



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

SOFTWARE DE RASTREIO DE SATÉLITES PARA UMA ESTAÇÃO SOLO DE TT&C

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBITI/INPE/CNPq)**

Felipe Cortez de Sá (UFRN, Bolsista PIBITI/CNPq)
E-mail: felipe.cortez@crn.inpe.br

Dr. José Marcelo Lima Duarte (INPE, Orientador)
E-mail: jmarcelo@crn.inpe.br

Julho de 2017

SUMÁRIO

| | Pág. |
|---|------|
| 1 INTRODUÇÃO..... | 1 |
| 2 OBJETIVOS DO TRABALHO | 1 |
| 3 VISÃO GERAL DO SISTEMA..... | 2 |
| 3.1. INSERÇÃO E EDIÇÃO DE DADOS | 2 |
| 3.2. VISUALIZAÇÃO DE DADOS E ESTADO | 2 |
| 4 MATERIAIS E MÉTODOS UTILIZADOS | 2 |
| 4.1. METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE | 2 |
| 4.2. FRAMEWORK DE INTERFACE GRÁFICA | 3 |
| 4.3. BUG-TRACKING | 4 |
| 4.4. DOCUMENTAÇÃO | 4 |
| 4.5. BIBLIOTECAS | 4 |
| 4.6. PROTOTIPAÇÃO | 5 |
| 4.7. SIMULAÇÃO..... | 5 |
| 5 ANÁLISES E RESULTADOS..... | 5 |
| 6 CONCLUSÕES..... | 6 |
| APÊNDICE A..... | 7 |
| GLOSSÁRIO | 9 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 10 |

LISTA DE ILUSTRAÇÕES

| | Pág. |
|--|------|
| Figura 1 – Editor de interfaces | 7 |
| Figura 2 – Imagem do software anteriormente desenvolvido | 7 |
| Figura 3 – Imagem do novo software com interface gráfica | 8 |

LISTA DE SÍMBOLOS E ABREVIATURAS

| | |
|------|---|
| INPE | Instituto Nacional de Pesquisas Espaciais |
| CRN | Centro Regional do Nordeste |
| TT&C | <i>Telemetry, Tracking and Commanding</i> |
| TLE | <i>Two-Line Element set</i> |
| STK | <i>Systems Tool Kit</i> |
| IDE | <i>Integrated Development Environment</i> |

RESUMO

Este trabalho descreve o processo de desenvolvimento de um software com interface gráfica utilizando o *framework* Qt para comandar uma estação multi-missão. O programa mantém uma lista de satélites a serem rastreados com prioridade ajustável e, com base nas passagens desses satélites, movimenta uma antena através de comandos enviados por uma porta serial a um dispositivo Arduino. Foi feito um teste prático com sucesso, posicionando uma antena para rastrear um satélite. Resta corrigir *bugs*, adicionar novas funcionalidades, elaborar uma documentação para desenvolvedores e escrever um manual para usuários.

Palavras-chave: Software. Interface gráfica. Rastreo de satélites.

1 INTRODUÇÃO

O INPE CRN possui em Natal uma estação de TT&C dedicada ao satélite SACI. Infelizmente, essa estação está inativa desde a falha no lançamento desse satélite. Num esforço para reaproveitar essa estrutura, o INPE CRN vem desenvolvendo uma estação solo que possa atender múltiplas missões utilizando componentes da estação de TT&C do SACI. Recentemente, um software de rastreamento foi desenvolvido com interface em linha de comandos, recebendo como entrada dados gerados pela ferramenta STK – *Systems Tool Kit* e enviando comandos para um sistema de posicionamento continuamente.

Concluído seu desenvolvimento, desejou-se ter uma interface mais amigável para o usuário. Este trabalho trata do desenvolvimento de uma interface gráfica para o programa anteriormente escrito e adaptações feitas para facilitar a operação pelo usuário.

2 OBJETIVOS DO TRABALHO

O objetivo principal do trabalho foi desenvolver uma interface gráfica intuitiva para o código anteriormente escrito, adicionando novas funcionalidades ao programa.

Quanto à interface gráfica, o programa deve minimizar a interação necessária pelo usuário e obedecer às dez heurísticas de usabilidade para desenho de interfaces de usuário propostas por Nielsen (1994).

Para a implementação geral, o desenvolvimento do projeto é guiado pela árvore de atributos de qualidade de Boehm et al. (1977), atentando-se principalmente à confiabilidade, facilidade de entendimento e modificabilidade.

Após chegar-se a uma versão inicial estável, atualizações subsequentes devem adicionar melhoramentos como atualização automática de TLEs para

satélites rastreados através de conexão com servidor do CelesTrak e posicionamento manual da antena pelo usuário.

Para auxiliar futuros desenvolvedores, uma documentação referente ao código necessita ser elaborada. Para usuários do programa, deverá haver um manual de uso do software.

3 VISÃO GERAL DO SISTEMA

3.1. INSERÇÃO E EDIÇÃO DE DADOS

O programa permite o cadastro de TLEs, isto é, dados codificados em duas linhas referentes à órbita de um satélite, e a partir disso e da posição do observador — coordenadas da antena, no caso — calcula as efemérides para o rastreo do satélite com base no algoritmo SGP4 (*simplified perturbations models*), enviando-as para o sistema de controle de posição da antena através de conexão serial com um dispositivo Arduino.

Também é possível escolher a prioridade de rastreo de cada satélite cadastrado, isto é, caso haja sobreposição no tempo de duas ou mais passagens, a ordem na lista define qual dos satélites será apontado pela antena. É possível alterar a prioridade de um satélite selecionando-o e pressionando os botões [▲] e [▼].

3.2. VISUALIZAÇÃO DE DADOS E ESTADO

O usuário pode visualizar as informações das próximas passagens, como elevação máxima e horários de aquisição e perda de sinal através de uma tabela e visualização gráfica em formato de linha do tempo. Além disso, são constantemente visíveis o tempo restante para a próxima passagem, o satélite a ser rastreado, a elevação e azimute da antena, assim como a elevação e azimute alvos durante uma passagem.

4 MATERIAIS E MÉTODOS UTILIZADOS

4.1. METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

O ciclo de desenvolvimento consistiu na criação de um protótipo de interface gráfica refinado progressivamente para verificar se o modelo atendia aos requisitos. Chegando-se a um protótipo satisfatório, deu-se início à implementação funcional. Os módulos foram testados à medida que foram desenvolvidos, aperfeiçoando-os e corrigindo-se os *bugs*.

Para controle de versão, isto é, visualização de histórico de mudanças e possibilidade de reverter o código para suas versões anteriores, utilizou-se a ferramenta Git, também permitindo a distribuição de backups em servidores remotos.

Do software anteriormente escrito, aproveitaram-se os códigos referentes à movimentação da antena. A detecção de passagens e cálculo de rotas foram reescritos para facilitar a operação do software.

4.2. FRAMEWORK DE INTERFACE GRÁFICA

Decidiu-se usar o Qt entre os diversos *frameworks* de interface gráfica devido à maturidade, ou seja, tempo de existência da ferramenta e quantidade de atualizações, além de sua disponibilidade para gerar executáveis em múltiplas plataformas e sistemas operacionais, minimizando o custo de modificações caso seja necessário alterar o sistema operacional da máquina. O Qt também possui recursos para desenho 2D, persistência e redes, facilitando a criação de visualizações gráficas, configurações e conexão com a Internet.

Neste projeto, as rotinas de desenho foram utilizadas para criar uma linha do tempo visual com próximas passagens. As rotinas de persistência garantem que os dados de satélites rastreados não precisem ser reinseridos ao fechar e abrir novamente o programa. As rotinas de redes permitem que TLEs sejam atualizados automaticamente, calculando as posições de satélites com maior precisão.

Para escrever o código, compilar e encontrar *bugs*, foi utilizada a IDE Qt *Creator*. Para visualização de próximas passagens, desenvolveu-se um componente personalizado utilizando a classe *QGraphicsView*.

4.3. **BUG-TRACKING**

Para manter uma lista de *bugs* a serem corrigidos, foi criado um simples documento de texto, contendo também uma lista de tarefas em ordem de prioridade.

4.4. **DOCUMENTAÇÃO**

Para documentação de código do software, utilizou-se o formato Doxygen de comentários, o que permite que documentos HTML sejam gerados automaticamente descrevendo a estrutura do programa e a funcionalidade de cada classe, bem como seus métodos e variáveis. Os documentos servirão de referência para programadores entenderem a estrutura do projeto, facilitando futuras modificações. Foram inseridos comentários simples no código em partes de entendimento não trivial.

Além da documentação Doxygen para auxiliar programadores, será escrito um manual de referência para usuários descrevendo o modo de utilização do programa.

4.5. **BIBLIOTECAS**

Para calcular as rotas dos satélites cadastrados a partir de TLEs, foi feita uma pesquisa de bibliotecas que implementam o modelo de perturbações SGP4. Após a tentativa de compilação e testes com diversas bibliotecas para C++, foi selecionada a biblioteca desenvolvida por Warner (2014) por conter exemplos fornecidos que se adequam às necessidades do projeto. Foi necessário adaptar o código referente à obtenção de datas e horários para compilação no sistema operacional Windows. As trajetórias dos satélites calculadas foram comparadas com as anteriormente geradas pelo STK. Como o software anterior já havia sido testado com sucesso, esse

passo foi necessário para assegurar que os cálculos de passagens tinham a mesma precisão dos executados pelo STK. Também foi testada a biblioteca PyEphem, porém as rotas calculadas diferiam significativamente dos valores esperados.

4.6. **PROTOTIPAÇÃO**

Quando necessário, foi utilizado Python para escrita mais rápida de códigos. Como exemplo, para testar rapidamente a viabilidade de receber dados de satélites do CelesTrak, um *script* foi escrito para descobrir cabeçalhos de pacote necessários para uma solicitação ser aceita e respondida pelo servidor.

O protótipo da interface gráfica foi feito com ferramenta de edição *Forms* da IDE Qt Creator. Pretendia-se codificar a interface puramente por código após chegar a um modelo aceitável, porém foi possível construir o programa final sem comprometer o desempenho usando a interface criada com o editor.

4.7. **SIMULAÇÃO**

Para executar testes sem necessidade de manipulação da antena real, um modelo do sistema de controle de posição da antena foi codificado em Arduino, recebendo comandos no mesmo formato e enviando respostas de azimute e elevação simulando o comportamento da antena real.

5 **ANÁLISES E RESULTADOS**

O software já permite o cadastro de satélites a serem rastreados e manipulados em uma lista de prioridades básica e visualizações de próximas passagens tanto em modo tabela quanto modo gráfico, numa linha do tempo.

Na sexta-feira, 14/07/2017, foi realizado o primeiro teste de rastreamento de satélite com o novo software desenvolvido. O posicionamento da elevação apresentou pequenas oscilações indesejadas e a causa ainda não foi identificada. Apesar disso, a conexão com a antena funcionou e a antena

apontou com sucesso para um satélite cadastrado no sistema na hora programada.

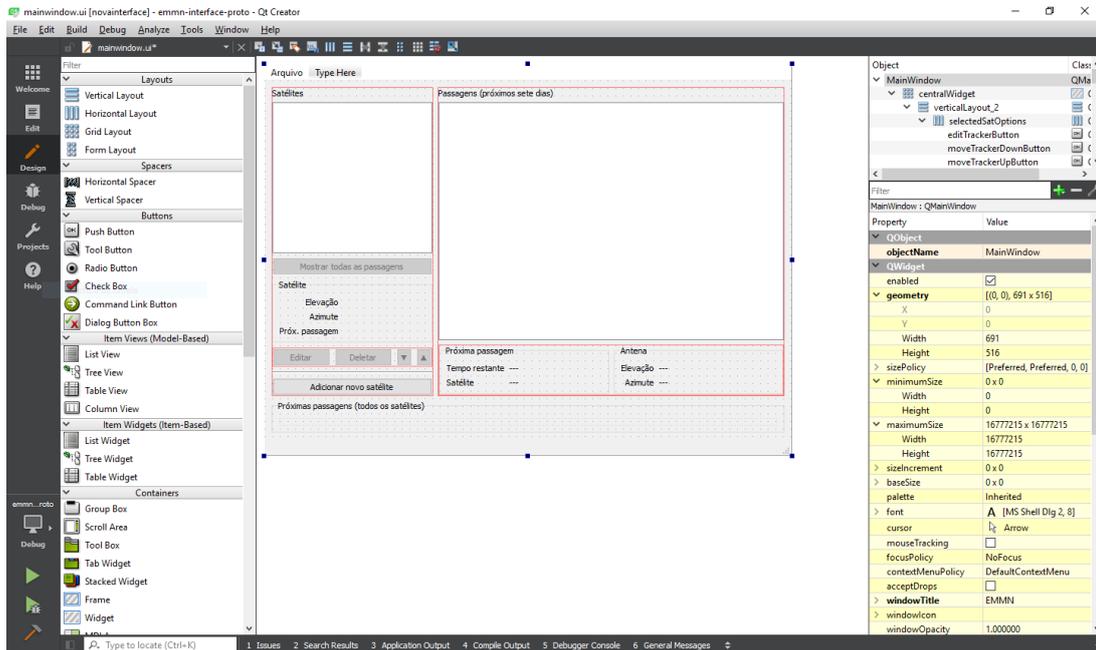
Foram identificados vazamentos de memória com o uso prolongado do programa que devem ser corrigidos em versões posteriores.

6 CONCLUSÕES

Foi possível desenvolver e adaptar o código sem grandes dificuldades, além de adicionar novos recursos. Um teste prático de integração foi realizado, rastreando um satélite com sucesso, porém com uma oscilação de posicionamento de elevação cuja causa ainda não foi identificada. É necessário resolver *bugs*, identificar e corrigir um caso de vazamento de memória e adicionar mais funcionalidades, como detecção de porta serial para realizar comunicação com Arduino e posicionamento manual da antena. Resta gerar a documentação através da ferramenta Doxygen, escrever um manual de operação para usuários e elaborar um artigo para congresso. Resta também realizar mais testes com a antena.

APÊNDICE A

Figura 1 – Editor de interfaces



Editor de interfaces (*Forms*) do Qt Creator.

Fonte: Produção do autor.

Figura 2 – Imagem do software anteriormente desenvolvido

```
Numero de passagens que ja ocorreram: 3 Total de passagens carregadas: 32
Informacoes da proxima passagem:
Data: 19/7/2017 Hora: 21:48:28 duracao: 889.820984 max_ele: 82.200996 Sat_name:
e: S_4_

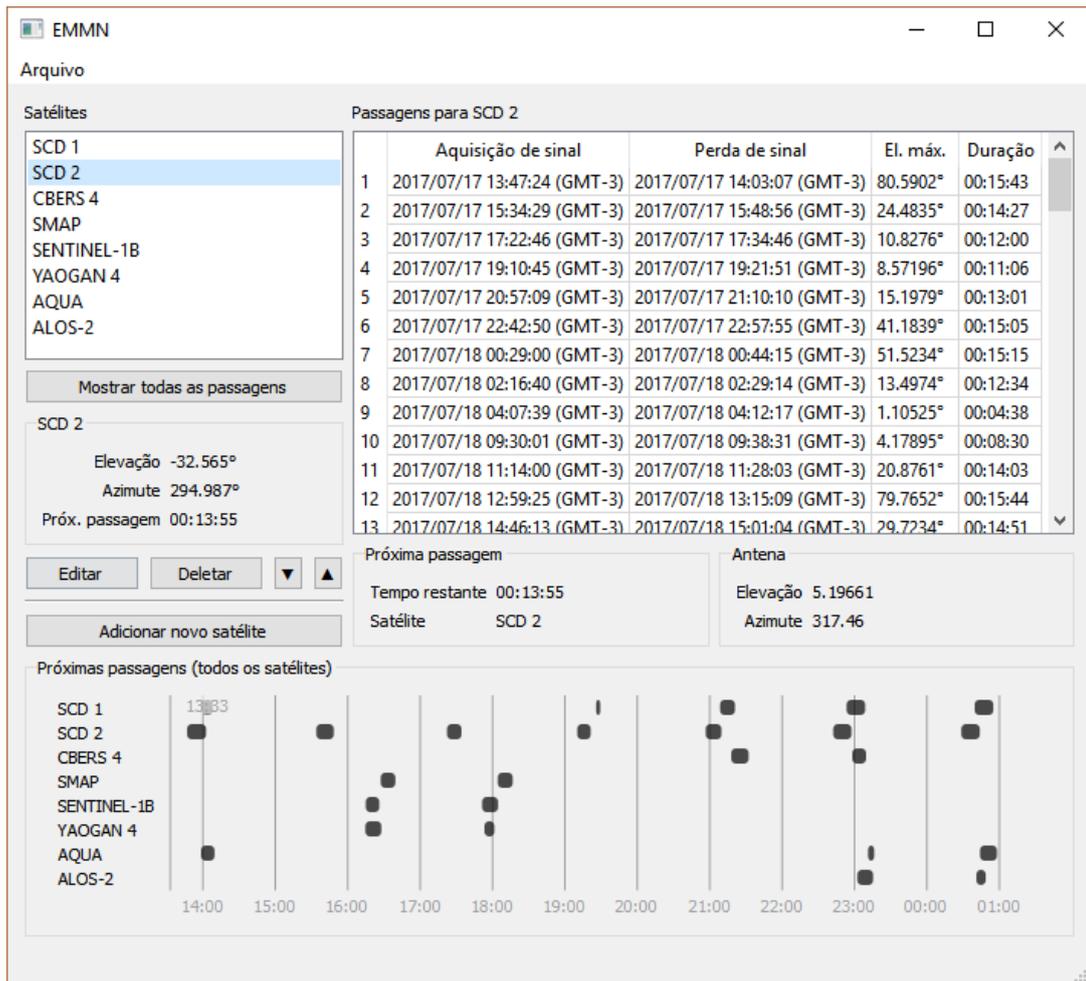
Tempo restante para a proxima passagem: 11:10:54
Input String: {87, 128, 0, 64, 0, 31, 136}
AZ_antena = 180.002747
ELE_antena = 90.001373

P:0 KA1: 1 KA2: 1 DAZ: 1 DEL: 1 M: 1 Erros: 0
Erros detectados: 0
```

Projeto antigo sendo executado no sistema operacional Windows, rastreando o satélite CBERS-4.

Fonte: Produção do autor.

Figura 3 – Imagem do novo software com interface gráfica



Projeto sendo executado no sistema operacional Windows, aguardando passagens.

Fonte: Produção do autor.

GLOSSÁRIO

framework - modelo base para desenvolvimento de um software

efeméride - posição de um satélite para um dado momento no tempo

biblioteca - conjunto de rotinas desenvolvidas para serem incorporadas em vários projetos.

script - programa escrito em linguagem de nível de abstração maior

REFERÊNCIAS BIBLIOGRÁFICAS

BOEHM, B. W. et al. Quantitative evaluation of software quality. **Proceedings of the 2nd international conference on Software engineering**, 1976. 592-605.

CRAIG RHODES, B. PyEphem. **PyEphem**, 2008. Disponível em: <<http://rhodesmill.org/pyephem/>>. Acesso em: 19 julho 2017.

KELSO, T. S. TLEs do NORAD. **CelesTrak**, 16 dezembro 2000. Disponível em: <<https://www.celestrak.com/NORAD/elements/>>. Acesso em: 19 julho 2017.

NIELSEN, J. Heuristic evaluation. **Usability inspection methods**, v. 17, 1994. 25-62.

WARNER, D. Biblioteca SGP4 para C++. **danrw.com**, 2014. Disponível em: <<https://www.danrw.com/sgp4/>>. Acesso em: 19 julho 2017.