

# **GESTOR E AUTOMATIZADOR PARA EQUIPAMENTOS CONFIGURÁVEIS POR LINHA DE COMANDOS**

**Pedro Cunha**

Licenciatura em Engenharia Informática,  
Universidade Lusófona de Humanidades  
e Tecnologias, Lisboa

**Data:** Julho 2011

**Relatório Técnico**

## INDICE

### **1 Introdução**

1.1 Contexto. ....	3
1.2 Proposta. ....	3

### **2 Método**

2. Planeamento, Requisitos, Desenvolvimento e Testes. ....	4
2.1 Planeamento e Gestão de Projecto. ....	4
2.2 Requisitos e Especificações. ....	5
2.3 Desenho de Software. ....	9
2.4 Validação e Testes. ....	14

### **3 Conclusões**

## **Abstract**

Nowadays, the vast majority of business communications equipment is configurable by command line (CLI). In certain product ranges, almost all the interaction with the unit is made in this particular way. It is especially in this business context, that arises the need to automate certain tasks performed massively and in a repetitive execution.

This work aimed to obtain an intuitive tool for minimally manage but fully automate tasks to interact with equipment or equipment groups that are configurable by command line. This solution should alleviate human resources in a way that they can dedicate their time and effort to more complex or motivating tasks, thus improving skills and optimizing time.

## **INTRODUÇÃO**

### **1.1 Contexto.**

Em qualquer grande operador de TI, uma boa gestão dos recursos técnicos e parque de equipamentos toma relevante importância, não só pela perspectiva da optimização e consequente rentabilização dos recursos e investimentos feitos, mas também porque de certa forma, a área e infra-estrutura técnica representa o núcleo duro sob o qual assentam todos os restantes serviços fornecidos pelas empresas de TI. A área técnica representa também uma parte da imagem da empresa de TI, na forma como esta se comporta.

Um sistema mal gerido ou uma infra-estrutura de comunicações que funciona de forma deficiente, não pode oferecer condições desejáveis à satisfação dos clientes nem fazer cumprir as suas expectativas. Se a rede de um operador funciona mal, torna-se mais difícil convencer os clientes a aderir aos serviços oferecidos, e seguramente, facilita-se eventuais fugas de clientes para a concorrência.

Parte significativa do tempo e recursos de uma equipa técnica é gasto na manutenção, configuração e eventual reconfiguração de equipamentos instalados e geridos. Tais tarefas são muitas vezes repetitivas e devem ser aplicadas quase simultaneamente a grandes quantidades de equipamentos ou dispositivos. Por vezes a tarefa em si é simples e pouco variável nas suas iterações, mas o facto de ter de ser feita em grandes quantidades e de forma repetida, desvia de tarefas mais importantes preciosos recursos humanos, que poderiam ser assignados a tarefas mais complexas de implementação, diagnóstico ou aprendizagem e que valorizam e tornam mais produtivos os elementos das equipas técnicas.

### **1.2 Proposta.**

A necessidade de automatizar certas tarefas, garantindo simultaneamente o normal decorrer das operações foram alguns dos motivos que impulsionaram o desenvolvimento de uma aplicação simples e intuitiva que permitisse minimamente identificar, tipificar e gerir parques de equipamentos, mas que em determinados contextos permitisse de forma completamente automática e controlada o envio de instruções, comandos, configurações completas ou parciais a equipamentos

configuráveis por linha de comandos. A escolha deste projecto faz ainda mais sentido, porque profissionalmente sou há vários anos responsável pela planificação, implementação, validação e manutenção de sistemas de grande dimensão com recurso a equipamentos configuráveis por linha de comandos.

Pretendeu-se assim criar um sistema que permitisse interagir com equipamentos, incluindo o envio ou a recolha de configurações ou partes de configurações de equipamentos geríveis por acesso IP (Telnet ou SSH). Não se trata propriamente de um emulador de terminal, mas sim uma ferramenta que permita a automatização de determinadas tarefas, principalmente aquelas que devem ser aplicadas em grandes quantidades de equipamentos em simultâneo.

Mesmo sabendo que existem determinadas aplicações que permitem a monitorização e eventualmente a gestão remota dos equipamentos, estas são normalmente proprietárias, limitadas na versatilidade e tipicamente pouco usadas pelas equipas técnicas de implementação e manutenção de projectos.

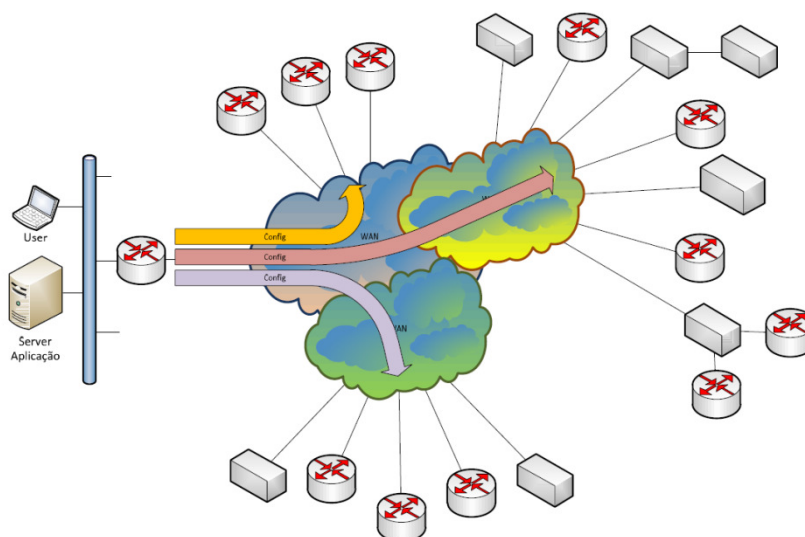


Fig.1 – Exemplo de Cenário Simplificado de Execução.

## 1. MÉTODO

### 2. Planeamento, Requisitos, Desenvolvimento e Testes.

Nas páginas seguintes serão detalhadas de forma não exaustiva as particularidades mais pertinentes do projecto desenvolvido. A descrição servirá principalmente para traçar as linhas orientativas utilizadas ao longo do projecto.

#### 2.1 Planeamento e Gestão de Projecto.

##### 2.1.1 Elementos a Entregar.

No final do prazo definido para a entrega dos elementos do projecto, deverão estar disponíveis para entrega um protótipo funcional da aplicação bem como o código fonte desenvolvido. Deverá ainda ser entregue este relatório em formato papel e electrónico.

### 2.1.2 Processo de Desenvolvimento.

O modelo de desenvolvimento escolhido mistura aspectos do modelo Rapid Application Development e do modelo Waterfall. Sendo um projecto para um tempo de execução relativamente curto considerou-se que estes são os modelos que mais se adequavam ao projecto. Apesar de ser de relativamente baixa complexidade ainda permite uma prototipagem rápida, sendo este um elemento mandatório no projecto. Por se tratar de um projecto individual, todas as fases e tarefas tiveram responsabilidade única.

### 2.1.3 Ferramentas e Técnicas.

A aplicação foi desenvolvida em Java com recurso ao IDE Netbeans, pelas ferramentas que disponibiliza e pela forma como permite integrar os diversos elementos. A necessidade da utilização do java surge na sequência de um dos requisitos que era garantir o funcionamento de forma completamente agnóstica em relação ao sistema operativo onde seria executado. Assim, desde que exista uma plataforma de suporte Java instalada no sistema operativo em execução, a aplicação deverá desempenhar as suas tarefas sem problemas ou restrições.

Para o desenvolvimento foram utilizadas classes nativas do java e em dois casos muito específicos classes “open-source” disponibilizadas gratuitamente pelos seus fabricantes.

Numa primeira fase foram tratados os aspectos gráficos, de disponibilização e apresentação da informação sendo posteriormente tratadas as questões relacionadas com os processos internos de funcionamento e interligação entre os elementos no código.

Ficou decidida a não utilização de um sistema de gestão de bases de dados como ferramenta necessária à execução da aplicação, primeiro porque se pretendia uma ferramenta que fosse portátil e funcionasse de forma autónoma, em segundo porque não traria benefício evidente à performance e ao desempenho da função principal da aplicação (automatização da interacção com equipamentos), e finalmente porque era possível cumprir os requisitos e atingir os objectivos propostos através de outros meios mais simples e sem descurar a fiabilidade ou afectar a prestação da aplicação durante a execução.

### 2.1.4 Gestão de Riscos, Esforço e Configurações

Os riscos previstos para o processo de desenvolvimento deste projecto eram mínimos. A quantidade de stakeholders era reduzida e os requisitos necessários ficaram claramente definidos. A solução assenta sobre aplicações robustas e já validadas em produção, principalmente a linguagem de programação escolhida. O desenvolvimento foi feito numa linguagem de programação já conhecida e experimentada. Este facto reduz significativamente o tempo de desenvolvimento e a probabilidade da ocorrência de atrasos.

### 2.1.5 Prazos.

O prazo de entrega estava definido desde o início como sendo o final do segundo semestre lectivo (normalmente até 29 Julho), devendo este prazo ser cumprido.

## 2.2 Requisitos e Especificações.

Tratando-se de uma aplicação principalmente para uso interno da empresa, uma pequena parcela dos requisitos foram obtidos por entrevista informal a responsáveis técnicos e a outros departamentos por si designados, e que poderiam estar envolvidos na utilização do sistema.

Grande parte dos requisitos tem como origem a experiência e necessidades próprias, uma vez que profissionalmente sou também responsável pela planificação, implementação, validação e manutenção de sistemas de grande dimensão com recurso a equipamentos configuráveis por linha de comandos.

### 2.2.1 Requisitos Externos.

Descrição do Requisito	Importância	Disponível
A aplicação terá interface gráfica. Todavia, é obrigatória a utilização teclado para preencher os campos com os dados necessários ao normal funcionamento. A utilização de rato facilita a navegação pelos diversos elementos da aplicação.	Elevada	1ª entrega
Deverá permitir o funcionamento em multi-plataforma, não devendo estar restringida pelo tipo de sistema operativo do dispositivo. Deverá no mínimo suportar Windows e sistemas operativos baseados em Linux.	Elevada	1ª entrega
Obrigatório o suporte para Ipv4	Elevada	1ª entrega
Suporte para “dual-stack” (Ipv4 + Ipv6).	Baixa	A anunciar
A aplicação deverá interagir com os equipamentos usando sessões telnet e se possível preparar caminho para suporte ssh.	Elevada	1ª entrega

### 2.2.2 Requisitos de Funcionalidades.

Descrição do Requisito	Importância	Disponível
O principal objectivo desta aplicação é a automatização das interacções com equipamentos. Por isso, é necessária identificação e tipificação dos mesmos e de algumas das suas particularidades técnicas, mas também do contexto	Elevada	1ª entrega

em que estão inseridos. Deve ser isso necessário manter de forma não volátil informação acerca de cada um dos equipamentos, dos grupos ou projectos em que se inserem e das tarefas a executar.		
Paralelamente à introdução manual da informação, deixar ficando em aberto a possibilidade futura de introdução automática de registos de equipamentos.	Média	A anunciar
Introdução, alteração e remoção de registos.	Elevada	1ª entrega
Deve ficar garantida a unicidade dos registos, devendo ser negada a introdução de elementos repetidos.	Elevada	1ª entrega
Verificação e validação de valores introduzidos, principalmente datas, endereços IP ou subnets.	Elevada	1ª entrega
Possibilidade de apresentar listagens dos registos e impressão das mesmas.	Elevada	1ª entrega
Possibilidade de exportar listas de registos para outros formatos (csv, sql, outros).	Baixa	A anunciar
Possibilidade de guardar ou restaurar cópias de segurança dos dados e do estado da execução das tarefas em agendamento.	Elevada	1ª entrega
Deverá existir um registo de eventos genérico que servirá de indicador para ocorrências relacionadas com o funcionamento da aplicação. Tenhamos por exemplo situações em que a aplicação foi parada ou iniciada, indicação da execução das tarefas ou outros eventos. Este registo(log) deverá também ele ser permanente mesmo em caso de falha da aplicação, devendo der recuperado quando esta for reiniciada.	Elevada	1ª entrega
Em caso de falha da aplicação, as tarefas pendentes e ainda não executadas devem ser retomadas ou reagendadas de imediato assim que a aplicação esteja novamente em execução.	Elevada	1ª entrega
Possibilidade de cancelar e remover tarefas pendentes de execução.	Elevada	1ª entrega
GUI em Português com possibilidade de apresentação em Inglês.	Elevada	Inglês A Anunciar
Cada tarefa depois de executada deve gerar um ficheiro log com o input e output gerado durante a execução da tarefa.	Elevada	1ª entrega

### 2.2.3 Atributos do Sistema.

Descrição do Requisito	Importância	Disponível
Garantir o funcionamento de forma completamente agnóstica em relação ao sistema operativo onde seja executado. Obrigatório funcionamento em Microsoft Windows e SO's baseados em Linux.	Elevada	1ª entrega
Aplicação deve garantir o tratamento de excepções e suportar eventuais erros dos utilizadores no que toca à introdução de dados.	Elevada	1ª entrega
Em situações normais a aplicação deve funcionar de forma ininterrupta e autónoma.	Elevada	1ª entrega
Deve suportar execução de várias tarefas em simultâneo, sem que haja degradação significativa de performance ou indisponibilidade de serviço.	Elevada	1ª entrega
Não pode de forma alguma afectar o funcionamento de outras aplicações ou do sistema em que estiver inserido.	Elevada	1ª entrega
Cópias de segurança e ficheiros guardados em disco não podem ser editáveis ou manipuláveis excepto pela própria aplicação.	Elevada	1ª entrega
Palavras-chave introduzidas nunca devem ser visíveis ou de certa forma passíveis de ser obtidas a partir do momento que são introduzidas.	Elevada	1ª entrega
Ficheiros gravados pela aplicação devem poder ser eliminados em caso de necessidade, permitindo a cópia ou distribuição da aplicação sem que sejam necessariamente entregues os dados nela contidos.	Elevada	1ª entrega
Código deve ser gerado de forma a poder existir reutilização, e deve permitir actualizações ou implementação de novas funcionalidades com o mínimo de esforço possível.	Média	1ª entrega

### 2.2.4 Suporte de Dados.

Os dados introduzidos, utilizados e arquivados pela aplicação são registados em ficheiros externos comprimidos. Pelas razões já descritas anteriormente, foi posta de parte a utilização de um sistema de gestão de bases de dados como ferramenta necessária à execução da aplicação. Estes ficheiros podem ser transportados ou eliminados sem prejuízo para o funcionamento da aplicação. Apenas contemplam os dados introduzidos pelos utilizadores. Este modelo garante melhores condições para a instalação e distribuição da aplicação, uma vez que o conteúdo está dissociado da aplicação em si.



## 2.3 Desenho de Software.

### 2.3.1 Arquitectura de Sistema.

O sistema desenvolvido pode ser decomposto em várias porções ou subsistemas menores. Cada um destes subsistemas foi desenvolvido para desempenhar tarefas específicas na aplicação e nas diferentes fases da sua utilização. Existem ainda vários micro-sistemas cujo objectivo é prestar pequenos serviços aos subsistemas maiores.

**Grafical User Interface (GUI)** – O subsistema que permite ao utilizador interagir com a aplicação e efectuar trocas informação. Através dele o utilizador manipula os dados e a informação utilizada pela aplicação, prepara e agenda a execução de tarefas e observa o estado e acontecimentos ocorridos enquanto a aplicação está em funcionamento. Do ponto de vista do utilizador, este é sempre o ponto de partida para qualquer interacção com o sistema.

**Repositório de Dados** – durante a execução, os dados estão carregados em memória e são sempre estes que são utilizados pela aplicação na realização das suas tarefas. Quando existe a necessidade de materializar estes dados existentes em memória, foram implementados mecanismos que permitem guardar toda a informação em ficheiros. Estes mecanismos guardam os dados mantendo escrupulosamente a sua estrutura interna, tornando assim a sua posterior recuperação sem qualquer diferença em relação aos dados iniciais. No caso das tarefas agendadas para execução, foi necessário guardar também o estado dos agendamentos (Timers) para que fosse posteriormente possível restaurar completamente os agendamentos e o seu estado. Este procedimento garantia que mesmo quando a aplicação não estivesse em execução todas as tarefas agendadas seriam resumidas e executadas assim que a aplicação voltasse a estar activa. Todos os ficheiros são guardados recorrendo à natureza serializável da maioria dos objectos em java. Conjuntamente com a compressão dos próprios ficheiros, na altura da gravação “esconde-se” o seu conteúdo protege-se assim informação crítica do utilizador (por exemplo as credenciais dos equipamentos).

**Executor de Tarefas (Scheduler)** – Representa a essência da aplicação e a razão principal para a sua existência. É o subsistema que garante o agendamento e a execução das tarefas em escrupuloso cumprimento do que foi previamente decidido pelos utilizadores. As tarefas são preparadas pelos utilizadores e assim que estas são submetidas, são agendadas de acordo com a(s) data(s) estipuladas. O controlo da execução é feito internamente pelo ambiente Java que garante o correcto despoletar da tarefa na hora especificada.

**Classes de Apoio** – Diversas classes ou micro-sistemas que permitem dar suporte aos subsistemas maiores. Existe por exemplo uma classe dedicada à interpretação e desmultiplicação das subnets IP e respectivos hosts, para que na altura da execução da tarefa, independentemente de o utilizador ter especificado um grupo ou uma subnet, seja possível obter individualmente todos os hosts representados de forma agrupada. Outro exemplo é o registo de ocorrências (event log) da aplicação. Este registo de ocorrências permite obter informação do sistema e registar o início da execução das tarefas. O decodificador de scripts permite por exemplo interpretar as regras definidas pelo utilizador para a geração do script e criar em tempo de execução da tarefa, todos os scripts para cada host e sem nova intervenção do utilizador.

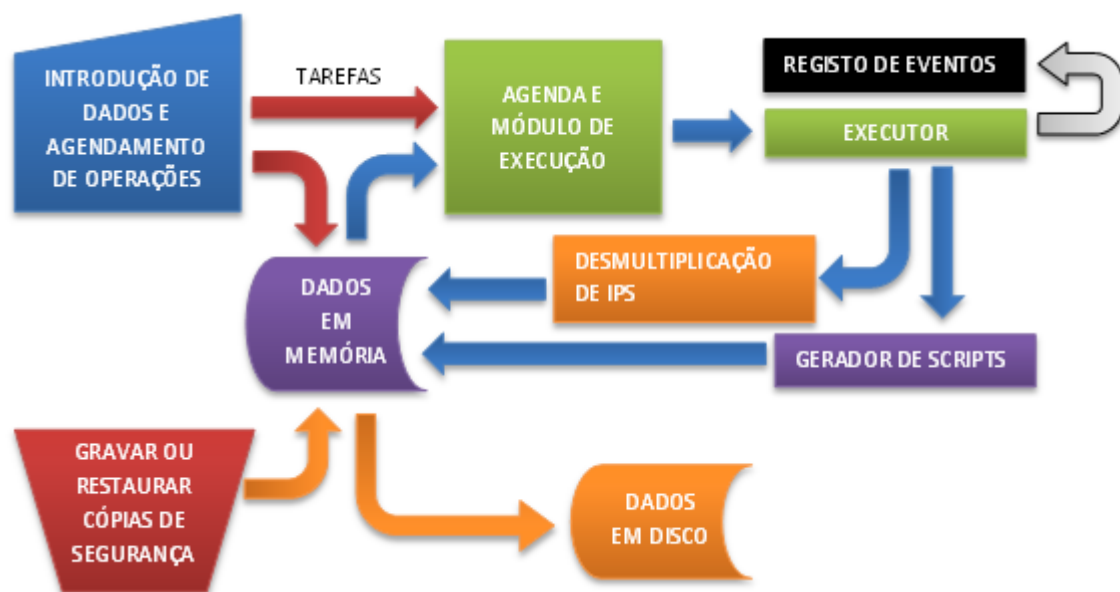


Fig.2 –Arquitectura de Sub-Sistemas e Fluxos Típicos de Execução.

## 2.3.2 Diagrama de Classes.

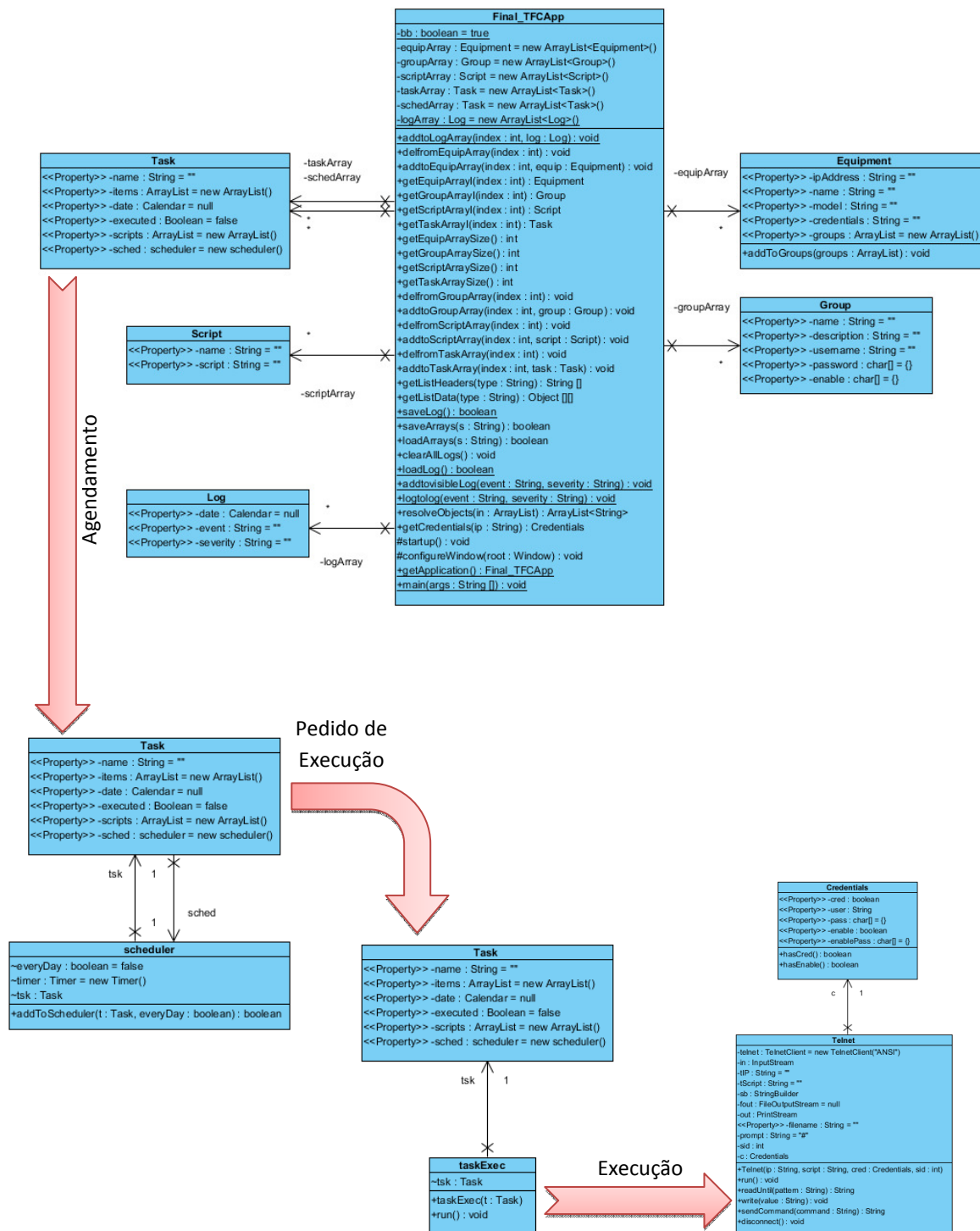
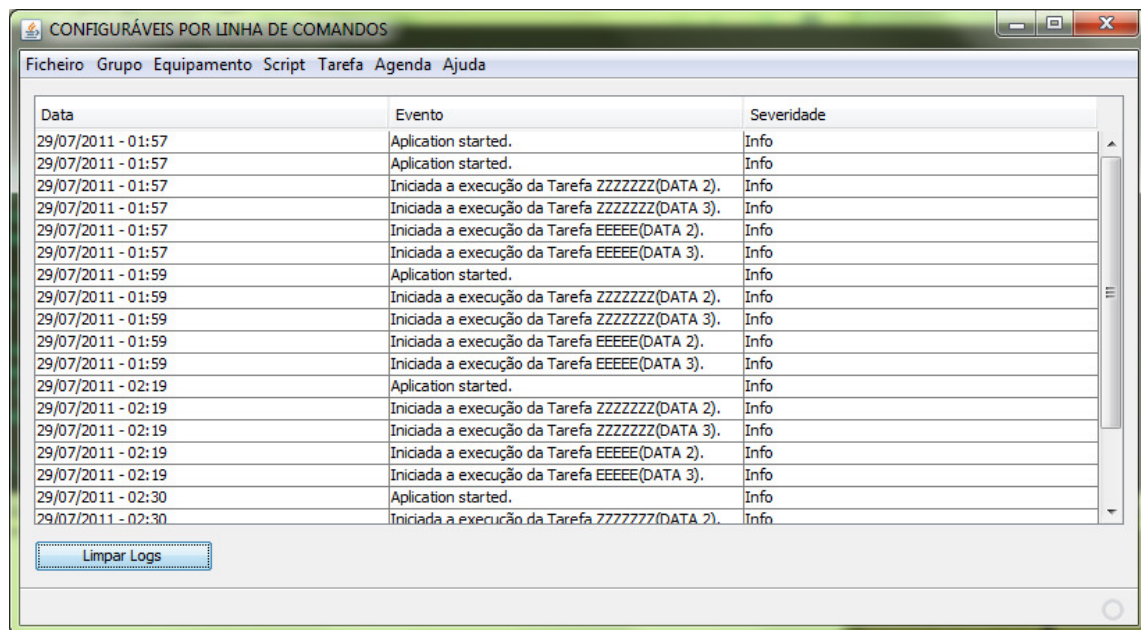


Fig.2 – Diagramas de Classes Mais importantes e Cenário Típico de Execução.

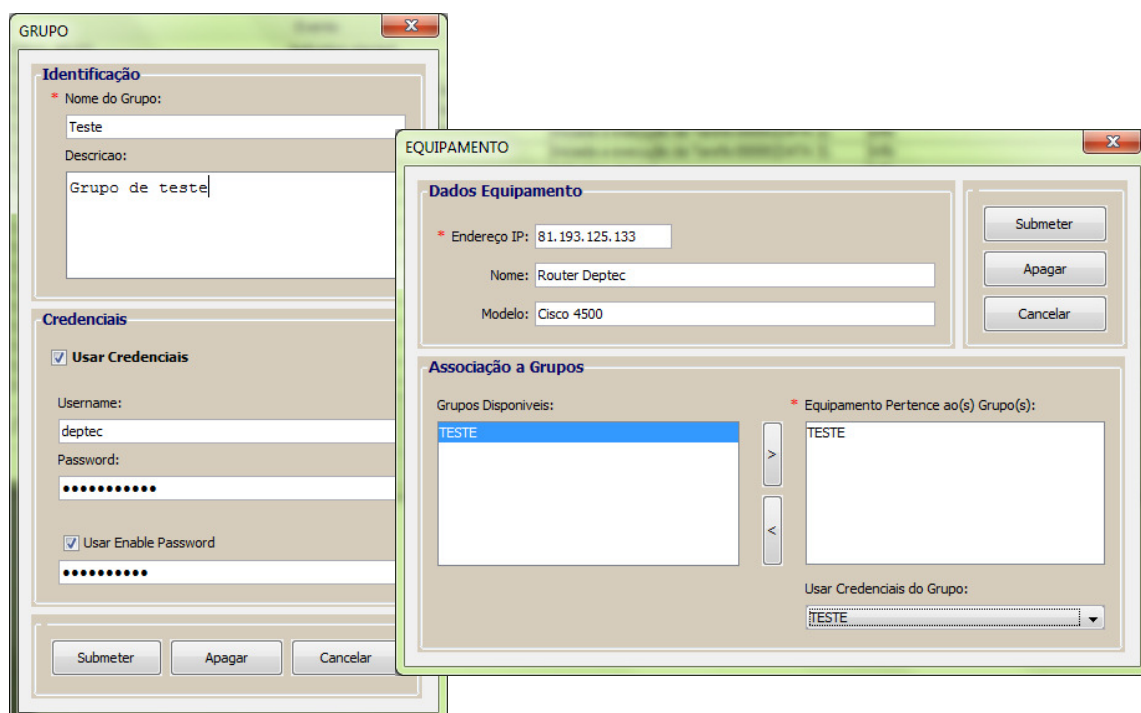
### 2.3.3 Capturas de Ecrã GUI.



The screenshot shows a window titled 'CONFIGURÁVEIS POR LINHA DE COMANDOS' with a menu bar (Ficheiro, Grupo, Equipamento, Script, Tarefa, Agenda, Ajuda) and a table of system events. The table has three columns: Data, Evento, and Severidade. The events are listed chronologically from 29/07/2011 - 01:57 to 29/07/2011 - 02:30. A 'Limpar Logs' button is located at the bottom left of the table area.

Data	Evento	Severidade
29/07/2011 - 01:57	Aplication started.	Info
29/07/2011 - 01:57	Aplication started.	Info
29/07/2011 - 01:57	Iniciada a execução da Tarefa ZZZZZZ(DATA 2).	Info
29/07/2011 - 01:57	Iniciada a execução da Tarefa ZZZZZZ(DATA 3).	Info
29/07/2011 - 01:57	Iniciada a execução da Tarefa EEEEE(DATA 2).	Info
29/07/2011 - 01:57	Iniciada a execução da Tarefa EEEEE(DATA 3).	Info
29/07/2011 - 01:59	Aplication started.	Info
29/07/2011 - 01:59	Iniciada a execução da Tarefa ZZZZZZ(DATA 2).	Info
29/07/2011 - 01:59	Iniciada a execução da Tarefa ZZZZZZ(DATA 3).	Info
29/07/2011 - 01:59	Iniciada a execução da Tarefa EEEEE(DATA 2).	Info
29/07/2011 - 01:59	Iniciada a execução da Tarefa EEEEE(DATA 3).	Info
29/07/2011 - 02:19	Aplication started.	Info
29/07/2011 - 02:19	Iniciada a execução da Tarefa ZZZZZZ(DATA 2).	Info
29/07/2011 - 02:19	Iniciada a execução da Tarefa ZZZZZZ(DATA 3).	Info
29/07/2011 - 02:19	Iniciada a execução da Tarefa EEEEE(DATA 2).	Info
29/07/2011 - 02:19	Iniciada a execução da Tarefa EEEEE(DATA 3).	Info
29/07/2011 - 02:30	Aplication started.	Info
29/07/2011 - 02:30	Iniciada a execução da Tarefa ZZZZZZ(DATA 2).	Info

Fig.3 –Registo de Eventos de Sistema.



The screenshot shows two overlapping windows. The 'GRUPO' window is in the background, showing fields for 'Nome do Grupo' (Teste), 'Descricao', and 'Grupo de teste'. The 'EQUIPAMENTO' window is in the foreground, showing fields for 'Endereço IP' (81.193.125.133), 'Nome' (Router Deptec), and 'Modelo' (Cisco 4500). It also has a section for 'Associação a Grupos' with a list of available groups (TESTE) and a dropdown for 'Equipamento Pertence ao(s) Grupo(s)'. Both windows have 'Submeter', 'Apagar', and 'Cancelar' buttons.

Fig.4 – Front-End para Introdução de Grupos e de Equipamentos.

**Script**

\* Nome:

\* Script:  

```
access-list 110 permit tcp any 10.10.0.0 0.0.255.255 established
access-list 110 deny ip any 10.10.0.0 0.0.255.255
access-list 110 permit ip any any
```

**Exemplos de para geradores de valores ou substituições**

**[SEQ:10-1000:2]** Em cada iteração (script), substitui esta string por um numero de 10 até 1000 em saltos de 2. EX: 10, 12, 14...

**[RND:5-100]** Em cada script gerado, substitui esta string por numeros aleatórios entre 5 e 100.

**[VAL:XX;xpto;teste;55]** Em cada script gerado, substitui esta string pelos valores especificados.

**[FILE:/teste.txt]** Em cada script gerado, substitui esta string por uma linha lida do ficheiro especificado

**NOTA:** Os valores são aplicados ciclicamente. Assim que se chega ao ultimo valor, o ciclo recomeça a partir do 1º valor.

Buttons: Submeter, Apagar, Cancelar

Fig.5 – Frontend para introdução de Scripts.

**TAREFA**

Parte 1 | Parte 2

**Identificação**

\* Nome / Descrição:

**Datas de Execução**

☐ Executar Todos os Dias

Data:

Hora:  :

**Scripts**

Scripts Disponíveis:

\* Scripts a Executar Por Cada Item:

Buttons: Submeter, Cancelar

**Notas**

Fig.6 – Frontend para Agendamento de Tarefas.

## **2.4 Validação e Testes.**

Foram feitos testes de introdução de valores e verificação da consistência dos mesmos durante e após introdução. A análise incidiu ainda sobre o cumprimento de boas práticas de programação, bem como análise semântica e código e relações entre classes. Foram também feitos testes de simulações de utilização real, recorrendo-se a equipamentos em produção mas com tarefas cujo resultado seria inócuo.

Estes testes permitiram a atempada detecção e correcção de faltas numa fase onde o seu impacto e esforço de resolução era mínimo. A introdução de valores de teste permitiu que fossem comprovadas a grande maioria das funcionalidades relacionadas com a validação de inputs. Não foram feitos testes de execução em massa (carga, outros...), mas houve possibilidade de comprovar a execução de múltiplas tarefas em simultâneo sem degradação de performance.

Foram feitos testes de execução em diferentes sistemas operativos, sendo dois deles Microsoft e um Linux Fedora, observando-se sempre o normal funcionamento da aplicação. Desta forma comprovou-se a execução completamente transparente no que toca ao sistema operativo existente, confirmando-se também a portabilidade entre sistemas.

## **3. CONCLUSÕES**

A elaboração deste projecto permitiu aquisição e consolidação de conhecimentos nas diversas áreas de aprendizagem abordadas durante os diversos períodos lectivos. Houve certamente alguma investigação acerca da forma como algumas coisas poderiam ter sido feitas, mas as bases foram em grande parte os conteúdos programáticos lectivos.

A natureza quase pessoal e o facto de estar prevista a sua real utilização em projectos de vertente profissional, tornaram de certa forma este projecto mais aliciente e permitiram um esforço adicional na sua execução. Todavia, ainda permite se sejam feitas muitas outras abordagens que fomentem a continuação do desenvolvimento do projecto mesmo depois de terminada a sua avaliação.

## Referencias

Este trabalho foi elaborado tendo por base a documentação e material didático fornecido pelos docentes das diversas disciplinas ao longo dos vários períodos lectivos. Esporadicamente, houve necessidade de recorrer a consultas de fóruns na internet, relacionados com aspectos muito específicos da programação em Java. Todo restante desenvolvimento advém dos conhecimentos técnicos adquiridos nas aulas ou no exercício da actividade laboral.

Sites de referência mais importantes:

Páginas no moodle das disciplinas de ACS, Engenharia de Software e Linguagens de Programação II.

[moodle.ulusofona.pt](http://moodle.ulusofona.pt)

Outras páginas:

[www.java.net/forum](http://www.java.net/forum)

[www.java-forums.org](http://www.java-forums.org)

[download.oracle.com](http://download.oracle.com)

[www.rosindia.net](http://www.rosindia.net)

## Bibliografía

Declan Delaney and Stephen Brown, *Document Templates for Student Projects in Software Engeneering*, Department of Computer Science, National University of Ireland, Maynooth, 2002

IEEE Std. 1012a-1998 *IEEE Standard for Software Verification and Validation - Supplement to 1012-1998 - Context Map to IEEE 12207.1*

IEEE-1058.1-1987 *IEEE Standard for Software Project Management Plans Final Year Project Handbook*, Department of Computer Science, National University of Ireland, Maynooth, 2001