

Sistema FIEB



PELO FUTURO DA INOVAÇÃO

CENTRO UNIVERSITÁRIO SENAI CIMATEC

Engenharia de Computação

Engenharia de Controle e Automação

Engenharia Elétrica

Trabalho de Conclusão de Curso - Metodologia TheoPrax

**Projeto BIRHex: Plataforma robótica autônoma  
para inspeção de ambientes que segue a morfologia  
Rhex**

Apresentada por: Amã Vickto Sacramento Fair  
Etevaldo Andrade Cardoso Neto  
Paulo da Veiga Dultra Bisneto  
Uelinton Vitor Conceição da Silva

Orientador: Prof. Marco Reis, M.Eng.

Fevereiro de 2021

Amã Vickto Sacramento Fair  
Etevaldo Andrade Cardoso Neto  
Paulo da Veiga Dultra Bisneto  
Uelinton Vitor Conceição da Silva

# Projeto BIRHex: Plataforma robótica autônoma para inspeção de ambientes que segue a morfologia Rhex

Trabalho de Conclusão de Curso - Metodologia TheoPrax apresentada ao Programa de Graduação em Engenharia de Computação , Engenharia de Controle e Automação e Engenharia Elétrica do Centro Universitário SENAI CIMATEC, como requisito parcial para a obtenção do título de **Bacharel em Engenharia**.

Área de conhecimento: Robótica

Orientador: Prof. Marco Reis, M.Eng.

# Centro Universitário SENAI CIMATEC

Engenharia de Computação

Engenharia de Controle e Automação

Engenharia Elétrica

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação do Trabalho de Conclusão de Curso - Metodologia TheoPrax, intitulado “Projeto BIRHex: Plataforma robótica autônoma para inspeção de ambientes que segue a morfologia Rhex”, apresentada no dia 08 de Fevereiro de 2021, como requisito parcial para a obtenção do título de **Bacharel em Engenharia**.

Orientador:

---

Prof. Me. Marco Antonio dos Reis  
Centro Universitário SENAI CIMATEC

Orientador TheoPrax:

---

Prof. MBA João Lucas da Hora  
Centro Universitário SENAI CIMATEC

Coordenador de curso:

---

Prof. Me. Murilo Plínio Nogueira Ribeiro  
Centro Universitário SENAI CIMATEC

Coordenador de curso:

---

Prof. Me. Taniel Silva Franklin  
Centro Universitário SENAI CIMATEC

Coordenador de curso:

---

Prof. Dr. Sérgio Oliveira Pitombo  
Centro Universitário SENAI CIMATEC

---

## Agradecimentos

---

Gostaríamos de agradecer a área de robótica do SENAI CIMATEC por proporcionar a oportunidade de realizarmos esse projeto. Agradecemos ao nosso orientador, Marco Antonio dos Reis, por todas as significativas contribuições no presente projeto principalmente com o suporte na relação com os *stakeholders* e por tornar possível a criação de uma equipe multidisciplinar. Ao professor João Lucas da Hora pelo acompanhamento das nossas atividades e todo suporte quanto a metodologia TheoPrax. Finalmente gostaríamos de agradecer ao Centro Universitário SENAI CIMATEC por toda infraestrutura oferecida durante nosso período como alunos da graduação, assim como a todos os professores e outros profissionais que foram responsáveis pela nossa formação.

Eu, Amã Vickto Sacramento Fair, gostaria de agradecer primeiramente a todos os meus familiares, com destaque a meus pais, Renê Fair e Cristiane Sacramento, e aos meus avós, Averaldo Sacramento e Lindinalva Sacramento, por todo o suporte e dedicação, dando diariamente exemplos de responsabilidade, honestidade, amor e respeito ao próximo. Também agradeço a todos os meus amigos, em especial aos Pangolas, pelos momentos de parceria, suporte e união nos últimos anos. Gostaria de agradecer em especial a Leonardo Bacelar, pelo exemplo de amizade, pelos conselhos, ensinamentos e por se mostrar sempre disponível e presente em todos os momentos dessa trajetória. Por fim, um agradecimento especial a Branilson Luiz, pelos últimos quatro anos de ensinamentos, conselhos e contribuições significativas em minha formação profissional.

Eu, Etevaldo Neto, gostaria de agradecer a toda minha família, principalmente aos meus pais, Etevaldo Júnior e Regina Gabriel, e os meus avós, Etevaldo Cardoso e Antônia Lúcia, que sempre incentivaram e proporcionaram os meus estudos. Também agradeço aos meus amigos por todo suporte e apoio durante a graduação.

Eu, Paulo da Veiga Dultra Bisneto, gostaria de agradecer aos meus pais, Jarbas Santos Dultra e Sylvia Maria Marques Ribeiro Dultra, por sempre terem me apoiando e por terem me dado a oportunidade de estar aqui hoje me graduando. Agradeço aos meus colegas que estiveram lado a lado comigo durante toda a graduação.

Eu, Uelinton Vitor Conceição da Silva, deixo em meu relato um agradecimento primeiramente e especial à minha mãe, Janice Conceição da Silva, que dentre todas as circunstâncias e dificuldades que tive durante a vida me apoiou e orientou prezando sempre meu melhor. À minha irmã, Cristhiane Conceição da Silva, onde em conversas foi capaz de me ajudar a ter calma e contornar melhor as situações em que me causavam certa angústia e temor. Por fim, espero que essa pesquisa possa ser um ponto de estímulo futuro para meu filho, Heitor Silveira Conceição Mendes e Silva, no estudo da área da ciência e tecnologia.



---

## Resumo

---

Devido ao aumento de desastres ambientais do Brasil, evidenciados pelos casos de Mariana e de Brumadinho, foi questionado o uso dos sistemas robóticos autônomos nesse problema e também no desafio de projetar e desenvolver um robô móvel autônomo com a capacidade de se locomover sobre terrenos acidentados. Para isso, esse trabalho tem como objetivos estudos de plataformas já existentes, construir códigos e pacotes de percepção e navegação, além da criação dos esquemas elétricos e desenhos mecânicos. A metodologia utilizada separa o projeto em seis etapas: Levantamento de Requisitos, Criação do Conceito, Funcionalidades, Modelagem, Simulação e Desenvolvimento. Cada etapa dessa metodologia gera materiais para as próximas, com isso o desenvolvimento do projeto evolui. Ao final, é apresentado os resultados obtidos com a simulação do Gazebo integrado ao ROS, os esquemáticos e modelos do robô.

**Palavras-chave.** Robótica Móvel, SLAM Visual, Hexápodes.

---

## Abstract

---

In consequence of the increase in environmental disasters in Brazil, emphasized by Mariana and Brumadinho, was questioned about the use of autonomous robotic systems in this problem and also in the challenge of designing and developing a mobile robot autonomous with the ability to move over rough terrain. Therefore, this work aims to studies of existing platforms, build codes and packages of perception and navigation, in addition to creating electrical schemes and mechanical drawings. The used methodology separates the project into six stages: Survey of Requirements, Concept Creation, Functionalities, Modeling, Simulation, and Development. Each step of this methodology creates new materials for the next ones, due to this, the development of the project evolves. In the end, the results obtained are presented with the simulation of the Gazebo integrated with ROS, the schematics, and models of the robot.

**Keywords:** Mobile Robotics, Vision-Based SLAM, Hexapods

---

# Sumário

---

Lista de abreviatura e siglas	viii
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivos	2
1.1.1 Objetivos Específicos	2
1.2 Justificativa	2
<b>2 REFERENCIAL TEÓRICO</b>	<b>6</b>
2.1 Robótica Móvel	6
2.1.1 Robôs Terrestres	6
2.1.1.1 Robôs com Roda	6
2.1.1.2 Robôs com Esteira	8
2.1.1.3 Robôs com Pernas	9
2.1.2 Veículos Aéreos não Tripulados	9
2.1.3 Veículos Submarino Autônomo	10
2.2 Bio-Inspirados	11
2.2.1 Bípedes	12
2.2.2 Quadrúpedes	13
2.2.3 Hexápodes	15
2.3 Robôs de Modelo Rhex	16
2.3.1 Research-RHex	17
2.3.2 Shelley-RHex	18
2.3.3 Rugged-RHex	19
2.3.4 Edubot	20
2.4 Marchas de Locomoção para Robôs com Patas	21
2.4.1 A Marcha <i>Tripod</i>	21
2.5 <i>Framework</i> ROS	23
2.5.1 Nós e Tópicos	23
2.5.2 Serviços e <i>Actions</i>	24
<b>3 METODOLOGIA</b>	<b>25</b>
3.1 Requisitos	25
3.2 Conceito	26
3.3 Funcionalidades	26
3.4 Modelagem	26
3.5 Simulação	27
3.6 Desenvolvimento	27
<b>4 RESULTADOS E ANÁLISES</b>	<b>28</b>
4.1 Modelo	28
4.1.1 Eletrônica	28
4.1.2 Sensoriamento e Smart Devices	31
4.1.2.1 Sistema de Segurança	35
4.1.3 Mecânica	37
4.1.3.1 Upside Housing	38

---

4.1.3.2	Downside Housing . . . . .	41
4.1.3.3	Modelo Completo . . . . .	41
4.2	Simulação . . . . .	43
4.2.1	Modelo de Simulação do Robô . . . . .	43
4.2.2	Ambiente . . . . .	45
4.2.3	Estrutura de Software . . . . .	46
4.2.3.1	Pacote de Locomoção . . . . .	49
4.2.3.2	Pacote de Localização . . . . .	51
4.2.3.3	Pacote de Navegação . . . . .	54
4.2.4	Testes de desempenho . . . . .	55
4.2.4.1	Locomoção . . . . .	55
4.2.4.2	Localização e Navegação . . . . .	58
4.2.5	Resultados . . . . .	64
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>68</b>
5.1	Trabalhos Futuros . . . . .	69
<b>A</b>	<b>Diagrama I</b> . . . . .	<b>70</b>
<b>B</b>	<b>Diagrama II</b> . . . . .	<b>71</b>
<b>C</b>	<b>Esquemáticos eletrônicos</b> . . . . .	<b>72</b>
<b>D</b>	<b>Desenhos Mecânicos</b> . . . . .	<b>78</b>
	<b>Referências</b> . . . . .	<b>81</b>

---

## Lista de Tabelas

---

4.1	Resultados dos Comandos Lineares. . . . .	56
4.2	Resultados dos Comandos Angulares. . . . .	57
4.3	Resultados dos Comandos Mistos. . . . .	58
4.4	Tabela de Erro Ground Truth vs Odometria Visual . . . . .	60
4.5	Tabela de Erro Ground Truth vs LIDAR . . . . .	61
4.6	Tabela de Erro Ground Truth vs GPS . . . . .	62
4.7	Tabela de Erro Ground Truth vs Odometria Global . . . . .	63
4.8	Percurso sem obstáculos . . . . .	66
4.9	Percurso com obstáculos . . . . .	67
4.10	Percurso com obstáculos e terreno acidentado . . . . .	67

---

## Lista de Figuras

---

2.1	Veículo autônomo guiado. . . . .	7
2.2	Carro autônomo. . . . .	7
2.3	Robô aspirador criado pela iRobot. . . . .	8
2.4	Robô com esteira . . . . .	9
2.5	Arranjos de pernas de alguns animais. . . . .	9
2.6	Veículo Aéreo não Tripulado. . . . .	10
2.7	Show de drones . . . . .	10
2.8	Aplicações do AUV. . . . .	11
2.9	Robôs bio-inspirados . . . . .	12
2.10	HRP-2 . . . . .	13
2.11	JROB-1 . . . . .	14
2.12	PhantomX AX Hexapod Mark II . . . . .	15
2.13	Adaptive Robot . . . . .	16
2.14	Rhex . . . . .	17
2.15	Rhex no ambiente de testes . . . . .	18
2.16	Research-RHex . . . . .	18
2.17	Shelley-RHex . . . . .	19
2.18	Rugged-RHex . . . . .	20
2.19	EduBot . . . . .	20
2.20	Marcha definida por $p$ , $\delta$ e o tempo de ciclo. . . . .	22
2.21	Exemplo dos triângulos de sustentação. . . . .	23
2.22	Relação entre Nós e Tópicos. . . . .	24
3.1	Metodologia aplicada ao desenvolvimento do projeto. . . . .	25
4.1	BIRHex Board . . . . .	29
4.2	BIRHex Fuse Board . . . . .	30
4.3	BIRHex Test Board . . . . .	31
4.4	Phidgets . . . . .	32
4.5	Dynamixel . . . . .	33
4.6	RPLidar . . . . .	33
4.7	Stereo Zed . . . . .	34
4.8	FLIR . . . . .	34
4.9	GPS PPKSI . . . . .	35
4.10	IMU xSens . . . . .	35
4.11	BIRhex Fuse Board no robô . . . . .	36
4.12	Botão de Emergência . . . . .	37
4.13	Conjunto de pata . . . . .	38
4.14	Upside Housing . . . . .	39
4.15	Upside Housing Top . . . . .	40
4.16	Upside Housing Bottom . . . . .	40
4.17	Downside Housing . . . . .	41
4.18	Montagem Completa . . . . .	42
4.19	Visão Superior da Montagem Completa . . . . .	42
4.20	Visão Traseira da Montagem Completa . . . . .	42

4.21	Modelo Simplificado. . . . .	43
4.22	Funcionamento Câmera Estéreo. . . . .	44
4.23	Funcionamento LIDAR. . . . .	44
4.24	Ambiente Virtual do BiRhex . . . . .	45
4.25	Vítima e Túnel do Parque . . . . .	45
4.26	Obstáculos do Parque . . . . .	46
4.27	Bifurcações do Parque . . . . .	46
4.28	Caminhos sem Saída do Parque . . . . .	47
4.29	Fluxo de Software Autônomo. . . . .	48
4.30	Fluxo de Software Operado. . . . .	49
4.31	Exemplo de funcionamento do suavizador de transição. . . . .	51
4.32	Funcionamento do stereo_image_proc no pacote de localização . . . . .	52
4.33	Funcionamento do rtabmap_ros no pacote de localização . . . . .	52
4.34	Funcionamento do hector_mapping no pacote de localização . . . . .	53
4.35	Funcionamento do robot_localization no pacote de localização . . . . .	53
4.36	Funcionamento pacote de localização . . . . .	54
4.37	Funcionamento pacote de Navegação . . . . .	55
4.38	Degradação da Velocidade Angular . . . . .	57
4.39	Mapa do ambiente de simulação . . . . .	59
4.40	Gráfico de posição Ground Truth vs Odometria Visual . . . . .	60
4.41	Gráfico de posição Ground Truth vs LIDAR . . . . .	61
4.42	Gráfico de posição Ground Truth vs GPS . . . . .	62
4.43	Gráfico de posição Ground Truth vs Odometria Global . . . . .	63
4.44	Gráfico de posição Ground Truth vs Odometria Global (SLAM) . . . . .	64
4.45	Mapa sem obstáculos . . . . .	65
4.46	Mapa com obstáculos . . . . .	65

---

## Lista de abreviatura e siglas

---

**AGV** *Automated Guided Vehicle*

**AUV** *Autonomous Underwater Vehicle*

**BIR** *Brazilian Institute of Robotics*

**DARPA** *Defense Advanced Research Projects Agency*

**EKF** *Extended Kalman Filter*

**GPS** *Global Positioning System*

**IMU** *Inertial Measurement Unit*

**LIDAR** *Light Detect and Ranging*

**OFDA/CRED** *The Office of U.S.Foreign Disaster Assistance/Centre for Research on the Epidemiology of Disasters*

**PCI** *Placa de Circuito Impresso*

**QFD** *Quality Function Deployment*

**ROS** *Robotic Operating System*

**RTAB-Map** *Real-Time Appearance-Based Mapping*

**SLAM** *Simultaneous Localization and Mapping*

**SOTA** *State of the Art*

**UAB** *Unmanned Aerial Vehicles*

**UKF** *Unscented Kalman Filter*



## INTRODUÇÃO

---

Durante a história da humanidade um fato muito recorrente foi a exploração de novos ambientes, muitos desses perigosos e inóspitos, dificultando a busca de novos locais habitáveis.

Em 1845 ocorria uma expedição liderada pelo capitão Sir John Franklin (A... , 2016), 59 anos, com o objetivo atravessar a secção (única até então) da Passagem do Noroeste no ártico onde após três sucessivas expedições essa última culminou em tragédia. Todos os tripulantes faleceram, aproximadamente 128 homens.

Somente em 1906 uma exploração culminou em êxito e atravessou a Passagem do Noroeste com o comando do norueguês Roald Amundsen, tornando-se o primeiro a completar essa missão que muitos falharam.

Hoje, a Passagem do Noroeste já é conhecida, mas ainda não podendo ser usada comercialmente. Sua vantagem seria na diminuição de milhas em rotas, como: Nova Iorque e Tóquio em 3000 milhas; Londres e Tóquio de 15000 milhas para menos de 8000 milhas; Yokohama e Liverpool de 12250 para 7000 milhas.

Além das situações exploratórias têm-se também as inspeccionais. Devido às necessidades de inspeções em casos de incêndios, queimadas ambientais e queda de edifícios, que são momentos difíceis de averiguação e resgate de um ser vivo, tanto pelo tempo curto quanto pela complexidade em certos acessos. Depara-se constantemente com notícias que relatam mortes causadas por desabamentos/queimadas, e nessas situações opta-se por enviar bombeiros com o auxílio de cães, assumindo assim os riscos de que possam acontecer acidentes graves.

Em casos como os descritos acima, a utilização de robôs com alta mobilidade tem se tornado mais comum e ideal para a preservação da vida dos que fazem os resgates, já que os locais são hostis à presença humana. Na robótica móvel atual, existe uma grande variedade de plataformas como: humanoides, veículos terrestres, aeroespaciais e aquáticos. Dentre os veículos terrestres, pode se citar os robôs móveis com patas, alvo desse trabalho, que se destacam pela alta capacidade de deslocamento em terrenos não uniformes e desconhecidos. A grande vantagem nessa abordagem é a de um contato em pontos específicos com o solo e não mais uma alta área da superfície, trazendo assim diferentes movimentos locomotivos.

A aplicabilidade geral da robótica móvel se dá por: navegação - criar formas de representação do ambiente; reconhecimento - reconhecer o ambiente ao qual está incluído; aprendizado - calibração de sensores de forma autônoma para plena execução das tarefas; cooperação - possibilidade de pequenos robôs atuarem em conjunto e cognição - novas formas de armazenar o conhecimento e utilizar. Motivado por fatores como velocidade, economia de combustível, mobilidade e suavidade deu-se o interesse em construir robôs baseados em muitas pernas, como os hexápodes as aplicações dos robôs hexápodes e já se inspirando no comportamento de alguns insetos, onde o robô reage baseando-se na leitura de informações do ambiente através dos seus sensores.

## 1.1 Objetivos

Com esse projeto, espera-se a criação de uma simulação e projeto detalhado de robô hexápode capaz de se locomover em terrenos acidentados e utilizando uma câmera IR de captura de imagens térmicas, *Light Detect and Ranging* (LIDAR), uma câmera *stereo*, um *Inertial Measurement Unit* (IMU) e um *Global Positioning System* (GPS), seja possível que esse robô consiga se localizar e navegar para realizar inspeções em diferentes ambientes. Além da utilização do *framework Robotic Operating System* (ROS) para integrar esses componentes e funcionalidades.

### 1.1.1 Objetivos Específicos

- Estudar as versões de robô com estilo RHex já desenvolvidas.
- Estudar conceitos importantes da robótica móvel como navegação e localização.
- Desenvolver os *softwares* do robô e os ambientes de simulação no Gazebo integrado com o ROS.
- Criar esquemáticos elétricos e modelos mecânicos.
- Realizar testes de desempenho das funcionalidades implementadas na simulação.

## 1.2 Justificativa

Com o aumento dos números de desastres ambientais no país, um relatório da *The Office of U.S. Foreign Disaster Assistance/Centre for Research on the Epidemiology of Disasters* (OFDA/CRED) declarou que o Brasil tornou-se o décimo país com o maior número de vítimas em desastres ambientais (ONU..., 2015). A baixa manutenção da infraestrutura

associada ao clima tropical causa uma vulnerabilidade grande que propicia acidentes indesejados. Os desastres mais comuns são hidrológicos – inundações, enchentes e movimentos tectônicos – e meteorológicos – vendavais, tornados e alguns casos pequenos furacões -, onde com o crescimento populacional desordenado os novos habitantes concentraram-se em locais fisicamente desfavoráveis (COM... , 2018).

No Brasil, desastres além de serem causados naturalmente também ocorrem por conta da irresponsabilidade diária. No dia 5 de novembro de 2015, no Dique do Fundão (Mariana), ocorreu o rompimento da barragem e causou um nível de desastre alto. Especialistas consultados na época foram unânimes em afirmar que a utilização de técnicas modernas e manutenção adequadas teriam evitado totalmente o desastre e seus impactos diretos e indiretos. Esse caso, considerado por muitos a maior tragédia ambiental da história do país, deixou um legado de muitas mortes tanto dentre os que habitavam o local quanto aos que realizavam os resgates.

Empresas Brasileiras do setor tecnológico e industrial como a TINBOT, UMBÔ e ARMTEC<sup>1</sup>, vem ao longo do tempo buscando formas de apoiar a criação de robôs específicos para essas situações, aumentando sua eficiência produtiva e planejamento estratégico.

A aplicação da robótica autônoma tornou-se um tema importante e que se estende até a atualidade, diversificando o número de estudos na área e dando um grande salto tecnológico. Durante casos emergenciais envolvendo resgates e buscas, os avanços dentro desse campo de estudo cooptou um volume alto de investimentos e estudos tendo assim um crescimento inimaginável.

As inspirações para a construção e estudo nesse segmento vem dos casos em que os seres humanos passaram anteriormente e não possuíam as melhores estruturas e técnicas para a resolução do determinado problema, sendo que, diante desses cenários foi-se necessário avanço a partir das experiências anteriores enfrentadas. Ao longo dos anos o ser humano buscou o uso da tecnologia a seu favor, evitando e/ou diminuindo impactos externos de grandes escalas.

Em 17 de janeiro de 1995, um terremoto de escala 7,3 atingiu a região centro-sul do Japão, Osaka e Kobe, ferindo 40.000 e matando 6400 pessoas em 30 segundos (HAVE... , 2020). Mediante essa desastrosa ocorrência, o Japão articulou-se para a injeção de um grande volume em investimentos internos e externos nos setores de robótica com foco no resgate e busca de sobreviventes. Em pouco mais de 4 meses outro terremoto atingiu o Japão, em 19 de abril de 1995 deixando 500 feridos e 168 mortos, novamente com muitos escombros na região. Após esses impactos, o maior desafio existente encontra-se na busca

---

<sup>1</sup>A Tinbot Robótica é a startup criada em 2017 e desenvolvedora do Tinbot, o primeiro robô assistente brasileiro interativo; A Umbô é uma startup de robótica de atendimento ao consumidor; A ARMTEC atua nas áreas de Defesa e Segurança Pública, Pavimentação, Petróleo e Gás, Divisão Automotiva, e Entretenimento.

de sobreviventes - as primeiras 72 horas são as mais importantes no resgate: sucesso de 81% nas primeiras 24 horas e de 37% em até 48 horas – já que a dificuldade no acesso ao redor do desastre e a precária comunicação local elevam a dificuldade de tal tarefa, existindo ainda alto risco de acidentes e mortes com os buscadores. Casos como em Ankara (Turquia)([AVALANCHE. . . , 2020](#)), onde equipes de resgate sofreram sérios ferimentos, demonstram a real necessidade do avanço tecnológico para situações emergenciais.

Dentre esses acontecimentos de fomento a produção e desenvolvimento de robôs surgiu a RoboCup Rescue ([Tadokoro et al. ,](#) ), onde seu objetivo é a competição internacional de robôs de busca e resgate urbano, na qual os robôs têm como tarefa o encontro das “vítimas” em ambientes urbanos simulados e comparar a performance de diferentes modelos físicos e algoritmos para coordenar e controlar uma equipe de oito robôs. As missões designadas consistem em encontrar fisicamente (com maior precisão e agilidade) pontos que simulam os sinais vitais: visualização, movimentos espontâneos, temperatura corporal, nível de CO2 e sinal sonoro. As áreas de atuação com foco e desenvolvimento contínuo são: Óleo e gás; Águas profundas; Espacial; Busca e Resgate. Em especial no tema de Busca e Resgate, robôs terrestres é a classe mais comum para essa tarefa, preenchendo todos os requisitos necessários. O sistema autônomo Momaro ([BONN, ,](#) ), apresentado em 2017, foi projetado com a capacidade de abertura de portas, válvulas, cortes de paredes e drywall, mobilidade em terreno disforme e habilidade de subir escadas. O mesmo foi constituído de um sistema de pernas com 4 DOF's (graus de liberdade) com rodas em cada uma, dando assim uma capacidade de movimentação omnidirecional para transpor obstáculos. Diferente do veículo terrestre, os drones que por possuírem maior agilidade - mapeamento e identificação – e baixo custo de produção, ganharam um crescente número de pesquisas. Um exemplo foi o projeto desenvolvido por Silvagni, Tonoli, Zenerino, e Chiaberge (2017) ([SILVAGNI et al., 2017](#)), onde se descreveu a utilização de sistemas aéreos autônomos para o resgate de avalanches em áreas montanhosas.

Um dos pontos mais importantes no uso dessas tecnologias é a implementação da inteligência computacional em seu sistema, tornando a mesma cada vez mais capaz de executar tarefas complexas. A partir dessa primeira etapa atingida, novas funcionalidades podem ser adicionadas, dando novas capacidades como: ser arremessado ou cair sem causar danos à sua estrutura física, uma bateria de maior duração, áudio e transmissão de imagem em tempo real, melhorando assim sua usabilidade e contribuindo com uma maior eficácia em suas missões de resgate.

Nós dias atuais o uso dessas ferramentas autônomas nas buscas e resgates em desastres urbanos ainda são muito aquém do esperado, apesar do grau de pesquisa crescer exponencialmente. Um dos momentos mais significativos no seu uso ocorreu no epicentro do WTC (“World Trade Center” – 11/09/2001) quando houveram os ataques terroristas, onde cinco equipes de socorristas chegaram ao local e utilizaram robôs de resgate desenvolvidos pela

CRASAR ("Center for Robot-Assisted Search And Rescue") ([MURPHY, 2018](#)). Principalmente após o ataque, na atualidade as pesquisas são desenvolvidas de forma teórica e experimental com foco nas suas aplicações. Hoje as pesquisas convergem a criação de uma única plataforma ou ao menos uma criação de definidas subcategorias. Nos últimos anos, novas pesquisas de desenvolvimentos expandiram rapidamente as habilidades dos robôs que hoje podem atuar em condições adversas e não possíveis a seres humanos mesmo utilizando equipamentos de proteção.

O incremento da visão computacional a robôs aéreos; e também a outros modelos, permitiu muitas aplicações novas que hoje não se restringem somente a voos de reconhecimento em áreas abertas, com o fim de manter o sistema de GPS ativo mundialmente. As pesquisas em robôs com patas possuíram um avanço significativo, com novas abordagens para o controle e resposta dos atuadores fazendo possível transpor terrenos desafiadores com agilidade e segurança ([BELLICOSO et al., 2018](#)). Adicionalmente, o conceito atual de estudo iniciou-se na bio-inspiração, onde o modelo físico possui alguma semelhança com animais reais a fim de projetar uma vantagem biológica do animal a uma mecânica robótica. Ao nível que todos esses cenários avançam, as pessoas que necessitavam operar remotamente vão encerrando a atividade da tele operação, ocasionando a possibilidade de surgirem produtos comerciais com foco em inspeção ou tarefas sensoriais, podendo ser utilizadas para prevenir e preparar-se para um desastre iminente.

---

## REFERENCIAL TEÓRICO

---

O presente referencial teórico pretende embasar e fundamentar o tema pesquisado, utilizando-se dos estudos mais relevantes e uma seleção de materiais providos por pesquisadores experientes no tema. A importância desse referencial associa-se à coleta de dados e técnicas para a fundamentação de aspectos conceituais, teóricos e práticos do projeto robótico em questão, além de uma consolidação dos resultados encontrados no meio científico. Durante a investigação, conceitos como robótica móvel, robôs bio-inspirados e o robô Rhex foram estudados pela sua relevância e tornaram-se indispensáveis para a elaboração do projeto. A estruturação completa do referencial consiste em cinco grandes tópicos: robótica móvel; robô bio-inspirados; robô modelo Rhex; Marchas e *Framework* ROS.

### 2.1 Robótica Móvel

Robótica móvel é o ramo da robótica especializado em locomoção. Segundo ([MIHELJ et al., 2019](#)) um robô móvel é um robô capaz de mover-se através do ambiente, podendo para isso utilizar-se de rodas, patas, esteiras, ou uma combinação destes. Robôs móveis podem ser bio-inspirados, ou seja, eles tomam como inspiração os seres vivos. Eles também podem voar, nadar, se arrastar ou rolar. Nos próximos tópicos serão abordados os diversos tipos de robôs móveis, com o foco em robôs terrestres.

#### 2.1.1 Robôs Terrestres

Os robôs classificados nessa categoria são projetados para melhor eficiência em contato com o solo e o uso de seu centro de gravidade como um balanceamento passivo e estabilizador. Em sua maioria são subdivididos em: Robôs com roda; Robôs com esteira; Robôs com pernas.

##### 2.1.1.1 Robôs com Roda

Segundo ([SICILIANO et al., 2010](#)) a roda é o mecanismo mais comumente utilizado nas aplicações da robótica móvel. Existem diversas categorias de robôs que utilizam rodas para realizar sua locomoção, como será demonstrado a seguir.



*Automated Guided Vehicle* (AGV) são veículos autônomos que seguem linhas ou fios no chão, ou utilizam video, ímãs, ou *laser* para poder se locomover de um local ao outro sem se perder, ou colidir com outros objetos (MIHELJ et al., 2019). Ele é geralmente utilizado para realizar o transporte de cargas em fábricas e armazéns como é possível observar na figura 2.1.

Figura 2.1: Veículo autônomo guiado.



Fonte: [Robotics e News](#) (2020)

O carro autônomo segundo (JO et al., 2014) é um veículo que consegue perceber o ambiente a qual está inserido e Navega autonomamente por ele através de sensores, atuadores, e de algoritmos complexos de navegação e localização.

Figura 2.2: Carro autônomo.



Fonte: [Techcrunch](#) (2018)

Existem também aqueles robôs que auxiliam as pessoas com suas atividades diárias como os de limpeza e os robôs que auxiliam as enfermeiras a distribuir comida para seus paci-

entes. Eles funcionam de maneira semelhante os carros autônomos, apesar de serem mais simples.

Figura 2.3: Roomba<sup>1</sup>.



Fonte: [TIMES \(2019\)](#)

### 2.1.1.2 Robôs com Esteira

Os robôs que utilizam esteiras para sua locomoção segundo ([MICHAUD et al., 2005](#)) são bastante eficientes em terrenos desnivelados e superfícies macias como a lama e a neve. Devido a sua eficiência em se locomover em diversas superfícies ele é bastante utilizado pelo exército para poder fazer o reconhecimento de áreas hostis e a depender do robô ele pode ainda ser usado para desativar bombas e minas, realizar ataques e ajudar no resgate de vítimas.

---

<sup>1</sup>Robô aspirador criado pela iRobot.



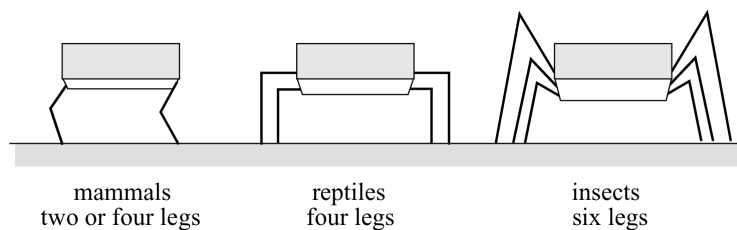
Figura 2.4: Robô com esteira

Fonte: [Sciences \(2019\)](#)

### 2.1.1.3 Robôs com Pernas

Segundo ([SIEGWART; NOURBAKHS; SCARAMUZZA, 2011](#)) robôs com pernas são bio-inspirados e possuem grande adaptabilidade e mobilidade, a quantidade de pernas normalmente variam de 2 a 6 como visto na figura 2.5. Eles são a melhor escolha para se locomover em terrenos de difícil mobilidade em que a utilização de um robô com roda ou com esteira seja inviável.

Figura 2.5: Arranjos de pernas de alguns animais.

Fonte: [Siegwart, Nourbakhsh e Scaramuzza \(2011\)](#)

### 2.1.2 Veículos Aéreos não Tripulados

Segundo ([NONAMI et al., 2010](#)) *Unmanned Aerial Vehicles* (UAV) é qualquer aeronave criada ou modificada para poder operar sem a necessidade da intervenção humana. Eles são bastante utilizados pelos militares para realizar reconhecimento, vigilâncias e ataques. Recentemente polícia começou a utilizar para auxiliar no monitoramento de algumas cidades e algumas empresas começaram a utilizá-los para realizar *shows* aéreos, demonstrado na figura 2.7.

Figura 2.6: Veículo Aéreo não Tripulado.



Fonte: [Harada \(2020\)](#)

Figura 2.7: Show de drones



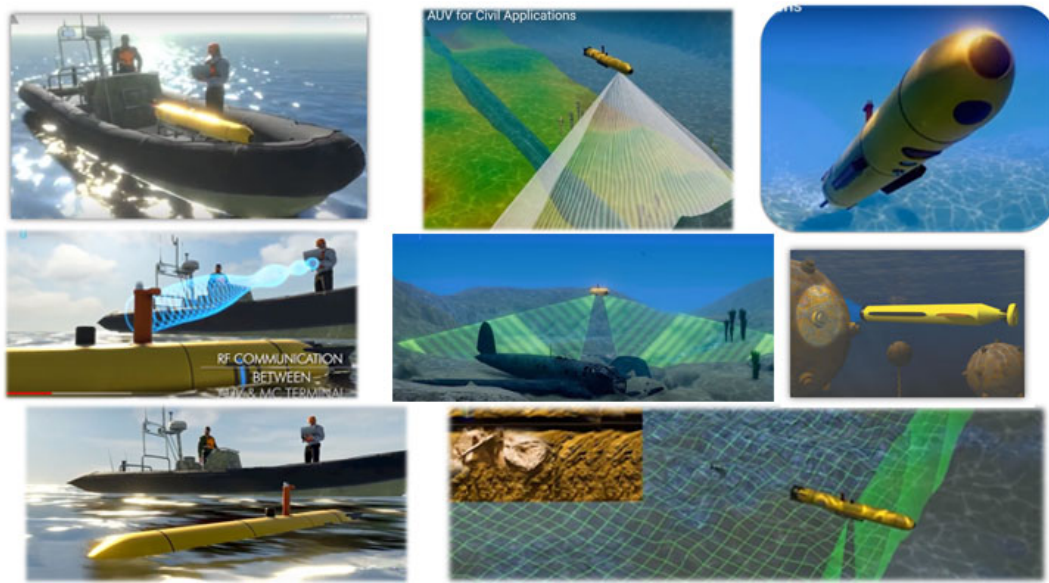
Fonte: [Intel \(2018\)](#)

### 2.1.3 Veículos Submarino Autônomo

O desenvolvimento dos *Autonomous Underwater Vehicle* (AUV) começou no início dos anos 70 e desde então ele vem crescendo em eficiência, tamanho, e a quantidade de memória dos computadores atuais aumentaram bastante o seu potencial ([PAULL et al., 2013](#)). O formato e o tamanho deles variam bastante eles podem ser feitos de uma maneira portátil para que um humano possa carregá-lo a até um com várias toneladas. Diferente dos robôs terrestres e aéreos os robôs subaquáticos sofrem com baixa largura de banda por isso eles têm que ter grande autonomia para poder completar o seu objetivo.

Existem diversas aplicações para o AUV eles podem ser usados para fazer o mapeamento de uma costa, para pesquisas oceanográficas, desativação de minas, no monitoramento da costa entre outros.

Figura 2.8: Aplicações do AUV.



Fonte: [Duman \(2019\)](#)

## 2.2 Bio-Inspirados

A bio-inspiração é a ideia de construir máquinas capazes de imitar características dos animais. Até meados do século XIX, no entanto, não havia a viabilidade científica para que o avanço fosse suficiente na criação de projetos e planos realistas ([RAIBERT, 1986](#)), fazendo com que tentativas verdadeiramente bem-sucedidas de fazer com que os robôs pudessem andar ou engatinhar aumentassem apenas nos últimos anos. O termo ‘inspirado biologicamente’ e o conceito atual de robótica se originaram nas últimas décadas do século XX.

Como os animais são estrutural e funcionalmente complexos, uma reprodução completa de qualquer animal em *hardware* e *software* não é possível. Portanto, há algum debate entre os bio roboticistas sobre onde traçar a linha das possibilidades atuais. Alguns pesquisadores adotam a abordagem de Ritzmann ([RITZMANN et al., 2000](#)), que sugeriram que o maior número possível de recursos de um animal fosse incorporado ao robô, mesmo que a vantagem funcional de um determinado recurso não seja clara ([CHAM; KARPICK; CUTKOSKY, 2004](#); [DILLMANN et al., 2007](#)). Nos últimos anos, essa abordagem tem sido chamada de robótica biométrica ([AYERS; WITTING, 2007](#)). O argumento é que muitos desses recursos realmente conferem atributos úteis ao robô, mesmo que essa utilidade não seja imediatamente aparente. Outros pesquisadores adotaram uma abordagem mais conservadora, sabendo que a inclusão de muitos recursos semelhantes ao dos animais em um robô poderia prejudicar o desempenho ([YONEDA; OTA, 2003](#)).

Em busca de uma maior adaptabilidade aos cenários do dia a dia, o desenvolvimento dos robôs inspiraram-se em seres vivos para uma aproximação biológica. A estrutura física de locomoção animal é versátil e altamente eficiente, dando aos robôs uma mecânica total de controle no ambiente inserido. A capacidade de criação e reprodução de estruturas físicas de animais já existem e pode-se citar de exemplos como grilo ([Birch et al., 2000](#)), galinha ([MEDERREG et al., 2003](#)), gorila ([KAJIMA et al., 2003](#); [DAVIS, 2003](#)), cachorro ([DAVIS, 2003](#); [PENG; LAM; COLE, 2003](#)), tartaruga ([HENNION; PILL; GUINOT, 2006](#)) e lagosta ([AYERS et al., 2000](#)).

Figura 2.9: Robôs bio-inspirados



Fonte: [Diego \(2020\)](#)

### 2.2.1 Bípedes

De início, os robôs bípedes formularam-se a partir da bio-inspiração em seres humanos e sua anatomia, apresentando esses modelos iniciais dificuldades em locomoção e estabilização. O primeiro modelo construído, o Wabot-1 (1973), trabalhava com o rebalanceamento do seu centro de massa em cada perna para executar sua locomoção. O sistema possuía um modo de locomoção estática em uma trajetória pré-planejada, onde conseguia manter seu centro de gravidade com o auxílio dos pés. Durante cada passo ocorria um leve desbalanceamento para rapidamente ocorrer transferência de energia de um pé a outro e em seguida entrando em equilíbrio passivo. Modelos avançados foram capazes de atingir até 0,5 m por minuto.



Outros modelos possuíam outras dificuldades como o da sua estabilização motora (já que alguns possuíam bio-inspiração em dinossauros), um exemplo se deu como demonstrado em Takita ([TAKITA; KATAYAMA; HIROSE, 2003](#)), devido a sua estrutura de equilíbrio encontrar-se em sua cauda e permitir a possibilidade de movimentação com altas velocidades.

Nos anos seguintes, o estudo dos robôs bípedes intensificaram e um grande número de publicações ([GEPPERT, 2004](#); [LÖFFLER; GIENGER; PFEIFFER, 2003](#); [KANEHIRA et al., 2002](#); [KIM; OH, 2004](#); [NISHIWAKI et al., 2000](#); [SAKAGAMI et al., 2002](#)) foram desenvolvidas. Devido ao incentivo do Japão com o HRP (*Humanoid Robotics Program*) um dos projetos mais famosos foi o HRP-2 ([YOKOI et al., 2004](#)), que era capaz de movimentar-se em superfícies irregulares, com aproximadamente 2/3 da velocidade humana, e atravessar locais íngremes utilizando-se da troca de marchas. Outro diferencial era que em caso de queda, ele era capaz de levantar-se sozinho devido a seu sistema físico de “tendões” e “coluna”.

Figura 2.10: HRP-2



Fonte: [Robots \(2020\)](#)

### 2.2.2 Quadrúpedes

Buscando uma evolução em comparação ao modelo anterior, o formato quadrúpede foi criado com a premissa de uma locomoção mais estável em terrenos não uniformes. Esse formato possui uma bio-inspiração nos mamíferos, onde os mesmos conseguem carregar altas cargas e executar múltiplas tarefas sem perda de estabilidade corpórea.

Em busca de melhorias do modelo bípede, pesquisadores do Departamento Mecânico-

Informática na Universidade de Tokyo desenvolveram o primeiro robô quadrúpede, JROB-1 (KAGAMI et al., 1998) para pesquisas em visão e percepção robótica. Um maior desempenho nessa classificação pesquisada teve seu grande salto evolutivo a partir dos anos 2000.

O primeiro aprimoramento desse segmento deu-se com Omata (OMATA; TSUKAGOSHI; MORI, 2002) que sugeriram o uso para execução de tarefas variadas e com maiores complexidades. Dentre as pernas disponíveis, duas seriam usadas para locomoção e outras duas para manipulação de objetos.

O robô quadrúpede para andar em alta velocidade requer uma movimentação dinâmica com menor fator de trabalho, sendo a utilização da marcha um dos padrões de caminhada dinâmica. Nessa configuração de marcha, dois pares de pernas diagonais fazem uma única fase de pé, tendo um controle dianteiro e traseiro. Inagaki e Kobayashi (INAGAKI; KOBAYASHI, 1994) estudaram a locomoção com utilização de marchas e investigaram um método de usar um oscilador de peso construído no centro do corpo para compensar o momento na direção lateral. O método é aplicável se a distância do centro de gravidade ao eixo de suporte for curta. Se a distância for um pouco maior, o efeito da compensação é desfavorável (KURAZUME; YONEDA; HIROSE, 2002). Deve-se destacar também que o método não compensa o momento no sentido longitudinal e parece incompreensível. Assim, o momento em que ocorrer um desbalanceamento no seu eixo de suporte o contrapeso é modelado considerando a direção lateral e a longitudinal, e o ângulo de rotação.

Figura 2.11: JROB-1



Fonte: Filho, Amaral e Pinto (2010)

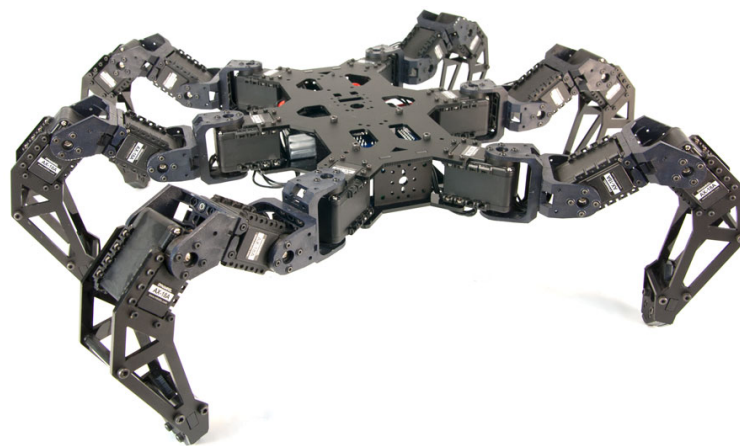
### 2.2.3 Hexápodes

Seguindo o modelo proposto por Omata, soluções similares no conceito hexápode foram propostas por Takahashi ([TADOKORO et al., 2000](#)) e Koyachi ([KOYACHI et al., 2002](#)), onde a composição com seis patas trouxeram maior estabilidade em uma única pata e maior capacidade de carga. Construído para ser um corredor, cada perna possui um único atuador localizado no nível médio do corpo. Outro ponto que favorece sua velocidade e adaptação é o de seu funcionamento contar com a marcha trípole, onde dificuldades como rotação das juntas são eliminadas.

Nessa classificação os robôs são projetados tendo um corpo central e seis pernas, cada uma com dois segmentos e três graus de liberdade, duas nas juntas com a “cintura” e uma conectada ao seu ponto de rotação. O ponto de inércia de cada perna é menor que a inércia do corpo inteiro ([ESPENSCHIED; QUINN, 1994](#)). Devido a isso, cada perna é tratada como um ponto estático de equilíbrio e movimento. Segundo Kenneth as reações que atuam no corpo na articulação do quadril de uma perna que está em seu curso de recuperação (balanço) são muito menores do que as reações na articulação do quadril de uma perna em posição, portanto, podem ser negligenciadas.

Como exemplo dessa classificação pode se citar hexápodes existentes com propriedades comparáveis ao robô PhantomX Mark II hexápode da fabricante Interbotix.

Figura 2.12: PhantomX AX Hexapod Mark II



Fonte: [Robotics \(2020\)](#)

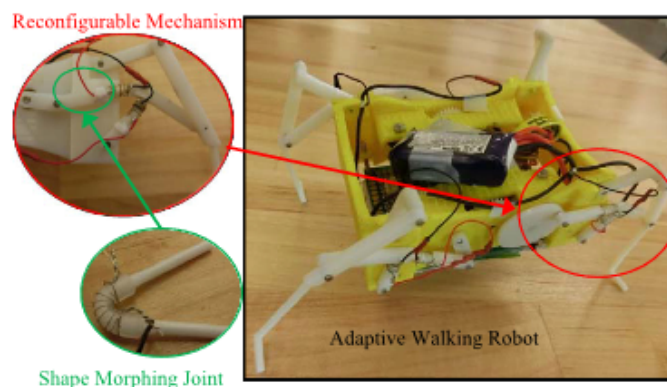
Os robôs hexápodes e suas variações utilizam um único atuador por perna, o que os tornam robustos e ainda permitem que eles executem diferentes marchas para elevação do corpo ou controle da velocidade. Suas articulações operam em forma quase passiva, igual a automóveis, o que aumenta o peso e a resistência da carga útil. Por outro lado, eles possuem uma leve dificuldade em terrenos acidentados devido à ausência de área de contato lateral da ponta dos pés ou com o solo. Robôs como o Vorpal ([VORPAL. . .](#)), que

possui pernas na configuração de inclinação, permitindo o posicionamento na ponta dos pés ao longo das superfícies. A marcha de um robô desse tipo quando executada ocorre um deslizamento na ponta dos pés a cada passada devido à sua geometria; isso resulta em maior dissipação de energia e, mais importante, em odometria muito imprecisa. Por outro lado, Vorpal é um veículo omnidirecional verdadeiro, além disso, com possibilidade de várias modificações, por exemplo, com torre de cabeça rotativa.

Outra classe de hexápodes são aqueles com três graus de liberdade por perna. Dessa forma se amplia a estrutura do tipo Vorpal adicionando junta. Entre esses robôs, foi mencionado o Hexie (THAO, 2018), onde sua construção geral é impulsionada pelo menor custo possível. O robô PhantomX também utiliza três graus de liberdade por perna, sendo essa construção mais flexível graças à grande área em que vários complementos podem ser colocados.

Existem também alguns hexápodes mais exóticos onde durante sua utilização podem mudar sua estrutura física de acordo com o terreno atual (SUN; ZHAO, 2019).

Figura 2.13: Adaptive Robot



Fonte: Sun e Zhao (2019)

### 2.3 Robôs de Modelo Rhex

Com o objetivo de ser uma plataforma simples e robusta para operações em terrenos acidentados (SARANLI; BUEHLER; KODITSCHKEK, 2001), o Rhex faz frente a plataformas como o PhantomX Mark II, exibido na seção 2.2.3. Desenvolvido por membros das Universidades de Michigan e McGill em parceria com a *Defense Advanced Research Projects Agency* (DARPA), o modelo possui apenas um atuador por perna, com isso, diminuindo a sua complexidade mecânica.

Em um *design* simples, composto somente por uma estrutura de alumínio retangular e



seis articulações com posições e orientações fixas, como pode ser visualizado na figura 2.14, o sistema possui 53 *cm* de comprimento e 7 *kg*.

Figura 2.14: Rhex



Fonte: [Saranli, Buehler e Koditschek \(2001\)](#)

Com esses atributos, o Rhex conseguiu ser o robô com seis patas mais rápido no período de seu desenvolvimento. Alcançando até 55 *cm/s* e possuindo uma relação de velocidade por comprimento de 1,04, onde nessa categoria perde somente para o robô Sprawlita ([SARANLI; BUEHLER; KODITSCHKEK, 2001](#)).

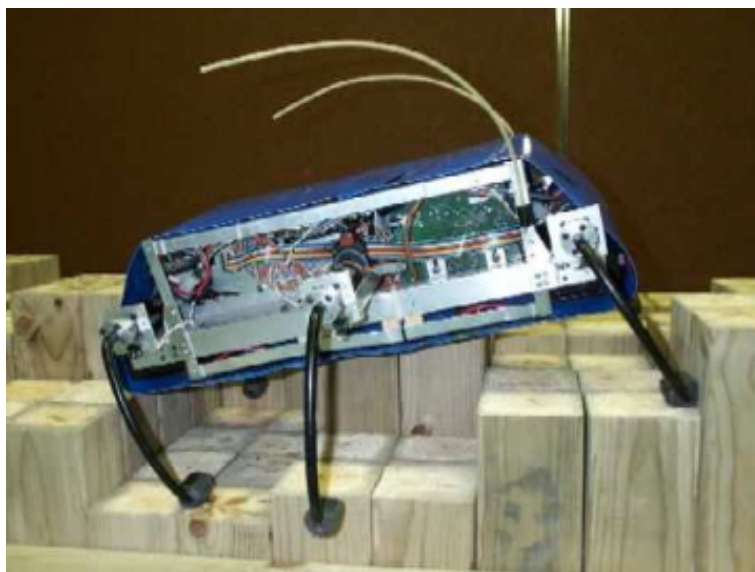
Além disso, surpreendendo seus criadores, esse modelo foi capaz de transpor obstáculos com mais de 20 *cm* de altura mantendo uma velocidade de 42 *cm/s*. A figura 2.15 abaixo exhibe o robô em seu ambiente de testes que possui superfícies com tamanhos aleatórios.

Apesar de suas conquistas, esse robô ainda tinha muito a evoluir. Ao longo dos anos novos modelos foram criados, cada um com um objetivo diferente, porém, sempre aprimorando a plataforma. Nas seções abaixo serão apresentadas algumas dessas variações.

### 2.3.1 *Research-RHex*

Em uma versão mais compacta e mais leve, pesando 6 *kg*, essa variação trouxe importantes melhorias para a plataforma. Atingindo uma velocidade máxima de 2,7 *m/s* e com acréscimo de 30 minutos no tempo de funcionamento, o Research-Rhex foi capaz de subir escadas, girar, pular, andar por tubulações e, além disso, ser apto a se locomover utilizando somente duas pernas ([PRAHACS et al., 2004](#)).

Figura 2.15: Rhex no ambiente de testes



Fonte: [Saranli, Buehler e Koditschek \(2001\)](#)

Contudo, a sua estrutura aberta, apresentada na figura 2.16, demonstrava-se insuficiente para testes em ambientes externos não controlados. Com isso, foi necessária a criação de novos modelos selados com a capacidade de resistir a chuva, poeira e lama.

Figura 2.16: Research-RHex



Fonte: [Prahacs et al. \(2004\)](#)

### 2.3.2 Shelley-RHex

Como uma proposta de proteger os componentes eletrônicos da umidade, sujeiras e impactos, Shelly-RHex substitui a estrutura aberta da antiga versão, o Research-RHex, por uma “concha<sup>2</sup>” completamente fechada, como mostrada pela figura 2.17.

<sup>2</sup>O termo vem da palavra de língua inglesa *Shell*, que origina o nome do robô: Shelley

Figura 2.17: Shelley-RHex



Fonte: [Prahacs et al. \(2004\)](#)

Construída em fibra de carbono, a estrutura possui superfícies completamente curvas por quatro razões principais: melhor distribuição dos impactos, diminuição da força de arrasto hidrodinâmico, reduzir a probabilidade de emperrar em algum obstáculo. Entretanto, a falta de superfícies lisas dificultava a fixação de componentes na parte interna do robô ([PRAHACS et al., 2004](#)).

Além disso, a armação de carbono apesar de atender as expectativas dos desenvolvedores, sendo capaz de nadar na superfície sem a perda de desempenho, abre falhas após repetitivos testes, impossibilitando a locomoção em ambientes aquáticos após isso.

### 2.3.3 *Rugged-RHex*

O Rugged-RHex, exibido na figura 2.18, foi desenvolvido objetivando uma plataforma para utilização comercial e militar. Com isso, alguns pré-requisitos foram estabelecidos, entre eles estão: peso abaixo de 10 *kg*, 2 *kg* de carga útil, 2 *km* de distância percorrida, seis horas de missão e ser resistente a água, ou seja, andar na chuva. Contudo, apesar de não ser requerido, a equipe colocou como objetivo tornar o projeto submersível em profundidade de até 10 metros.

Utilizando as lições aprendidas do projeto do Shelley-RHex (2.3.2), foi repensando a forma de realizar o esboço da carcaça e também seu material, abandonando a utilização de fibra de carbono e adotando o Nylon. Com isso, apesar de não alcançar a meta estabelecida, alcançou 5 *m* de profundidade, se tornando uma das melhores alternativas anfíbias da família RHex ([PRAHACS et al., 2004](#)).

Figura 2.18: Rugged-RHex

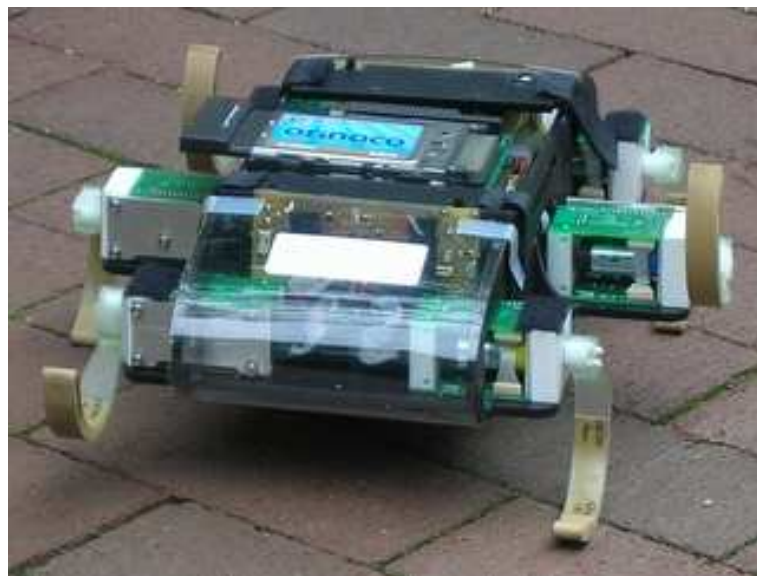


Fonte: [Prahacs et al. \(2004\)](#)

### 2.3.4 Edubot

Edubot, apresentado na figura 2.19, foi projetado com uma idealização diferente das versões anteriores, seu maior foco é criar o interesse nos alunos para o campo das engenharias. Com os números de estudantes diminuindo nos Estados Unidos, enquanto nos outros países aumentando, esse projeto se torna uma ferramenta para alterar essa realidade.

Figura 2.19: Edubot



Fonte: [Weingarten et al. \(2007\)](#)

Por essa razão, a plataforma foi focada em contorna as duas principais razões para a frustração dos projetos de robótica nesse desafio, que são: a falta de modularidade e a incapacidade de demonstrar a real complexidade dos problemas da área ([WEINGAR-](#)



TEN et al., 2007). Com isso, os estudantes conseguem, com o Edubot, alterar tanto os componentes mecânicos, como a sua programação, sendo possível customizar conforme a necessidade.

## 2.4 Marchas de Locomoção para Robôs com Patas

Causada pela grande quantidade de graus de liberdade, expressão relacionada ao número de juntas controláveis de um robô, e pelo critério da estabilidade, os robôs com pernas possuem uma complexidade maior em comparação aos que utilizam rodas, não somente em construção, mas também na eletrônica e no controle (GONZALEZ; SANTOS; ESTREMER, 2006). Uma alternativa utilizada para minimizar esse problema é através da abstração do movimento das pernas por meio do uso de marchas (HAYNES; RIZZI, 2006).

As marchas ou também *gaits*, são padrões de movimento cíclicos que produzem um deslocamento. As diferentes configurações de *Stance* e *Flight*, além dos parâmetros de marcha, proporcionam os distintos *gaits* que um robô possui. *Stance* é o nome empregado quando a perna está em contato com o solo, por outro lado, a condição de *Flight* acontece quando está em voo, ou seja, sem realizar força contra uma superfície.

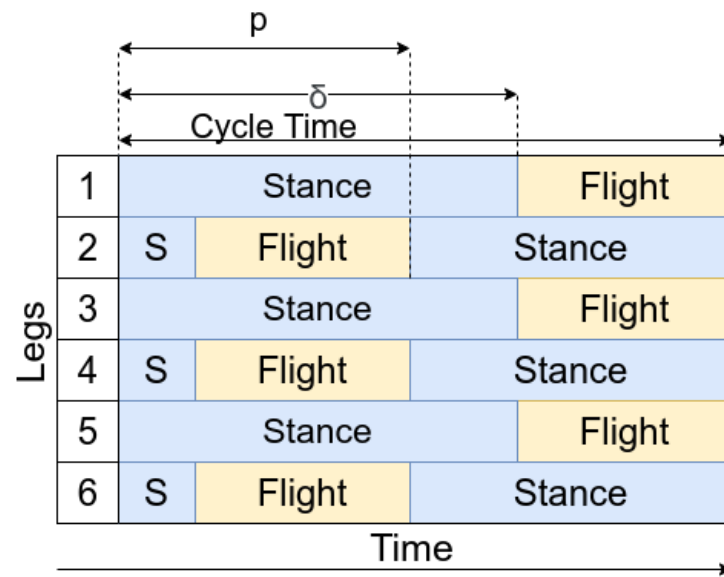
Entretanto, como apresentado por (GONZALEZ; SANTOS; ESTREMER, 2006), o comportamento da marcha é definido por seus parâmetros, que entre eles estão: a fase de *Stance* ( $p$ ), o período de trabalho ( $\delta$ ), e o tempo de ciclo. Na figura 2.20 é possível visualizar uma representação de marcha com essas variáveis explicitadas.

Manipulando esses valores, os robôs conseguem realizar movimentos como andar, subir e descer escadas, pular, correr, entre outros. Devido a isso, nas próximas subseções, será explorado as marchas *tripod* e *stair climbing*, as principais utilizadas em robôs Rhexs.

### 2.4.1 A Marcha Tripod

Inspirado nos movimentos de insetos com seis patas, a marcha *tripod* é a mais aplicada a robôs com essa mesma característica. Esse nome é utilizado devido a sua principal marca, que é o apoio sobre três patas enquanto as outras três estão em voo, como representado na figura 2.20. Para isso, as patas são separadas em dois grupos, onde cada um é composto por duas das extremidades de um lado e uma do meio do outro, como ilustrado na figura 2.21.

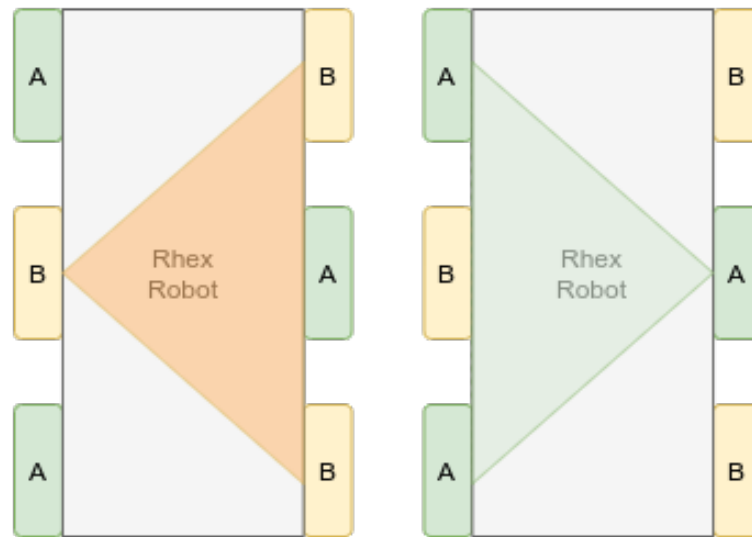
Figura 2.20: Marcha definida por  $p$ ,  $\delta$  e o tempo de ciclo.



Fonte: Própria

Com essa configuração o robô possui o triângulo mais estável para a sustentação nessa marcha, permitindo assim, o seu controle em malha aberta.

Figura 2.21: Exemplo dos triângulos de sustentação.



Fonte: Própria

## 2.5 Framework ROS

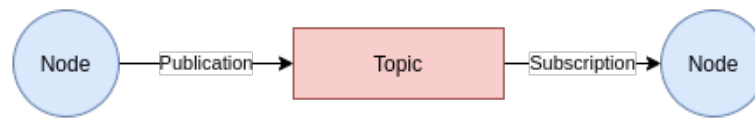
Iniciado no 2007 como uma solução para integração e compartilhamento de códigos e soluções para robótica, o ROS é definido de acordo com (DATTALO, 2018, tradução nossa), como um meta sistema operacional de código aberto para robôs. Através dessa ferramenta é possível reutilizar pacotes disponibilizados pela sua comunidade utilizando a integração que os tópicos e os nós proporcionam.

### 2.5.1 Nós e Tópicos

Os nós e os tópicos são os conceitos básicos, em cima deles são construídas as funcionalidades do ROS. Um nó é qualquer processo que executam alguma computação, como, por exemplo, controle de um motor, identificação de obstáculos ou o cálculo de carga utilizada (AARONMR, 2014). Um robô pode conter dezenas desses nós processando e compartilhando informações.

Para realizar esse compartilhamento é onde entra os tópicos, criados pelos nós no ato de publicar ou se inscrever, eles permitem o envio de mensagens entre um processo e outro. O ato de publicação e inscrição só podem ser realizado pelos nós durante a execução, na publicação o processo envia mensagens para todos os nós inscritos em determinado tópico, a figura 2.22 exemplifica o funcionamento.

Figura 2.22: Relação entre Nós e Tópicos.



Fonte: ROS Wiki ( adaptada )

### 2.5.2 Serviços e Actions

Serviços e *Actions* são ferramentas baseadas no conceito de cliente e servidor, ou seja, um processo solicita a outro uma informação e então recebe essa informação depois que é processada.

Apesar de os nós e os tópicos serem estruturas flexíveis para comunicação, sua natureza de “muitos para muitos” e unidirecional não são apropriadas para uma comunicação de requisição e resposta, essa tarefa é realizada pelos Serviços ([AARONMR, 2014](#)). Para o funcionamento um nó tem que fornecer o serviço e outro processo enviar uma requisição, esse cliente então aguarda até a resposta ser enviada. Essa espera pela resposta é a principal razão pela qual os serviços são utilizados para situações mais simples, que não exigem muito processamento.

Para comandos mais complexos, onde a requisição pode alterar a qualquer momento durante a execução, ou que necessitam de atualizações constantes do progresso é necessário se utilizar os *Actions*. O servidor é um nó que tem como objetivo a execução de uma determinada ação e até ser completada, quando retorna uma resposta de sucesso, envia atualizações para o cliente. Um bom exemplo disso é um servidor para movimentar um robô, o cliente requer um movimento de um metro e então o servidor atualiza o cliente o quanto da etapa foi concluída até finalmente o robô alcançar o valor desejado.

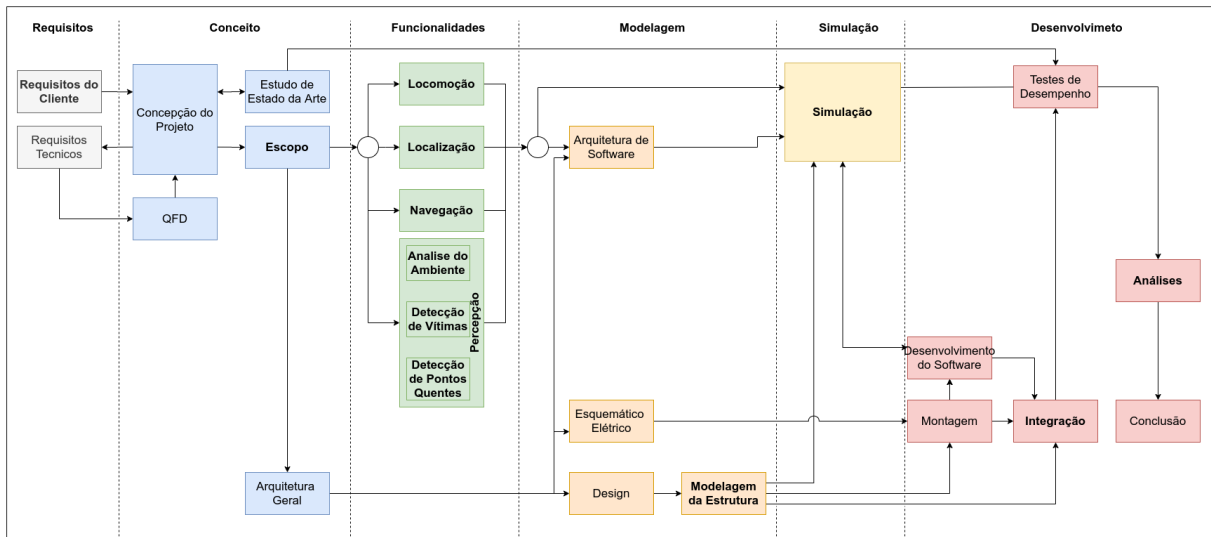
A principal diferença entre os Serviços e os *Actions* é quanto a sincronia com o cliente. Enquanto os Serviços são síncronos, ou seja, o cliente espera a resposta para continuar a sua execução, os *Actions*, por outro lado, são assíncronos, possibilitando a execução do código do lado do cliente enquanto a requisição é processada.



# METODOLOGIA

Concebida através da adaptação do *framework* de gerenciamento de projetos do *Brazilian Institute of Robotics (BIR)*, a metodologia aplicada tem como objetivo idealizar e conduzir o desenvolvimento desse trabalho. Como apresentada na figura 3.1, existem seis seções principais: Requisitos, Conceitual, Funcionalidades, Modelagem, Simulação e Desenvolvimento. Cada segmento será detalhado nos tópicos posteriores.

Figura 3.1: Metodologia aplicada ao desenvolvimento do projeto.



Fonte: Própria

## 3.1 Requisitos

Como o passo inicial, a etapa de Requisitos tem como objetivo a obtenção das necessidades do cliente e dos requerimentos técnicos do projeto. Para o primeiro, são fornecidas pelo cliente o objetivo e o desempenho desejado, os quais são a base para a concepção do projeto, da seção Conceito (3.2). Após essa etapa de idealização, os requerimentos técnicos são levantados, tais como: duração de bateria, capacidade de processamento, número de atuadores, peso, material, entre outros.

## 3.2 *Conceito*

Esta seção da metodologia tem como função transformar os dados obtidos na etapa anterior, Requisitos (3.1), nas ideias que norteiam o desenvolvimento. Essas informações adquiridas do cliente passa para o bloco de concepção do projeto onde as propostas para solução do problema são formuladas e estruturadas. Para auxiliar na realização dessa tarefa existem dois laços de interações, estruturas essas que possuem como característica a realimentação entre as etapas. A primeira acontece quando após a formulação das ideias são obtidos os requerimentos técnicos e então é retornado ao conceitual para a geração da matriz *Quality Function Deployment* (QFD), a qual é utilizada para reestruturar a concepção do projeto. Já a segunda, acontece na interação com o *State of the Art* (SOTA), onde é estudado os projetos com finalidades similares no intuito de entender o que e como foram construídos, com isso, é realizado novamente a reestruturação das ideias.

Após a conclusão da concepção do projeto, é iniciada a etapa Escopo. Nesse bloco é documentado tudo que será desenvolvido e entregue para o cliente, sendo assim possível especificar as funcionalidades e realizar a criação da Arquitetura Geral, que tem como resultado uma visão macro, porém simplificada da solução a ser desenvolvida.

## 3.3 *Funcionalidades*

As funcionalidades existentes no robô são atividades e processos desempenhados que garante que a solução, desenvolvida na seção anterior (3.2), seja executada da forma como foi idealizada. Para isso, nesta etapa é definido cada um desses elementos explicitando suas dependências, suas entradas e saídas, seu funcionamento e objetivo.

## 3.4 *Modelagem*

As atividades desenvolvidas na etapa Modelagem possuem como finalidade moldar a forma como às três principais áreas de conhecimento do projeto serão realizadas. A arquitetura de *Software* exibem as camadas de abstração e aplicação, assim como as relações entre os programas e pacotes. O esquemático elétrico expõe as ligações elétricas, tanto as de potência, responsáveis pela alimentação dos circuitos e motores, como as de informação, encarregadas de transportar os dados de comandos e sensoramento. Por fim, o *Design* e a Modelagem da Estrutura são os incumbidos do planejamento da carcaça e armação do robô, onde é descrito a posição, forma e material de cada elemento.

### **3.5 Simulação**

Uma simulação contém ambientes e robôs virtuais que auxiliam no desenvolvimento de projetos, principalmente no *software*. Essa ferramenta permite executar os algoritmos criados sem a necessidade de aquisição, conexão e montagem do robô, o que torna o processo de testes mais rápido, mais barato e até mesmo mais seguro. Com essas verificações dos programas é possível encontrar falhas e falta de desempenho com maior agilidade.

Para isso, a etapa Simulação é responsável pela criação virtual de todos os recintos no qual o robô irá atuar, além disso, é feita a utilização do modelo mecânico, desenvolvido na seção 3.4, para simular a física da plataforma com uma maior semelhança com o real. Por consequência, é possível ao final do projeto, comparar os desempenhos entre os dois modelos.

### **3.6 Desenvolvimento**

Com todo o projeto idealizado e planejado, é na etapa Desenvolvimento que acontece a criação dos pacotes, possuindo como finalidade reproduzir as funcionalidades discutidas na etapa da seção 3.3. Logo após, é realizada a integração entre a simulação e os pacotes das funcionalidades.

Por fim, com a simulação completa, é iniciada as etapas finais do projeto: testes, análises e conclusões. Nos testes é recolhido informações da execução das funcionalidades do robô, que então são comparadas com as versões estudadas no SOTA e com as encontradas nas simulações. Com isso, é feito as análises do resultado do trabalho e finalmente o projeto é concluído.

---

## RESULTADOS E ANÁLISES

---

### 4.1 Modelo

A plataforma robótica é dividida em 3 subsistemas principais: Eletrônica, Mecânica e Dispositivos Inteligentes (*Smart Devices*), que são integrados ao robô em duas *housings*<sup>1</sup>: *Upside Housing* e *Downside Housing*. Nessa seção, será abordado a concepção, *design* e integração de cada subsistema, bem como sua incorporação integral no robô.

#### 4.1.1 Eletrônica

A eletrônica do robô concentra todos os dispositivos elétricos do sistema, gerenciando e monitorando potência, e permitindo controle e medição de dados. Foi projetada de forma a manter o robô funcionando de forma segura e consistente, tendo embarcado dispositivos de proteção e monitoramento geral do sistema.

Para a alimentação geral do sistema se fez necessário a utilização de 2 Baterias Inteligentes de 14.4V, 6200mAh cada, ligadas a uma placa de distribuição, que paraleliza as cargas, dando uma capacidade total ao sistema de 12400mAh. Com isso, é possível alcançar cerca de uma hora de autonomia utilizando a plataforma em sua melhor *performance*, de acordo com o Diagrama I C.

As cargas do sistema são agrupadas em 3 conversores do tipo abaixador, que limitam às faixas de tensão em duas faixas de 12V e uma de 5V. No Diagrama 1, é detalhado a distribuição das cargas, bem como suas características elétricas de consumo e energização. O diagrama também ilustra as cargas atreladas a cada componente, bem como o modelo de comunicação utilizado em cada situação, como USB ou RS-485, por exemplo. De modo a não sobrecarregar a CPU, alguns dispositivos USBs foram alimentados diretamente pela bateria, mantendo-se apenas as conexões de sinais fixas no USB. O diagrama ainda separa as categorias pela cor, categorizando parte inteligente, parte de potência, sensoriamento e iluminação.

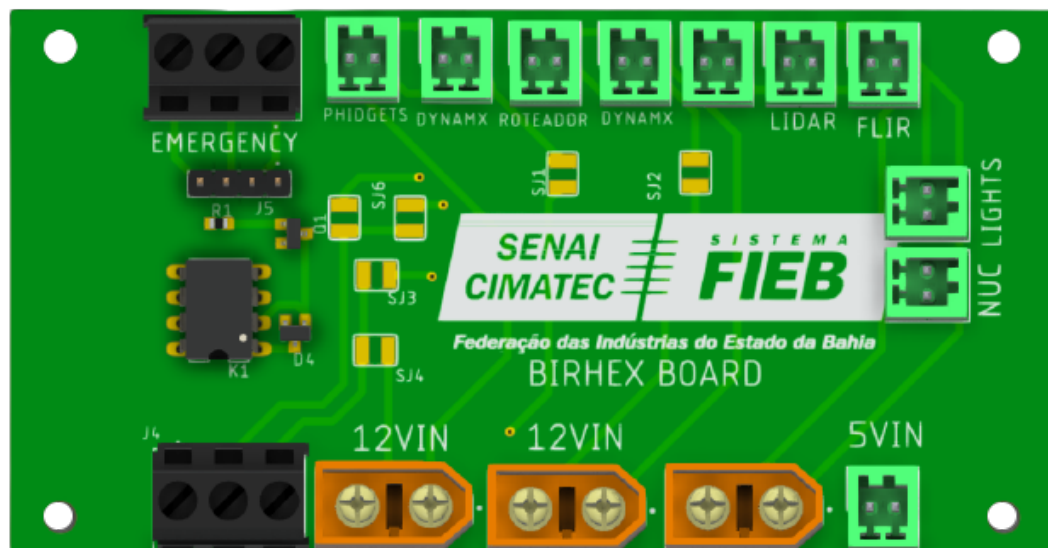
No Diagrama II, disponível no anexo B, é definido todas as ligações do sistema de potência do robô. Para alimentar o robô de forma segura, organizada e simples, projetou-se a

---

<sup>1</sup>Compartimento que agrupa um conjunto de componentes num protótipo

BIRHex Board (Figura 4.1), uma placa de circuito que concentra todas as ligações de potência do robô e distribui energia para todas as cargas do sistema. Projetou-se também a BIRHex Fuse Board (Figura 4.2), uma placa conectada à saída da BIRhex Board, que concentra apenas os fusíveis e é fixada na chapa superior da *Upside Housing*, a fim de fornecer fácil acesso e troca dos fusíveis. A BIRhex Board é composta por conectores de entrada, saída e uma chave relé conectada ao botão de emergência, que quando acionado, desativa eletronicamente todas as cargas ligadas ao sistema de movimentação do robô.

Figura 4.1: BIRHex Board

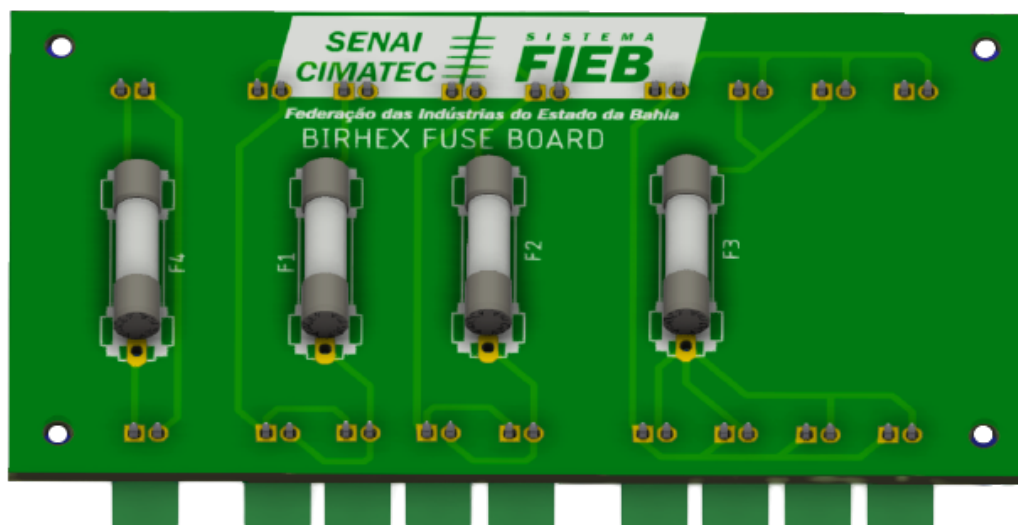


Fonte: Própria

De forma a facilitar a ligação dos componentes do robô, colocou-se legendas (Figura 4.1) em todos os conectores de entrada e saída presentes na placa. Tal organização ainda facilita na montagem e desmontagem da plataforma, bem como em todo o processo de manutenção do robô.

Com o intuito de simplificar o sistema de testes de potência do robô durante a montagem, projetou-se a *BIRhex Test Board*, uma placa de circuito integrada já com fusíveis, conectores de entrada, saída e um circuito de relé de estado sólido embutido. Além de ser uma placa relativamente maior do que a BIRHex Board, sua principal diferença se encontra na montagem e fabricação: a placa de teste possui apenas uma camada e simples *design*, sendo possível realizar sua fabricação à mão, sem a necessidade de terceirização de serviço. Com ela é possível ligar todo o sistema do robô e operar normalmente, sem restrições funcionais. Porém, operar com a mesma gera restrições a nível mecânico, como a dificuldade de acesso aos fusíveis, uma vez que a mesma precisa ser fixada no interior da *Downside Housing* para simplificar o processo de ligação dos componentes. A placa ainda possibilita manter o botão eletrônico e ligações de algumas cargas como uma ferramenta

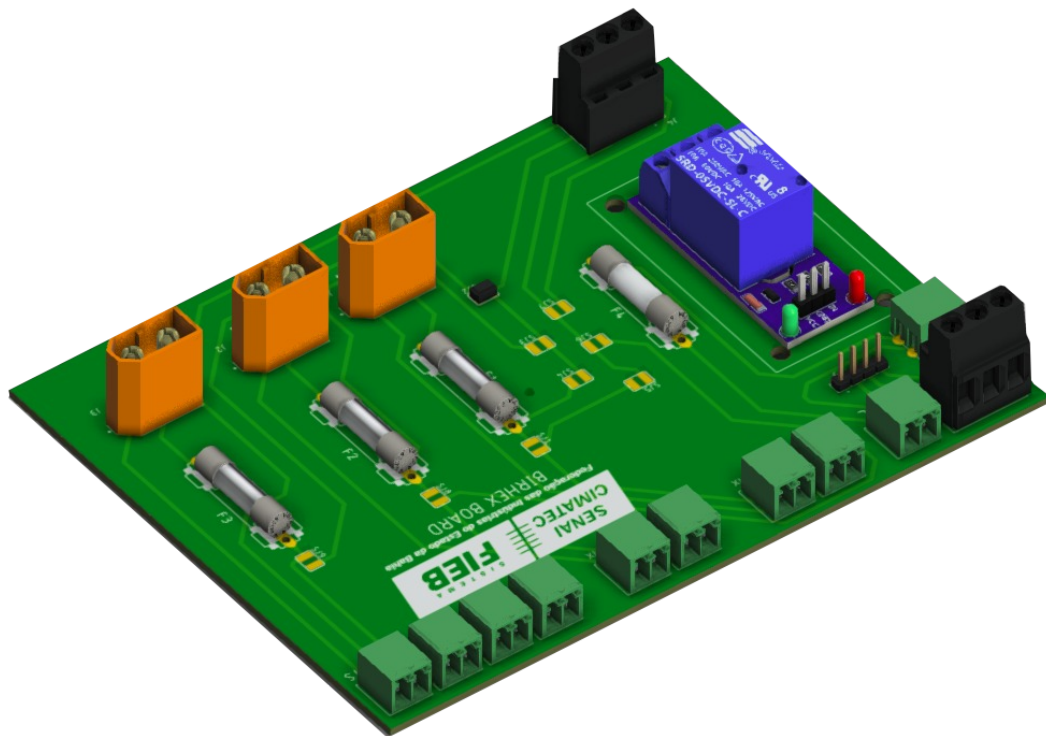
Figura 4.2: BIRHex Fuse Board



Fonte: Própria

opcional, através dos circuitos *solder jump*.

Figura 4.3: BIRHex Test Board



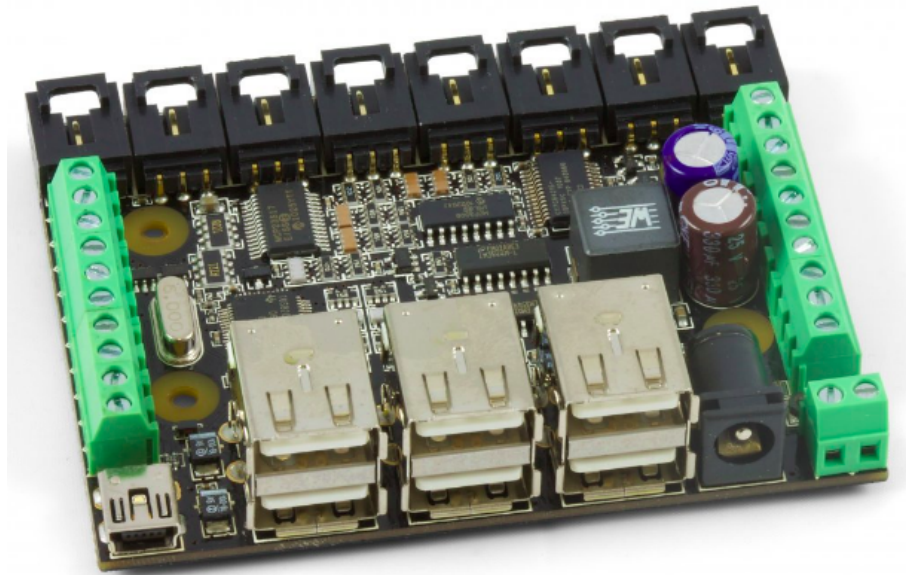
Fonte: Própria

### 4.1.2 Sensoriamento e Smart Devices

Nesse tópico será abordado o sistema de sensoriamento do robô e citar todos os dispositivos inteligentes existentes no protótipo. Mais detalhes de cada dispositivo, bem como suas aplicações e resultados nos testes em simulação, serão abordados em seções futuras.

Todo o sistema de sensoriamento do robô, que consta com sensores de tensão, corrente e temperatura são ligados na *Phidgets Interface*, figura 4.4, uma placa compacta, equipada com microcontrolador, que faz leitura e tratamento dos dados dos sensores e acionamento de dispositivos como *coolers* e botão eletrônico de emergência. A escolha da *Phidgets* se deu devido à fácil integração da plataforma com o ROS, bem como uma estrutura de *hardware* bem organizada sob uma *Placa de Circuito Impresso* (PCI) compacta e equipada com múltiplos conectores bem distribuídos e de fácil manuseio. O ponto de monitoramento mais crítico pelo qual a *Phidgets* é responsável são os Conversores DCDC embarcados no robô. Ao todo, existem três conversores, que alimentam todo o robô, e se faz necessário monitorar corrente, tensão e temperatura desses componentes, de forma a evitar que possíveis surtos de tensão, corrente ou superaquecimento venham a danificar algum dispositivo. Através desse monitoramento, a *Phidgets* é capaz de acionar o botão eletrônico de emergência de forma automática.

Figura 4.4: Phidgets



Fonte: [Phidgets](#) (2020)

Dentre os *Smart Devices*, têm-se como destaque os servomotores inteligentes Dynamixel MX-106 4.5. Foram especificados para esse projeto devido a seu alto torque de até 10N.m, sua pequena dimensão, e por possuir uma manuseabilidade simples e compatível com o *Framework* ROS. Além disso, os motores possuem sensoriamento interno de alta precisão, onde se pode ter *feedback* de tensão, corrente, temperatura, rotação, posição e velocidade do motor. Com tais dados é possível realizar monitoramento em tempo real dos dispositivos, bem como calcular a posição aproximada do robô à medida em que se move. Em contrapartida, os Dynamixel possuem uma rotação relativamente baixa para suprir a velocidade exigida por um robô hexápode, porém retorna um desempenho que satisfaz as exigências do cliente.



Figura 4.5: Dynamixel



Fonte: [Dynamixel \(2020\)](#)

O sistema de visão é integrado por 2 câmeras, uma é estéreo e a outra é infravermelho, além de um LIDAR. A câmera Estéreo Zed 4.7 é responsável pela odometria virtual, enquanto a câmera infravermelho FLIR 4.8, será direcionada a aplicações em ambientes mal iluminados, bem como para a detecção de superfícies quentes em ambientes a serem inspecionados. O LIDAR 4.6 tem como objetivo gerar uma nuvem de pontos para mapear o ambiente a ser inspecionado e auxiliar no deslocamento do robô.

Figura 4.6: RPLidar



Fonte: [slamtec \(2020\)](#)

Figura 4.7: Stereo Zed



Fonte: [StereoLabs \(2020\)](#)

Figura 4.8: FLIR



Fonte: [FLIR \(2020\)](#)

O sistema ainda possui integrado um sensor IMU xSens 4.10 e um GPS PIKSI 4.9. Ambos dispositivos possuem integração pronta com o ROS e oferecem alta precisão de medições em tempo real. A disposição de tais informações possibilita ao robô, realizar uma fusão de sensores que torna seu deslocamento e autonomia para tomada de decisões no ambiente, muito mais precisa. Resultados satisfatórios desse processo são demonstrados na simulação do Gazebo e são abordadas da Seção 4.2.4.2.

Figura 4.9: GPS PIKSI

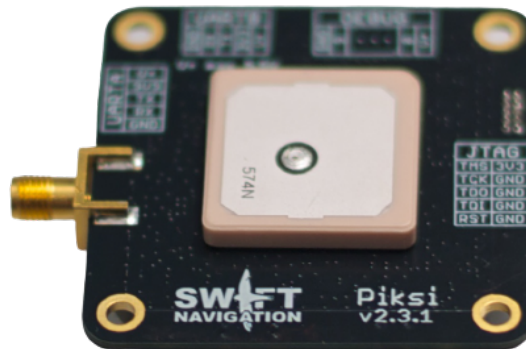
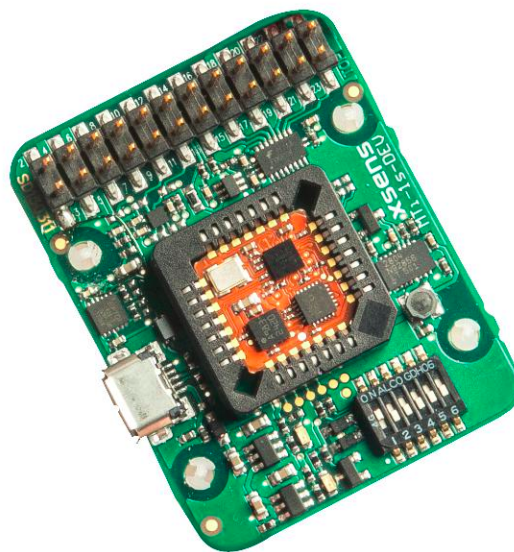
Fonte: [Navigation \(2020\)](#)

Figura 4.10: IMU xSens

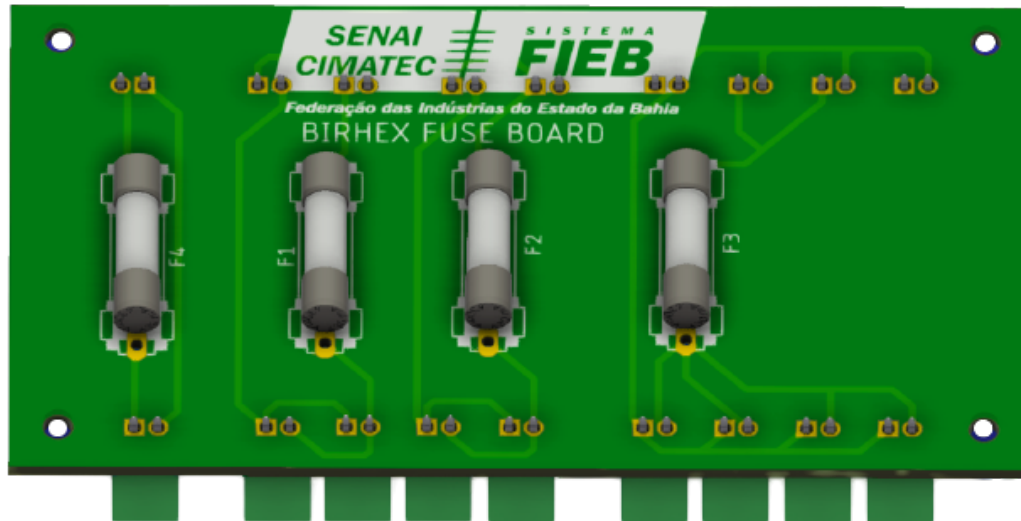
Fonte: [xSens \(2020\)](#)

#### 4.1.2.1 Sistema de Segurança

O sistema geral do robô possui uma proteção em três camadas: um em *hardware* e dois em *software*. A proteção em *hardware* acontece através dos fusíveis, incorporados na BIRhex Fuse Board, que se encontra fixa na tampa superior da *Upside Housing* 4.11. É possível realizar a troca dos fusíveis de forma rápida e simples, apenas retirando a tampa

de proteção, removendo o fusível desejado e inserindo um novo. A imagem abaixo ilustra a posição da *BIRhex Fuse Board* no robô.

Figura 4.11: BIRhex Fuse Board no robô



Fonte: Própria

As proteções em *software* são definidas em 2 estados: manual e automático, que se mantêm ligadas simultaneamente. A proteção automática atua através da Phidgets Interface, que monitora corrente, tensão e temperatura de todos os Conversores DCDC presentes no robô e é capaz de desativar automaticamente, através de um relé de estado sólido, as principais cargas do robô quando detectado alguma anormalidade nesses 3 pontos. A proteção manual se dá através do botão de emergência 4.12, que quando acionado, envia a uma entrada digital da Phidgets um degrau, que aciona o relé de estado sólido, desativando as principais cargas do robô.

O robô ainda possui alguns modos de segurança a serem programados em *software*, como o estado de economia de energia, que o possibilita suspender energia dos motores quando a bateria atinge nível crítico. Dessa forma, o robô mantém os principais sensores ligados, a fim de fornecer ao operador visual em comando a sua localização exata por mais tempo. Outro modo de segurança a ser inserido no sistema se dá através do monitoramento individual dos motores Dynamixel, que possuem sensoramento interno próprio. Caso algum dos motores apresente algum surto ou evidência de exaustão, se inicia o modo de descanso, onde o robô envia um alerta ao operador visual da condição atual e inicia uma contagem de 2 minutos até entrar em modo de suspensão temporária dos motores, onde se intensifica a atuação do sistema de refrigeração e se monitora os motores até que o sistema retorne a um nível seguro de trabalho.

Figura 4.12: Botão de Emergência



Fonte: Própria

### 4.1.3 Mecânica

Nessa seção será abordada as características construtivas e de posicionamento dos componentes internos e externos no protótipo BIRhex. Todos os componentes internos do robô são divididos em duas *housings*: *Upside Housing* e *Downside Housing*. Será comentado sobre os componentes externos do robô, bem como sobre cada *Housing* individualmente.

Na parte externa do robô estão os 6 conjuntos de patas, o LIDAR, a *BIRhex Fuse Board* e as tampas laterais, frontal e traseira do protótipo. Um conjunto de pata, é formado por: Suporte de fixação, motor e pata, impressa em 3D com PLA, como demonstrado na figura 4.13.

Para modelar as patas, utilizou-se como parâmetro as dimensões de altura e comprimento do robô, onde a altura determina o tamanho mínimo e o comprimento limita o tamanho máximo do componente. Dessa forma, foi modelado uma pata com 150mm de diâmetro e 5mm de espessura.

Figura 4.13: Conjunto de pata



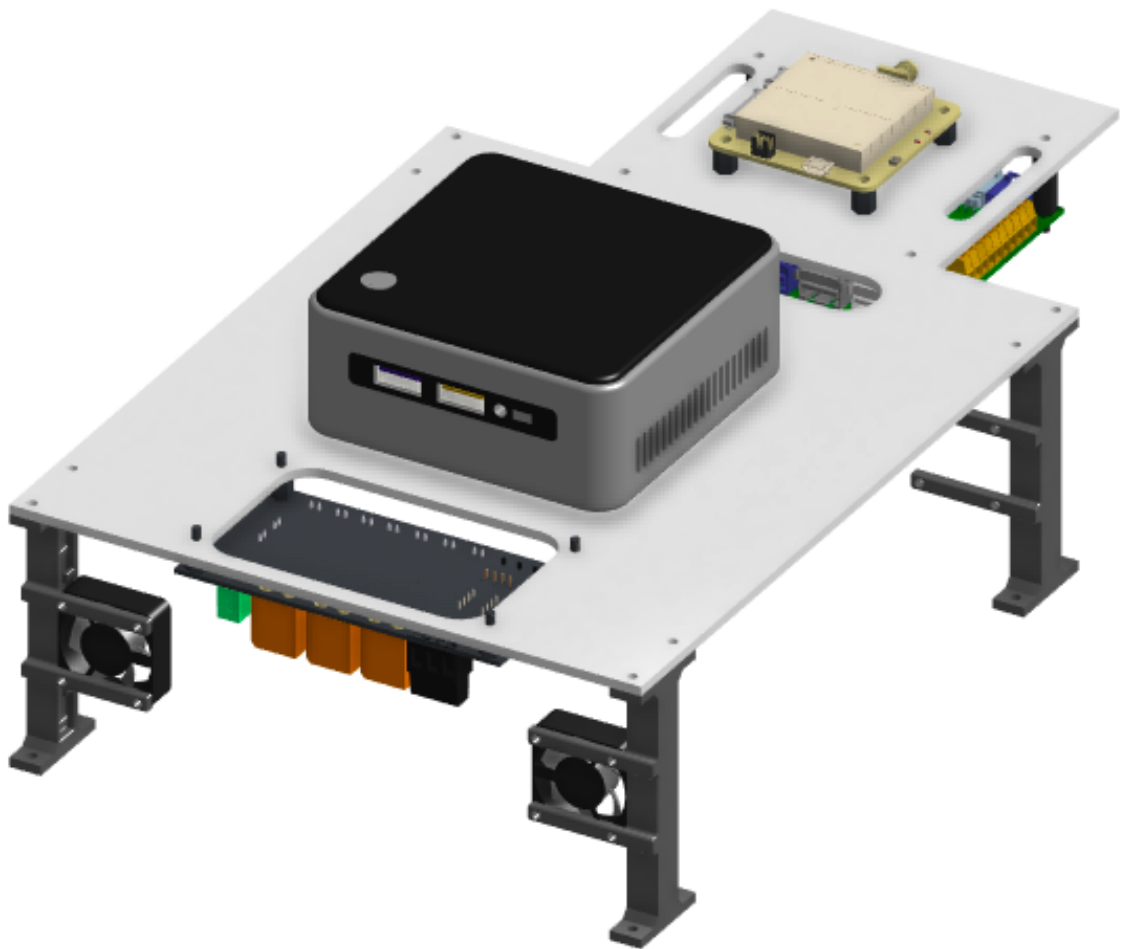
Fonte: Própria

#### 4.1.3.1 *Upside Housing*

No interior dessa *Housing*, na parte inferior, têm-se os *coolers*, que fazem refrigeração da *Downside Housing*, a *BIRhex board*, o *IMU* e a *Board Phidgets Interface*. As duas placas estão fixadas na tampa inferior da *Upside Housing* com o intuito de fazer uma conexão com a *Downside Housing*, que promove a alimentação para todo o protótipo.

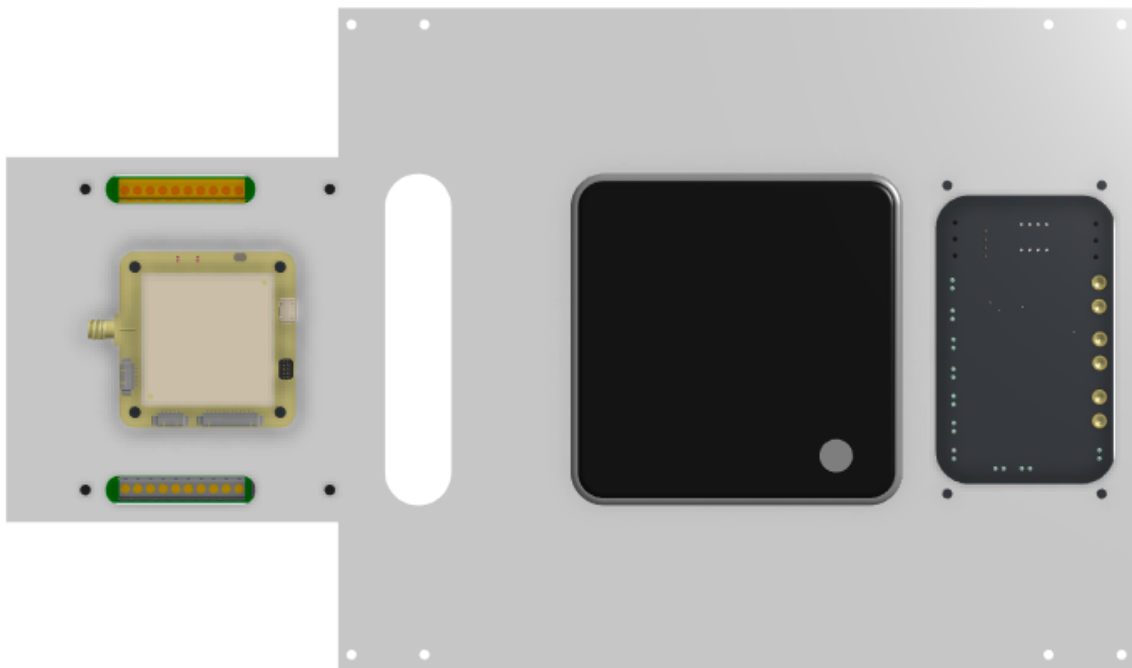
Na parte superior da *Upside Housing*, têm-se o *GPS PIKSI* e a *Nuke*, que estão posicionadas no lado superior da tampa central da *Housing* com o intuito de isolar tais componentes da parte de potência do robô. A principal atribuição dessa *Housing* é a de distribuição de energia e comunicação do robô, visto que se concentra nela a CPU, microcontrolador e a placa de distribuição de energia. Nas imagens [4.14](#), [4.15](#), [4.16](#) ilustramos as características construtivas dessa *Housing*.

Figura 4.14: Upside Housing



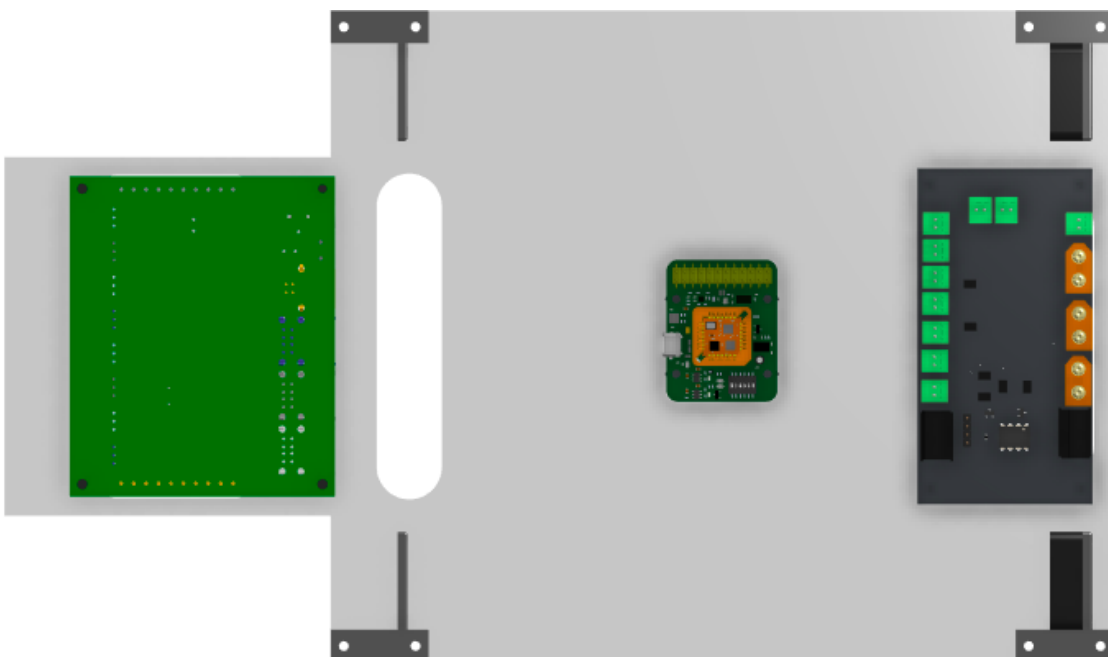
Fonte: Própria

Figura 4.15: Upside Housing Top



Fonte: Própria

Figura 4.16: Upside Housing Bottom



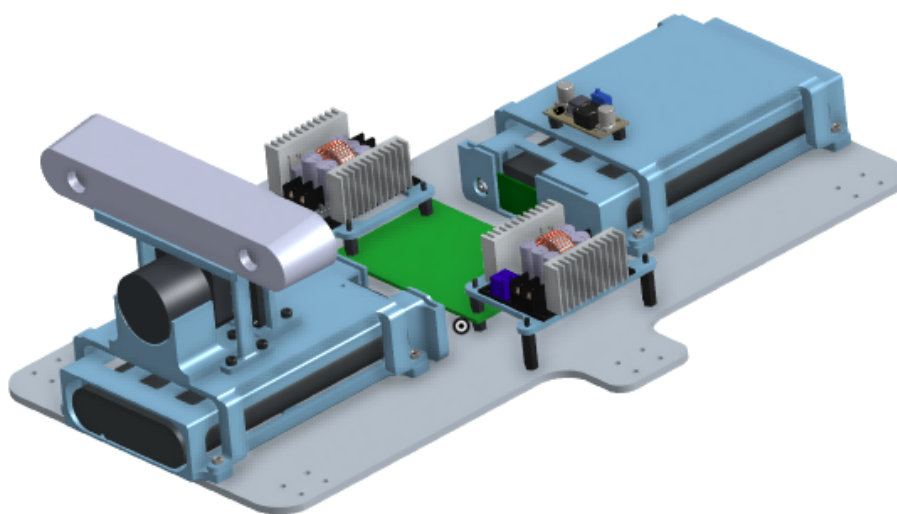
Fonte: Própria



### 4.1.3.2 *Downside Housing*

No interior dessa *housing*, se concentra todo o sistema de potência e as câmeras do robô. Por se encontrar na base do robô, uma característica muito importante dessa *housing* é a simetria de distribuição dos componentes fixos na tampa inferior. Tal simetria promove uma melhor distribuição de carga ao longo do protótipo garantindo melhor estabilidade nos movimentos do robô. Essa *housing* possui apenas uma câmera térmica e uma câmera estéreo, duas baterias, três conversores DCDC e uma placa de paralelização para as baterias, conforme imagem abaixo.

Figura 4.17: Downside Housing



Fonte: Própria

### 4.1.3.3 *Modelo Completo*

A montagem completa do BiRhex é a união das *housings*, parte externa do robô e as tampas laterais, frontal e traseira. Essa montagem possui peso de aproximadamente 8kg e dimensões de 400x220x140 mm, de comprimento, largura e altura respectivamente. Alcança uma velocidade de 15 cm/s e possui a cor amarela, com intuito de ser facilmente visível em qualquer ambiente. As imagens [4.16](#), [4.17](#), [4.18](#), [4.19](#) e [4.20](#) ilustram o modelo renderizado da montagem completa do BiRhex.

Figura 4.18: Montagem Completa



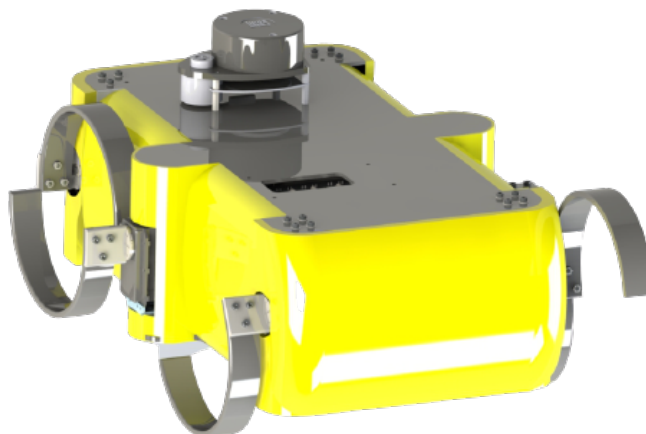
Fonte: Própria

Figura 4.19: Visão Superior da Montagem Completa



Fonte: Própria

Figura 4.20: Visão Traseira da Montagem Completa



Fonte: Própria

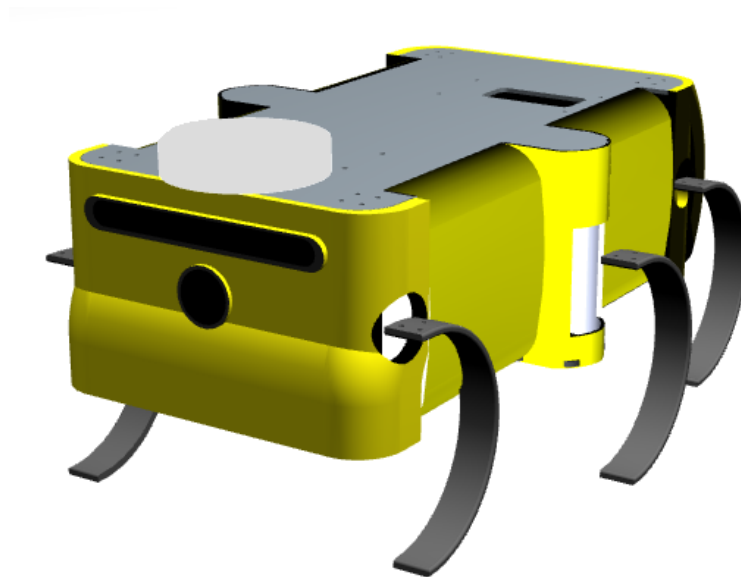
## 4.2 Simulação

Nessa seção será mostrado como a simulação do projeto, etapa descrita na seção 3.5, foi realizada. Para isso, o será esmiuçado o modelo do robô utilizado e os seus *plug-ins*: câmera, LIDAR, IMU, entre outros. Além disso, também será detalhado como ambiente onde o robô será simulado foi construído e quais seus elementos.

### 4.2.1 Modelo de Simulação do Robô

A seção 4.1 detalha a modelagem mecânica do robô, que foi exibida na figura 4.18, contudo, esse modelo não se mostrou eficiente para a simulação devido ao custo computacional para sua execução. Por essa razão, foi desenvolvida sua forma simplificada retirando os elementos visuais que não diminuam a fidelidade da simulação e configurando como estrutura de colisão elementos geométricos básicos. A figura 4.21 mostra a versão simplificada do BIRHex para a simulação.

Figura 4.21: Modelo Simplificado.



Fonte: Própria

Além do modelo do corpo, é necessária a implementação dos *plugins* dos sensores, código que implementa o comportamento do sensor. A câmera estéreo como mostrado na seção 4.1.2, tem como objetivo a aquisição de duas imagens ao mesmo tempo, e para isso se faz uso do *plugin libgazebo ros multicamera* para conseguir captar as imagens dentro do ambiente simulado, um exemplo da imagem gerada pelo programa pode ser vista na figura 4.22.

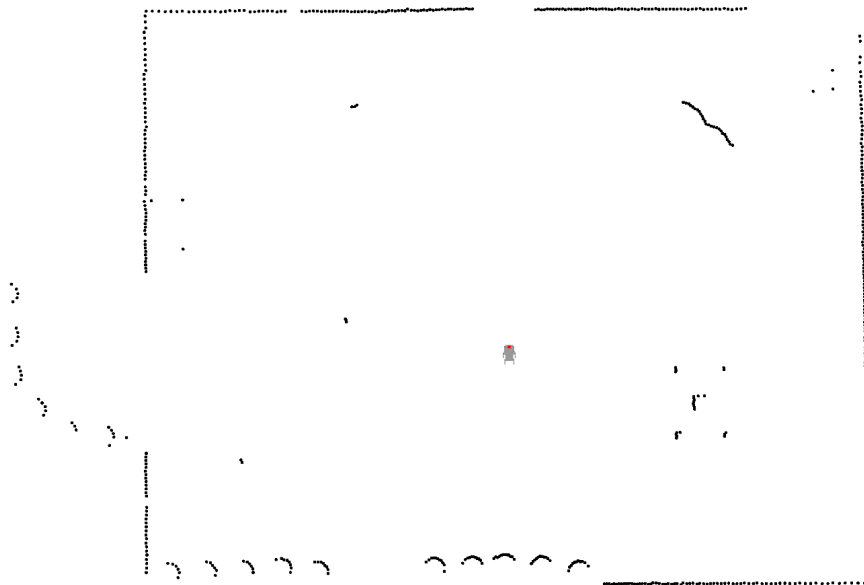
Figura 4.22: Funcionamento Câmera Estéreo.



Fonte: Própria

Para fazer a virtualização do LIDAR foi utilizado o *plugin libgazebo ros gpu laser*, e seu funcionamento se dar enviando a distância do sensor até um objeto na cena em diferentes ângulos. Para fazer a demonstração da figura 4.23, foi utilizado o mapa da figura 4.24.

Figura 4.23: Funcionamento LIDAR.



Fonte: Própria

O processo de simulação do IMU e GPS foi realizado utilizando as informações da simulação, velocidade angular, posição e aceleração, e acrescentando um desvio padrão a esse valor.

### 4.2.2 Ambiente

Para executar a simulação é necessário inserir o robô em um ambiente virtual. Para este projeto foi desenvolvido o mundo representados na figura 4.24. Esse modelo foi nomeado de *Park* ou Parque e tem como objetivo representar uma situação real, na qual, o BiRhex estará inserido com outros elementos no mundo como objetos, pessoas, plantas e paredes.

Figura 4.24: Ambiente Virtual do BiRhex



Fonte: Própria

Para simular uma situação com vítimas, uma pessoa foi posicionada atrás de alguns escombros, onde para o robô alcançar ele deve passar por um pequeno “túnel” ou por um caminho com obstáculos, esses obstáculos foram representados por dois troncos deitados, pedras e latas de bebidas para criar desnivelamento no solo, elementos exibidos nas figuras 4.25 e 4.26.

Figura 4.25: Vítima e Túnel do Parque



Fonte: Própria



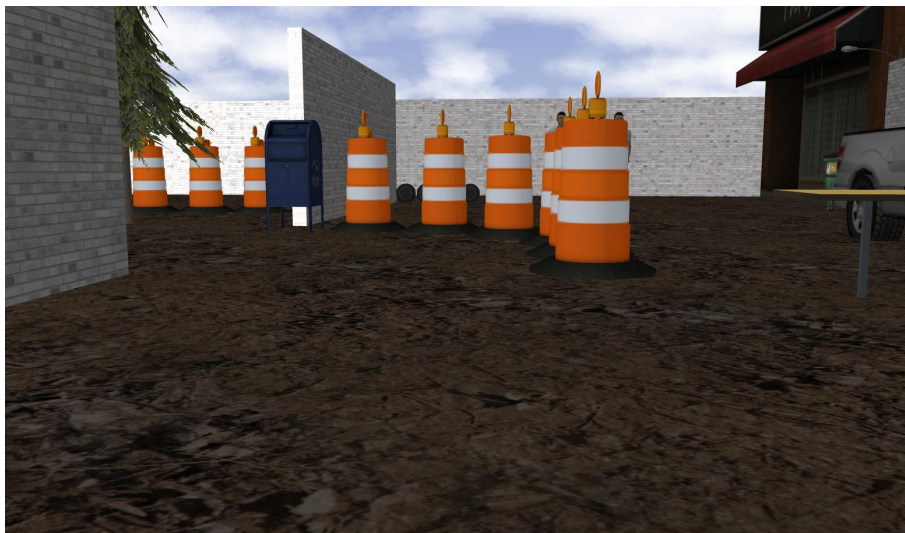
Figura 4.26: Obstáculos do Parque



Fonte: Própria

Para fazer melhor uso dos pacotes de navegação foi inserido no mapa bifurcações e caminhos sem saída. Com esses elementos o robô possui um desafio maior na hora de traçar a sua rota até determinado objetivo, sendo obrigado a selecionar um dos caminhos ou retornar, um exemplo de bifurcação e caminho sem saída está na figura 4.27 e 4.28.

Figura 4.27: Bifurcações do Parque

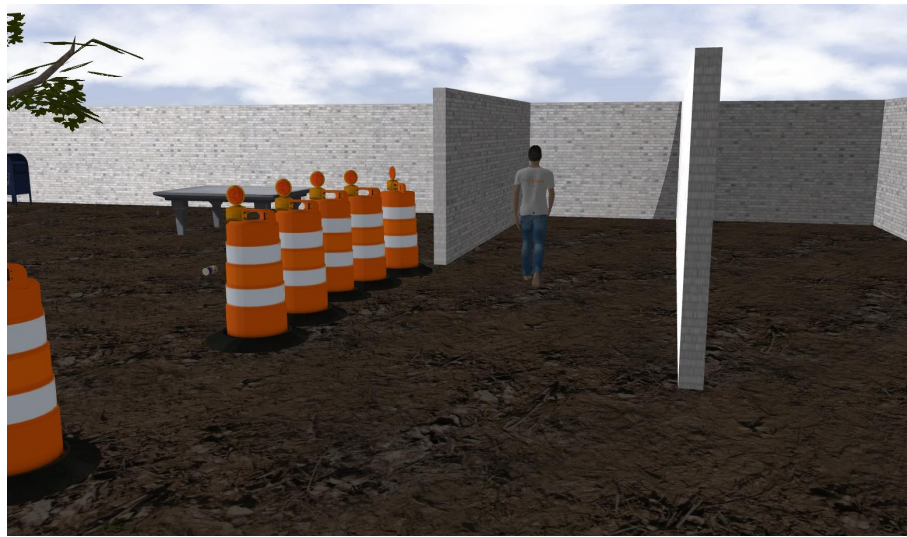


Fonte: Própria

### 4.2.3 Estrutura de Software

O ROS foi utilizado como base do projeto da arquitetura de *software* do BiRhex, a capacidade de reutilização de códigos, pacotes e procedimentos, além do suporte a diferentes

Figura 4.28: Caminhos sem Saída do Parque



Fonte: Própria

linguagens e ambientes para a simulação foram os pontos principais para essa escolha. Devido a isso, as funcionalidades do robô foram implementados em diferentes pacotes que se complementam durante sua execução, o nome e a descrição de cada pacote está resumida abaixo.

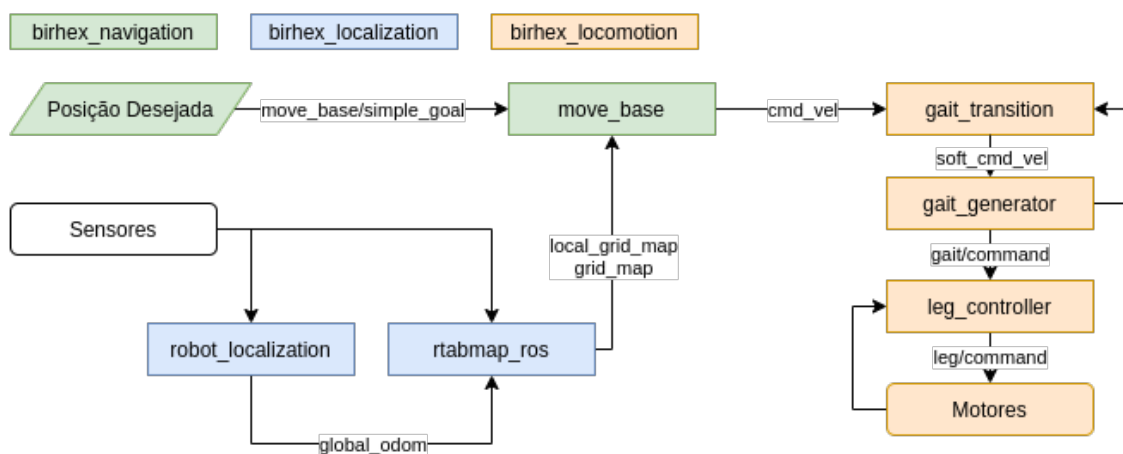
- **birhex**: Este é um *meta-package*<sup>2</sup> que tem como dependência todos os outros pacotes, permitindo assim criar uma dependência de todos os pacotes somente com um;
- **birhex\_description**: Todas as informações do robô necessárias para o funcionamento dos algoritmos ou das simulações, algumas delas apresentadas na seção 4.2.1, são implementadas nesse pacote;
- **birhex\_gazebo**: Os ambientes apresentados na seção 4.2.2 são armazenados nesse pacote e, além disso, possui os *launchers* para a inicialização da simulação no Gazebo;
- **birhex\_ihm**: Qualquer ferramenta para a interação entre um operador e o BiRhex deve ser implementada dentro deste pacote, até o momento da escrita deste trabalho somente a teleoperação do robô está implementada;
- **birhex\_localization**: Contém os arquivos de configuração e iniciação da localização;
- **birhex\_locomotion**: Os arquivos de marcha, como a trípode, são desenvolvidas nesse pacote;

<sup>2</sup>Um *meta-package* é um pacote especial que não instala itens, não possui códigos ou quaisquer outros arquivos se não os arquivos de manifesto.

- **birhex\_msgs**: Contém os arquivos para a geração das mensagens customizadas utilizadas em todos os outros pacotes do robô;
- **birhex\_navigation**: Contém os arquivos de configuração e iniciação da navegação.

O BiRhex possui navegação autônoma, ou seja, ele pode ir a um destino sem a interferência de um operador. Para isso, como descrito acima, os pacotes funcionam em conjunto para atingir esse objetivo, a organização e o fluxo do programa podem ser visualizados na figura 4.29 abaixo. Nessa arquitetura o sistema espera como entrada do operador uma posição que o robô deve estar, esta entrada é então recebida pelo planejador de trajetórias que utiliza como entradas os dados de localização e mapeamento dos sensores para enviar o comando para o pacote de locomoção, nesse pacote é recebido um comando de velocidade linear e angular que é transformado em velocidade angular para os motores de cada perna.

Figura 4.29: Fluxo de Software Autônomo.

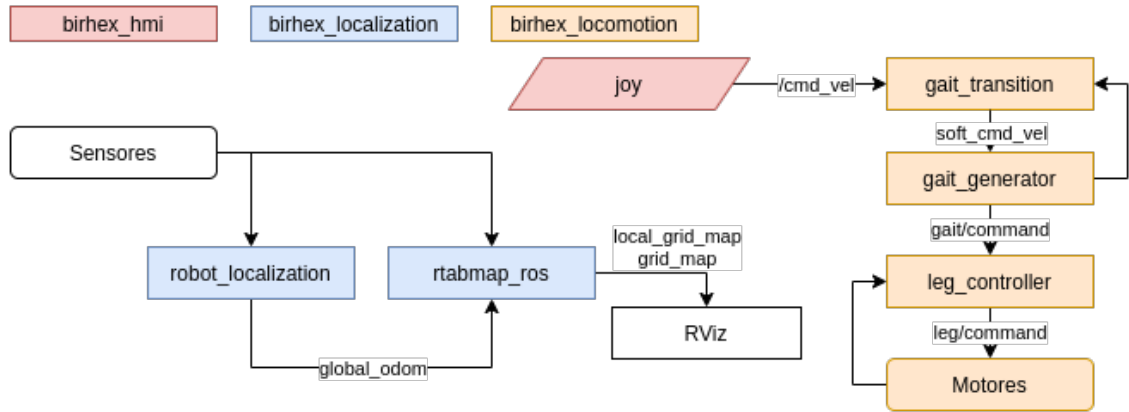


Fonte: Própria

Para o funcionamento teleoperado do robô, algumas mudanças são necessárias, o diagrama modificado exibido na figura 4.30 explicita que a principal alteração acontece na entrada do sistema, nessa versão o usuário seleciona diretamente as velocidades desejadas para o robô através de um *joystick*. As informações de mapeamento e localização do robô podem ser vistos pelo usuário através do RViz, ferramenta de visualização do ROS.



Figura 4.30: Fluxo de Software Operado.



Fonte: Própria

#### 4.2.3.1 Pacote de Locomoção

O pacote de locomoção do robô possui um lugar de destaque entre os pacotes listados anteriormente na seção 4.2.3. Os pacotes de localização e navegação utilizam muitos recursos e programas já desenvolvidos pela comunidade de robótica do ROS, contudo, o *birhex\_locomotion* não tem essa característica, pois foi desenvolvida pelos autores desse trabalho. O funcionamento desse pacote acontece com a junção de duas funcionalidades principais, o gerador de marchas e o suavizador de transições entre marchas.

O gerador de marcha possui como responsabilidade calcular a posição e velocidade angular de cada perna para atender um valor de velocidade do robô desejado. Para isso é feita uma abstração do modelo cinemático do robô utilizando um modelo diferencial, tornando possível assim enviar comandos de velocidade simplificados com velocidade linear e angular. As curvas das marchas respeitam o sistema de equações 4.1, sendo  $\Theta$  a posição angular,  $\omega$  a velocidade,  $t_c$  o tempo de ciclo,  $t_d$  o tempo de trabalho das patas,  $\Theta_d$  a posição de trabalho,  $\Theta_{\text{offset}}$  o deslocamento angular e  $\Theta_{\text{differential offset}}$  o deslocamento diferencial entre os lados.

$$\begin{aligned}
 \text{se } t \leq t_d \left\{ \begin{array}{l} \omega(t) = \frac{\Theta_d}{t_d} \\ \Theta(t) = -\frac{\Theta_d}{2} + \omega(t)t + \Theta_{\text{offset}} + \Theta_{\text{differential offset}} \end{array} \right. \\
 \text{se } t > t_d \left\{ \begin{array}{l} \omega(t) = (2\pi - \Theta_d)/(t_c - t_d) \\ \Theta(t) = \frac{\Theta_d}{2} + \omega(t)(t - t_d) + \Theta_{\text{offset}} + \Theta_{\text{differential offset}} \end{array} \right.
 \end{aligned} \tag{4.1}$$

Contudo, para essas equações acima se relacionarem com as velocidades lineares e angula-

res são necessárias mais quatro, as equações 4.2 e 4.3 relacionam a velocidade linear com o tempo de ciclo e o tempo de ciclo com o tempo de trabalho respectivamente, sendo  $d_f$  o fator de trabalho e possui valores de zero até um.

$$t_c = (\alpha|v_x|)^{-1}, \text{ para } \alpha \text{ e } v_x \neq 0 \quad (4.2)$$

$$t_d = t_c * d_f + t_{d_{\text{differential}}} \quad (4.3)$$

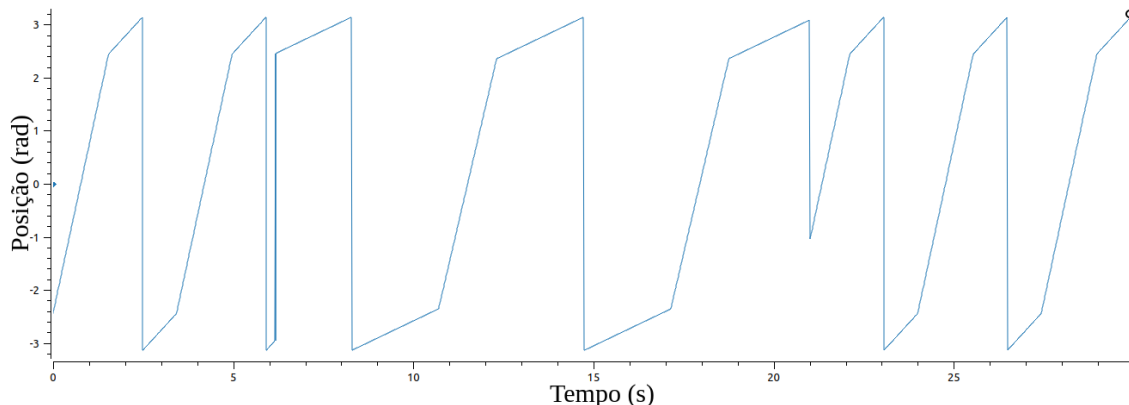
Para definir os valores das variáveis diferenciais,  $t_{d_{\text{differential}}}$  e  $\Theta_{\text{differential offset}}$ , são usadas as equações 4.4 e 4.5 abaixo. Os valores de  $\alpha$ ,  $\beta$  e  $\gamma$  foram encontrados de forma empírica.

$$t_{d_{\text{differential}}} = \frac{\beta v_\omega t_c d_f}{\beta v_\omega + \frac{\Theta}{t_c * d_f}} \quad (4.4)$$

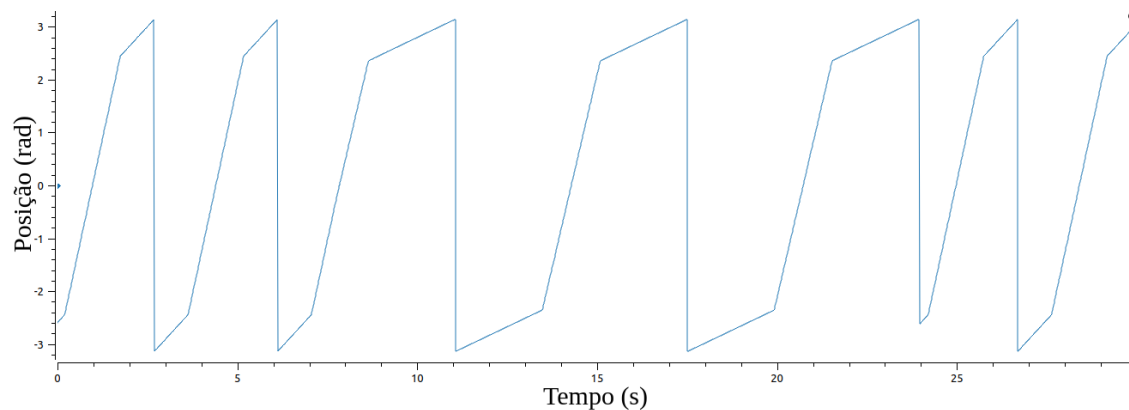
$$\Theta_{\text{differential offset}} = \gamma v_\omega \quad (4.5)$$

Por causa dos vários parâmetros mostrados acima, a mudança entre movimentos lineares para movimentos puramente rotacionais causa uma variação abrupta nos valores de posição e velocidade angular. Esse efeito resulta em desestabilização do robô nesses momentos de transição. Para atuar nesse problema, o suavizador de transições entre marchas aguarda o melhor momento para repassar o comando de velocidade, diminuindo assim os problemas de transição. Um exemplo do funcionamento do suavizador está na figura 4.31 abaixo, onde na primeira imagem mostra uma transição não suavizada enquanto na segunda ocorre a suavização.

Figura 4.31: Exemplo de funcionamento do suavizador de transição.



(a) Transição não Suavizada



(b) Transição Suavizada

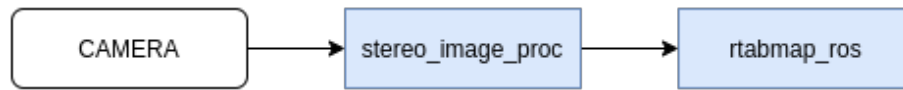
Fonte: Própria

#### 4.2.3.2 Pacote de Localização

Um dos problemas de criar um robô móvel autônomo é a capacidade de fazer sua localização. Por essa razão, desenvolvedores utilizam o ROS para criar programas reutilizáveis que resolver esse problema. Nesta seção será apresentado os programas utilizados no pacote de localização.

- **stereo\_image\_proc**: A principal função do `stereo_image_proc`, em nosso pacote de localização, é fazer a comunicação entre o *drive* da câmera estéreo e os nós de processamento visual. Será feita a retificação das imagens para que possam ser usadas pelos nós de processamento visual, que em nosso caso, será o pacote do `rtabmap`.

Figura 4.32: Funcionamento do stereo\_image\_proc no pacote de localização

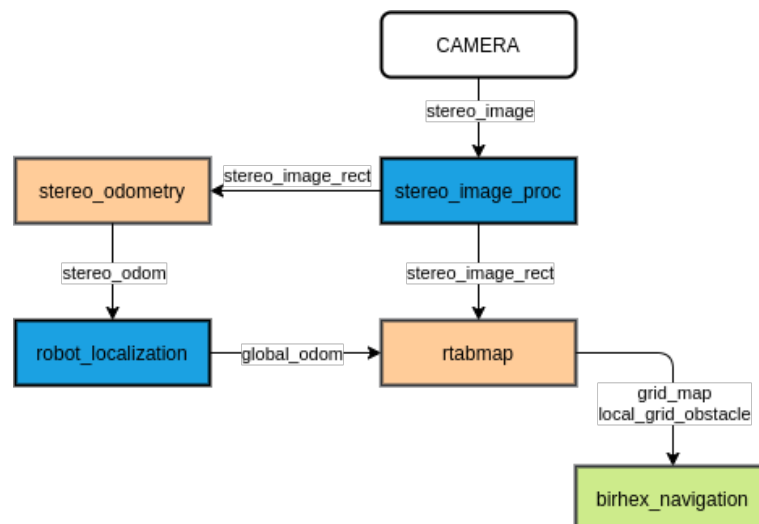


Fonte: Própria

- **rtabmap\_ros**: O pacote rtabmap\_ros é um ROS *wrapper* do *Real-Time Appearance-Based Mapping* (RTAB-Map) que é um RGB-D, *stereo* e *lidar graph-based Simultaneous Localization and Mapping (SLAM)*. Este pacote pode ser usado para gerar nuvens de pontos 3D do ambiente e/ou criar mapas de ocupação 2D para a navegação. O rtabmap\_ros possui o nó principal que é o rtabmap e outros nós secundários para geração de odometria. Dentre os quais, será utilizado o stereo\_odometry.

O stereo\_odometry gerará a odometria visual através das imagens retificadas pelo stereo\_image\_proc. Essa odometria será utilizada pelo pacote do robot\_localization para o cálculo da odometria global. O rtabmap utilizará as imagens retificadas da câmera estéreo assim como a odometria global, para realizar a reconstrução do ambiente em 3D e para criação do grid\_map e o local\_grid\_obstacle. No fluxograma da figura 4.33 é possível ver o funcionamento do pacote rtabmap\_ros em nosso sistema.

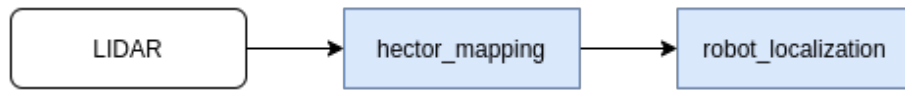
Figura 4.33: Funcionamento do rtabmap\_ros no pacote de localização



Fonte: Própria

- **hector\_mapping**: Hector\_mapping é um pacote capaz de fazer SLAM mesmo sem odometria e em robôs com movimentos em roll e *pitch*. Apesar de o hector\_mapping está gerando um mapa em 2D do ambiente, a sua principal função nesse pacote será a de transformar a nuvem de pontos gerada pelo LIDAR em uma posição estimada do robô em relação ao ambiente. Na figura 4.34 é possível observar o funcionamento do hector\_mapping em nosso pacote de localização.

Figura 4.34: Funcionamento do hector\_mapping no pacote de localização

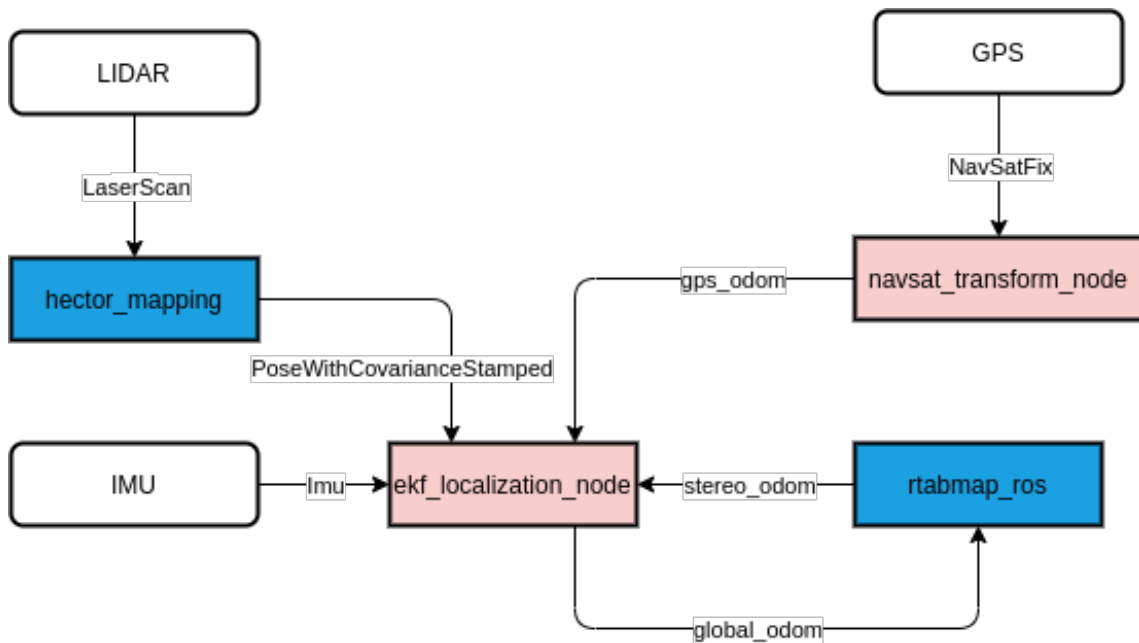


Fonte: Própria

- robot\_localization:** O pacote do robot\_localization utiliza-se de algoritmos como o *Extended Kalman Filter* (EKF) e o *Unscented Kalman Filter* (UKF), para fazer a fusão sensorial e assim gerar medições mais precisas de posição e velocidade. Esse pacote possui três nós sendo dois deles o ekf\_localization\_node e o ukf\_localization\_node, realizam fusão sensorial e o terceiro, o navsat\_transform\_node, transforma as informações de latitude, longitude e altitude adquiridas por um GPS em odometria.

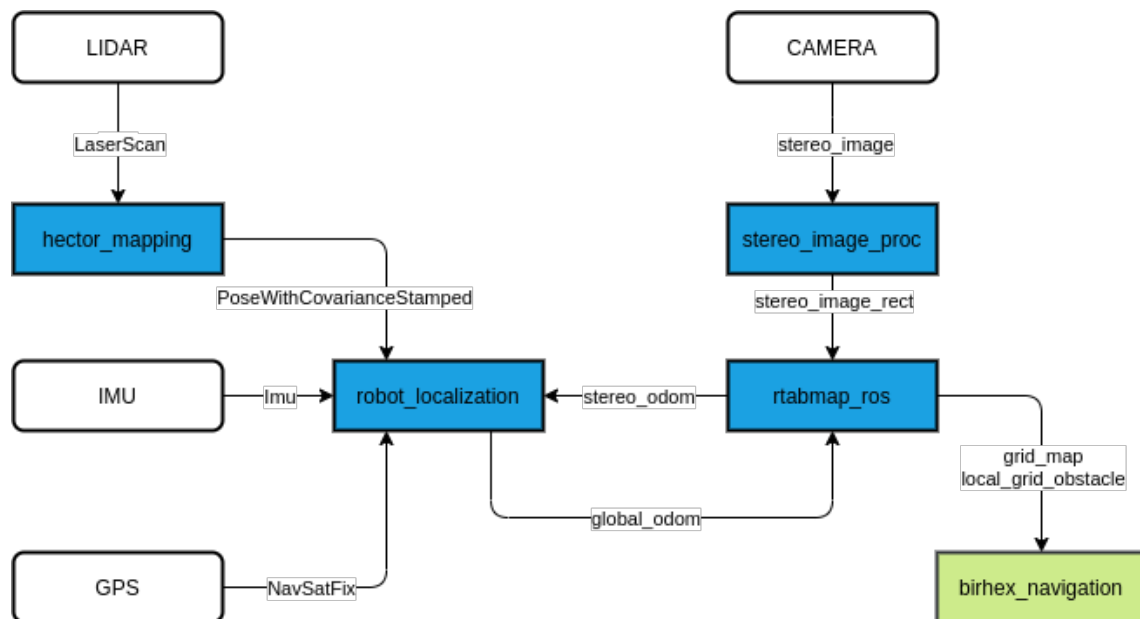
Em nosso pacote o robot\_localization está utilizando o ekf\_localization\_node para a fusão sensorial do IMU, do LIDAR, da odometrias visual e da odometria do GPS para gerar a odometria global, além de estar utilizando o navsat\_transform\_node. A figura 4.35 exibe o funcionamento do robot\_localization no pacote.

Figura 4.35: Funcionamento do robot\_localization no pacote de localização



Fonte: Própria

Figura 4.36: Funcionamento pacote de localização



Fonte: Própria

### 4.2.3.3 Pacote de Navegação

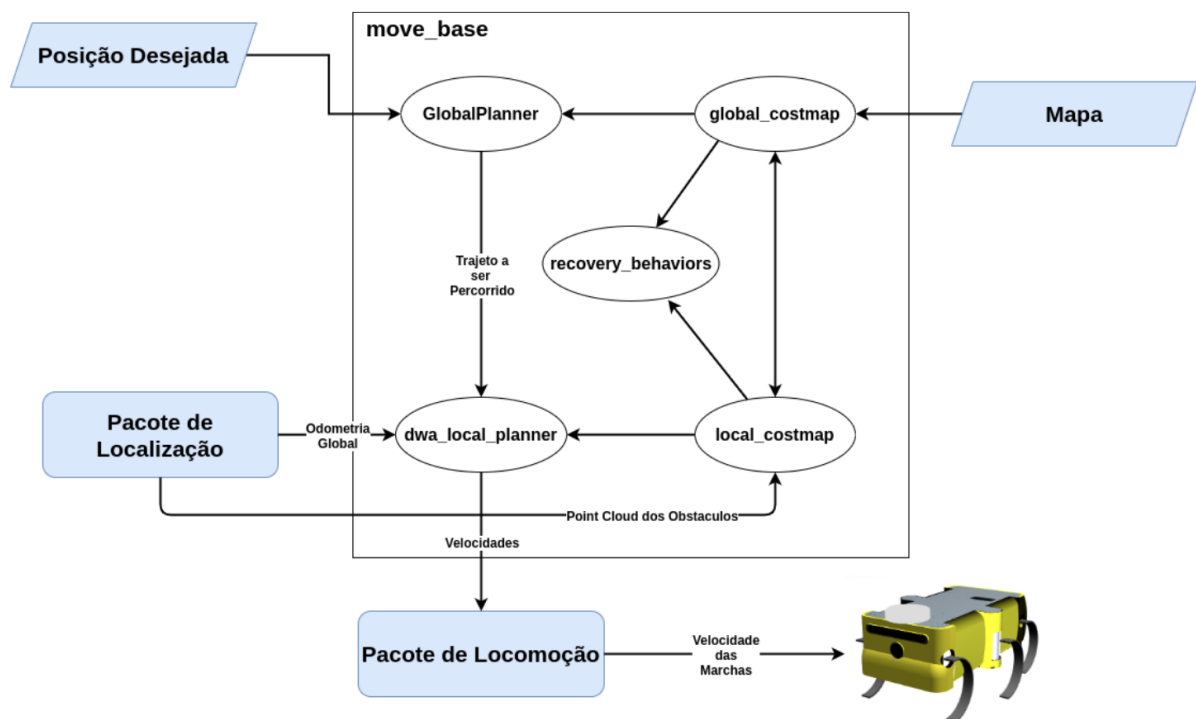
Uma grande característica de um robô móvel autônomo, é a capacidade de realizar a navegação sem interferência humana. Para desenvolver essa capacidade, assim como foi para o pacote de localização, desenvolvedores criaram variados programas para o ROS. Diferente do pacote de localização, que foram utilizados vários pacotes, a navegação foi desenvolvido inteiramente em cima de um único pacote.

O pacote de navegação depende dos pacotes de locomoção e de localização para seu funcionamento, como demonstrado na figura 4.37. A base do pacote de navegação foi o pacote `move_base`.

O pacote `move_base` faz a implementação de uma *action* que receberá uma posição e irá alcançá-la com o robô. Para fazer esta tarefa o pacote utiliza um planejador de trajetórias global e um local. O planejador local escolhido foi o `dwa_local_planner` e o global foi o `global_planner`. Além dos planejadores o `move_base` possui dois *costmaps*<sup>3</sup>, um para cada planejador. Em nosso pacote há três *costmaps*, um para o planejador local e dois para o global, sendo que esses dois não são utilizados ao mesmo tempo. Um deles é para a exploração em ambiente desconhecido e o outro para navegação em ambientes mapeados previamente, podendo o usuário escolher entre os dois modos de operação. O funcionamento do pacote de navegação pode ser visto na figura 4.29.

<sup>3</sup> *Costmaps* ou Mapas de Custos, são mapas que armazenam o custo para se locomover através de determinado local

Figura 4.37: Funcionamento pacote de Navegação



Fonte: Própria

#### 4.2.4 Testes de desempenho

Essa seção pretende mostrar o desempenho dos pacotes desenvolvidos nesse projeto. Será exibido resultado de testes dos pacotes de locomoção, localização e navegação no ambiente de simulação.

##### 4.2.4.1 Locomoção

Como descrito na seção 4.2.3.1, esse pacote possui o objetivo de tornar o comando de velocidade do robô em um perfil de movimento das suas pernas. Com isso, esse teste consistirá no envio de comandos de velocidade para o robô como os valores de referência e a saída a ser monitorada será a velocidade alcançada pelo robô.

Serão separados três categorias de testes diferenciados pela natureza do comando de velocidade, podendo ser: linear, angular ou misto. Os comandos lineares possuem comandos somente de translação, os angulares somente de rotação, já os mistos possuem os dois ao mesmo tempo.

Para as velocidades somente com componentes lineares foram enviados onze comandos diferentes para o robô na simulação e armazenado os dados de velocidade de referência e velocidade real<sup>4</sup> para então serem processados. A tabela 4.1 mostra os valores de comando e as velocidades médias durante a execução.

Tabela 4.1: Resultados dos Comandos Lineares.

<b>Comando Linear (m/s)</b>	<b>Comando Angular (rad/s)</b>	<b>Linear Média (m/s)</b>	<b>Angular Média (rad/s)</b>
0,00	0,00	-0,0007	-0,0003
0,01	0,00	0,0145	-0,0002
0,02	0,00	0,0272	-0,0004
0,04	0,00	0,0593	-0,0006
0,05	0,00	0,0489	-0,0004
0,06	0,00	0,0697	-0,0005
0,07	0,00	0,0800	-0,0007
0,08	0,00	0,0905	-0,0010
0,09	0,00	0,1006	-0,0012
0,10	0,00	0,1111	-0,0012
0,11	0,00	0,1246	-0,0017

Fonte: Própria

Como mostra a tabela 4.1, a velocidade linear real segue o comportamento do comando, contudo, ainda existe um erro, diferença entre a velocidade desejada e a velocidade média alcançada, que aumenta de forma absoluta com o aumento da velocidade de comando, mas diminui de forma percentual. Já as velocidades angulares seguiram o esperado e se mantiveram muito próximas a zero.

Os comandos angulares seguem o comportamento das lineares, tabela 4.2, com os valores reais seguindo a referência, mas apresentando um erro que escala de forma absoluta com a velocidade e diminui de forma relativa. Um comportamento importante de se ressaltar é a diminuição da velocidade média quando o comando angular está em 0,9. Isso acontece, pois, acima de 0,8 existe uma degradação do movimento de rotação em consequência do limite de velocidade do nosso atuador.

<sup>4</sup>A velocidade real nesse caso, é a alcançada pelo robô na simulação.



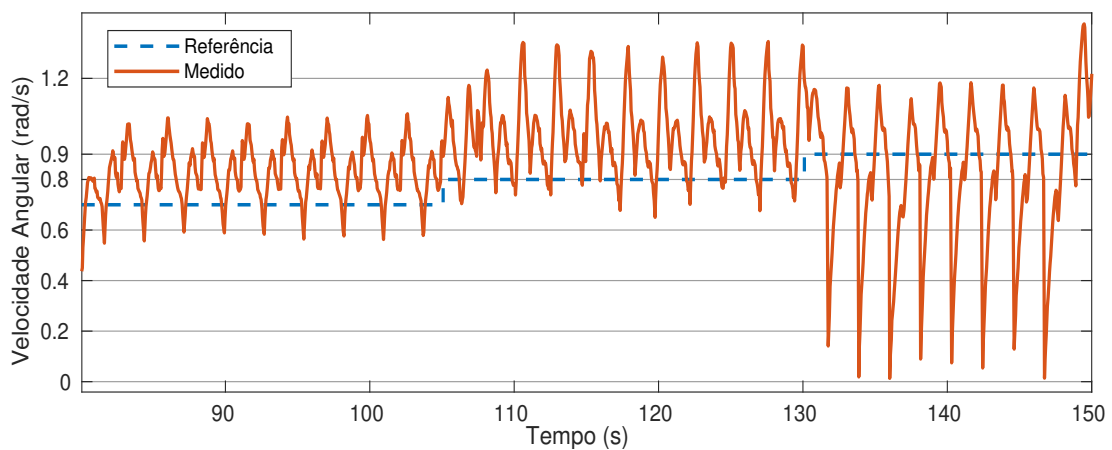
Tabela 4.2: Resultados dos Comandos Angulares.

Comando Linear (m/s)	Comando Angular (rad/s)	Linear Média (m/s)	Angular Média (rad/s)
0,00	0,00	0,0005	0,0048
0,00	0,10	0,0006	0,0783
0,00	0,30	0,0010	0,2682
0,00	0,50	0,0026	0,5209
0,00	0,70	0,0031	0,8233
0,00	0,80	0,0071	0,9750
0,00	0,90	-0,0009	0,8277

Fonte: Própria

Na figura 4.38 é mostrado um gráfico com as curvas de velocidade angular medida e de referência, e é possível perceber que apesar do comportamento oscilatório em todos os comandos, causada pela forma circular da pata do robô, quando a referência está em 0,9, a amplitude é maior.

Figura 4.38: Degradação da Velocidade Angular



Fonte: Própria

Para o cálculo dos comandos mistos, foram realizadas as mesmas etapas dos tipos anteriores e os dados da tabela 4.3 mostram os resultados obtidos. A principal informação retirada dele é como uma velocidade está relacionada a outra, ou seja, elas não se alteram de forma isolada como deveria acontecer. Outra informação importante está relacionada aos valores de velocidade angular quando combinada com a linear: o robô não suporta movimentos superiores a  $0,1 \frac{rad}{s}$ , devido à velocidade de seus atuadores.

Com essas informações é possível notar que o pacote de locomoção funciona, contudo,

Tabela 4.3: Resultados dos Comandos Mistos.

<b>Comando Linear (m/s)</b>	<b>Comando Angular (rad/s)</b>	<b>Linear Média (m/s)</b>	<b>Angular Média (rad/s)</b>
0,00	0,00	-0,0003	-0,0011
0,10	0,00	0,1031	0,0004
0,10	0,07	0,1186	0,0467
0,07	0,07	0,0804	0,0893
0,07	0,09	0,0824	0,1281
0,10	0,099	0,1203	0,0777

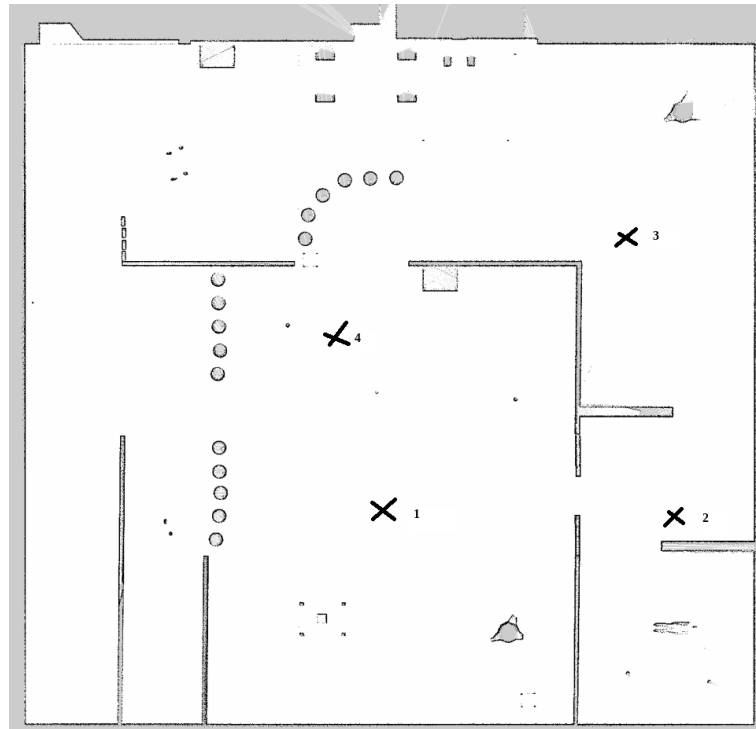
Fonte: Própria

possui alguns erros em seus comandos. Esses erros são provavelmente consequências das escolhas dos valores de  $\alpha$ ,  $\beta$  e  $\gamma$  da seção 4.2.3.1.

#### 4.2.4.2 *Localização e Navegação*

O mapa da figura 4.24 foi utilizado para realizar os teste de localização e navegação. Esse teste consiste em realizar várias voltas em um percurso predeterminado. Este percurso foi feito utilizando o mapa criado através do SLAM realizado pelo pacote de localização e utilizando o pacote de navegação, que ao receber 4 coordenadas, realiza o percurso autonomamente. As coordenadas e o mapa utilizado podem ser visualizados na figura 4.39. Dentro deste percurso, existem trechos com alguns obstáculos que simulam um terreno de difícil locomoção como exibido na figura 4.26. Para se avaliar a performance dos pacotes, foram coletados os dados dos sensores individualmente e os dados da fusão sensorial, que gera a odometria global, como será demonstrado nas figuras e tabelas abaixo.

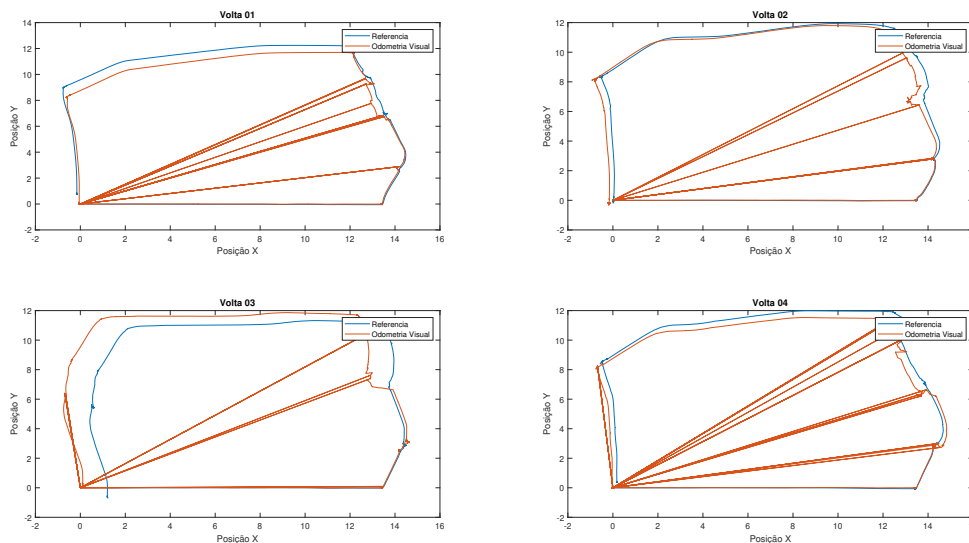
Figura 4.39: Mapa do ambiente de simulação



Fonte: Própria

- **Odometria Visual:** Pelos gráficos da figura 4.40 é possível observar que no início do percurso até aproximadamente a posição (14;7) do gráfico, a diferença entre o ground truth e a odometria visual é praticamente nula. A Partir desse ponto a odometria visual começa a apresentar maiores erros de medição, isso acontece devido ao fato de que neste ponto o robô encontra o terreno acidentado da figura 4.26, e para poder passar por ele o robô acaba fazendo alguns movimentos que fazem com que a câmera perca visão ou vire para cima, isso faz com que a câmera não consiga detectar características suficientes do ambiente para poder realizar a localização. Quando isso acontece a câmera perde a capacidade de realizar odometria e acaba mandando zero como posição e por isso existem trechos no gráfico que a odometria visual volta para zero. Fica visível que depois que o robô passa pelo terreno acidentado o erro de medição diminui, mas não volta ao seu estado original. Através das médias da tabela 4.4 é demonstrado uma precisão média de 0,2979 metros na odometria visual.

Figura 4.40: Gráfico de posição Ground Truth vs Odometria Visual



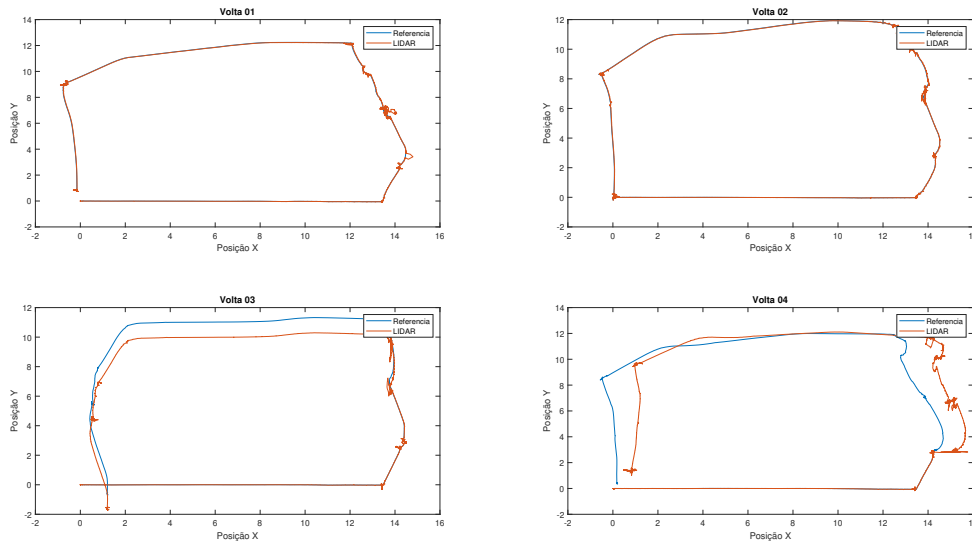
Fonte: Própria

Tabela 4.4: Tabela de Erro Ground Truth vs Odometria Visual

	Media(m)	Desvio Padrão
Volta 01	0,3852	0,6682
Volta 02	0,1772	0,4409
Volta 03	0,3648	0,5051
Volta 04	0,2642	0,6480
Media Total	0,2979	0,5656

- LIDAR:** Assim como a odometria visual o LIDAR praticamente não tem erro de medição antes do terreno acidentado mas diferente da odometria visual o LIDAR não apresenta erros todas as vezes que ele passa pelo terreno acidentado, como demonstrado nos gráficos da Volta Um e Volta Dois da figura 4.41. É importante salientar que a apesar do LIDAR ter resultados muitos bons, de até 0,01 metros de erro na Volta Um, também pode se ter resultados muito ruins como na Volta Quatro, que apresentou um erro médio de 0,72 metros, como mostra a tabela 4.5.

Figura 4.41: Gráfico de posição Ground Truth vs LIDAR



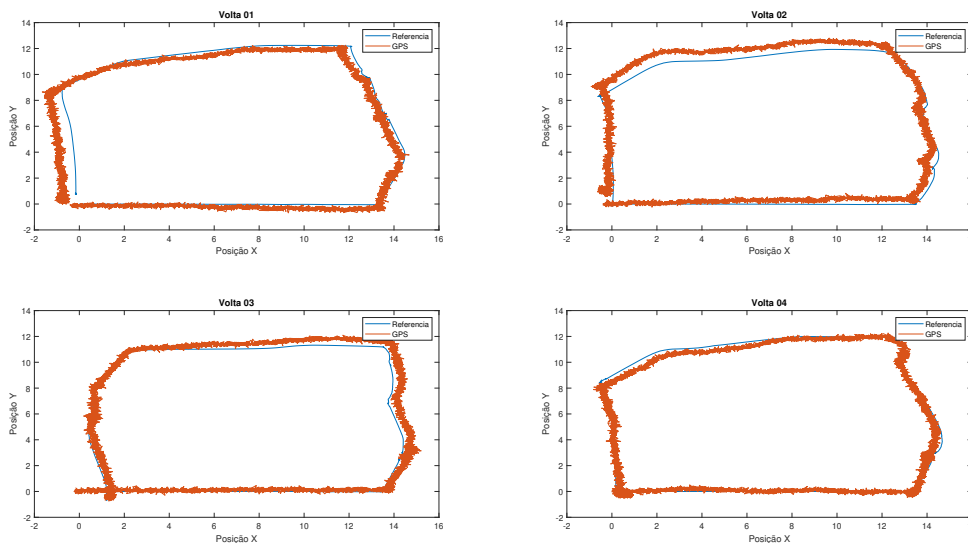
Fonte: Própria

Tabela 4.5: Tabela de Erro Ground Truth vs LIDAR

	Media(m)	Desvio Padrão
Volta 01	0,01296	0,02961
Volta 02	0,01372	0,03936
Volta 03	0,4328	0,4323
Volta 04	0,7222	0,5182
Media Total	0,2954	0,2549

- **GPS:** Diferente do LIDAR e da odometria visual o GPS já apresenta um erro quase que constante desde o início do percurso, já que suas medições oscilam bastante, como demonstrado nos gráficos da figura 4.42. Apesar do GPS gera uma medição com uma precisão inferior aos outros sensores as suas medições não são afetadas quando o robô está passando pelo terreno acidentado. Pelo fato de o GPS não ser afetado pelo terreno acidentado ele conseguiu ter um média de erro muito parecida com as do outros sensores, apesar de ser um pouco maior, como mostra a tabela 4.6.

Figura 4.42: Gráfico de posição Ground Truth vs GPS



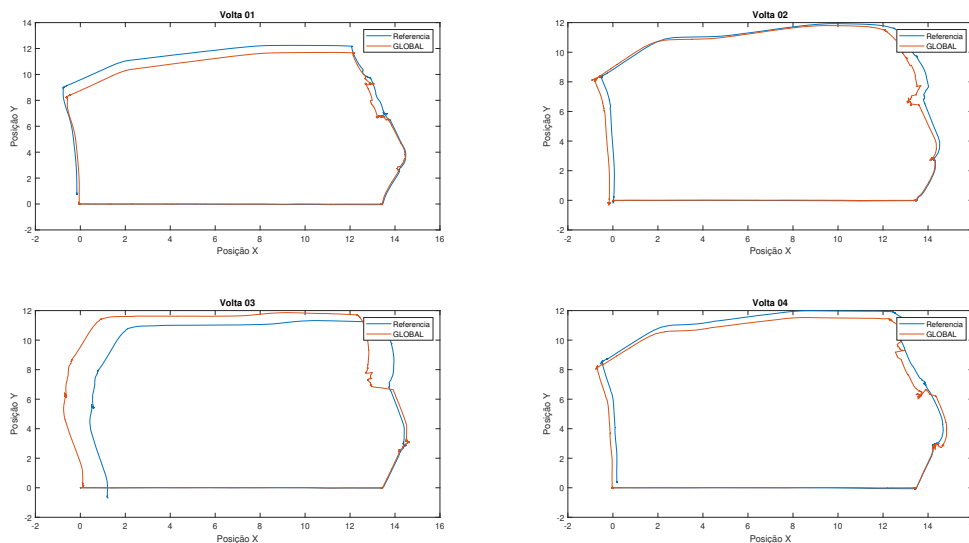
Fonte: Própria

Tabela 4.6: Tabela de Erro Ground Truth vs GPS

	Media(m)	Desvio Padrão
Volta 01	0,2798	0,1660
Volta 02	0,4197	0,3644
Volta 03	0,2873	0,1639
Volta 04	0,2247	0,1731
Media Total	0,3029	0,2169

- Odometria Global:** A partir dos dados coletados acima perceptível que todos os sensores possuem algum tipo de erro, para mitiga-los foi utilizado o pacote do `robot_localization` para a fusão sensorial e gerar a odometria global a partir dela. É notório, através dos gráficos da figura 4.43, que a fusão sensorial consegue resolver a maioria dos problemas encontrados nos sensores anteriores. Da odometria visual ela resolve o problema de perda de odometria, do GPS ela resolve o problema de oscilação das medições, e do LIDAR ela consegue mitigar os grandes erros de medições que algumas vezes acontece, que em nossos testes foram 50% das vezes. Pelos gráficos da figura 4.43 e pela tabela 4.7 é observável que fora as Voltas Um e Dois do LIDAR, a odometria global foi a que teve a melhor performance, como esperado.

Figura 4.43: Gráfico de posição Ground Truth vs Odometria Global



Fonte: Própria

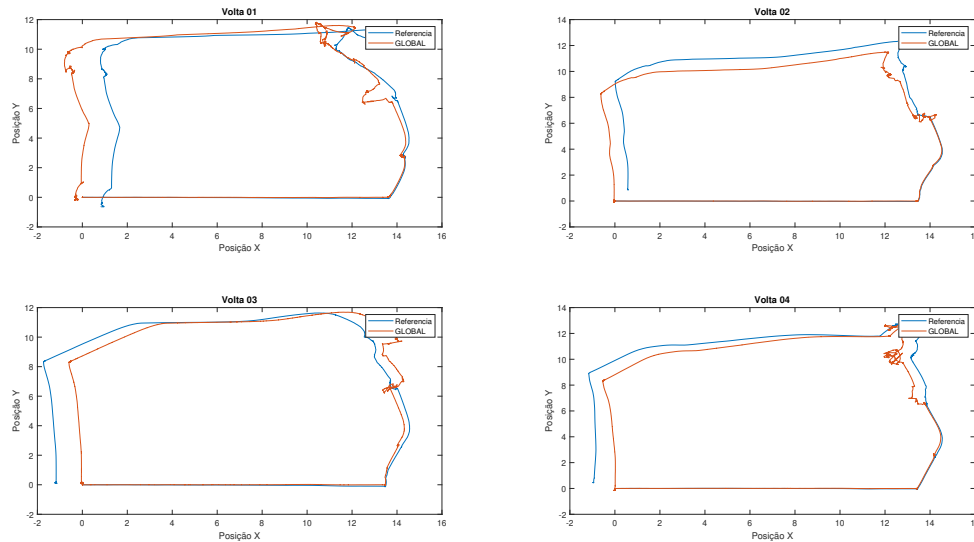
Tabela 4.7: Tabela de Erro Ground Truth vs Odometria Global

	Media(m)	Desvio Padrão
Volta 01	0,2394	0,1947
Volta 02	0,3535	0,3420
Volta 03	0,1656	0,1410
Volta 04	0,3610	0,2962
Media Total	0,2799	0,2435

- Navegação:** No teste de navegação foram realizadas 8 voltas no percurso anterior, sendo 4 delas feitas com o mapa da figura 4.39 e as outras 4 sem mapa e utilizando o SLAM. Em todas as voltas o Birhex conseguiu completar o percurso.

Fica evidente então que o pacote de navegação é extremamente dependente do pacote de localização, já que se for considerar os dados de localização da odometria global ideais, o robô estaria sempre chegando às coordenadas desejadas. Nos gráficos das figuras 4.43 e 4.44 isso é demonstrando, já que o robô sempre volta para o ponto zero da odometria global.

Figura 4.44: Gráfico de posição Ground Truth vs Odometria Global (SLAM)



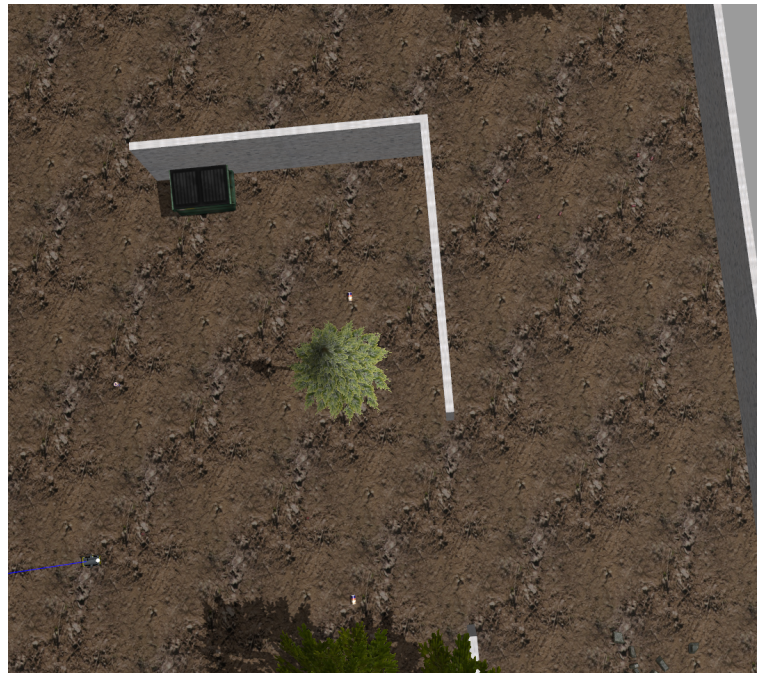
Fonte: Própria

### 4.2.5 Resultados

Para poder avaliar o desempenho do software como um todo foi simulado uma situação hipotética na qual o Birhex deveria percorrer um determinado percurso e voltar para o seu ponto de partida. Para isso foi realizado 30 voltas no mesmo percurso mostrado na secção 4.2.4.2, sendo que 10 voltas foram realizadas com o mapa sem obstáculos, como mostra a figura 4.45, outras 10 voltas com o mapa com obstáculos, como mostra a figura 4.46, e 10 voltas com o mapa com obstáculos e com o terreno acidentado, como foi feito anteriormente. Nos testes foi avaliado a capacidade do Birhex de realizar um determinado percurso e voltar para a sua posição inicial, o tempo necessário para percorrer o percurso, e o erro médio da localização durante o percurso.



Figura 4.45: Mapa sem obstáculos



Fonte: Própria

Figura 4.46: Mapa com obstáculos



Fonte: Própria

Os testes realizados com o mapa sem obstáculos apresentaram bons resultados, com um erro médio durante o trajeto de apenas 0,12 centímetros, o Birhex sempre conseguiu realizar o trajeto predeterminado e voltar próximo a sua posição inicial com um erro médio de apenas 22 centímetros, como podemos observar na tabela 4.8.

Os resultados encontrados no mapa com obstáculos não foram tão bons quanto os resultados encontrados no mapa sem obstáculos. O erro médio encontrado durante o percurso das 10 voltas foi de 35 centímetros e o erro médio de sua posição final foi de 73,27

centímetros, como mostrado na tabela 4.9. Apesar de ter um aumento no erro, de até 3 vezes em relação ao mapa sem obstáculos, o Birhex ainda conseguiu realizar o percurso em todas as voltas no mapa com obstáculos.

Há simulação feita com o mapa com obstáculos e com o terreno acidentado foi a que apresentou os piores resultados. Com um erro médio durante o percurso de 43,92 centímetros e um erro médio na sua posição final de 98,63 centímetros, apesar disso o robô ainda foi capaz de realizar todo o percurso.

Em todos os testes o tempo para percorrer todo o trajeto foram muito parecidos, com o mapa com obstáculos com o maior tempo médio, de 11,62 minutos, e o mapa sem obstáculos com o menor tempo médio, de 9,62 minutos. Isso quer dizer que o Birhex consegue lidar relativamente bem com os obstáculos já que houve apenas 20,7% de acréscimo do tempo utilizado para percorrer o percurso.

Tabela 4.8: Percurso sem obstáculos

	<b>Posição Final(metros)</b>	<b>Erro Medio(metros)</b>	<b>Tempo(minutos)</b>
<b>Volta 1</b>	0,149	0,1069	9,285
<b>Volta 2</b>	0,255	0,06796	8,441
<b>Volta 3</b>	0,3461	0,1388	9,915
<b>Volta 4</b>	0,1959	0,2109	9,918
<b>Volta 5</b>	0,00483	0,1258	8,95
<b>Volta 6</b>	0,5373	0,1857	9,091
<b>Volta 7</b>	0,3846	0,1663	8,449
<b>Volta 8</b>	0,1619	0,06194	10,23
<b>Volta 9</b>	0,02568	0,05015	10,77
<b>Volta 10</b>	0,2118	0,0777	11,17
<b>Media Total</b>	0,2272	0,119215	9,6219

Fonte: Própria

Tabela 4.9: Percurso com obstáculos

	Posição Final(metros)	Erro Medio(metros)	Tempo(minutos)
<b>Volta 1</b>	0,4124	0,2516	10,87
<b>Volta 2</b>	0,4977	0,1547	11,19
<b>Volta 3</b>	0,7218	0,2746	10,61
<b>Volta 4</b>	1,608	0,9819	13,12
<b>Volta 5</b>	0,2309	0,09144	9,551
<b>Volta 6</b>	0,2862	0,1866	10,86
<b>Volta 7</b>	0,22	0,05624	10,74
<b>Volta 8</b>	0,1954	0,2567	14,08
<b>Volta 9</b>	1,465	0,5607	10,85
<b>Volta 10</b>	1,69	0,688	14,19
<b>Media Total:</b>	0,7327	0,3502	11,62

Fonte: Própria

Tabela 4.10: Percurso com obstáculos e terreno acidentado

	Posição Final(metros)	Erro Medio(metros)	Tempo(minutos)
<b>Volta 1</b>	0,9495	0,3202	11,15
<b>Volta 2</b>	1,358	0,537	10,20
<b>Volta 3</b>	1,038	0,2452	13,04
<b>Volta 4</b>	1,061	0,4779	9,61
<b>Volta 5</b>	0,8437	0,8164	11,46
<b>Volta 6</b>	2,065	0,8762	11,57
<b>Volta 7</b>	0,7361	0,361	10,56
<b>Volta 8</b>	0,2434	0,1656	10,07
<b>Volta 9</b>	1,207	0,3535	10,58
<b>Volta 10</b>	0,3611	0,2394	11,03
<b>Media Total:</b>	0,9863	0,4392	10,93

Fonte: Própria

Com esses testes foi possível observar que o Birhex pode ser usado como um robô autônomo para tarefas que não exijam uma precisão muito grande, já que apesar de ele ter tido um resultado bom no ambiente sem obstáculos, o provável cenário que ele irá encontrar será o com obstáculos e ou com um terreno acidentado.

## CONCLUSÃO

Este trabalho apresentou a construção do projeto de *software*, eletrônica e mecânica de um robô modelo RHex com a capacidade de andar sobre terrenos acidentados, nomeado BIRHex. A plataforma foi equipada com bateria, circuitos de proteção e ajustes de corrente e tensão, sensores, uma unidade de processamento e seis servos motores inteligentes.

O projeto eletrônico resultou no desenho de duas placas de circuito impresso, uma para receber as conexões da bateria e fazer a distribuição com os diferentes conversores *DC-DC* e outra para a utilização de fusíveis de proteção. Os desenhos e modelos mecânicos criados detalham a estrutura do robô que foi separada em duas camadas principais, a camada inferior ficou responsável por acoplar a bateria, conversores *DC-DC*, estruturas de fixação e resfriamento, enquanto a camada superior ficou responsável pelo computador, sistema microprocessado e sensores de percepção. Com isso criando uma divisão entre sistemas de potências e os sistemas de inteligência e processamento de dados.

O *framework* ROS que foi utilizado nos códigos e pacotes e que também foi o responsável pela integração com o Gazebo, *software* utilizado para a execução da simulação, se mostrou primordial para a criação de toda a arquitetura do robô em um tempo extremamente curto graças a grande comunidade envolvida no *framework*. Os pacotes foram divididos quanto a sua funcionalidade, localização, simulação, navegação, locomoção e de *interface* com o operador. Dentre desses, a locomoção é um pacote que merece destaque, pois foi desenvolvida e detalhada dentro deste projeto.

Os pacotes de locomoção, localização e navegação passaram por testes de desempenho para mostrar de forma objetiva seus desempenhos. A locomoção se mostrou efetiva na execução, principalmente nos movimentos lineares do robô, todavia, existiram erros inseridos por sintonização incorreta de parâmetros. A localização apresenta alguns erros, principalmente sobre terrenos acidentados ou em caso de colisão, porém, também foi evidenciado que a fusão sensorial permite que os sensores se complementem, onde mesmo durante um erro temporário de medição as informações de posição se mantêm coerentes. A navegação se mostrou eficiente mesmo sobre terrenos acidentados, entretanto é muito dependente dos outros pacotes, onde erros em sistemas anteriores pode afetar a capacidade de movimentação autônoma do robô.

Esse projeto trouxe a toda a equipe grandes desafios técnicos e de gerenciamento de projetos. Como desafios técnicos enfrentados teve-se a integração entre as três áreas

do robô, mecânica, eletrônica e *software*, a montagem da arquitetura geral e também a configuração dos pacotes de percepção do robô. Já no gerenciamento teve desafios na divisão das tarefas e o acompanhamento, e para isso foram utilizados o Kaban e reuniões periódicas para reportar a atividade. A participação a partir da concepção de um projeto de robótica até sua simulação e desenvolvimento do detalhado enriqueceu a experiência na área de todos os participantes da equipe.

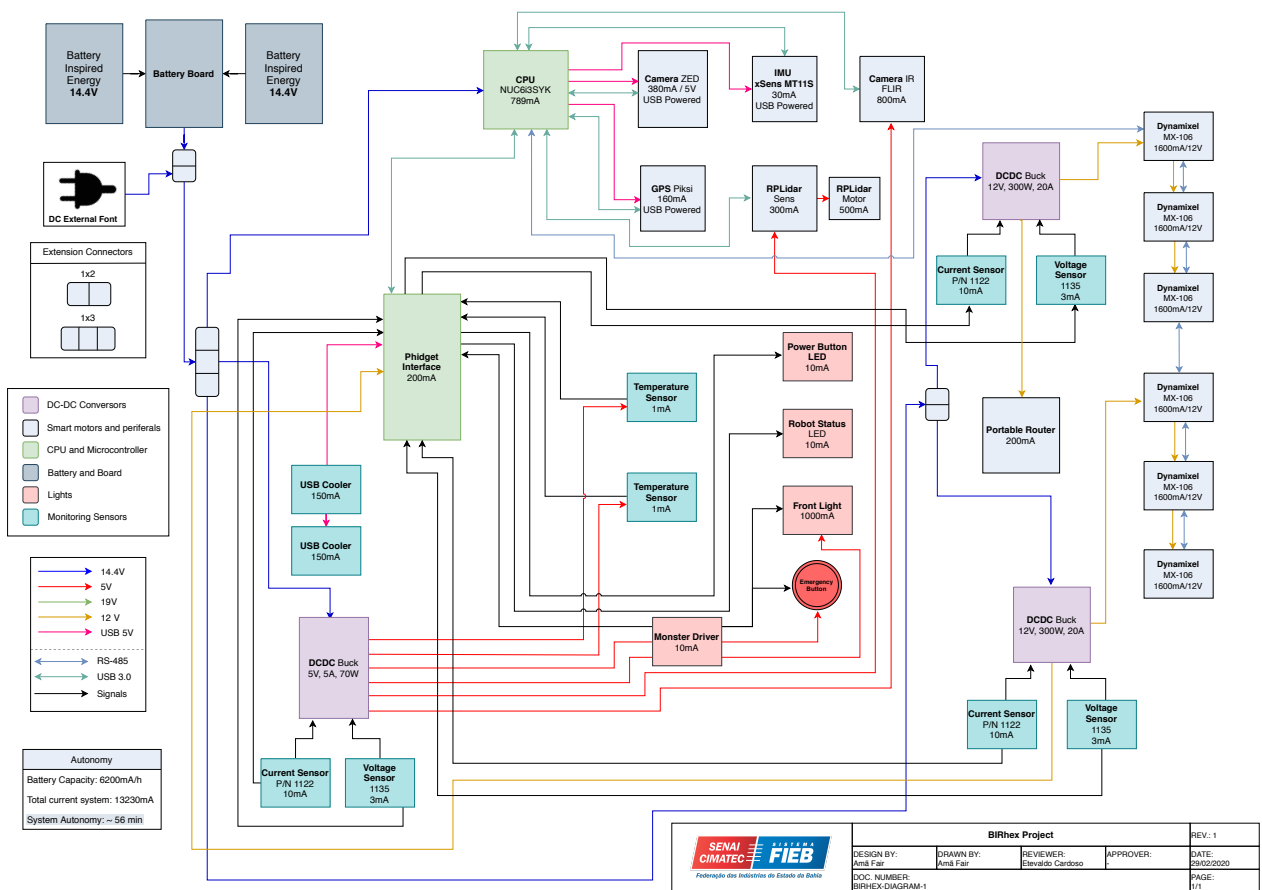
## 5.1 *Trabalhos Futuros*

Ao decorrer desse projeto foi encontrado pontos interessantes a serem estudados para possíveis trabalhos futuros.

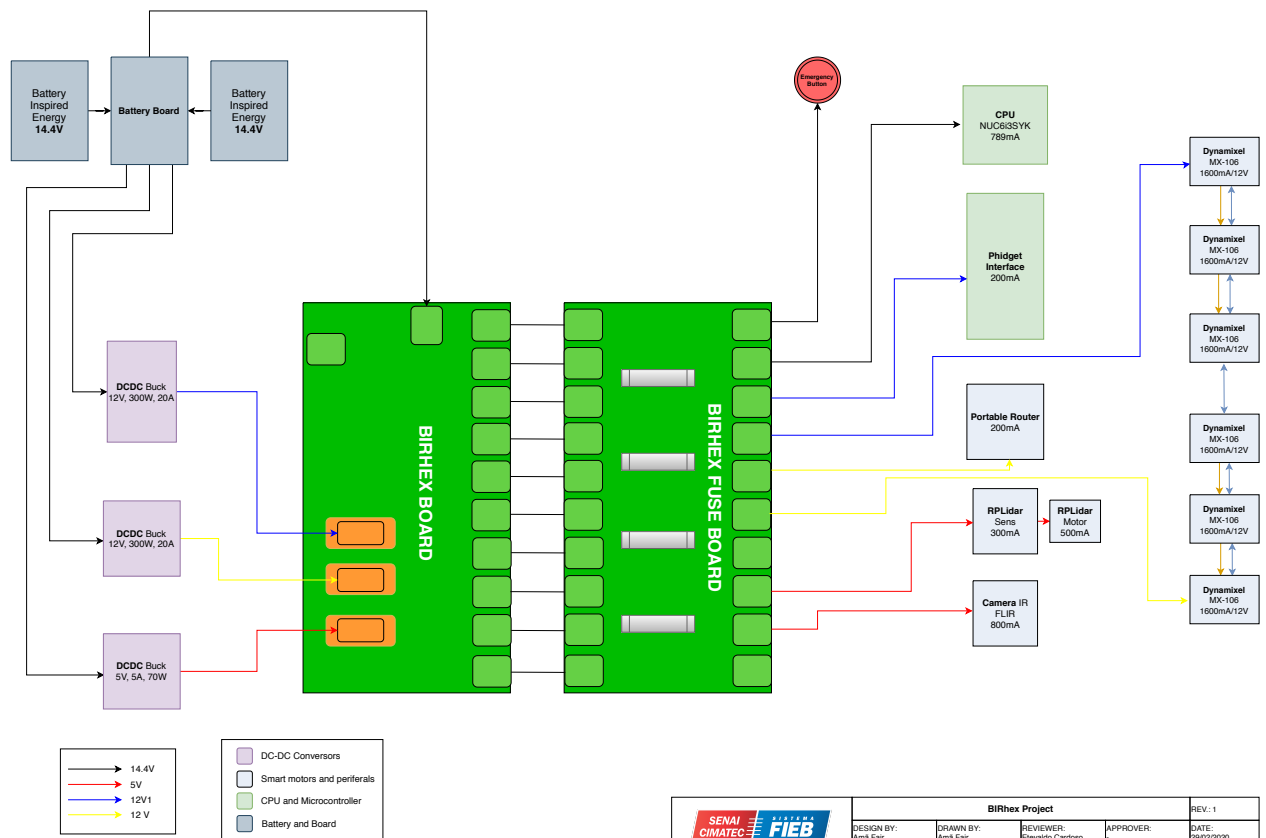
- **Criação do pacote da câmera IR:** O esquemático e modelo mecânico já contemplam a existência da câmera IR, contudo, não foi desenvolvido um pacote utilizar as imagens na percepção.
- **Melhorias no *design*:** A estrutura do robô pode ser melhorada para se tornar resistente ou impermeável à água, permitindo assim andar em terrenos barrosos. Também é possível a implementação de luzes para o funcionamento em ambientes cavernosos ou de baixa luminosidade.
- **Novas marchas:** O BIRHex permite a execução de diferentes marchas, uma das mais interessantes seria a marcha de subir escadas.
- **Aumento de Velocidade:** O robô pode se beneficiar bastante com o aumento das velocidades de seus motores, permitindo uma navegação mais estável.

Por fim, pode-se afirmar que o trabalho correspondeu a expectativa, as metas e entregas propostas, a criação do projeto eletrônico, mecânico e do *software*, foram realizadas, o que permite navegação autônoma ao robô mesmo sobre ambientes acidentados. Apesar de algumas limitações e problemas no desenvolvimento a equipe está satisfeita com o resultado técnico entregue no projeto do BIRHex.

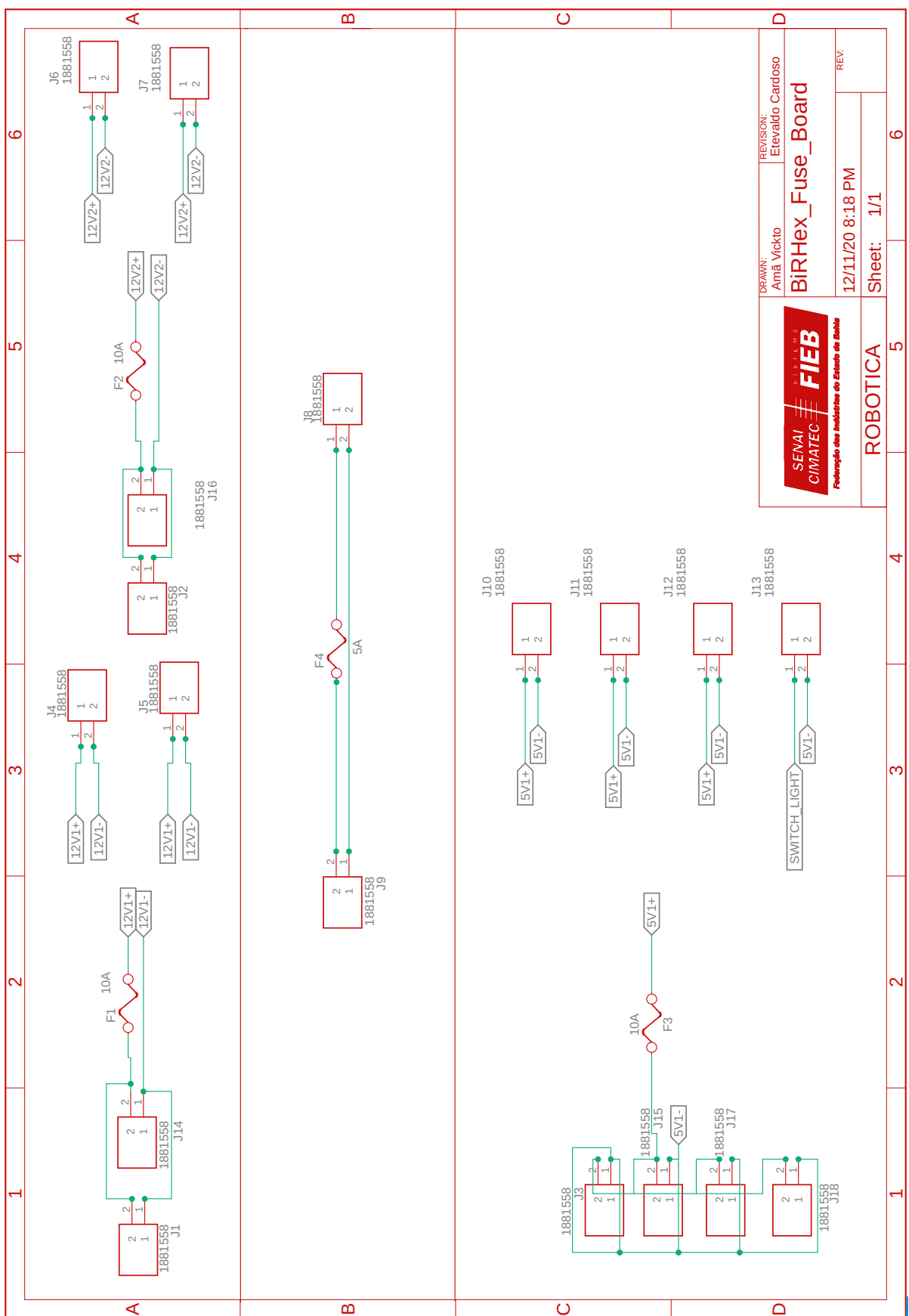
# Diagrama I



# Diagrama II



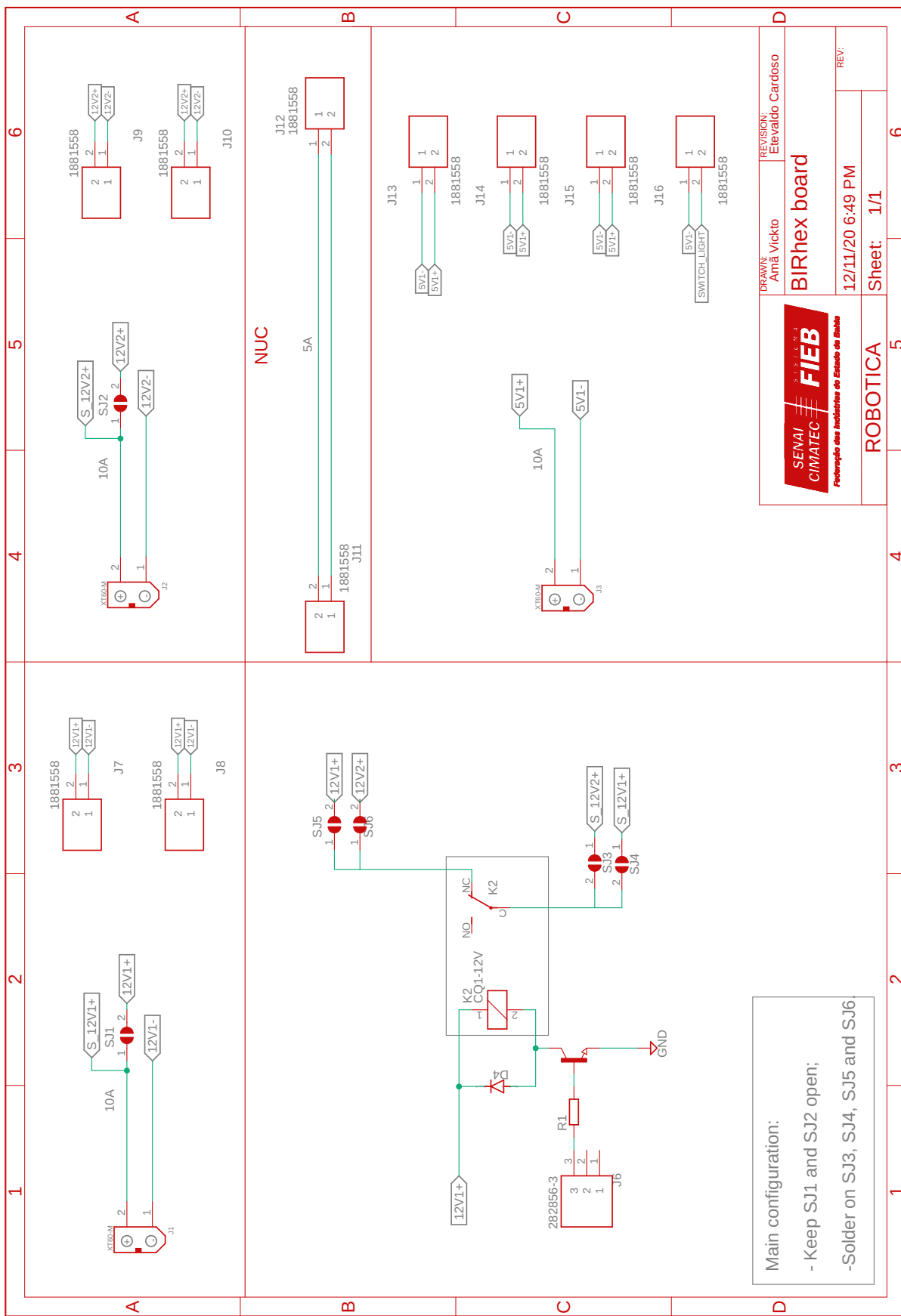
# Esquemáticos eletrônicos



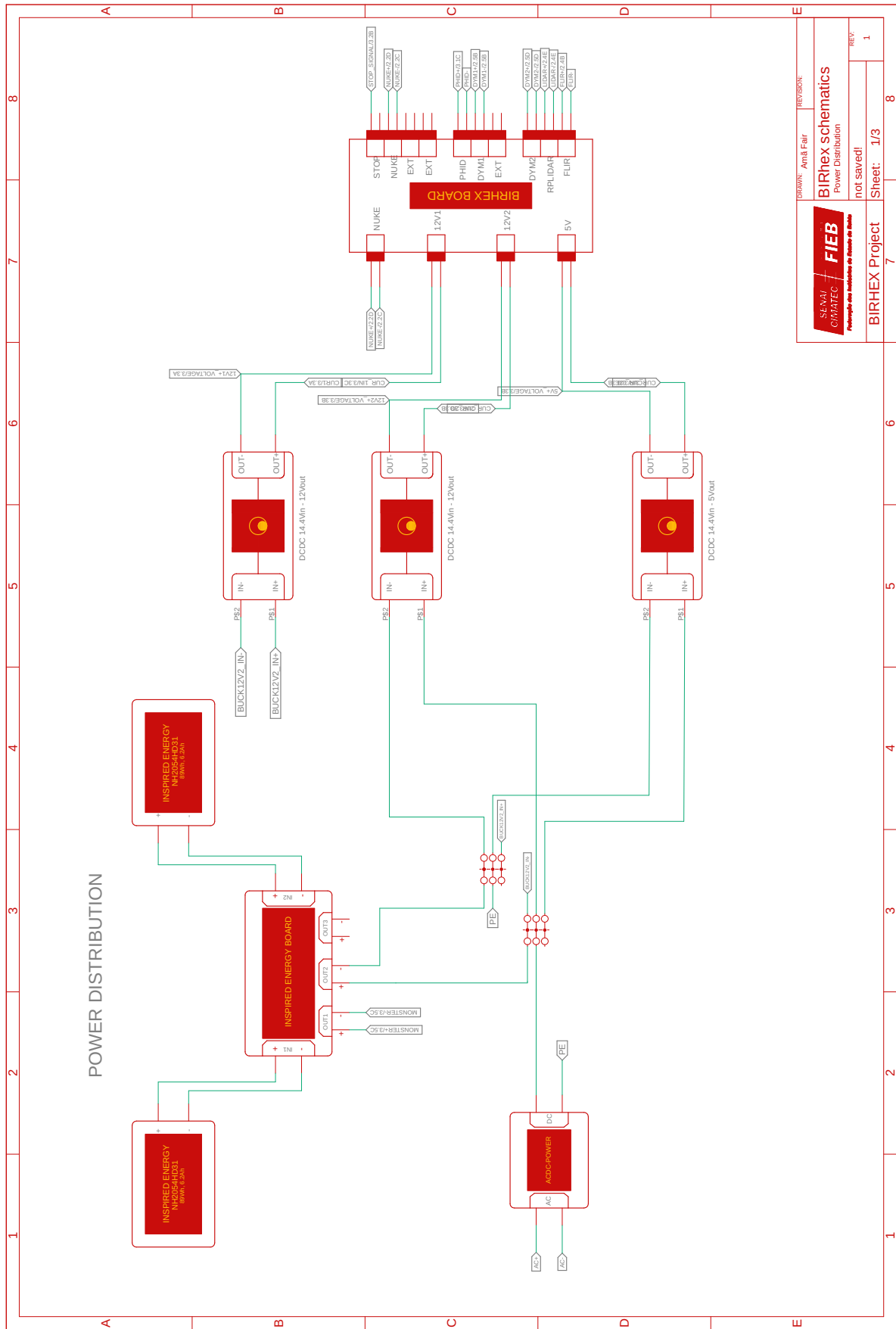
 <b>SENAI CIMATEC</b> <small>Federação dos Institutos do Estado de Bahia</small>	DRAWING: Amíl Vickto	REVISION: Etevaldo Cardoso
	<b>BiHex_Fuse_Board</b>	
12/11/20 8:18 PM		REV:
Sheet: 1/1		6

ROBOTICA

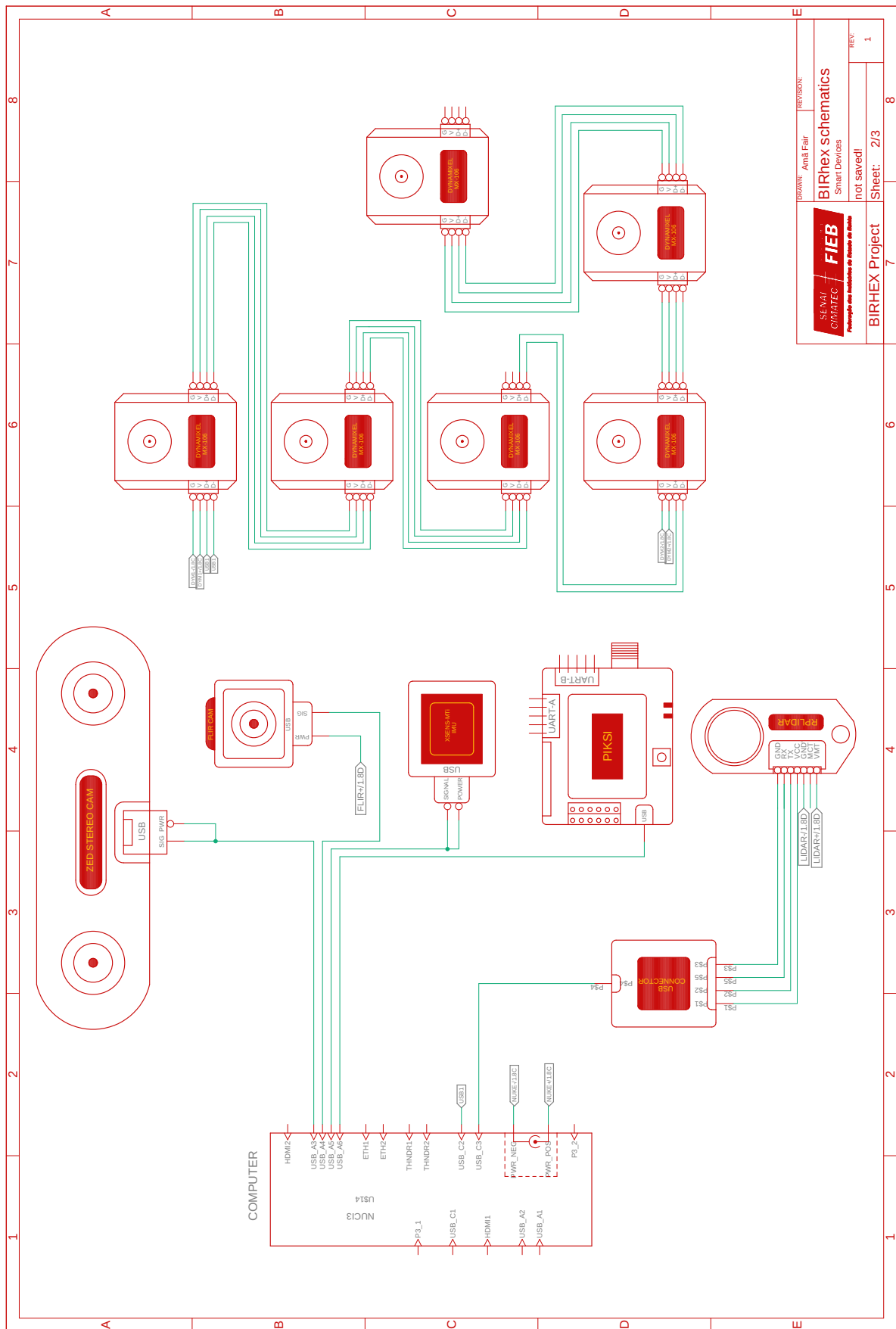




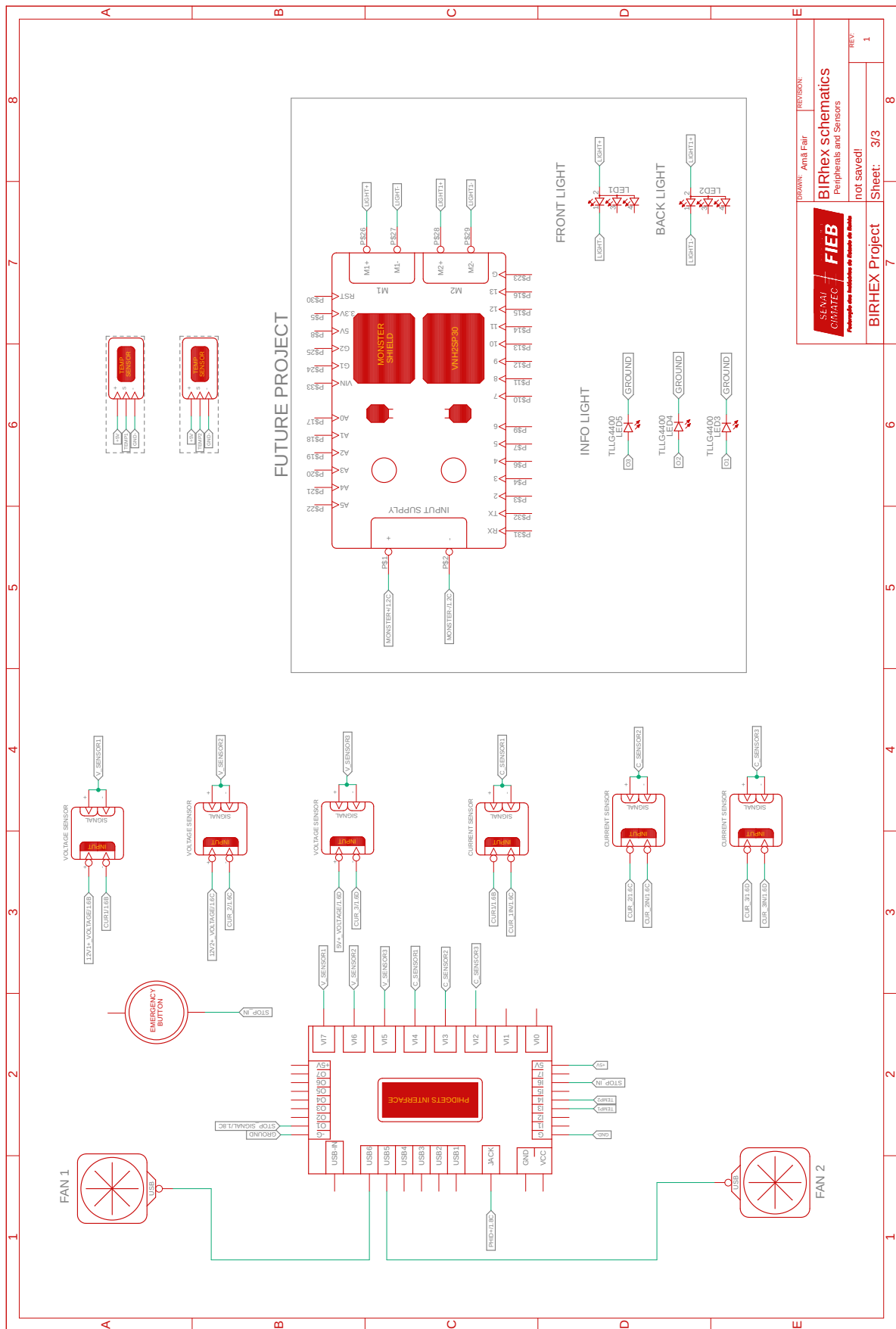
	DRAWING: Amá Vickto	REVISION: Ezevaldo Cardoso
	<b>BIRhex board</b>	
12/11/20 6:49 PM		REV:
ROBOTICA		Sheet: 1/1



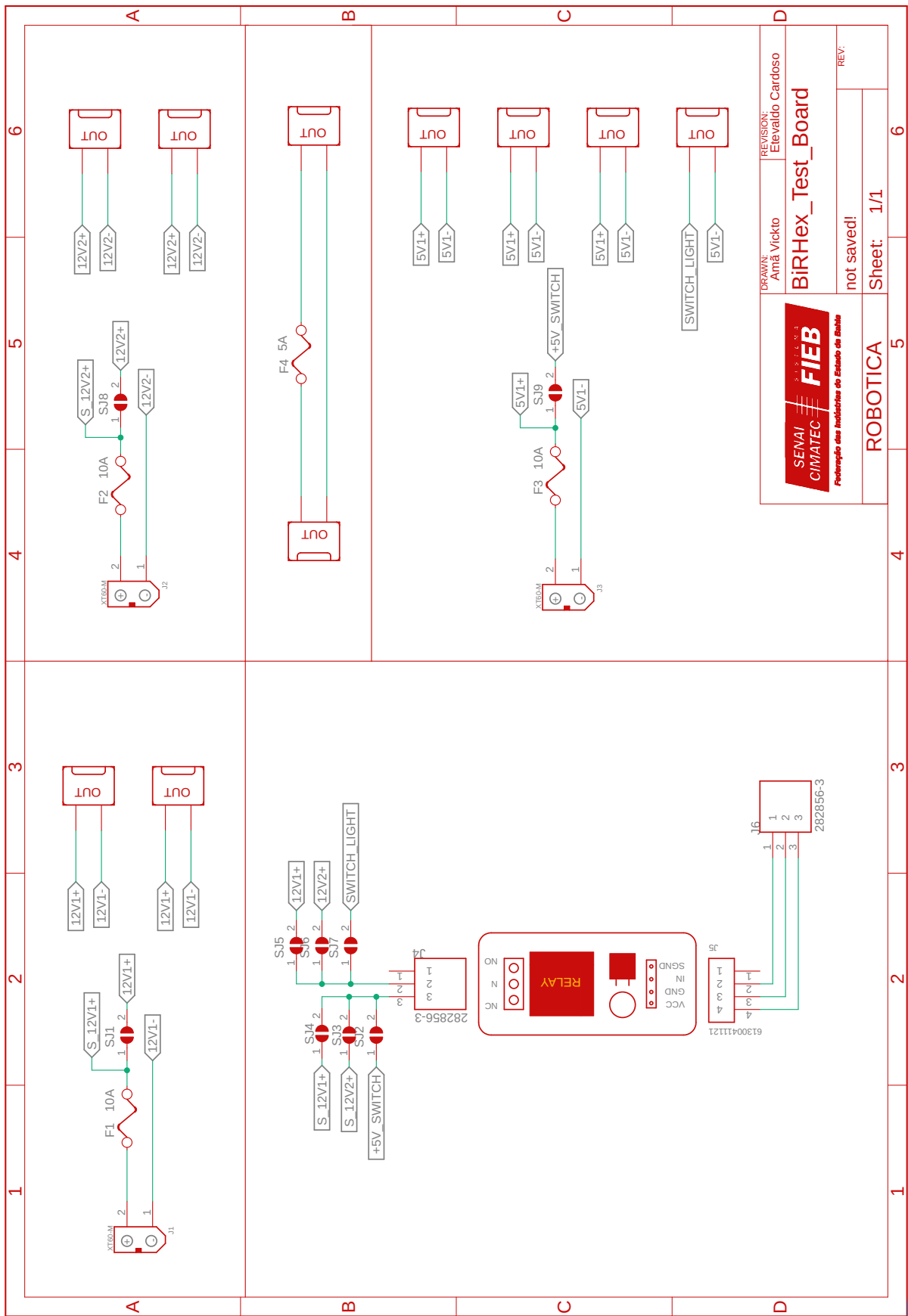
SEWAL COMATIC	BIRHEX Project	Sheet: 1/3	8
<b>FIEB</b> FIEB Engenharia e Instalações de Sistemas de Energia		not saved! REV: 1	8
DRAWN: André Faria REVISION:	<b>BIRhex schematics</b> Power Distribution	Sheet: 1/3 REV: 1	8



		DRAWN: Amir Fair REVISION:
	BIRhex schematics Smart Devices	not saved! Sheet: 2/3
BIRHEX Project		REV: 1

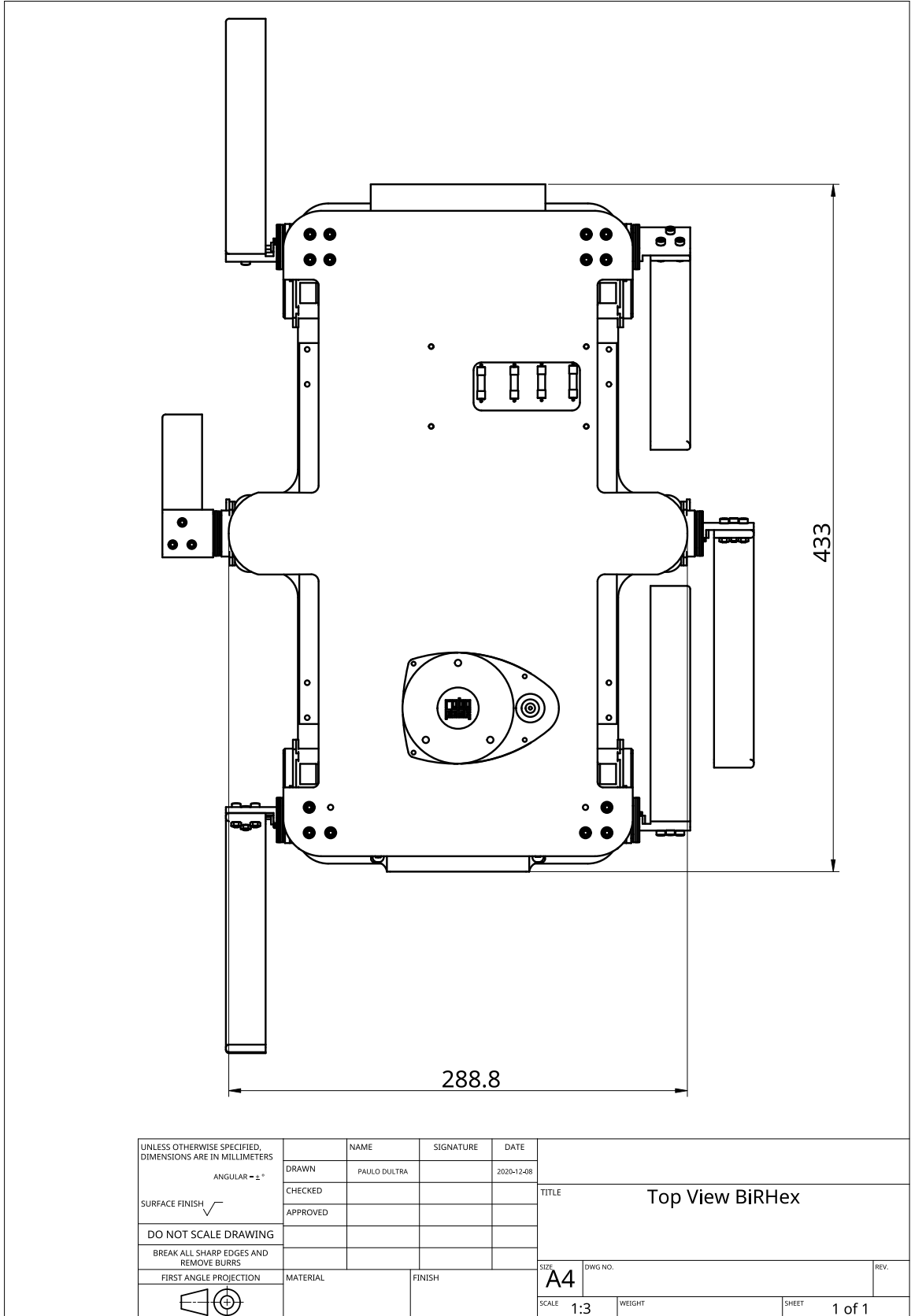


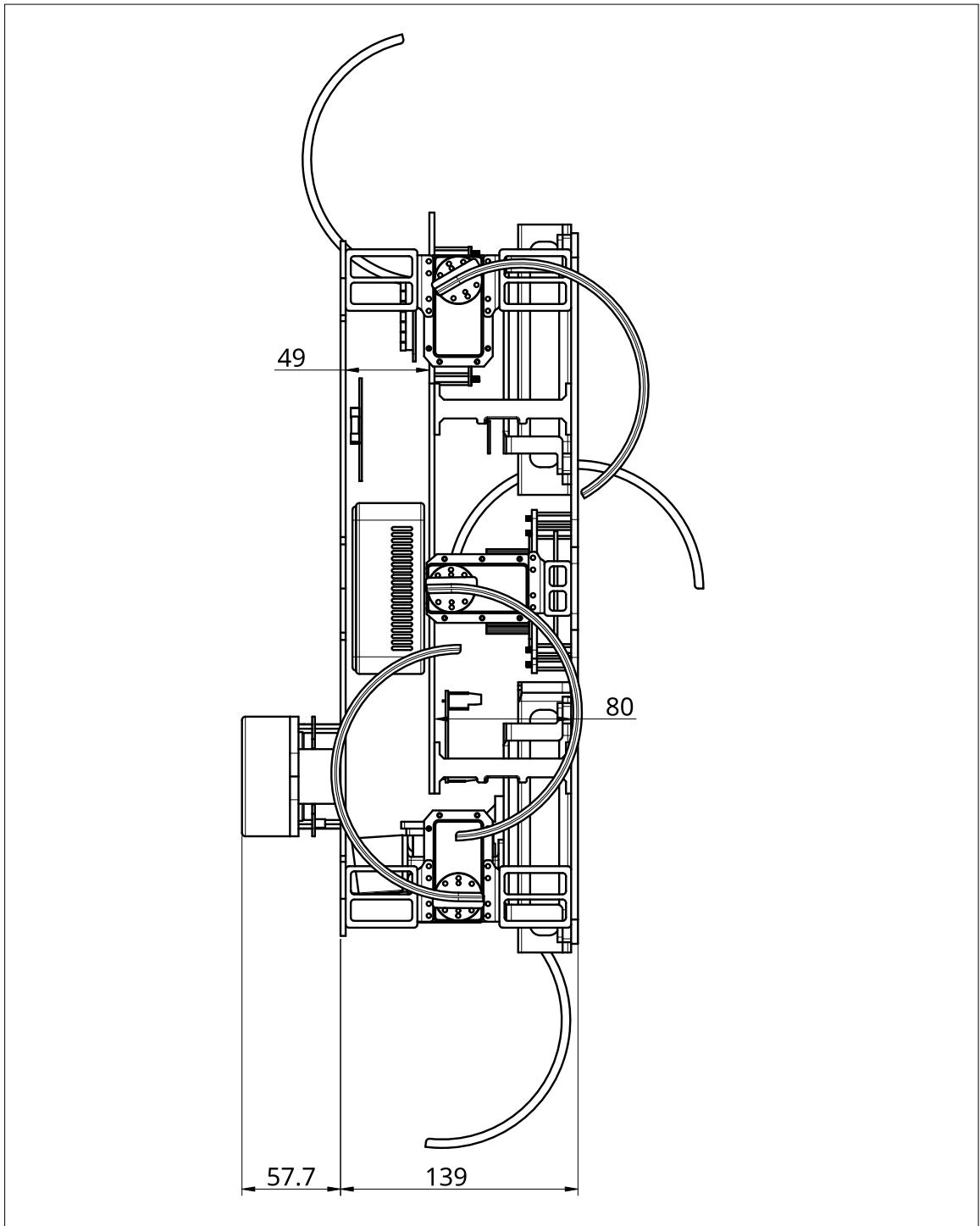
SEWAL COMATIC	<b>FIEB</b>	REV: 1
BIRHEX schematics Peripherals and Sensors		Sheet: 3/3
not saved!		8
BIRHEX Project		7
DRAWN: André Fair		REVISION:




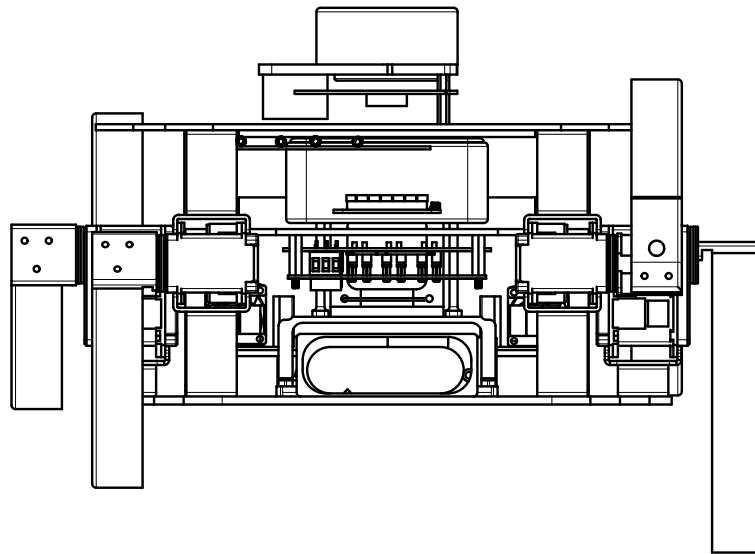
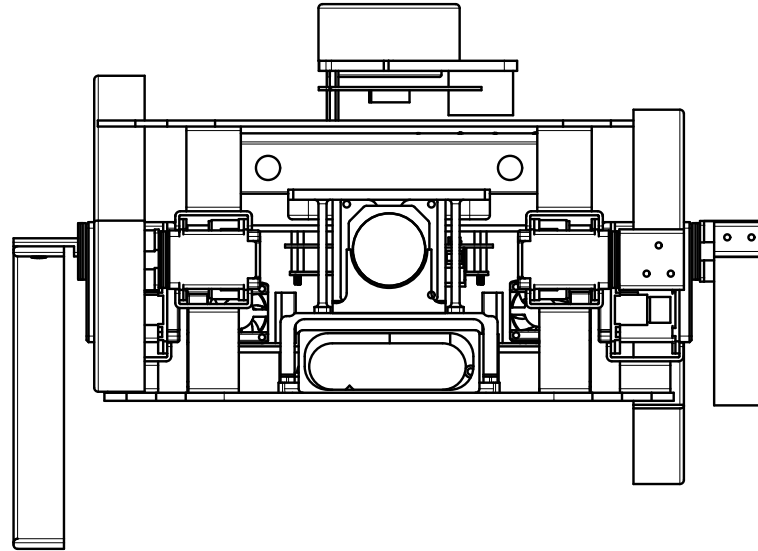
	DESIGNER: Amã Vickto	REVISION: Ezevaldo Cardoso
	<b>BIRHex_Test_Board</b>	
not saved!		REV:
ROBOTICA		Sheet: 1/1



Desenhos Mecânicos





UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = °  SURFACE FINISH $\sqrt{\quad}$  DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION  		NAME	SIGNATURE	DATE	TITLE <b>Right View BiRHex</b>		
	DRAWN	PAULO DULTRA		2020-12-09			
	CHECKED						
		APPROVED				SIZE	DIWG NO.
		MATERIAL	FINISH		<b>A4</b>		
					SCALE	WEIGHT	SHEET
					<b>1:3</b>		<b>1 of 1</b>



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ± °  SURFACE FINISH   DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION 		NAME	SIGNATURE	DATE	TITLE Front and Back View BiRHex
	DRAWN	PAULO DULTRA		2020-12-09	
	CHECKED				
	APPROVED				
	MATERIAL	FINISH		SIZE <b>A4</b>	DWG NO. _____
				SCALE 1:8	WEIGHT _____
				SHEET 1 of 1	REV. _____



---

## Referências Bibliográficas

---

A expedição norte-americana. 2016. WorldDigitalLibrary. Disponível em: <<https://www.wdl.org/pt/item/7308/>>. 1

AARONMR (Ed.). *Introduction - ROS Wiki*. 2014. Disponível em: <<http://wiki.ros.org/ROS/Concepts>>. 23, 24

AVALANCHE in Turkey wipes out rescue team: 38 dead overall. 2020. <<https://abcnews.go.com/International/wireStory/killed-avalanche-eastern-turkey-missing-68769581>>. Accessed: 2020-03-27. 4

AYERS, J.; WITTING, J. Biomimetic approaches to the control of underwater walking machines. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, v. 365, p. 273–95, 02 2007. 11

AYERS, J. et al. Lobster robots. Citeseer, 2000. 12

BELLICOSO, C. D. et al. Advances in real-world applications for legged robots. *Journal of Field Robotics*, v. 35, n. 8, p. 1311–1326, 2018. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21839>>. 5

Birch, M. C. et al. Design of a cricket microrobot. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. [S.l.: s.n.], 2000. v. 2, p. 1109–1114 vol.2. 12

BONN, U. of. *Momaro*. IEEE. Disponível em: <<https://robots.ieee.org/robots/momaro/>>. 4

CHAM, J. G.; KARPICK, J. K.; CUTKOSKY, M. R. Stride period adaptation of a biomimetic running hexapod. *The International Journal of Robotics Research*, SAGE Publications, v. 23, n. 2, p. 141–153, Feb 2004. Disponível em: <<http://dx.doi.org/10.1177/0278364904041323>>. 11

COM quase metade da população em áreas com risco de desabamento e alagamento, Salvador lidera índice nacional do IBGE. 2018. G1 GLOBO. Disponível em: <<https://g1.globo.com/ba/bahia/noticia/com-quase-metade-da-populacao-em-areas-com-risco-de-desabamento-e-alagamento-salvador-lidera-ghml>>. 3

DATTALO, A. (Ed.). *Introduction - ROS Wiki*. 2018. Disponível em: <<http://wiki.ros.org/Packages>>. 23

DAVIS, S. The biomimetic design of soft mechatronic systems. In: *Proc. IEEE Int. Conf. on Advanced Robotics, Coimbra, 2003*. [S.l.: s.n.], 2003. p. 720–725. 12

DIEGO, U. da Califórnia em S. *Robôs Bio-inspirados*. 2020. Disponível em: <<http://web.eng.ucsd.edu/ngravish/MAE207/syllabus/>>. 12

- DILLMANN, R. et al. Biologically inspired walking machines: design, control and perception. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 365, n. 1850, p. 133–151, 2007. Disponível em: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2006.1914>. 11
- DUMAN. *AUTONOMOUS UNDERWATER VEHICLE SYSTEMS*. 2019. Disponível em: <http://www.dumansystems.com/autonomous-underwater-vehicle-systems>. 11
- DYNAMIXEL. *Dynamixel™*. 2020. Disponível em: <https://www.robotis.us/dynamixel/>. 33
- ESPENSCHIED, K. S.; QUINN, R. D. Biologically-inspired hexapod robot design and simulation. In: . [S.l.: s.n.], 1994. 15
- Filho, A. B.; Amaral, P. F. S.; Pinto, B. G. M. A four legged walking robot with obstacle overcoming capabilities. In: *3rd International Conference on Human System Interaction*. [S.l.: s.n.], 2010. p. 374–379. 14
- FLIR. *FLIR ADK™*. 2020. Disponível em: <https://www.flir.com/products/adk/>. 34
- GEPPERT, L. Qrio, the robot that could. *Ieee Spectrum*, v. 41, n. 5, p. 34–37, 2004. 13
- GONZALEZ, P.; SANTOS, E. G. D.; ESTREMER, J. *Quadrupedal locomotion: An introduction to the control of four-legged robots*. [S.l.: s.n.], 2006. 21
- HARADA, E. *Drone*. 2020. Disponível em: <https://www.tecmundo.com.br/produto/149016-drone-comprar-conheca-melhores-opcoes-marca-dji.htm>. 10
- HAVE we learned enough from the 1995 Kobe quake? 2020. <https://japantimes.co.jp/opinion/2020/01/16/editorials/learned-enough-1995-kobe-quake/>. Accessed: 2020-03-28. 3
- HAYNES, G. C.; RIZZI, A. A. Gaits and gait transitions for legged robots. *Proceedings - IEEE International Conference on Robotics and Automation*, v. 2006, 2006. 21
- HENNION, B.; PILL, J.; GUINOT, J.-C. A biologically inspired model for quadruped locomotion. In: *Climbing and Walking Robots*. [S.l.]: Springer, 2006. p. 49–56. 12
- INAGAKI, K.; KOBAYASHI, H. Adaptive wave gait for hexapod synchronized walking. In: IEEE. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. [S.l.], 1994. p. 1326–1331. 14
- INTEL. *Show de Drones*. 2018. Disponível em: <https://newsroom.intel.com.br/news-releases/intel-planeja-quebrar-recorde-com-show-de-luzes-de-1500-drones-no-aniversario-de-50-anos-english-#gs.mtnthc>. 10
- JO, K. et al. Development of autonomous car—part i: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, IEEE, v. 61, n. 12, p. 7131–7140, 2014. 7
- KAGAMI, S. et al. Design and development of a legged robot research platform job-1. In: IEEE. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*. [S.l.], 1998. v. 1, p. 146–151. 14

KAJIMA, H. et al. Study on brachiation controller for the multi-locomotion robot: redesigning behavior controllers. In: IEEE. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*(Cat. No. 03CH37453). [S.l.], 2003. v. 2, p. 1388–1393. 12

KANEHIRA, N. et al. Design and experiments of advanced leg module (hrp-2l) for humanoid robot (hrp-2) development. In: IEEE. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2002. v. 3, p. 2455–2460. 13

KIM, J.-H.; OH, J.-H. Realization of dynamic walking for the humanoid robot platform khr-1. *Advanced Robotics*, Taylor & Francis, v. 18, n. 7, p. 749–768, 2004. 13

KOYACHI, N. et al. Control of walk and manipulation by a hexapod with integrated limb mechanism: Melmantis-1. In: IEEE. *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No. 02CH37292). [S.l.], 2002. v. 4, p. 3553–3558. 15

KURAZUME, R.; YONEDA, K.; HIROSE, S. Feedforward and feedback dynamic trot gait control for quadruped walking vehicle. *Autonomous robots*, Springer, v. 12, n. 2, p. 157–172, 2002. 14

LÖFFLER, K.; GIENGER, M.; PFEIFFER, F. Sensors and control concept of walking “johnnie”. *The International Journal of Robotics Research*, SAGE Publications, v. 22, n. 3-4, p. 229–239, 2003. 13

MEDERREG, L. et al. The robocoq project: Modelling and design of bird-like robot. In: CITESEER. *6th International Conference on Climbing and Walking Robots, CLAWAR*. [S.l.], 2003. p. 17–19. 12

MICHAUD, F. et al. Multi-modal locomotion robotic platform using leg-track-wheel articulations. *Autonomous Robots*, Springer, v. 18, n. 2, p. 137–156, 2005. 8

MIHELJ, M. et al. *Robotics*. [S.l.]: Springer, 2019. 6, 7

MURPHY, R. *NBC Nightly News Feature on CRASAR “Robot rescue: Behind the technology deployed for disaster relief”*. 2018. CRASAR. Disponível em: <http://crasar.org/nbc-nightly-news-feature-on-crasar-robot-rescue-behind-the-technology-deployed-for-disaster-relief/>. 5

NAVIGATION, S. *Swift Navigation*. 2020. Disponível em: <https://www.swiftnav.com>. 35

NISHIWAKI, K. et al. Design and development of research platform for perception-action integration in humanoid robot: H6. In: IEEE. *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*(Cat. No. 00CH37113). [S.l.], 2000. v. 3, p. 1559–1564. 13

NONAMI, K. et al. *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. [S.l.]: Springer Science & Business Media, 2010. 9

OMATA, T.; TSUKAGOSHI, K.; MORI, O. Whole quadruped manipulation. In: IEEE. *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No. 02CH37292). [S.l.], 2002. v. 2, p. 2028–2033. 14

ONU: Brasil está entre os 10 países com maior número de afetados por desastres nos últimos 20 anos. 2015. Observatoriodoclima. Disponível em: <http://www.observatoriodoclima.eco.br/>

onu-brasil-esta-entre-os-10-paises-com-maior-numero-de-afetados-por-desastres-nos-ultimos-20-anos  
>. 2

PAULL, L. et al. Auv navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, IEEE, v. 39, n. 1, p. 131–149, 2013. 10

PENG, S.; LAM, C.; COLE, G. A biologically inspired four legged walking robot. In: IEEE. *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. [S.l.], 2003. v. 2, p. 2024–2030. 12

PHIDGETS. *Phidgets*. 2020. Disponível em: <<https://www.phidgets.com>>. 32

PRAHACS, C. et al. Towards legged amphibious mobile robotics. *Proceedings of the Canadian Engineering Education Association (CEEA)*, 2004. 17, 18, 19, 20

RAIBERT, M. H. *Legged Robots that Balance*. 1986. MIT Press. 11

RITZMANN, R. E. et al. Insect Walking and Biorobotics: A Relationship with Mutual Benefits. *BioScience*, American Institute of Biological Sciences, v. 50, n. 1, p. 23 – 33, 2000. Disponível em: <[https://doi.org/10.1641/0006-3568\(2000\)050\[0023:IWABAR\]2.3.CO;2](https://doi.org/10.1641/0006-3568(2000)050[0023:IWABAR]2.3.CO;2)>. 11

ROBOTICS; NEWS, A. *Automated guided vehicles market size forecast to be worth 7.3 billion by 2025*. 2020. Disponível em: <<https://roboticsandautomationnews.com/2019/02/25/automated-guided-vehicles-market-size-forecast-to-be-worth-7-3-billion-by-2025/20106/>>. 7

ROBOTICS, T. *Hex MK2*. 2020. Disponível em: <<https://www.trossenrobotics.com/hex-mk2/>>. 15

ROBOTS, I. *HRP2*. 2020. Disponível em: <<https://robots.ieee.org/robots/hrp2/>>. 13

SAKAGAMI, Y. et al. The intelligent asimo: System overview and integration. In: IEEE. *IEEE/RSJ international conference on intelligent robots and systems*. [S.l.], 2002. v. 3, p. 2478–2483. 13

SARANLI, U.; BUEHLER, M.; KODITSCHKEK, D. E. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, SAGE Publications, v. 20, n. 7, p. 616–631, 2001. 16, 17, 18

SCIENCES, O. *Tracked robots*. 2019. Disponível em: <<https://www.online-sciences.com/robotics/tracks-military-technology-tracked-robots-uses-advantages-disadvantages/>>. 9

SICILIANO, B. et al. *Robotics: modelling, planning and control*. [S.l.]: Springer Science & Business Media, 2010. 6

SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011. 9

SILVAGNI, M. et al. Multipurpose uav for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk*, v. 8, n. 1, p. 18–33, 2017. 4

SLAMTEC. *SlamTec*. 2020. Disponível em: <<https://www.slamtec.com/en/Lidar/A1>>. 33

STEREOLABS. *StereoLabs™*. 2020. Disponível em: <<https://www.stereolabs.com/zed/>>. 34

SUN, J.; ZHAO, J. An adaptive walking robot with reconfigurable mechanisms using shape morphing joints. *IEEE Robotics and Automation Letters*, IEEE, v. 4, n. 2, p. 724–731, 2019. 16

Tadokoro, S. et al. The robocup-rescue project: a robotic approach to the disaster mitigation problem. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. [S.l.: s.n.]. 4

TADOKORO, S. et al. The robocup-rescue project: A robotic approach to the disaster mitigation problem. In: IEEE. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. [S.l.], 2000. v. 4, p. 4089–4094. 15

TAKITA, K.; KATAYAMA, T.; HIROSE, S. Development of dinosaur-like robot titrus - the efficacy of the neck and tail of miniature dinosaur-like robot titrus-iii-. In: . [S.l.: s.n.], 2003. v. 2, p. 2466 – 2471 vol.2. ISBN 0-7803-7736-2. 13

TECHCRUNCH. *Waymo's autonomous vehicles are driving 25,000 miles every day*. 2018. Disponível em: <https://techcrunch.com/2018/07/20/waymos-autonomous-vehicles-are-driving-25000-miles-every-day/>. 7

THAO, N. M. High-fidelity modeling of hexapod walking robot locomotion. In: *Workshop Proceedings*. [S.l.: s.n.], 2018. p. 116–123. 16

TIMES, S. *How the Roomba is making me a better person*. 2019. Disponível em: <https://www.seattletimes.com/explore/at-home/how-the-roomba-is-making-me-a-better-person/>. 8

VORPAL Home Page. [https://vorpaltrobotics.com/wiki/index.php/Vorpal\\_The\\_Hexapod](https://vorpaltrobotics.com/wiki/index.php/Vorpal_The_Hexapod). Accessed: 27/04/2020. 15

WEINGARTEN, J. D. et al. Robotics as the delivery vehicle: A contextualized, social, self paced, engineering education for life-long learners. 2007. 20, 21

XSENS. *xSens*. 2020. Disponível em: <https://www.xsens.com/inertial-sensor-modules>. 35

YOKOI, K. et al. Humanoid robot applications in hrp. *International Journal of Humanoid Robotics*, World Scientific, v. 1, n. 03, p. 409–428, 2004. 13

YONEDA, K.; OTA, Y. Non-bio-mimetic walkers. *The International Journal of Robotics Research*, SAGE Publications, v. 22, n. 3–4, p. 241–249, Mar 2003. Disponível em: <http://dx.doi.org/10.1177/0278364903022003008>. 11