



Universidade do Porto
Faculdade de Engenharia

FEUP



Filipe Guedes

MIB (management information base) UTRAN



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



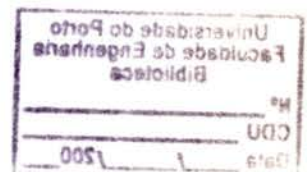
MIB3G

Filipe Rodrigo Bastos Guedes

Relatório do Estágio Curricular da LEIC 2004/2005

Orientador na FEUP: Prof. Manuel Ricardo
Orientador na Optimus: Engenheira Adelaide Coutinho

Setembro de 2005



004(047.3) LEIC/EIC5202 2005/SUG f

Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº 81405
CDU
Data 15/03/2006

Aos meus pais,

Resumo

O projecto enquadra-se no âmbito da proposta de Estágio Curricular MIB (Management Information Base) UTRAN. Tem como principal objectivo a especificação e desenvolvimento de uma aplicação informática que permita construir e manter a base de dados de parâmetros, da rede UTRAN.

O sistema de acesso rádio de uma rede UMTS apresenta uma grande complexidade do ponto de vista dos parâmetros que é necessário manter, para cada um dos seus elementos de rede.

Existe a necessidade de visualização e manipulação dessa informação em tempo real, o que tornou fundamental o mantimento e a correcta actualização dos dados nessa MIB.

O projecto surgiu, devido ao facto do fornecedor do equipamento UTRAN, ainda não disponibilizar uma solução deste tipo.

Agradecimentos

Em primeiro lugar quero agradecer a toda a equipa de suporte da Optimus pois foi com eles que trabalhei diariamente ao longo dos seis meses de estágio, para que este projecto fosse avante

Agradeço igualmente aos meus orientadores, Prof. Manuel Ricardo (FEUP) e Eng.º António Araujo (OPTIMUS), pela disponibilidade que manifestaram no apoio solicitado para a elaboração deste documento

Por fim, agradeço à minha namorada Catarina por todo o seu apoio, e paciência e aos meus pais por me terem ajudado nas outras responsabilidades da vida

Muito Obrigado a todos

Glossário de Termos, Siglas e Acrónimos

- **Backup** – Cópia de Segurança;
- **Parser** - Serve para analisar a estrutura gramatical de uma dada entrada (input);
- **Benchmark** – É um padrão de referência utilizado para se medir a performance de determinada tecnologia em resposta a diferentes situações;
- **String** – Sequência de caracteres;
- **Script** – São linguagens de programação cujo código fonte é interpretado pelo programa em tempo de execução;
- **XML (Extensible Markup Language)** – Linguagem que permite a criação de tags próprias não existentes no HTML. Com o XML, pode-se criar links apontando para mais de um documento, diferentemente do que ocorre com os links do HTML, que sempre se referem a um endereço único;
- **UMTS – (Universal Mobile Telecommunications System)** é uma das tecnologias de terceira geração dos telemóveis.
- **Restore** – Mecanismos para restaurar uma copia de segurança
- **Default.** – Valor por defeito
- **Balloon-Tips** – Ajuda para o utilizador sobre a forma de pequenos balões.
- **File-Select** – Menu de entrada para a escolha de um ficheiro.
- **Check** – Verificar uma determinada acção.
- **Commit** – Validar uma determinada acção.
- **Log** – Registo de uma determinada acção
- **Cron.**- Programa que executa comandos agendados no sistema operativo UNIX

Índice de Conteúdos

Glossário de Termos, Siglas e Acrónimos	vi
1 Introdução.....	10
1.1 Optimus Telecomunicações SA.....	10
1.2 MIB3G.....	11
1.3 Motivação.....	12
1.4 Organização do documento	12
2 O Projecto.....	14
2.1 Limitações e necessidades	14
2.2 Planeamento.....	15
2.2.1 Fase 1.....	15
2.2.2 Fase 2.....	15
2.2.3 Fase 3.....	15
2.2.4 Fase 4.....	16
3 Análise Tecnológica	17
3.1 Introdução.....	17
3.2 LMT (Local Management Tool).....	17
3.3 PERL	18
3.4 MySQL.....	18
3.5 Interface – Módulo PERL/Tk.....	19
3.6 XML	19
4 Solução Proposta	20
4.1 Arquitectura.....	20
4.1.1 Arquitectura Física.....	21
4.2 Funcionalidades.....	22
4.2.1 Menu Inicial	22
4.2.2 Ferramenta Administrativa.....	23
4.3 Painel Principal MIB3G	24
4.4 Operações sobre base de dados	25
4.5 Validação de Parâmetros.....	26
4.6 Commit.....	27
4.7 Log	27
4.8 Exportar para XML.....	28
4.9 Backup da Base de dados	29
5 Implementação da solução proposta.....	30
5.1 Interface.....	31
5.2 Cron	31
5.3 Base de dados	31
5.4 Ficheiro de entrada	32
5.5 Parsing.....	33
5.5.1 1ª Fase do Parser.....	33
5.5.2 2ª Fase do Parser.....	33
5.5.3 3ª Fase do Parser.....	34

- 5.6 Validação de Parâmetros.....35
 - 5.6.1 Parse da gama de valores..... 36
- 5.7 Interacção com Base de dados.....37
 - 5.7.1 Função de Inserção..... 37
 - 5.7.2 Função de Remoção 38
 - 5.7.3 Função de Modificação 39
- 6 Validação de Requisitos 40
- 7 Conclusões 41
- Referências e Bibliografia..... 42
- ANEXO A: Código da Interface..... 43
- ANEXO B: Código da cron..... 44
- ANEXO C: Exemplo de algumas tabelas 45
- ANEXO D: Apresentação final à Optimus..... 46

Índice de Figuras

Figura 1. Local Management Tool..... 17

Figura 2. Menu Inicial 22

Figura 3. Ferramenta administrativa..... 23

Figura 4. Painel Principal da aplicação do MIB3G 24

Figura 5. Escolha de ficheiro de entrada..... 25

Figura 6. Validação de Parâmetros..... 26

Figura 7. Log integral da aplicação. 27

Figura 8. Exportação para XML..... 28

Figura 9. Backup da base de dados 29

Figura 10. Exemplo de um ficheiro de entrada..... 32

Figura 11. Função Parser do comando ADD 34

Figura 12. Ficheiro de validação de parâmetros. 35

Figura 13. Função que examina gama de valores 36

Figura 14. Função de inserção de valores 37

Figura 15. Função de remoção de valores..... 38

Figura 16. Função de modificação de valores..... 39

Figura 17. Gráfico de Benchmark MIB3G 40

1 Introdução

Neste capítulo pretende-se transmitir uma imagem global da instituição onde este estágio teve lugar e do sistema que está na origem do âmbito e título deste estágio – MIB3G. Além disso, são apresentadas a motivação do autor na escolha deste estágio e a organização geral deste documento

1.1 Optimus Telecomunicações SA

A Optimus nasceu da união de empresas que integram os maiores e melhores grupos económicos nacionais e internacionais, do qual faz parte o Grupo Sonae, o maior accionista da operadora. A Optimus é ainda participada pelo segundo maior operador europeu de telecomunicações, a Orange / France Telecom.

A entrada da Optimus no mercado, a 15 de Setembro de 1998, assinalou uma nova fase no sector das Telecomunicações em Portugal, atingindo nas primeiras cinco semanas, uma base de 100 mil clientes. Um dos melhores, senão o melhor, resultado jamais conseguido por um operador de comunicações móveis.

1.2 MIB3G

O projecto enquadra-se no âmbito da proposta de Estágio Curricular MIB (Management Information Base) UTRAN. Tem como principal objectivo a especificação e desenvolvimento de uma aplicação informática que permita construir e manter a base de dados de parâmetros, da rede UTRAN.

O sistema de acesso rádio de uma rede UMTS apresenta uma grande complexidade do ponto de vista dos parâmetros que é necessário manter, para cada um dos seus elementos de rede.

Existe a necessidade de visualização e manipulação dessa informação em tempo real, o que tornou fundamental o mantimento e a correcta actualização dos dados nessa MIB.

O projecto surgiu, devido ao facto do fornecedor do equipamento UTRAN, ainda não disponibilizar uma solução deste tipo.

Para o seu desenvolvimento para adoptada um mecanismo de compilador, ou seja, tal como vamos ver nos capítulos seguintes, a aplicação não é mais do que um parser enorme, pois consegue definir numa linguagem de bases de dados todo o conjunto de informação presente em toda a parameterização que é feita numa base diária na equipa de suporte e manutenção da Optimus Telecomunicações SA.

Tal como o leitor irá facilmente perceber foi feito um trabalho muito intenso e minucioso já que estamos a lidar com configurações extremamente críticas, pois uma configuração mal efectuada poderá ter imensos prejuízos do ponto de vista financeiro.

Para o seu desenvolvimento o autor contou com varias tecnologias entre as quais destaca uma arquitectura assente num servidor SUN SOLARIS 8 que desde início mostrou ser extremamente fiável para a aplicação em causa.

O sistema está pensado para integrar interfaces com os vários elementos da equipa de suporte assegurando o controlo interno e a manutenção por essa mesma equipa.

1.3 Motivação

De entre as propostas de estágio disponíveis no âmbito da cadeira homónima da LEIC, pesou na decisão do autor, em primeiro lugar, o modelo de negócio em que a OPTIMUS se insere – uma empresa de telecomunicações, mas com uma relação estreita com a informática, área que é fonte de entusiasmo e curiosidade para o autor. Aliado a este factor, agradou igualmente o pensamento da equipa integrante que cedo fez perceber que engenharia de software e evolução tecnológica seriam sempre premissas de topo na hierarquia de objectivos da empresa, entendendo por isso o autor que poderia auspiciar a um estado de crescimento constante como pessoa e como futuro Engenheiro Informático.

O ambiente e condições de trabalho proporcionadas, bem como a personalidade e *Know How* dos elementos da equipa da OPTIMUS foram também factores de relevo na decisão.

1.4 Organização do documento

Para além deste capítulo inicial, onde se introduz a empresa e o sistema com os quais se prende o estágio, o presente documento inclui outros cinco capítulos na sua organização.

Nos pontos iniciais do capítulo seguinte é feita uma apresentação geral do projecto, explicando-se quais as metas traçadas para o mesmo e o consequente plano de trabalhos. No último ponto é feita uma referência ao enquadramento e importância do projecto na empresa.

No terceiro capítulo são apresentadas as tecnologias afectas ao projecto, sendo a solução adoptada para este apresentada no quarto capítulo. Aqui, o leitor encontrará uma abordagem às funcionalidades implementadas, ilustrações das mesmas, reservando-se o capítulo seguinte para a explanação dos pormenores de implementação que conduziram a esta solução.

O sexto e último capítulo é reservado para as ilações finais do estágio curricular, enquanto experiência profissional primária, assim como do projecto desenvolvido.

Existem ainda secções adicionais do documento (glossário e anexos) que têm a finalidade de esclarecer o leitor nas eventuais dúvidas que possam surgir.

2 O Projecto

Neste capítulo é feita uma apresentação geral do projecto, procurando-se situar o leitor nas necessidades e objectivos que existiam à partida e que deram origem à definição inicial do plano de trabalhos.

2.1 Limitações e necessidades

O sistema MIB3G apresenta uma grande complexidade do ponto de vista dos parâmetros que são necessários manter. Não só pela sua imponentia pois estamos a lidar com milhares de configurações mas também pela complexidade que as mesmas apresentam.

À medida que se ia conhecendo o sistema, numa primeira fase de análise e estudo das suas limitações e necessidades, cedo deu para perceber que se tratava de um projecto muito ambicioso mas ao mesmo tempo representava um desafio para o autor.

Em termos práticos era necessário o desenvolvimento de um compilador que conseguisse traduzir para linguagem de base de dados todas as configurações da rede UMTS de uma forma automática e quase transparente para o utilizador. Desta forma foi feito inicialmente um vasto estudo de tecnologias para que a aplicação MIB3G introduzisse uma mais valia mas ao mesmo tempo que fosse um sistema fácil e sem grande necessidade de formação.

2.2 Planeamento

2.2.1 Fase 1

- Aprender o “modus operandi” do RANS&O
 - Tomar contacto com as rotinas de Operação e Manutenção
 - Ficar familiarizado com as ferramentas de OSS 3G: LMT (Local Management Tool) OMCu
 - Conhecer o processo de interacção com as bases de dados existentes (2G e 3G)
- Stagging da máquina SUN SPARC ULTRA 60
 - Clean install Solaris
 - Instalação do software MySQL
 - Instalação do software DBI DBD para a interacção MySQL – PERL
 - Instalação do servidor Apache
 - Instalação do PHP
- Contacto com os dados MIB3G
 - Conhecer os ficheiros exportados dos RNCs: CFGMML.txt
 - Conhecer o ficheiro de partilha de informação: CFGMML.xml
 - Conhecer a base de dados MySQL existente

2.2.2 Fase 2

- Modelização da Base de Dados
 - Definição da estrutura da base de dados
 - Mapeamento dos parâmetros 3G em tipo e gama de valores
- Aplicação de interacção com a base de dados

2.2.3 Fase 3

- Operação e Manutenção
 - Testes Funcionais/Operacionais
 - Backup e Restore
 - Testes de Robustez e Fiabilidade
- Pesquisa de Add-ons possíveis à aplicação

2.2.4 Fase 4

- Add-ons
- Documentação detalhada do projecto.

3 Análise Tecnológica

3.1 Introdução

Neste capítulo são apresentadas as várias tecnologias que, de alguma forma, estão envolvidas no desenvolvimento do projecto e sempre que se justifique, são apresentadas as vantagens e as desvantagens que cada uma delas acarreta para o mesmo.

Diga-se que nos limitamos a uma apresentação genérica daquelas, evitando longas e massudas descrições do funcionamento de cada tecnologia.

3.2 LMT (Local Management Tool)

Esta ferramenta disponibilizada pela Motorola é responsável pela injeção dos parâmetros na rede UTRAN.

A aplicação MIB3G desenvolvida pelo autor, surge de forma análoga a esta ferramenta pois permite que todos essas configurações de rede sejam manipuladas e visualizadas em tempo real e estejam sempre disponíveis aos elementos do grupo de suporte da Optimus Telecomunicações SA.

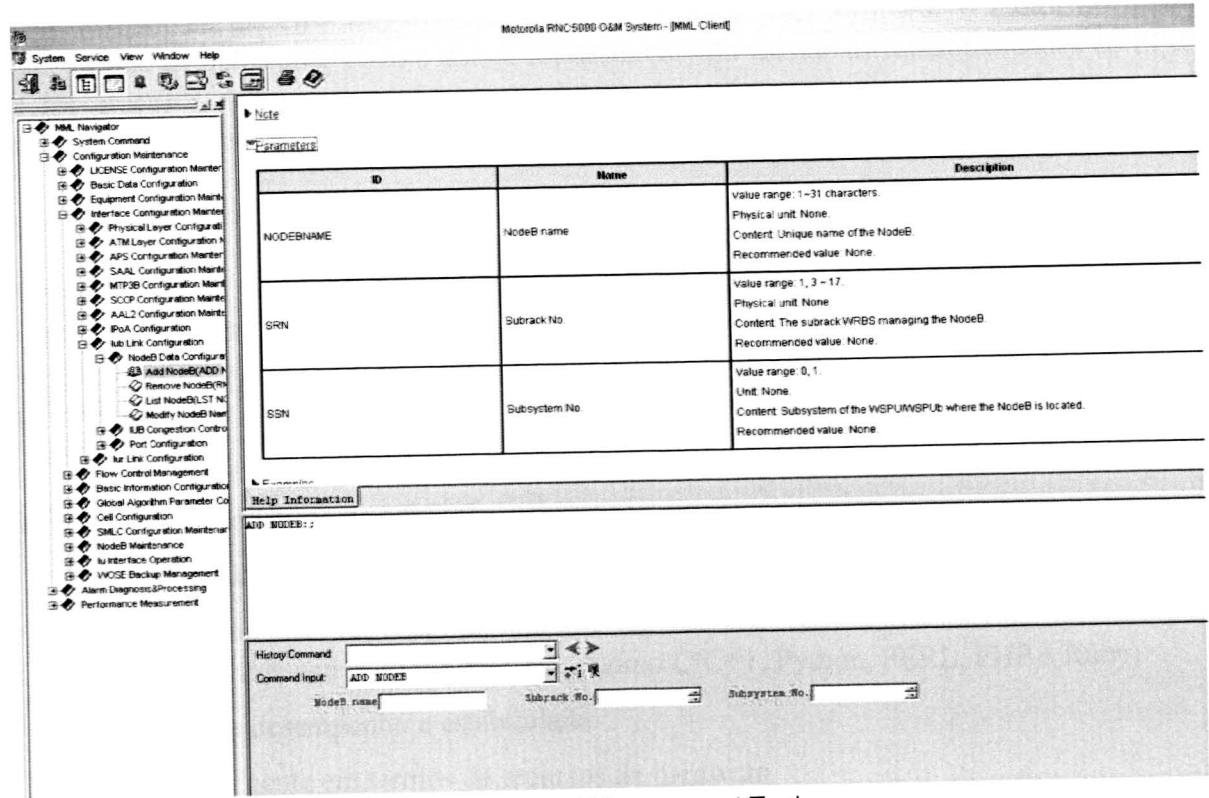


Figura 1. Local Management Tool.

3.3 PERL

PERL é a abreviação de *Practical Extracting and Reporting Language*, que indica que se trata de uma linguagem de programação muito prática para extrair informação de arquivos de texto e gerar informação a partir do conteúdo dos arquivos.

Podemos considerar os seguintes pontos fortes do PERL:

- Declaração de variáveis implícita. O primeiro carácter distingue se a variável é um escalar, um array ou um array associativo.
- Strings e arrays não necessitam de definição de tamanho. Eles são criados dinamicamente e crescem conforme a necessidade.
- Todas as variáveis são inicializadas com um valor default.
- Possui um conjunto muito rico de operações para busca de padrões em textos.
- Um conjunto de funções internas à linguagem com diversas funcionalidades.
- Uma sintaxe exótica, semelhante em alguns aspectos ao “C”, mas bem diferente em outros.
- O uso de expressões regulares.
- Fácil acesso a qualquer base de dados.

O PERL foi de extrema importância para este projecto pois tornou-se o motor de todo o mecanismo da aplicação. Todo o parser foi desenvolvido usando as funcionalidades do PERL e foi graças à sua extrema eficácia na busca de padrões em textos (através do uso de expressões regulares) que todo este trabalho foi possível.

3.4 MySQL

O MySQL é um sistema de gestão de base de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) como interface. É actualmente uma das bases de dados mais populares, com mais de 4 milhões de instalações pelo mundo.

Principais características do MySQL:

- Portabilidade (suporta praticamente qualquer plataforma atual)
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como C/C++, Python, PERL, PHP e Ruby)
- Excelente desempenho e estabilidade
- Pouco exigente em termos de recursos de hardware
- Facilidade de uso
- É um Software Livre
- Suporte a vários engines (como MyISAM e InnoDB)

3.5 Interface – Módulo PERL/Tk

A utilização deste módulo foi de extrema importância já que se trata de uma ferramenta de muito fácil utilização e compreensão. Para além disto foi muito relevante a sua completa integração com o PERL e com os scripts desenvolvidos.

3.6 XML

Outra tecnologia existente é o XML, acrónimo para *Extensible Markup Language*, formato de texto simples e muito flexível, permitindo a estruturação e a descrição da informação.

A opção da utilização do XML na aplicação MIB3G, deve-se ao facto de ser um formato *standard*, de permitir a definição de uma gramática própria à medida da natureza do problema e de existir um leque grande de ferramentas disponíveis para o seu processamento.

Na aplicação, o seu uso foi útil pois permite a exportação de uma qualquer tabela da base de dados para extracção de informação.

4 Solução Proposta

Neste capítulo apresenta-se a solução adoptada que visa dar resposta aos objectivos deste projecto. Neste sentido, o leitor irá deparar com uma abordagem referente às funcionalidades implementadas, vantagens e desvantagens das mesmas, reservando-se o capítulo seguinte para a explanação técnica dos pormenores de implementação que conduziram a esta solução.

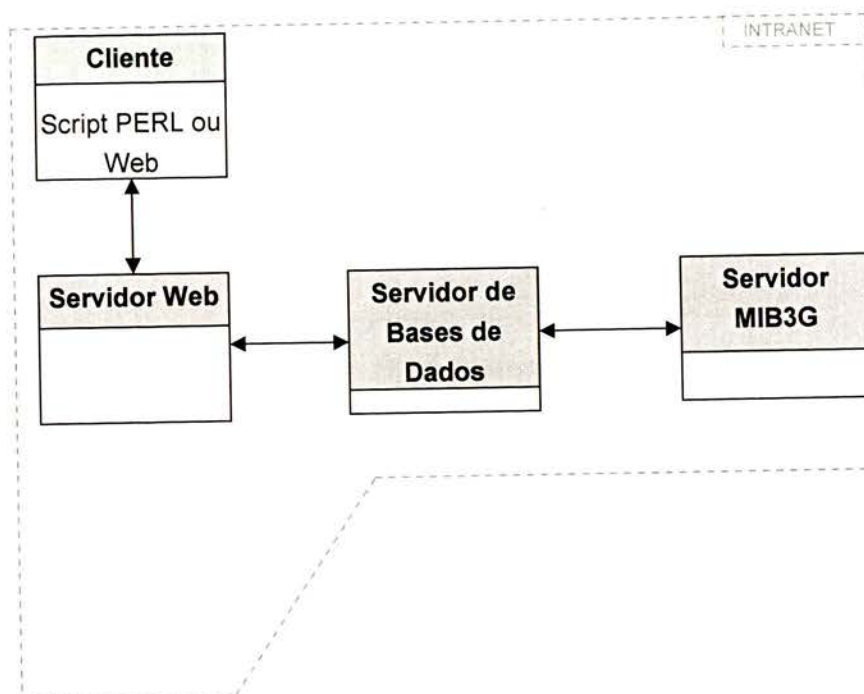
4.1 Arquitectura

O assumir das necessidades do sistema esteve na base da justificação deste estágio, pelo que a solução proposta para o projecto deriva de muita investigação por parte do autor e também pelo largo conhecimento da equipa de suporte da Optimus.

Eram requisitos fundamentais deste projecto o incremento de funcionalidades e que acima de tudo que fosse implementada uma solução fiável e bastante consistente valorizando-se ainda o desenvolvimento de componentes reutilizáveis sempre que tal fosse possível.

Assim, a arquitectura física do sistema, esquematizada no ponto seguinte, foi alvo de estudo e de total responsabilidade do autor já que lhe foi dada desde início a possibilidade de decidir qual o caminho a percorrer durante a realização deste estágio.

4.1.1 Arquitectura Física



O Servidor de base de dados (MySQL) é o servidor que mais se enquadra no propósito deste trabalho, já que é nele que ficam alojadas todas as configurações da rede UMTS. Além disto, este servidor é a ponte entre o cliente (utilizador) e o restante do sistema, relacionando-se directamente com o PERL, (servidor MIB3G) o servidor mais importante no sistema, pois nele reside todo o motor da aplicação.

Por estas razões, estes dois servidores merecem um cuidado especial na definição de critérios de performance e estratégias que garantam robustez e integridade da informação.

4.2 Funcionalidades

4.2.1 Menu Inicial



Figura 2. Menu Inicial

Depois de iniciarmos a aplicação com palavra-chave de root o utilizador terá de escolher entre duas hipóteses. Uma delas será a GUI, ou seja, a interface com a aplicação MIB3G e uma ferramenta administrativa onde poderá realizar operações críticas.

4.2.2 Ferramenta Administrativa

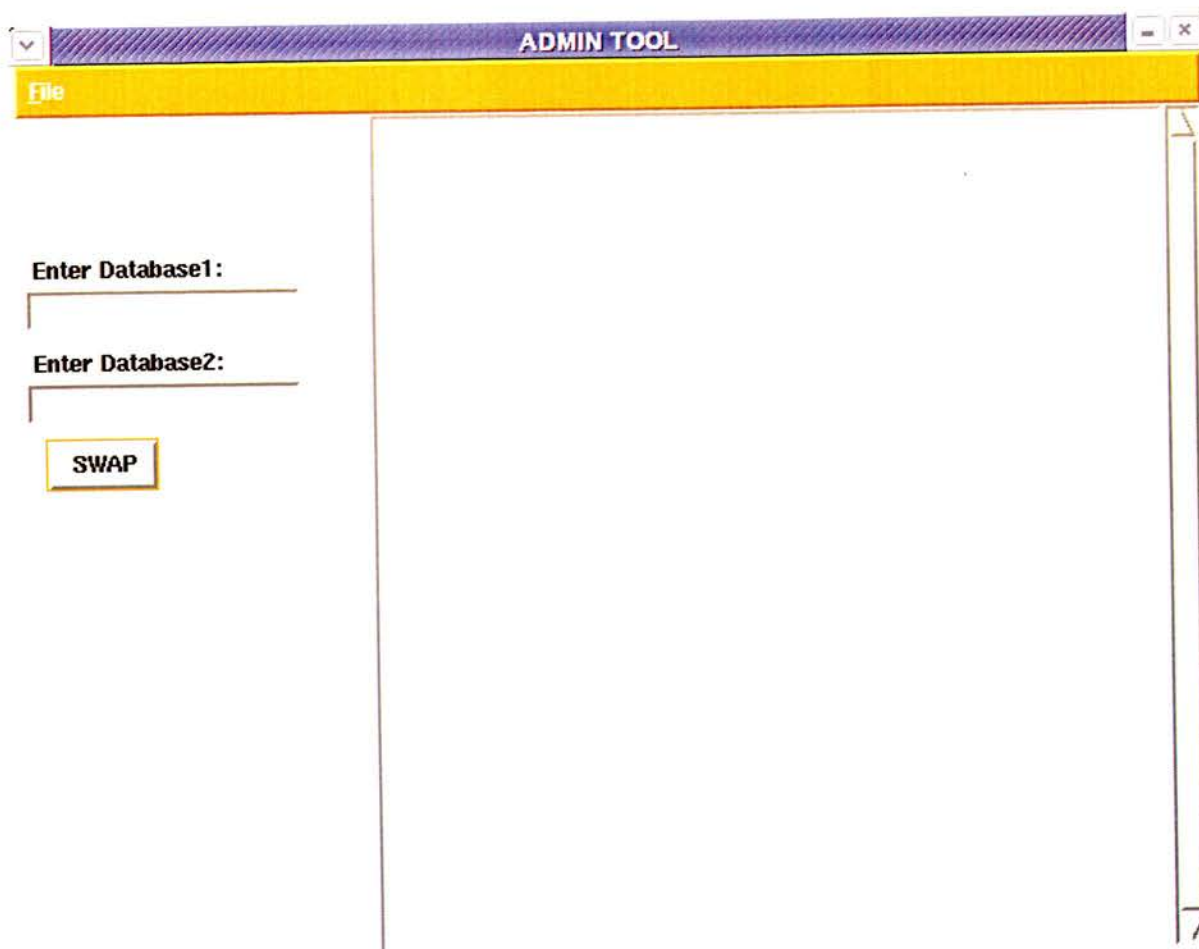


Figura 3. Ferramenta administrativa.

Nesta ferramenta de uso muito cuidado podemos realizar operações críticas mas fundamentais como apagar uma determinada base de dados ou realizar uma troca.

Uma troca (swap) consiste em trocar informação de duas bases de dados. Esta operação pode ser de extrema importância pois o utilizador poderá querer efectuar uma cópia de segurança e manter sempre numa base de dados as configurações realizadas na última operação.

4.3 Painel Principal MIB3G

O painel é sem dúvida a principal referência em termos de navegação na aplicação, de forma a que o utilizador possa sempre aceder a todas as restantes funcionalidades de forma rápida e intuitiva.

Como se pode observar na figura seguinte o painel principal é constituído por um conjunto de ícones devidamente comentados com *balloon-tips* centralizando nesse painel todas as funcionalidades.

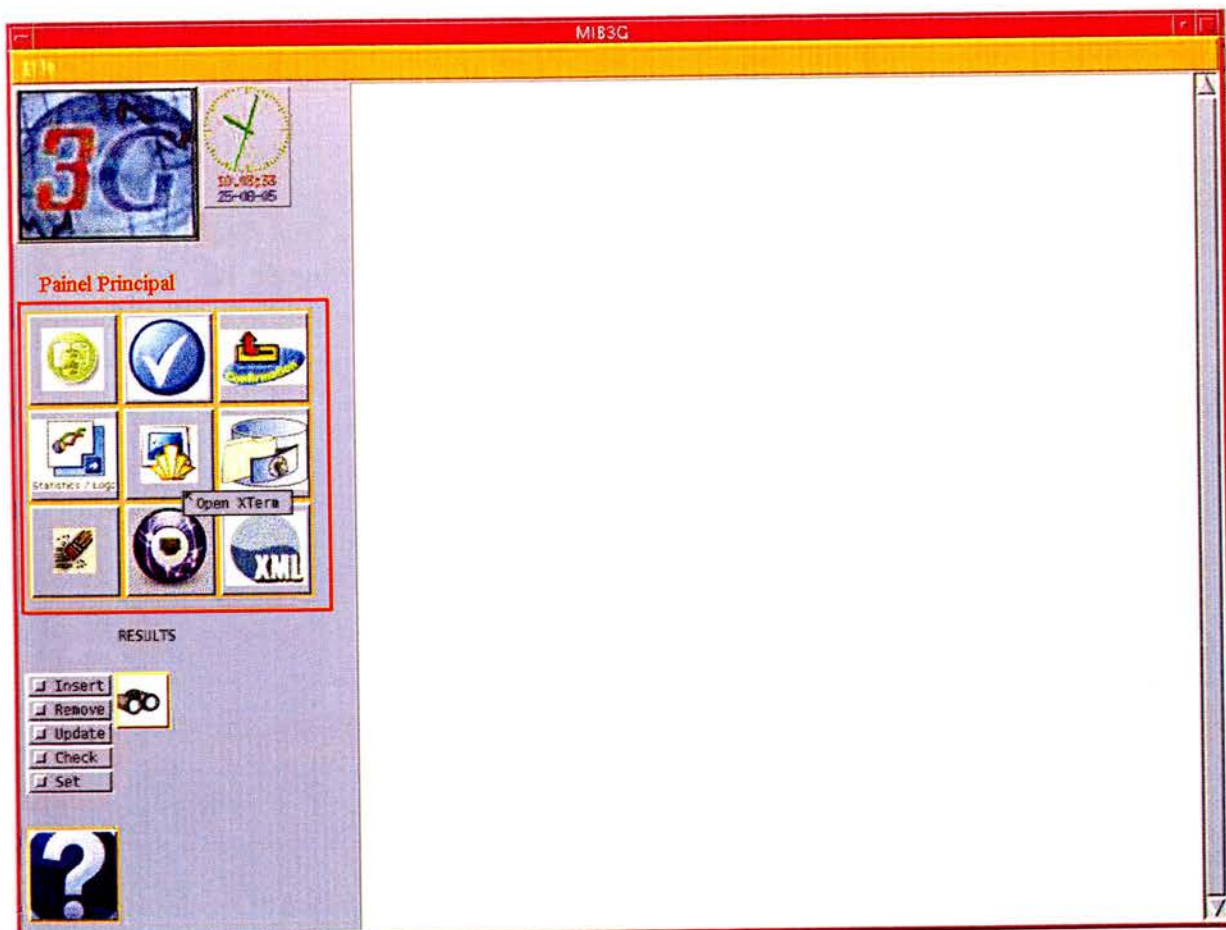


Figura 4. Painel Principal da aplicação do MIB3G



4.4 Operações sobre base de dados

Será aqui que é feito todo o *parsing* do ficheiro de entrada e respectivas operações na base de dados. Assim sendo, o utilizador irá ser confrontado com um menu do tipo *fileselect* em que terá apenas que escolher o seu ficheiro de entrada. Quando essa operação tiver terminado a aplicação MIB3G irá apresentar ao utilizador os resultados das operações efectuadas.

Na figura seguinte podemos ver a escolha de um ficheiro de entrada e seus respectivos resultados.

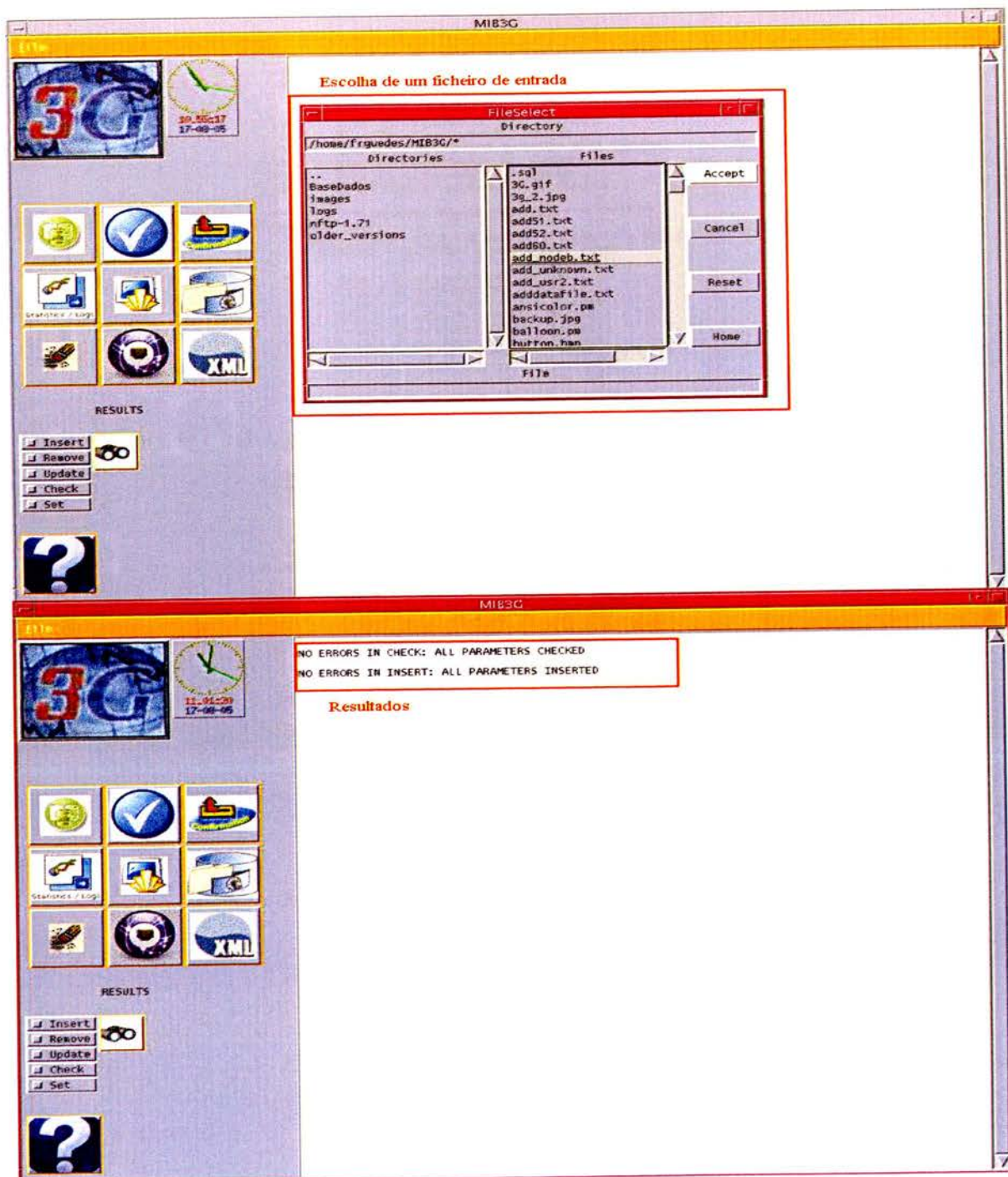


Figura 5. Escolha de ficheiro de entrada



4.5 Validação de Parâmetros

Tendo em conta que o utilizador MIB3G poderá querer apenas um check do seu ficheiro de entrada, ou seja, que esse mesmo ficheiro respeite todas as regras quer de sintaxe quer de semântica, poderá fazê-lo recorrendo a esta funcionalidade. Novamente terá apenas que escolher o seu ficheiro de entrada e no fim observar os resultados mais detalhadamente recorrendo ao log dessa operação. A figura seguinte ilustra com clareza essa operação.

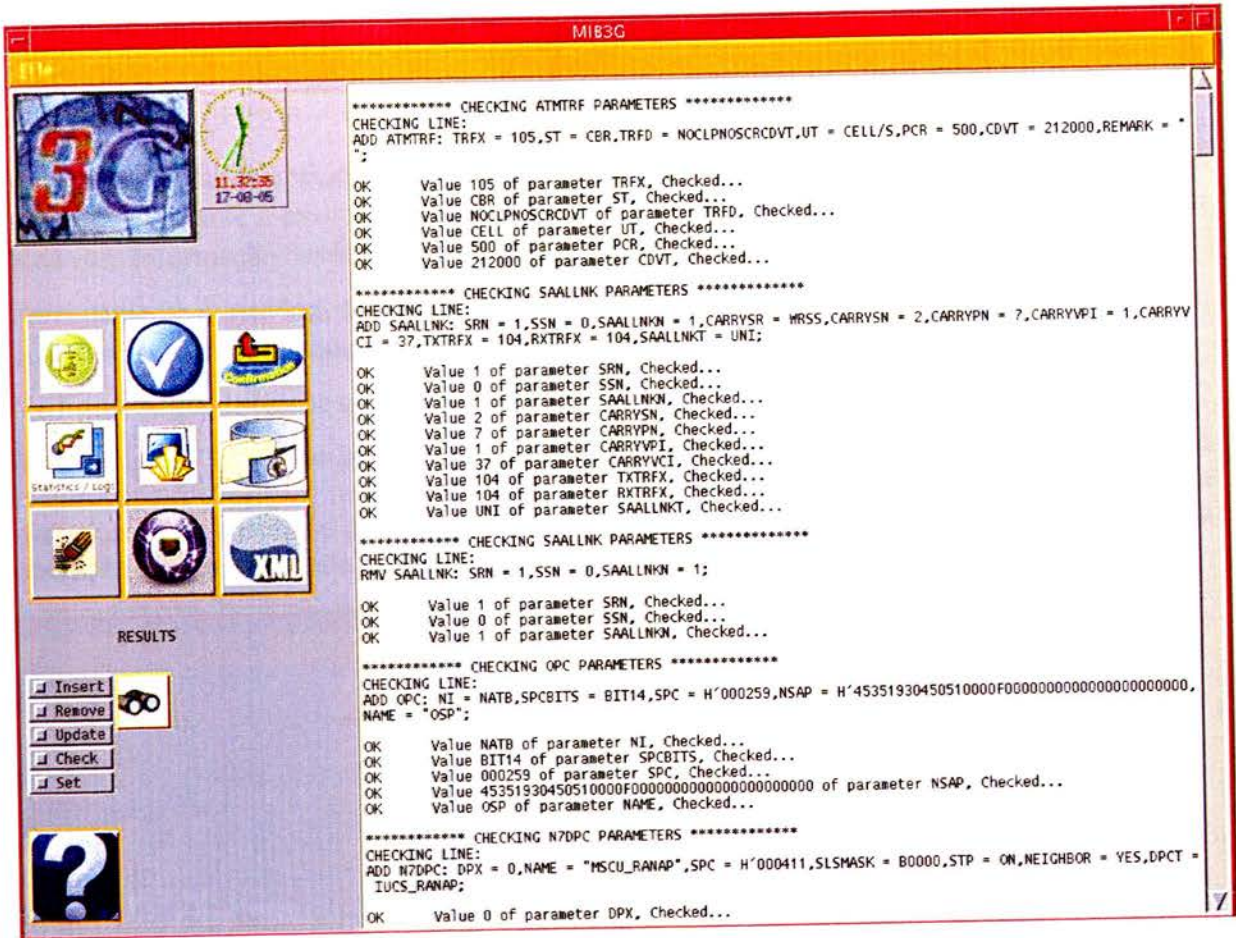


Figura 6. Validação de Parâmetros



4.6 Commit

Tal como foi explicado anteriormente todas as operações são realizadas numa base de dados virtual para que posteriormente possa ser feito a exportação de toda essa informação para a base de dados real. Esta funcionalidade permite essa mesma situação.



Statistics / Logs

4.7 Log

Fazia sentido numa aplicação deste tipo poder guardar num log todo o registo das operações efectuadas durante a execução do programa. Assim sendo o denominado log integral contém toda essa informação devidamente formatado para uma pesquisa rápida e intuitiva.

Paralelamente a este log, foi embutido igualmente um log referente apenas à sessão, ou seja, guarda apenas todas as operações relativas à sessão activa.

Na figura seguinte temos uma pequena parte desse log.

No retângulo vermelho podemos ver a data da sessão realizado e o respectivo ficheiro de entrada que foi inserido nessa sessão. No retângulo azul temos o tipo de operação realizada neste caso um check de parâmetros. Por fim no retângulo verde temos o resultado dessa operação e se foi realizado com sucesso (OK) ou com algum tipo de erro (ERROR).

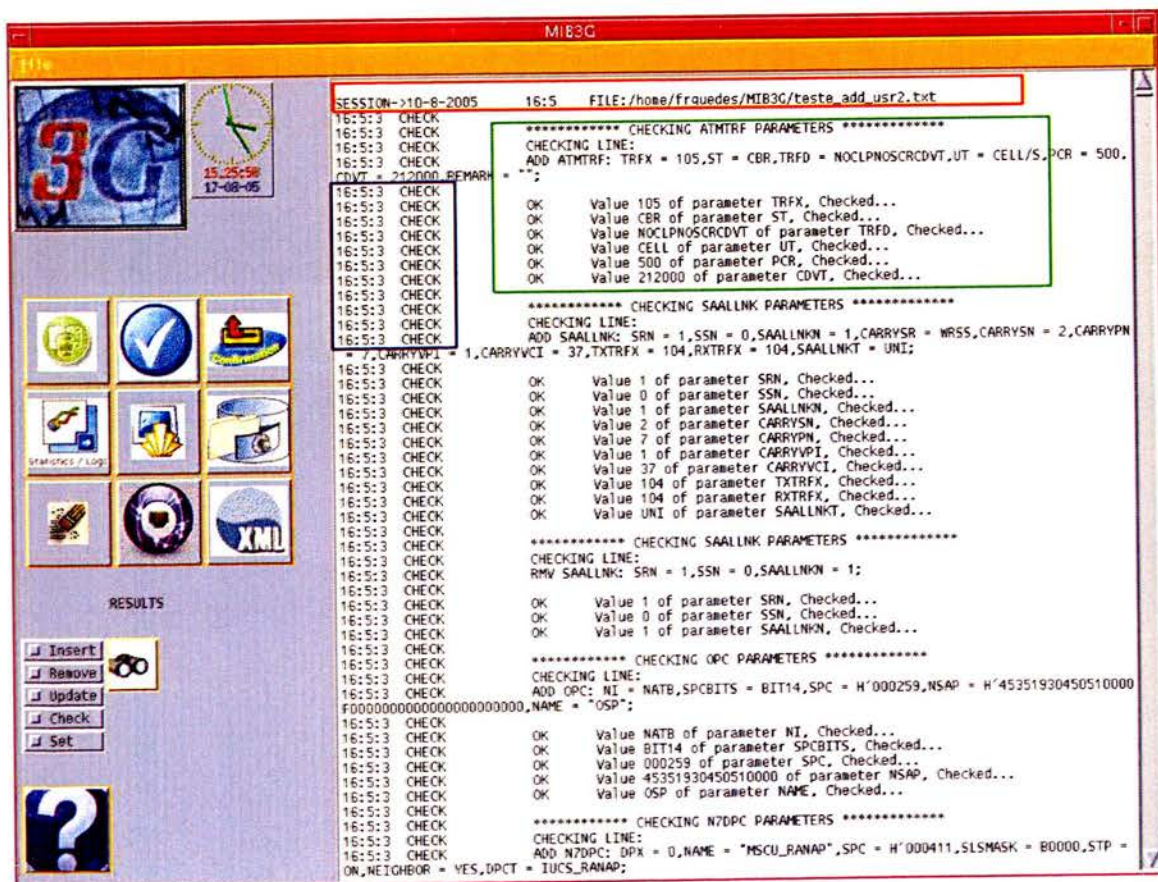


Figura 7. Log integral da aplicação.

4.8 Exportar para XML



Esta funcionalidade permite ao utilizador exportar toda a informação contida na base de dados para um ficheiro XML para futuramente ser editado por uma qualquer ferramenta.

Na figura seguinte podemos ver um ficheiro XML depois de editado na própria aplicação.

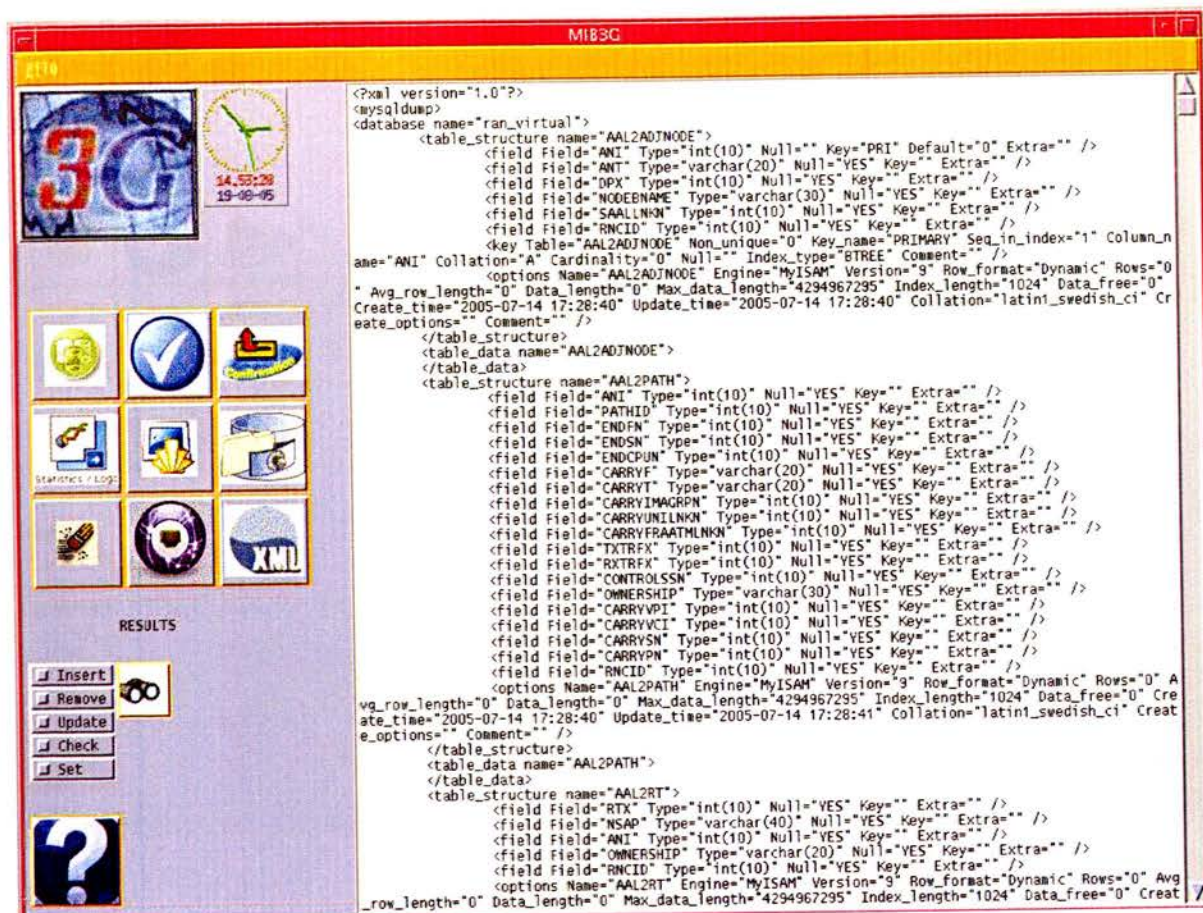


Figura 8. Exportação para XML



4.9 Backup da Base de dados

A partir desta funcionalidade, e por motivos de segurança, é possível em qualquer momento guardas uma cópia de segurança da base de dados.

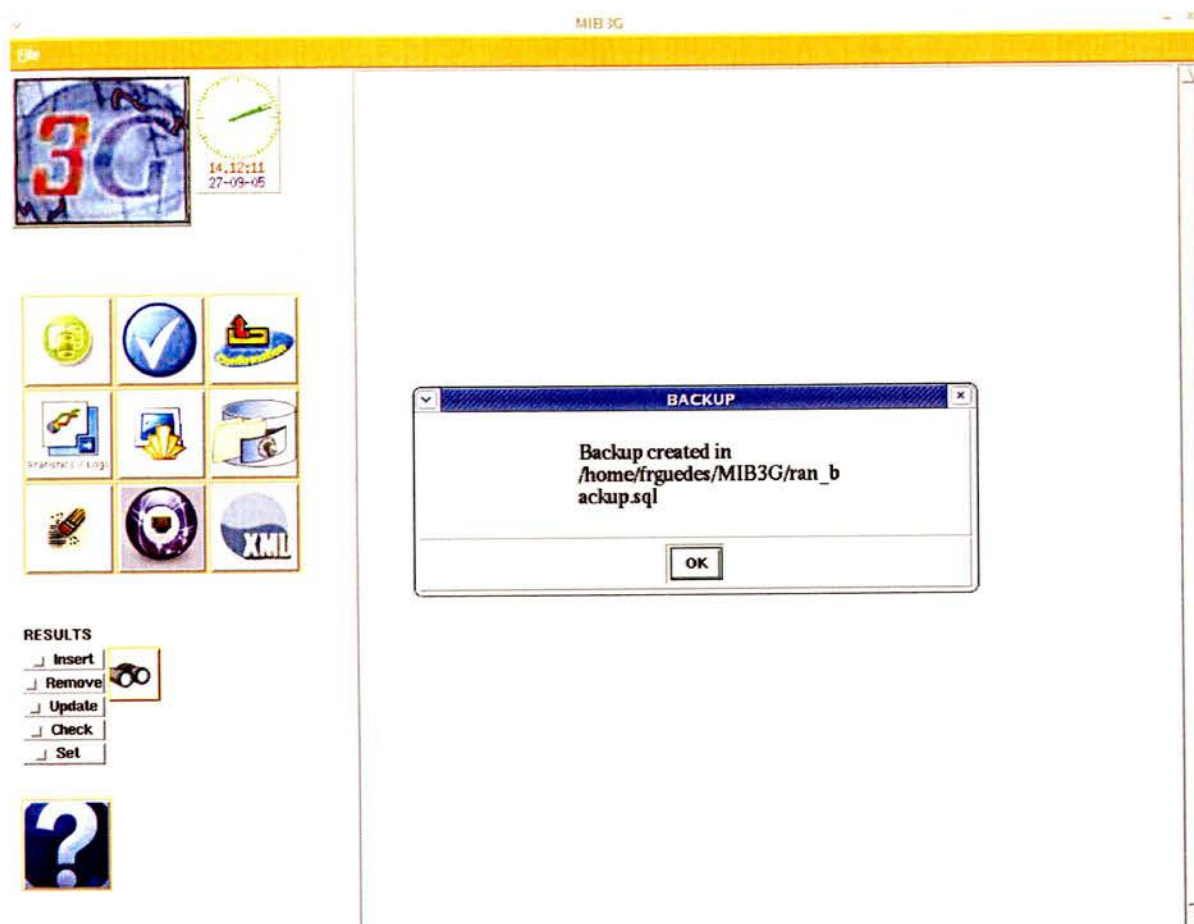


Figura 9. Backup da base de dados

5 Implementação da solução proposta

Este capítulo destina-se às especificações de implementação da solução apresentada no capítulo anterior, pelo que a sua organização tende a seguir o mesmo modelo.

É normal que num qualquer processo de desenvolvimento de software, o programador depare com situações inesperadas no decurso da implementação da solução, nunca sendo possível manter o desenvolvimento com a mesma cadência de resultados, mesmo que se recorra a estudos prévios exaustivos.

Explicam-se igualmente as dificuldades encontradas e, quando oportuno e possível, demonstram-se as funções e métodos que solucionaram tais barreiras.

O autor apenas apresenta algumas das funções utilizadas, pois a maioria tratam-se de funções cujo sigilo profissional deverá ser mantido.

5.1 Interface

Toda a interface foi desenvolvida utilizando o módulo PERL-Tk.

Trata-se de uma extensão explicitamente desenvolvida para o uso de interfaces, daí o seu fácil manuseamento e complexidade.

O autor optou por anexar o código da interface pela sua possível utilidade para o leitor.

Ver anexo A.

5.2 Cron

Paralelamente a toda a aplicação foi igualmente implementado um script que permite ao utilizador, sem qualquer tipo de intervenção deste, realizar uma operação diária de encher a base de dados com toda a informação da rede. O seu uso torna-se completamente transparente para o utilizador e apenas terá que definir uma tarefa denominada cron usando o programa crontab do unix apontando para o script. O código desta função esta apresentado no anexo B.

5.3 Base de dados

Na implementação da base de dados o autor deparou-se com varias dificuldades sobretudo relativo com o número gigantesco de campos a utilizar. Para combater esta dificuldade foi usado um ficheiro SQL (ver anexo C) que é carregado dinamicamente cada vez que a aplicação o exige.

Devido á grande complexidade das consultas usadas foram usados índices nos campos em que essa procura é mais intensa.

Como o leitor deve imaginar estamos a lidar com mais de 100 tabelas, com referencias entre elas o que exigiu por parte do autor uma grande minuciosidade e atenção para que nenhum campo pudesse ficar esquecido.

5.4 Ficheiro de entrada

Aquilo que iremos denominar de ficheiro de entrada, será o ficheiro que contém toda a parameterização necessária para fazer subir, por exemplo, um RNC.

Toda a informação disponível neste ficheiro de entrada terá de ser processada de maneira que seja possível reconhecer o seu padrão, sintaxe e semântica, e futura interacção com a base de dados.

```

1 //System RNCID: 51:
2 SET SYS: SYSDESC = "RNC51",SYSOBJECTID = "UPT001",SYSLOCATION = "Porto-SHO",SYSSERVICES = "WCDMA BGA";
3 ADD RNCBASIC: RNCID = 51,MCC = "268",MNC = "03";
4 SET WMPUIP: WMPUIPADDR = "10.75.113.41",MASK = "255.255.255.0",BAHIPADDR = "10.75.113.9",CONFIRM = Y;
5 SET WLPUSUBT: SN = 2,SUBN = 0,SUBT = STM-1;
6 SET WLPUSUBT: SN = 2,SUBN = 1,SUBT = STM-1;
7 SET WLPUSUBT: SN = 2,SUBN = 2,SUBT = STM-1;
8 SET WLPUSUBT: SN = 2,SUBN = 3,SUBT = STM-1;
9 SET WLPUSUBT: SN = 3,SUBN = 0,SUBT = STM-1;
10 SET WLPUSUBT: SN = 3,SUBN = 1,SUBT = STM-1;
11 SET WLPUSUBT: SN = 3,SUBN = 2,SUBT = STM-1;
12 SET WLPUSUBT: SN = 3,SUBN = 3,SUBT = STM-1;
13 SET WLPUSUBT: SN = 4,SUBN = 0,SUBT = STM-1;
14 SET WLPUSUBT: SN = 4,SUBN = 1,SUBT = STM-1;
15 SET WLPUSUBT: SN = 4,SUBN = 2,SUBT = STM-1;
16 SET WLPUSUBT: SN = 4,SUBN = 3,SUBT = STM-1;
17 SET WLPUSUBT: SN = 5,SUBN = 0,SUBT = STM-1;
18 SET WLPUSUBT: SN = 5,SUBN = 1,SUBT = STM-1;
19 SET WLPUSUBT: SN = 5,SUBN = 2,SUBT = STM-1;
20 SET WLPUSUBT: SN = 5,SUBN = 3,SUBT = STM-1;
21 SET WLPUSUBT: SN = 6,SUBN = 0,SUBT = STM-1;
22 SET WLPUSUBT: SN = 6,SUBN = 1,SUBT = STM-1;
23 SET WLPUSUBT: SN = 6,SUBN = 2,SUBT = STM-1;
24 SET WLPUSUBT: SN = 6,SUBN = 3,SUBT = STM-1;
25 SET WLPUSUBT: SN = 9,SUBN = 0,SUBT = STM-1;
26 SET WLPUSUBT: SN = 9,SUBN = 1,SUBT = STM-1;
27 SET WLPUSUBT: SN = 9,SUBN = 2,SUBT = STM-1;
28 SET WLPUSUBT: SN = 9,SUBN = 3,SUBT = STM-1;
29 SET OPT: OPT = SONET;
30 ADD REDPORT: LSN = 2,LPN = 0,HSN = 3,HFN = 0,REDT = EXTERNAL;
31 ADD REDPORT: LSN = 2,LPN = 1,HSN = 3,HFN = 1,REDT = EXTERNAL;
32 ADD REDPORT: LSN = 2,LPN = 7,HSN = 3,HFN = 7,REDT = EXTERNAL;
33 ADD REDPORT: LSN = 2,LPN = 15,HSN = 3,HFN = 15,REDT = INTERNAL;
34 ADD SUBRACK: SRN = 1,SRNAME = "WRBS1",WLPUSN = 2,WLPUPN = 15,WXIET = WOSE,REDT = NO;
35 SET RFNSRC: SRN = 1;
36 SET PWRALNSW: SRN = 1,PWRALMSSW = BIT0-14BIT1-14BIT2-14BIT3-14BIT6-14BIT7-14BIT8-14BIT9-14BIT12-14BIT13-14BIT14-14BIT15-14BIT16;
37 SET PWRLED: SRN = 1,PWRMONLED1SW = OFF,PWRMONLED2SW = OFF,PWRMONLED3SW = OFF;
38 SET PWRBUZZERSW: SRN = 1,PWRMONBUZZERSW = ON;
39 SET PWRPARA: SRN = 1,NO1M48ALMTHDUP = -57,NO1M48ALMTHDDOWN = -43,NO2M48ALMTHDUP = -57,NO2M48ALMTHDDOWN = -43,TEMPERATUREALMTHDU;
40 SET FANSPEED: SRN = 1,FANSPEEDMODE = AUTO_TEMPERATURE_CONTROL;
41 SET FANSPEED: SRN = 2,FANSPEEDMODE = AUTO_TEMPERATURE_CONTROL;
42 SET FANALNSW: SRN = 1,FANALMSSW = BIT0-14BIT1-14BIT3-14BIT4-14BIT5-14BIT6-14BIT7-1;
43 SET FANALNSW: SRN = 2,FANALMSSW = BIT0-14BIT1-14BIT3-14BIT4-14BIT5-14BIT6-14BIT7-1;
44 SET CLK: SRN = 1,SN = 0,BT = WOSE,REF2CLKSRC = 15,E1T1CLKSRC = BACKPLANE_CLOCK,E1T1OPTICALCLKSRC = BACKPLANE_CLOCK,CESCLKSW = ;
45 SET CLK: SRN = 1,SN = 15,BT = WOSE,REF2CLKSRC = 15,E1T1CLKSRC = BACKPLANE_CLOCK,E1T1OPTICALCLKSRC = BACKPLANE_CLOCK,CESCLKSW = ;

```

Figura 10. Exemplo de um ficheiro de entrada

5.5 Parsing

O método de parse foi sem dúvida a mais complexa operação no desenvolvimento deste projecto. Não só pelo grande número de parâmetros que são necessários verificar como também a sua interacção com a base de dados.

Antes de explicarmos todas as fases de parse, autor tentara neste resumo dar uma ideia do ciclo de vida para uma linha de entrada.

Tendo um ficheiro de entrada, que contem uma série de configurações (linhas) capazes, por exemplo, de fazer subir um RNC, o motor de parse terá de percorrer linha a linha todo o ficheiro de entrada e ser capaz de identificar para cada linha qual a operação a realizar na base de dados.

Assim sendo o autor tentara de forma breve, explicar todo esse mecanismo.

Desta forma podemos dividir esse mecanismo em 3 fases

5.5.1 1ª Fase do Parser

Nesta primeira fase o parser esta encarregue de analisar o tipo de comando da linha do ficheiro de entrada, no caso poderá ser, um ADD, SET, RMV ou MOD. Para cada um destes tipos de comandos existem parâmetros obrigatórios e opcionais que o parser terá de prever e inspeccionar.

5.5.2 2ª Fase do Parser

Posteriormente será feito um reconhecimento parâmetro principal que estamos a lidar. Esta operação é igualmente importante pois com ela o parser será capaz de identificar posteriormente quais os sub-parâmetros que pertencem a este parâmetro principal.

5.5.3 3ª Fase do Parser

Nesta fase um pouco mais complexa o parser terá de analisar todos os sub-parâmetros pertencentes ao parâmetro principal sejam estes mandatórios ou opcionais. Para isso, foi usada uma função para cada tipo de comando.

No caso, por exemplo, de ser detectado um ADD NODEB, existe uma função que pesquisa toda a linha de entrada desse comando e separa os seus sub-parâmetros e respectivos valores.

Isto fez-se porque o formato da linha da entrada como vimos, é um formato standard, mudando só o nome e o seu valor.

Recorrendo á ferramenta auxiliar LMT o parser consegue para esse parâmetro principal (no caso NODEB) identificar os parâmetros obrigatórios, ou seja, os parâmetros que têm obrigatoriamente estar presentes na linha.

```
sub Parser_add_NODEB
{
    my($line,$rncid,$opcao)=@_;
    my(@args)=();
    my(@fields)=();
    my(@values)=();
    #ADD NODEB: NODEBNAME="teste12". SRN=1, SSN=1;
    if($line =~ /ADD NODEB: NODEBNAME = \"(.+)\", SRN = (\d+), SSN = (\d+)/)
    {
        @args=split(/\./,$line);
        my($fail)=0;
        for(my($i)=1;$i<@args;$i++)
        {
            #print "$args[$i]\n";
            if($args[$i] =~ /[^\w]+) = ([^\w]+)/)
            {
                $args[$i]=$1;
                $fields[$i]=$1;
                $values[$i]=$2;
                for($fields[$i])
                {
                    s/$fields[$i]/$fields[$i],/
                }
                for($values[$i])
                {
                    s/$values[$i]/$values[$i],/
                }
            }
            elsif($args[$i] =~ /[^\w]+) = \"(.+)\")/)
            {
                $args[$i]=$1;
                $fields[$i]=$1;
                $values[$i]=$2;
                for($fields[$i])
                {
                    s/$fields[$i]/$fields[$i],/
                }
                for($values[$i])
                {
                    s/$values[$i]/$values[$i],/
                }
            }
            else
            {
                print "failure on NODEB $line\n";
                print LOG_ADD "ERROR\tFailure on NODEB line $line\n";
            }
        }
    }
    Parser_Range_NODEB(\@fields,\@values,$rncid,$line,$opcao);
}
```

Figura 11. Função Parser do comando ADD

5.6 Validação de Parâmetros

Depois de concluído todo o mecanismo de parse, a aplicação ira tratar de validar todos os parâmetros e sub-parâmetros do ficheiro de entrada.

Isto é conseguido através de um ficheiro de configuração que contem todas as definições e valores permitidos para todos os parâmetros. Assim sendo, o programa ira comparar os valores dos parâmetros do ficheiro de entrada com a gama de valores permitidos do ficheiro de configuração e retornar dessa forma a possibilidade ou não de interação com a base de dados.

PARAMETERS VALUE RANGES

```
CELLRLACTIME: CELLID = 0-268435455; ACTTIMEDEFOFFVAL = 1-255; ACTTIMEDEFOFFVALH0 = 1-255; RNCID = 51,52,60

AICH: CELLID = 0-268435455; PRACHPHYCHID = 0-255; PHYCHID = 0-255; AICHTXTIMING = 0-1; STTDIND = TRUE,FALSE; R

BCH: CELLID = 0-268435455; TRCHID = 0-255; BCHPOWER = -350-150; RNCID = 51, 52, 60

CELLACCESSSTRICT: CELLID = 0-268435455; CELLRESERVEDFOROPERATORUSE = RESERVED, NOT_RESERVED; CELLRESERVATIONEX

CELLALGOSWITCH: CELLID = 0-268435455; NBMSWITCH = INTRA_FREQUENCY_LDB, INTER_FREQUENCY_LDB, PUC, LCC, DOFFC, C

CELLAMRC: CELLID = 0-268435455; ULMEASFILTERCOEF = D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D11, D13, D15, D17,

CELLCAC: CELLID = 0-268435455; CELLENVTYPE = CASE1, CASE2, CASE3; ORTHOFACTOR = 0-1000; ULCONVTHD = 0-1000; UL

CELLCMCF: CELLID = 0-268435455; CMCFCELLTYPE = WALKING_SPEED_AND_HOT_SPOT_CELL, MID_SPEED_AND_HOT_SPOT_CELL, H

CELLDCCC: CELLID = 0-268435455; TCPMRFILTERCOEF = D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D11, D13, D15, D17,

CELLFRC: CELLID = 0-268435455; ULPSTRAFFDECTHS = D8, D32, D64, D128, D144, D256, D384; DLPSTRAFFDECTHS = D8, D

CELLINTERFREQH0: CELLID = 0-268435455; FILTERCOEF = D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D11, D13, D15, D17

CELLINTERRATH0: CELLID = 0-268435455; FILTERCOEF = D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D11, D13, D15, D17,

CELLINTRAFFREQH0: CELLID = 0-268435455; FILTERCOEF = D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D11, D13, D15, D17

CELLLCC: CELLID = 0-268435455; LCCMRTHD1 = 0-100; LCCMRTHD2 = 0-100; RNCID = 51, 52, 60
```

Figura 12. Ficheiro de validação de parâmetros.

Como podemos ver, para cada parâmetro principal estão definidos os valores possíveis para todos os seus sub-parâmetros.

5.6.1 Parse da gama de valores

Esta operação é conseguida, recorrendo a uma função que examina os novos valores a inserir pelo utilizador com os permitidos representados no ficheiro. De realçar que este ficheiro é totalmente configurável, ou seja, poderá ser alterado sempre que desejável.

```
sub Parser_Range_CELL
{
    #log_check_error='/home/irguedes/MIB3G/logs/log_check_error.txt';
    $log_check='/home/irguedes/MIB3G/logs/log_check.txt';
    open(LOG_CHECK,">>{$log_check}") || die("Cannot open file log_check.txt");

    print"\n***** CHECKING CELL PARAMETERS ***** \n";
    print LOG_CHECK "\n***** CHECKING CELL PARAMETERS ***** \n";

    $range='/home/irguedes/MIB3G/range.txt';

    ##### Open dos Ficheiros #####
    open(RANGEDATA,"<{$range}") || die("Cannot open file range.txt");
    @data=<RANGEDATA>;
    my($fields,$values,$rncid,$line,$opcao)=@_;
    my($check_flag2)=0;
    @columns=@$fields;
    @values2=@$values;
    @values=@$values;
    print LOG_CHECK "CHECKING LINE:\n$line\n";
    foreach $line (@data)
    {
        if ($line =~ /CELL: /)
        {
            my(@args)=();
            @args=split(/\s:/,$line);
            for (my($i)=1;$i<@args;$i++)
            {
                #print "$args[$i]\n";
                if($args[$i] =~ /([-\w]+) = (.+)/)
                {
                    $args[$i]=$1;
                    $range_columns[$i]=$1;
                    $range_values[$i]=$2;
                }
            }

            $query=("insert into NODEB(@columns,RNCID) values (@values,$rncid)");
            $sth = $dbh->prepare($query) or die "Unable to prepare: ". $dbh->errstr."\n";
            print"QUERY\t$query\n";
            print LOG_INSERT "QUERY\t$query\n";
            if(!$sth->execute)
            {
                print"ERROR\t" . $dbh->errstr."\n";
                print LOG_INSERT "ERROR in query $query\t" . $dbh->errstr."\n";
            }
            else
            {
                print"OK\tQUERY INSERTED\n";
                print LOG_INSERT "OK\tQUERY INSERTED\n";
            }
        }
    }
}
```

Figura 13. Função que examina gama de valores

5.7 Interação com Base de dados

Depois de concluído todo o processo de identificação de uma linha de entrada a aplicação terá de interagir com a base de dados e proceder á operação que lhe foi destinada.

Desta forma, existem funções que foram implementadas para este propósito, de maneira que, possa ser possível, em qualquer altura visualizar toda essa informação na base de dados.

Assim sendo, o autor irá apresentar as três principais funções de interação com a base de dados para que seja possível a sua compreensão.

5.7.1 Função de Inserção

Esta função, talvez a mais simples delas todas irá inserir os novos valores de entrada na tabela correspondente ao parâmetro principal que estamos a inserir. Por exemplo, para a linha:

ADD NODEB: NODEBNAME = "PORTO_AV_REPUBLICA_UMTS",SRN = 1,SSN = 0;

A função é a seguinte:

```
$query=("insert into NODEB(@columns,RNCID) values (@values,$rncid)");
$sth = $dbh->prepare($query) or die "Unable to prepare: ". $dbh->errstr."\n";
print "QUERY\t$query\n";
print LOG_INSERT "QUERY\t$query\n";
if(!$sth->execute)
{
    print "ERROR\t" . $dbh->errstr. "\n";
    print LOG_INSERT "ERROR in query $query\t" . $dbh->errstr. "\n";
}
else
{
    print "OK\tQUERY INSERTED\n";
    print LOG_INSERT "OK\tQUERY INSERTED\n";
}
```

Figura 14. Função de inserção de valores

Como podemos perceber pela função acima representada, a função o que faz é inserir os novos valores na tabela NODEB nas colunas correspondentes. Para além disto, é feito um controle rigoroso de todos os possíveis erros para o log da função. Isto torna-se extremamente útil para efeitos de depuração de erros.

5.7.2 Função de Remoção

```

$query=("delete from NODES where NODEENAME = '$values[0]'");
$stmt = $dbh->prepare($query) or die "Unable to prepare: ". $dbh->errstr."\n";
print "QUERY\t$query\n";
print LOG_REMOVE "QUERY\t$query\n";
if(!$stmt->execute)
{
    print "ERROR\t" . $dbh->errstr."\n";
    print LOG_REMOVE "ERROR\t" . $dbh->errstr."\n";
}
else
{
    print "OK\tQUERY INSERTED\n";
    print LOG_REMOVE "OK\tQUERY INSERTED\n";
}

```

Figura 15. Função de remoção de valores

A única diferença desta função para a anterior esta no tipo de comando a interagir com a base de dados. Neste caso lidamos com uma remoção e tal como a anterior, obrigatoriamente é feito um log de todos os possíveis erros.

5.7.3 Função de Modificação

```

$query=("select NODENAME from NODES where NODEID = $values[0] and PNCD = $pnoid");
print "$query\n";
print LOG_UPDATE "QUERY\t$query\n";
$stmt = $dbh->prepare($query);
$stmt->execute();
$rows=$stmt->fetchrow_array;
if($row[0])
{
    if($row[0] ne $values[1])
    {
        $query2=("update NODES set NODENAME= '$values[1]' where NODEID = $values[0] and PNCD = $pnoid");
        print "$query2\n";
        print LOG_UPDATE "QUERY\t$query2\n";
        $stmt = $dbh->prepare($query2);
        print "Updating value $values[1] in column $columns[1]\n";

        if(!$stmt->execute)
        {
            print "ERROR\t" . $dbh->errstr . "\n";
            print LOG_UPDATE "ERROR in query $query2\t" . $dbh->errstr . "\n";
        }
        else
        {
            print "OK\tQUERY INSERTED\n";
            print LOG_UPDATE "OK\tQUERY INSERTED\n";
        }
    }
    else
    {
        print "WARNING\tValue of $values[1] already exists in database...\n";
        print LOG_UPDATE "WARNING\tValue of $values[1] already exists in database...\n";
    }
}
else
{
    print "ERROR\tValue must exist in database\n";
}

```

Figura 16. Função de modificação de valores

Esta função bastante mais complexa, consiste em guardar numa estrutura todos valores actuais presentes na base de dados para uma determinada tabela, e comparar esses valores como os novos que vão ser inseridos. No caso destes valores serem diferentes faz a respectiva modificação na base de dados.

De salientar que para esta operação poder ser realizada terá que existir sempre um elemento comum, que será a chave primária.

6 Validação de Requisitos

A validação de requisitos é uma actividade de extrema importância, uma vez que é possível garantir a qualidade do processo de desenvolvimento do produto de software através de um documento de especificação claro, correcto e sem problemas de ambiguidade.

Conhecendo esses factores e as necessidades da empresa Optimus SA em constante melhoria no processo de elaboração e validação de requisitos surge este capítulo que tem como objectivo demonstrar ao leitor de que forma os requisitos foram validados.

Dentre as técnicas de detecção e remoção de desconformidades, o autor tentou através de testes exaustivos, detectar requisitos inconsistentes e incorrectos no sistema antes destes se tornarem base para o projecto e implementação, evitando fazer tudo de novo.

Assim sendo ao longo de todo o projecto, os testes foram efectuados de forma controlada e sistemática, definindo-se um “guião” de cenários para capturar um conjunto vasto de configurações de maneira que fosse possível depurar a existência de possíveis erros.

Neste caso, e como se torna impossível prever quais as operações previstas no ficheiro de entrada, os testes realizados contêm sempre as três operações possíveis, ou seja, ADD, RMV e MOD, referentes á inserção, remoção e modificação de parâmetros.

O volume do ficheiro de entrada e as suas configurações foram as variáveis alteradas durante a medição desta grandeza, por forma a que obtivéssemos os resultados que espelhassem a realidade, o mais aproximadamente possível.

Desta forma foi elaborado um gráfico, apenas para mera ilustração de maneira que o leitor fica com uma ideia da performance da aplicação mais concretamente de todo o mecanismo de parser.

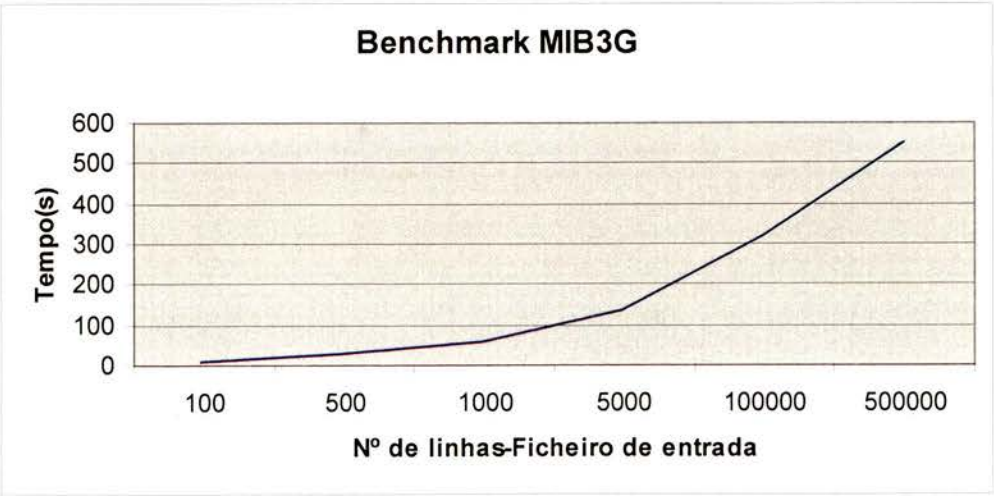


Figura 17. Gráfico de Benchmark MIB3G

De referir mais uma vez que este capítulo trata uma pequena amostra dos vários testes que foram realizados, ao longo do projecto e que a maior parte deles apenas são visíveis com a própria execução da aplicação.

7 Conclusões

O espaço do capítulo conclusivo deste documento reserva-se para se tecerem as considerações finais que podem ser retidas desta experiência de estágio curricular, focando-se o nível de sucesso atingido no cumprimento dos objectivos, numa revisão das dificuldades encontradas, resumindo-se também as vantagens e utilidade do produto desenvolvido para a instituição de estágio e seus clientes

Serão igualmente abordados aspectos de ordem técnica, como as ilações que se podem retirar da tecnologia usada no trabalho, assim como aspectos pessoais que dizem respeito ao quotidiano laboral em que este projecto se inseriu, qualificando-se as condições, recursos e ambiente de trabalho proporcionados.

Assim, do ambiente de trabalho encontrado e dos recursos disponibilizados só se pode aferir o melhor. Desde as ferramentas de trabalho, ao apoio, formação e integração no seio da equipa, tudo é merecedor de um apreço imenso, ao qual apenas se pode corresponder com trabalho, alegria e empenho.

Finalmente, resta aferir acerca do funcionamento do estágio curricular em si, experiência que em primeira instância se pode considerar muito positiva e enriquecedora no geral, pelo que se endereçam neste momento felicitações aos responsáveis máximos pela LEIC, ao manterem esta aposta como componente curricular mais importante. Para quem sai do ambiente académico é muito importante poder contar com um período de integração na vida profissional, segundo uma metodologia progressiva de assimilação dos processos normais ao funcionamento diário das empresas.

Logicamente que alguns reparos poderão ser feitos, mas também é certo que não há modelo algum que sirva o todo incondicionalmente. Neste sentido, prefere-se deixar uma palavra de motivação para a continuidade do bom trabalho nas várias instâncias da LEIC.

Referências e Bibliografia

PERL Praticial Extracting and Report Language.

<http://www-cgi.cs.cmu.edu/cgi-bin/perl-man>

Motorola UMTS Customers Manual USR1.0

Motorola UMTS Customers Manual USR2.0

MySQL Reference Manual.

<http://dev.mysql.com/doc/mysql/en/index.html>

Comprehensive Perl Archive Network.

<http://www.cpan.org>

Perl TK Home Page.

<http://www.perltk.org>

Sun Microsystems.

<http://www.sun.com>

McGraw-Hill (2001) Perl – The Complete Reference.

O'Reilly (2002) Perl Cookbook

ANEXO A: Código da Interface

```
1: #!/usr/local/bin/perl
2:
3:
4: BEGIN
5: {
6:     push(@INC, "/home/fguedes/MIB3G");
7: }
8:
9:
10: #use strict;
11: use Tk;
12: use Tk::JPEG;
13: use Tk::Photo;
14: use Tk::Clock;
15: use Tk::FileSelect;
16: use Tk::Dialog;
17: use Tk::Balloon;
18: use Tk::ProgressBar;
19: use Tk::DialogBox;
20: use Tk::Scrollbar;
21: use Term::ReadKey;
22: use Term::ANSIColor;
23: use parser_file qw(&Parser_File);
24:
25:
26: system "clear";
27: $path='/home/fguedes/MIB3G/gui_mib3g.pl';
28: open(LOCKPATH, "<< $path") or die "can't open filename: $!";
29: flock(LOCKPATH, 2);
30: $value_commit = 0;
31:
32:
33: $mw->Button(-text => 'Exit', -command => sub { exit },
34:             #-image => $image)->place(-x => 0, -y => 27);
35:
36:
37: #### Verify User
38: print "Enter your root password: ";
39: ReadMode 'noecho';
40: $password = ReadLine 0;
41: chomp $password;
42: ReadMode 'normal';
43:
44: print "\n";
45:
46: ($username, $encrypted) = ( getpwuid $< )[0,1];
47:
48: if (crypt($password, $encrypted) ne $encrypted)
49: {
50:     die "You are not $username\n";
51: }
52: else
53: {
54:     print "Welcome $username to MIB3G\n";
55:     my $mw = MainWindow->new();
56:     $mw->geometry('0x0');
57:     $image = $mw->Photo(-file => "logo2.gif");
58:     $dialog = $mw->Dialog(-image => $image, -text => "Please Choose:\n", -bitmap
=> 'question', -title => 'MIB3G', -default_button => 'GUI', -buttons=>[ "GUI",
"ADMIN TOOL" ] );
59:     $result = $dialog->Show;
60:     #chomp( $result = <STDIN> );
61:     if($result eq 'GUI')
62:     {
63:         destroy $mw;
64:         gui();
65:     }
66:     else
```

```
: {
:     destroy $mw;
:     system "/usr/local/bin/perl admin.pl";
: }
:
:
: ### GUI ###
: sub gui
: {
:
:     ##ERASE ALL THE DATA IN THE VIRTUAL DATABASE##
:
:     print"DROPPING AND CREATING VIRTUAL DATABASE...PLEASE WAIT\n";
:     my $dbh = DBI->connect("DBI:mysql:database=ran_virtual;host=localhost",
:                           "frguedes", "riutort",
:                           {'RaiseError' => 1});
:     $query=("drop database ran_virtual");
:     $query2=("create database ran_virtual");
:     $sth = $dbh->prepare($query);
:     $sth2 = $dbh->prepare($query2);
:     $sth->execute();
:     $sth2->execute();
:     $ranb='/home/frguedes/MIB3G/temp/ran_b.sql';
:     open(RANB,">$ranb") || die("Cannot open file ran_b.sql");
:     print"RECONSTRUCTING DATABASE DATA AND STRUCTURE...PLEASE WAIT\n";
:     system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual < tabelas.sql");
:     system("/usr/local/mysql/bin/mysqldump -uroot -pmib3g --no-create-info ran >
: '/home/frguedes/MIB3G/temp/ran_b.sql'");
:     system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual <
: '/home/frguedes/MIB3G/temp/ran_b.sql'");
:     print"ALL OK\n";
:     sleep(1);
:
:     #Create the main window
:
:     my $mw = MainWindow->new(-relief => 'groove', -title=>'MIB3G');
:     $mw->geometry('1024x768');
:     $mw->bind('<Configure>' => sub {
:         $xe = $mw->XEvent;
:         $mw->maxsize($xe->w, $xe->h);
:         $mw->minsize($xe->w, $xe->h);
:     });
:
:     # Create the menubar
:     my $menubar = $mw->Frame(-relief => 'raised',
:                             -background => 'orange',
:                             -borderwidth => 2
:     )->pack(-side => 'top', -fill => 'x');
:
:     # Create the menubuttons
:     my $menu file = $menubar->Menubutton(-text => 'File',
:                                           -background => 'orange',
:                                           -foreground => 'white',
:                                           -underline => 0,
:                                           -tearoff => 0
:     )->pack(-side => 'left');
:
:     # File menu items
:     $menu file->command(-label => 'Open logs',
:                       -command => \&open_file,
```

```
133:         -underline => 1);
134:
135:
136: $menu file->command(-label => 'LOAD CFGMML',
137:         -command => \&load_cfgmml,
138:         -underline => 1);
139:
140: $menu file->command(-label => 'ERASE CFGMML',
141:         -command => \&erase_cfgmml,
142:         -underline => 1);
143:
144: $menu_file->separator();
145:
146: $menu file->command(-label => 'Exit',
147:         -command => \&quit,
148:         -underline => 1);
149:
150:
151:
152:
153:
154: #Area de texto
155:
156: my $scrollbar = $mw->Scrollbar( );
157: my $text = $mw->Text(-width => 100,
158:         -background=>'white',
159:         -height => 43,
160:         -yscrollcommand => ['set' => $scrollbar]);
161:
162: $scrollbar->configure(-command => ['yview' => $text]);
163:
164: $scrollbar->pack(-side => 'right', -fill => 'y');
165: $text->pack(-side => 'right', -fill => 'both');
166:
167:
168: $text->pack( -fill => "both", -expand => 1 );
169:
170: ##Imagem 3G##
171:
172: $image_3g = $mw->Photo(-file => "3g 2.jpg");
173: $mw->Button(-image => $image_3g)->place(-x => 0, -y => 30);
174:
175: #my $image2 = $mw->Label(-image => $image);
176:
177:
178: #Buttons
179:
180: $image insert = $mw->Photo(-file => 'insert.jpg');
181: $image logs = $mw->Photo(-file => 'logs.jpg');
182: $image check = $mw->Photo(-file => 'check.jpg');
183: $image erase = $mw->Photo(-file => 'erase.jpg');
184: $image xterm = $mw->Photo(-file => 'xterm.jpg');
185: $image connect = $mw->Photo(-file => 'connect.jpg');
186: $image help = $mw->Photo(-file => 'help.jpg');
187: $image commit = $mw->Photo(-file => 'commit.jpg');
188: $image backup = $mw->Photo(-file => 'backup.jpg');
189: $image xml = $mw->Photo(-file => 'xml.jpg');
190: $image_errors = $mw->Photo(-file => 'errors.jpg');
191:
192:
193: $button insert = $mw->Button(-text => 'INSERT IN DATABASE', -width => 70, -
height => 70, -image => $image insert,-relief => 'raised',
194:         -command => \&FileChooser_Insert,-highlightbackground => 'orange') -
>place(-x => 10, -y => 220);
195:
196: $button check = $mw->Button(-text => "PARAMETERS\nCHECK",-width => 70, -height
=> 70, -image => $image check,-relief => 'raised',
197:         -command => \&FileChooser_Check,-highlightbackground => 'orange')-
>place(-x => 90, -y => 220);
```

```

198:
199: $button logs = $mw->Button(-text => "LOGS", -width => 70, -height => 70, -image
=> $image logs, -relief => 'raised',
200:     -command => \&ReadLogs, -highlightbackground => 'orange')->place(-x
=> 10, -y => 300);
201:
202: $button xterm = $mw->Button(-text => "XTERMS", -width => 70, -height => 70, -
image => $image xterm, -relief => 'raised',
203:     -command => \&xterm, -highlightbackground => 'orange')->place(-x =>
90, -y => 300);
204:
205: $button erase = $mw->Button(-text => "Clear Screen", -width => 70, -height => 70,
-image => $image erase, -relief => 'raised',
206:     -command => \&erase, -highlightbackground => 'orange')->place(-x =>
10, -y => 380);
207:
208: $button connect = $mw->Button(-text => "Connect", -width => 70, -height => 70, -
image => $image connect, -relief => 'raised',
209:     -command => \&connect_mysql, -highlightbackground => 'orange')-
>place(-x => 90, -y => 380);
210:
211: $button help = $mw->Button(-text => "Help", -width => 70, -height => 70, -image
=> $image help, -relief => 'raised',
212:     -command => \&help_about, -highlightbackground => 'orange')->place(-x
=> 5, -y => 650);
213:
214: $button commit = $mw->Button(-text => "Commit", -width => 70, -height => 70, -
image => $image commit, -relief => 'raised',
215:     -command => \&commit, -highlightbackground => 'orange')->place(-x =>
170, -y => 220);
216:
217: $button backup = $mw->Button(-text => "Backup", -width => 70, -height => 70, -
image => $image backup, -relief => 'raised',
218:     -command => \&backup, -highlightbackground => 'orange')->place(-x =>
170, -y => 300);
219:
220: $button xml = $mw->Button(-text => "Rollback", -width => 70, -height => 70, -
image => $image xml, -relief => 'raised',
221:     -command => \&export_xml, -highlightbackground => 'orange')->place(-x
=> 170, -y => 380);
222:
223: $button errors = $mw->Button(-text => "ERRORS", -width => 50, -height => 50, -
image => $image errors, -relief => 'raised',
224:     -command => \&errors, -highlightbackground => 'orange')->place(-x =>
140, -y => 520);
225:
226: ##Ballons##
227:
228: $balloon = $mw->Balloon();
229: $balloon->attach($button insert, -balloonmsg => "Insert file in database");
230: $balloon->attach($button check, -balloonmsg => "Parameters Check");
231: $balloon->attach($button logs, -balloonmsg => "Open Mainlog");
232: $balloon->attach($button xterm, -balloonmsg => "Open XTerm");
233: $balloon->attach($button connect, -balloonmsg => "Connecto to MySQL");
234: $balloon->attach($button erase, -balloonmsg => "Clear Screen");
235: $balloon->attach($button help, -balloonmsg => "Help...in construction");
236: $balloon->attach($button commit, -balloonmsg => "Commit all the changes");
237: $balloon->attach($button backup, -balloonmsg => "Backup database");
238: $balloon->attach($button xml, -balloonmsg => "Export to XML");
239: $balloon->attach($button_inject, -balloonmsg => "Load CFGMML to database");
240:
241:
242: ##CLOCK
243:
244:
245: my $clock = $mw->Clock()
246: ->place(-x => 160, -y => 32);
247:

```

```

248:
249:
250:
251:
252:
253: ##RadioButtons and View Button
254:
255:
256: $mw->Label(-text => "RESULTS")->place(-x => 5, -y => 500);
257: # $mw->Label(-text => "ERRORS")->place(-x => 140, -y => 500);
258:
259:
260: $mw->Checkbutton(-text => "Insert", -variable => \$checkboxbutton_insert,-relief =>
'raised',-width => 7,
261:     -command => \&Results)->place(-x => 5, -y => 520);
262:
263: $mw->Checkbutton(-text => "Remove", -variable => \$checkboxbutton_remove,-relief =>
'raised',-width => 7,
264:     -command => \&Results)->place(-x => 5, -y => 540);
265:
266: $mw->Checkbutton(-text => "Update", -variable => \$checkboxbutton_update,-relief =>
'raised',-width => 7,
267:     -command => \&Results)->place(-x => 5, -y => 560);
268:
269: $mw->Checkbutton(-text => "Check ", -variable => \$checkboxbutton_check,-relief =>
'raised',-width => 7,
270:     -command => \&Results)->place(-x => 5, -y => 580);
271:
272: $mw->Checkbutton(-text => "Set  ", -variable => \$checkboxbutton_set,-relief =>
'raised',-width => 7,
273:     -command => \&Results)->place(-x => 5, -y => 600);
274:
275:
276:
277: $image_view = $mw->Photo(-file => 'view.jpg');
278:
279: $button_view = $mw->Button(-text => 'VIEW LOG', -width => 40, -height => 40, -
image => $image_view,-relief => 'raised',
280:     -background => 'orange', -foreground => 'white',
281:     -command => \&Buttonview) ->place(-x => 80, -y => 520);
282:
283: $balloon->attach($button_view, -balloonmsg => "View Selected Log");
284:
285:
286: MainLoop;
287:
288:
289: sub errors
290: {
291:
292:
293:     system("grep ERROR '/home/fguedes/MIB3G/logs/log_insert.txt' >
'/home/fguedes/MIB3G/logs/insert_error.txt'");
294:     #system("grep WARNING log insert.txt > insert_warning.txt");
295:     $insert_error = '/home/fguedes/MIB3G/logs/insert_error.txt';
296:     open(INSERT_ERROR,"<$insert_error") || die("Cannot open file
insert_error.txt ");
297:     @insert_error=<INSERT_ERROR>;
298:     $text->delete('1.0', 'end');
299:     if(@insert_error)
300:     {
301:         $item = "***** INSERT ERRORS
*****\n";
302:         $text->insert('end', $item);
303:         foreach $line (@insert_error)
304:         {
305:             $item = $line;

```

```
306:         $text->insert('end', $item);
307:     }
308:
309: }
310: #system("grep ERROR log insert.txt > insert_error.txt");
311: #system("grep ERROR log _insert.txt > insert_error.txt");
312:
313:
314:
315:
316: }
317:
318:
319: ## Function to open any file ##
320: sub open_file
321: {
322:     $FSref = $mw->FileSelect(-directory => '/home/fguedes/MIB3G/logs', -height =>
323: 25, -width =>25);
324:     $file = $FSref->Show;
325:     if($file)
326:     {
327:         open(FILEDATA,$file) || die("Cannot open file $file");
328:         @datafile=<FILEDATA>;
329:         $text->delete('1.0', 'end');
330:         foreach $line (@datafile)
331:         {
332:             $item = $line;
333:             $text->insert('end', $item);
334:         }
335:     }
336:     else
337:     {
338:         print"Cancel\n";
339:     }
340: }
341: }
342:
343: sub load_cfgmml
344: {
345:     $result=$mw->Dialog(-title => 'COMMIT',-text => 'YOU SHOULD COMMIT ALL THE
346: CHANGES BEFORE LOAD CFGMML!CONTINUE?(WILL ERASE ALL DATABASES)', -default_button
347: => 'no',-bitmap => 'warning',-buttons=>[ "YES","NO"]->Show();
348:     if($result =~/YES/)
349:     {
350:         $cfgmml='/home/fguedes/MIB3G/temp/ran_cfgmml.sql';
351:         open(CFGMML,">$cfgmml") || die("Cannot open file cfgmml.sql");
352:
353:         $item = "ERASING DATABASES...PLEASE WAIT\n";
354:         $text->insert('end', $item);
355:         my $dbh = DBI->connect("DBI:mysql:database=ran_virtual;host=localhost",
356:                               "fguedes", "riutort",
357:                               {'RaiseError' => 1});
358:         $query=("drop database ran_virtual");
359:         $query2=("create database ran_virtual");
360:         $sth = $dbh->prepare($query);
361:         $sth2 = $dbh->prepare($query2);
362:         $sth->execute();
363:         $sth2->execute();
364:         system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual <
365: tabelas.sql");
366:         $FSref = $mw->FileSelect(-directory => '/home/fguedes/MIB3G/CFGMML', -
367: height => 25, -width =>25);
368:         $file = $FSref->Show;
369:         if($file)
370:         {
371:             $item = "LOADING...PLEASE WAIT!\n";
372:             $text->insert('end', $item);
373:         }
374:     }
375: }
```

```

369: open (example, $file) || die ("Could not open file $file!");
370: $system_rncid = <example>;
371: if($system_rncid =~ /\//System RNCID: (\d+)/)
372: {
373:     open(FILEDATA,$file) || die("Cannot open file $file");
374:     @datafile=<FILEDATA>;
375:     foreach $line (@datafile)
376:     {
377:         if($line =~ /\//System RNCID: (\d+)/)
378:         {
379:             $rncid = $1;
380:             $opcao = insert;
381:             Parser_File($file,$opcao,$rncid);
382:         }
383:     }
384: }
385: else
386: {
387:     $item = "\nNO System RNCID in header of file $file\n";
388:     $text->insert('end', $item);
389: }
390: }
391:
392: system("/usr/local/mysql/bin/mysqldump -uroot -pmib3g --no-create-info
ran virtual > '/home/fguedes/MIB3G/temp/ran_cfgmml.sql'");
393: system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_cfgmml <
'/home/fguedes/MIB3G/temp/ran_cfgmml.sql'");
394: check_logs();
395: log_geral();
396:
397: }
398: }
399:
400: sub erase_cfgmml
401: {
402:     my $dbh2 = DBI->connect("DBI:mysql:database=ran_cfgmml;host=localhost",
403:                             "fguedes", "riutort",
404:                             {'RaiseError' => 1});
405:     $query3=("drop database ran_cfgmml");
406:     $query4=("create database ran_cfgmml");
407:     $sth3 = $dbh2->prepare($query3);
408:     $sth4 = $dbh2->prepare($query4);
409:     $sth3->execute();
410:     $sth4->execute();
411:     system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_cfgmml < tabelas.sql");
412:     $item = "\nCFGMML ERASED FROM DATABASE\n";
413:     $text->insert('end', $item);
414: }
415:
416: ## Quit Function ##
417: sub quit
418: {
419:     if($value_commit eq 0)
420:     {
421:         $result=$mw->Dialog(-title => 'EXIT',-text => 'ALL CHANGES ARE NOT SAVED,
EXIT ANYWAY?', -default_button => 'NO',-bitmap => 'warning',-buttons=>[ "YES",
"NO"])->Show();
422:         if($result =~ /YES/)
423:         {
424:             $mw->deiconify;
425:             $mw->destroy;
426:         }
427:     }
428:     else
429:     {
430:         $mw->deiconify;
431:         $mw->destroy;
432:     }

```

```
433:
434: }
435:
436:
437: ## MySQL Connect ##
438: sub connect_mysql
439: {
440:
441:
442:     my $dbh = DBI->connect("DBI:mysql:database=ran;host=localhost",
443:                             "frguedes", "rlutort",
444:                             {'RaiseError' => 1});
445:
446: }
447: # View Button
448: sub Buttonview
449: {
450:     $value_button = 1;
451:     Results($value_button);
452: }
453:
454:
455: ## Check the results
456: sub Results
457: {
458:     if(($checkboxbutton_insert == 1)&&($value_button == 1))
459:     {
460:         $log_insert = '/home/frguedes/MIB3G/logs/log_insert.txt';
461:         $count_insert = `wc -l < $log_insert`;
462:         chomp($count_insert);
463:         $count_insert = $count_insert + 0;
464:         $item = open(LOG_INSERT,"<$log_insert");
465:         if($count_insert ne 0)
466:         {
467:             @log_insert=<LOG_INSERT>;
468:             $text->delete('1.0', 'end');
469:             foreach $line (@log_insert)
470:             {
471:                 $item = $line;
472:                 $text->insert('end', $item);
473:             }
474:             $checkboxbutton_insert = 0;
475:         }
476:     }
477:     else
478:     {
479:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
480: default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
481:         $checkboxbutton_insert = 0;
482:     }
483:     #clear_log_insert();
484: }
485: elseif(($checkboxbutton_check == 1)&&($value_button == 1))
486: {
487:     $log_check = '/home/frguedes/MIB3G/logs/log_check.txt';
488:     $count_check = `wc -l < $log_check`;
489:     chomp($count_check);
490:     $count_check = $count_check + 0;
491:     $item = open(LOG_CHECK,"<$log_check");
492:     if($count_check ne 0)
493:     {
494:         @log_check=<LOG CHECK>;
495:         $text->delete('1.0', 'end');
496:         foreach $line (@log_check)
497:         {
498:             $item = $line;
499:             $text->insert('end', $item);
500:         }
501:         $checkboxbutton_check = 0;
```

```
433:
434: }
435:
436:
437: ## MySQL Connect ##
438: sub connect_mysql
439: {
440:
441:
442:     my $dbh = DBI->connect("DBI:mysql:database=ran;host=localhost",
443:                             "frguedes", "riutort",
444:                             {'RaiseError' => 1});
445:
446: }
447: # View Button
448: sub Buttonview
449: {
450:     $value_button = 1;
451:     Results($value_button);
452: }
453:
454:
455: ## Check the results
456: sub Results
457: {
458:     if(($checkbutton_insert == 1)&&($value_button == 1))
459:     {
460:         $log_insert = '/home/frguedes/MIB3G/logs/log_insert.txt';
461:         $count_insert = `wc -l < $log_insert`;
462:         chomp($count_insert);
463:         $count_insert = $count_insert + 0;
464:         $item = open(LOG_INSERT,"<$log_insert");
465:         if($count_insert ne 0)
466:         {
467:             @log_insert=<LOG_INSERT>;
468:             $text->delete('1.0', 'end');
469:             foreach $line (@log_insert)
470:             {
471:                 $item = $line;
472:                 $text->insert('end', $item);
473:             }
474:             $checkbutton_insert = 0;
475:         }
476:     }
477:     else
478:     {
479:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
480: default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
481:         $checkbutton_insert = 0;
482:     }
483:     #clear_log_insert();
484: }
485: elseif(($checkbutton_check == 1)&&($value_button == 1))
486: {
487:     $log_check = '/home/frguedes/MIB3G/logs/log_check.txt';
488:     $count_check = `wc -l < $log_check`;
489:     chomp($count_check);
490:     $count_check = $count_check + 0;
491:     $item = open(LOG_CHECK,"<$log_check");
492:     if($count_check ne 0)
493:     {
494:         @log_check=<LOG CHECK>;
495:         $text->delete('1.0', 'end');
496:         foreach $line (@log_check)
497:         {
498:             $item = $line;
499:             $text->insert('end', $item);
500:         }
501:         $checkbutton_check = 0;
502:     }
503: }
```

```
500:     }
501:     else
502:     {
503:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
504:         $checkboxbutton_check = 0;
505:     }
506:     #clear_log_check();
507: }
508: elsif(($checkboxbutton_update == 1)&&($value_button == 1))
509: {
510:     $log_update = '/home/fguedes/MIB3G/logs/log_update.txt';
511:     $count_update = `wc -l < $log_update`;
512:     chomp($count_update);
513:     $count_update = $count_update + 0;
514:     $item = open(LOG_UPDATE,"<$log_update");
515:     if($count_update ne 0)
516:     {
517:         @log_update=<LOG_UPDATE>;
518:         $text->delete('1.0', 'end');
519:         foreach $line (@log_update)
520:         {
521:             $item = $line;
522:             $text->insert('end', $item);
523:         }
524:         $checkboxbutton_update = 0;
525:     }
526:     else
527:     {
528:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
529:         $checkboxbutton_update = 0;
530:     }
531:     #clear_log_update();
532: }
533: elsif(($checkboxbutton_remove == 1)&&($value_button == 1))
534: {
535:     $log_remove = '/home/fguedes/MIB3G/logs/log_remove.txt';
536:     $count_remove = `wc -l < $log_remove`;
537:     chomp($count_remove);
538:     $count_remove = $count_remove + 0;
539:     $item = open(LOG_REMOVE,"<$log_remove");
540:     if($count_remove ne 0)
541:     {
542:         @log_remove=<LOG_REMOVE>;
543:         $text->delete('1.0', 'end');
544:         foreach $line (@log_remove)
545:         {
546:             $item = $line;
547:             $text->insert('end', $item);
548:         }
549:         $checkboxbutton_remove = 0;
550:     }
551:     else
552:     {
553:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
554:         $checkboxbutton_remove = 0;
555:     }
556:     #clear_log_remove();
557: }
558: elsif(($checkboxbutton_set == 1)&&($value_button == 1))
559: {
560:     $log_set = '/home/fguedes/MIB3G/logs/log_set.txt';
561:     $count_set = `wc -l < $log_set`;
562:     chomp($count_set);
563:     $count_set = $count_set + 0;
564:     $item = open(LOG_SET,"<$log_set");
```

```

565:         if($count_set ne 0)
566:         {
567:             @log_set=<LOG SET>;
568:             $text->delete('1.0', 'end');
569:             foreach $line (@log_set)
570:             {
571:                 $item = $line;
572:                 $text->insert('end', $item);
573:             }
574:             $checkboxbutton_set = 0;
575:         }
576:     else
577:     {
578:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
579:         $checkboxbutton_set = 0;
580:     }
581:     #clear_log_remove();
582: }
583: elsif(($checkboxbutton_check_error == 1)&&($value_button == 1))
584: {
585:     $log_check_error = '/home/fguedes/MIB3G/logs/log_check_error.txt';
586:     $count_check_error = `wc -l < $log_check_error`;
587:     chomp($count_check_error);
588:     $count_check_error = $count_check_error + 0;
589:     $item = open(LOG CHECK ERROR,"<$log_check_error");
590:     if($count_check_error ne 0)
591:     {
592:         @log_check_error=<LOG CHECK ERROR>;
593:         $text->delete('1.0', 'end');
594:         foreach $line (@log_check_error)
595:         {
596:             $item = $line;
597:             $text->insert('end', $item);
598:         }
599:         $checkboxbutton_check_error = 0;
600:     }
601:     else
602:     {
603:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
604:         $checkboxbutton_check_error = 0;
605:     }
606:     #clear_log_check_error();
607: }
608: elsif(($checkboxbutton_update_error == 1)&&($value_button == 1))
609: {
610:     $log_update_error = '/home/fguedes/MIB3G/logs/log_update_error.txt';
611:     $count_update_error = `wc -l < $log_update_error`;
612:     chomp($count_update_error);
613:     $count_update_error = $count_update_error + 0;
614:     $item = open(LOG UPDATE ERROR,"<$log_update_error");
615:     if($count_update_error ne 0)
616:     {
617:         @log_update_error=<LOG UPDATE_ERROR>;
618:         $text->delete('1.0', 'end');
619:         foreach $line (@log_update_error)
620:         {
621:             $item = $line;
622:             $text->insert('end', $item);
623:         }
624:         $checkboxbutton_update_error = 0;
625:     }
626:     else
627:     {
628:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
629:         $checkboxbutton_update_error = 0;

```

```

630:     }
631:     #clear_log_update_error();
632: }
633: elseif((Scheckbutton_remove_error == 1)&&($value_button == 1))
634: {
635:     $log_remove_error = '/home/fguedes/MIB3G/logs/log_remove_error.txt';
636:     $count_remove_error = `wc -l < $log_remove_error`;
637:     chomp($count_remove_error);
638:     $count_remove_error = $count_remove_error + 0;
639:     $item = open(LOG_REMOVE_ERROR,"<$log_remove_error");
640:     if($count_remove_error ne 0)
641:     {
642:         @log_remove_error=<LOG_REMOVE_ERROR>;
643:         $text->delete('1.0', 'end');
644:         foreach $line (@log_remove_error)
645:         {
646:             $item = $line;
647:             $text->insert('end', $item);
648:         }
649:         $checkbutton_remove_error = 0;
650:     }
651:     else
652:     {
653:         $mw->Dialog(-title => 'ERROR',-text => 'LOG IS EMPTY', -
default button => 'OK',-bitmap => 'error',-buttons=>[ "OK"])->Show();
654:         $checkbutton_remove_error = 0;
655:     }
656:     #clear_log_remove_error();
657: }
658:
659: $value_button = 0;
660: }
661:
662: sub erase
663: {
664: $text->delete('1.0', 'end');
665: }
666:
667:
668: sub xterm
669: {
670:     system("/usr/openwin/bin/xterm");
671: }
672:
673: sub display
674: {
675:     my ($item) = @ ;
676:     $text->delete('1.0', 'end');
677:     $text->insert('end', $item);
678: }
679:
680: sub display_checkbutton
681: {
682:     my $item = "Checkbutton1: $cb1\n";
683:     $item .= "Checkbutton2: $cb2\n";
684:     $text->delete('1.0', 'end');
685:     $text->insert('end', $item);
686: }
687:
688: sub display_radiobutton {
689:     my $item = "$rb selected";
690:     $text->delete('1.0', 'end');
691:     $text->insert('end', $item);
692: }
693:
694: sub teste
695: {
696:     #my $image = $mw->Photo('-format' => 'jpeg', -file =>
'/home/fguedes/MIB3G/logo.jpg');

```

```

697: my $Dialog = $mw->Dialog(-text => "Escolha RNC", -bitmap => "question", -title
=> 'File Dialog', -default_button => 'RNC01',-buttons=>[ 'RNC01', "RNC02",
"RNC03" ])->pack();
698: my $result = $Dialog->Show;
699: if($result)
700: {
701:     print"$result\n";
702: }
703:
704: }
705:
706: sub FileChooser_Insert
707: {
708:
709: $FSref = $mw->FileSelect(-directory => '/home', -height => 25, -width =>25);
710: $file = $FSref->Show;
711:
712:
713: if($file)
714: {
715:
716:     open (example, $file) || die ("Could not open file $file ");
717:     $system = <example>;
718:     if($system =~/\//System RNCID: (\d+)/)
719:     {
720:         open(FILEDATA,$file) || die("Cannot open file $file");
721:         @datafile=<FILEDATA>;
722:         foreach $line (@datafile)
723:         {
724:             if($line =~/\//System RNCID: (\d+)/)
725:             {
726:                 $rncid = $1;
727:                 $opcao = insert;
728:                 Parser_File($file,$opcao,$rncid);
729:             }
730:         }
731:     }
732: }
733: else
734: {
735:     $item = "\nNO System RNCID in header of file $file\n";
736:     $text->insert('end', $item);
737: }
738: check_logs();
739: log_geral();
740: }
741: }
742:
743: sub FileChooser_Check
744: {
745:
746: $FSref = $mw->FileSelect(-directory => '/home', -height => 25, -width =>25);
747: $file = $FSref->Show;
748: if($file)
749: {
750:     open(FILEDATA,$file) || die("Cannot open file $file");
751:     @datafile=<FILEDATA>;
752:     $opcao = check;
753:     Parser_File($file,$opcao,$rncid);
754:
755:     #elsif($line!~/System RNCID: /)
756:     #{
757:         #$Dialog = $mw->DialogBox(-title => 'Choose RNC', -default_button => '51',
-headers=>[ "51", "52", "60"]);
758:         #$rncid = $Dialog->Show;
759:         #$opcao = insert;
760:         #Parser_File($file,$opcao,$rncid);
761:     }

```

```
762:
763:
764:
765:     check_logs();
766:     log_geral();
767: }
768:
769: }
770:
771: sub Logs
772: {
773:
774:     my $mw2 = MainWindow ->new(-screen => $ARGV[0] || $ENV{DISPLAY});
775:     $mw2->Button(-text=> 'Teste')->pack;
776:
777: }
778:
779:
780:
781:
782: sub CheckResult
783: {
784:     my($result)=@_;
785:
786:     if($result =~/YES/)
787:     {
788:         Parser File($file,$opcao,$text);
789:         check_logs();
790:         log_geral();
791:     }
792:
793:     else
794:     {
795:         exit;
796:     }
797:
798: }
799:
800: sub ReadLogs
801: {
802:     {
803:         #SFSref = $mw->FileSelect(-directory => '/home/frguedes/MIB3G/logs');
804:         #SFSref->Show;
805:
806:         $log_geral='/home/frguedes/MIB3G/logs/log_geral.txt';
807:         my $item = open(LOG_GERAL,"<$log_geral") || die("Cannot open file
808: $log_geral");
809:         @log_geral=<LOG_GERAL>;
810:         $text->delete('1.0', 'end');
811:         foreach $line (@log_geral)
812:         {
813:             $item = $line;
814:             $text->insert('end', $item);
815:         }
816:
817:
818:
819: }
820:
821:
822: sub check_logs
823: {
824:
825:
826: $logmain='/home/frguedes/MIB3G/logmain.txt';
827: $logmod='/home/frguedes/MIB3G/logmod.txt';
828: $logrmv='/home/frguedes/MIB3G/logrmv.txt';
```

```
829: $logfails='/home/fguedes/MIB3G/logfails.rmv';
830:
831: #Verificar se existem linhas nos logs
832:
833:
834: $countmain = `wc -l < $logmain`;
835: $countfails = `wc -l < $logfails`;
836: $countrmv = `wc -l < $logrmv`;
837: $countmod = `wc -l < $logmod`;
838:
839: chomp($countmain);
840: chomp($countfails);
841: chomp($countrmv);
842: chomp($countmod);
843:
844:
845: }
846:
847:
848: sub log_geral
849: {
850:
851: $log_geral='/home/fguedes/MIB3G/logs/log_geral.txt';
852: $log_check='/home/fguedes/MIB3G/logs/log_check.txt';
853: $log_insert='/home/fguedes/MIB3G/logs/log_insert.txt';
854: $log_update='/home/fguedes/MIB3G/logs/log_update.txt';
855: $log_remove = '/home/fguedes/MIB3G/logs/log_remove.txt';
856: $log_set = '/home/fguedes/MIB3G/logs/log_set.txt';
857: $log_databaseops = '/home/fguedes/MIB3G/logs/log_databaseops.txt';
858: $log_add = '/home/fguedes/MIB3G/logs/log_add.txt';
859: $log_rmv = '/home/fguedes/MIB3G/logs/log_rmv.txt';
860:
861: open(LOG_GERAL,">>$log_geral") || die("Cannot open file log_geral.txt");
862: open(LOG_CHECK,"<$log_check") || die("Cannot open file log_check.txt");
863: open(LOG_INSERT,"<$log_insert") || die("Cannot open file log_insert.txt");
864: open(LOG_UPDATE,"<$log_update") || die("Cannot open file log_update.txt");
865: open(LOG_REMOVE,"<$log_remove") || die("Cannot open file log_remove.txt");
866: open(LOG_SET,"<$log_set") || die("Cannot open file log_set.txt");
867: open(LOG_DATABASEOPS,">>$log_databaseops") || die("Cannot open file
log_databaseops.txt");
868: open(LOG_ADD,"<$log_add") || die("Cannot open file log_add.txt");
869: open(LOG_RMV,"<$log_rmv") || die("Cannot open file log_rmv.txt");
870:
871: @log_check=<LOG_CHECK>;
872: @log_insert=<LOG_INSERT>;
873: @log_remove=<LOG_REMOVE>;
874: @log_update=<LOG_UPDATE>;
875: @log_set=<LOG_SET>;
876: @log_databaseops=<LOG_DATABASEOPS>;
877: @log_add=<LOG_ADD>;
878: @log_rmv=<LOG_RMV>;
879:
880: $linecount_check = `wc -l < $log_check`;
881: $linecount_insert = `wc -l < $log_insert`;
882: $linecount_remove = `wc -l < $log_remove`;
883: $linecount_update = `wc -l < $log_update`;
884: $linecount_set = `wc -l < $log_set`;
885: $linecount_add = `wc -l < $log_add`;
886: $linecount_rmv = `wc -l < $log_rmv`;
887:
888: chomp($linecount_check);
889: chomp($linecount_insert);
890: chomp($linecount_remove);
891: chomp($linecount_update);
892: chomp($linecount_set);
893: chomp($linecount_add);
894: chomp($linecount_rmv);
895:
```

```
896: $linecount check = $linecount check + 0;
897: $linecount insert = $linecount insert + 0;
898: $linecount remove = $linecount remove + 0;
899: $linecount update = $linecount update + 0;
900: $linecount set = $linecount set + 0;
901: $linecount add = $linecount add + 0;
902: $linecount_rmv = $linecount_rmv + 0;
903:
904:
905: ($sec,$min,$hours,$day,$month,$year) = (localtime)[0,1,2,3,4,5];
906: $month=$month+1;
907: $year=$year+1900;
908:
909:
910: $text->delete('1.0', 'end');
911:
912:
913: print LOG_GERAL "\nSESSION->$day-$month-$year\t$hours:$min\tFILE:$file\n";
914:
915: if($linecount_check > 0)
916: {
917:     my($count check) = 0;
918:     my($count check error) = 0;
919:     my($count check warning) = 0;
920:     foreach $line (@log_check)
921:     {
922:         print LOG_GERAL "$hours:$min:$sec\tCHECK\t\t\t$line";
923:         if($line =~/OK/)
924:         {
925:             $count_check++;
926:         }
927:         elsif($line =~/ERROR/)
928:         {
929:             $count_check_error++;
930:         }
931:         }
932:         elsif($line =~/WARNING/)
933:         {
934:             $count_check_warning++;
935:         }
936:     }
937: }
938: if($count_check_error eq 0)
939: {
940:     $item = "\nNO ERRORS IN CHECK: ALL PARAMETERS CHECKED\n";
941:     $text->insert('end', $item);
942: }
943: if($count_check_warning ne 0)
944: {
945:     $item = "\nCHECK RESULTS (WARNINGS)->$count_check_warning in FILE
$file\nPlease check log check.txt\n";
946:     $text->insert('end', $item);
947: }
948: if($count_check_error ne 0)
949: {
950:     $mw->Dialog(-title => 'ERROR', -text => "$count_check_error error(s) in
log check.txt\nPlease check line:\n$line")->Show();
951:     $item = "\nCHECK RESULTS (ERRORS)->$count_check_error in FILE
$file\nPlease check log check.txt\n";
952:     $text->insert('end', $item);
953: }
954: }
955:
956:
957: if($linecount_insert > 0)
958: {
959:     my($count insert) = 0;
960:     my($count_insert_error) = 0;
```

```
961: my($count_insert_warning) = 0;
962: foreach $line (@log_insert)
963: {
964:     print LOG GERAL "$hours:$min:$sec\tINSERT\t\t\t$line";
965:     if($line =~/OK/)
966:     {
967:         $count_insert++;
968:     }
969:     elsif($line =~/ERROR/)
970:     {
971:         $count_insert_error++;
972:     }
973:     }
974:     elsif($line =~/WARNING/)
975:     {
976:         $count_insert_warning++;
977:     }
978: }
979:
980: if($count_insert_error eq 0)
981: {
982:     $item = "\nNO ERRORS IN INSERT: ALL PARAMETERS INSERTED\n";
983:     $text->insert('end', $item);
984: }
985: if($count_insert_error ne 0)
986: {
987:     $item = "\nINSERT RESULTS (ERRORS)->$count_insert_error in FILE
$files\nPlease check log insert.txt\n";
988:     $text->insert('end', $item);
989: }
990: if($count_insert_warning ne 0)
991: {
992:     $item = "\nINSERT RESULTS (WARNINGS)->$count_insert_warning in FILE
$files\nPlease check log insert.txt\n";
993:     $text->insert('end', $item);
994: }
995: }
996:
997: if($linecount_remove > 0)
998: {
999:     my($count_remove) = 0;
1000:     my($count_remove_error) = 0;
1001:
1002:     foreach $line (@log_remove)
1003:     {
1004:         print LOG GERAL "$hours:$min:$sec\tREMOVE\t\t\t$line";
1005:         if($line =~/OK/)
1006:         {
1007:             $count_remove++;
1008:         }
1009:         elsif($line =~/ERROR/)
1010:         {
1011:             $count_remove_error++;
1012:         }
1013:     }
1014: }
1015: if($count_remove_error eq 0)
1016: {
1017:     $item = "\nNO ERRORS IN REMOVE: ALL PARAMETERS REMOVED\n";
1018:     $text->insert('end', $item);
1019: }
1020: if($count_remove_error ne 0)
1021: {
1022:     $item = "\nREMOVE RESULTS (ERRORS)->$count_remove_error in FILE
$files\nPlease check log remove.txt\n";
1023:     $text->insert('end', $item);
1024: }
1025: }
```

```
1026:
1027:
1028: if($linecount_update > 0)
1029: {
1030:     my($count_update) = 0;
1031:     my($count_update_error) = 0;
1032:     foreach $line (@log_update)
1033:     {
1034:         print LOG_GERAL "$hours:$min:$sec\tUPDATE\t\t\t$line";
1035:         if($line =~/OK/)
1036:         {
1037:             $count_update++;
1038:         }
1039:         elsif($line =~/ERROR/)
1040:         {
1041:             $count_update_error++;
1042:         }
1043:     }
1044:
1045:     if($count_update_error eq 0)
1046:     {
1047:         $item = "\nNO ERRORS IN UPDATE: ALL PARAMETERS UPDATED\n";
1048:         $text->insert('end', $item);
1049:     }
1050:     if($count_update_error ne 0)
1051:     {
1052:         $item = "\nUPDATE RESULTS (ERRORS)->$count_update_error in FILE
1053: $file\nPlease check log update.txt\n";
1054:         $text->insert('end', $item);
1055:     }
1056:
1057: if($linecount_set > 0)
1058: {
1059:     my($count_set) = 0;
1060:     my($count_set_error) = 0;
1061:     foreach $line (@log_set)
1062:     {
1063:         print LOG_GERAL "$hours:$min:$sec\tSET\t\t\t$line";
1064:         if($line =~/OK/)
1065:         {
1066:             $count_set++;
1067:         }
1068:         elsif($line =~/ERROR/)
1069:         {
1070:             $count_set_error++;
1071:         }
1072:     }
1073:
1074:     if($count_set_error eq 0)
1075:     {
1076:         $item = "\nNO ERRORS IN SET COMMAND: ALL PARAMETERS UPDATED\n";
1077:         $text->insert('end', $item);
1078:     }
1079:     if($count_set_error ne 0)
1080:     {
1081:         $item = "\nSET RESULTS (ERRORS)->$count_set_error in FILE $file\nPlease
1082: check log set.txt\n";
1083:         $text->insert('end', $item);
1084:     }
1085:
1086: if($linecount_add > 0)
1087: {
1088:     my($count_add_error) = 0;
1089:     foreach $line (@log_add)
1090:     {
1091:         #print LOG_GERAL "$hours:$min:$sec\tADD\t\t\t$line";
```

```

1092:         if($line =~/ERROR/)
1093:         {
1094:             $count_add_error++;
1095:         }
1096:     }
1097: }
1098:
1099: if($count_add_error ne 0)
1100: {
1101:     $item = "\nADD RESULTS (ERRORS)->$count_add_error in FILE $file\nPlease
check log add.txt\n";
1102:     $text->insert('end', $item);
1103: }
1104: }
1105:
1106: if($linecount_rmv > 0)
1107: {
1108:     my($count_rmv_error) = 0;
1109:     foreach $line (@log_rmv)
1110:     {
1111:         #print LOG_GERAL "$hours:$min:$sec\tSET\t\t$line";
1112:         if($line =~/ERROR/)
1113:         {
1114:             $count_rmv_error++;
1115:         }
1116:     }
1117: }
1118:
1119: if($count_rmv_error ne 0)
1120: {
1121:     $item = "\nRMV RESULTS (ERRORS)->$count_rmv_error in FILE $file\nPlease
check log rmv.txt\n";
1122:     $text->insert('end', $item);
1123: }
1124: }
1125:
1126:
1127: foreach $line (@log_databaseops)
1128: {
1129:     print LOG_GERAL "$hours:$min:$sec\tCOMMIT\t\t$line";
1130: }
1131:
1132: $count_errors = $count_check_error + $count_update_error + 0;
1133:
1134: if($count_errors > 0)
1135: {
1136:     if(($count_check_error ne 0)&&($count_update_error eq 0))
1137:     {
1138:         # $item = "$count_check_error error(s) in line: $line: Please read
log check.txt\n";
1139:         # $text->delete('1.0', 'end');
1140:         # $text->insert('end', $item);
1141:         print color("red"),"$count_check_error error(s): Please read
log check.txt\n", color("reset");
1142:     }
1143:
1144:     elsif(($count_update_error ne 0)&&($count_check_error eq 0))
1145:     {
1146:         #my $item = "$count update error(s): Please check log_update.txt\n";
1147:         # $text->delete('1.0', 'end');
1148:         # $text->insert('end', $item);
1149:         print color("red"),"$count update_error (s): Please check
log update_error.txt\n", color("reset");
1150:     }
1151:     elsif(($count_update_error ne 0)&&($count_check_error ne 0))
1152:     {
1153:         #my $item = "$count_errors(s): Please check log_update_error.txt and
log_check_error.txt\n";

```

```

1154:         #Stext->delete('1.0', 'end');
1155:         #Stext->insert('end', $item);
1156:         print color("red"), "Scount update error(s): Please check
log update_error.txt and log_check_error.txt\n", color("reset");
1157:     }
1158:     elsif($scount_insert_error ne 0)
1159:     {
1160:         #my $item = "Scount_insert_error errors(s): Please check
log insert.txt\n";
1161:         #Stext->delete('1.0', 'end');
1162:         #Stext->insert('end', $item);
1163:         print color("red"), "Scount_insert_error errors(s): Please check
log insert.txt\n", color("reset");
1164:     }
1165:     elsif($scount_remove_error ne 0)
1166:     {
1167:         #my $item = "Scount_remove_error errors(s): Please check
log remove.txt\n";
1168:         #Stext->delete('1.0', 'end');
1169:         #Stext->insert('end', $item);
1170:         print color("red"), "Scount_remove_error errors(s): Please check
log remove.txt\n", color("reset");
1171:     }
1172:     elsif($scount_set_error ne 0)
1173:     {
1174:         #my $item = "Scount set error errors(s): Please check log_set.txt\n";
1175:         #Stext->delete('1.0', 'end');
1176:         #Stext->insert('end', $item);
1177:         print color("red"), "Scount_remove_error errors(s): Please check
log set.txt\n", color("reset");
1178:     }
1179: }
1180: else
1181: {
1182:     #my $item = "Scount errors errors: All successful\n";
1183:     #Stext->insert('end', $item);
1184:     #print color("green"), "Scount_errors errors: All sucessful\n", color("reset");
1185: }
1186: }
1187:
1188:
1189: }
1190:
1191: sub commit
1192: {
1193:     my $dbh = DBI->connect("DBI:mysql:database=ran_virtual;host=localhost",
1194:                           "frguedes", "riutort",
1195:                           {'RaiseError' => 1});
1196:     my $dbh2 = DBI->connect("DBI:mysql:database=ran;host=localhost",
1197:                            "frguedes", "riutort",
1198:                            {'RaiseError' => 1});
1199:
1200:
1201:     $log_databaseops = '/home/frguedes/MIB3G/logs/log_databaseops.txt';
1202:
1203:
1204:     $result=$mw->Dialog(-title => 'COMMIT',-text => 'ARE YOU SURE YOU WANT TO
COMMIT ALL THE CHANGES?', -default_button => 'yes',-bitmap => 'warning',-
buttons=>[ "YES","NO"])->Show();
1205:     if($result =~/YES/)
1206:     {
1207:         $value commit = 1;
1208:         system("/usr/local/mysql/bin/mysqldump -uroot -pmib3g --no-create-info
ran_virtual > 'ran_virtual temp.sql");
1209:         system("/usr/local/mysql/bin/mysqldump -uroot -pmib3g --no-create-info ran >
ran temp.sql");
1210:         $query=("drop database ran_virtual");
1211:         $query2=("create database ran_virtual");

```

```

1212: $sth = $dbh->prepare($query);
1213: $sth2 = $dbh->prepare($query2);
1214: $sth->execute();
1215: $sth2->execute();
1216: $query3=("drop database ran");
1217: $query4=("create database ran");
1218: $sth3 = $dbh2->prepare($query3);
1219: $sth4 = $dbh2->prepare($query4);
1220: $sth3->execute();
1221: $sth4->execute();
1222: system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual <
tabelas.sql");
1223: system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran < tabelas.sql");
1224: system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran <
ran_virtual temp.sql");
1225: system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual <
ran temp.sql");
1226: print LOG DATABASEOPS "COMMITTED ALL THE CHANGES\n";
1227: my $item = "COMMIT SUCCESSFUL\n";
1228: $text->delete('1.0', 'end');
1229: $text->insert('end', $item);
1230: }
1231: }
1232: }
1233:
1234: sub export_xml
1235: {
1236:     my $dbh = DBI->connect("DBI:mysql:database=ran_virtual;host=localhost",
1237:                             "frguedes", "riutort",
1238:                             {'RaiseError' => 1});
1239:     $query=("drop database ran_virtual");
1240:     $query2=("create database ran_virtual");
1241:     $sth = $dbh->prepare($query);
1242:     $sth2 = $dbh->prepare($query2);
1243:     $sth->execute();
1244:     $sth2->execute();
1245:     system("/usr/local/mysql/bin/mysqldump --xml -uroot -pmib3g ran > ran.xml");
1246:     $mw->Dialog(-title => 'BACKUP', -text => "XML created in
/home/frguedes/MIB3G/ran.xml")->Show();
1247: }
1248: }
1249:
1250: sub backup
1251: {
1252:     system("/usr/local/mysql/bin/mysqldump -uroot -pmib3g ran > ran_backup.sql");
1253:     $mw->Dialog(-title => 'BACKUP', -text => "Backup created in
/home/frguedes/MIB3G/ran_backup.sql")->Show();
1254: }
1255:
1256: sub help_about
1257: {
1258:     $mw->Dialog(-title => 'ABOUT MIB3G', -text => "MIB3G 1.1a\nDeveloped by Guedes,
Filipe\nOptimus Telecomunicações SA 2005\nAll rights Reserved\n", -
default button => 'OK', -buttons=>[ "OK"])->Show();
1259:     $mw->Label(-text => "MIB3G 1.1a\nDeveloped by Guedes, Filipe@Optimus
Telecomunicações SA 2005\nAll rights Reserved\n")->pack( );
1260: }
1261: }
1262: }
1263: }
1264: }
1265: }
1266: }
1267: }
1268: }
1269: }
1270: }
1271: }

```

272:

ANEXO B: Código da cron

ANEXO B: Código da cron

```

1: #!/usr/local/bin/perl
2:
3:
4: use parser_file qw(&Parser_File);
5: use DBI;
6: use Tk;
7: use Tk::JPEG;
8: use Tk::FileSelect;
9: use Tk::Dialog;
10: use Term::ReadKey;
11: use Term::ANSIColor;
12:
13:
14:
15: system "clear";
16:
17:
18: my($Nargs);
19: $Nargs=@ARGV;
20: if($Nargs == 0)
21: {
22:
23:
24:     print "Please insert the cfgmml file\n";
25:     chomp($file = <STDIN> );
26:     open (example, $file) || die ("Could not open file $file!");
27:     $system_rncid = <example>;
28: }
29: elsif(1 == $Nargs)
30: {
31:     $file=$ARGV[0];
32: }
33:
34:
35:     print"$file\n";
36:     my $dbh = DBI->connect("DBI:mysql:database=ran_virtual;host=localhost",
37:                           "frguedes", "riutort",
38:                           {'RaiseError' => 1});
39:     print"DROPPING...\n";
40:     $query=("drop database ran_virtual");
41:     print"CREATING...\n";
42:     $query2=("create database ran_virtual");
43:     $sth = $dbh->prepare($query);
44:     $sth2 = $dbh->prepare($query2);
45:     $sth->execute();
46:     $sth2->execute();
47:     print"RECONSTRUCTING...\n";
48:     system("/usr/local/mysql/bin/mysql -uroot -pmib3g ran_virtual <
tabelas.sql");
49:     open(FILEDATA,$file) || die("Cannot open file $file");
50:     @datafile=<FILEDATA>;
51:     foreach $line (@datafile)
52:     {
53:         if($line =~ /\sSystem RNCID: (\d+)/)
54:         {
55:             $rncid = $1;
56:             $opcao = insert;
57:             Parser_File($file,$opcao,$rncid);
58:         }
59:     }
60:
61:     my $dbh2 = DBI->connect("DBI:mysql:database=ran_cfgmml;host=localhost",
62:                           "frguedes", "riutort",
63:                           {'RaiseError' => 1});
64:     $query3=("drop database ran_cfgmml");
65:     $query4=("create database ran_cfgmml");
66:     $sth3 = $dbh2->prepare($query3);
67:     $sth4 = $dbh2->prepare($query4);

```

```
8:      $sth3->execute();
9:      $sth4->execute();
10:     system("/usr/local/mysql/bin/mysql -uuser -p'libby ran cignmi < table1.sql");
11:     system("/usr/local/mysql/bin/mysql -uuser -p'libby --no-create-info
    ran virtual > '/home/fguedes/MIB3G/temp/ran_mysql.sql");
12:     system("/usr/local/mysql/bin/mysql -uuser -p'libby ran_mysql <
    '/home/fguedes/MIB3G/temp/ran_mysql.sql");
13:     print"success!\n";
14:
15:
16:
```

ANEXO C: Exemplo de algumas tabelas

```
1: CREATE TABLE CELLSETUP (
2: CELLID INT(10),
3: CELLNAME VARCHAR(40),
4: MAXTXPOWER VARCHAR(40),
5: UARFCNUPLINK VARCHAR(40),
6: UARFCNDOWNLINK VARCHAR(40),
7: TCELL VARCHAR(40),
8: NINSYNCIND VARCHAR(40),
9: NOUTSYNCIND VARCHAR(40),
10: TRFAILURE VARCHAR(40),
11: DLTPCPATTERN01COUNT VARCHAR(40),
12: POWERRAISELIMIT VARCHAR(40),
13: DLPOWERAVERAGEWINDOWSIZE VARCHAR(40),
14: NODESYNSWITCH VARCHAR(40),
15: PSCRAMBCODE VARCHAR(40),
16: SUPSSDT VARCHAR(40),
17: TXDIVERSITYMODE VARCHAR(40),
18: TXDIVERSITYIND VARCHAR(40),
19: SSN VARCHAR(40),
20: CFGRACIND VARCHAR(20),
21: NODEBNAME VARCHAR(40),
22: LOCELL VARCHAR(40),
23: SERVICEIND VARCHAR(40),
24: LAC VARCHAR(40),
25: RAC VARCHAR(40),
26: SAC VARCHAR(40),
27: RNCID INT(10),
28: PRIMARY KEY (CELLID,RNCID) TYPE=INNODB;
29:
30: CREATE TABLE 2G_NEIS_3G (
31: source_bsc VARCHAR(20),
32: source_site VARCHAR(32),
33: source_cell VARCHAR(20),
34: source_lac SMALLINT UNSIGNED,
35: source_ci SMALLINT UNSIGNED,
36: source_cgi VARCHAR(20),
37: target_cell VARCHAR(20),
38: target_lac SMALLINT UNSIGNED,
39: target_ci SMALLINT UNSIGNED,
40: target_rncid SMALLINT UNSIGNED,
41: target_cgi VARCHAR(20),
42: fdd_arfcn SMALLINT UNSIGNED,
43: scr_code SMALLINT UNSIGNED,
44: diversity_enabled SMALLINT UNSIGNED,
45: PRIMARY KEY (source_bsc,source_ci,target_ci) TYPE=INNODB;
46:
47: CREATE INDEX source_bsc_index USING BTREE ON 2G_NEIS_3G (source_bsc);
48: CREATE INDEX source_ci_index USING BTREE ON 2G_NEIS_3G (source_ci);
49: CREATE INDEX target_ci_index USING BTREE ON 2G_NEIS_3G (target_ci);
50:
51: CREATE TABLE 2G_CELLS (
52: bsc varchar(20),
53: btsid SMALLINT ,
54: btsname VARCHAR(32),
55: name VARCHAR(20),
56: lac SMALLINT UNSIGNED,
57: ci SMALLINT UNSIGNED,
58: cgi VARCHAR(20),
59: bcch SMALLINT UNSIGNED,
60: bsic SMALLINT UNSIGNED,
61: qsearch_i SMALLINT,
62: inter_rat_enabled SMALLINT,
63: qsearch_c_initial SMALLINT,
64: fdd_qoffset SMALLINT,
65: fdd_qmin SMALLINT,
66: nnei SMALLINT,
67: PRIMARY KEY (bsc,ci) TYPE=INNODB;
68:
```

```
69: CREATE INDEX bsc_index USING BTREE ON 2G CELLS (bsc);
70: CREATE INDEX ci_index USING BTREE ON 2G CELLS (ci);
71:
72: CREATE TABLE 2G_NEIS (
73: source bsc VARCHAR(20),
74: source site VARCHAR(32),
75: source cell VARCHAR(20),
76: source lac SMALLINT UNSIGNED,
77: source ci SMALLINT UNSIGNED,
78: source cgi VARCHAR(20),
79: target cell VARCHAR(20),
80: target lac SMALLINT UNSIGNED,
81: target ci SMALLINT UNSIGNED,
82: target cgi VARCHAR(20),
83: bsic SMALLINT UNSIGNED,
84: bcchfrequencynum SMALLINT UNSIGNED,
85: ho_margin_cell SMALLINT,
86: ho_margin_rxqual SMALLINT,
87: ho_margin_rxlev SMALLINT,
88: ms_txpwr_max_cell SMALLINT,
89: rxlev_min_cell SMALLINT,
90: synchronized SMALLINT UNSIGNED,
91: adj_chn_intf_tst SMALLINT,
92: adj_chn_rxlev_dif SMALLINT,
93: neighbouringrange SMALLINT,
94: adap_trg_pbgt_nbr SMALLINT,
95: pbgtsrcellhreqave SMALLINT,
96: pbgt_alg_type SMALLINT,
97: pbgt_alg_data0 SMALLINT,
98: pbgt_alg_data1 SMALLINT,
99: pbgt_alg_data2 SMALLINT,
100: ho_margin_type5 SMALLINT,
101: dr_allowed SMALLINT,
102: congest_ho_margin SMALLINT,
103: interfering_nei SMALLINT,
104: inner_zone_thresh SMALLINT,
105: inner_zone_margin SMALLINT,
106: BA_TYPE smallint UNSIGNED,
107: frequency_type SMALLINT UNSIGNED,
108: PRIMARY KEY (source_bsc,source_ci,target_ci)) TYPE=INNODB;
109:
110: CREATE INDEX source_bsc_index USING BTREE ON 2G_NEIS (source_bsc);
111: CREATE INDEX source_ci_index USING BTREE ON 2G_NEIS (source_ci);
112: CREATE INDEX target_ci_index USING BTREE ON 2G_NEIS (target_ci);
113:
114: CREATE TABLE NODEB(
115: NODEBID INT(10) auto increment,
116: NODEBNAME VARCHAR(35),
117: SRN INT(10),
118: SSN INT(10),
119: RNCID INT(10),
120: PRIMARY KEY (NODEBID,NODEBNAME)) TYPE=INNODB;
121:
122: CREATE TABLE NRNC(
123: RNCID INT(10),
124: MCC VARCHAR(10),
125: MNC VARCHAR(10),
126: SHOHOTRIG VARCHAR(20),
127: SERVICEIND VARCHAR(30),
128: PRIMARY KEY(RNCID)) TYPE=INNODB;
129:
130: CREATE TABLE INTERFREQCELL(
131: CELLID INT(10),
132: NCELLID INT(10),
133: READSFNIND VARCHAR(20),
134: CELLINDIVIDALOFFSET INT(10),
135: QOFFSET1SN INT(10),
136: QOFFSET2SN INT(10),
```

```
37: RNCID INT(10),
38: FOREIGN KEY (CELLID) REFERENCES CELLSETUP (CELLID) on delete cascade,
39: PRIMARY KEY (CELLID,NCELLID) TYPE=INNODB;
40:
41: CREATE INDEX CELLID_INDEX USING BTREE ON INTERFREQCELL (CELLID);
42:
43: CREATE TABLE INTRAFREQCELL(
44: CELLID INT(10),
45: NCELLID INT(10),
46: READSFNIND VARCHAR(20),
47: CELLINDIVIDALOFFSET INT(10),
48: CELLSFORBIDDEN1A VARCHAR(20),
49: CELLSFORBIDDEN1B VARCHAR(20),
50: QOFFSET1SN INT(10),
51: QOFFSET2SN INT(10),
52: RNCID INT(10),
53: FOREIGN KEY (CELLID) REFERENCES CELLSETUP (CELLID) on delete cascade,
54: PRIMARY KEY (CELLID,NCELLID) TYPE=INNODB;
55:
56: CREATE INDEX CELLID_INDEX USING BTREE ON INTRAFREQCELL (CELLID);
```

ANEXO D: Apresentação final à Optimus



OPTIMUS

MIB3G

Apresentação Técnica

22 de Agosto de 2005



MIB3G



Funcionalidades

Revisão Tecnológica

Solução Proposta

Questões

AnalFcodeProjectFcodeNameDataAuthor

Page 3

MIB3G



Funcionalidades

Revisão Tecnológica

Solução Proposta

Questões

AnalFcodeProjectFcodeNameDataAuthor

Page 2

MIB3G



Funcionalidades

Revisão Tecnológica

Solução Proposta

Questões

AnalFcodeProjectFcodeNameDataAuthor

Page 4

MIB3G

OPTIMUS

Funcionalidades

Revisão Tecnológica

Solução Proposta

Questões

Area/Code/Project/Code/Name/Date/Author

Page: 6

Revisão Tecnológica

OPTIMUS

Linguagens utilizadas

Perl – Scripting/Parsing

MySQL – Base Dados

Perl/TK - Interface





Area/Code/Project/Code/Name/Date/Author

Page: 7

Funcionalidades

OPTIMUS

Visualização em tempo real

Interface fácil e intuitiva

Single-user

Facilidade de upgrade

Area/Code/Project/Code/Name/Date/Author

Page: 8

Revisão Tecnológica

OPTIMUS

Perl – Pontos Fortes

Muito pratica para extrair informação

Gerar informação a partir do conteúdo dos arquivos

Capacidade de evoluir

Open Source

MySQL – Pontos Fortes

Desempenho rápido e consistente

Fiabilidade elevada

Manuseamento extremamente fácil

Open Source

Area/Code/Project/Code/Name/Date/Author

Page: 8

Revisão Tecnológica



• Mecanismo de Parsing

Exemplo de linha de entrada

• ADD CELLPLWR, CELLO = 1471 CNDOMAINID = CS_DOMAIN, MAXIBRATE = 3400, RLMAXDLPWR = -15, RLMINDLPWR = -150, DLSF = 0126

• 1ª Fase

- Parsing do tipo de comando (ADD, RMV, SET ou MOD)
- ADD

• 2ª Fase

- Parsing do parâmetro principal
- CELLPLWR

• 3ª Fase

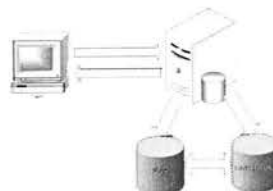
- Parsing dos parâmetros obrigatórios e opcionais
- CNDOMAINID, MAXIBRATE, RLMAXDLPWR, RLMINDLPWR, DLSF

Revisão Tecnológica



• Base de dados – MySQL

• 2 Base de dados



Revisão Tecnológica



Mecanismo de Check

• 1ª Fase

- Check dos parâmetros obrigatórios
- CNDOMAINID, MAXIBRATE, RLMAXDLPWR, RLMINDLPWR, DLSF

• 2ª Fase

- Check dos ranges dos parâmetros

• Ficheiro de ranges dos parâmetros

• Exemplo:

• CELLPLWR, CELLO = 0-26843455, CNDOMAINID = CS_DOMAIN, PS_DOMAIN, MAXIBRATE = 0-18000000, RLMAXDLPWR = 350-150, RLMINDLPWR = 350-150

Solução Proposta





???



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000081405