



Ministério da
Ciência e Tecnologia



DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA PARA O
SISTEMA DE CONTROLE DA ANTENA DA ESTAÇÃO
MULTIMISSÃO DE NATAL – EMMN

Moisés Cirilo de Brito Souto

Orientador: Manoel Jozeane Mafra de Carvalho

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)

CRN / INPE
Natal, Rio Grande do Norte
2009

Publicado por:

Instituto Nacional de Pesquisas Espaciais (INPE)

Gabinete do Diretor – (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 – CEP 12.245-970

São José dos Campos – SP – Brasil

Tel.: (012) 3945-6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br



Ministério da
Ciência e Tecnologia



DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA PARA O SISTEMA DE CONTROLE DA ANTENA DA ESTAÇÃO MULTIMISSÃO DE NATAL – EMMN

Moisés Cirilo de Brito Souto

Orientador: Manoel Jozeane Mafra de Carvalho

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)

CRN / INPE
Natal, Rio Grande do Norte
2009

“[...] senti que estava seguindo os passos de centenas de cientistas e outros acadêmicos. Pessoas que construíram seu trabalho apoiando-se em outros”.

Apoiando-se nos ombros de gigantes, nas palavras de Sir Isaac Newton.

Linus Benedict Torvalds

Dedico este trabalho a minha família, Paula (Mãe), Francisco Cirilo (pai) e Marcos (irmão), sem a qual não poderia sequer imaginar um dia me tornar engenheiro. Minha avó paterna, Teresinha Souto, In memoriam, que foi uma mãe para mim. Meus avós maternos Nilson e Maria Alice. Minha namorada Maria Fernanda, pela paciência e carinho.

AGRADECIMENTOS

Prof. Eng. Msc. Manoel, do INPE, por ter me dado a primeira oportunidade, permitir realizar este trabalho, sob sua supervisão, no INPE. Prof. Eng. Dr. Oscar Gabriel Filho, que sempre foi sábio em seus conselhos e muito incentivou minha participação na vida acadêmica. Ao Eng. Dr. Alexandre Guirland Nowosad, pelas revisões textuais, sugestões e críticas. Todos os colegas do CRN/INPE, que contribuíram de alguma forma na realização deste trabalho.

RESUMO

Este trabalho descreve uma interface gráfica supervisória (IGS), em plataforma GNU/Linux, C/C++ e biblioteca gráfica Qt3, para rastreamento e gestão do sistema de controle da antena da Estação Multimissão de Natal – EMMN. Foi construída usando software livre, distribuída em rede sob paradigma cliente/servidor, usando o protocolo TCP/IP. A interface gráfica supervisória (IGS), comunica-se com o módulo de controle em hardware, tornando o processo de operação de rastreamento de satélites simples e robusto.

ABSTRACT

This paper describes graphical supervisory interface (GSI), in GNU/Linux platform, C/C++ and UI framework Qt3, for tracking and management of antenna control system in Estação Multimissão de Natal – EMMN (Natal Multimission Station). It has been developed with free software, distributed in Local Network over Server/client paradigm, using TCP/IP protocol. The graphical supervisory interface (GSI) communicates with hardware control module making the operation of satellite tracking, simple and robust.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS.....	25
LISTA DE TABELAS	29
LISTA DE SIGLAS E ABREVIATURAS	25
1 INTRODUÇÃO.....	35
1.1 A empresa.....	35
1.2 Atividades desenvolvidas.....	36
2 SUBSISTEMA DE CONTROLE E RASTREIO (SCR).....	39
2.1 Módulo Posicionador	39
2.2 Módulo de Potência	40
2.3 Módulo de Comando.....	41
2.4 Módulo de Controle.....	41
3 SOFTWARE DE RASTREIO DE SATÉLITES (SRS).....	43
3.1 A Interface Gráfica Supervisória (IGS) do Software de Rastreo de Satélites (SRS).....	44
3.2 Programa Servidor TCP/IP	46
3.3 Interfaces ao Usuário.....	50
3.3.1 Menus	52
3.3.2 Telas	57
3.3.3 Ações	60
3.4 Interfaces de software.....	61
3.5 Interfaces de comunicação	62
3.6 SSC.....	62
4 CONCLUSÕES E TRABALHOS FUTUROS	63
REFERÊNCIAS BIBLIOGRÁFICAS	64

APÊNDICE A – Protocolo de Comunicação da EMMN.....	66
--	----

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1: Instalações do INPE no Brasil, em destaque CRN Natal-RN	35
Figura 2: Estrutura de Hardware do Subsistema de Controle e Rastreo (SCR) da EMMN.....	39
Figura 3: Antena Parabólica da EMMN vista de ângulos diferentes	40
Figura 4: Vista do módulo de potência em gaveta no rack da estação e os inversores que controlam os motores	40
41	
Figura 5: Vista do módulo de comando no rack da estação e a parte traseira com as conexões com outros módulos do SCR.....	41
Figura 6: Camadas de abstração	43
Figura 7: Estrutura do servidor TCP/IP (STP).....	44
Figura 8: Interface em texto com biblioteca ncurses do SSC.....	45
Figura 9: Interface em Qt3.....	46
Figura 10: Exemplo de servidor TCP/IP aguardando conexões	48
Figura 11: Simulação de conexão ao serviço.....	48
Figura 12: Servidor reconhece que recebeu uma conexão e lista o IP.....	48
Figura 13: Enviado a string Teste, e o servidor responde com mensagem em formato de teste, simbolizando apenas que recebeu a string corretamente.....	49
Figura 14: Servidor exibe a mensagem na tela e diz o número de bytes recebidos	49
Figura 15: Simulação de envio de comando próximo ao formato que será adotado.....	49
Figura 16: Resposta recebida no cliente a partir de dados enviados ao servidor TCP/IP que os processou e retornou essa mensagem.....	50

Figura 17: Menu da tela principal do SRS.....	51
Figura 18: Aviso inicial sobre a configuração de placa conversora ADDA no SSC	51
Figura 19: Aviso alertando de que a placa conversora ADDA está pronta	51
Figura 20: Tela inicial para autenticar no sistema, o acesso aos menus é liberado após ter sido autenticado.....	52
Figura 21: Menu Serviços, detalhes vide Tabela 1	52
Figura 22: Submenu Passagens, referente as efemêrides. Opção de visualizar as já cadastradas e adicionar novas	53
Figura 23: Menu controle, onde as principais ações sobre a estação são feitas	53
Figura 24: Menu Administrador com opções avançadas de configuração, sendo permitido acesso a tais opções apenas a usuário com privilégios administrativos.....	54
Figura 25: Menu logs, onde pode-se encontrar os relatórios dos principais eventos registrados no SRS.....	54
Figura 26: Submneu Usuários, onde pode-se encontrar relatórios dos principais eventos relacionados a usuários do SRS.....	55
Figura 27: Submenu Erros, onde pode-se encontrar relatório dos principais eventos relacionados a erros do SRS	55
Figura 28: Submenu Sistema, onde pode-se encontrar relatórios dos principais eventos relacionados aos sistema SRS	56
Figura 29: Menu Ajuda, pode-se conseguir auxílio sobre utilização, configurações e dicas sobre o SRS	56
Figura 30: Informação sobre os desenvolvedores do SRS e SCR da EMMN ..	57
Figura 31: Tela Adicionar usuário, acesso a esta tela unicamente para usuários com privilégios administrativos.....	57

Figura 32: Tela de ajuste dos parâmetros do controlador PI e do controlador PID, acesso unicamente com privilégios administrativos	58
Figura 33: Tela de acompanhamento das efemérides cadastradas, sendo possível filtrar entre ativas, inativas e canceladas	58
Figura 34: Tela para acrescentar arquivos efemérides para rastreamento de satélites	59
Figura 35: Tela para Movimentação manual da Antena.....	59
Figura 36: Tela para teste dos leds da gaveta de pilotagem e acerto do relógio do sistema	60
Figura 37: Ligar ou desligar o sistema de potência da EMMN.....	60
Figura 38: Sincronizar o relógio do sistema com o relógio UTC	61
Figura 39: Alerta do teste dos leds.....	61

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1. Protocolo de comunicação da EMMN.....	47
Tabela 2: Menus da tela principal	50

LISTA DE SIGLAS E ABREVIATURAS

AD/DA	Analógico Digital/Digital Analógico
CBERS	Satélite Sino-Brasileiro de Recursos Terrestres (<i>China-Brazil Earth Resources Satellite</i>)
COMEDI	Control and Measurement Device Interface
CRN	Centro Regional do Nordeste
EMMN	Estação Multimissão de Natal
ETR	Estação Terrena de Recepção de Sinais de Satélite
IGS	Interface Gráfica do SCR
INPE	Instituto Nacional de Pesquisas Espaciais
MTRs	Minitransmissores Remotos
PCD	Plataforma de Coleta de Dados
PCI	Interconector de Componentes Periféricos (<i>Peripheral Component Interconnect</i>)
SBCD	Sistema Brasileiro de Coleta de Dados
SCD	Satélite de Coleta de Dados
SCR	Subsistema de Controle e Rastreamento
SDR	Software Defined Radio
SRS	Sistema de Rastreamento de Satélites
SSC	Software de Supervisão e Controle
SID	Serviço de Informação e Documentação
SPG	Serviço de Pós-Graduação
TDI	Teses e Dissertações Internas

1 INTRODUÇÃO

Este trabalho visa o desenvolvimento de software para a interface gráfica do Subsistema de Controle e Rastreo (SCR) (da antena) da Estação Multi-Missão de Natal (EMM-Natal), localizada no Centro Regional Nordeste (CRN) / Instituto Nacional de Pesquisas Espaciais (INPE).

A Estação Multi-Missão de Natal (EMM-Natal) é um projeto de pesquisa e desenvolvimento tecnológico, com origem no reaproveitamento de uma estação francesa destinada aos Satélites de Aplicação Científica (SACI) desenvolvido pelo INPE. Ela está sendo projetada para agregar as funcionalidades de centro de controle e missão, com base em computadores interconectados através de uma rede Ethernet. A EMM-Natal é composta por vários sistemas, dentre eles, o sistema de controle de sua antena para o rastreo de satélites, (Subsistema de Controle e Rastreo – SCR) [1].

O SCR é formado por quatro módulos: o módulo posicionador (motores, engrenagens, sensores de posição, etc.), o módulo de potência (inversores, chave contactora, etc), o módulo de comando (relés, fontes, fusíveis, etc.) e o módulo de controle (computador e placa conversora AD/DA). Neste último encontra-se o Software de Supervisão e Controle (SSC) da estação, que atualmente dispõe de interface em modo texto, desenvolvida para suprir as necessidades emergenciais de operação e passagem de parâmetros pelo servidor TCP/IP [2].

1.1 A empresa



Figura 1: Instalações do INPE no Brasil, em destaque CRN Natal-RN

O Centro Regional Nordeste do INPE, com sede em Natal, foi estabelecido por volta de 1970, quando este chamava-se Comissão Nacional de Pesquisas Espaciais (CNAE) e assinou um convênio com o Governo do Estado e Universidade Federal do RN, visando estabelecer um núcleo de apoio aos lançamentos de foguetes e balões operados na Barreira do Inferno. Ao longo dos anos, passou a desenvolver equipamentos destinados à conexão com satélites, entre outros, montou uma Plataforma de Coleta de Dados (PCD) para operar com o Sistema CLS/ARGOS e já em 1983 a primeira PCD brasileira era testada e homologada nos laboratórios da CNES (Centre National d'Études Spatiales) - França. De lá para cá a atividade de coleta de dados via satélite tem tomado dimensões nacionais, hoje com 350 unidades operando em todo o Brasil, inclusive com os satélites brasileiros.

Paralelamente, nestes últimos vinte anos, foram montados o INPE de Fortaleza- CE, São Luís-MA e Campina Grande-PB, todas, unidades subordinadas a Natal e com leque grande de atividades ligadas aos projetos mais importantes do INPE

1.2 Atividades desenvolvidas

Durante o período de estágio, compreendido entre março a junho de 2008, o aluno realizou estudos das partes constituintes da estação EMMN, em seguida, passou ao estudo das especificações técnicas que compreendem a implementação da Interface Gráfica do SRC (IGS), que objetiva facilitar a operação da Estação, especificamente do Subsistema de Controle e Rastreo (SRC).

O estagiário atuou diretamente no projeto da Interface Gráfica do SRC (IGS) da EMM-Natal, realizando atividades em diversas áreas, como: interfaces gráfica em *Qt*, comunicação em rede utilizando Sockets (sys/socket, biblioteca padrão da linguagem de programação C++) em ambiente Linux utilizando protocolo *TCP/IP* e desenvolvimento de *software*. Algumas das atividades realizadas no decorrer do estágio foram:

- Estudo dos manuais descritivos e operacionais da EMMN;
- Estudo aprofundado de *Qt*;
- Estudo aprofundado de interfaces em *Qt*;
- Estudo dos módulos componentes (SCR e SSC);
- Estudo de comunicações e protocolos de rede *Ethernet*;
- Implementação de aplicações em rede usando sockets em C++ (sys/socket, biblioteca padrão do C++/LINUX);
- Implementação de comunicação em rede entre Interface Gráfica e Servidor TCP/IP;

- Implementação de Subversion, para versionamento das versões do SRS;
- Estudo de autômatos e máquinas de estado em software;
- Estudo de viabilidade técnica para implementação de software gerador automático de analisador léxico (Flex);
- Implementação de analisador léxico para filtragem de conteúdo recebido e enviado pelo servidor TCP/IP;
- Tratamento de conteúdo recebido no servidor TCP/IP com uso de expressões regulares;
- Testes de estabilidade no analisador léxico;

2SUBSISTEMA DE CONTROLE E RASTREIO (SCR)

O Subsistema de controle e rastreo (SCR) é o sistema responsável na Estação multimissão de Natal (EMMN) pelo rastreo de satélites, isto é por controlar a antena da estação para seguir a passagem de um satélite desejado, durante sua fase observável. Abaixo, Figura 2, temos o esquema de hardware do Subsistema de Controle e Rastreo (SCR).

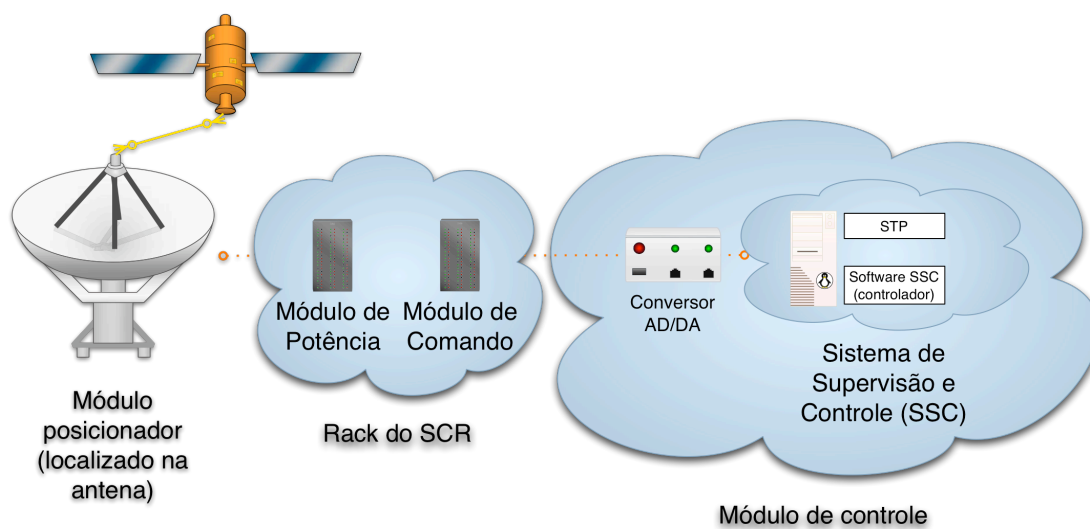


Figura 2: Estrutura de Hardware do Subsistema de Controle e Rastreo (SCR) da EMMN

Formado por quatro módulos: o módulo posicionador (motores, engrenagens, sensores de posição, etc.), o módulo de potência (inversores, chave contactora, etc), o módulo de comando (relés, fontes, fusíveis, etc.) e o módulo de controle (computador e placa conversora AD/DA)[2].

2.1 Módulo Posicionador

O módulo posicionador da antena permite que a mesma se movimente nos dois eixos, azimute e elevação, de forma independente. Cada eixo é movido por um conjunto motor mais redutor coaxial. Os motores são do tipo autossíncronos (ímãs permanentes), equipados com encoders de fábrica.

Dispositivos de fim de curso, mecânicos e elétricos, instalados nos dois eixos, limitam o deslocamento da antena parabólica, aos valores limites de segurança. O motor responsável pelo deslocamento no sentido de elevação possui um freio eletromagnético para garantir o seu travamento nas paradas. A figura 3 apresenta à antena parabólica da estação [2].



Figura 3: Antena Parabólica da EMMN vista de ângulos diferentes

2.2 Módulo de Potência

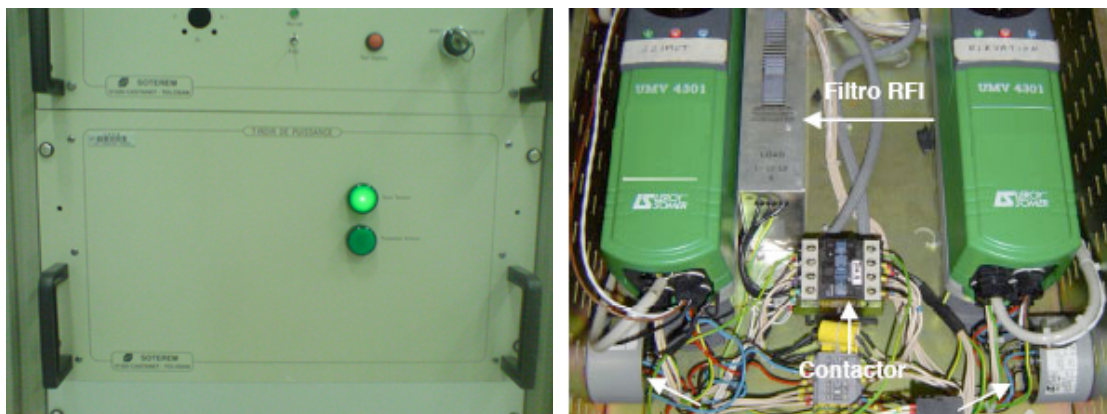


Figura 4: Vista do módulo de potência em gaveta no rack da estação e os inversores que controlam os motores

No módulo de potência estão instalados os dois inversores, modelo UMV 4301 da Leroy Somer, responsáveis pelo acionamento e controle de velocidade dos motores. Os inversores (ou drives) recebem um sinal de controle, proveniente do módulo de controle, entre -10V e +10V. Uma tensão de +10V equivale a velocidade nominal do motor (3000 rpm) num sentido, -10V no sentido contrário, e 0V a 0 rpm. O sistema de controle interno do inversor garante a convergência da velocidade real do motor com a velocidade de referência.

A alimentação dos dois inversores é controlada através de uma chave contactora, acionada pelo módulo de comando. O sinal para cortar a alimentação dos inversores, pode ser gerado devido ao acionamento de alguma proteção, como a de fim de curso, ou da chave que desliga o sistema [2].

2.3 Módulo de Comando



Figura 5: Vista do módulo de comando no rack da estação e a parte traseira com as conexões com outros módulos do SCR

O módulo de comando suporta todos os circuitos que controlam a alimentação dos inversores, a placa conversora SOTEREM 2266-1, os LEDs para visualização de eventos, e as fontes de alimentação do sistema. Ela também abriga um autotransformador (230V/48V), que alimenta o circuito de aquecimento do modulo posicionador, de modo a evitar que seus componentes sejam danificados em situações de clima frio [2].

Logo após as devidas configurações no módulo de controle (calibração da placa AD/DA, configuração e inicialização do sistema), a chave principal do módulo de comando deve ser acionada. Com isso, as fontes de alimentação são ligadas, e o software de rastreo poderá ser utilizado. A figura 7 apresenta a vista frontal do modulo de comando.

2.4 Módulo de Controle

Este módulo é composto pelo computador e pela placa conversora AD/DA (PCI 6025E) da National Instruments, que dispõe de 32 canais digitais para leitura (input) ou escrita (output), além de duas saídas analógicas. Dentre os canais digitais, 24 são fornecidos através do CI 82C55 (Programmable Peripheral Interface - PPI), dispostos em 3 portas (PA, PB e PC) de 8 canais cada. O software que realiza a supervisão e controle da estação EMM-Natal pertence a este modulo [2].

A Interface Gráfica (IGS) atua como abstração para que o operador não entre em contato diretamente com o hardware, evitando assim uma parcela de erros que são tratados com o uso da IGS.

3 SOFTWARE DE RASTREIO DE SATÉLITES (SRS)

Usuários



Figura 6: Camadas de abstração

A principal função do Software de Rastreamento de Satélites (SRS) é servir de interface entre as requisições do usuário na execução de um dado evento (posicionar a antena numa determinada posição, por exemplo) e repassá-las para o software de supervisão e controle (SSC) que é responsável por processar as requisições recebidas controlando o hardware. O Software de Rastreamento de Satélites (SRS) é composto de Interface Gráfica Supervisória (IGS) e módulo cliente de comunicação em rede Ethernet. O Software de Rastreamento de Satélites (SRS) deve apresentar ao usuário a situação atual da estação, bem como um sinal de sucesso ou fracasso ao término de cada evento solicitado. O status da EMMN é monitorado pelo software de supervisão e controle (SSC) e em seguida repassado ao Software de Rastreamento de Satélites (SRS) para apresentação ao operador, que tem a visualização a partir da interface gráfica supervisória (IGS). A Figura 6, mostra o fluxo de comunicação e troca de informações no sistema.

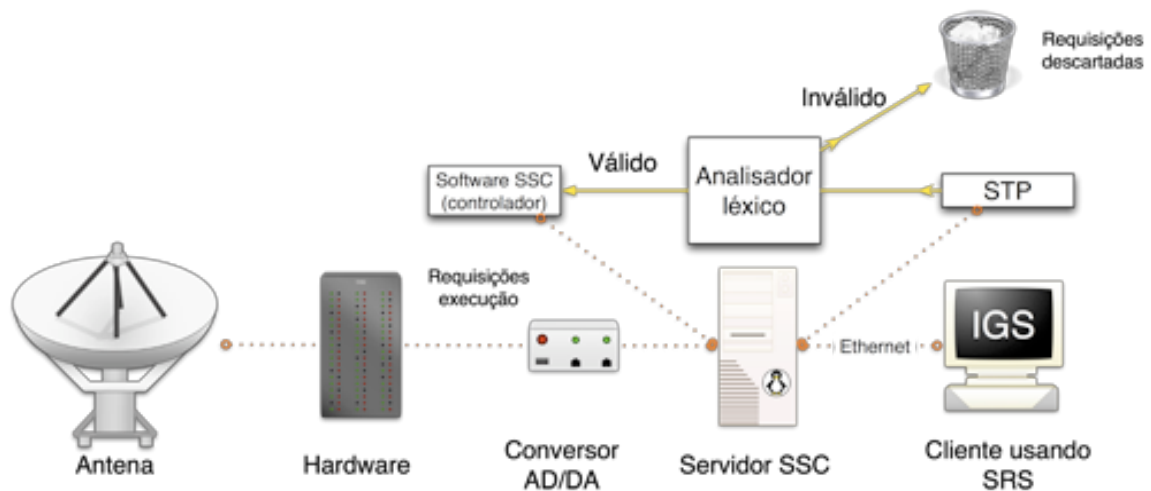


Figura 7: Estrutura do servidor TCP/IP (STP)

O retorno de informação do Software de Rastreamento de Satélites (SRS) (especificamente na Interface Gráfica Supervisória IGS) para o software de supervisão e controle (SSC) e os recursos por ele (SSC) monitorados são repassados ao SRS usando a comunicação em rede Ethernet, usando o protocolo TCP/IP. O programa responsável pela filtragem e validação de requisições é o Servidor TCP/IP que será chamado de STP.

Existem três etapas até que a requisição que o usuário fez no SRS seja entregue e executada pelo SSC. Quando o usuário executa a ação no SRS (computador cliente), através da IGS (interface gráfica supervisória), uma requisição é criada, a IGS usando o cliente TCP/IP tenta conectar-se através da porta TCP 4950 ao STP, programa em execução no Servidor do SSC, que server pra filtrar as requisições, como foi explicado anteriormente. Caso a conexão seja estabelecida a requisição é enviada apartir do SRS ao STP. Quando recebida no STP este submete a mensagem ao analisador léxico para verificar se a requisição recebida é válida e se deve ser repassado ao SSC para execução. Sendo a requisição validada pelo analisador léxico contido no STP, a requisição pode então ser tratada pelo SSC.

O STP não trata por exemplo, os valores válidos de ângulos para a movimentação da antena, já que este é um intermediário e não possui comunicação direta com a antena. Esta separação de tratamento em etapas torna o processo de adaptação e recompilação do sistema mais rápida e simples. facilitando a distribuição deste sistema para outras estações do INPE.

3.1A Interface Gráfica Supervisória (IGS) do Software de Rastreamento de Satélites (SRS)

A Interface Gráfica do SRS (IGS) tem como objetivo promover uma camada de abstração entre o SSC e os usuários da EMMN, permitindo a realização de tarefas de operação, manutenção, rastreamento e posicionamento da antena, sem atuar diretamente no Hardware da estação. Toda a operação é feita a partir de uma interface em modo texto, usando a biblioteca ncurses, desenvolvido em [1]. O operador necessariamente deve possuir familiaridade com o GNU/Linux, já que toda a aplicação de supervisão é executada sobre um terminal interpretador de comandos, conforme visualização na Figura 8.

```
***** Software de Rastreo e Controle da Estacao EMMN do INPE-CRN *****
Acoes(F1) Configuracao(F2) Agendar(F3) Ajuda(F4)
Informa??es sobre o Sistema:
Nome da fun??o  ELEVA??O  AZIMUTE
?ngulo atual   0.000000  0.000000
Tens?o aplicada 0.000000  0.000000
Corrente      0.000000  0.000000
Erro          0.000000  0.000000
Referencia    0.000000  0.000000
```

Figura 8:Interface em texto com biblioteca ncurses do SSC

A EMMN agregará diversas funcionalidades e servirá principalmente como centro de controle para as diversas missões que serão realizadas para o Sistema Brasileiro de Coleta de Dados. Muitos operadores, com diferentes graus de conhecimento sobre o GNU/Linux, devem atuar sobre a mesma, portanto, é requisito que o sistema seja de fácil utilização e de rápido aprendizado. Definiu-se então o modelo de distribuição baseado em rede com interface gráfica, conforme Figura 9.

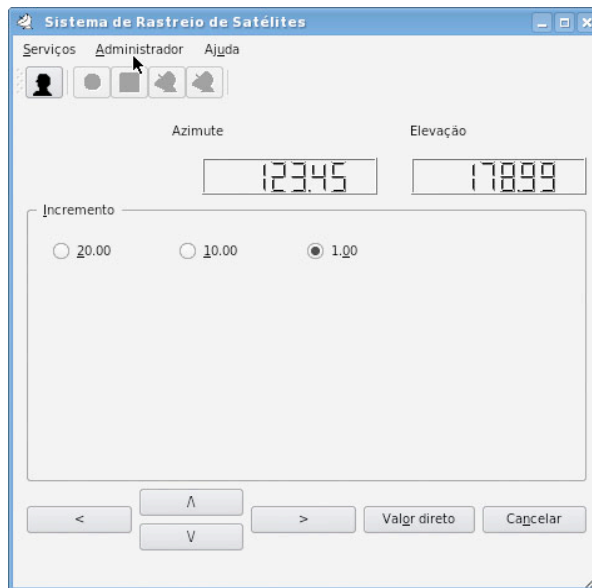


Figura 9: Interface em Qt3

A interface gráfica foi criada utilizando a biblioteca de classes, sob licença GPL (General Public License), para C++ e conjunto de ferramentas para construção de aplicações gráficas, QT 3, como framework para a parte gráfica da aplicação [10]. Este trabalho é a continuação do desenvolvimento em [3]. O BSD Sockets foi a biblioteca utilizada na criação dos sockets e o GNU/Linux é o sistema operacional usado para desenvolver e executar a aplicação.

3.2 Programa Servidor TCP/IP

O principal componente do STP é um analisador léxico que tem como função validar ou rejeitar as requisições recebidas através da rede. Neste trabalho o analisador léxico analisará não código fonte de um programa, mas sim requisições recebidas em formato de string ASCII enviadas através da rede utilizando sockets TCP/IP.

Para validar o comando o analisador léxico, localizado no STP, utilizando-se de expressões regulares que descrevem o protocolo desenvolvido para comunicação entre IGS e SSC, se a requisição feita atende os requisitos previstos. Uma mostra de alguns comandos deste protocolo podem ser vistos na Tabela 1, o protocolo completo pode visto no Apêndice A.

Comando	Parâmetro	Descrição	Retorno ao sock	Retorno ao GUI
MV	<valueAZ>:<valueEL>	Move a antena manualmente até a posição escolhida. Se um	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'

		eixo não for movido, usar "." no lugar do valor. Ex. MV :.120		
RC		Cancela o rastreo em execução.	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'

Tabela 1. Protocolo de comunicação da EMMN

Para o comando MV do protocolo, conforme Tabela 1, para que o analisador léxico durante o processo de varredura esteja apto a encontrar este padrão de texto, é necessário descrevê-lo como sendo uma expressão regular. Portanto, a expressão regular correspondente seria:

MV valueAzimute:valueElevação, padrão de texto para expressão regular (3.2.1)

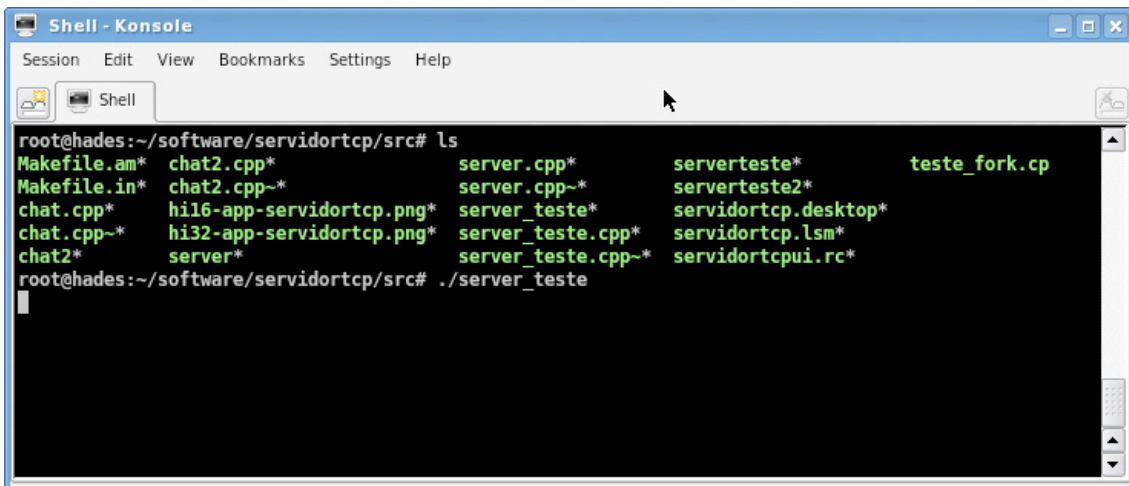
Digito={0-9}, usando facilitador de expressão regular, nomeando expressão (3.2.2)

(MV)+()+digito+(:)digito+ , Expressão regular para o comando (3.3.3)

Evitando que por exemplo alguém através da rede requisiite que o conteúdo do disco rígido do servidor seja deletado, Isso porque o analisador léxico contido no STP irá descartar a requisição.

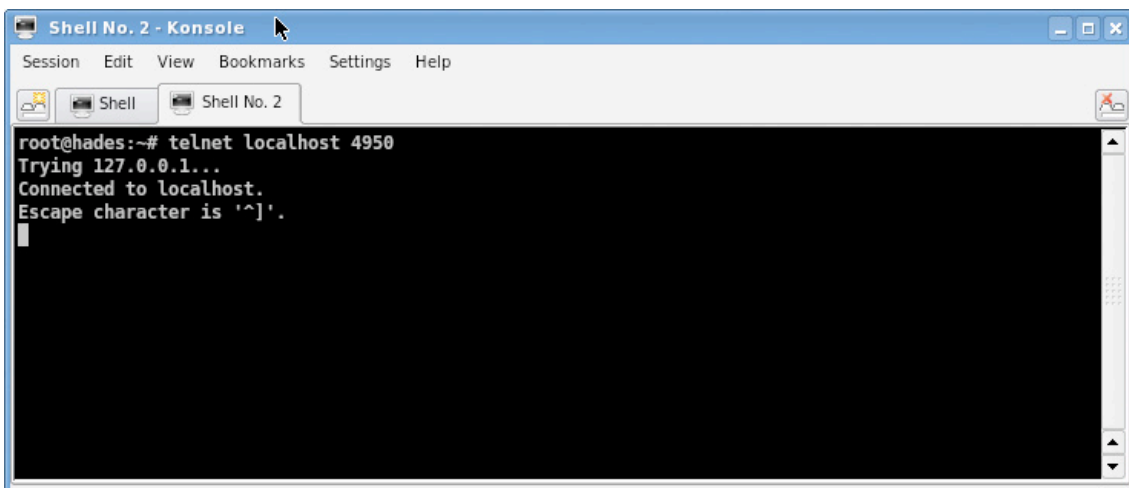
A comunicação com o hardware é feita através do SSC, que utilizando-se do *comedi*, oferece drivers para uma variedade de dispositivos de aquisição de dados e da biblioteca *open source comedlib* que oferece facilidade para manipulação do dispositivo interfaceado [11]. Essas bibliotecas tem como função principal permitir a comunicação com o *hardware* através da placa conversora AD/DA, PCI 6025E da *National Instruments*.

A seguir, da Figura 10 à Figura 16: Resposta recebida no cliente apartir de dados enviados ao servidor TCP/IP que os processou e retornou essa mensagem, temos o esquema simulado da comunicação do programa STP.



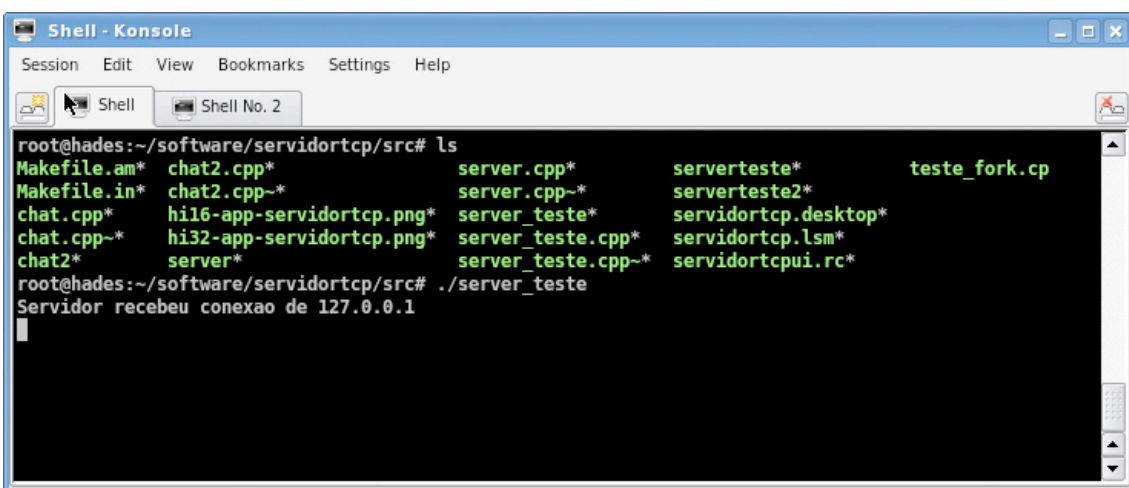
```
Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell
root@hades:~/software/servidortcp/src# ls
Makefile.am* chat2.cpp* server.cpp* serverteste* teste_fork.cp
Makefile.in* chat2.cpp~* server.cpp~* serverteste2*
chat.cpp* hi16-app-servidortcp.png* server_teste* servidortcp.desktop*
chat.cpp~* hi32-app-servidortcp.png* server_teste.cpp* servidortcp.lsm*
chat2* server* server_teste.cpp~* servidortcpui.rc*
root@hades:~/software/servidortcp/src# ./server_teste
```

Figura 10: Exemplo de servidor TCP/IP aguardando conexões



```
Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell Shell No. 2
root@hades:~# telnet localhost 4950
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
```

Figura 11: Simulação de conexão ao serviço



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell Shell No. 2
root@hades:~/software/servidortcp/src# ls
Makefile.am* chat2.cpp* server.cpp* serverteste* teste_fork.cp
Makefile.in* chat2.cpp~* server.cpp~* serverteste2*
chat.cpp* hi16-app-servidortcp.png* server_teste* servidortcp.desktop*
chat.cpp~* hi32-app-servidortcp.png* server_teste.cpp* servidortcp.lsm*
chat2* server* server_teste.cpp~* servidortcpui.rc*
root@hades:~/software/servidortcp/src# ./server_teste
Servidor recebeu conexao de 127.0.0.1
```

Figura 12: Servidor reconhece que recebeu uma conexão e lista o IP


```

root@hades:~# telnet localhost 4950
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
teste
$STATUS:<AZres><ELEres><LEDS:11111110><MANUAL:0><CHAVE_EMERGENCY:0><KA:><KB:><KAZ:><KEL:>
Connection closed by foreign host.
root@hades:~#

```

Figura 13: Enviado a string Teste, e o servidor responde com mensagem em formato de teste, simbolizando apenas que recebeu a string corretamente

```

root@hades:~/software/servidortcp/src# ls
Makefile.am*  chat2.cpp*          server.cpp*          serverteste*          teste_fork.cp
Makefile.in*  chat2.cpp~*         server.cpp~*         serverteste2*
chat.cpp*     hi16-app-servidortcp.png*  server_teste*       servidortcp.desktop*
chat.cpp~*    hi32-app-servidortcp.png*  server_teste.cpp*   servidortcp.lsm*
chat2*        server*              server_teste.cpp~*  server_teste.rc*
root@hades:~/software/servidortcp/src# ./server_teste
Servidor recebeu conexao de 127.0.0.1
0 numero de bytes eh: 7
Recebido: teste
teste

```

Figura 14: Servidor exibe a mensagem na tela e diz o número de bytes recebidos

```

root@hades:~/software/servidortcp/src# ./server_teste
Servidor recebeu conexao de 127.0.0.1
0 numero de bytes eh: 17
Recebido: MV 124.455:178.99MV 124.455:178.99

```

Figura 15: Simulação de envio de comando próximo ao formato que será adotado

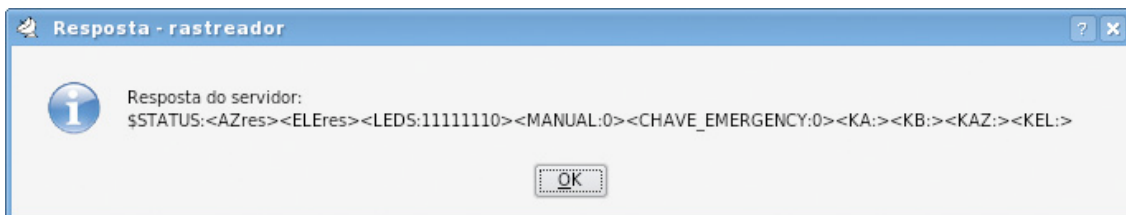


Figura 16: Resposta recebida no cliente a partir de dados enviados ao servidor TCP/IP que os processou e retornou essa mensagem

3.3 Interfaces ao Usuário

O SRS apresenta uma janela principal ao usuário composta de *menus* que são acessados através das teclas de funções ou do *mouse*. Cada item presente nos *menus* também pode ser acessado através do *mouse* ou com uso de teclas de atalho compostas, no formato *alt+[letra inicial do item]*. A Tabela 2, apresenta os *menus* e os respectivos itens que aparecem na janela principal do SRS.

Item	Serviços	Administrador	Ajuda
1	Passagens - Visualizar passagens - Adicionar passagens	Usuários	Ajuda
2	Controle -Ligar o sistema de potência -Mover Antena Manualmente -Testar leds da gaveta de pilotagem -Acertar relógio do sistema com o UTC -Monitorar	Log - Usuários - Visualizar - Limpar - Erros - Visualizar - Limpar - Sistema - Visualizar - Limpar	Sobre o rastreador
3	Sair	Controlador	Sobre KDE

Tabela 2: Menus da tela principal



Figura 17: Menu da tela principal do SRS

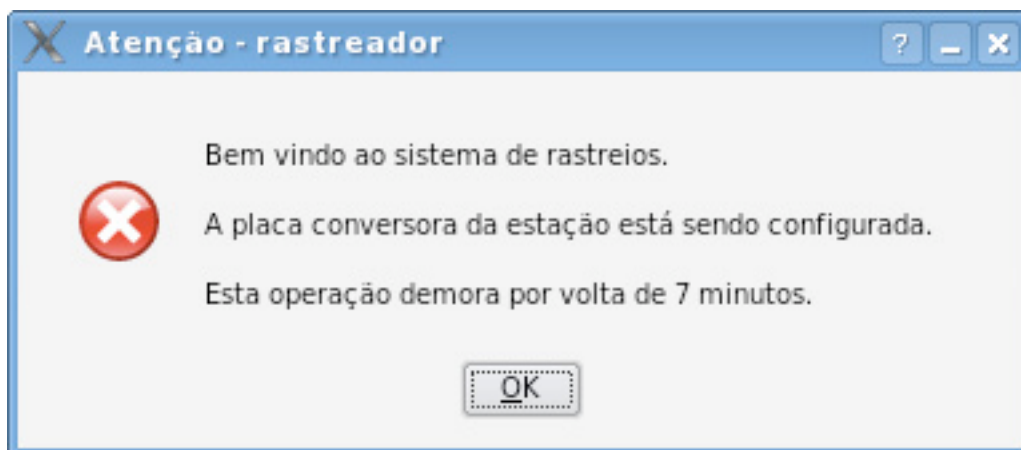


Figura 18: Aviso inicial sobre a configuração de placa conversora ADDA no SSC

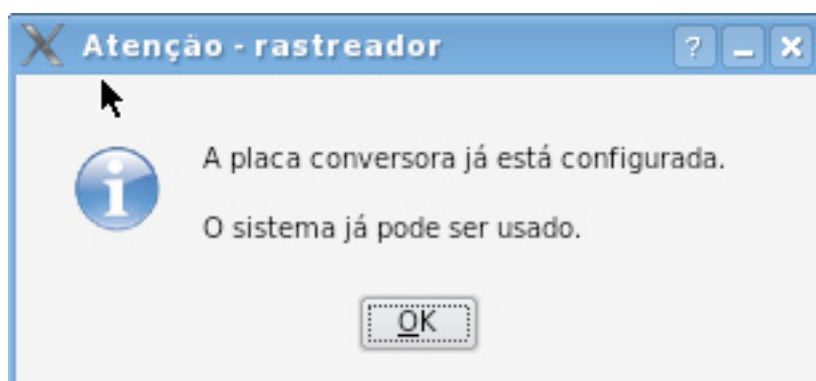


Figura 19: Aviso alertando de que a placa conversora ADDA está pronta

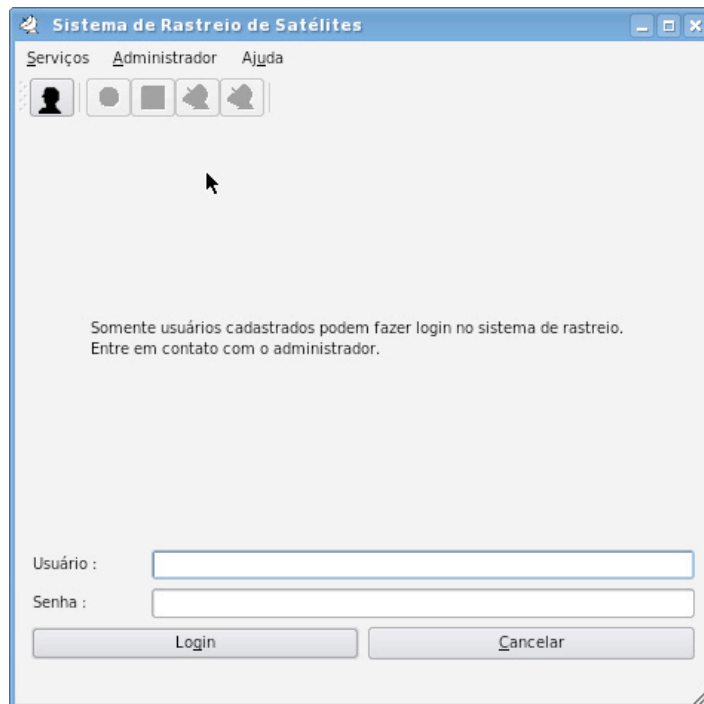


Figura 20: Tela inicial para autenticar no sistema, o acesso aos menus é liberado após ter sido autenticado

3.3.1 Menus

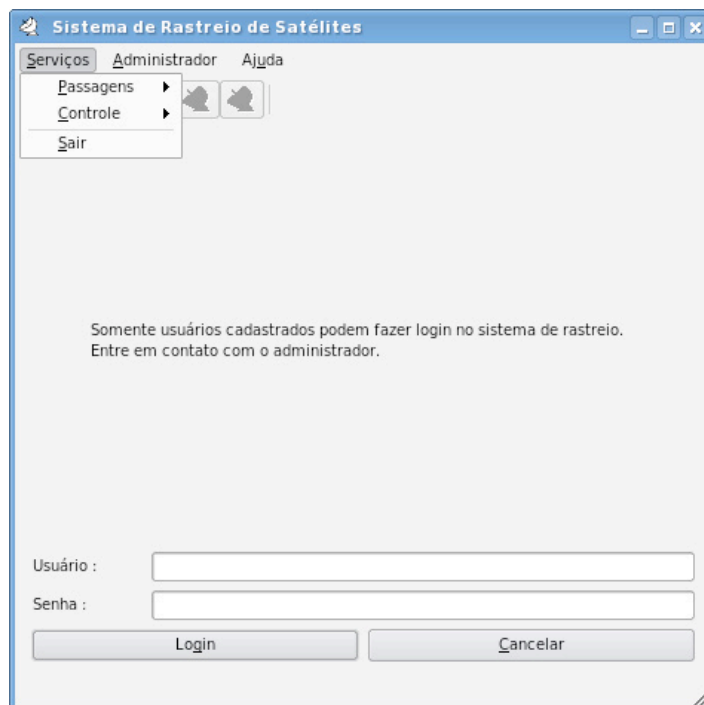


Figura 21: Menu Serviços, detalhes vide Tabela 1

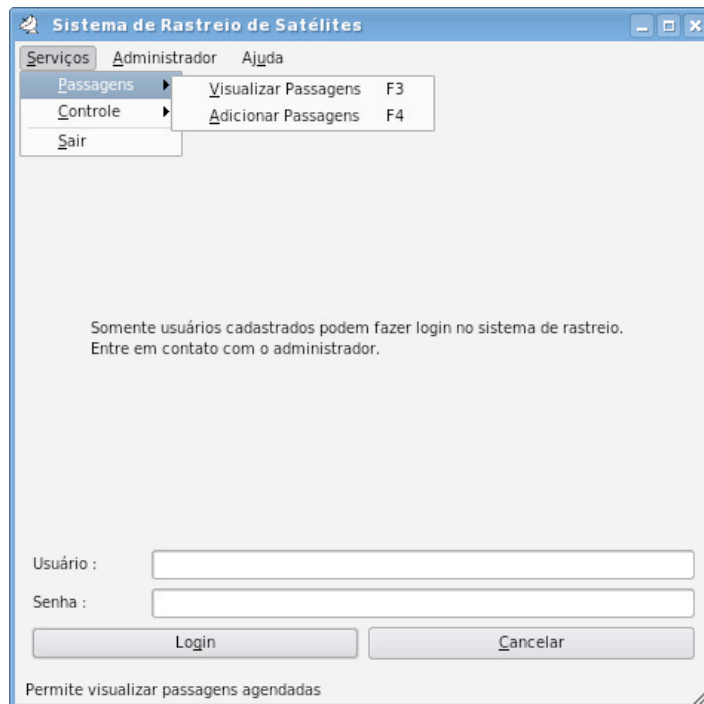


Figura 22: Submenu Passagens, referente as efemêrides. Opção de visualizar as já cadastradas e adicionar novas

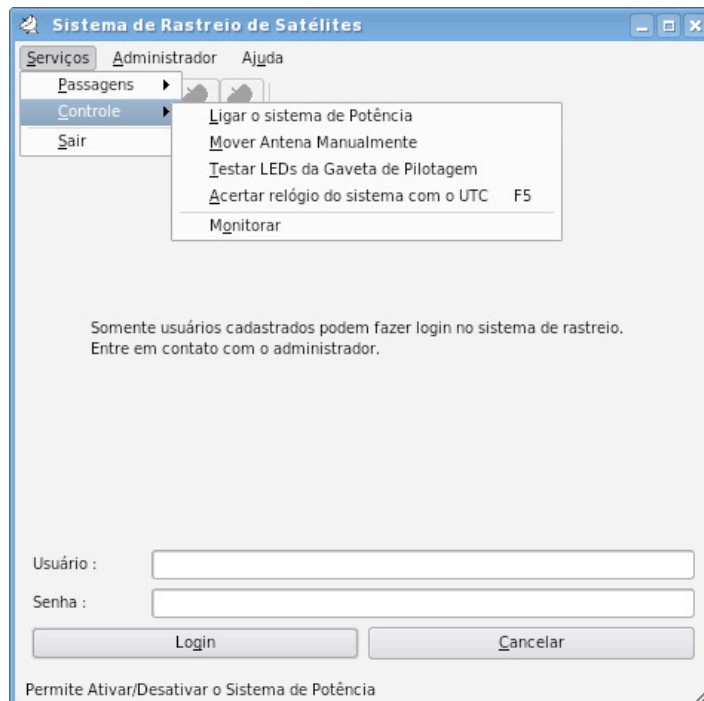


Figura 23: Menu controle, onde as principais ações sobre a estação são feitas

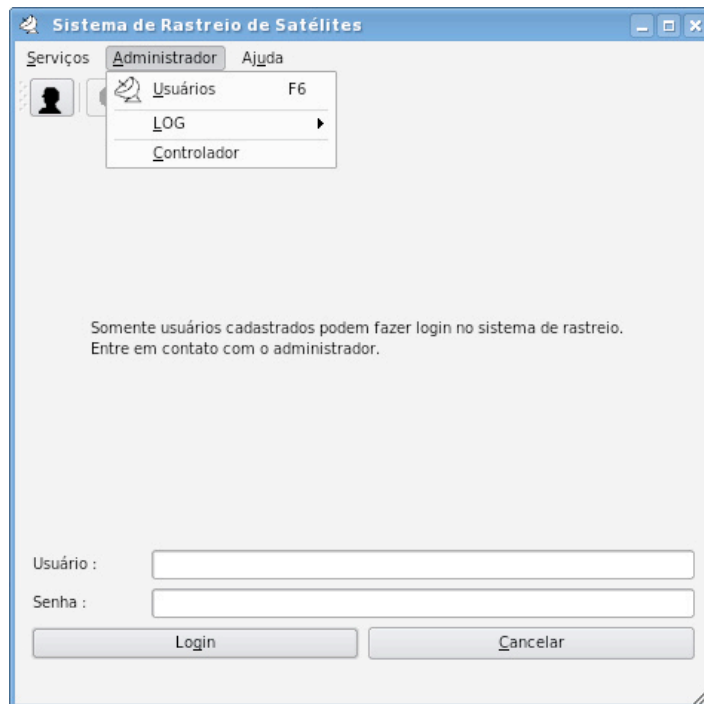


Figura 24: Menu Administrador com opções avançadas de configuração, sendo permitido acesso a tais opções apenas a usuário com privilégios administrativos

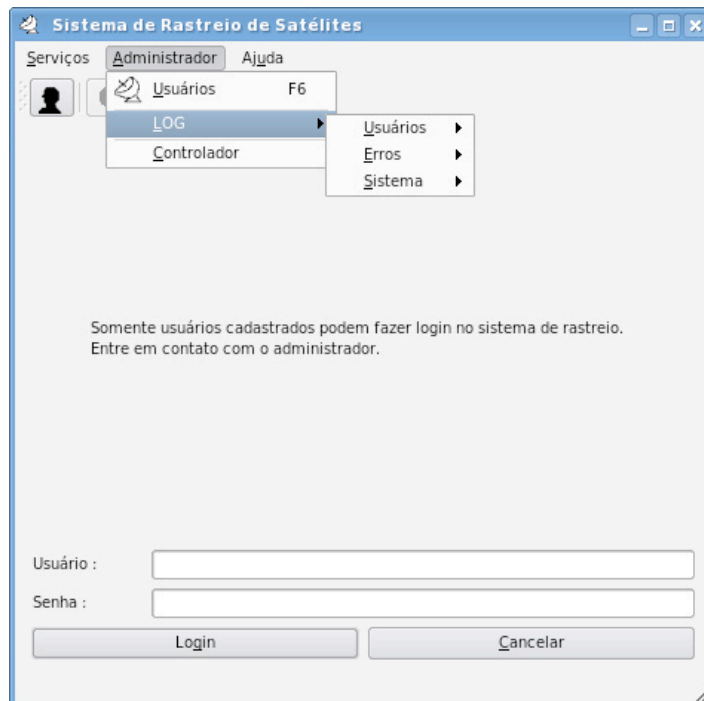


Figura 25: Menu logs, onde pode-se encontrar os relatórios dos principais eventos registrados no SRS

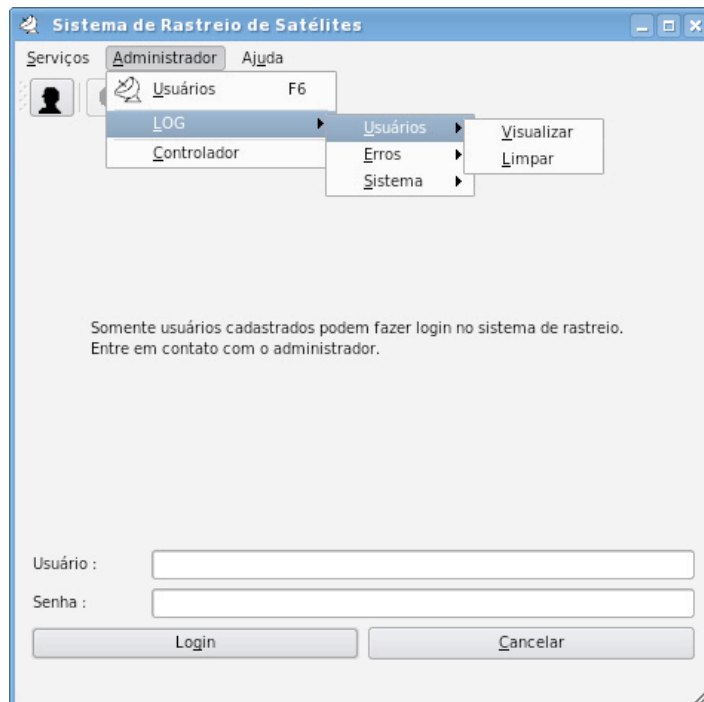


Figura 26: Submenu Usuários, onde pode-se encontrar relatórios dos principais eventos relacionados a usuários do SRS

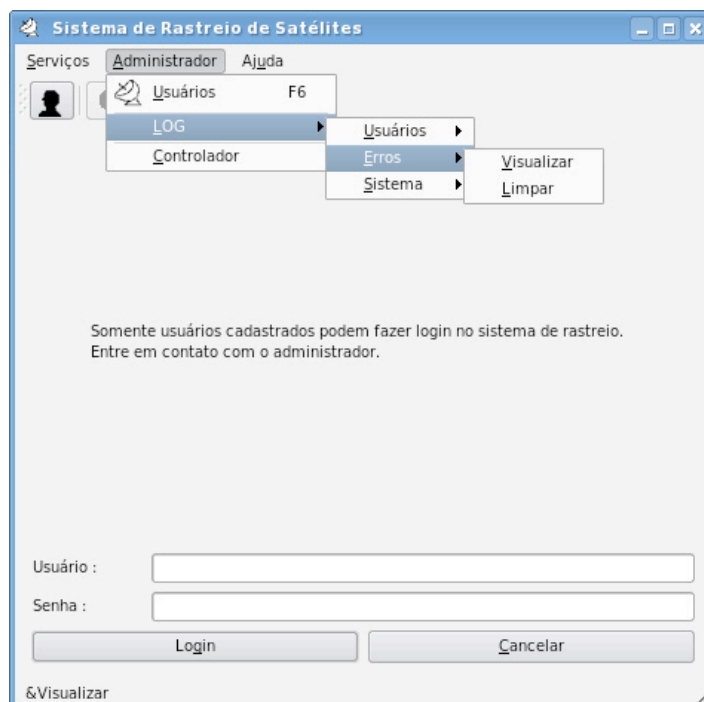


Figura 27: Submenu Erros, onde pode-se encontrar relatório dos principais eventos relacionados a erros do SRS

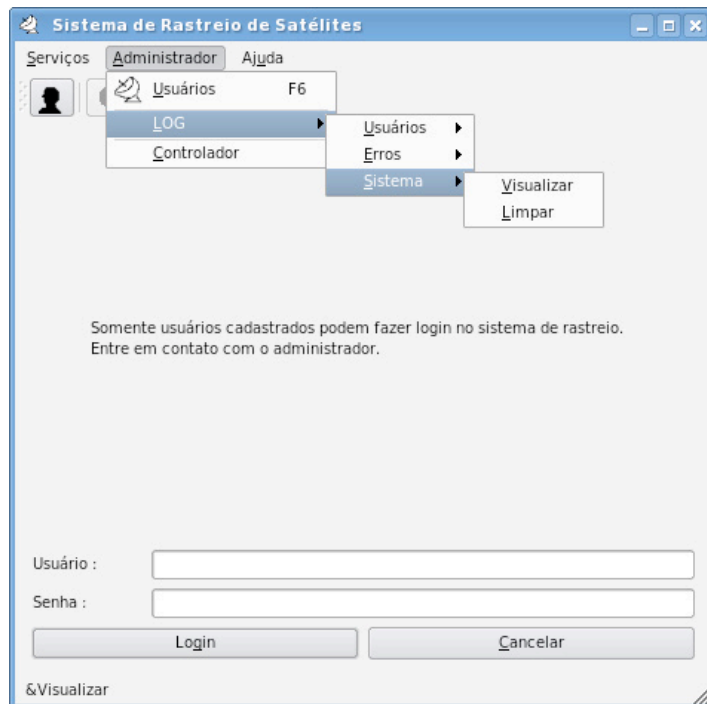


Figura 28: Submenu Sistema, onde pode-se encontrar relatórios dos principais eventos relacionados aos sistema SRS

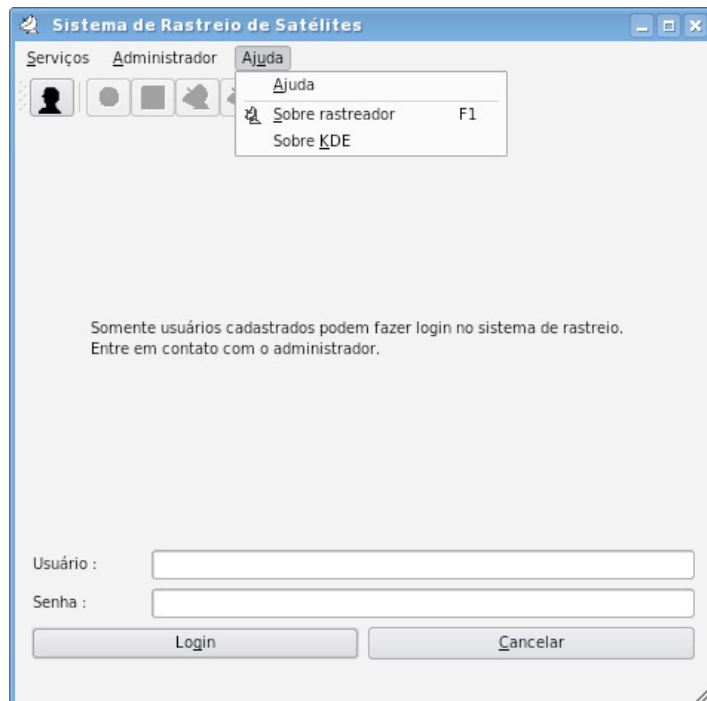


Figura 29: Menu Ajuda, pode-se conseguir auxílio sobre utilização, configurações e dicas sobre o SRS

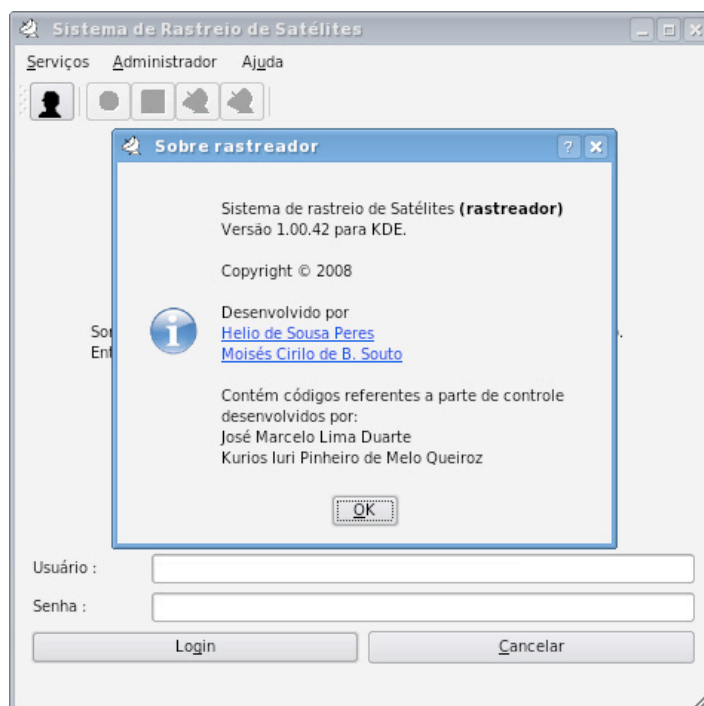


Figura 30: Informação sobre os desenvolvedores do SRS e SCR da EMMN

3.3.2 Telas

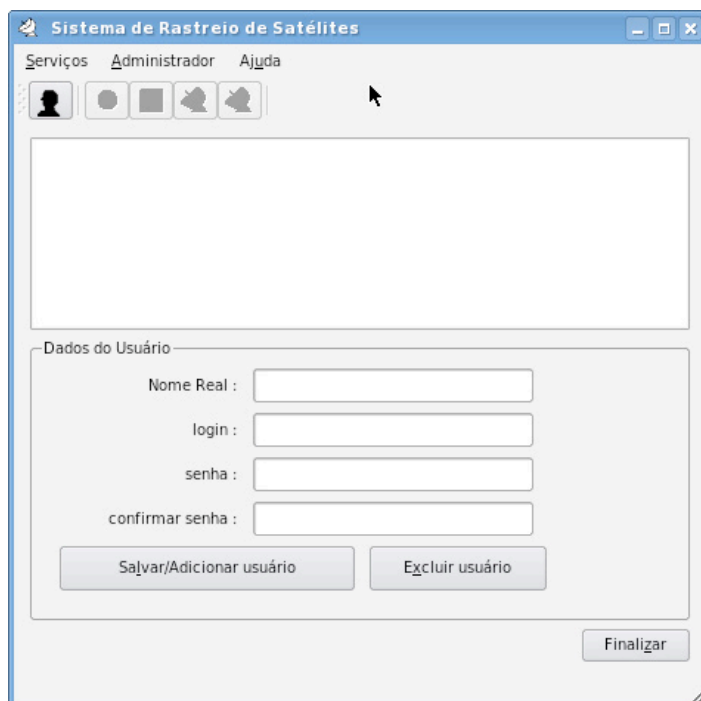


Figura 31: Tela Adicionar usuário, acesso a esta tela unicamente para usuários com privilégios administrativos



Figura 32: Tela de ajuste dos parâmetros do controlador PI e do controlador PID, acesso unicamente com privilégios administrativos

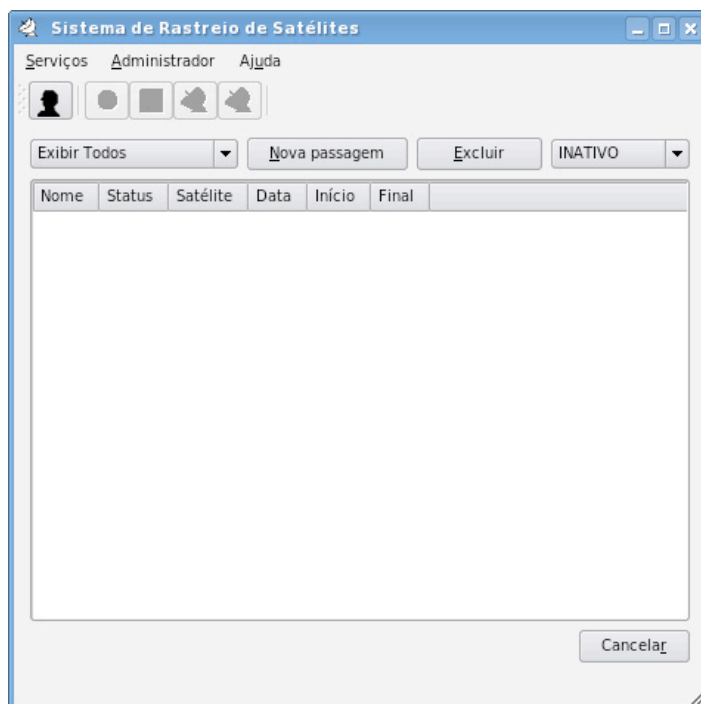


Figura 33: Tela de acompanhamento das efemérides cadastradas, sendo possível filtrar entre ativas, inativas e canceladas

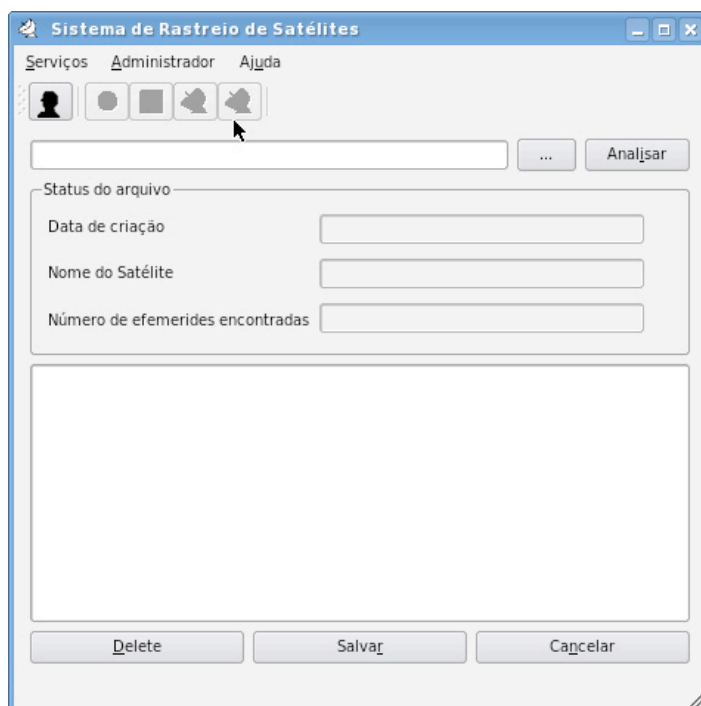


Figura 34: Tela para acrescentar arquivos efemérides para rastreo de satélites



Figura 35: Tela para Movimentação manual da Antena



Figura 36: Tela para teste dos leds da gaveta de pilotagem e acerto do relógio do sistema

3.3.3 Ações

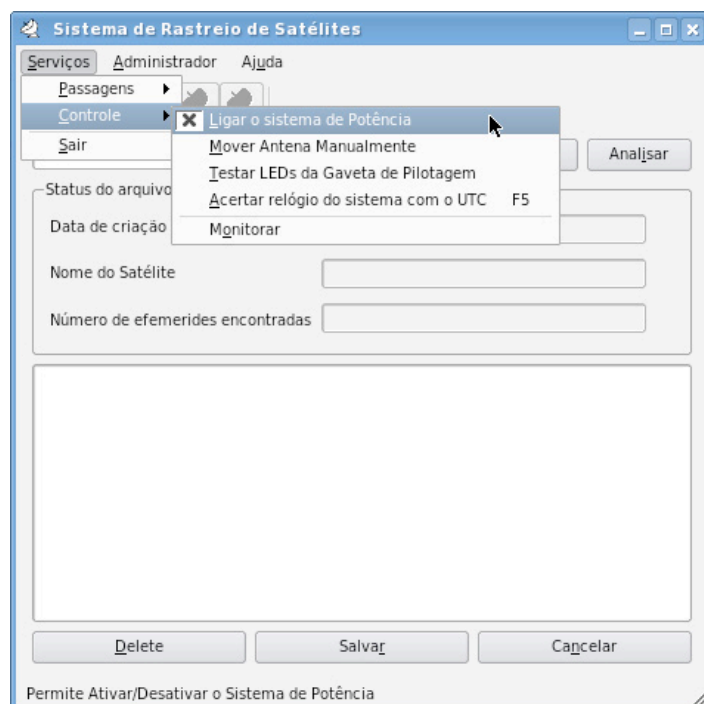


Figura 37: Ligar ou desligar o sistema de potência da EMMN

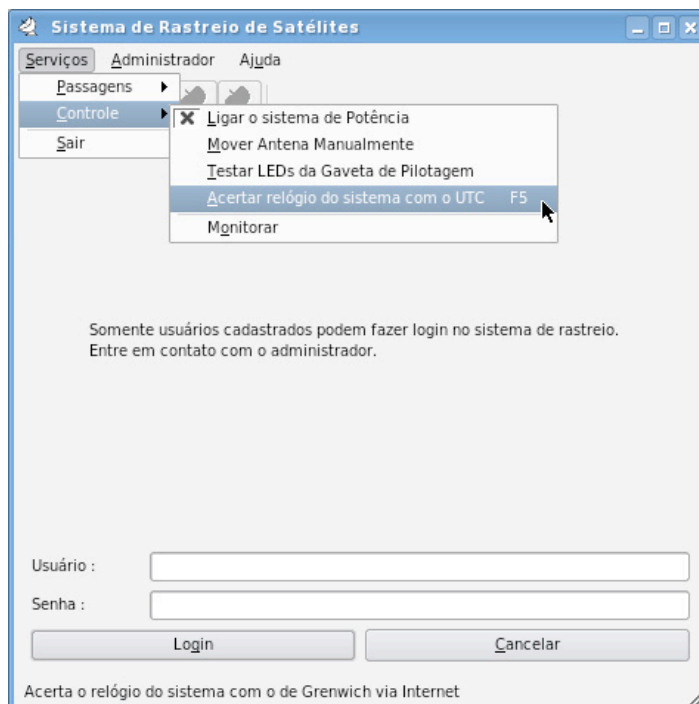


Figura 38: Sincronizar o relógio do sistema com o relógio UTC

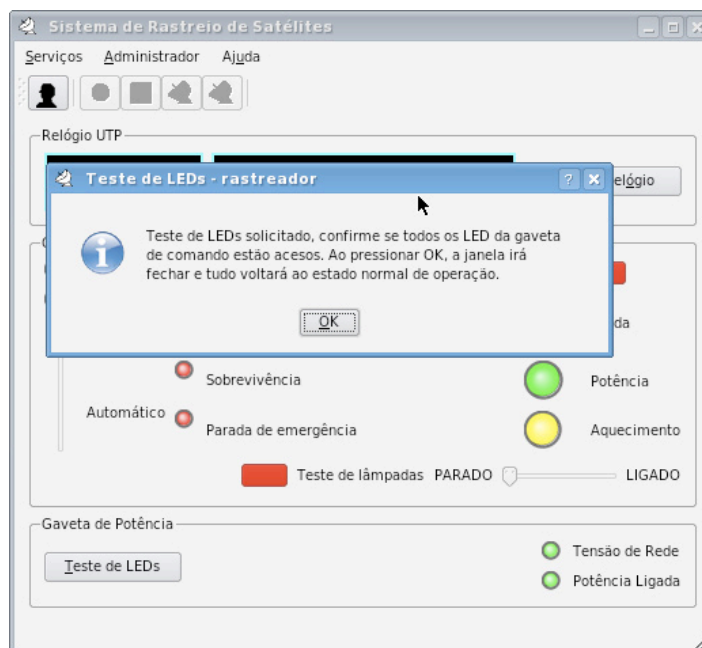


Figura 39: Alerta do teste dos leds

3.4 Interfaces de software

O SRS tem como requisito para o correto funcionamento os seguintes softwares:

Sistema Operacional: GNU/Linux
Distribuição: Slackware e Debian
Versões: Slackware 12 e 12.1; Debian Sarge (3.1r2)
Ambiente gráfico: K Desktop Environment (KDE)
Versão: 3.3

3.5 Interfaces de comunicação

O SRS comunica-se com o SSC através de uma conexão TCP/IP pela porta 4950.

3.6 SSC

O detalhamento do funcionamento do SSC pode ser melhor compreendido no trabalho de Kurios Iuri Pinheiro de Melo Queiroz [2].

4 CONCLUSÕES E TRABALHOS FUTUROS

O trabalho exigiu um estudo minucioso de diversos assuntos relevantes a produção de sistemas modernos e robustos. De todos os objetivos previstos a Interface Gráfica está 90% pronta, em continuidade ao trabalho desenvolvido em [3]. Desenvolvemos comunicação entre interfaces gráfica e STP, sobre rede Ethernet. Atualmente, a fase de desenvolvimento é o analisador léxico.

Interagindo com colegas de trabalho com sólida formação e vasta experiência no desenvolvimento de novos produtos e produção científica, tive a oportunidade de conhecer as metodologias do desenvolvimento de projetos científico e com isso abrir leque de novas idéias e conhecimentos.

O INPE sempre permitiu, incentivou e proporcionou ambiente propício ao desenvolvimento técnico – científico, dando sempre infraestrutura e conhecimento para o engrandecimento profissional do estagiário.

Espera-se conseguir renovação da bolsa para que se possa continuar o desenvolvimento de tão importante projeto.

- Tratamento de conteúdo recebido no servidor TCP/IP com uso de expressões regulares;
- Testes de estabilidade no analisador léxico;
- Passagem de parâmetros por referência para o software de controle (SCR);
- Integração em programa de controle por parâmetros via *socket* com interface gráfica;
- Testes operacionais e documentação;

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] DUARTE, J. M. L e CARVALHO, M. J. M. **Programa para Rastreo de Satélite da Estação EMMN**. Natal, RN: INPE, 2005;
- [2] Queiroz, K. I. P. M. e Carvalho, M. J. M. **Sistema de Controle de Apontamento para uma Antena da Estação TT&C de Natal**. Natal, RN: INPE, 2006;
- [3] PERES, H. S. e CARVALHO, M. J. M. **Desenvolvimento de aplicação para Linux utilizando Kdevelop e Qt**. Natal, RN: INPE, 2007;
- [4] BELLOMO, M. **Administração do Linux**. Rio de Janeiro, RJ: Campus, 2000;
- [5] TANENBAUM, A. S. e WOODHULL, A. S. **Sistemas Operacionais: Projeto e implementação**. 2^a Ed. Porto Alegre, RS: Bookman, 2000;
- [6] ALVES, M. M. **Sockets Linux**. Rio de Janeiro, RJ: Brasport, 2008;
- [7] TANENBAUM, A. S. **Redes de Computadores**. 4^a Ed. Rio de Janeiro, RJ: Elsevier, 2003;
- [8] Holzner, S. *C++ Black Book*, 1^a Ed. Markron Books, 2001;
- [9] LOUDEN, K. C. **Compiladores: Princípios e práticas**. São Paulo, SP: Pioneira Thomson Learning, 2004;
- [10] BLANCHETTE, J. e SUMMERFIELD, M. **C++ GUI Programming With QT3.1** 1^a Ed. Pearson Education, 2004;
- [11] SCLEEF, D., HESS MORI, F. e ABBOTT, I. **Comedi: linux control and measurement device interface**. Disponível em <<http://www.comedi.org/>> Acesso em 11 abril de 2009;
- [12] STEVENS,W. R. **TCP/IP Illustrated. Vol. 1: The protocols**, 1^a Ed. Addison Wesley, 1994;
- [13] STEVENS,W. R. **UNIX Network Programming**, 1^a Ed., Prentice Hall, 1990;

APÊNDICE A – Protocolo de Comunicação da EMMN

Comando	Parâmetro	Descrição	Retorno ao sock	Retorno ao GUI
MV	<valueAZ>:<valueEL>	Movimenta a antena manualmente até a posição escolhida. Se um eixo não for movido, usar "." no lugar do valor. Ex. MV .:120	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'
RC		Cancela o rastreamento em execução.	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'
RS	'/dir/file'	Executa imediatamente o rastreamento do arquivo.	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'
STATUS		Retorna vários parâmetros de status do sistema.	0: OK 20: Erro 255: Erro	\$STATUS:<AZres><ELEres><LEDS:11111110><MANUAL:0><CHAVE_EMERGENCY:0><KA:0><KB:0><KAZ:0><KEL:0>
NTPDATE	<value>	Envia comando para ajuste do relógio pelo servidor NTP ou busca a data e hora do servidor. 0 : retorna a hora/data 1 : ajusta a hora/data	0: OK xx: Erro 20: Erro 255: Erro	\$NTP:TIME=hh:mm;DATE=dd/mm/yy
GETFILE	<value>:'/dir/file'	Envia ou recebe arquivo pela rede. O diretório digitado sempre corresponde ao servidor e a cópia sempre é executada via scp. 0 : Grava/sobrescreve '/dir/file' do servidor; 1 : Retorna o arquivo no diretório '/tmp' do cliente.	0: OK xx: Erro 20: Erro 255: Erro	\$FILE:'/dir/file'
GETVAR	'VAR_NAME'	Retorna o valor de uma variável de ambiente do servidor.	0: OK 20: Erro 10: Erro 255: Erro	\$VAR:VAR_NAME=value
SETVAR	'VAR_NAME=value'	Grava valor de uma ou mais variáveis de ambiente no servidor.	0: OK 20: Erro 10: Erro 255: Erro	\$VAR:VAR_NAME=value

UPDATE		Força a atualização das passagens do servidor no crontab	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'
LEDS	<value>	Liga ou desliga todos os LED da gaveta de comando. 0 : Estado normal de operação 1 : Todos os LEDS acesos	0: OK 20: Erro 50: Erro 255: Erro	\$MSGBOX:<type>,'message'
RESET		Força o reinício de todo o sistema do servidor para o estado 0.	0: OK 20: Erro 50: Erro 255: Erro	\$HANG
REBOOT		Reinicia todo o sistema sem chegar ao estado 0.	0: OK 20: Erro 50: Erro 255: Erro	\$HANG
DEFAULTS		Retorna todas as variáveis de ambiente ao estado 0.	0: OK 20: Erro 50: Erro 255: Erro	\$MSGBOX:<type>,'message'
LOGIN	'username:password'	Pede conexão ao servidor via login.	0: OK 25: Erro 255: Erro	Se retornar ao sock !=0: \$MSGBOX:<type>,'message' Se retornar ao sock =0: \$UID:'hexa_value'
LOGOFF	UID	Pede desconexão do servidor	0: OK 20: Erro 255: Erro	\$MSGBOX:<type>,'message'
ERROR		Retorna todos os erros ativos. O erro 20 informa uma interrupção requisitada pelo controlador.	0: OK 20: Erro 255: Erro	\$ERROR:number,...,number
GETERROR	<error_number>	Retorna a última mensagem para o GUI do erro selecionado	0: OK 255: Erro	Variável. Depende do erro.
GETERROR	20	Requisição do controlador. Ocorre em erros não monitorados, avisos de sistema ou fora do escopo. Requisitando GETERROR 20, pode gerar qualquer tipo de retorno ao GUI.	20: Erro	Variável. Depende do caso.

