



## **DESENVOLVIMENTO DE SOFTWARE EMBARCADO EM PLATAFORMAS CUBESAT**

### **RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA PIBIC/INPE - CNPq/MCT**

**Thiago Alberto de Souza** - Bolsista PIBIC/INPE – CNPq/MCT

**Sensores**

(TEC)/INPE – MCT

E-mail: thiago.souza@inpe.br

**Dr. Maria de Fátima Mattiello-Francisco** – Orientadora

**Coordenação de Gestão Tecnológica**

(TEC)/INPE – MCT

**Instituto Nacional de Pesquisas Espaciais**

INPE - MCT

E-mail: fatima.mattiello@inpe.br

**INPE**  
**São José dos Campos**  
**Julho de 2015**

**RELATÓRIO FINAL DE INICIAÇÃO CIENTÍFICA DO PROGRAMA:  
PIBIC/INPE – CNPq/MCT**

**PROJETO**

**DESENVOLVIMENTO DE SOFTWARE EMBARCADO EM  
PLATAFORMAS CUBESAT**

**Relatório elaborado por:**

**Thiago Alberto de Souza** – Bolsista PIBIC/INPE –  
CNPq/MCT

Acadêmico do Curso de Engenharia de Computação

E-mail: thiago.souza@inpe.br

**Dr. Maria de Fátima Mattiello-Francisco** – Orientadora  
**Coordenação de Gestão Tecnológica**  
(TEC)/INPE – MCT

E-mail: fatima.mattiello@inpe.br

**Me. Marcelo Essado** – Co-Orientador  
**EMSISTI - Sistemas Espaciais & Tecnologia**

E-mail: marcelo@emsisti.com.br

## DADOS DE IDENTIFICAÇÃO

**Projeto:**

**DESENVOLVIMENTO DE SOFTWARE EMBARCADO EM PLATAFORMAS CUBESAT**

**Bolsista:**

**Thiago Alberto de Souza**

Acadêmico do Curso de Engenharia Computação

ETEP - FACULDADES

**Orientadora:**

**Dra. Maria de Fátima Mattiello-Francisco**

Coordenação de Gestão Tecnológica - (TEC)/INPE – MCT

**Co-Orientador:**

**Me. Marcelo Essado**

Engenheiro de Sistemas

EMSISTI – MCT

**Colaboradores/ Acadêmicos:**

**Cristiano Strieder** – Me. Computação Aplicada – INPE

**Local de Trabalho/Execução do Projeto:**

Laboratório de Simulação – LabSim/INPE – São José dos Campos, SP.

Laboratório de Verificação e Validação – LabVVSIS;/INPE – São Jose dos Campos. SP



## **AGRADECIMENTOS**

Agradeço à minha Orientadora, Dra. Maria de Fátima Mattiello-Francisco e ao meu Co-Orientador Me. Marcelo Essado pela grande oportunidade e apoio prestados nos momentos de dificuldades encontrados no decorrer do trabalho desenvolvido, proporcionando um grande crescimento pessoal e profissional.

Meus sinceros agradecimentos: À colega do Laboratório Verificação & Validação de Sistemas Mariam Acuña e ao Cristiano Strieder do Laboratório de Simulação pelo apoio e disponibilização de tempo; ao Programa PIBIC/INPE – CNPq/MCT pelo recurso financeiro e ao Dr. Otávio Santos Cupertino Durão coordenador geral de Engenharias e Tecnologias Espaciais do Programa NanosatC-BR – Desenvolvimento de CubeSats.

## SUMÁRIO

SUMÁRIO.....	6
LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	8
RESUMO.....	9
1. INTRODUÇÃO.....	10
2. FUNDAMENTAÇÃO.....	11
2.1 - SATÉLITES.....	11
2.2 - PROGRAMA NANOSATC-BR, DESENVOLVIMENTO DE CUBESATS ...	11
2.2.1 - MISSÃO NANOSATC-BR2.....	13
2.3 - COMPUTADOR DE BORDO (OBC) DO NANOSATC-BR2.....	13
2.3.1 - FreeRTOS.....	13
2.3.2 - NANOMIND A712.....	13
2.4 - DESENVOLVIMENTO DE SISTEMAS.....	15
2.5 - UML.....	15
3. SIMULADOR DE GESTÃO DE BORDO.....	17
3.1 - Metodologia.....	17
3.2 - SIMULADOR DO SOFTWARE DE DATA HANDLING DO NANOSATC-BR2	20
4. CONSIDERAÇÕES FINAIS.....	23
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	24

## LISTA DE FIGURAS

Figura 1 - Logo do Programa Fonte: NanosatC-BR.....	12
Figura 2 - Diagrama de Blocos Nanomind A712 Fonte: Nanomind .....	14
Figura 3 - Diagrama de Classe .....	18
Figura 4 - Máquina de estados Fonte: Campnogara 2013.....	19
Figura 5 - Simulador em modo Nominal .....	21
Figura 6 - Tela de Carga Util.....	22

## **LISTA DE TABELAS**

Tabela 1 - Componentes do NanoMind A712.....	14
Tabela 2 - Modos de Operação e Tarefas (threads) do sistema.....	20



## **RESUMO**

Com o objetivo de prover uma ferramenta para simular a operação do computador de bordo e sua interação com os experimentos científicos e tecnológicos embarcados na família de nanosatélites que compõem o Programa NanosatC-BR, o presente trabalho apresenta a versão 2.0 do simulador desenvolvido cujo foco foi o primeiro nanossatélite científico brasileiro - NanosatC-BR1. Descreve-se os modos de operação do satélite modelados em máquina de estados e a solução arquitetural do simulador.

## 1. INTRODUÇÃO

O presente relatório de pesquisa descreve as atividades desenvolvidas no período de 02/2015 a 06/2015, com foco nas funcionalidades do Software de Data Handling (Gestão de Bordo) para o satélite NanosatC-BR2. O NanosatC-BR2 é uma missão com objetivo de desenvolvimento tecnológico e científico em parceria com Centro Regional Sul -CRS, Universidade Federal de Santa Maria, Secretaria de Ciência, Inovação e Desenvolvimento Tecnológico do Rio Grande do Sul, Santa Maria Design House, Agência Espacial Brasileira, Instituto Tecnológico de Aeronáutica, CTI Renato Archer, União Santamariense de Radioamadores e LABRE-SP Liga de Amadores Brasileiros de Radioemissão.

Consagrado pelo nome OBDH – On Board Data Handling, os sistemas computacionais de supervisão de bordo basicamente implementam as funções de “comando e controle” e “telemetria” a bordo de aeronaves espaciais. Estes têm por objetivo principal verificar a saúde dos demais subsistemas da aeronave e efetuar seus controles quando necessário. Por exemplo, isto é típico em missões de pequenos satélites onde se utiliza grande capacidade do processamento em bordo objetivando eliminar hardware que consome potência e/ ou reduzir custo de operações em solo. (MATTIELLO-FRANCISCO, 2003),

A clareza na especificação dos requisitos do comportamento esperado do software de gestão de bordo é elemento chave para o projeto do software. Assim, a necessidade de levantar as demandas operacionais dos diferentes experimentos que compõem a carga útil do NanosatC-BR2 motivou a concepção e desenvolvimento de um simulador operacional que também será extremamente útil para apoiar o processo de Verificação e Validação do NanosatC-BR2.

O projeto tem por objetivo a formação de competências na área de engenharia de software e no processo de desenvolvimento do software embarcado em missões críticas. Pretende-se com o projeto consolidar os requisitos de operação do NanosatC-BR2.

## 2. FUNDAMENTAÇÃO

Neste capítulo é apresentado o embasamento teórico deste projeto. Aqui descreveremos alguns conceitos e funcionalidades assim como as ferramentas utilizadas para o desenvolvimento da aplicação.

### 2.1 - SATÉLITES

Pode-se afirmar que um satélite é qualquer objeto que orbita ao redor de outro, esse denominado principal. Existem dois tipos de satélites:

a) Natural

Um satélite natural é um astro (ou corpo celeste) que não foi feito pelo homem ou outro ser vivo. Pode-se citar como exemplo a lua que é o único satélite natural da terra.

b) Artificial

O satélite artificial é um astro (ou corpo celeste) que orbita em volta de um planeta ou similar, e que foi desenvolvido pelo homem. Pode-se citar o Cbers satélite sino-brasileiro.

Os satélites artificiais podem ser catalogados ou agrupados segundo sua massa, como mostrado abaixo:

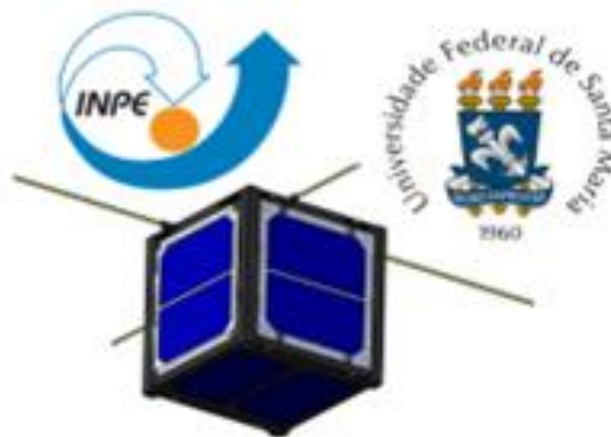
Tabela 1. Classificação satélites artificiais. Fonte INPE 2015

Classificação	Peso
Grandes satélites	Maior 1000 kg
Satélites médios	Entre 500 e 1000 kg
Minissatélites	Entre 100 e 500 kg
Microssatélites	Entre 10 e 100 kg
Nano satélites	Entre 1 e 10 kg
Pico satélite	Entre 0,1 e 1 kg
Femto satélite	Menor 100 g

### 2.2 - PROGRAMA NANOSATC-BR, DESENVOLVIMENTO DE CUBESATS

O Programa NANOSATC-BR, DESENVOLVIMENTO DE CUBESATS, visa alcançar alguns objetivos, tanto no âmbito acadêmico quanto no âmbito científico (NanosatC-BR), do projeto, lista-se alguns dos objetivos.

Figura 1: Logo do Programa Fonte: NanosatC-BR



Capacitação de Recursos Humanos, em nível de Graduação Universitária, para a realização de pesquisa e desenvolvimento com instrumentação espacial.

Capacitação tecnológica das instituições nacionais que participam da Missão, promovendo o desenvolvimento das áreas de ciências, engenharias e tecnologias espaciais;

Monitoramento do Geoespaço das condições geomagnéticas na superfície e em órbita sobre as regiões da Anomalia Magnética do Atlântico Sul – AMAS e do Eletro jato Ionosférico Equatorial sob a qual o Brasil se encontra;

Qualificar no espaço circuitos eletrônicos miniaturizados e Circuitos Integrados no âmbito do Projeto;

Pesquisas científicas relacionadas à fenomenologia do Geoespaço e Clima Espacial, tanto em aspectos globais gerais do espaço quanto na superfície sobre o Brasil;

### 2.2.1 - MISSÃO NANOSATC-BR2

A missão NanosatC-Br2 tem como objetivo o desenvolvimento tecnológico e científico:

- **Missão Científica**

Coletar dados do campo magnético terrestre, principalmente na região da Anomalia Magnética da América do Sul.

- **Missão Tecnológica**

Testar em voo circuitos integrados projetos no Brasil para resistência à radiação com um objetivo maior de futuramente serem utilizados em missões com outros satélites Brasileiros de maior porte.

### 2.3 - COMPUTADOR DE BORDO (OBC) DO NANOSATC-BR2

O Computador de Bordo (OBC - *On Board Computer*) do NanosatC-Br2 é o **NanoMind A712**, cujo o sistema operacional é o FreeRTOS, que fara interface dos aplicativos Data Handling e ADCS (*Attitude Determination and Control Subsystem*) com o hardware. O OBC implementa as funcionalidades de gestão de bordo (*software Data Handling*) dos subsistemas da plataforma do satélite e da carga útil que inclui funções de comando, manipulação de dados e aquisição de telemetria; e o ADCS que realiza a determinação e o controle de atitude.

#### 2.3.1 - FreeRTOS

É um Sistema Operacional de Tempo Real (RTOS – Real Time Operating System) para dispositivos de baixo consumo de potência e tem suporte para até 33 arquiteturas (FreeRTOS, 2015)

#### 2.3.2 - NANOMIND A712

O computador de bordo NanoMind A712 é desenvolvido pela empresa GomSpace ApS. A Tabela 1 apresenta lista os componentes que o compõe, diferenciando-os em itens opcionais e inclusos.

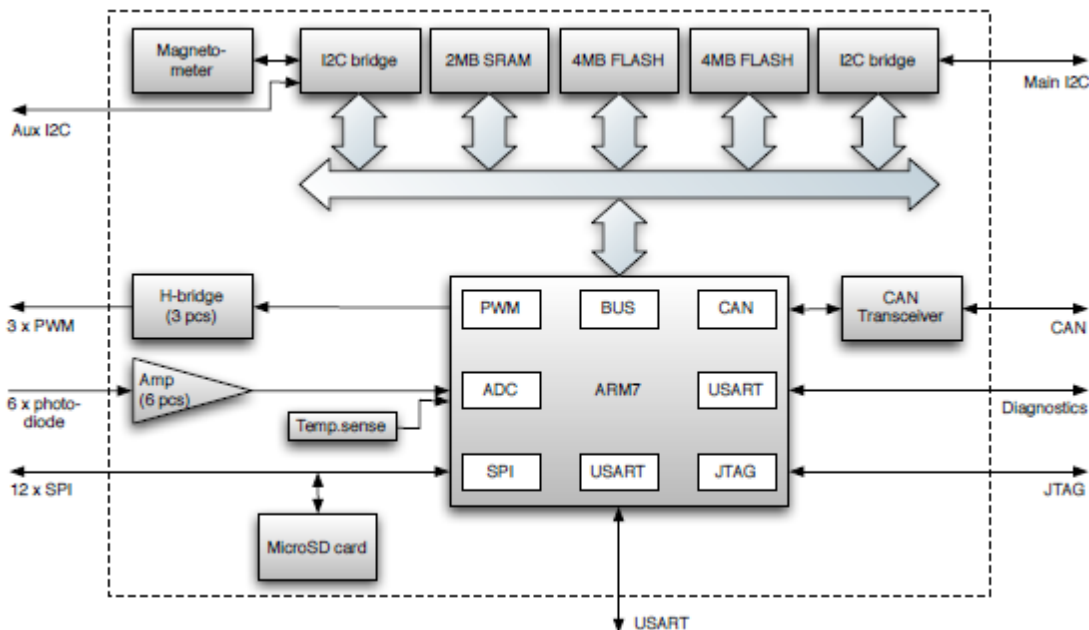
Tabela 1 - Componentes do NanoMind A712 Fonte: Nanomind.

Feature	A712C
ARM7 8-40MHz RISC CPU	•
2 MB SRAM	•
4MB parallel FLASH memory for code storage	•
4MB parallel FLASH memory for code and data storage	•
MicroSD card socket	•
I <sup>2</sup> C interface	•
CAN interface	○
Serial diagnostics interface with USB adapter	•
3 PWM outputs with bi-directional H-bridge drivers	•
6 analogue photo-diode amplifiers connected to AD-converters	•
SPI interface (for NanoPower Solar 100 panels with gyroscopes)	•
USART interface	•
2 on-board temperature sensors	•

- : Included feature
- : Optional feature (may be omitted)

A Figura 2 mostra o diagrama de blocos do sistema e as suas interfaces.

Figura 2 - Diagrama de Blocos Nanomind A712 Fonte: Nanomind



## 2.4 - DESENVOLVIMENTO DE SISTEMAS

Sem um bom planejamento e levantamento das informações, um sistema corre o risco de ser mal desenvolvido, não atendendo às necessidades reais. Na fase de levantamento de requisitos de software crítico são realizadas entrevistas com usuários, especialistas e analistas de negócios para determinar quais são os requisitos funcionais e de dependabilidade (desempenho, disponibilidade e segurança no funcionamento) e conhecer os processos que devem ser automatizados.

## 2.5 - UML

A UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) surgiu da união de três métodos de modelagem: o método de Booch, o método OMT de Jacobson e o método OOSE de Rumbaugh. Estes eram, até meados da década de 90, os três métodos de modelagem orientada a objetos mais populares entre os profissionais da área de desenvolvimento de software. A união dessas metodologias contou com o amplo apoio da Rational Software, que incentivou e financiou a união das três metodologias (MEDEIROS 2004).

A UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos.

O objetivo da UML é ajudar a definir as características do software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado.

Todas essas características são definidas por meio da UML antes do software começar a ser realmente desenvolvido (GUEDES 2004).

Alguns dos principais propósitos da UML são:

- **Especificação:** A UML pode ser utilizada para especificar "o que" é necessário para um sistema e "como" um sistema pode ser implementado. Ele captura os requisitos, análise, projeto e implementação de decisões muito importantes que precisam ser estabelecidos durante um ciclo de vida de desenvolvimento do sistema.

- **Visualização:** A natureza gráfica da UML permite a visualização de sistemas antes de serem implementadas. Usando as formas bem definidas para se comunicar com um público mais amplo de forma mais sucinta do que uma

narrativa descritiva e de forma mais abrangente do que o que muitas vezes pode ser representado por uma linguagem de programação.

- **Construção:** A UML pode ser usada para orientar e elaborar a implementação de um sistema complicado. Além disso, com a ajuda de várias ferramentas de caso no mercado, é possível gerar o código de forma orientado objeto a partir de modelos UML e também possível fazer a engenharia reversa e transformar um código pronto em uma UML.

- **Documentação:** A UML oferece um meio de capturar o conhecimento e documentação de resultados, tais como documentos de requisitos, especificações funcionais e planos de teste. Estes são todos críticos para controlar, medir e comunicar um sistema ao longo do seu ciclo de vida.

A UML é composta por diferentes tipos de diagrama, cada um representando o sistema sob uma determinada ótica. A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros.

Os artefatos UML foram adotados no projeto do simulador objeto do desenvolvimento desse trabalho.

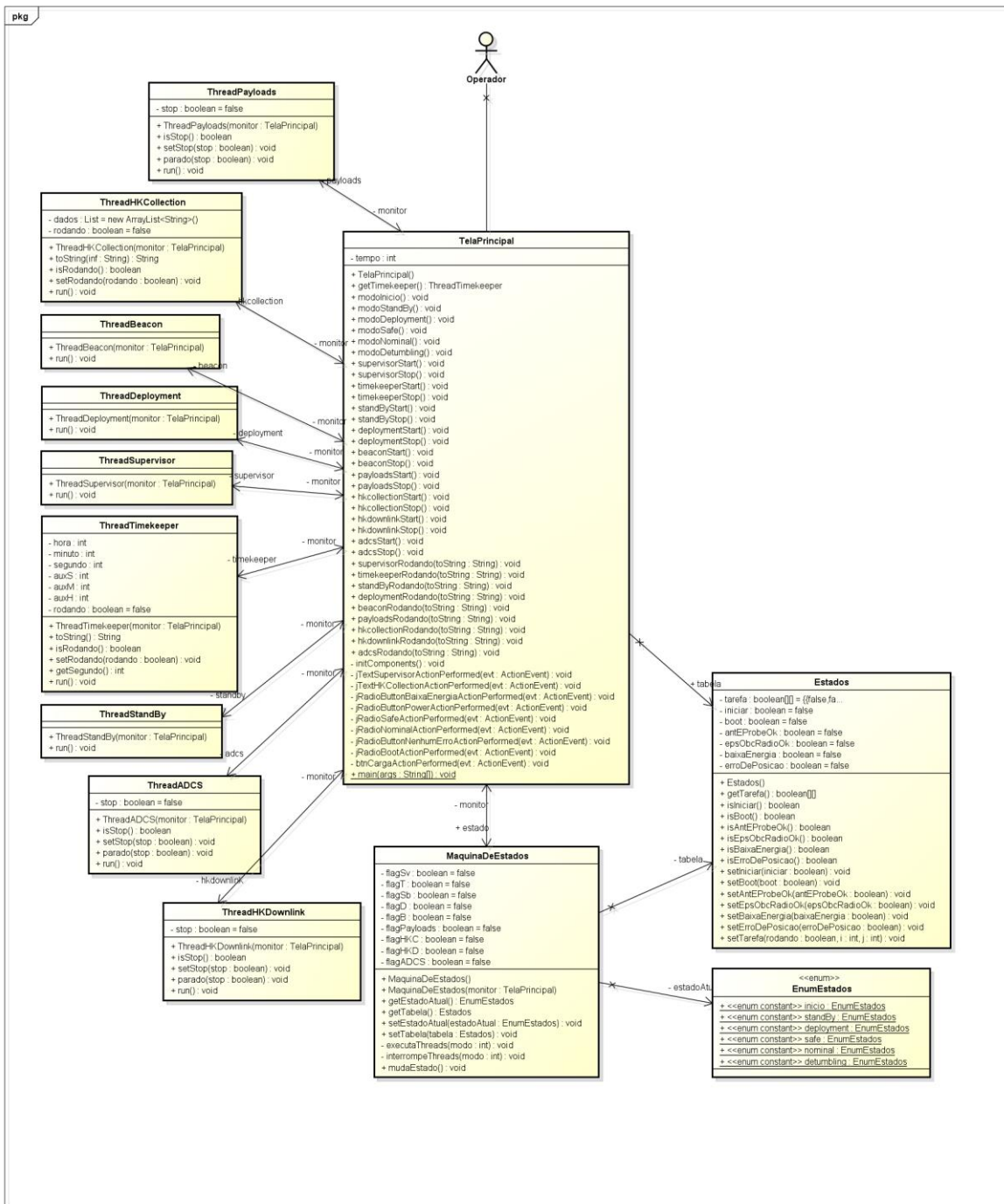


### **3. SIMULADOR DE GESTÃO DE BORDO**

#### **3.1 - METODOLOGIA**

A metodologia adotada no desenvolvimento do simulador considerou a experiência na operação do NanoSatC-B1 atualmente em órbita e um protótipo de simulador do protocolo I2C desenvolvido em projeto anterior de iniciação científica. Foi realizada uma engenharia reversa do programa para se conhecer todas as classes e seus respectivos relacionamentos. Com a realização do processo foi obtido o seguinte resultado apresentado na Figura 2.

Figura 3 - Diagrama de Classe

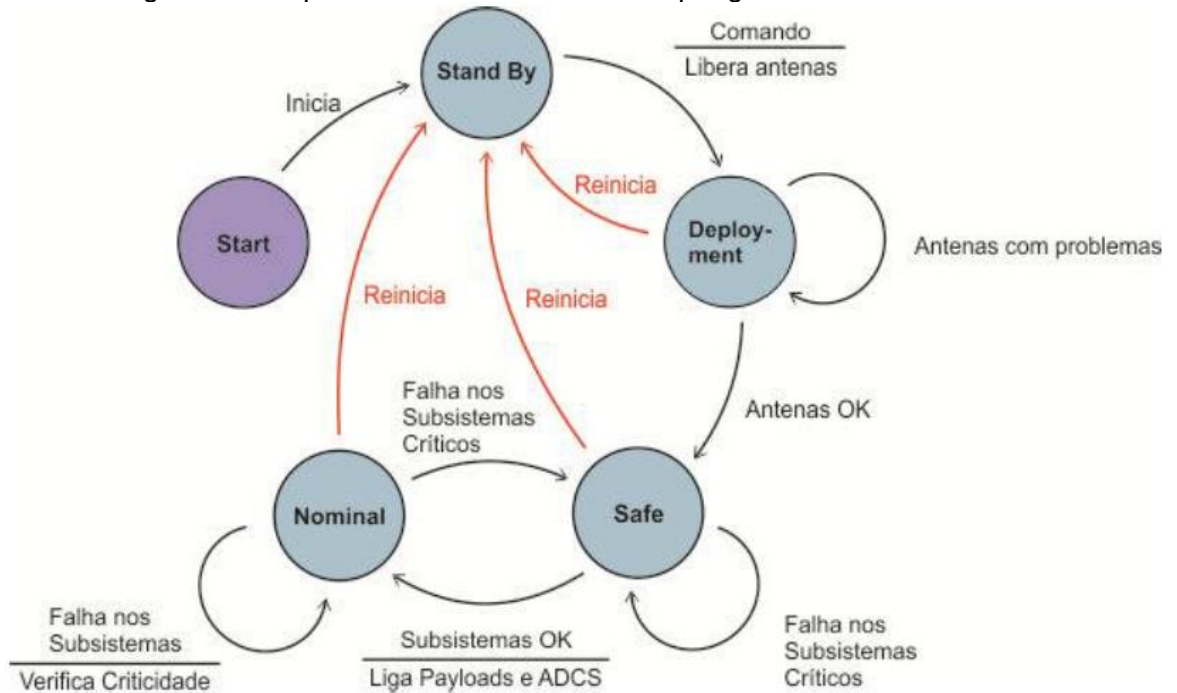


Com o diagrama de classe em mãos foi possível realizar uma análise mais sustentável de como o programa se comportava e assim possibilitando a remoção e implementação de alguns elementos.

A arquitetura de software apresentada na figura 2 foi estendida para implementar uma nova funcionalidade no simulador, que consiste na interação direta do operador com as cargas úteis. Essa interação permite ao operador escolher uma carga útil e simular sua operação remota: ativá-la, desativá-la e também escolher um telecomando para ser enviado e aguarda sua resposta.

O diagrama comportamental do Computador de Bordo apresentado na figura 3, concebido no escopo do projeto do simulador por Campnogara 2013, foi utilizado como referência.

Figura 4 - Máquina de estados Fonte: Campnogara 2013



Após análise da máquina de estados apresentada na Figura 3, identificou-se que o modo de operação que permite a interação com as cargas uteis é o Nominal.

Definido o estado de operação, foi dado início ao desenvolvimento das melhorias e as adições que seriam testadas.

### 3.2 - SIMULADOR DO SOFTWARE DE DATA HANDLING DO NANOSATC-BR2

Para um melhor entendimento do funcionamento do aplicativo de Gestão de Bordo, foi estudado e dado continuidade ao desenvolvimento do código Java que simula as tarefas de um OBDH associadas aos modos de operação, com base na arquitetura do NanosatC- Br1.

O simulador desenvolvido foi configurado para realizar os modos de operação do NanosatC-Br2, conforme apresentado na Tabela 2, onde as linhas representam os modos de operação do OBDH e as colunas as tarefas (Threads) em operação em cada modo. As células em verde indicam que a tarefa deve estar em operação no correspondente modo e as células em vermelho indicam que a tarefa deve estar desabilitada no correspondente modo.

Tabela 2 - Modos de Operação e Tarefas (threads) do sistema Fonte: Campnogara 2013

Modos de operação	SUPERVISO	TIMEKEEPER	BEACON	PAYLOAD	TRASCEPTOR		ADCS
					HKCOLLECTION	HKDOWNLINK	
STAND BY	Verde	Verde	Verde	Verde	Verde	Verde	Verde
DEPLOYMENT	Verde	Verde	Verde	Verde	Verde	Verde	Verde
SAFE	Verde	Verde	Verde	Verde	Verde	Verde	Verde
NOMINAL	Verde	Verde	Verde	Verde	Verde	Verde	Verde

Os modos de operação considerados são descritos a seguir:

**Standy By:** nesse modo o software de Data Handling já foi inicializado, as tarefas SUPERVISÃO, TIMEKEEPER e HKCOLLECTION encontram-se ativadas e aguarda o comando para liberação da antena;

**Deployment:** nesse modo as antenas estão abertas e o Sistema da Antena está sendo checado;

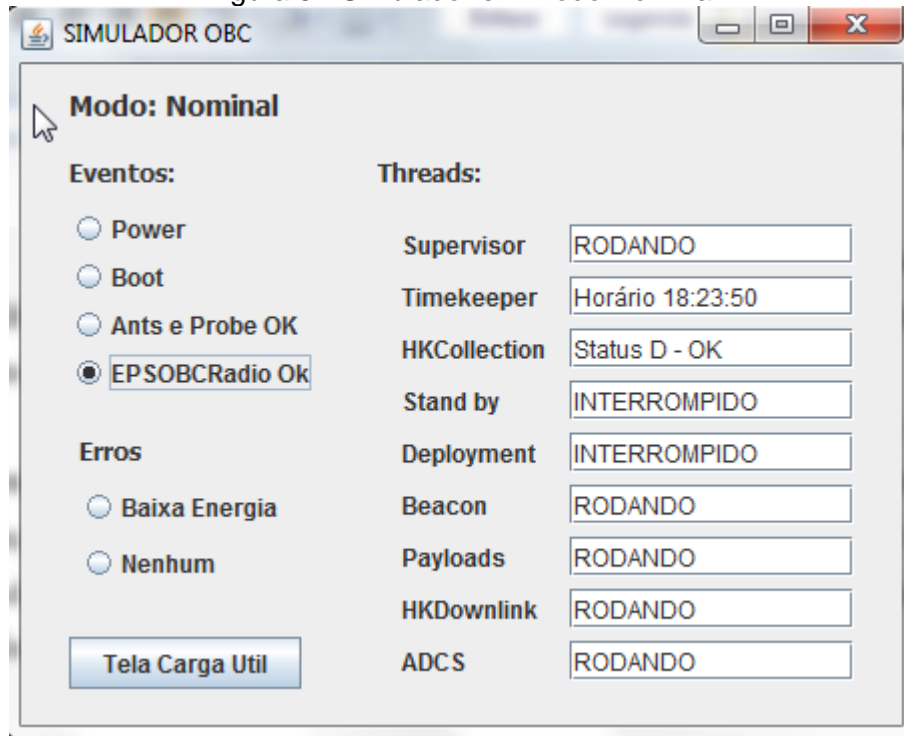
**Safe:** nesse modo o Sistema da Antena encontra-se validado e as tarefas SUPERVISÃO, TIMEKEEPER e HKCOLLECTION encontram-se ativadas;

**Nominal:** nesse modo os Sistemas de Potência e de Comunicação e o OBC encontram-se com sua integridade checada, o aplicativo de controle e atitude é iniciado e as tarefas SUPERVISÃO, TIMEKEEPER, HKCOLLECTION, HKDOWNLINK E ADCS encontram-se ativadas.

A Figura 4 mostra a primeira tela de interação com o usuário, onde o mesmo poderá interagir com o simulador seguindo uma ordem de execução que permite a

visualização abstrata do comportamento do OBDH associadas aos modos de operação.

Figura 5 - Simulador em modo Nominal



**Threads:** No simulador, não desempenham suas reais funções. Apenas são iniciadas ou não e retornam uma mensagem de sua operação de acordo com o modo em que o simulador se encontra.

- **Supervisor:** Ele aguarda ocorrências de eventos que resultam em mudança de estado.

- **Timekeeper:** Responsável pela manutenção do relógio do OBC.

- **HKCollection:** Sua função é coletar dados referentes a saúde dos demais subsistemas do satélite;

- **StandBy:** Apenas aponta quando o modo StandBy está ativo.

- **Beacon:** É responsável por enviar telemetrias em um espaço de tempo determinado.

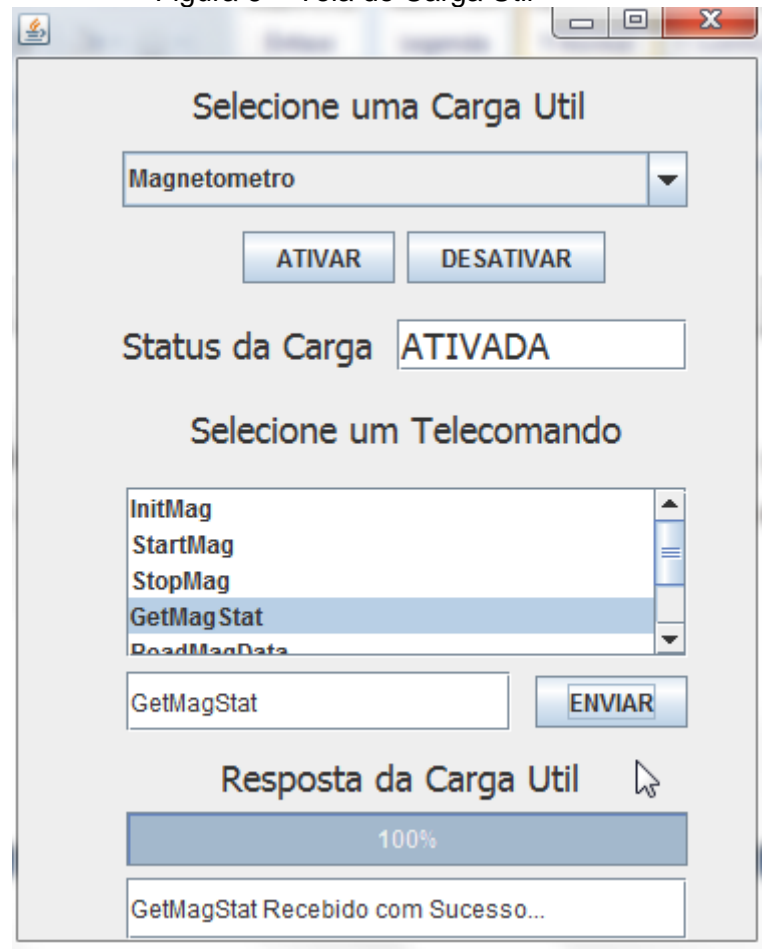
- **Payloads:** Coleta dados relativos as cargas úteis;

- **HKDownlink:** Transmite telemetria para a estação solo, durante o momento em que ela está visível pela Estação Solo.

- **ADCS:** Sistema de determinação e controle de atitude acionado ativo.

Em modo nominal o simulador habilita o botão Tela Carga Útil que dá acesso a uma segunda tela (Figura 5) que permite a interação do usuário com as cargas úteis.

Figura 6 - Tela de Carga Util



Esta tela do simulador tem por objetivo permitir ao usuário a seleção de uma carga útil e a partir dela selecionar um telecomando enviá-lo e aguardar sua resposta (telemetria). Da mesma forma que o simulador apenas representa a transição de estados respondendo a ações pré-determinadas, aqui se obtêm a mesma situação, todas as ações estão pré-determinadas no código fonte, onde o mesmo apenas realiza uma leitura externa de um arquivo .TXT em que estão definidos os telecomandos.

#### **4. CONSIDERAÇÕES FINAIS**

O relatório descreveu as atividades desenvolvidas pelo aluno no projeto de **“DESENVOLVIMENTO DE SOFTWARE EMBARCADO EM PLATAFORMAS CUBESAT”** no período de setembro de 2014 a julho de 2015.

Os estudos realizados referentes ao Software de Data Handling e ao Simulador de Data Handling foram apresentados neste Relatório. Os estudos realizados possibilitaram dar continuidade ao desenvolvimento do simulador com melhorias na documentação do projeto desenvolvido por Campnagara 2013 e estender suas funcionalidades buscando uma maior fidelidade do simulador com o NanoSatC-BR2

Atualmente o bolsista atua na elaboração de um documento de requisitos de um simulador genérico para missões de nanosatélites.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

CAMPONOGARA, Ândrei. **SOFTWARE EMBARCADO EM NANOSATÉLITES (SENa)**. Relatório Final de Projeto de Iniciação Científica PIBIC/INPE - CNPq/MCT. Jun. 2013. 42 f.

COSTA, L. L. **Projeto de um Aplicativo de Bordo para Missão NanosatC-Br**. Relatório Final de Projeto de Iniciação Científica PIBIC/INPE - CNPq/MCT. Jun. 2011. 64p.

GUEDES, G. T. A. **UML - Uma abordagem prática**. 1. Ed. São Paulo:Novatec, 2004. 319.

FREERTOS. **FreeRTOS**. Disponível em: <<http://www.freertos.org>>. Acesso em: 21 jun 2015.

GOMSPACE. **GOMX-Platform**. Disponível em: <<http://gomspace.com/index.php?p=products-platforms>>. Acesso em: 20 jun 2015.

INPE. **Nano Satélites**. Disponível em: <<http://www.crn2.inpe.br/conasat1/nanosatt.php>>. Acesso em: 20 jun 2015.

MATTIELLO-FRANCISCO, M. F.. **Sistemas Computacionais em Aplicações Espaciais**. INPE-9604-PUD/125. Fev. 2003.

MEDEIROS, E. S. **Desenvolvendo software com UML 2.0**. 1. ed. São Paulo: MAKRON Books, 2004. 264.

Nanomind, **Nanomind A712 DataSheet**. Disponível em:

<<http://gomspace.com/documents/GS-DS-NM712C-1.1.pdf>> Acesso em 22 jun 2015

NanosatC-BR, **Desenvolvimento de Cubesats**. Página Internet, online. Disponível em: <<http://www.inpe.br/crs/nanosat/NanoSatCBR2.php>> Acesso em: 15 jan. 2015



SCHUCH, N. J., et al. **Projeto NANOSATC-BR – Desenvolvimento de CubeSats**. Documento de Projeto. Centro Regional Sul de Pesquisas Espaciais. Santa Maria – RS, Junho 2010.