & Associates, 1st edition, 1999.

[5] Blank-Edelman D. *Perl for System Administration.* O'Reilly & Associates, 1st edition, 2000.

[6] Box D., Ehnebuske D., Kakivaya G., Layman A., Mendelson N., Nielsen H., Thatte S., and Winer D. *Simple Object Access Protocol (SOAP) 1.1* . Disponível em http://www.w3.org/TR/SOAP/, May 2000.

[7] Apache-SSL Documentation. *Apache-SSL Documentation.* http://www.apache-ssl.org/docs.html, 2001.

[8] Aas G. and Koster M. *lwpcook - libwww-perl cookbook.* Disponível em http://www.sn.no/libwww-perl, 1997.

[9] Stern H., Eisler M., and Labiaga R. *Managing NFS and NIS.* O'Reilly & Associates, 2nd edition, 2001.

[10] Abbey J. *The Group Administration Shell and the GASH Network Computing Environment.* Proceedings of the 8th USENIX System Administration Conference, 1994.

[11] Franks J., Hallam-Baker P., Hostetler J., Leach P., Luotonen A., Sink E., and Stewart L. *RFC2069 - An Extension to HTTP.* Disponível em http://www.ietf.org/rfc.html, 1997.

[12] Franks J., Hallam-Baker P., Hostetler J., Lawrence S., Leach P., Luotonen A., and Stewart L. *RFC2617 - HTTP Authentication.* Disponível em http://www.ietf.org/rfc.html, 1999.

[13] Coar K. *The WWW Common Gateway Interface - Version 1.1.* Disponível em http://cgi-spec.golux.com/draft-coar-cgi-v11-03-clean.html, 1999.

[14] MacFarlane K. *Parsing HTML with HTML::Parser.* The Perl Journal #18, 2000.

[15] Rohit K. and Adam R. *X Marks the Spot.* Disponível em http.//www.cs.caltech.edu/~adam/papers/xml/x-marks-the-spot.html, 1997.

[16] Stein L. *CGI.pm - a PERL5 CGI Library, version 2.752.* Disponível em http://stein.cshl.org/WWW/software/CGI/, 2 March 2001.

[17] Stein L. *Network Programming with Perl.* Adisson Wesley, 1st edition, 2000.

[18] Wall L., Christiansen T., and Orwant J. *Programming Perl.* O'Reilly & Associates, 3rd edition, 2000.

[19] Burgess M. *Principles of Network and System Administration.* John Wiley & Sons, Ltd, 1st edition, 2000.

[20] Harlander M. *Central System Administration in a Heterogeneous Unix Environment: GeNUAdmin.* USENIX System Administration Conference (LISA), 1994.

[21] Rosenstein M., Geer D., and Levine P. *The Athena Service Management Service.* Winter USENIX, 1988.

[22] Wilcox M. *Implementing LDAP.* McGrawHill, 1st edition, 1999.

[23] IBM Manuals. *PSSP Administration Guide - Document number: SA22-7348-01.* IBM.

[24] IBM Manuals. *System Management Concepts - Operating Systems and devices.* IBM, 2nd edition, 2000.

[25] IBM Manuals. *Distributed SMIT 2.2 for AIX - Guide and Reference.* IBM, 3nd edition, June 1995.

[26] IBM Manuals. *RS/6000 Scalable POWERparallel System: Scientific and Technical Computing Overview.* IBM, October 1996.

[27] GeNUA mbH. *GeNUA mbH - Home Page.* http://www.genua.de/.

[28] Comprehensive Perl Active Network. *CPAN's Home Page.* http://www.cpan.org.

[29] Tim Niemueller. *Webmin Module Writers Help.* http://www.niemueller.de/webmin/writers-help/index.html.

[30] Hewlett Packard. *HP-UX System Administration Tasks - HP9000 Series 700/800.* Hewlett Packard, 1995.

[31] Clinton Pierce. *The Igor System Administration Tool.* Proceedings of the 10th USENIX System Administration Conference, 1996.

[32] Fielding R., Gettys J., and Mogul J. *RFC2616 - Hypertext Transfer Protocol - HTTP/1.1.* Disponível em http://www.ietf.org/rfc.html, 1999.

[33] Fielding R., Gettys J., Mogul J., Frystyk H., and Berners-Lee T. *RFC2068 - Hypertext Transport Protocol - HTTP/1.1.* Disponível em http://www.ietf.org/rfc.html, 1997.

[34] Finkel R. and Sturgill B. *Tools for System Administration in a Heterogeneous System.* Proceedings of Workshop on Large Installation System Administration III - USENIX, 1989.

[35] Finkel R. and Sturgill B. *SAT - System Administrator's Manual.* Internal report - University of Kentucky, Department of Computer Science, 2000.

[36] Finkel R., Sturgill B., and Stenn H. *Experience with a System-Administration Tool.* Software Practice and Experience v.29 p.953-971; Disponível em http://dblp.uni-trier.de/db/journals/spe/spe29.html, 1999.

[37] Ramakrishnan R. *Database Management Systems.* McGrawHill, 2nd edition, 1998.

[38] Yarger R., Reese G., and King T. *MySQL and mSQL.* O'Reilly & Associates, 1st edition, 1999.

[39] Burke S. *Scanning HTML.* The Perl Journal #19, 2000.

[40] Burke S. *Trees.* The Perl Journal #18, 2000.

[41] Srinivasan S. *Advanced Perl Programming.* O'Reilly & Associates, 1st edition, 1997.

[42] St.Laurent S., Johnston J., and Dumbill E. *Programming Web Services with XML-RPC.* O'Reilly & Associates, 1st edition, June 2001.

[43] Caldera Systems. *eDesktop Users's Guide - Using Webmin.* http://www.calderasystems.com/edesktop/usersguide/ch10.html.

[44] Miller T., Stirlen C., and Nemeth E. *satool - A System Administrator's Cockpit, An Implementation.* Proceedings of the 7th USENIX System Administration Conference (LISA VII), 1993.

[45] Gregory Thomas, James Schroeder, Merrilee Orcutt, Desiree Johnson, Jeffrey Simmelink, and John Moore. *UNIX Host Administration in a Hetrogeneous Distributed Computing Environment.* Proceedings of the 10th USENIX System Administration Conference, 1996.

[46] WEBMIN. *Webmin's Home Page.* http://www.webmin.com/webmin/.

[47] Leihai Xu and Finkel R. *System Administration Tools in PERL.* Master Project by Leihai Xu, University of Kentucky, Department of Computer Science, 1999.