

Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA

## Projeto de P11

Autor: Arthur Henrique Vieira, Erick Tavares, Guilherme Resende, Hugo Queiroz, Ives Monteiro, João Victor da Silva, Kaed Canizo, Leonardo Henrique, Lucas Gama, Luis Eduardo Castro, Luiza Maluf, Millena de Abreu, Natan Almeida, Paulo Henrique Lamounier, Renan Vieira, Sophia Souza, Sophia Ferreira, Vitor Feijó

Orientador: Prof. Ricardo Ajax Dias Kosloski

Brasília, DF

2025



Arthur Henrique Vieira, Erick Tavares, Guilherme Resende, Hugo Queiroz,  
Ives Monteiro, João Victor da Silva, Kaed Canizo, Leonardo Henrique, Lucas  
Gama, Luis Eduardo Castro, Luiza Maluf, Millena de Abreu, Natan Almeida,  
Paulo Henrique Lamounier, Renan Vieira, Sophia Souza, Sophia Ferreira,  
Vitor Feijó

## **Projeto de PI1**

Trabalho submetido à disciplina de Projeto Integrador de Engenharia 1 da Universidade de Brasília.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Ricardo Ajax Dias Kosloski

Coorientador: Prof. Diogo Caetano Garcia, Prof. Juliana Petrocchi  
Rodrigues, Prof. Lui Txai Calvoso Habl e Prof. Rafael Rodrigues da Silva

Brasília, DF

2025

# Resumo

O projeto Controle de Trajetória de Foguetes d'Água foi desenvolvido no âmbito do Projeto Integrador de Engenharia 1 da Faculdade UnB Gama, com a colaboração de estudantes das engenharias de Software e Aeroespacial. O objetivo principal é construir um foguete reutilizável que utilize água como combustível e atinja distâncias fixas de 10, 20 e 30 metros com precisão de  $\pm 0,5$  metro, além de uma plataforma de lançamento automatizada que garanta segurança operacional.

O sistema incorpora um conjunto de sensores para medição em tempo real de parâmetros como pressão, ângulo de lançamento, velocidade e altitude, integrados a um microcontrolador ESP32 para processamento e transmissão de dados. A solução prevê ainda a persistência das informações em um banco de dados para calibração e análise de trajetórias. Entre os principais requisitos, destacam-se a automação eletromecânica do lançamento, a reutilização do foguete em três missões e o cumprimento de normas de segurança, como distanciamento mínimo de 5 metros de pessoas.

**Palavras-chaves:** Projeto Integrador de Engenharia 1, Faculdade UnB Gama, Foguete d'água, Controle de trajetória, Sistema embarcado.

# Lista de ilustrações

Figura 1 – EAP Geral . . . . .	15
Figura 2 – Pacote de Trabalho 1.1 – Planejamento e Gerenciamento . . . . .	16
Figura 3 – Pacote de Trabalho 1.1 – VPMN . . . . .	16
Figura 4 – Caso de Teste 01 - Importação de Arquivo JSON Válido . . . . .	17
Figura 5 – Caso de Teste 02 - Importação de JSON com campos ausentes . . . . .	17
Figura 6 – Caso de Teste 03 - Importação de formato inválido . . . . .	18
Figura 7 – Caso de Teste 05 - Exportação de Dados para CSV . . . . .	18
Figura 8 – Caso de Teste 06 - Responsividade da Interface . . . . .	19
Figura 9 – Caso de Teste 07 - Validação dos sensores no lançamento do foguete . . . . .	19
Figura 10 – Caso de Teste 08 - Funcionamento do mecanismo de lançamento . . . . .	20
Figura 11 – Caso de Teste 04 - Geração dos gráficos . . . . .	20
Figura 12 – Caso de Teste 24 - Integração entre ESP32 e Sensores . . . . .	21
Figura 13 – Caso de Teste 25 - Integração entre Foguete e Base (LoRa) . . . . .	21
Figura 14 – Caso de Teste 26 - Integração Energética – Estabilidade no Sistema Completo . . . . .	22
Figura 15 – Caso de Teste 09 - Parser de JSON Válido . . . . .	22
Figura 16 – Caso de Teste 10 - Parser de JSON com Campos Ausentes . . . . .	23
Figura 17 – Caso de Teste 27 - Parser de JSON com Dados Absurdos . . . . .	23
Figura 18 – Caso de Teste 11 - Aplicação de Filtro de Média Móvel . . . . .	24
Figura 19 – Caso de Teste 12 - Exportação de Dados para CSV . . . . .	24
Figura 20 – Caso de Teste 13 - Geração de gráfico . . . . .	25
Figura 21 – Caso de Teste 14 - Validação do Sensor MPX5700 (Pressão) . . . . .	25
Figura 22 – Caso de Teste 15 - Validação do Sensor MPU-6500 (Acelerômetro) . . . . .	26
Figura 23 – Caso de Teste 16 - Leitura do Sensor de Massa . . . . .	26
Figura 24 – Caso de Teste 17 - Validação do Dimensionamento Energético . . . . .	27
Figura 25 – Caso de Teste 18 - Validação do Atuador - Válvula Solenóide . . . . .	27
Figura 26 – Caso de Teste 19 - Validação do Atuador - Compressor . . . . .	28
Figura 27 – Caso de Teste 20 - Validação do Atuador - Buzzer . . . . .	28
Figura 28 – Caso de Teste 21 - Validação da Conectividade LoRa (RFM95W) . . . . .	29
Figura 29 – Caso de Teste 22 - Validação do Módulo de Armazenamento MicroSD . . . . .	29
Figura 30 – Caso de Teste 23 - Validação do Display LCD 16x4 . . . . .	30
Figura 31 – Pacote de Trabalho 1.2 – Estruturas . . . . .	30
Figura 32 – Pacote de Trabalho 1.3 – Software . . . . .	31
Figura 33 – Pacote de Trabalho 1.3 – Casos de Uso . . . . .	32
Figura 34 – Pacote de Trabalho 1.3 – Diagrama de Entidade-Relacionamento . . . . .	33
Figura 35 – Pacote de Trabalho 1.3 – Atributos das Entidades . . . . .	33

Figura 36 – Arquitetura do Software . . . . .	34
Figura 37 – Pacote de Trabalho 1.4 – Hardware . . . . .	35
Figura 38 – Pacote de Trabalho 1.5 – Energia . . . . .	35
Figura 39 – Pacote de Trabalho 1.6 – Integração . . . . .	36

# Lista de tabelas

Tabela 1 – Requisitos Gerais . . . . .	11
Tabela 2 – Requisitos de Software . . . . .	11
Tabela 3 – Requisitos de Hardware . . . . .	11
Tabela 4 – Requisitos de Custos e Materiais . . . . .	11
Tabela 5 – Composição da equipe. . . . .	14
Tabela 33 – Tabela de Requisitos do Projeto . . . . .	31
Tabela 34 – Requisitos Não-Funcionais . . . . .	32
Tabela 35 – Cronograma geral. . . . .	37
Tabela 36 – Cronograma de Atividades do Projeto (Parte 1) . . . . .	38
Tabela 37 – Cronograma de Atividades do Projeto (Parte 2) . . . . .	39

# Lista de abreviaturas e siglas

EAP	Estrutura Analítica do Projeto
TAP	Termo de Abertura do Projeto
SWOT	Forças ( <i>Strength</i> ), Fraquezas ( <i>Weakness</i> ), Oportunidades ( <i>Opportunities</i> ), Ameaças ( <i>Threats</i> )
PMBOK	<i>Project Management Body of Knowledge</i> (guia das melhores práticas do gerenciamento de projetos)
CAD	Ferramenta para criação de desenhos técnicos
DER	Diagrama de Entidade e Relacionamento
BACKLOG	Lista de atividades de execução do projeto
FCTE	Faculdade de Ciências e Tecnologias de Engenharia
KICKOFF	uma reunião com a equipe do projeto e o cliente do projeto
BPMN	Business Process Model and Notation
PET	Tipo de plástico usado para fazer garrafas
TAP	Termo de Abertura de Projetos
AEB	Agência Espacial Brasileira
ESP32	Placa de desenvolvimento de hardware aberto
CLI	Interface de Linha de Comando
API	Interface de Programação de Aplicações
GUI	Interface Gráfica do Usuário

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
<b>2</b>	<b>TERMO DE ABERTURA DO PROJETO</b>	<b>10</b>
2.1	Dados do projeto	10
2.2	Objetivos	10
2.3	Mercado-alvo	10
2.4	Requisitos	11
2.5	Justificativa	12
2.6	Indicadores	12
<b>3</b>	<b>EQUIPE DE TRABALHO</b>	<b>14</b>
<b>4</b>	<b>PROJETO CONCEITUAL DO PRODUTO</b>	<b>15</b>
<b>4.1</b>	<b>Características gerais</b>	<b>15</b>
4.1.1	Decomposição Inicial	15
4.1.2	Planejamento e Gerenciamento	15
<b>4.2</b>	<b>Casos de Teste</b>	<b>17</b>
4.2.1	Estruturas	30
4.2.2	Software	30
4.2.3	Hardware	35
4.2.4	Energia	35
4.2.5	Integração	36
<b>5</b>	<b>CRONOGRAMA DO PROJETO</b>	<b>37</b>
	<b>REFERÊNCIAS</b>	<b>40</b>



# 1 Introdução

A Universidade de Brasília (UnB), por meio da Faculdade de Ciências e Tecnologias em Engenharia (FCTE), promove uma formação interdisciplinar com foco em inovação tecnológica e desenvolvimento aplicado, integrando as engenharias Aeroespacial, Automotiva, de Energia, de Software e Eletrônica. No contexto do curso de Engenharia, a disciplina de Projeto Integrador 1 propõe a realização de projetos colaborativos, desafiando os estudantes a aplicarem conhecimentos técnicos, gerenciais e científicos na resolução de problemas reais. A ementa da disciplina compreende noções de projeto e gestão, modelos de ciclo de vida, gerenciamento de escopo, tempo, qualidade, recursos humanos e riscos, além da prática por meio de projetos interdisciplinares.

Entre as propostas de desafio prático, destaca-se o desenvolvimento de um sistema de lançamento de foguetes d'água com controle de trajetória. Essa atividade visa estimular competências de engenharia aplicada, prototipagem, controle e automação, além de considerar aspectos de segurança e reusabilidade. O objetivo do projeto é construir um foguete com propulsão hidrostática capaz de atingir distâncias predefinidas de 10m, 20m e 30m, com precisão igual ou inferior a  $\pm 0,5$  m, utilizando uma base de lançamento automatizada. A solução deve assegurar distanciamento mínimo de 5 metros de pessoas envolvidas e reaproveitamento do foguete em três ciclos de lançamento.

A utilização de propulsão à base de água (ou propulsão hidrostática) como método educacional tem sido recorrente em instituições de ensino devido ao seu baixo custo, segurança e potencial para experimentação científica. Diversos trabalhos acadêmicos relatam o uso de foguetes PET e sistemas automatizados em ambientes escolares e universitários. (YUKIMITSU, 2020), do Instituto Federal de Campinas, desenvolveram uma base lançadora automatizada para participação em olimpíadas científicas, empregando sensores e microcontroladores para controle de pressão e segurança do disparo, demonstrando a viabilidade de sistemas eletromecânicos embarcados nesse tipo de aplicação. De forma semelhante, Guimarães e Francisco (GUIMARÃES; FRANCISCO, 2020) propuseram uma plataforma automatizada com redução de torque para lançamento de foguetes PET, visando maior estabilidade e controle sobre o ângulo de disparo, destacando a importância da calibração e automação nos mecanismos de acionamento.

Do ponto de vista regulatório, embora foguetes experimentais de pequeno porte não sejam classificados como artefatos sujeitos à regulação pela Agência Espacial Brasileira (AEB), normas de segurança devem ser observadas. Segundo a National Association of Rocketry (National Association of Rocketry, 2012), recomenda-se uma distância mínima de cinco metros para lançamentos de foguetes de baixa potência. Tais diretrizes são

utilizadas como base para a definição dos parâmetros de segurança neste projeto.

O desenvolvimento de soluções próprias, em detrimento do uso de sistemas prontos, é estimulado no escopo da disciplina Projeto Integrador I, promovendo originalidade, inovação e domínio tecnológico. Neste contexto, a integração entre sensores de pressão, giroscópios, acelerômetros e módulos de comunicação sem fio, todos conectados a um microcontrolador como o ESP32, permite realizar medições em tempo real, persistir dados para análises futuras e otimizar a trajetória do foguete por meio de ajustes iterativos.

Dessa forma, o presente projeto justifica-se pela sua contribuição ao ensino prático de engenharia, pelo incentivo à construção de soluções seguras, reutilizáveis e tecnicamente viáveis, e pela proposta de um projeto que combina as engenharias de Software, Automotiva, Aeroespacial, Eletrônica e Energia. Além disso, atende a uma demanda acadêmica por experiências educacionais multidisciplinares que promovam competências técnicas e habilidades de trabalho em equipe.

Dessa forma, o projeto justifica-se pela necessidade de um sistema didático acessível para o ensino prático de engenharia, que integre eletrônica, software, estruturas e energia em uma solução reutilizável e segura. A proposta substitui sistemas prontos por uma base de lançamento desenvolvida do zero, com controle automatizado e coleta de dados em tempo real, permitindo experimentação e análise. Ademais, atende à demanda acadêmica por projetos interdisciplinares, promovendo competências técnicas e habilidades de trabalho em equipe.

## 2 Termo de Abertura do Projeto

### 2.1 Dados do projeto

**Nome do Projeto:** Foguete d'Água com Base Automatizada

**Data de abertura:** 23/04/2025

**Código:** 2-A

**Patrocinador:** Universidade de Brasília

**Gerente do projeto:** Vitor Feijó Leonardo

Matrícula: 221008516

E-mail: 221008516@aluno.unb.br

Telefone: +55 (61) 99243-6348

### 2.2 Objetivos

Desenvolver, até 16 de julho de 2025, um sistema original e funcional de lançamento automatizado de foguetes d'água, com controle de trajetória baseado em dados reais (pressão, ângulo, altitude, velocidade, aceleração), assegurando a reutilização em três lançamentos com precisão de até  $\pm 0,5$  metro nas distâncias de 10 m, 20 m e 30 m, respeitando o limite orçamentário de R\$ 1.000,00.

### 2.3 Mercado-alvo

O projeto tem como mercado-alvo principal instituições educacionais e grupos de pesquisa que atuam nas áreas de engenharia, física aplicada e educação científica, especialmente em níveis médio, técnico e superior. Entre os potenciais beneficiários estão professores e estudantes envolvidos em atividades práticas de ensino de ciências, clubes de astronomia e olimpíadas científicas, que podem utilizar o sistema como ferramenta didática para explorar conceitos de física, matemática, programação e controle de sistemas dinâmicos.

Além do ambiente acadêmico, o projeto atende também às necessidades de pesquisadores e entusiastas da engenharia experimental que necessitam de plataformas acessíveis para testes de controle de trajetória, aquisição de dados em tempo real e automação de

sistemas mecatrônicos. A proposta oferece um modelo de baixo custo e alta reusabilidade, adequado para experimentação segura em ambientes controlados.

De forma indireta, o sistema pode ainda inspirar iniciativas voltadas à popularização da ciência e inovação tecnológica, sendo compatível com projetos de extensão universitária e feiras científicas, contribuindo para a disseminação do conhecimento e estímulo à formação científica em contextos educacionais diversos.

## 2.4 Requisitos

Tabela 1 – Requisitos Gerais

Código	Descrição
RF1	O sistema deve executar, de forma totalmente automática, três lançamentos reutilizáveis de um mesmo foguete d'água, programados para alcançarem precisamente 10 m, 20 m e 30 m, com tolerância de $\pm 0,5$ m.
RF2	O software embarcado deve coletar, em tempo real, sinais de volume de água, pressão interna, ângulo de lançamento, posição e altitude, velocidade e aceleração (conforme especificações de precisão).
RF3	O firmware deve timestampar cada amostra com resolução e transmitir via Wi-Fi ou MicroSD para a unidade de processamento principal.
RF4	O módulo de processamento principal deve armazenar os dados em banco JSON, garantindo integridade transacional e histórico completo de cada voo.

Tabela 2 – Requisitos de Software

Código	Descrição
RQ01	Importar dados de voo em JSON de dispositivo externo
RQ02	Exibir gráfico de velocidade vertical vs. tempo.
RQ03	Exibir gráfico de aceleração vertical vs. tempo.
RQ04	Exibir gráfico de dispersão da trajetória no plano X e Y.
RQ05	Exibir gráfico de altitude vs. tempo.
RQ06	Exibir valores máximos e mínimos de aceleração, velocidade e ângulo.
RQ07	Exibir tempo de execução e intervalos de amostragem.
RQ08	Exportar dados do voo em CSV.
RQ09	Importar arquivos CSV para comparação e simulação.
RQ10	Aplicar filtro de média móvel nos dados dos sensores.

Tabela 3 – Requisitos de Hardware

Código	Descrição
RH1	Base de lançamento em alumínio ou aço inox, peso máximo de 2 kg, dimensões não superiores a $30 \times 30 \times 40$ cm.
RH2	Atuador de torque maior que 5 Nm para ajuste de ângulo, suportando 12 V DC.
RH3	Microcontrolador de 32 bits com ADC de 12 bits ou superior, memória RAM maior que 64 kB e Flash maior que 256 kB.
RH4	Fonte portátil (bateria LiPo 3S) capaz de alimentar todo o sistema por, ao menos, 4 h contínuas.

Tabela 4 – Requisitos de Custos e Materiais

Código	Descrição
RC1	Custo total de componentes eletrônicos, mecânicos e estruturais não deve exceder R\$ 1.000,00.
RC2	Materiais de construção (tubos, fixadores, conexões) devem ser reutilizáveis e modulares, facilitando montagem/desmontagem em menos de 1 h.

## 2.5 Justificativa

O projeto justifica-se pela necessidade de promover a aplicação prática e integrada de conhecimentos das diferentes engenharias da Faculdade de Ciências e Tecnologias em Engenharia (FCTE), por meio do desenvolvimento de uma solução original, segura e funcional para o lançamento automatizado de foguetes d'água.

Além disso, o projeto representa uma oportunidade para suprir a carência de sistemas didáticos acessíveis e reutilizáveis voltados ao ensino de física aplicada, automação e controle de trajetória em ambientes educacionais. Desse modo, integrando sensores e ferramentas de análise, a solução contribui para o avanço de práticas experimentais no ensino de engenharia e ciências, podendo ser adotada por instituições de ensino e centros de pesquisa como recurso pedagógico.

## 2.6 Indicadores

1. **Precisão da distância atingida:** diferença absoluta entre a distância real alcançada e as metas de 10m, 20m e 30m, com tolerância máxima de 0,5m.
2. **Número de lançamentos bem-sucedidos:** total de disparos realizados com o mesmo foguete sem falhas estruturais ou funcionais (meta: 3 lançamentos).
3. **Tempo de montagem e preparação para o lançamento:** tempo médio entre o início da montagem até a execução do disparo (em minutos).
4. **Número de dados coletados por voo:** total de amostras válidas registradas por sensores durante cada lançamento (em  $N$  leituras).
5. **Taxa de perda de dados:** percentual de dados esperados que não foram coletados ou foram invalidados por falha de sensor, ou transmissão.
6. **Tempo total de voo:** duração média entre o lançamento e o retorno do foguete ao solo, medido por sensores (em segundos).
7. **Desvio angular no lançamento:** diferença entre o ângulo configurado e o ângulo real de disparo detectado (em graus).
8. **Número de erros críticos no sistema embarcado:** quantidade de falhas que impedem a coleta, transmissão ou registro de dados em um lançamento.
9. **Número de funcionalidades do software implementadas com sucesso:** total de recursos entregues em relação ao planejado (ex.: gráficos, filtro de ruído, exportação de dados, visualização histórica, etc.).

10. **Atraso no cronograma:** dias de diferença entre o plano de entregas e a data real de finalização de cada etapa do projeto (em dias corridos).

### 3 Equipe de Trabalho

Tabela 5 – Composição da equipe.

Iniciais	Nome	Matrícula	Curso	Telefone	E-mail	Atribuições
AV	Arthur Vieira	23/1034064	Eng. de Software	(61) 98604-5333	231034064@aluno.unb.br	Engenheiro de Estruturas
EM	Erick Moreira	22/2022073	Eng. de Aeroespacial	(61) 99121-2563	222022073@aluno.unb.br	Engenheiro de estruturas
GC	Guilherme Carmona	22/1008070	Eng. de Aeroespacial	(61) 98128-8772	221008070@aluno.unb.br	Engenheiro de Estruturas
HM	Hugo Melo	23/1034064	Eng. de Software	(61) 99547-0505	231034064@aluno.unb.br	<b>Gerente</b> de <i>Software</i>
IL	Ives Lemos	22/1008122	Eng. de Aeroespacial	(61) 99966-5115	221008122@aluno.unb.br	<b>Gerente</b> de Estruturas
JF	João Farias	22/1022604	Eng. de Software	(61) 92002-3350	221022604@aluno.unb.br	Engenheiro de <i>Software</i>
KP	Kaed Prado	22/1008220	Eng. de Aeroespacial	(92) 98149-3839	221008220@aluno.unb.br	Engenheiro de estruturas
LJ	Leonardo Junior	23/1035428	Eng. de Software	(61) 98531-1586	231035428@aluno.unb.br	Engenheiro de Software
LB	Lucas Bottino	22/1008267	Eng. de Software	(61) 99149-8776	221008267@aluno.unb.br	Engenheiro de <i>Hardware</i>
LL	Luis Lima	22/1008285	Eng. de Software	(61) 99961-8010	221008285@aluno.unb.br	<b>Gerente</b> de Hardware
LA	Luiza Amorim	22/1008294	Eng. de Software	(61) 99845-4712	221008294@aluno.unb.br	Engenheiro de <i>Software</i>
MQ	Millena Queiroz	20/2046283	Eng. de Software	(61) 99250-1026	202046283@aluno.unb.br	Engenheiro de <i>Software</i>
NA	Natan Almeida	22/2006169	Eng. de Software	(61) 99841-7222	222006169@aluno.unb.br	Engenheiro de <i>software</i>
PD	Paulo Dantas	22/1022408	Eng. de Software	(61) 98156-3664	221022408@aluno.unb.br	Engenheiro de <i>Software</i>
RG	Renan Guedes	22/1031363	Eng. de Software	(61) 99542-2260	221031363@aluno.unb.br	Engenheiro de Hardware
SC	Sophia Cordeiro	22/1008454	Eng. de Aeroespacial	(18) 99783-6296	221008454@aluno.unb.br	Engenheiro de Estruturas
SS	Sophia Silva	23/1026886	Eng. de Software	(61) 98260-6252	231026886@aluno.unb.br	Engenheiro de <i>software</i>
VL	Vitor Leonardo	22/1008516	Eng. de Software	(61) 99243-6348	221008516@aluno.unb.br	<b>Gerente</b> do projeto

## 4 Projeto Conceitual do Produto

### 4.1 Características gerais

A Estrutura Analítica do Projeto (EAP), conforme definida pelo PMBOK Guide – Sexta Edição ([Project Management Institute, 2017](#)), consiste em uma decomposição hierárquica e orientada a entregas do escopo total do projeto. Sua função é organizar e subdividir o trabalho em partes menores e gerenciáveis, facilitando o planejamento, a execução eo controle das entregas do projeto.

A seguir, apresenta-se a EAP desenvolvida para o projeto *Foguete d'Água com Base Automatizada*, estruturada com base em seis pacotes principais de trabalho, que refletem os pilares técnicos e gerenciais do projeto.

#### 4.1.1 Decomposição Inicial

A decomposição do escopo do projeto resultou nos seguintes pacotes principais: Planejamento e Gerenciamento, Estruturas, Software, Hardware, Energia e Integração. A Figura 1 apresenta o primeiro e o segundo níveis da EAP.

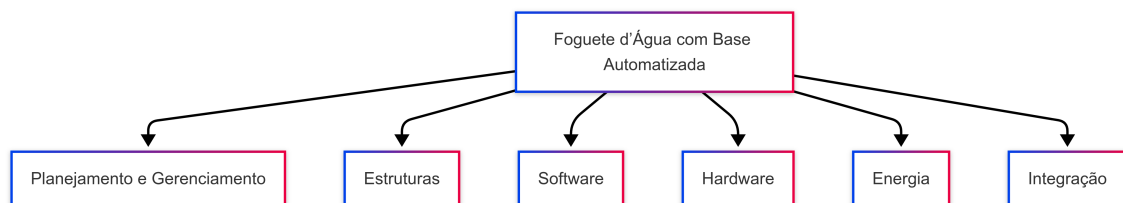


Figura 1 – EAP Geral

#### 4.1.2 Planejamento e Gerenciamento

Este pacote de trabalho contempla os processos de iniciação, planejamento e monitoramento do projeto. Abrange a elaboração do Termo de Abertura do Projeto (TAP), a consolidação da EAP e dos cronogramas setoriais, o planejamento orçamentário, os relatórios de acompanhamento (planejado x realizado) e as atividades de encerramento, como eventos de avaliação SWOT, lições aprendidas e avaliação de desempenho da equipe, conforme a Figura 2.





Figura 2 – Pacote de Trabalho 1.1 – Planejamento e Gerenciamento

Eventualmente, o pacote de trabalho de Planejamento e Gerenciamento foi estendido para o modelo de Fluxo de Valor do Produto (VPMN), conforme ilustrado na Figura 3. O VPMN é uma abordagem que visa otimizar o fluxo de valor ao longo do ciclo de vida do produto, garantindo que cada etapa agregue valor ao cliente e ao negócio.

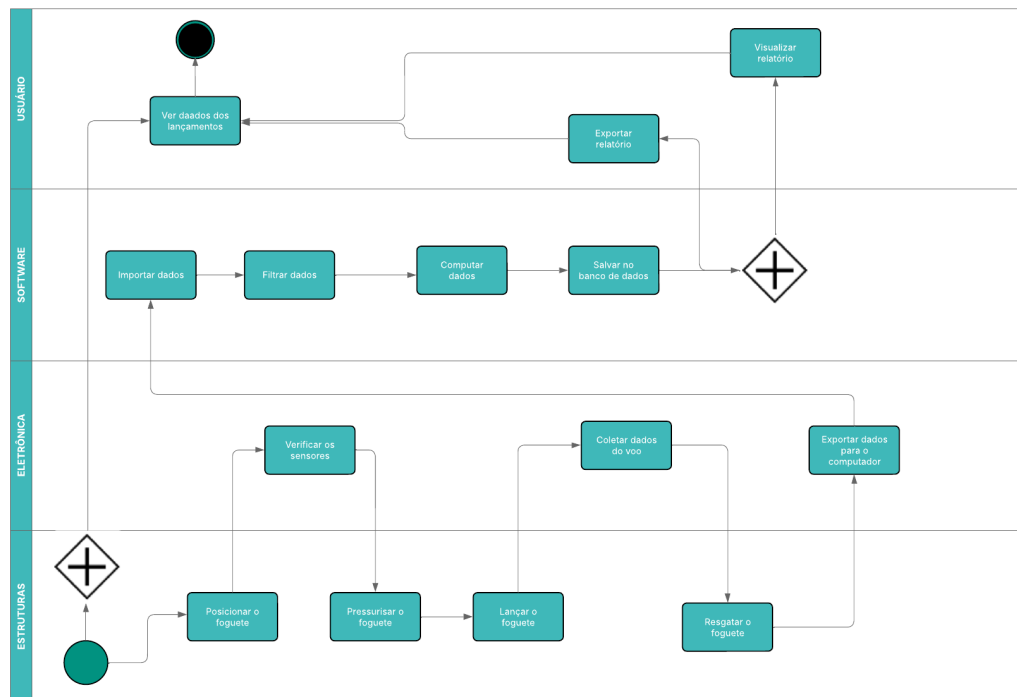


Figura 3 – Pacote de Trabalho 1.1 – VPMN

## 4.2 Casos de Teste

### Testes de Sistema

<b>CT01 - Importação de Arquivo JSON Válido</b>	
<b>Tipo</b>	Teste de sistema
<b>Objetivo</b>	Verificar se o sistema é capaz de importar corretamente um arquivo JSON com dados válidos.
<b>Pré-condições</b>	O sistema deve estar iniciado. Deve haver um arquivo "database.json" com dados completos e corretos no formato esperado.
<b>Procedimentos</b>	Abrir o sistema. Acessar a funcionalidade "Importar Dados". Selecionar o arquivo "database.json". Confirmar importação.
<b>Resultado esperado</b>	O sistema exibe mensagem de sucesso. Os dados são carregados na interface. Nenhum erro é exibido.
<b>Especificação do Reparo</b>	Verificar se o parser JSON está tratando corretamente os campos esperados. Validar se o caminho de acesso ao arquivo está correto. Corrigir o tratamento de erros silenciosos na importação.
<b>Resultado Após Reparo</b>	O sistema importa corretamente arquivos válidos e exibe os dados esperados. Teste é reexecutado e aprovado.

Figura 4 – Caso de Teste 01 - Importação de Arquivo JSON Válido

<b>CT02 - Importação de JSON com campos ausentes</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Verificar o comportamento do sistema ao importar registros incompletos.
<b>Pré-condições</b>	Arquivo "dados_incompletos.json" com campos obrigatórios ausentes.
<b>Procedimentos</b>	Acessar "Importar Dados". Selecionar "dados_incompletos.json". Confirmar importação.
<b>Resultado esperado</b>	Sistema identifica e ignora os registros inválidos.
<b>Especificação do Reparo</b>	Adicionar validação de campos obrigatórios durante o parsing.
<b>Resultado após Reparo</b>	Registros incompletos tratados corretamente.

Figura 5 – Caso de Teste 02 - Importação de JSON com campos ausentes

<b>CT03 - Importação de formato inválido</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Garantir que o sistema rejeita arquivos com extensão JSON mas conteúdo inválido (ex: CSV).
<b>Pré-condições</b>	Arquivo "dados_errados.json" contendo dados CSV.
<b>Procedimentos</b>	Tentar importar "dados_errados.json".
<b>Resultado esperado</b>	Mensagem de erro de formato exibida.
<b>Especificação do Reparo</b>	Validar estrutura JSON no momento da importação.
<b>Resultado após Reparo</b>	Sistema rejeita corretamente arquivos malformados.

Figura 6 – Caso de Teste 03 - Importação de formato inválido

<b>CT05 - Exportação de Dados para CSV</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Garantir que os dados processados sejam exportados corretamente para CSV.
<b>Pré-condições</b>	Dados processados e disponíveis na interface.
<b>Procedimentos</b>	Clicar em "Exportar Dados". Selecionar "Formato CSV". Salvar arquivo.
<b>Resultado esperado</b>	Arquivo CSV é gerado com os dados exibidos.
<b>Especificação do Reparo</b>	Corrigir função de exportação e formatação de colunas.
<b>Resultado após Reparo</b>	Arquivo CSV correto é gerado.

Figura 7 – Caso de Teste 05 - Exportação de Dados para CSV

<b>CT06 – Responsividade da Interface</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Verificar o comportamento da interface em diferentes resoluções da tela.
<b>Pré-condições</b>	Sistema em execução.
<b>Procedimentos</b>	Acessar a aplicação em diferentes tamanhos de tela. Navegar pelas funcionalidades.
<b>Resultado esperado</b>	Componentes se ajustam corretamente, sem sobreposições ou corte.
<b>Especificação do Reparo</b>	Ajustar responsividade.
<b>Resultado após Reparo</b>	Interface se adapta corretamente a todas as resoluções testadas.

Figura 8 – Caso de Teste 06 - Responsividade da Interface

<b>CT07 - Validação dos sensores no lançamento do foguete</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Verificar se os dados coletados pelos sensores (pressão, ângulo, massa) estão sendo corretamente lidos e registrados no início do lançamento.
<b>Pré-condições</b>	Sistema embarcado ligado. Foguete pronto para lançamento. Todos os sensores conectados corretamente.
<b>Procedimentos</b>	Ligar o sistema de aquisição de dados. Acionar o lançamento do foguete. Verificar os dados registrados pelos sensores na interface de análise.
<b>Resultado Esperado</b>	Os dados de pressão, ângulo e massa devem ser capturados corretamente, sem valores nulos ou inconsistentes, e armazenados no JSON gerado.
<b>Especificação do Reparo</b>	Verificar conexões físicas dos sensores, calibrar sensores com valores reais, corrigir erros de leitura no firmware.
<b>Resultado Após Reparo</b>	Os sensores coletam os dados de forma correta e os valores são exibidos e armazenados conforme esperado.

Figura 9 – Caso de Teste 07 - Validação dos sensores no lançamento do foguete

<b>CT08 - Funcionamento do mecanismo de lançamento</b>	
<b>Tipo</b>	Sistema
<b>Objetivo</b>	Verificar o funcionamento do mecanismo de lançamento.
<b>Pré-condições</b>	Sistema embarcado ligado. Foguete carregado com água e pressurizado.
<b>Procedimentos</b>	Acionar o sistema de lançamento via interface. Observar o funcionamento do atuador eletromecânico.
<b>Resultado Esperado</b>	O foguete deve ser lançado automaticamente após o comando, sem falhas mecânicas ou atraso.
<b>Especificação do Reparo</b>	Verificar fiação, tensão e código do acionamento do atuador.
<b>Resultado Após Reparo</b>	Atuador funciona corretamente e o foguete é lançado de forma imediata e estável.

Figura 10 – Caso de Teste 08 - Funcionamento do mecanismo de lançamento

## Testes de Integração

<b>CT04 - Geração dos gráficos</b>	
<b>Tipo</b>	Integração
<b>Objetivo</b>	Verificar se o sistema gera corretamente o gráfico de altitude após a importação dos dados.
<b>Pré-condições</b>	Dados válidos já importados.
<b>Procedimentos</b>	Acessar a interface de visualização de gráficos.
<b>Resultado Esperado</b>	Gráfico exibido com dados coerentes.
<b>Especificação do Reparo</b>	Ajustar lógica de plotagem ou eixos do gráfico.
<b>Resultado após Reparo</b>	Gráfico é exibido corretamente.

Figura 11 – Caso de Teste 04 - Geração dos gráficos

<b>CT24 - Integração entre ESP32 e Sensores</b>	
<b>Tipo</b>	Integração
<b>Objetivo</b>	Verificar se os sensores (pressão, ângulo, massa) estão integrados corretamente ao ESP32 e geram dados coerentes.
<b>Pré-condições</b>	Firmware embarcado finalizado. Todos os sensores ligados ao ESP32.
<b>Procedimentos</b>	Iniciar sistema embarcado. Coletar dados em tempo real dos sensores. Verificar se os dados são salvos e transmitidos corretamente.
<b>Resultado esperado</b>	Leituras coerentes e disponíveis para transmissão e armazenamento.
<b>Especificação do Reparo</b>	Verificar drivers de sensores, conexões físicas e timing de leitura.
<b>Resultado após Reparo</b>	Sistema lê e integra dados sem falhas ou atrasos.

Figura 12 – Caso de Teste 24 - Integração entre ESP32 e Sensores

<b>CT25 - Integração entre Foguete e Base (LoRa)</b>	
<b>Tipo</b>	Integração
<b>Objetivo</b>	Garantir que os dados gerados no voo são transmitidos e recebidos pela base sem perdas relevantes.
<b>Pré-condições</b>	Firmware de transmissão e recepção finalizado. Antenas e módulos LoRa funcionando.
<b>Procedimentos</b>	Iniciar voo de teste (ou simulação). Observar dados transmitidos em tempo real (base recebe JSON). Comparar dados recebidos com dados salvos no MicroSD.
<b>Resultado esperado</b>	Dados coerentes e transmissão confiável.
<b>Especificação do Reparo</b>	Ajustar parâmetros LoRa (velocidade, potência) e revisar firmware.
<b>Resultado após Reparo</b>	Comunicação estável e sem falhas críticas.

Figura 13 – Caso de Teste 25 - Integração entre Foguete e Base (LoRa)

<b>CT26 - Integração Energética – Estabilidade no Sistema Completo</b>	
<b>Tipo</b>	Integração
<b>Objetivo</b>	Verificar se a fonte de energia dimensionada suporta o consumo de todos os componentes ao mesmo tempo.
<b>Pré-condições</b>	Sistema montado com todos os sensores e atuadores ativos. Fonte de energia de 3,3 V e corrente maior que 300 mA.
<b>Procedimentos</b>	Ligar todos os módulos simultaneamente (ex: pressão, atuadores, display, etc.). Observar estabilidade de tensão e corrente durante 30 segundos. Verificar se não ocorrem quedas ou falhas.
<b>Resultado esperado</b>	Tensão estável ( $\pm 6\%$ ) e corrente $\leq$ capacidade máxima (300)mA.
<b>Especificação do Reparo</b>	Substituir a fonte por uma de maior capacidade ou revisar fiação.
<b>Resultado após Reparo</b>	Sistema opera sem oscilações ou falhas de alimentação.

Figura 14 – Caso de Teste 26 - Integração Energética – Estabilidade no Sistema Completo

## Testes de Unidade

<b>CT09 - Parser de JSON Válido</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar leitura de JSON com dados completos.
<b>Pré-condições</b>	Função de parser implementada e disponível no ambiente de desenvolvimento. Arquivo de teste válido com dados completos.
<b>Procedimentos</b>	Rodar função de parser diretamente com arquivo de teste válido. Verificar se estrutura retornada está correta.
<b>Resultado esperado</b>	Dados carregados na memória, sem exceções.
<b>Especificação do Reparo</b>	Revisar lógica de leitura e formatação do JSON. Corrigir tratamento de exceções ou formatação incorreta.
<b>Resultado após Reparo</b>	Parser reconhece corretamente arquivos válidos e gera a estrutura esperada sem exceções.

Figura 15 – Caso de Teste 09 - Parser de JSON Válido

<b>CT10 - Parser de JSON com Campos Ausentes</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Garantir tratamento de dados incompletos.
<b>Pré-condições</b>	Função de parser em funcionamento. JSON de teste com campos ausentes.
<b>Procedimentos</b>	Rodar parser com JSON que tenha campos ausentes. Verificar se registros inválidos são removidos.
<b>Resultado esperado</b>	Retorno apenas dos registros completos.
<b>Especificação do Reparo</b>	Corrigir verificação de campos obrigatórios. Implementar descartes controlados de registros inválidos.
<b>Resultado após Reparo</b>	Parser filtra registros incompletos sem falhas.

Figura 16 – Caso de Teste 10 - Parser de JSON com Campos Ausentes

<b>CT27 - Parser de JSON com Dados Absurdos</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar descarte de dados absurdos (ex: altitude negativa).
<b>Pré-condições</b>	Função de validação em funcionamento. Lista de dados brutos com registros absurdos.
<b>Procedimentos</b>	Rodar função de validação com lista de dados brutos. Observar se registros absurdos são filtrados.
<b>Resultado esperado</b>	Lista resultante sem registros absurdos.
<b>Especificação do Reparo</b>	Corrigir regra de validação de campos. Implementar descarte de dados absurdos.
<b>Resultado após Reparo</b>	Função remove corretamente dados absurdos sem impactar o restante.

Figura 17 – Caso de Teste 27 - Parser de JSON com Dados Absurdos



<b>CT11 - Aplicação de Filtro de Média Móvel</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar aplicação correta de filtro.
<b>Pré-condições</b>	Função de filtro implementada. Dados de teste com ruído disponíveis.
<b>Procedimentos</b>	Rodar função de filtro com dados de teste (com ruído). Comparar saída com valores esperados (média móvel conhecida).
<b>Resultado esperado</b>	Dados suavizados sem distorção indevida.
<b>Especificação do Reparo</b>	Ajustar janela ou lógica de média móvel. Corrigir cálculos que gerem resultados errados.
<b>Resultado após Reparo</b>	Filtro suaviza corretamente os dados e remove ruídos.

Figura 18 – Caso de Teste 11 - Aplicação de Filtro de Média Móvel

<b>CT12 - Exportação de Dados para CSV</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Validar formatação e consistência do CSV.
<b>Pré-condições</b>	Função de exportação implementada e em funcionamento. Dados em memória para exportação.
<b>Procedimentos</b>	Rodar função de exportação com dados em memória. Abrir CSV e verificar cabeçalhos, formato e integridade dos dados.
<b>Resultado esperado</b>	Arquivo CSV gerado sem erros.
<b>Especificação do Reparo</b>	Corrigir escrita do arquivo CSV (formato e separadores). Ajustar cabeçalhos e ordem de dados.
<b>Resultado após Reparo</b>	CSV exportado corretamente e sem inconsistências.

Figura 19 – Caso de Teste 12 - Exportação de Dados para CSV

<b>CT13 - Geração de gráfico</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Validar coerência visual dos gráficos gerados.
<b>Pré-condições</b>	Função de geração de gráficos implementada. Dados de teste consistentes.
<b>Procedimentos</b>	Rodar função de geração de gráficos com dados de teste. Conferir se gráfico está correto visualmente e numericamente.
<b>Resultado esperado</b>	Gráfico gerado com dados coerentes e layout adequado.
<b>Especificação do Reparo</b>	Ajustar lógica de plotagem (eixos e escalas). Corrigir erros de indexação de dados.
<b>Resultado após Reparo</b>	Gráficos exibem dados de forma coerente e confiável.

Figura 20 – Caso de Teste 13 - Geração de gráfico

<b>CT14 - Validação do Sensor MPX5700 (Pressão)</b>	
<b>Tipo</b>	Unidade (hardware/embarcado)
<b>Objetivo</b>	Garantir leitura coerente da pressão durante carregamento.
<b>Pré-condições</b>	Sistema embarcado ligado e calibrado.
<b>Procedimentos</b>	Pressurizar gradualmente a câmara. Ler valores do sensor de pressão. Comparar com medidor externo confiável.
<b>Resultado esperado</b>	Leituras consistentes, sem desvios bruscos.
<b>Especificação do Reparo</b>	Calibrar sensor ou corrigir conversão no firmware.
<b>Resultado após Reparo</b>	Leituras de pressão confiáveis e coerentes.

Figura 21 – Caso de Teste 14 - Validação do Sensor MPX5700 (Pressão)

<b>CT15 - Validação do Sensor MPU-6500 (Acelerômetro)</b>	
<b>Tipo</b>	Unidade (hardware/embarcado)
<b>Objetivo</b>	Verificar se a leitura de aceleração está coerente com o movimento.
<b>Pré-condições</b>	Sistema embarcado funcional.
<b>Procedimentos</b>	Simular movimentos suaves e bruscos do foguete. Observar leituras de aceleração.
<b>Resultado esperado</b>	Dados refletem as variações reais de movimento.
<b>Especificação do Reparo</b>	Calibrar sensor e revisar código de leitura.
<b>Resultado após Reparo</b>	Sensor reporta acelerações reais de forma confiável.

Figura 22 – Caso de Teste 15 - Validação do Sensor MPU-6500 (Acelerômetro)

<b>CT16 - Leitura do Sensor de Massa</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Garantir leitura correta da massa do foguete antes do lançamento.
<b>Pré-condições</b>	Sensor de massa calibrado. Massa real conhecida.
<b>Procedimentos</b>	Colocar o foguete com massa conhecida na plataforma. Ler os dados da célula de carga via firmware. Comparar leitura com massa real medida em balança.
<b>Resultado esperado</b>	Erro máximo aceitável ( $\pm 10g$ ).
<b>Especificação do Reparo</b>	Verificar calibração e conexão do sensor. Corrigir leituras incorretas no firmware.
<b>Resultado após Reparo</b>	Leitura de massa consistente com o valor real.

Figura 23 – Caso de Teste 16 - Leitura do Sensor de Massa

<b>CT17 - Validação do Dimensionamento Energético</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar se o fornecimento de energia atende ao consumo calculado com margem de segurança.
<b>Pré-condições</b>	Sistema embarcado montado com todos os sensores e atuadores conectados. Fonte de alimentação dimensionada.
<b>Procedimentos</b>	Ligar o sistema completo. Monitorar tensão e corrente durante 30 segundos. Verificar se a tensão permanece estável. Verificar se a corrente total está dentro da faixa de consumo.
<b>Resultado esperado</b>	Sistema opera sem quedas de tensão ou falhas de alimentação.
<b>Especificação do Reparo</b>	Substituir fonte de alimentação ou ajustar conexões elétricas.
<b>Resultado após Reparo</b>	Sistema estável durante todo o tempo de operação.

Figura 24 – Caso de Teste 17 - Validação do Dimensionamento Energético

<b>CT18 - Validação do Atuador - Válvula Solenóide</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar se a válvula solenóide abre/fecha corretamente sob comando.
<b>Pré-condições</b>	Sistema embarcado funcional. Compressor e pressão dentro do esperado.
<b>Procedimentos</b>	Acionar a válvula via relé (comando direto no firmware). Observar se há liberação/fechamento imediato de ar.
<b>Resultado esperado</b>	Válvula responde rapidamente ao comando sem falhas.
<b>Especificação do Reparo</b>	Verificar conexões elétricas e integridade do relé.
<b>Resultado após Reparo</b>	Válvula abre/fecha conforme esperado.

Figura 25 – Caso de Teste 18 - Validação do Atuador - Válvula Solenóide

<b>CT19 - Validação do Atuador - Compressor</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar funcionamento do compressor 12 V durante a fase de pressurização.
<b>Pré-condições</b>	Sistema montado e conectado ao relé.
<b>Procedimentos</b>	Enviar comando para ativar compressor. Observar operação estável e sem superaquecimento.
<b>Resultado esperado</b>	Compressor enche a câmara e atinge pressão esperada.
<b>Especificação do Reparo</b>	Verificar tensão de alimentação, relé e conexões físicas.
<b>Resultado após Reparo</b>	Compressor opera normalmente sem falhas.

Figura 26 – Caso de Teste 19 - Validação do Atuador - Compressor

<b>CT20 - Validação do Atuador - Buzzer</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar se o buzzer emite contagem regressiva sonora.
<b>Pré-condições</b>	Firmware com contagem regressiva implementada.
<b>Procedimentos</b>	Acionar comando de contagem regressiva. Observar sequência de sons de 10 a 0.
<b>Resultado esperado</b>	Sons nítidos e no tempo correto.
<b>Especificação do Reparo</b>	Verificar ligação do buzzer e ajuste do firmware.
<b>Resultado após Reparo</b>	Contagem sonora clara e funcional.

Figura 27 – Caso de Teste 20 - Validação do Atuador - Buzzer

<b>CT21 - Validação da Conectividade LoRa (RFM95W)</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Garantir comunicação confiável entre ESP32 da base e do foguete.
<b>Pré-condições</b>	Ambientes de teste prontos (foguete e base).
<b>Procedimentos</b>	Ligar ambos os dispositivos. Iniciar transmissão de dados de teste do foguete para a base. Observar recepção sem perdas significativas.
<b>Resultado esperado</b>	Dados JSON recebidos sem interrupções relevantes.
<b>Especificação do Reparo</b>	Ajustar parâmetros de transmissão ou antenas.
<b>Resultado após Reparo</b>	Comunicação estável e confiável via LoRa.

Figura 28 – Caso de Teste 21 - Validação da Conectividade LoRa (RFM95W)

<b>CT22 - Validação do Módulo de Armazenamento MicroSD</b>	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Garantir que os dados de voo são salvos corretamente no cartão.
<b>Pré-condições</b>	Sistema do foguete funcional e MicroSD inserido.
<b>Procedimentos</b>	Iniciar simulação de voo. Salvar dados de sensores no MicroSD. Remover cartão e abrir no computador.
<b>Resultado esperado</b>	Arquivo JSON salvo e legível.
<b>Especificação do Reparo</b>	Verificar comandos de escrita no firmware e integridade do MicroSD.
<b>Resultado após Reparo</b>	Dados salvos corretamente e legíveis.

Figura 29 – Caso de Teste 22 - Validação do Módulo de Armazenamento MicroSD

CT23 - Validação do Display LCD 16x4	
<b>Tipo</b>	Unidade
<b>Objetivo</b>	Verificar exibição correta da contagem visual de 10 a 0.
<b>Pré-condições</b>	Display conectado e inicializado.
<b>Procedimentos</b>	Acionar contagem regressiva no firmware. Observar exibição da contagem no display.
<b>Resultado esperado</b>	Dígitos exibidos sem falhas ou cortes.
<b>Especificação do Reparo</b>	Ajustar comunicação I2C e revisão do código.
<b>Resultado após Reparo</b>	Contagem exibida de forma clara e precisa.

Figura 30 – Caso de Teste 23 - Validação do Display LCD 16x4

#### 4.2.1 Estruturas

Responsável pela modelagem e construção da fuselagem do foguete e da base física de lançamento. Este pacote inclui atividades como elaboração de desenhos técnicos em CAD, levantamento de materiais, montagem estrutural, e realização de experimentos e testes de integração estrutural. Também contempla a avaliação do desempenho de fornecedores, conforme ilustrado na Figura 31.



Figura 31 – Pacote de Trabalho 1.2 – Estruturas

#### 4.2.2 Software

Este pacote compreende a elicitação de requisitos funcionais e não funcionais, a modelagem da arquitetura do sistema, a construção e testes de software. Inclui ainda a elaboração de diagramas (casos de uso, estados, BPMN), BACKLOG, DER, protótipos navegáveis e relatórios de testes de unidade e integração, conforme apresentado na Figura 32.

O software desenvolvido tem como principal objetivo coletar, armazenar, processar e apresentar os dados dos lançamentos do foguete de água. Ele permite que os usuários importem os dados registrados pelos sensores do foguete (armazenados em um pendrive ou cartão SD), visualizem esses dados em forma de relatórios e gráficos e mantenham um histórico dos lançamentos realizados.

Além disso, o software possui duas interfaces: uma interface em linha de comando (CLI) voltada para usuários que preferem comandos diretos e uma interface gráfica (GUI) desenvolvida para tornar a visualização dos dados mais acessível e amigável.

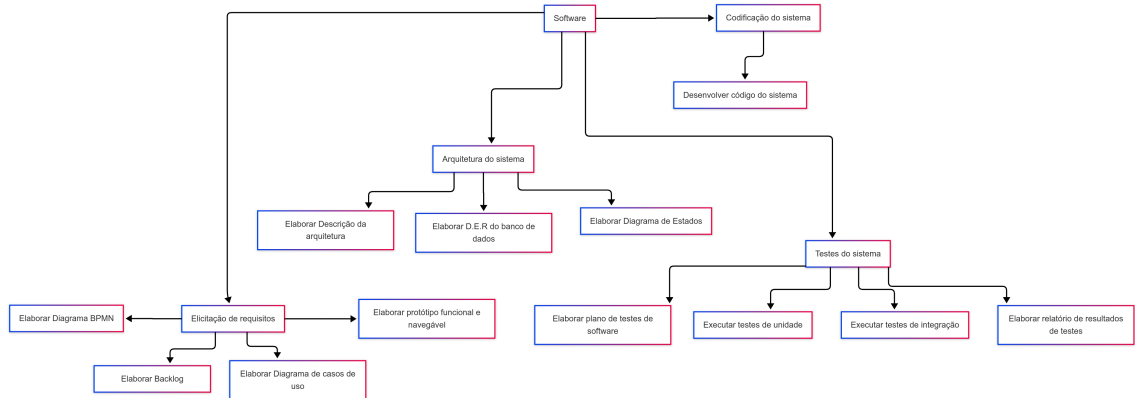


Figura 32 – Pacote de Trabalho 1.3 – Software

O pacote MOSCOW é uma técnica de priorização de requisitos que classifica as funcionalidades em quatro categorias: Must have (deve ter), Should have (deveria ter), Could have (poderia ter) e Won't have this time (não terá desta vez). Essa abordagem ajuda a focar no que é essencial para o sucesso do projeto, garantindo que os recursos sejam alocados de forma eficiente. A Figura 33 ilustra a aplicação dessa técnica no contexto do projeto.

Tabela 33 – Tabela de Requisitos do Projeto

ID	DESCRIÇÃO	PRIORIDADE
RQ01	Importar dados de voo em JSON de dispositivo externo	Must have
RQ02	Exibir gráfico de velocidade vertical vs. tempo.	Must have
RQ03	Exibir gráfico de aceleração vertical vs. tempo.	Must have
RQ04	Exibir gráfico de dispersão da trajetória no plano X e Y.	Must have
RQ05	Exibir gráfico de altitude vs. tempo.	Must have
RQ06	Exibir valores máximos e mínimos de aceleração, velocidade e ângulo.	Should have
RQ07	Exibir tempo de execução e intervalos de amostragem.	Should have
RQ08	Exportar dados do voo em JSON.	Must have
RQ09	Importar arquivos JSON para comparação e simulação.	Must have
RQ10	Aplicar filtro de média móvel nos dados dos sensores.	Should have

Os requisitos não-funcionais são critérios que definem a qualidade e as restrições do sistema, como desempenho, segurança, usabilidade e portabilidade. Eles são essenciais para garantir que o sistema atenda às expectativas dos usuários e funcione de maneira eficiente em diferentes ambientes. A Tabela 34 apresenta os requisitos não-funcionais identificados para o projeto.

O pacote de Casos de Uso é uma técnica de modelagem que descreve as interações entre os usuários (atores) e o sistema, detalhando como os requisitos funcionais serão atendidos. Os casos de uso ajudam a identificar as funcionalidades essenciais do sistema e a garantir que todas as partes interessadas tenham uma compreensão clara dos objetivos



Tabela 34 – Requisitos Não-Funcionais

ID	DESCRIÇÃO
RNF01	Os dados dos lançamentos devem ser armazenados localmente de forma que não possam ser corrompidos em caso de desligamento inesperado.
RNF02	O software deve validar a integridade dos arquivos JSON antes de processar os dados.
RNF03	O software deve ser multiplataforma, funcionando em pelo menos Windows, Linux e MacOS.
RNF04	Não deve depender de conexão com internet para funcionar.
RNF05	A geração de relatórios e gráficos não deve exceder 3 segundos para arquivos de até 100 lançamentos.
RNF06	A interface gráfica (GUI) deve ser intuitiva, com botões claros para importar dados, gerar gráficos e acessar os lançamentos anteriores.
RNF07	O CLI deve ter comandos simples.

do projeto. A Figura 33 apresenta um exemplo de diagrama de casos de uso utilizado no projeto.

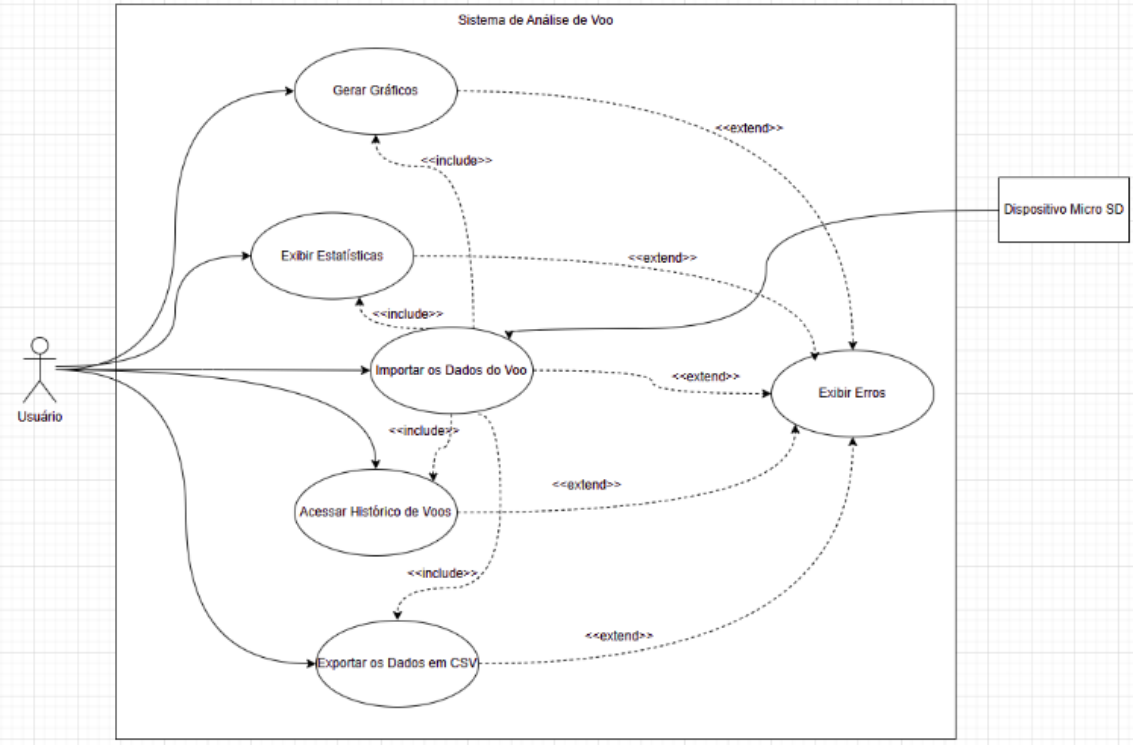


Figura 33 – Pacote de Trabalho 1.3 – Casos de Uso

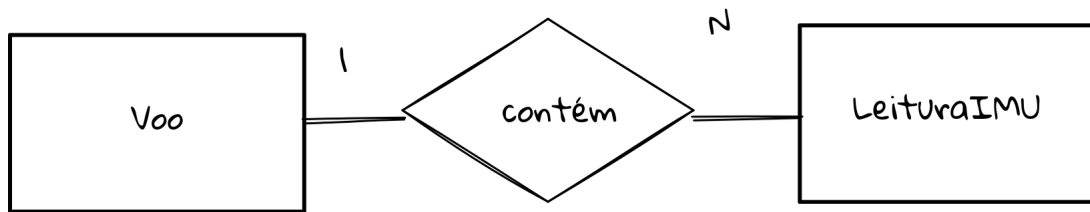


Figura 34 – Pacote de Trabalho 1.3 – Diagrama de Entidade-Relacionamento

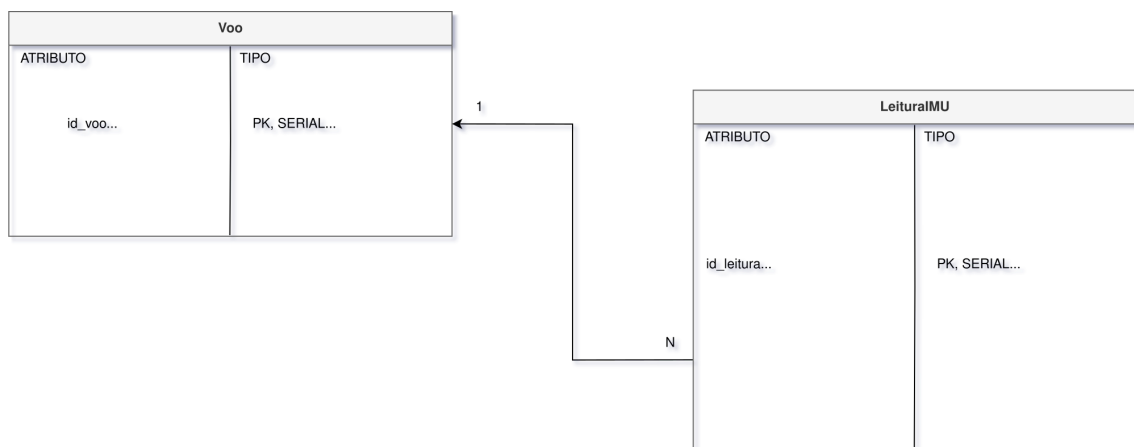


Figura 35 – Pacote de Trabalho 1.3 – Atributos das Entidades

O Diagrama de Entidade-Relacionamento (DER) é uma técnica de modelagem de dados que representa as entidades do sistema e os relacionamentos entre elas. É fundamental para a construção do banco de dados e para garantir que os dados sejam organizados de forma eficiente. A Figura 34 ilustra um exemplo de DER utilizado no projeto.

Após isso, podemos modelar os atributos de cada entidade, conforme os dados que receberemos da ERP32, que calculará movimentos de unidade de medição inercial pelo sensor de 6 eixos, como o acelerômetro e o giroscópio. A Figura 35 apresenta um exemplo de atributos modelados para as entidades do sistema.

O padrão arquitetural escolhido é o Monolítico, por se tratar de um sistema simples, com poucos módulos e baixa complexidade. A adoção de uma arquitetura mais robusta, como microsserviços, não se justifica neste contexto, uma vez que todo o processamento ocorre localmente e o software não possui dependências distribuídas.

A arquitetura monolítica permite concentrar todas as funcionalidades como leitura de dados, processamento, geração de relatórios e apresentação dos resultados em um único sistema, facilitando o desenvolvimento, testes e manutenção dentro dos requisitos do projeto, como visto na Figura 36.

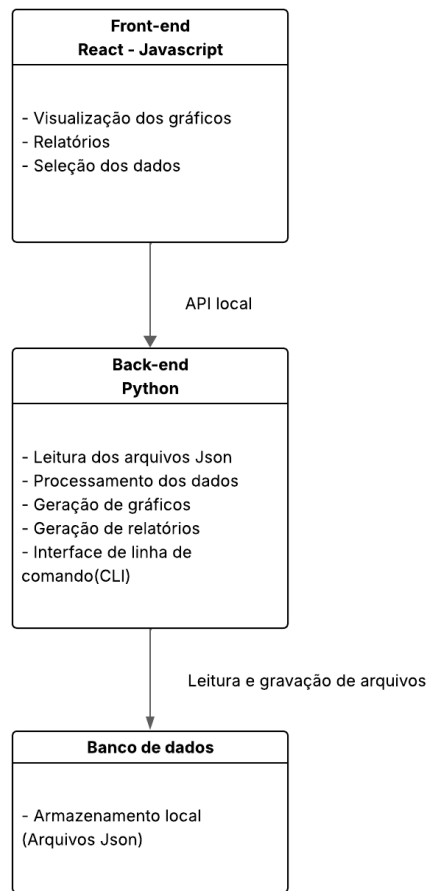


Figura 36 – Arquitetura do Software

As linguagens de programação escolhidas para o desenvolvimento do software são Python e JavaScript, cada uma com suas responsabilidades específicas dentro do sistema.

Python é responsável pelo desenvolvimento do backend e da interface de linha de comando (CLI). Utilizado para processamento dos dados, geração de relatórios, análises estatísticas e manipulação dos arquivos JSON. Utilizamos as bibliotecas Pandas para manipulação e análise de dados. NumPy para cálculos matemáticos e operações com arrays. E Matplotlib para criação de gráficos e visualizações dos dados.

JavaScript é utilizado no desenvolvimento do frontend (GUI), através da biblioteca React, para apresentar visualmente os dados dos lançamentos de forma interativa e amigável.

Neste projeto, não será utilizado um banco de dados tradicional. Os dados dos lançamentos serão armazenados localmente em arquivos JSON, por se tratar de uma solução leve, simples e suficiente para o volume de dados do projeto. Cada arquivo JSON conterá os dados de um lançamento específico, organizados de forma que possam ser facilmente acessados tanto pela CLI quanto pela interface gráfica.

### 4.2.3 Hardware

Este pacote contempla a concepção eletrônica do sistema, incluindo a elaboração de diagramas de blocos e esquemáticos, a instalação de sensores e atuadores no foguete e na base de lançamento, além da realização de experimentos de hardware e de integração. Envolve também a documentação técnica e a avaliação do desempenho dos fornecedores, conforme Figura 37.

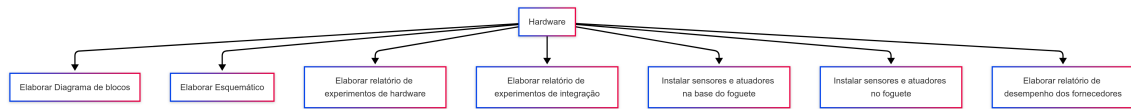


Figura 37 – Pacote de Trabalho 1.4 – Hardware

### 4.2.4 Energia

Foca na análise do consumo energético do sistema, na seleção e validação da fonte de alimentação e na condução de experimentos relacionados à autonomia e estabilidade energética. Assim como nas demais frentes técnicas, inclui relatório de desempenho dos fornecedores de componentes de energia, conforme Figura 38.

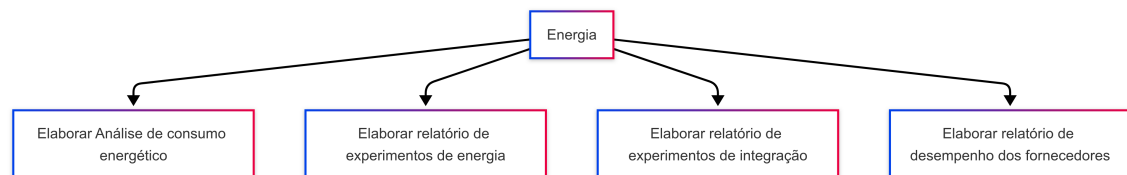


Figura 38 – Pacote de Trabalho 1.5 – Energia

### 4.2.5 Integração

Este pacote agrupa atividades de integração entre as frentes técnicas, validação dos critérios de sucesso do projeto (precisão de trajetória, reutilização, segurança), além da preparação da apresentação final e do vídeo demonstrativo. Essa fase garante que o sistema na totalidade atenda ao escopo e aos requisitos definidos, conforme representado na Figura 39.

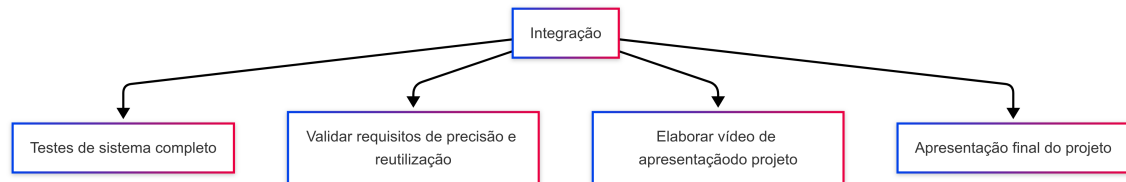


Figura 39 – Pacote de Trabalho 1.6 – Integração

## 5 Cronograma do projeto

Tabela 35 – Cronograma geral.

<b>Geral</b>	<b>Previsto</b>	<b>Realizado</b>
Início do projeto	21/04/2025	21/04/2025
Fim do projeto	18/07/2025	-

Tabela 36 – Cronograma de Atividades do Projeto (Parte 1)

N	Atividade	Início Prev.	Início Real.	Fim Prev.	Fim Real.	Prev.	Responsáveis
1	Realizar Kickoff	21/04	21/04	24/04	24/04		VL
2	Elaborar TAP do projeto	21/04	21/04	04/05	04/05	1	VL
3	Unificar EAP	01/05	01/05	03/05	03/05		VL
4	Unificar Cronogramas	03/05	03/05	05/05	05/05	3	VL
5	Formatar o relatório ED1	01/05	01/05	05/05	05/05		VL, JF, RG, MQ, LL
6	Elaborar relatório de orçamento geral	06/05	-	29/05	-		VL, HM, IL, LL
7	Elaborar Desenho em CAD	06/05	-	29/05	-		EM, GC, IL, KP, SC
8	Elaborar inventário de materiais	06/05	-	29/05	-		EM, GC, IL, KP, SC
9	Relatório de decisões de projeto (estrutura)	06/05	-	29/05	-	7,8	EM, GC, IL, KP, SC
10	Elaborar Diagrama BPMN	06/05	28/05	29/05	30/05		EM, GC, IL, KP, SC
11	Elaborar Backlog funcional	06/05	05/04	29/05	30/05		HM, IL, LL, VL
12	Elaborar Backlog não-funcional	06/05	05/04	29/05	30/05		HM, IL, LL, VL
13	Diagrama de casos de uso	06/05	21/05	29/05	31/05		NA, SS, AV, JF, MQ, LA, LJ
14	Elaborar Diagrama de blocos	06/05	29/05	29/05	31/05		VL, RG, LL, PD, LB
15	Esquemático de eletrônica	06/05	-	29/05	-		VL, RG, LL, PD, LB
16	Descrição da arquitetura	06/05	29/05	29/05	30/05	10-13	NA, SS, AV, JF, MQ, LA, LJ
17	D.E.R do banco de dados	06/05	28/05	29/05	25/05	16	NA, SS, AV, JF, MQ, LA, LJ
18	Diagrama de Estados	06/05	19/05	29/05	30/05	16	NA, SS, AV, JF, MQ, LA, LJ
19	Protótipo funcional navegável	06/05	15/05	29/05	30/05	11-13	NA, SS, AV, JF, MQ, LA, LJ
20	Plano de testes de software	06/05	23/05	29/05	30/05		NA, SS, AV, JF, MQ, LA, LJ
21	Análise de consumo energético	06/05	-	29/05	-	15	NA, SS, AV, JF, MQ, LA, LJ
22	Comprar materiais	06/05	-	02/06	-	6	VL, SC, IL
23	Unificar relatórios para ED2	30/05	01/05	02/06	02/05		VL, JF, RG, MQ, LL, LA
24	Formatar o relatório ED2	30/05	01/05	02/06	02/05		VL, JF, RG, MQ, LL, LA

Tabela 37 – Cronograma de Atividades do Projeto (Parte 2)

N	Atividade	Início Prev.	Início Real.	Fim Prev.	Fim Real.	Prev.	Responsáveis
25	Código do sistema Interface	13/05	-	02/07	-	16	HM, NA, SS, AV, JF, MQ, LA, LJ
26	Código do software embarcado	13/05		02/07		14,15	VL, RG, LL, PD
27	Montar fuselagem do foguete	13/05	-	02/07	-	7	EM, GC, IL, KP, SC
28	Montar base do foguete	13/05	-	02/07	-	7	EM, GC, IL, KP, SC
29	Sensores na base do foguete	13/05	-	02/07	-	28	VL, RG, LL, PD, LB
30	Sensores no foguete	13/05	-	02/07	-	27	VL, RG, LL, PD, LB
31	Testes de unidade	13/05	-	02/07	-	20	HM, NA, SS, AV, JF, MQ, LA, LJ
32	Testes de integração	02/07	-	11/07	-	20	HM, NA, SS, AV, JF, MQ, LA, LJ
33	Resultados dos testes	02/07	-	11/07	-	31,32	MQ
34	Relatório de hardware	02/07	-	11/07	-		VL, RG, LL, PD, LB
35	Relatório de estrutura	02/07	-	11/07	-		IL, SC, GC
36	Relatório de integração	02/07	-	11/07	-		Todos os integrantes
37	Desempenho de fornecedores	02/07	-	11/07	-		VL, SC, IL
38	Relatório de energia	02/07	-	11/07	-		LB
39	Experimentos de integração	02/07	-	11/07	-		Todos os integrantes
40	Validar testes completos	12/07	-	15/07	-	30,29	Todos os integrantes
41	Validar precisão e reutilização	12/07	-	15/07	-		VL, HM, IL, LL
42	Vídeo de apresentação	12/07	-	15/07	-	41	Todos os integrantes
43	Relatório planejado vs. realizado	12/07	-	15/07	-		VL, HM, IL, LL
44	Avaliação SWOT do projeto	12/07	-	15/07	-		Todos os integrantes
45	Lições aprendidas	12/07	-	15/07	-		Todos os integrantes
46	Avaliar desempenho da equipe	12/07	-	15/07	-		VL, HM, IL, LL
47	Relatório final ED3	12/07	-	16/07	-		VL, JF, RG, MQ, LL, LA
48	Apresentação final	16/07	-	18/07	-	41,42,46,47	Todos os integrantes



## Referências

GUIMARÃES, S.; FRANCISCO, G. Automação para a base de lançamento de um foguete pet com redução de torque. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, v. 2, n. 8, 2020. Disponível em: <<https://www.nucleodoconhecimento.com.br/engenharias/foguete-pet>>. Citado na página 8.

National Association of Rocketry. *Model Rocket Safety Code*. 2012. Disponível em: <<https://www.nar.org/safety-information/model-rocket-safety-code/>>. Citado na página 8.

Project Management Institute. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK)*. 6. ed. Newtown Square, PA: Project Management Institute, 2017. Tradução oficial. Citado na página 15.

YUKIMITSU, G. Automação de uma base lançadora de foguetes com propulsão hidrostática para participação em olimpíadas. In: *Anais do CONICT - Congresso de Inovação, Ciência e Tecnologia*. Campinas, SP: Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, 2020. Disponível em: <<https://ocs.ifsp.edu.br/conict/xiconict/paper/view/6690/1754>>. Citado na página 8.