UNIVERSIDADE FEDERAL DO RIO GRANDE DE SUL
ESCOLA DE ENGENHARIA - INSTITUTO DE FÍSICA

Víctor Fernandes Gandara

TRENA DIGITAL COM GIROSCÓPIO PARA RECONSTRUÇÃO DE PLANTA BAIXA DE AMBIENTES

Porto Alegre
2023

VÍCTOR FERNANDES GANDARA

TRENA DIGITAL COM GIROSCÓPIO PARA RECONSTRUÇÃO DE PLANTA BAIXA DE AMBIENTES

Projeto Final de Curso em Engenharia Física submetido à Universidade Federal do Rio Grande do Sul como requisito necessário para obtenção do grau de Bacharel em Engenharia Física.

Porto Alegre
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

VÍCTOR FERNANDES GANDARA

Esta monografia foi julgada adequada para a obtenção do título de Bacharel em Engenharia Física, sendo aprovada em sua forma final pela banca examinadora:

Orientador: Prof. Dr. Gustavo Gil da Silveira
Universidade Federal do Rio Grande do Sul

Prof. Dr. Cristiano Krug
Universidade Federal do Rio Grande do Sul

Prof. Dr. Marcelo Barbalho Pereira
Universidade Federal do Rio Grande do Sul

Porto Alegre, 6 de Abril de 2023

Agradecimentos

Gostaria de agradecer primeiramente à minha família, que sempre foi a base na qual eu me sustento e de onde retiro todo o suporte necessário. Agradeço também aos professores Gustavo Gil da Silveira, Rafael Peretti Pezzi e Cristiano Krug, que acompanharam e instruíram essa longa caminhada, ao Centro de Tecnologia Acadêmica do Instituto de física e seus integrantes passados, presentes e futuros, em especial Renan Ritter Soares, e também aos amigos e colegas que também trilharam esse caminho junto. Desses, trago especial agradecimento àqueles que durante a jornada tornaram-se também família. Por fim, gostaria de agradecer a Rogério Lima Manganelli, cujo trabalho esteve sempre presente quando do desenvolvimento deste.

Resumo

A presente monografia elabora o desenvolvimento de um dispositivo capaz de realizar medidas de distância e então, utilizando um giroscópio digital, recriar o perímetro do ambiente medido (planta baixa) através de um algoritmo de reconstrução. A hipótese a ser averiguada é a de que a adição do giroscópio como medida auxiliar permite sanar as dificuldades atuais do processo de automação da recriação de ambientes em duas dimensões presentes nos dispositivos comerciais. Por meio do projeto e manufatura de um protótipo, demonstrou-se que a adição desse sensor auxiliar remove a necessidade de entrada de informações por parte do usuário final durante a tomada de dados e, junto do algoritmo proposto, realiza corretamente a digitalização do ambiente. O desenvolvimento e produção do protótipo, assim como seu *firmware* e relação com o *software* de reconstrução, são relatados nesta monografia, descrevendo a solução proposta, seus testes e resultados como meio de averiguar a hipótese levantada e realizar uma análise de sua viabilidade como solução.

Palavras-chave: Reconstrução digital de ambientes, metrologia, sensor de distância, giroscópio, *motion-tracking device*, dispositivos digitais, algoritmo, software.

Abstract

The following engineering project proposes the design of a device capable of automating the reconstruction process of rooms using the values of perimeter measurement and a newly proposed algorithm, outputting the room's blueprint. The hypothesis studied in this work is whether the addition of a gyroscope as a mean to provide an auxiliary measurement to already estabilished devices in the market allows for the automation of the recriation of rooms in two dimensions. Through the project and the execution of this prototype it is shown that by adding this auxiliar sensor it's possible to fully remove the necessity of any data entry by the user during the measurement process and together with the proposed algorithm also draw correctly the digitalized blue print for the room. The development and production of said prototype, as well as its firmware and its relation to the reconstruction software, are all explained in this work, thus describing the proposed solution to the problem as well as the tests carried out and their results as a means to assess the main hypothesis and to analyze the feasibility of the solution.

Keywords: Digital environment mapping, metrology, distance sensors, gyroscope, motion-tracking device, digital devices, algorithm, software.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Medir com exatidão o tamanho de objetos e espaços é uma necessidade que acompanha a humanidade através de sua história. Inicialmente, partindo de métodos que utilizavam as próprias partes de seu corpo para obter uma estimativa grosseira [Klein 2011], passa-se então a utilizar ferramentas para auxiliar neste processo. Um dos primeiros equipamentos utilizados, a régua [Klein 2011], traz consigo a capacidade de ter uma métrica invariante entre usuários e começa a trazer ao mundo a necessidade de padronização de medidas como garantia da universalidade [Smeaton 2000]. Apesar de possuir diferentes implementações que aumentam sua precisão de medida, como no caso do paquímetro, as réguas e outros dispositivos com marcações fixas possuem limitações frente medidas de grandes distâncias. Para maiores distâncias de medida, o tamanho destes aparelhos aumenta e os mesmos tornam-se impráticos, não portáteis e acabam por perder seu propósito útil. Dentro de soluções portáteis, com uso simples e individual, surge a trena métrica, tendo sua forma mais comum contendo uma fita metálica enrolada de forma compacta por um sistema de molas [Bangs 1864]. Apesar de um avanço bastante importante, as trenas ainda possuem certos empecilhos em seu uso. Muitas vezes, usuários apresentam dificuldades na sua utilização autônoma para distâncias maiores que a envergadura humana, sendo necessário uma segunda pessoa ou aparato para auxiliar no processo. Além disso, o processo de averiguação da medida em si continuava idêntico àquele das réguas e, por conseguinte, pouco preciso para medidas pequenas e sempre sujeito ao erro sistemático devido ao processo depender da examinação visual simples por parte do usuário.

Junto ao desenvolvimento de dispositivos eletrônicos, por volta da segunda metade do século XX, passa-se do uso de trenas métricas com fitas para trenas digitais [Soule 1977]. Essas, por sua vez, possuem capacidades de medida bastante grandes e não necessitam de um aumento significativo em seu tamanho para alcançar maiores medidas. Sensores de diversos tipos podem ser utilizados para obter a medida de distância, cada um possuindo diferentes acurácias, valores mínimos e máximos de distância, capacidades de uso e preços. No entanto, algo comum a todos é a facilidade do processo de medida, agora digital, e a eliminação da necessidade de exame visual como método de definição da medida e, portanto, mais acuradas. Por fim, para grandes distâncias em que a medida não é feita em um piso plano e não se tem pontos de apoio suficientes, a trena digital também se torna preferível, uma vez que ela acaba por não deformar-se frente o seu próprio peso e divergir o valor medido daquele real, problema presente em trenas de fita.

Atualmente, possuímos trenas digitais que utilizam sensores de distância que eliminam completamente a necessidade de fitas [Dumas 1981]. Em decorrência disso, de sua fácil utilização e precisão, elas encontram amplo uso comercial dentro de áreas como

engenharia civil, arquitetura e design de interiores. Apesar de seu amplo uso e de sua natureza digital, há ainda uma falta de avanço por parte desses dispositivos no quesito de automação do processo de medida e reconstrução do ambiente como um todo. Visto que esse processo é suficientemente simples, repetível, com ponto inicial e final bem definidos e já utiliza de dispositivos digitais, têm-se uma oportunidade para implementar automações em diversos pontos do processo. Dentre essas áreas passíveis de automação, podemos dividi-las entre duas grandes áreas: tomada de dados e reconstrução digitalizada do ambiente. Essa primeira foca em tornar o processo atual de tomada de medidas mais fácil, ou seja, a colocação do dispositivo de medida em cada uma das paredes ou quinas para, assim, ter-se o conjunto de medidas de distância. Geralmente isso implica em reformulações do dispositivo de medida como um todo e também em mudanças no processo de medida em si. Além disso, visando restringir e reduzir o escopo do processo de realização das medidas a algo que seja inteiramente automatizável, essa abordagem acaba por acarretar em simplificações e pressupostos sobre o ambiente de medida e o processo que acabam por reduzir a utilização ampla dessas soluções em ambientes reais. Já a segunda tem como intuito resolver, de maneira automatizada, o processo de recriar o ambiente medido utilizando das medidas obtidas no processo anterior. Assim, possui maiores chances de não resultar em mudanças drásticas no processo de medida em si, visto que trata somente com os valores de saída do dispositivo de medida, e também no próprio dispositivo. Ao manter inalterado o processo de tomada de medidas, tal abordagem minimiza a necessidade de adaptação do usuário final ao novo dispositivo ou método, não alterando drasticamente seu processo de trabalho.

Esta monografia se propõe a projetar e executar a produção de um protótipo demonstrativo da automação do processo de recriação digitalizada de um ambiente. Busca-se uma maneira de implementar a recriação da planta baixa do ambiente agregando ao dispositivo o mínimo de alterações em relação aos dispositivos já presentes no mercado. Junto de um profissional da área de design de interiores, analisaram-se as características do problema e o como se dá o uso desse tipo de dispositivos para, assim, traçar um projeto de prova de conceito por meio de um protótipo. Através dele, busca-se demonstrar que a solução desenvolvida é capaz de corretamente desempenhar o processo de automação utilizando poucas modificações a nível de *hardware* do dispositivo.

Motivação

Esta monografia apresenta um projeto de uma solução de engenharia ao problema da automação da recriação digitalizada de plantas baixas a partir de dispositivos digitais de medição de distâncias. Tais dispositivos já se encontram bastante solidificados no mercado graças às suas capacidades de medida plenamente satisfatórias e estão constantemente integrando-se mais a outros instrumentos de trabalho dos seus usuários, como *tablets*, *smartphones* e computadores pessoais. Essa integração, no entanto, se dá em maior parte através do envio e estruturação das medidas realizadas pelo usuário de forma concisa em uma única interface [Bosch, Instruments]. A presente monografia argumenta que essas capacidades de conectividade e integração de sensores já presentes dão as bases materiais a nível de *hardware* para que soluções de automação sejam desenvolvidas e não sendo necessária reformulação significativa nos dispositivos já encontrados no mercado. Ainda mais, visto que os sensores de distância atualmente empregados já possuem grande precisão na sua medida de distância e que, em função da necessidade de nivelamento, muitas vezes já possuem algum sistema digital de medição de inclinação, então torna-se factível a implementação de automações do desenho da planta baixa do cômodo medido. Essa falta de automação foi trazida por um profissional da área de design de interiores ao relatar que a anotação de medidas, e o desenho inicial de plantas baixas de cômodos internos, ainda era feita de maneira manual em seu processo de trabalho. Alguns dispositivos [Bosch, ADA Instruments 2012] sanam parte desse problema, ao disponibilizarem dentro de seus aplicativos para *smartphones* e *tablets*, interfaceamento entre desenho técnico e a utilização das medidas retiradas pelo dispositivo [Instruments]. Junto desse profissional de design de interiores, averiguou-se a existência de um potencial para melhorias nesse processo por meio da criação de algoritmos que utilizassem das medidas dos sensores do dispositivo para anotar automaticamente cada uma das medidas retiradas, atribuindo-as às suas respectivas paredes. Através dessa solução, busca-se deixar inalterado o processo de tomadas de medidas; porém, obtendo-se, de forma automática e sem necessidade constante de atribuições do usuário, a planta baixa do cômodo ao final do procedimento. Têm-se como objetivo principal desta monografia o estudo da viabilidade de desenvolvimento e implementação de uma solução satisfatória ao problema de automação descrito, analisando as necessidades de adequação do *hardware* de dispositivos atuais frente as demandas do algoritmo a ser desenvolvido. Para tal finalidade, relata-se o projeto e produção de um protótipo capaz de testar tal solução e levantar dados sobre a capacidade da solução e suas limitações. Por meio desses, busca-se, então, concluir acerca da viabilidade da solução em si e também das mudanças necessárias à sua adequação como produto. Ainda mais, busca-se que esse protótipo possua o menor número possível de modificações a nível de *hardware* e de execução do processo de tomada de medidas frente os dispositivos atuais, visando não alterar fundamentalmente esse processo e assim garantir melhor adequação ao uso atual pelo usuário.

2 Concepção de solução

Como um primeiro passo para o início do projeto, buscou-se a idealização de uma solução a partir do problema e suas especificações. Assim, junto ao cliente, caracterizou-se o problema abordado e as necessidades específicas em relação ao dispositivo. Tais definições permitiram definir uma solução na forma de um dispositivo idealizado. Durante esse processo, não atentou-se para a viabilidade de projeto e construção dele, uma vez que a razão por trás dessa concepção seria justamente levantar todas as necessidades possíveis para um produto final já maduro. Partindo dessa solução idealizada, foi examinado como cada uma das suas características impactava a determinação da viabilidade do dispositivo em solucionar o problema. Através dessa análise traçou-se, então, uma solução demonstrativa que conta somente com o básico necessário para determinar a viabilidade da solução através dos testes e seus resultados. Abaixo segue a descrição em maiores detalhes de cada uma dessas etapas.

2.1 Caracterização do problema

O processo de medidas relatado pelo cliente consiste em medidas sucessivas e ordenadas de comprimentos de paredes que definem o perímetro de um recinto, podendo ser um cômodo, sala ou loja comercial. Com isso, o problema a ser resolvido se refere à recriação automatizada da geometria de ambientes internos através do uso de medidas de distância do seu perímetro e sensores auxiliares, gerando como resultado um conjunto de pontos que definem os vértices do ambiente medido. Cabe também ao escopo do problema a busca desses sensores auxiliares necessários para garantir o mínimo viável para o algoritmo ser capaz de recriar a geometria. Posteriormente, esse conjunto de vértices que definem o cômodo deverá ser utilizado para digitalizar a planta baixa do ambiente de maneira automática.

Foi relatado que essas geometrias não são necessariamente convexas, porém em sua absoluta maioria são compostos somente por ângulos retos. Dessa forma, restringe-se inicialmente seu escopo a essas geometrias compostas por ângulos internos e externos de $90°$, sendo elas convexas ou não, visto que isso compõe quase a totalidade dos casos reais de uso. Constatou-se, também, que o processo atual de medida realizado implica na medição de todas as paredes, uma a uma, mesmo em casos em que alguns dos valores poderiam ser inferidos matematicamente por restrições geométricas do ambiente. A sequência ordenada dessas medidas, como relatado pelo possível cliente, segue ou uma ordenação do tipo horária ou anti-horária em relação ao interior do ambiente. Isto é, do ponto de vista de um observador dentro do cômodo, a escolha da ordem da sequência das paredes a serem

medidas segue naturalmente uma orientação, percorrendo o perímetro do ambiente, vértice a vértice, em um determinado sentido. Ainda, segundo o profissional, o uso de barreiras para determinar o final de uma parede quando a mesma não apresenta um delimitação clara é o comportamento padrão no processo. Para o caso específico de uso por parte do cliente, escalas de medidas entre 3 m e 5 m foram relatadas como sendo suficientes para a maioria dos ambientes com os quais ele trabalha. Contudo valores maiores chegando a 10 m são possíveis, o que não é necessário para o projeto atual.

O processo de reconstrução do ambiente após a tomada de medidas tem como produto final um conjunto de coordenadas capaz de reconstruir, sem perdas de informação ou ambiguidades, o ambiente medido. Cabe ao estudo proposto neste projeto elencar maneiras de minimizar – ou até mesmo remover totalmente – a necessidade do usuário proporcionar valores ou informações de entrada para que o algoritmo possa realizar a correta reconstrução. Esse *software* deverá ser uma parte integrante do produto e, por consequência, não tem intenções de ser utilizado de maneira separada dele. Assim, o mesmo poderá utilizar pressupostos relativos à geometria do dispositivo, dos seus sensores e do processo de medida definido nesta seção para realizar este processo. Visto tal relação entre pressupostos do algoritmo e a projeto do *hardware*, têm-se que as etapas possuem caráter complementar. Dessa maneira, está embarcado ao problema proposto resolver de maneira conjunta as duas demandas.Essas necessidades de cada área para com a outra deverão ser o objeto de estudo para o levantamento das necessidades mínimas que o *software* tem para com o *hardware* e, assim, das necessidades mínimas de adições ou modificações nos *hardwares* dos dispositivos presentes atualmente no mercado.

As definições acima caracterizam o objetivo geral do projeto a respeito do processo de medida e papel do *hardware* e do *software*. Para levar tais requisitos básicos àquilo que se define como uma solução aplicável à realidade, no entanto, é necessária uma maior especificação das demandas de cada parte individual e sua relação para a comprovação, ou não, das capacidades da solução apresentada. Essas demandas particulares guiam diretamente as tomadas de decisão do projeto do protótipo, visto que delineiam de maneira mais clara como cada funcionalidade se relaciona com as capacidades finais do dispositivo. Separou-se tais demandas em termos de sua origem, podendo ser do dispositivo em si e de seu uso, ou do algoritmo de reconstrução e seu ambiente de execução.

2.2 Especificação

Abaixo seguem as demandas que não se restringem à formulação simples do problema, mas que definem sua aplicação no ambiente em que ela se propõe a ser inserida. Elas tem função também de definir mais precisamente o escopo do problema abordado e, assim, traçar garantias de que a solução proposta possui o mínimo de adequação necessária para que ela não somente resolva o problema de reconstrução, mas que também seja passível de futuramente tornar-se um produto.

A primeira série de especificações abordadas é de ordem do dispositivo, de seu uso, além do processo de tomada de medidas e do ambiente em que ele ocorre:

- No quesito tamanho, o dispositivo deve ser compacto a ponto de ser facilmente carregado e manuseado por uma única pessoa.
- Não pode ser alimentado pela rede elétrica doméstica do local.
- O processo inteiro de tomada de medidas deve poder ser desempenhado por uma única pessoa.
- O dispositivo deve repassar os dados para um outros dispositivo do usuário, como *notebook*, *tablet* ou *smartphone*.

A segunda série de especificações é de ordem do *software* de reconstrução, seu uso e sua integração com o *hardware*:

- Não há necessidade por parte do usuário que o *software* seja embarcado ao *hardware*, sendo processado por seu microprocessador. Assim, o software de reconstrução pode ser executado diretamente no computador ou *tablet* do usuário, deixando ao *hardware* somente a demanda do envio de dados crus a esse *software*.
- O resultado final do algoritmo é um desenho representativo da planta baixa do cômodo, representando sua geometria e medidas de cada parede.

Esse conjunto de especificações serve de base para traçar a ideia inicial de solução aplicada ao problema. Elas também foram utilizadas para traçar os requisitos de sistema, abordagem esta que será explicada no Capítulo 3.

A partir das especificações acima, deu-se início à concepção de uma solução idealizada. Seu propósito útil passa por delimitar um horizonte possível de desenvolvimento final do projeto como um produto maduro, não servindo como um objetivo deste trabalho.

2.3 Definição de solução ideal

Junto ao profissional da área de design, definiu-se uma solução ideal que abrange não somente as necessidades básicas, mas também aquelas de caráter de completude de um produto final e maduro de engenharia. Através dela, tornou-se possível discernir mais facilmente quais de suas características são indispensáveis como parte do estudo de sua viabilidade como solução de automação. Suas partes e componentes que não são relevantes à validade da solução serão removidas ou adaptadas para, assim, gerar uma definição de solução demonstrativa que será, então, o protótipo definido na seção 2.4 e que deverá ser projetada e desenvolvida. Para fins de clareza, elencou-se tais características dessa solução idealizada já subdividindo-as em *hardware* e *software*:

Hardware

- Capacidade de medida do sensor de distância entre 10 cm e 10 m.
- Capacidade de medida em superfícies foscas, vidros translúcidos e espumas.
- Invariabilidade da medida frente a presença de poeira suspensa no ambiente.
- Geometria que auxilie o processo de medição, tanto no nível do solo, como em ponto acima dele, com o dispositivo sendo segurado pelo usuário.
- Alimentação por meio de baterias recarregáveis.
- Dimensionamento menor que 15 cm em todas as direções.
- Invólucro de plástico removível protegendo a eletrônica e com acesso fácil à alimentação do sistema.
- Laser alinhado ao eixo de medida com frequência no visível com o intuito de demonstrar de maneira simples a presença de obstáculos.
- Indicadores do nivelamento do dispositivo em relação ao assoalho do ambiente.
- Botão de gatilho de medida.
- Indicação visual por meio de LEDs ou *buzzers* de início e finalização de uma medição individual.
- *Display* eletrônico embarcado ao dispositivo, onde poderão ser alteradas configurações, realizadas outras necessidades de interfaceamento com o mesmo e retornado ao usuário o valor da medição individual realizada.
- Envio de dados por transmissão sem fio.

Software

- Remoção total de entradas de dados auxiliares por parte do usuário para a correta reconstrução da geometria.
- Algoritmo de recriação em tempo real e integrado a outros *softwares* de desenho.
- *Software* compatível com dispositivos portáteis como *tablets* e *smartphones*.
- Capacidade de correções de medidas já realizadas por meio de interfaceamento com o dispositivo e/ou aplicação.

Algumas das especificações ideais relatadas acima são de caráter de uso e experiência do usuário. Muitas delas já encontram pleno uso por trenas digitais atualmente e, por consequência, não teriam caráter inovativo se embarcadas ao projeto. A seleção de capacidades necessárias ao projeto, utiliza como critério principal seu valor de pesquisa e inovação para com a automação do processo de recriação de ambientes. Necessidades básicas ainda deverão ser postas como essenciais, visto que seu caráter é restritivo, ou seja, sem elas não seria possível sequer realizar a construção de um dispositivo que atenda às necessidades mínimas de um usuário ou até mesmo criar um protótipo demonstrativo.

2.4 Definição de solução demonstrativa

A partir das demandas idealizadas, analisou-se a relevância de tais requisitos para uma prova inicial de conceito por meio de um protótipo demonstrativo. Através da remoção, alteração ou manutenção das características da solução idealizada, buscou-se definir os reais requisitos de um sistema com caráter de mínimo produto viável, capaz de servir de prova de conceito da solução. Cada um dos requisitos da solução idealizada foi analisado de acordo com sua necessidade para com a prova da solução em si ou sua restringibilidade frente a utilização do dispositivo. Para garantir que essas mudanças não acarretassem em uma má adequação às demandas básicas de uma prova de conceito utilizável em testes, cada mudança feita possui uma justificativa de sua alteração ou remoção na solução demonstrativa (Apêndice A). A Tabela 1 concentra de forma concisa a relação entre cada uma das demandas da solução ideal e da demonstrativa. Na coluna *Razão*, ela possui um breve resumo do motivo do descarte, modificação ou manutenção relatado na coluna *Necessidade*. Além disso, um *link* para a sub-seção respectiva dentro do Apêndice A é dado para cada requisito, sendo possível encontrar maiores detalhes sobre a razão.

Atributo	**Necessidade**	**Razão**
Capacidade máxima de medida	Parcial	Reduzida para prova de conceito (A.1.1)
Medição em diversas superfícies	Descartada	É alterada pela troca de sensores (A.1.2)
Robustez à presença de poeira	Descartada	É alterada pela troca de sensores (A.1.3)
Geometria	Total	A geometria define pressupostos e informações do processo de reconstrução (A.1.4)
Alimentação	Parcial	Reduzido à demanda de não ser alimentado diretamente pela rede doméstica (A.1.5)
Dimensionamento	Parcial	Sem restrições qualitativas, desde que o dispositivo se mantenha portátil (A.1.6)
Invólucro	Parcial	Não há necessidade de fácil acesso às baterias (A.1.7)
Laser	Descartada	Possui caráter estritamente auxiliar (A.1.8)
Indicadores de nivelamento	Parcial	Deverá possuir garantia de nivelamento ou indicativo (A.1.9)
Botão de gatilho	Parcial	Funcionalidade deverá estar presente, mas não necessita ser através de um botão (A.1.10)
Indicação de início e finalização de medida	Descartada	Poderá ser utilizado um tempo fixo (A.1.11)
Display eletrônico	Descartada	De caráter puramente estético (A.1.12)
Envio de dados sem cabo	Parcial	Poderá utilizar-se cabos para o envio (A.1.13)
Remoção total de entradas de dados	Parcial	Deverá minimizar-se a quantidade, preferencialmente eliminando totalmente (A.2.1)
Tempo real e integração com outros *softwares*	Descartada	Não são demonstrativos da capacidade de reconstrução em si (A.2.2)
Compatibilidade com dispositivos móveis	Descartada	Não é demonstrativo da capacidade de reconstrução em si (A.2.3)
Correções dentro do *software*	Descartada	Não é demonstrativo da capacidade de reconstrução em si (A.2.4)

Tabela 1 – Tabela de necessidades da definição demonstrativa ante as demandas da solução ideal. As necessidades poderão ser mantidas de maneira total ou parcial, ou ainda serem descartadas.

A partir das modificações descritas na Tabela 1, traçou-se os requisitos do sistema que deve ser projetado e produzido para fim de solucionar o problema proposto e também averiguar o grau de sua adequação através de testes. Essas características presentes na solução demonstrativa servem de base para a formulação dos requisitos de sistema, que são a espinha dorsal da metodologia de desenvolvimento escolhida e descrita no próximo capítulo (Capítulo 3).

3 Metodologia de desenvolvimento

Visando a criação do protótipo com requisitos definidos na seção 2.4, dentro de um período de quatro meses, definiu-se uma metodologia de desenvolvimento baseada em sistemas. A primeira parte dessa metodologia consistiu na concepção da solução idealizada para ter-se os requisitos mínimos da solução demonstrativa ao reduzir sua complexidade. Partindo das definições de como é o sistema e o que se espera dele, tornou-se possível utilizar uma metodologia baseada em sistemas e testes de suas partes individuais como estratégia de desenvolvimento do protótipo. Através dela, buscou-se uma maneira de assegurar o projeto de cada uma das partes que compõem o protótipo em conformidade com os requisitos traçados. Assim, dividiu-se o projeto em dois grupos, *hardware* e *software*, subdivididos em partes menores. O *hardware* teve sua subdivisão em: eletrônica e peça mecânica (também chamada de *invólucro*). O primeiro englobando todos os sensores utilizados, assim como quaisquer outras funções analógicas ou eletrônicas do dispositivo. A peça mecânica, por sua vez, se delimita às definições e desenhos técnicos do invólucro plástico que deverá conter a eletrônica. Já o *software* foi subdividido em *firmware* e algoritmo de reconstrução. O primeiro é o código embarcado ao microcontrolador do *hardware* que orquestra as funções da eletrônica do sistema e, ao fim, realiza a comunicação dos dados obtidos ao algoritmo de reconstrução. Esse, por sua vez, é o *script* que se encarrega de, fora do controlador, utilizar dos dados obtidos pelo dispositivo para reconstruir a geometria do cômodo medido. Essas quatro partes definem o sistema a ser desenvolvido.

Visando assegurar o resultado final do dispositivo como sistema de partes independentes, porém interagentes, foram definidos requisitos a nível de sistema (seção 3.2) que seriam desdobrados em requisitos individuais de cada parte. Os requisitos de eletrônica, peça mecânica, *firmware* e algoritmo de reconstrução, ao definirem em um nível mais baixo as demandas do sistema, foram averiguados por meio de uma série de testes, garantindo seu cumprimento. Uma vez cumpridos e testados todos os requisitos de baixo nível, têm-se o cumprimento dos requisitos de sistemas e passou-se à condução de testes a esse nível, visando garantir a correta integração de suas partes. Quando da não adequação de algum requisito, a respectiva parte foi reprojetada e produzida visando consertar os erros anteriores que resultaram na falha de teste. Uma vez que os requisitos de cada área, apesar de bastante compartimentalizados, possuem algum grau de dependência com as outras áreas, como é o caso do *software* com o *hardware*, algumas escolhas de projeto foram feitas tendo em vista mais de uma área.

Em termos de ordem sequencial de desenvolvimento dessas partes, por haver sobreposições entre as áreas do dispositivo nos quesitos de demandas e pressupostos, fez-se necessário ter como metodologia de desenvolvimento o projeto sequencial das partes

do dispositivo. A ordem dessa sequência foi elencada com base nas dependências de cada área em relação às demais. Isto é, áreas com nenhum ou um menor número de dependências foram projetadas primeiro. Uma vez resolvidas as dependências de uma área, seu desenvolvimento pode ser feito de maneira simultânea ao das outras áreas as quais dependiam dessa. Por questão de clareza, nos capítulos de desenvolvimento *hardware* e *software* (Capítulo 4 e Capítulo 5, respectivamente) a ordem de suas seções respeitou essa sequência, mesmo quando houve concomitância no projeto ou execução entre áreas. Também nesses capítulos, melhor se identificou cada caso onde escolhas de projeto foram tomadas levando em conta mais de uma área.

A definição da ordem de desenvolvimento teve como primeiro critério o pertencimento da parte ao *hardware* ou *software*. Iniciou-se por partes atreladas ao *hardware*, visto que o algoritmo de reconstrução e o *firmware* precisam de definições dessa etapa do projeto para que o seu desenvolvimento ocorra, como quais sensores estarão disponíveis e a geometria do dispositivo. Assim, iniciou-se o projeto pela eletrônica do sistema, visto que a peça mecânica, apesar de definir a geometria do dispositivo, tem como primeiro objetivo justamente armazenar, alinhar e proteger os sensores e demais componentes. Essas duas partes compartilham de uma definição importante ao projeto, que é a geometria e sua relação com os sensores. Como isso afeta diretamente também os pressupostos e funcionamento de ambas partes do *software*, optou-se por definir, já no início do projeto da eletrônica, não somente quais sensores seriam utilizados, mas também como seria sua relação com a geometria do dispositivo.

Cada uma das partes conta com uma seção dedicada ao relato de seu ciclo de desenvolvimento. As etapas que compõem esse ciclo são seu projeto inicial, produção do *hardware*/*software* projetados, teste e avaliação, e, mediante ocorrência de falha nos testes, reformulação da parte para adequação aos requisitos. Ao final do ciclo de desenvolvimento, espera-se ter uma versão final da parte que seja capaz de ser integrada ao sistema como um todo sem erro, capacidade essa que será atestada por testes de sistema no Capítulo 6.

A condução dos testes de cada requisito pode ser feita mediante três abordagens diferentes: verificação simples, análise de documentação e testes de medida ou funcionalidade. O primeiro é aplicado a requisitos que exigem a simples presença de algum componente ou capacidade facilmente averiguável por meio de examinação. O segundo é aplicado àqueles requisitos em que a análise de documentações acerca dos componentes envolvidos é suficiente para verificar sua adequação. Por último, testes de medida ou de funcionalidades são utilizados para confirmar a adequação de requisitos de caráter mais funcional, onde sua simples inspeção ou de sua documentação não é suficiente para obter resultados qualitativos ou quantitativos aferíveis. Abaixo, utilizou-se dessa divisão para agrupar quais requisitos pertencem a cada tipo de teste e também definir o escopo de cada um deles.

3.1 Solução implementada

Para o desenvolvimento do dispositivo como um sistema com partes individuais integradas, é necessário primeiro haver uma definição de seu funcionamento e processos. Durante o estudo inicial de dispositivos passíveis de atenderem aos requisitos descritos na seção 2.4, passo esse anterior ao projeto de qualquer protótipo, diversas ideias iniciais com diferentes níveis de maturidade foram levantadas. Inicialmente, partindo da demanda por um menor número de alterações no *hardware* frente a dispositivos já presentes no mercado, foram avaliadas soluções que utilizassem somente sensores já presentes nesses dispositivos. Assim, a utilização de um sensor de variação angular foi incluída ao projeto para obter as informações necessárias sobre a geometria do cômodo. Utilizando-se o sensor de distância e um medidor de ângulo, ou de sua variação, seria possível obter medidas de distância e ângulo e, através delas, reconstruir a geometria medida. Importante salientar que a variação de ângulo medida é aquela no plano perpendicular ao assoalho do cômodo, ignorando outros eixos de rotação. Essa escolha também passou pela constatação de que diversos dispositivos atuais apresentam componentes eletrônicos para medir seu nivelamento [Bosch]. Isso indica a viabilidade de implementação de sensores de variação angular dentro do dispositivo, indo ao encontro da minimização de mudanças no mesmo.

Dado o caráter de prova de conceito e um período de tempo de quatro meses para projeto, execução e testes do protótipo, realizou-se, já nesse ponto inicial de desenvolvimento, a escolha dos tipos de sensores de distância e variação angular. Dispositivos de implementação mais simples foram avaliados, cumprindo, assim, com um caráter mais demonstrativo. Por meio deles, buscou-se demonstrar a possibilidade de implementação da solução proposta como um todo e não do uso dos exatos componentes escolhidos como melhores implementações desse processo. Logo, para o sensor de distância, um sensor do tipo sonar ultrassônico foi escolhido e para o sensor de variação angular um giroscópio digital do tipo MEMS (*microelectromechanical system*).

3.2 Requisitos de sistema

As demandas trazidas pelo cliente em relação ao dispositivo se manifestam dentro do desenvolvimento através de requisitos de sistema. Eles residem em um nível acima daqueles de *hardware* e *software*, definindo o comportamento esperado do sistema como um todo e não de suas partes integrantes, visando garantir o cumprimento das demandas do cliente. Os requisitos dele seguem abaixo:

- **(S1):** O dispositivo deve ser portátil;
- **(S2):** O dispositivo deve contar com alimentação independente;

- **(S3):** O uso do dispositivo deve poder ser desempenhado por uma única pessoa;
- **(S4):** O dispositivo deve ter um gatilho acionável pelo usuário para a tomada de um dado individual;
- **(S5):** O dispositivo deve ser capaz de medir a distância que define o tamanho de uma parede;
- **(S6):** O dispositivo deve ser capaz de medir variação angular do dispositivo no plano perpendicular ao assoalho;
- **(S7):** Os valores de erro médio e desvio padrão do sensor de distância devem ser conhecidos;
- **(S8):** Os valores de erro médio e desvio padrão do sensor de variação angular devem ser conhecidos;
- **(S9):** O dispositivo deve enviar de forma sequencial e ordenada os dados coletados por seus sensores ao computador do usuário no momento do acionamento do gatilho;
- **(S10):** O processo de reconstrução deve utilizar somente os valores advindos do dispositivo;
- **(S11):** O resultado do processo de reconstrução deve ser o desenho da geometria do cômodo medido com anotações de distância em cada uma das arestas.

Todos os requisitos a nível de sistema foram identificados com um **S** em seu identificador único. À frente, menções feitas a um requisito de sistema serão realizados através desse identificador. A eletrônica e a peça mecânica, por serem ambos partes componentes do *hardware*, possuem um primeiro identificador **H** seguido de um identificador de sua parte individual (**E** para eletrônica e **I** para invólucro). A mesma lógica se aplica para as partes componentes do *software*, onde são primeiramente identificados pelas letras **Sw**, seguido de **F** e **A** para o *firmware* e algoritmo, respectivamente.

Uma vez traçados os requisitos a nível de sistema, realiza-se o desdobramento deles em requisitos pertencentes a cada uma de suas quatro partes integrantes. Um requisito a nível de sistema pode ter a si atrelado mais de um requisito, tanto entre suas partes como dentro dela. Tal fato serve como base para a garantia de que, havendo o cumprimento de todos os requisitos de baixo nível, os requisitos de sistema estejam cumpridos. Como última medida de garantia disso, é realizado um teste de sistema final onde se busca tornar evidente se o dispositivo é capaz de desempenhar a função para a qual ele foi projetado.

4 Desenvolvimento do *hardware*

4.1 Eletrônica

A área da eletrônica dentro do projeto se delimita à concepção de uma placa de circuito impresso contendo todos os sensores e periféricos necessários à implementação da função de coleta de dados. Seu funcionamento é definido pela coleta de dados dos sensores de distância e do giroscópio mediante o pressionamento de um botão de gatilho e o subsequente envio desses dois valores por meio de comunicação serial. Para isso, é necessário também que esteja presente na eletrônica uma unidade de processamento capaz de gerenciar todas essas funções, em especial a de envio de dados. A placa de circuito impresso define em parte a geometria do dispositivo e, principalmente, a localização de cada um dos sensores e do cabo de alimentação e envio de dados. Visto a inerente relação entre a geometria do dispositivo e os pressupostos os quais o algoritmo de reconstrução poderá utilizar para seu desenvolvimento, é de extrema importância já nessa primeira etapa de desenvolvimento atentar-se a isso.

Requisitos

Os requisitos **(S1-9)** são passíveis de serem afetados pela eletrônica do sistema e, portanto, devem possuir desdobramento nessa parte. Cada um deles será individualmente abordado e os requisitos da eletrônica traçados para garantir o cumprimento do mesmo.

- **Requisito (S1):** Para cumprir com esse requisito de sistema, que estipula que o dispositivo deve ser portátil, foram traçados os seguintes requisitos de *hardware*:
  - **(HE1)**: Todos os componentes eletrônicos devem suportar alimentação por voltagens entre 3,3 V e 5 V;
  - **(HE2)**: A eletrônica deve conter uma unidade de processamento própria;
  - **(HE3)**: A massa de todos os componentes eletrônicos e da placa de circuito impresso somados deve ser de aproximadamente 100 g.
- **Requisito (S2):** Esse requisito de sistema estipula que o dispositivo deve possuir alimentação independente, logo, foi traçado o seguinte requisito de *hardware*:
  - **(HE4)**: A alimentação do dispositivo deverá ser providenciada através de um cabo USB e assim prover entre 3,3 V e 5 V para todos os demais componentes.
- **Requisito (S3):** Estipula que o dispositivo deve poder ser utilizado por uma única pessoa. Traçou-se os seguintes requisitos de *hardware*:

  - **(HE5)**: O gatilho que inicializa a coleta de um dado individual deve poder ser ativável utilizando somente a mão que segura o dispositivo.

- **Requisito (S4):** O requisito de sistema estipula que o dispositivo deve ter um gatilho acionável pelo usuário para dar início ao processo de tomada de uma medida. Dessa forma, foi traçado o seguinte requisito de *hardware*:

  - **(HE6)**: Um botão do tipo *push-button* deverá ser utilizado como gatilho do início do processo de tomada de um conjunto de medidas.

- **Requisito (S5):** Estipula que o dispositivo deve ser capaz de medir a distância que define o tamanho de uma parede. Para isso, foram traçados os seguintes requisitos de *hardware*:

  - **(HE7)**: O dispositivo deverá conter um sensor de distância;
  - **(HE8)**: O sensor de distância deve ser do tipo sonar ultrassônico;
  - **(HE9)**: O sensor de distância deve possuir valor mínimo de medida de 2,5 m.

- **Requisito (S6):** Para cumprir com esse requisito de sistema, que estipula que o dispositivo deve possuir uma medida de variação angular, foram traçados os seguintes requisitos de *hardware*:

  - **(HE10)**: O dispositivo deve conter um giroscópio de tipo MEMS (*microelectromechanical systems*);
  - **(HE11)**: O giroscópio deve possuir no mínimo um eixo de rotação.

- **Requisito (S7):** Esse requisito de sistema estipula que os valores de erro médio e desvio padrão do sensor de distância devem ser conhecidos. Traçou-se o seguinte requisito:

  - **(HE12)**: Os valores de média e desvio padrão dos erros das medidas do sensor de distância devem ser verificados por inspeção na sua documentação ou, havendo falta de algum dos dados, averiguados por testes.

- **Requisito (S8):** O requisito estipula que os valores de erro médio e desvio padrão do sensor de variação angular devem ser conhecidos. Foram traçados os seguintes requisitos de *hardware*:

  - **(HE13)**: O giroscópio deverá ser capaz de discernir entre medidas defasadas de 45°;
  - **(HE14)**: Os valores de erro médio e desvio padrão do erro associado às medidas do sensor de variação angular devem ser verificados por inspeção na sua documentação ou, havendo falta de algum dos dados, averiguados por testes.

- **Requisito (S9):** Estipula que o dispositivo deve enviar de forma ordenada os dados coletados por seus sensores ao computador do usuário. O seguinte requisito de *hardware* foi desdobrado:
  - **(HE15)**: A eletrônica deverá contar com um cabo USB para envio de dados.

Testes

Verificação simples

- HE2: Deve-se aferir a existência de uma unidade de processamento no esquemático da eletrônica;
- HE3: Deve-se utilizar uma balança para verificar a massa final da placa de circuito impresso com seus componentes já soldados;
- HE4: Deve-se aferir a existência de uma entrada USB que permita a alimentação conjunta de todos os componentes, fornecendo 5 V a eles;
- HE5: Deve-se aferir a possibilidade do gatilho ser ativado utilizando somente da força exercida por uma única mão;
- HE6: Deve-se aferir a existência de um botão do tipo *push-button* no esquemático da eletrônica;
- HE7: Deve-se aferir a existência de um sensor de distância no esquemático da eletrônica;
- HE15: Deve-se aferir a existência de uma entrada USB que permita o envio de dados.

Análise de documentação

- HE1: Deve-se verificar que cada um dos componentes da eletrônica podem ser alimentados pelos valores de voltagem descritos – deve-se referenciar onde no *datasheet* de cada um dos componentes é explicitado esse valor;
- HE8: Deve-se verificar que o sensor de distância escolhido é do tipo sensor ultrassônico e referenciar onde em seu *datasheet* isso é esclarecido;
- HE10: Deve-se verificar que o giroscópio escolhido é do tipo MEMS e referenciar onde em seu *datasheet* isso é esclarecido;
- HE11: Deve-se verificar o *datasheet* do giroscópio para verificar a quantidade de eixos de rotação suportados e referenciar onde esse dado se encontra.

Teste de medida/funcionalidade

Uma vez que testes de medida ou funcionalidade acarretam em maior desprendimento de tempo, maior preparação e que um único teste pode averiguar a adequação de mais de requisito, eles foram divididos por teste e não por requisito.

Teste de medida do sensor de distância: Deve-se montar a eletrônica do sistema junto de um *firmware* temporário para retirada simples de medidas de distância. Deverão ser feitas 10 medidas em quatro regimes diferentes de distância: muito abaixo de 2,5 m (1,0 m), abaixo de 2,5 m (2,0 m), exatamente 2,5 m e acima de 2,5 m (3,0 m). Uma vez que são conhecidos os valores alvo para cada regime, deve ser calculado o erro de cada medição. A partir desse conjunto de erros deve ser calculada sua média e desvio padrão. Ambas as métricas devem ser calculadas também para os valores de erro absoluto. Então, deve ser feita a calibração dos valores obtidos e novamente realizado o cálculo, obtendo-se, assim, os valores finais dessa métrica para o sensor. Com isso, cumpre-se com os requisitos **HE9** e **HE12**.

Teste de medida do giroscópio: Deve-se montar a eletrônica do sistema junto de um *firmware* temporário capaz de adquirir dados do giroscópio. Deverão ser conduzidas quatro rotinas de medida. Uma rotina de medida é definida pela sequência de 17 medidas entre um intervalo de 360°. Essa quantidade de medidas é composta das nove medidas necessárias para completar uma rotação completa com incrementos de 45°, contendo o ângulo inicial de 0° e 360°. As demais oito medidas são o caminho de volta dessa rotação, realizando-a no sentido contrário e não incluindo novamente a medida em 360°. Uma vez que o incremento é conhecido e fixo, torna-se possível calcular o erro do valor de variação angular medida pelo giroscópio a cada ponto. Para garantir também a invariância frente ao sentido de rotação, das quatro rotinas, metade delas deverão ocorrer em um sentido de rotação e a outra metade em outro. Os resultados obtidos nestes testes deverão gerar uma estimativa de média e desvio padrão do erro da medida (cumprindo assim com **HE14**) e servir de base para determinar se o giroscópio conforma com o requisito **HE13**.

Execução

O primeiro passo do desenvolvimento da eletrônica do sistema se deu através da listagem de componentes necessários para cumprir com a sua função e requisitos. Os requisitos **HE2**, **HE6**, **HE7** e **HE10** exigem diretamente a presença dos seguintes componentes: Placa de prototipagem ou microcontrolador, *push-button*, sensor de distância e giroscópio, respectivamente. Para cada um deles, é descrito seu processo de escolha do componente final, demonstrando os critérios utilizados e relacionando-os à sua função dentro do dispositivo. O funcionamento básico de cada componente também é descrito, demonstrando suas capacidades e os requisitos que se espera cumprir. Como peça principal da união de todos os componentes, ao fim é explicado o processo de projeto da placa de

circuito impresso, as ferramentas usadas e os esquemáticos finais da eletrônica, junto de outras ilustrações de apoio. Como mencionado previamente, seu projeto já define a nível de sistema algumas características da geometria do dispositivo assim como disposição dos sensores e demais componentes. Dessa forma, a projeto final da placa de circuito impresso não se limita somente à simples implementação do seu circuito, mas está diretamente relacionada às projeções feitas para a peça mecânica (descritas na seção 4.2).

Placa de prototipagem

A escolha de uma unidade de processamento está diretamente relacionada aos requisitos **HE1**, **HE4** e **HE15**. São fatores de importância dessa escolha também o custo, fácil obtenção, facilidade de acesso a bibliotecas de *firmware* para os sensores de distância e giroscópio, além de suas dimensões, buscando-se a redução do tamanho do dispositivo. Placas de prototipagem são soluções que já embarcam em si diversas das demandas dessa área, como comunicação serial, alimentação e envio de dados concentrados em um único cabo USB, tamanho enxuto e a presença de uma unidade de processamento. Visto o caráter de prova de conceito e as vantagens listadas, optou-se pela escolha de uma dessas placas em detrimento de um microcontrolador ou microprocessador não integrado a uma delas.

Inicialmente, elencou-se três possíveis candidatos que apresentavam indícios de se conformarem às necessidades: NodeMCU, Arduino Uno e Arduino Nano. Dentre eles, optou-se pelo último, já que ele possui dimensões mais reduzidas, maior disponibilidade de bibliotecas dos sensores e pinos na sua face inferior, o que facilita na produção da placa de circuito impresso. Possuir um formato alongado também foi um fator determinante, alinhando-se bastante às expectativas de geometria do dispositivo final. Ele é capaz de, uma vez alimentado via cabo USB, prover 5 V aos outros componentes através de seu pino número 27 (justamente intitulado `+5V`), providenciando, assim, alimentação dentro da faixa de operação desejada e cumprindo com **HE1** e **HE4**. Ele é capaz de desempenhar comunicação serial utilizando o cabo USB que o alimenta, cumprindo com **HE15**, e possui poder de processamento suficiente para orquestrar os demais componentes e realizar essa comunicação sem haver latência significativa no processo.

Botão de gatilho

O desenvolvimento do botão de gatilho está atrelado aos requisitos **HE1**, **HE5** e **HE6**. Devido ao seu fácil pressionamento, tamanho enxuto e ampla disponibilidade, optou-se por utilizar um *push-button* de 6 mm. Por razões de diminuição de complexidade e redução de número de componentes como medida para melhor uso do espaço interno e da placa, optou-se por não implementar soluções de *hardware* para problemas de *bouncing* do botão[a]. Ficou relegado ao *firmware* do sistema, se necessário, tratar digitalmente o

[a] *Bouncing* é o processo através do qual o sinal elétrico de um botão acaba por trocar de estado de maneira não ideal. Devido à presença de peças mecânicas no botão e de dinâmicas de curta duração

sinal do botão. Para garantir que não houvesse curto-circuitos quando da abertura do botão, adicionou-se um resistor de 1 kΩ, limitando a corrente no circuito do botão a 5 mA. A presença do componente e suas características fazem com que os requisitos **HE5** e **HE6** sejam cumpridos inteiramente e que o requisito **HE1** seja cumprido para esse componente.

Sensor de distância

Os requisitos de eletrônica relacionados ao sensor de distância são: **HE1**, **HE7**, **HE8**, **HE9** e **HE12**. O sensor escolhido foi o `HC-SR04`, sendo uma solução de baixo custo e que atende aos requisitos. Ele possui capacidade de medida de até 4 m, sendo alimentado por 5 V e sendo um componente pequeno (45×20×15 mm) [ElecFreaks]. Por inspeção de sua documentação, podemos ver que ele conforma aos requisitos **HE1**, **HE7**, **HE8** e **HE9**. Esse último requisito ainda deverá ser comprovado juntamente com o **HE12** por meio do **Teste de medida do sensor de distância**.

O seu processo de tomada de medidas é bastante simples. O componente possui quatro pinos: um de alimentação (`VCC`), um de aterramento (`GND`), um de gatilho (`Trig`) e um de chegada de sinal de eco (`Echo`). Para gerar o sinal sonoro ultrassônico que inicia o processo de medida é necessário providenciar um pulso de 10 μS no pino `Trig` [ElecFreaks]. Com isso, a eletrônica embarcada ao componente produz uma rajada de oito ciclos de um sinal sonoro na faixa de 40 kHz [ElecFreaks]. Esse sinal sonoro é, então, medido de volta por um segundo sensor ultrassônico, gerando um sinal eletrônico no pino `Echo`. Utilizando a velocidade do som v como 340 m/s, pode-se traçar a relação entre a distância entre o sensor e a barreira que gerou o retorno do eco (limite da parede) por:

$$d = \frac{T_{echo} \cdot v}{2} = \frac{T_{echo} \cdot 340\,\mathrm{m/s}}{2} = T_{echo} \cdot 170\,\mathrm{m/s}, \tag{4.1}$$

onde T_{echo} é o tempo entre o final da geração da rajada e a chegada do sinal no pino `Echo` [ElecFreaks].

Giroscópio

Assim como o sensor de distância, certas escolhas acerca das especificidades do sensor já haviam sido feitas na criação de seus requisitos. Dessa forma, os requisitos da eletrônica relativos a esse componente são: **HE1**, **HE10**, **HE11**, **HE13** e **HE14**. O giroscópio escolhido foi o `MPU-6050`, que é considerado um *Motion Tracking device* (em português, dispositivo de rastreamento de movimentos). Ele possui capacidades além daquelas descritas nos requisitos, possuindo três eixos de rotação e três eixos de aceleração [InvenSense 2012].Um dos principais motivos de sua escolha foi sua ampla

nos primeiros momentos de contato, o sinal gerado pelo pressionamento acaba não transicionando automaticamente entre os estados, apresentando breves momentos onde o sinal oscila entre o estado aberto e fechado.

disponibilidade e fácil acesso, somados ao fato do componente se demonstrar apto a cumprir com seus requisitos. Devido à maneira como giroscópios MEMS são produzidos, torna-se difícil e economicamente inviável encontrar componentes com somente um eixo de rotação, sendo esse o motivo da escolha pelo componente com três eixos.

O `MPU-6050` é capaz de suportar alimentação de 5 V e realiza sua comunicação de dados por meio de protocolo I2C [InvenSense 2012]. Para isso, ele utiliza dois pinos, sendo necessários mais dois outros pinos para alimentação (`VCC` e `GND`). O Arduino Nano utilizado possui suporte a esse protocolo. Uma vez que a eletrônica embarcada do `MPU-6050` não executa a integração de seus sinais crus de saída em valores de ângulos, é necessário o uso de uma biblioteca externa que realize esse processo.

Placa de circuito impresso

A partir da seleção dos componentes, foi projetado o esquemático da eletrônica do sistema. Primeiramente, junto do *datasheet* de cada componente, foi realizado um desenho esquemático que ilustrasse a conexão de cada um junto do Arduino Nano, já definindo sua pinagem (Figura 1).

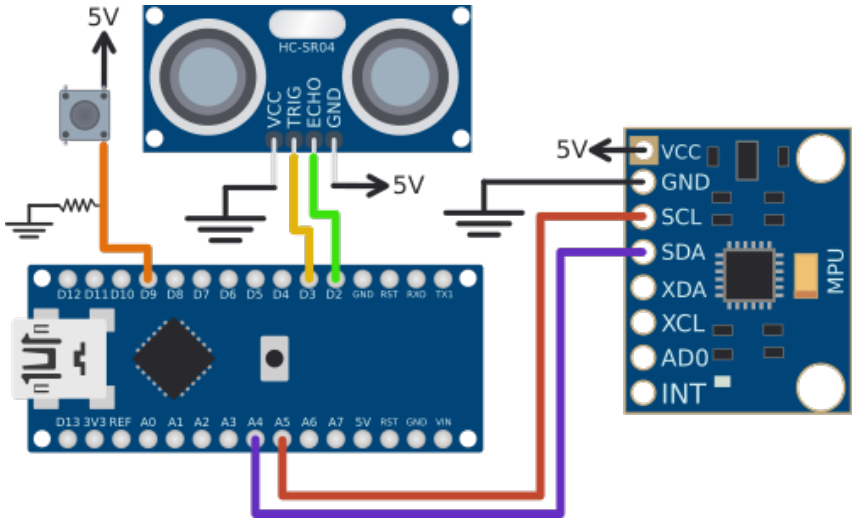

Figura 1 – Representação ilustrativa da eletrônica do sistema (proporções fora de escala).

Com ele, foi desenvolvido um esquemático da eletrônica utilizando a ferramenta KiCAD (Figura 2). Ela é uma ferramenta de design auxiliado por computadores, do inglês CAD (*computer-aided design*). Ele agrupa todos os componentes citados e desenha suas ligações, além de evidenciar todos os pinos de cada um. Esse esquemático serve de base dentro do KiCAD para referenciar as ligações entre cada componente durante o processo de criação de um esquemático de placa de circuito impresso, averiguando que não haverá erros e automatizando parte do processo de desenho de trilhas.

Desse modo, criou-se o esquemático da placa de circuito impresso dentro da ferramenta. Nele, são traçadas as trilhas que ligam cada componente, além de definir o

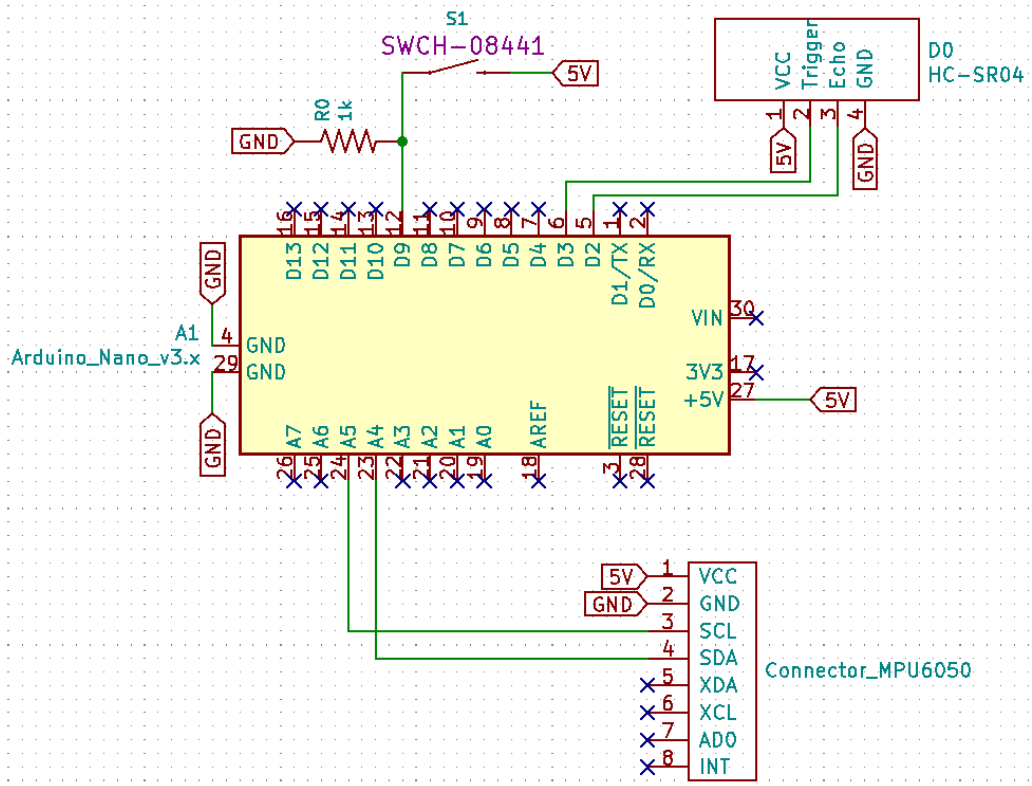

Figura 2 – Esquemático da eletrônica do sistema projetado na ferramenta KiCAD.

formato da placa e a localização de cada componente nela. Naturalmente, uma vez que esse passo já começa a definir posições de componentes e um formato aproximado do dispositivo final, certas escolhas em seu *design* foram feitas já com as demais partes do sistema em mente. Por exemplo, visando que o dispositivo final fosse mais longo do que largo ou alto, decidiu-se por colocar o Arduino Nano na extremidade da placa, com a sua saída de alimentação/envio de dados virada para a extremidade da placa. Além de garantir um acesso mais fácil a esse conector, isso faz com que a dimensão mais longa do Arduino Nano esteja alinhada com dimensão mais longa da placa/dispositivo. Em seguida, colocou-se o giroscópio ao final da outra extremidade do Arduino Nano, mantendo ainda o formato da placa alongado. Por fim, em uma escolha integrada à parte de Peça Mecânica do sistema, decidiu-se que a conexão do sensor de distância seria feita não diretamente na placa, mas, sim, através de fios soldados em barramentos conectados à placa. Como será melhor explicado na seção 4.2, essa escolha foi feita de maneira a permitir que, através da presença de fios e não de um barramento fixo, seja possível retirar e colocar com mais facilidade o sensor dentro do invólucro. A nível de eletrônica e esquemático, não há diferença entre o uso de barramento e a conexão direta desse componente na placa de circuito impresso. Uma vez que o botão de gatilho deve possuir conexão com a placa de circuito impresso, porém ao mesmo tempo deve estar na região externa do invólucro, fez-se necessário também realizar a sua conexão por meio de barramentos. Por ser uma conexão bastante pequena geometricamente e por necessidades de conexão de trilhas da eletrônica

da placa, optou-se colocar os conectores do botão no espaço entre o Arduino Nano e o giroscópio. Um desenho ilustrativo do resultado final da placa de circuito impresso e da disposição de seus componentes nela pode ser visto na Figura 3, que também possui o tamanho projetado para a mesma.

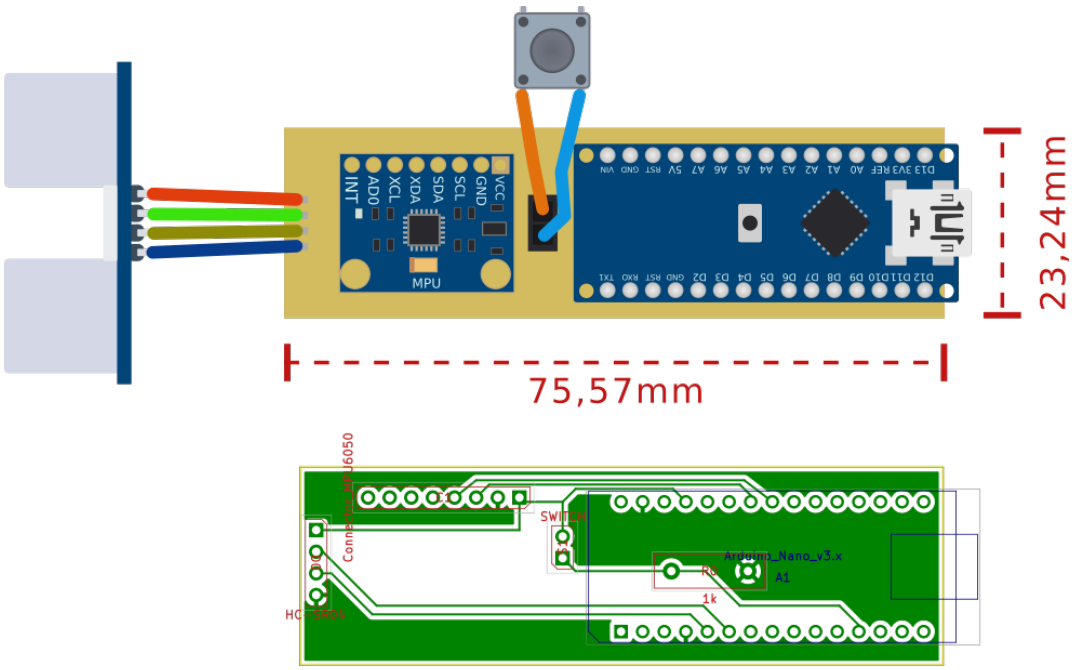

Figura 3 – Desenho representativo da eletrônica do sistema já montada na placa de circuito impresso com medidas estimadas pelo KiCAD. Abaixo, uma representação das trilhas de cobre que ligam os componentes exportados diretamente do KiCAD. Note que o resistor se encontra soldado entre a placa e o Arduino Nano, assim, utilizando melhor o espaço superior da placa.

Solução

Como maneira de averiguar a conformidade da eletrônica aos seus requisitos, a placa de circuito impressa projetada foi produzida no Centro de Tecnologia Acadêmica do Instituto de Física (CTA). Para isso, utilizou-se a Fresadora João-de-barro [Postal], fresando a placa em uma chapa de fibra de vidro com cobre em uma de suas faces. Em seguida, os componentes foram individualmente soldados à placa diretamente ou por meio de conectores ou fios e, então, uma camada de estanho foi adicionada às trilhas da placa manualmente para evitar sua oxidação. Por fim, com o auxílio de um multímetro, atestou-se que não haviam curtos-circuitos ou conexões indevidas entre as trilhas. Cronologicamente, testou-se primeiro somente a capacidade dos sensores de coletar seus respectivos dados. Para a condução dos testes, já era necessário a produção da peça mecânica, uma vez que é ela que alinha o sensor de distância. Com isso, primeiro garantiu-se a funcionalidade básica da obtenção de dados da eletrônica para, então, fazer o projeto e a produção do invólucro descrito na seção 4.2. Posteriormente, testou-se a eletrônica do sistema de acordo com seus

requisitos. Essa inversão entre a ordem do desenvolvimento e das seções desse capítulo são frutos da inerente concomitância do processo de projeto e produção dessas duas áreas, não alterando substancialmente a ordem lógica do desenvolvimento nem invalidando o processo de testes dessa primeira área.

O primeiro teste conduzido foi o **Teste de medida do sensor de distância**. O ambiente escolhido para os testes de todas as medidas foi uma mesma parede com mais de 3 m. O dispositivo foi colocado em cima de uma mesa com um suporte que o elevasse e garantisse seu nivelamento. Com uma fita métrica mediu-se a distância alvo de cada um dos regimes de medida e o dispositivo foi posicionado de forma que o extremo contrário ao sensor de distância permanecesse no limite dessa medida. Essa conformação serve para que a calibração das medidas respeitasse a lógica de uso do dispositivo, que será colocado com essa extremidade rente à quina que delimita o final de uma ou duas paredes. Dessa forma, é um resultado esperado do teste que uma calibração linear por meio de uma função $f(x) = ax + b$ resulte em um valor de b maior que 0 cm e próximo do tamanho do dispositivo, justamente contabilizando por essa distância entre a posição do sensor de distância e o final do dispositivo.

Uma vez coletados os dados de cada regime, utilizou-se um modelo de regressão linear para estimar a melhor curva de calibração. A função, com os valores de coeficiente e interseção estão explicitados na equação abaixo:

$$f(x) = ax + b = 0{,}98x + 24{,}26\,\text{cm}. \tag{4.2}$$

Visualmente, o efeito da calibração pode ser visto na Figura 4. Como podemos ver, ambos valores são razoáveis, especialmente o valor b, que se enquadra dentro da previsão feita anteriormente acerca de seu valor. As métricas de média e desvio padrão do erro para antes da calibração e após a calibração, contando também com os valores de erro absoluto, podem ser vistos na Tabela 2.

	Média	Desvio Padrão	Média (abs.)	Desvio Padrão (abs.)
Sem calibração	$-19{,}79$	4,09	19,79	4,09
Com calibração	$-3{,}55 \times 10^{-14}$	3,66	2,70	2,44

Tabela 2 – Métricas do sensor de distância antes e depois do processo de calibração. A demarcação *(abs.)* indica que os valores de erro absoluto foram utilizados para o cálculo. Todas as unidades estão em cm.

Utilizando a Tabela 2, a Figura 4 e o *datasheet* do componente, afere-se a adequação do sensor de distância aos seus requisitos. O requisito **HE1** pode ser averiguado para esse componente na primeira página do seu *datasheet* [ElecFreaks], assim como o requisito **HE8**. Por fim, os requisitos **HE9** e **HE12** também são cumpridos, uma vez que podemos perceber, por meio dos resultados do teste, que o sensor possui capacidade de medir até 2,5 m e que são conhecidas suas métricas de erro.

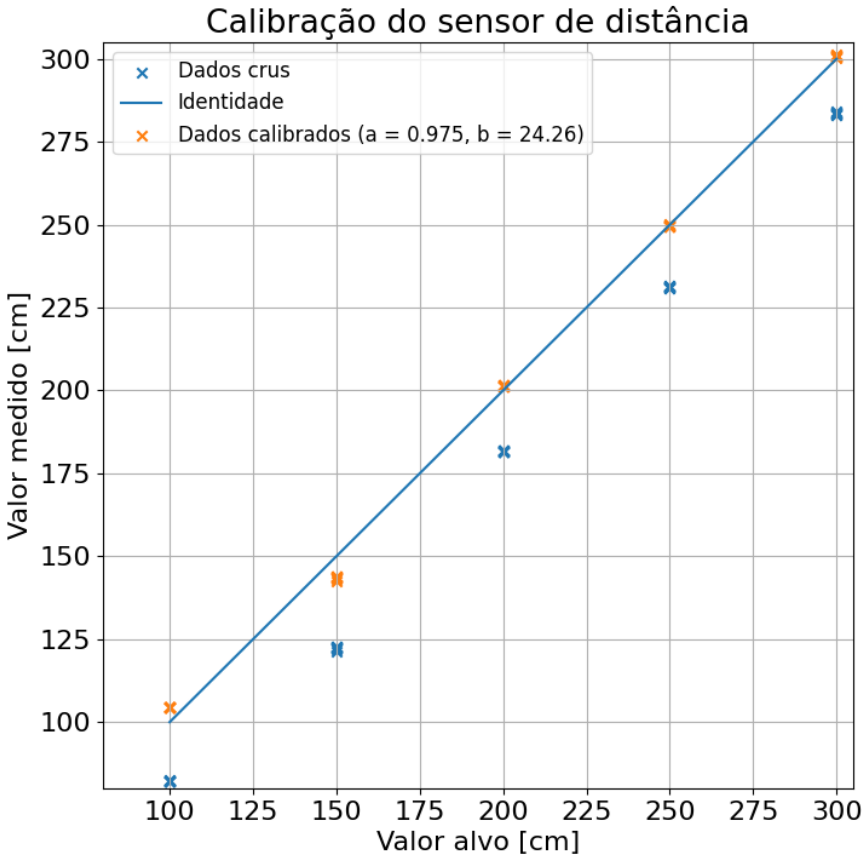

Figura 4 – Gráfico demonstrativo do resultado da calibração. A função identidade em azul ilustra o comportamento esperado de um sensor perfeitamente calibrado.

Em seguida, realizou-se o **Teste de medida do giroscópio**. O ambiente de testes foi criado utilizando uma superfície de formato quadrado com 70 cm de lado. Nela, foram traçadas com fita adesiva linhas ligando suas extremidades, como na Figura 5. Dessa forma, ao colocar o dispositivo ao final de uma marcação, com seu sensor de distância virado para o centro da superfície, foi possível garantir 45° de defasagem entre cada medida. Esse formato de testes também se assemelha ao processo de medida desempenhado pelo usuário, uma vez que a rotação não ocorre inteiramente no eixo central do dispositivo, mas envolve o seu translado entre os pontos de medida. Esse processo está também ilustrado na Figura 6, onde é possível ver um exemplo de rotina de medidas para o sentido positivo de rotação (anti-horário).

Com quatro rotinas de medidas, duas no sentido inicial positivo e duas no negativo, utilizou-se os dados angariados para analisar sua divergência para com os valores esperados. Os valores de média, desvio padrão e erro máximo e mínimo para os valores absolutos e não absolutos são mostrados na Tabela 3. Como é possível averiguar, o erro associado ao sensor permite que o mesmo seja usado sem problemas dentro do processo de reconstrução,

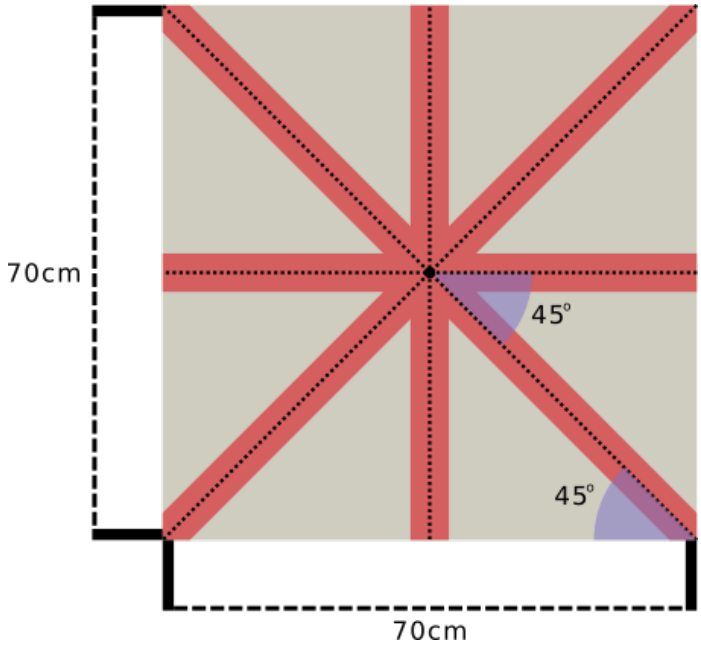

Figura 5 – Marcações feitas na superfície onde foram conduzidos os teste de medida do giroscópio. Utilizando-se do fato de que a superfície possui formato quadrado, garantiu-se por meio das marcações que o ângulo entre cada medida subsequente possui de 45° de diferença.

cumprindo com **HE13** e **HE14**. Além do mais, na página 7 de seu *datasheet* [InvenSense 2012] é possível averiguar seu cumprimento com **HE1**, **HE10** e **HE11**.

	Média	Desvio Padrão	Máximo	Mínimo
Não absoluto	0,15	2,01	4,09	−5,33
Absoluto	1,59	1,22	5,33	0,04

Tabela 3 – Métricas de erro (em graus) do giroscópio geradas a partir dos dados do **Teste de medida do giroscópio**.

Por fim, para ser possível cumprir com todos os requisitos da eletrônica, ainda é necessário aferir aqueles delegados à verificação simples. Como vimos em seção 4.1, **HE2** se encontra cumprido, uma vez que possuímos uma unidade de processamento no Arduino Nano. Como último requisito ainda não cumprido, pesou-se a placa em uma balança digital de precisão e verificou-se um valor de massa de 36 g, cumprindo com **HE3**. Dessa forma, a eletrônica do sistema cumpre com todos os requisitos do seu desenvolvimento e passa-se às próximas partes do sistema.

4.2 Peça mecânica

Esse componente caracteriza-se pela peça mecânica que engloba e protege a eletrônica do sistema enquanto também define a geometria do dispositivo. Os requisitos de sistema **(S1)**, **(S3)** e **(S4)** são os requisitos que serão desdobrados. De maneira geral, eles

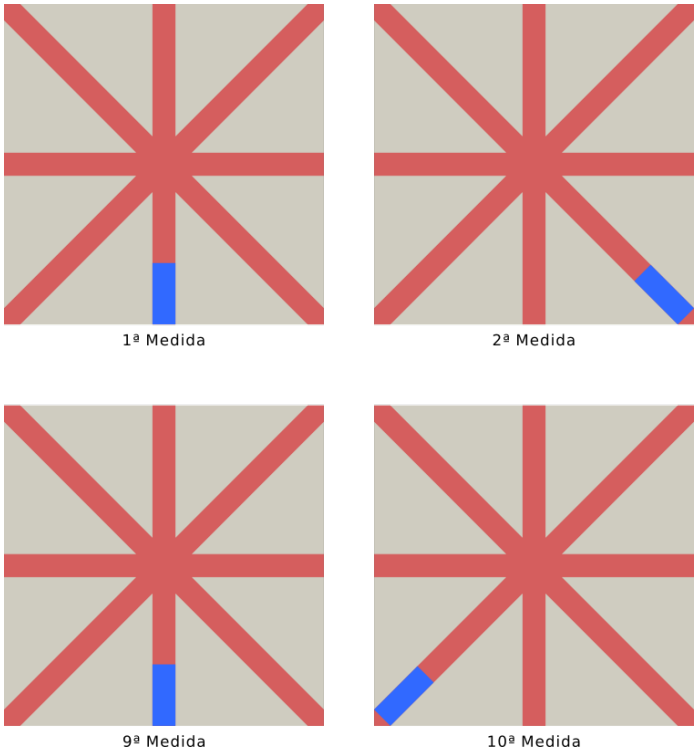

Figura 6 – Exemplificação ilustrada do processo de tomada de dados para o teste no sentido positivo de rotação, com o dispositivo em azul. As medidas omitidas entre a 2^a e a 9^a seguem a ordem estabelecida. A 9^a e 10^a medida foram ilustradas para tornar evidente como e quando ocorre o processo de volta, repetindo os pontos anteriores agora no sentido de rotação contrário, até retornar ao ponto de origem das medidas novamente, totalizando 17 medidas por rotina.

tratam da geometria do dispositivo e de como ela afeta o uso do aparelho e os pressupostos acerca disso, que o algoritmo de reconstrução poderá utilizar.

Requisitos

- **Requisito (S1):** Para cumprir com esse requisito de sistema, que estipula que o dispositivo deve ser portátil, foram traçados os seguintes requisitos do invólucro:
  - **(HI1)**: O invólucro deve ser feito de material plástico;
  - **(HI2)**: O invólucro deve ter massa menor do que 200 g;
  - **(HI3)**: O invólucro deve possuir peças removíveis para acesso ao seu interior;
  - **(HI4)**: O invólucro deve englobar inteiramente a eletrônica do sistema.
- **Requisito (S3):** O requisito de sistema estipula que o uso do dispositivo deve poder ser desempenhado por uma única pessoa. Assim, foram traçados os seguintes requisitos de *hardware*:

  - **(HI5)**: O cabo de alimentação e transferência de dados deve ser localizado de forma a permitir que as laterais e traseira do dispositivo possam ser colocadas rente às paredes do cômodo;
  - **(HI6)**: Todas peças removíveis durante a montagem e desmontagem do dispositivo devem ser fixáveis, de forma a não haver peças soltas durante o seu uso.

- **Requisito (S4):** Para o cumprimento desse requisito de sistema, que estipula que o dispositivo deve ter um gatilho acionável pelo usuário para a tomada de um dado individual, traçaram-se os seguintes requisitos de *hardware*:

  - **(HI7)**: O botão de gatilho deve ser colocado na região exterior do invólucro;
  - **(HI8)**: O botão de gatilho deve estar fixado de maneira a minimizar a mudança do ângulo de medida do sensor de distância em relação ao assoalho quando do seu pressionamento.

Testes

Assim como nos testes de eletrônica, cada requisito pode ser testado mediante as mesmas três diferentes abordagens: verificação simples, análise de documentação e testes de medida ou funcionalidade. Utilizou-se essa divisão para organizar os testes a seguir.

Verificação simples

- HI1: Deve-se aferir que o material utilizado para a produção da peça é plástico;
- HI2: Utilizando-se uma balança, deve-se medir a massa da peça e verificar que está dentro do limite estipulado;
- HI3: Deve-se aferir que é possível inserir e remover a eletrônica do sistema através da remoção de peças móveis do invólucro;
- HI4: Deve-se aferir que todas as partes da eletrônica do sistema estão fixadas e englobadas pelo invólucro, com exceção do botão de gatilho e da área ativa do sensor de distância;
- HI6: Deve-se averiguar que todas as peças, tanto do invólucro como da eletrônica, são fixáveis quando o dispositivo está montado;
- HI7: Deve-se aferir que o botão de gatilho se localiza em uma região externa da peça.

Análise de documentação

Não foram elencados testes de análise de documentação para essa área. Isso decorre, principalmente, do fato de que a peça será inteiramente projetada desde seu princípio, sem a utilização de outros componentes já prontos e, portanto, com documentação.

Teste de medida/funcionalidade

Teste de nivelamento

Como os requisitos **HI5** e **HI8** são ambos referentes ao nivelamento do dispositivo, foi traçado um teste para averiguá-los. Nele, inicialmente, serão feitas 10 medidas de um valor conhecido de distância com o dispositivo apoiado em cima de uma superfície firme. Esse conjunto de dados é considerado como sem deflexão devido ao aperto do botão, uma vez que o dispositivo se encontra devidamente apoiado. Em seguida, sem o auxílio desse apoio, o teste é conduzido novamente, dessa vez com o operador segurando o dispositivo em sua mão enquanto aperta o botão com essa mesma mão. O *firmware* e *software* de leitura de dados serial utilizados não devem implementar calibração do dispositivo, utilizando somente os valores crus captados pelo sensor de distância. Ao final, deve-se utilizar um teste de hipótese estatístico ou o levantamento de um intervalo de confiança para determinar a conformidade com o requisito **HI8**. O local onde ocorre essa medida deve ser a interseção entre duas paredes de forma a gerar um ângulo de 90°. Isso é feito de maneira concomitante às medições anteriores para conduzir também testes de caráter qualitativo acerca da adequação da geometria do invólucro ao requisito **HI5**. Assim, testa-se se a posição da saída do cabo de alimentação e transferência de dados está em um local propício e não apresenta empecilhos ao processo.

Execução

O processo de desenvolvimento da peça mecânica foi iniciado pela adequação aos requisitos **HI3** e **HI4**. Primeiramente, utilizando outra ferramenta do tipo CAD, o FreeCAD [Riegel], fez-se o design de um retângulo de dimensões alguns milímetros maior que as dimensões planares do sensor de distância (45×20 mm). Esse retângulo sofreu uma extrusão ao longo do eixo normal de sua dimensão planar. O dimensionamento dessa extrusão, 170 mm, foi escolhido tendo em mente o tamanho da placa de circuito impresso, somado ao tamanho remanescente do sensor de distância e de um espaço extra para a saída do cabo de alimentação. Visando uma maior facilidade de acesso à eletrônica do sistema quando do desmonte da peça, ela foi tornada oca e foram removidas as suas faces frontais e inferiores, como podemos ver na Figura 7. Doravante, essa peça será chamada de *case*.

Visando uma maior facilidade de montagem e desmontagem, optou-se por projetar

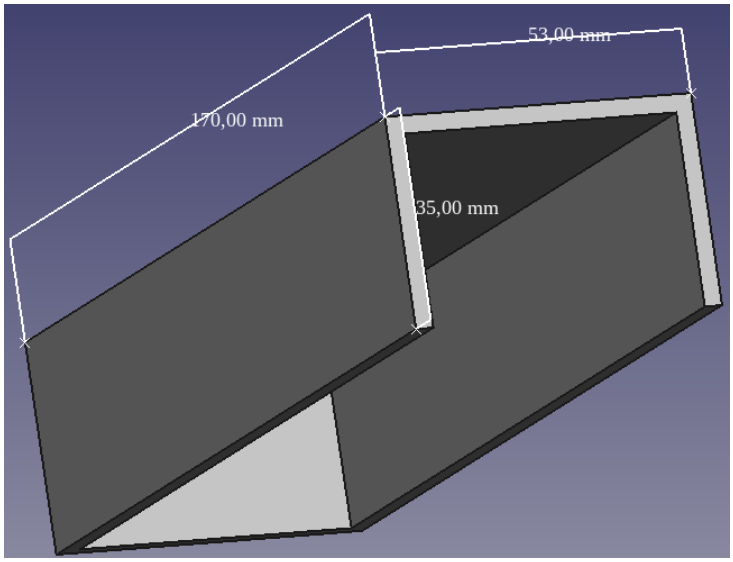

Figura 7 – Formato inicial da tampa da peça mecânica. A espessura de todas as faces é de 3 mm.

uma peça móvel somente para suportar e encaixar o sensor de distância. Essa peça, doravante chamada de *slider*, é fixada ao restante do invólucro por meio de uma canaleta escavada na região interna da ponta da peça. Como podemos ver na Figura 8, ela é um retângulo plano com uma abertura para os dois sensores ultrassônicos e buracos para fixação por meio de parafusos. O seu formato interno é pensado de maneira a permitir que as faces da peça e da eletrônica do sensor se encostem inteiramente.

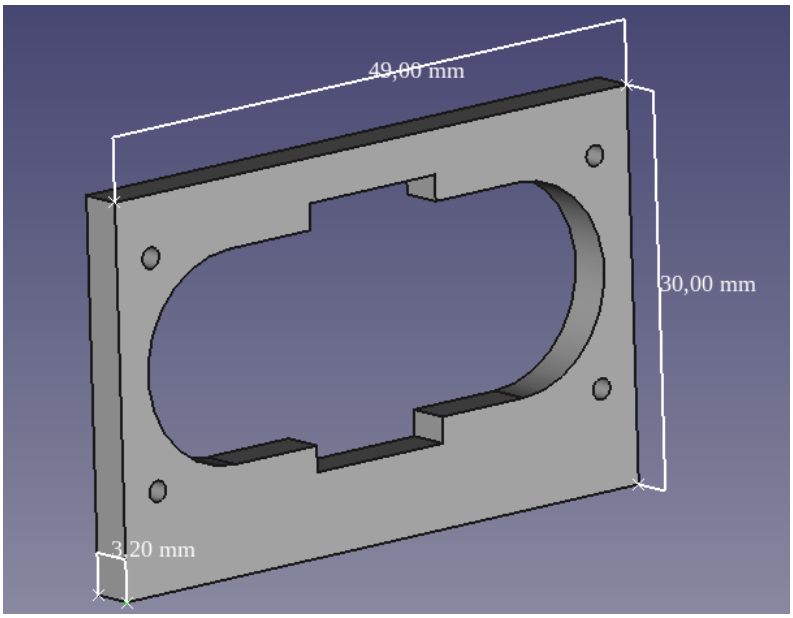

Figura 8 – Formato e medidas do *slider*.

A Figura 9 ilustra a canaleta adicionada ao *case* para fixar o *slider*. Suas dimensões são ligeiramente maiores que aquela do *slider* para que exista uma pequena folga. Ela não altera o alinhamento do sensor e permite com que a remoção do mesmo se dê maneira

mais fácil, evitando a necessidade do uso de força e possíveis danos. O valor de 13,50 mm foi escolhido de forma a que o sensor não ficasse com seus sensores de distância para fora do *case*, assim evitando possíveis colisões e danos.

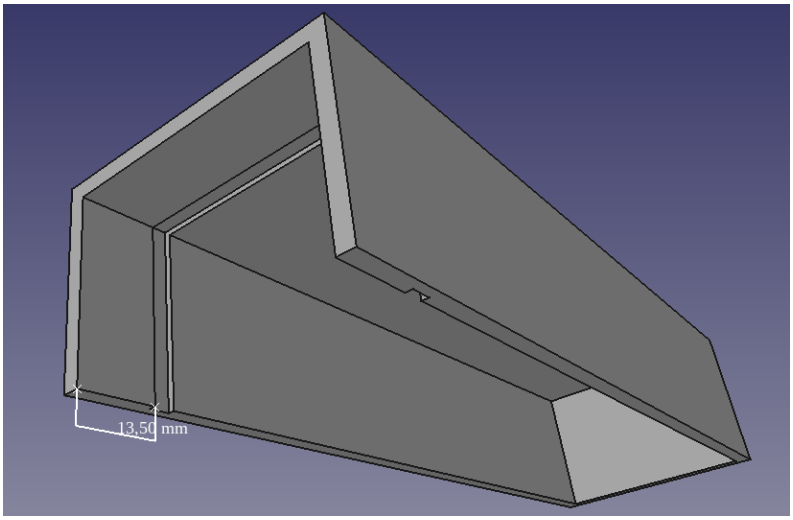

Figura 9 – *Case* com adição da canaleta de suporte para o *slider*.

Além do sensor de distância, o botão de gatilho é o único outro componente eletrônico que ficará localizado em uma região externa ou aberta da peça mecânica. Assim, passou-se para o projeto de sua localização no *case*. Visando a minimização de qualquer movimento do dispositivo ao ser pressionado o botão, optou-se por colocá-lo em uma região mais próxima do centro do dispositivo. Além disso, essa escolha facilita também o uso do dispositivo por qualquer uma das mãos do usuário. O botão do tipo *push-button* escolhido possui quatro pinos, sendo que os dois pinos de cada lado definem uma ligação. Assim, criou-se um espaço de baixo relevo com o formato do botão no meio da face superior do *case* e duas pequenas fendas foram abertas em duas de suas laterais, transpassando até o lado interno, como pode ser visto na Figura 10.

Havendo o *case* e o *slider*, ainda é necessário uma peça que tampe a parte inferior do dispositivo e fixe a eletrônica nela. Tal posição faz com que a peça que segura a eletrônica possa ser inteiramente removida, permitindo total e irrestrito acesso ao circuito. Isso reduz estresses mecânicos e possíveis danos por aplicação de força à placa de circuito impresso durante a fixação dela à peça e também quando da sua fixação junto ao *case*. Assim, como primeira parte do projeto dessa peça, iniciou-se com um retângulo de 3 mm de altura. A ele foram adicionados pequenos suportes de 3,18 mm nas laterais, na posição onde se localiza a canaleta no *case*. Além de dar um ponto extra de encaixe à peça, isso também garante um melhor alinhamento do sensor de distância. A extremidade oposta, possui um orifício de 6 mm de raio onde é possível passar o cabo USB. Essa primeira parte do design da peça, doravante chamada de *tampa*, se encontra ilustrada na Figura 11.

Para fixar essa peça ao restante do invólucro, pequenos parafusos de fixação são

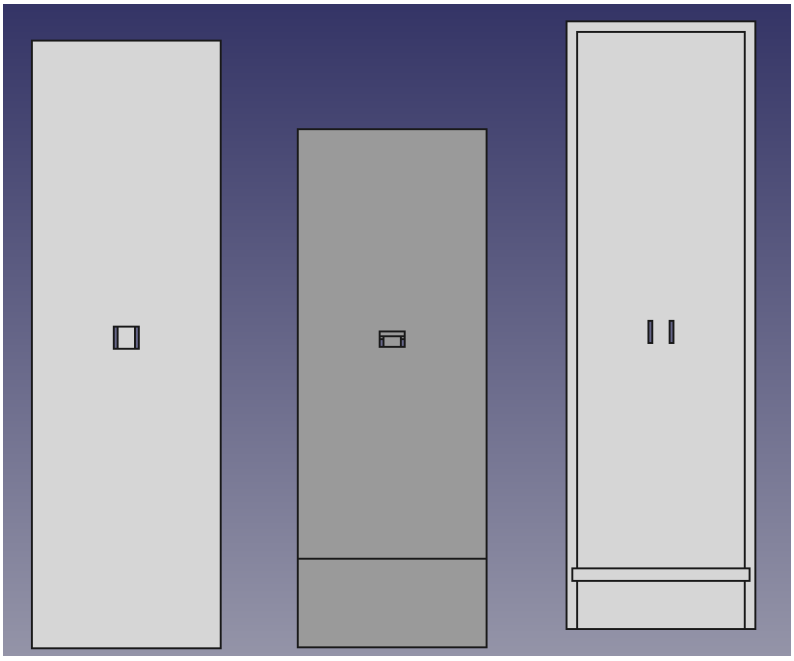

Figura 10 – Visão superior, em 45° e inferior respectivamente do *case* com a adição do baixo relevo e fenda para o botão de gatilho.

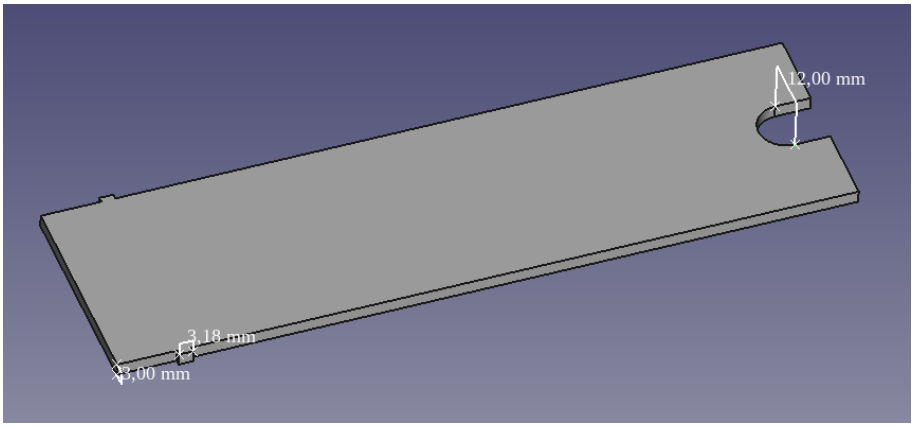

Figura 11 – Fase inicial do projeto da tampa inferior, antes da adição do suporte para a placa de circuito impresso.

usados nas laterais. Para isso, foram adicionados dois semicírculos às laterais da tampa. Eles possuem diâmetro interno de 1,5 mm e estão localizados de maneira a também serem englobados pelo *case* quando o dispositivo estiver montado. A peça resultante, ilustrada na Figura 12, exige que o *case* seja perfurado em suas laterias de maneira alinhada aos furos da tampa.

A eletrônica, primariamente em função do giroscópio, deve permanecer nivelada e devidamente fixada. O peso da placa de circuito impresso auxilia na fixação da mesma, uma vez que ele aplica pressão na direção da peça que a segura, fato o qual não seria verdade para as demais faces do invólucro do dispositivo. Utilizando isso, optou-se por fazer com que o encaixe da placa de circuito impresso fosse justo, não havendo peças

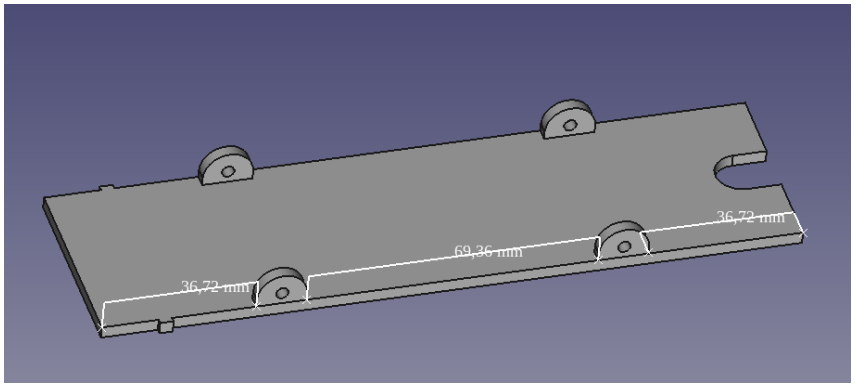

Figura 12 – Tampa inferior com adição de suportes laterais para aparafusamento.

mecânicas móveis, nem encaixes que aplicassem força extra. Assim, uma região de alto relevo com dimensões maiores que a da placa de circuito impresso foi criada. Uma segunda zona, agora de baixo relevo e do tamanho da placa, foi então escavada a partir do topo da zona de alto relevo, como podemos ver na Figura 13.

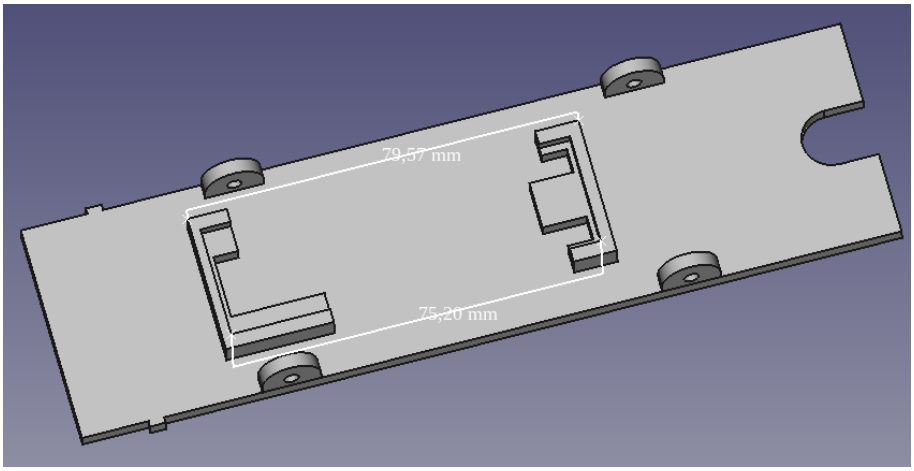

Figura 13 – Tampa inferior com suporte para a placa de circuito impresso.

As áreas de relevo intermediário tem também função de suporte. Elas contornam regiões onde existem pinos de componentes, fazendo com que os pontos de suporte inferior da placa não tensionem eles. Isso impede que a parte inferior da placa tenha contato direto com a tampa por meio dos seus pinos, deixando que todas as forças sejam aplicadas à placa de circuito impresso em si. O conjunto como um todo exerce a função de suporte e também de fixação da placa. Suas dimensões, justas ao tamanho da placa de circuito impresso, realizam essa função de fixação.

Também dentro do FreeCAD foi realizada a montagem da peça, verificando o correto encaixe. A peça mecânica final pode ser observada na Figura 14 em diferentes ângulos, demonstrando os encaixes das peças.

O método escolhido para a produção da peça foi a impressão 3D por filamento. A

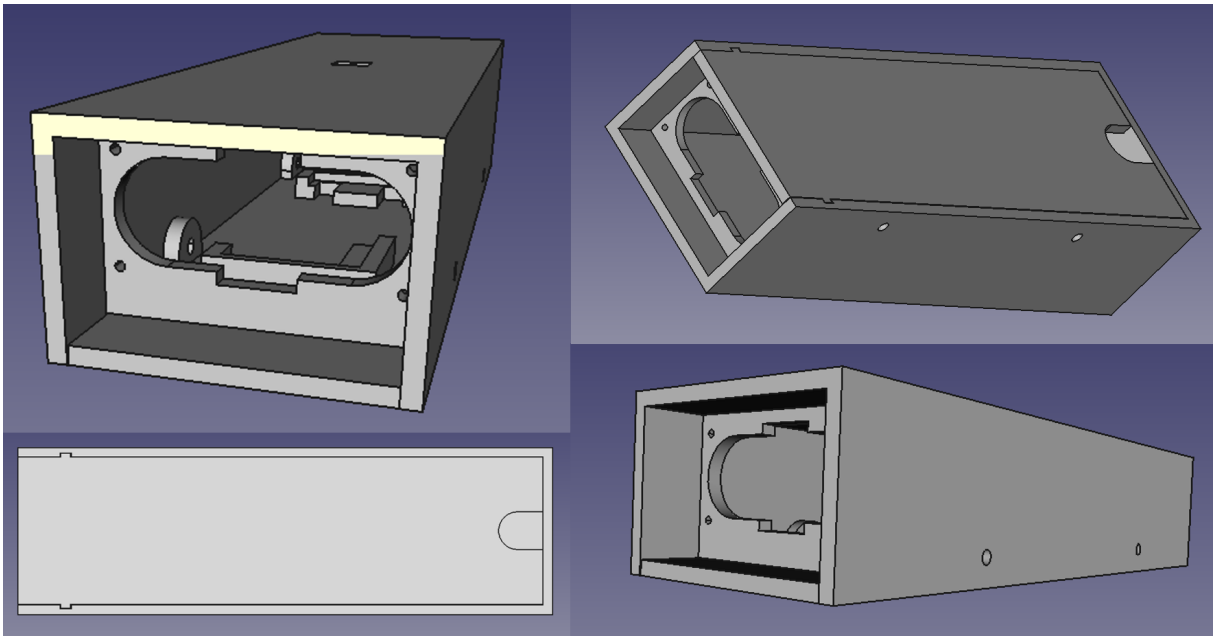

Figura 14 – Encaixe das peças que compõem o a peça mecânica do dispositivo.

geometria da peça permite tal tipo de técnica sem a necessidade de adição de suportes e o material escolhido, poli(ácido láctico) (PLA), é um plástico leve, porém resistente, conformando com o requisito **HI1** e **HI2**. A produção dessa peça foi feita junto do CTA, onde foi disponibilizado tanto a impressora (modelo Sethi3D S2), como o material para impressão.

Solução

A peça foi produzida de acordo com o desenvolvimento relatado sem empecilhos. O processo de soldar os fios do botão de gatilho foi exitoso, porém notou-se que a utilização do ferro de solda em proximidade ao plástico do invólucro durante essa operação deve ser repensada em iterações futuras, pois pequenos derretimentos e deformações são geradas na região interna em torno do botão de gatilho.

O passo seguinte foi iniciar o processo de testagem; primeiro, pelos testes de verificação simples. Uma vez que o invólucro foi produzido em PLA, temos **HI1** satisfeito. A pesagem de todas peças do invólucro resultou numa massa de 87 g, estando dentro do limite estipulado por **HI2**. Verificou-se que processo de montagem do *hardware* envolvia, primeiramente, a fixação do *slider* ao sensor de distância, seguido da fixação da placa de circuito impresso à tampa, a conexão dos fios do botão à placa e o fechamento e fixação da tampa. Essa última etapa se mostrou mais complicada, sendo recomendado, primeiro, realizar um encaixe inicial do *slider* na sua canaleta para, então, fechar a tampa. Os fios do botão possuem 10 cm de comprimento, o que os tornam suficientemente longos para que seja possível conectá-los à eletrônica, evitando tensioná-los demais e rompê-los ou sua

solda. Constatou-se, também, que não é recomendado aumentar o tamanho dos fios para valores muito além de 10 cm, pois eles começam a se tornar incômodos durante o processo de fechamento da tampa, além de ocuparem muito espaço interno e gerarem tensões em si mesmos ou nos outros componentes. Ainda assim, foi verificado durante a execução de todos os testes que a solda e fixação desses fios eram uma área de problema, acarretando com certa frequência no rompimento da sua solda ao botão. Como isso está também diretamente atrelado à remontagem constante da peça mecânica em decorrência dos testes, ação essa não esperada durante o uso comum do dispositivo, considerou-se tal constatação como um ponto de melhoria futura, mas não de inviabilização ou má-conformação do invólucro aos seus requisitos. A fixação por parafusos se demonstrou plenamente suficiente para fixar o conjunto de peças do dispositivo. Notou-se, também, a necessidade de em uma iteração futura de desenvolvimento adicionar regiões de baixo relevo que impeçam que a cabeça dos parafusos destaquem-se para fora das paredes laterais do dispositivo. Tal impasse não foi considerado como suficiente para não conformar com **HI5**, uma vez que, quando há alguma forma de rotação devido à presença das cabeças dos parafusos, ela se dá no eixo longitudinal do dispositivo, o que não afeta significativamente a medição de distância. Assim, por verificação simples, é possível aferir que **HI3** está sendo cumprida, bem como **HI5**, uma vez que a saída do cabo USB se localiza na região inferior da peça mecânica. Com a fixação da tampa inferior com uso dos parafusos, é possível averiguar que todos os componentes da eletrônica, incluindo sua placa de circuito impresso, estão devidamente fixados e englobados pela peça mecânica, cumprindo com **HI4** e **HI6**. Por fim, uma vez que a localização do botão se encontra na região externa do invólucro, é possível verificar que **HI7** está sendo cumprido,

Para a realização do **Teste de nivelamento**, apoiou-se o dispositivo em uma superfície plana e nivelada para a aquisição do primeiro conjunto de dados. O local de testes era uma quina interna de um cômodo, de forma que duas das quatro laterais do dispositivo ficassem em contato com alguma parede. Para o segundo, o dispositivo foi segurado pelo usuário ao invés de apoiado na superfície nivelada. Utilizou-se da geometria do invólucro para assegurar o nivelamento do dispositivo, segurando-o contra a quina. Para determinar se o apertar do botão deflete suficientemente a medida, a ponto de alterar significativamente o seu erro, um intervalo de confiança foi calculado para a diferença entre as médias dos valores de distância medidos nas duas conformações. Ele é definido por [Statology.org 2020]:

$$IC = (\bar{x}_1 - \bar{x}_2) \pm t\sqrt{\left[\left(\frac{s_p^2}{n_1}\right) + \left(\frac{s_p^2}{n_2}\right)\right]}, \tag{4.3}$$

onde $\bar{x}_1$ e $\bar{x}_2$ são as médias de cada grupo em cm, t é o t-valor crítico para um dado valor de confiança (não possui unidade de medida) e s_p^2 é a variância agrupada (do inglês, *pooled*

variance) em cm^2, sendo definida como:

$$s_p^2 = \frac{(n_1 - 1)\, s_1^2 + (n_2 - 1)\, s_2^2}{n_1 + n_2 - 2}, \tag{4.4}$$

onde, assim como na equação anterior, n_i representa a quantidade de amostras no grupo i e s_i^2 representa a variância amostral desse grupo em cm^2. Ao final, IC será um intervalo de confiança em unidades de cm.

Tratando o primeiro grupo como os 10 pontos com o dispositivo apoiado e o segundo como os 10 com ele segurado em mãos, calculou-se os valores de média e variância amostral como sendo:

$$\bar{x}_1 = 57{,}33\,\text{cm}, \tag{4.5}$$

$$\bar{s}_1^2 = 0{,}02\,\text{cm}^2, \tag{4.6}$$

$$\bar{x}_2 = 57{,}58\,\text{cm}, \tag{4.7}$$

$$\bar{s}_2^2 = 0{,}04\,\text{cm}^2. \tag{4.8}$$

Sabendo que $n_1 = n_2 = 10$ e utilizando um valor de confiança de 99% com uma distribuição bicaudal, foi obtido o intervalo $[-0{,}4674, -0{,}0286]$ cm. Esse resultado pode ser interpretado como, para um nível de 99% de confiança, pode-se dizer, a partir dos dados levantados, que a diferença entre as médias dos dois grupos se encontra dentro desse intervalo. Como o valor conhecido de erro médio absoluto do sensor (2,70 cm, obtido no **Teste de medida do sensor de distância**) é maior que os valores contidos no intervalo, não é possível afirmar que o pressionamento do botão gere deflexões significativas para além de um erro já esperado do sensor. Dessa forma, têm-se como cumpridos os requisitos **HI5** e **HI8** atrelados a esse teste e é dado como finalizado o desenvolvimento, não somente da peça mecânica, mas também do *hardware* como um todo.

5 Desenvolvimento do *software*

5.1 *Firmware*

O *firmware* é o código embarcado ao dispositivo para garantir a execução ordenada de suas funções a nível de componentes de *hardware*. Como o algoritmo de reconstrução do ambiente será executado pelo computador pessoal do usuário, o papel do *firmware* se enquadra em simplesmente coletar os dados e enviá-los por comunicação serial. Dessa forma, os requisitos **(S3), (S4)** e **(S9)** serão desdobrados nessa parte do sistema.

Requisitos

- **Requisito (S3):** Para cumprir com esse requisito de sistema, que estipula que o uso do dispositivo deve poder ser desempenhado por uma única pessoa, foi traçado o seguinte requisito de *software*:

  - **(SwF1)**: Todas as entradas de instruções e gatilhos do usuário deverão ser feitas a nível do *hardware* do dispositivo.

  Esse requisito assegura que não haverão partes do uso do dispositivo que dependam do usuário fornecer qualquer tipo de entrada além do pressionamento do botão.

- **Requisito (S4):** Estipula o seguinte requisito de sistema: o dispositivo deve ter um gatilho acionável pelo usuário para a tomada de um dado individual. Para isso, foram traçados os seguintes requisitos de *software*:

  - **(SwF2)**: O *firmware* deve utilizar o sinal elétrico do botão para demarcar o início do processo de tomada e envio dos dados;
  - **(SwF3)**: O *firmware* deve produzir um único conjunto individual de dados por pressionamento contínuo do botão de gatilho.

- **Requisito (S9):** Esse requisito de sistema estipula o envio de forma ordenada dos dados coletados ao computador do usuário no momento do acionamento do gatilho. Assim, foram traçados os seguintes requisitos de *software*:

  - **(SwF4)**: O *firmware* deve enviar em uma única mensagem o conjunto de dados de distância e ângulo;
  - **(SwF5)**: O *firmware* deve realizar o envio do dado imediatamente após a sua coleta.

Testes

Assim como nos testes de *hardware*, cada requisito pode ser testado mediante as mesmas três diferentes abordagens: verificação simples, análise de documentação e testes de medida ou funcionalidade. Algumas verificações simples utilizarão um diagrama de bloco, o qual se encontra em ilustrado na Figura 15. A seguir temos os testes elencados para essa parte.

Verificação simples

- SwF1: Deve-se verificar dentro do diagrama de bloco do *firmware* que a única fonte de entrada de informação por parte do usuário ocorre por meio do uso do *hardware*, eliminando suas interações diretamente com o *firmware*;
- SwF2: Deve-se verificar dentro do diagrama de bloco do *firmware* que o sinal gerado pelo pressionamento do botão é utilizado como critério de decisão do início do processo de tomada de um conjunto individual de dados e seu subsequente envio;
- SwF5: Deve-se verificar dentro do diagrama de bloco do *firmware* que não existem subprocessos intermediários entre o final da coleta de dados e o início de seu envio.

Análise de documentação

Não foram elencados testes de análise de documentação para essa área. Isso decorre do fato de que o *firmware* é inteiramente produzido dentro do projeto, não havendo documentação externa para ser analisada. Mesmo havendo o uso de bibliotecas externas para a implementação das funções dos sensores, qualquer inconformidade já fora resolvida durante os testes da etapa anterior.

Teste de medida/funcionalidade

Teste de coleta e envio de dados

Esse teste consiste da averiguação das capacidades básicas de comunicação serial e envio e tomada de dados controlados por um gatilho. Apesar de diretamente testar os requisitos **SwF3** e **SwF4**, o teste também indiretamente ajuda a confirmar a adequação aos demais requisitos do *firmware*. Ele consiste na tentativa seguida da tomada e envio de dados e da análise de sua resposta. Para isso, em uma IDE (*Integrated Development Environment*, em português Ambiente de Desenvolvimento Integrado) ou em um *script* simples que implemente comunicação serial, deve ser averiguada a capacidade do *firmware* de estabelecer essa comunicação. Em seguida, deve-se utilizar do botão de gatilho para realizar o envio de 12 dados, garantindo que a comunicação de cada um é realizada de forma única e ordenada. Deve-se aferir, também, que, independente do tempo de

pressionamento do botão, somente um conjunto de dados é enviado. Para isso, duas variáveis temporais são analisadas: a duração do pressionamento do botão e o intervalo entre pressionamentos. Para a primeira variável, optou-se por três regimes: longo (mais de cinco segundos pressionamento), normal (aproximadamente um segundo, imitando um pressionamento regular esperado do usuário) e rápido (pressionamento o mais rápido possível pelo usuário). Para a segunda, em todos os regimes da variável anterior, deve-se realizar a repetição três vezes. Entre a primeira medida e a segunda, deve-se permitir um intervalo de tempo longo, de cerca de 30 s. Do segundo pressionamento para o terceiro, deve-se deixar um período curto de tempo, de cerca de 1 s. Por fim, do terceiro pressionamento para o quarto, deve-se deixar um período de tempo o mais curto possível para o operador. Dessa forma, são testadas as conformações de pressionamento longo, esperado e curto em diferentes intervalos entre eles. Para o teste ser considerado como aprovado, deve-se observar a ocorrência de um único envio por pressionamento do botão e do envio ordenado de cada dado imediatamente após ele.

Execução

Para o desenvolvimento do *firmware*, foi utilizada a IDE Arduino [Arduino] em um sistema operacional Linux, distribuição Mint, versão 19.3 de 64 bits. Ela permite o controle de versões e *download* direto de bibliotecas de terceiros facilmente. Dentre elas, utilizou-se da biblioteca `Wire`, responsável por fornecer rotinas para a implementação do protocolo de comunicação I2C e a `MPU6050_light`, responsável por implementar a tradução do sinal elétrico cru do giroscópio em um valor de variação angular. O desenvolvimento dentro da IDE se dá através de um ou mais arquivos com terminação `.ino`. Esse desenvolvimento se dá em linguagem `C++`, utilizando sintaxe própria do Arduino construída com base nessa linguagem.

Dado que existem bibliotecas dedicadas para o giroscópio e que a função de coleta de dados do sensor de distância é suficientemente simples, optou-se por utilizar um único arquivo `.ino` para o *firmware*. Como é padronizado na sintaxe Arduino, uma função é executada no momento da inicialização do código (quando a placa de prototipagem é ligada ou reiniciada) e a função principal, que é executada continua e repetidamente após a inicialização. Dessa forma, rotinas de inicialização dos componentes, variáveis, protocolos de comunicação e funções que devem ser executadas uma única vez estão definidas dentro da rotina de *setup*, como podemos ver na Figura 15.

Com essa inicialização, passa-se ao laço que é executado repetidamente dentro do código (na Figura 15, a região em roxo claro denominada *Loop*). Ele se encarrega de verificar se houve o gatilho de tomada de dados, a tomada de dados em si e a atualização constante dos valores do giroscópio, uma vez que esse componente precisa ser atualizado a cada laço. Para garantir que flutuações na porta de entrada do botão de gatilho não

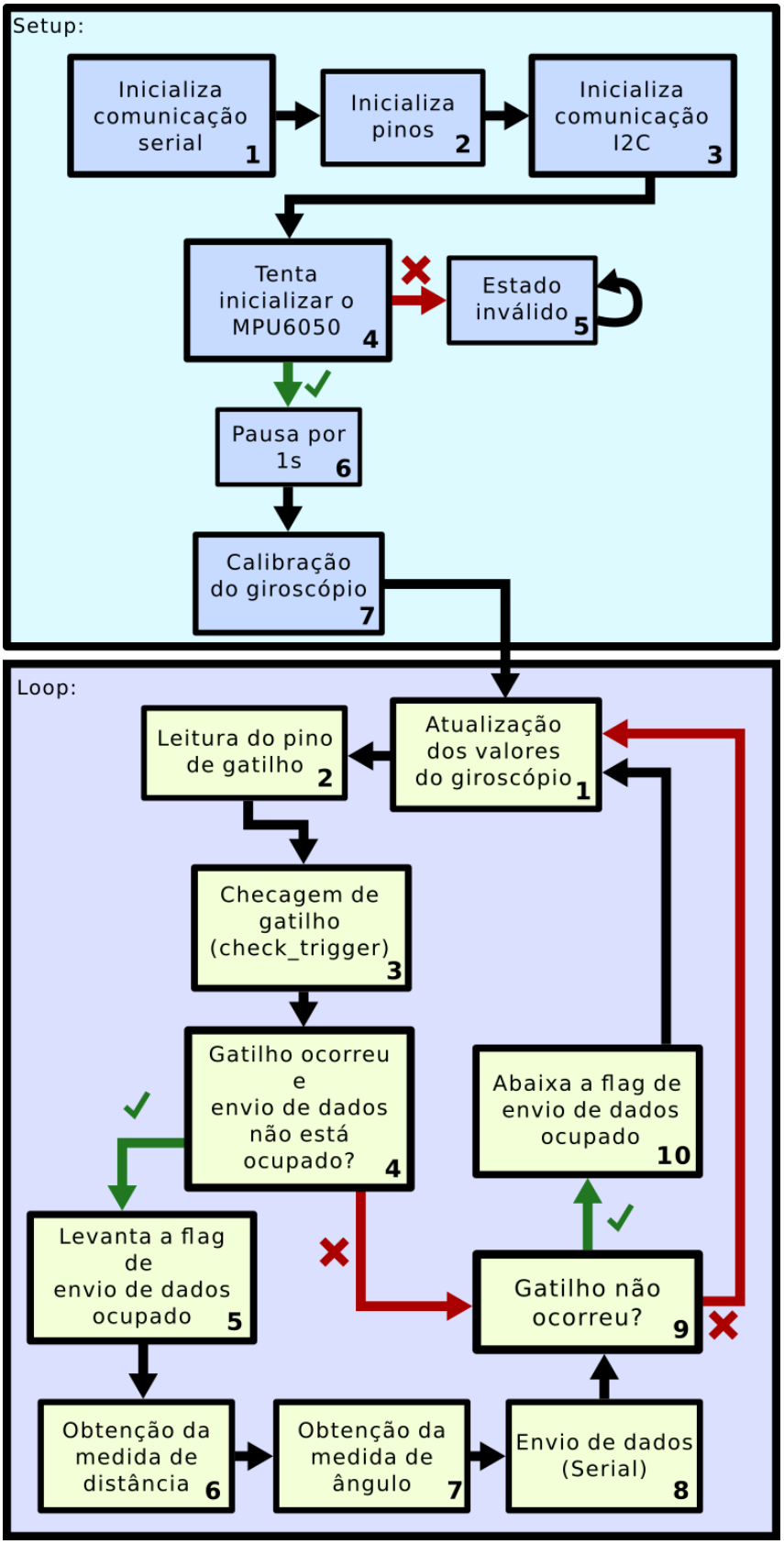

Figura 15 – Diagrama de bloco do *firmware* demonstrando o fluxo de informação do *setup* e laço de execução.

iniciassem indevidamente o processo de coleta de dados, foi desenvolvido um filtro digital. Ele se encontra no passo de número 3, sendo implementado pela função `check_trigger`. Sua implementação se baseia em armazenar os cinco últimos valores lidos no pino do botão e retornar que ocorreu o gatilho somente quando todos esses valores forem verdadeiros. No entanto, essa checagem não é o único critério de decisão para o início da tomada de dados. O *firmware* possui uma variável interna denominada `busy` (do inglês, ocupado) que serve de *flag* indicativa do estado atual do envio de dados. Essa *flag* inicia com valor *Verdadeira* e se torna *Falsa* após o primeiro ciclo, como será visto à frente. Se o seu valor é falso e a função `check_trigger` indicou a ocorrência de um gatilho, inicia-se a tomada de dados. Isso atribui um valor *Verdadeiro* à *flag*, indicando que um dado foi gerado e logo o dispositivo está ocupado. Desse ponto em diante, até que um valor *Falso* seja atribuído a essa *flag*, o dispositivo não realizará novamente a coleta e envio de dados. Esse comportamento garante que para um único pressionar do botão de gatilho, um único conjunto de dados é gerado. Para que esse valor *Falso* seja novamente atribuído a ela, é necessário que o valor apresentado no passo 9 seja *Verdadeiro*. Ou seja, é preciso que não ocorra um gatilho. Assim, fica garantido que mesmo que o botão seja indefinidamente pressionado, têm-se um único conjunto de dados.

Como pode ser notado, o *firmware* realiza somente a coleta e o envio continuado dos dados, não possuindo de um controle de fluxo que indique um término no processo de medição. Tal funcionalidade ficou relegada ao algoritmo de reconstrução, visando, assim, reduzir a complexidade do *firmware*, uma vez que esse controle é mais facilmente realizado diretamente pelo computador do usuário.

Solução

O conteúdo do arquivo `.ino`, que define o *firmware*, se encontra na página do GitHub do projeto [a]. Uma vez feito seu carregamento para o Arduino Nano, inicia-se a condução de testes e a verificação do diagrama para o caso de requisitos verificáveis por meio de inspeção. O requisito **SwF1** exige que não haja, dentro do diagrama de blocos, nenhuma entrada de dados extra por parte do usuário além do gatilho de medição. O passo número 2 do *Loop* é o único momento em que o usuário insere alguma entrada de dados no *firmware*, portanto cumpre-se com esse requisito. De mesma forma, o requisito **SwF2** pode ser verificado dentro do diagrama de bloco nos blocos 2, 3 e 4 do *Loop*. O bloco número 2 é a única entrada de dados, sendo comandada pelo botão e o critério de decisão em 4 é criado em 3 utilizando somente esse sinal. Dessa forma, torna-se evidente que o requisito é cumprido, uma vez que somente o sinal do botão é usado como critério de decisão. Por fim, temos **SwF5** como o último requisito verificável por inspeção. Como podemos ver no diagrama da Figura 15, após a positiva do bloco 4, com exceção da atribuição do

[a] https://github.com/Vfgandara/pfcef_software

valor *Verdadeiro* à *flag* de coleta de dados, passo esse necessário para garantir **SwF3**, não existem tarefas não relacionadas à coleta ou envio dos dados até o passo `9`, onde já se encontra finalizado o envio de dados por comunicação serial. Dessa forma, o requisito é tido como cumprido.

O **Teste de coleta e envio de dados** foi conduzido já utilizando um *script* em linguagem `Python`, versão 3.10, desempenhando comunicação serial. Dessa forma, foi possível aferir a capacidade do *firmware* de realizar a comunicação serial e, concomitantemente, também testar as capacidades de recebimento dos dados por parte do futuro algoritmo de reconstrução de forma incipiente. Foi averiguado que os 12 dados enviados foram recebidos sequencialmente e sem o envio de mais dados em todos os regimes, cumprindo, assim, com **SwF3** e **SwF4**.

Dessa forma, cumpre-se com todos os requisitos do *firmware* e o desenvolvimento do Algoritmo de reconstrução pode ser executado. Neste ponto de desenvolvimento, o dispositivo se encontra já em capacidade de atuação e cabe ao algoritmo utilizar os valores que ele o envia para realizar seu papel de reconstrução.

5.2 Algoritmo de reconstrução

O algoritmo de reconstrução é um *script* de código executado no computador do usuário, o qual é responsável por interpretar os dados advindos do dispositivo para recriar o desenho do cômodo medido. Para cumprir com o seu papel dentro do sistema, os requisitos **(S3)**, **(S10)** e **(S11)** serão desdobrados.

Requisitos

- **Requisito (S3):** Para cumprir com esse requisito de sistema, que estipula que o uso do dispositivo deve poder ser desempenhado por uma única pessoa, foi traçado o seguinte requisito de *software*:
  - **(SwA1)**: O algoritmo de reconstrução deve ser executado sem entradas do usuário durante a tomada de medidas.
- **Requisito (S10):** Esse requisito de sistema estipula que o processo de reconstrução deve utilizar somente os valores advindos do dispositivo. Dessa forma, foram traçados os seguintes requisitos de *software*:
  - **(SwA2)**: O *script* deve desempenhar o recebimento dos dados do dispositivo por comunicação serial;
  - **(SwA3)**: Após inicializado, o *script* deve ser executado de início a fim utilizando somente mensagens advindas da comunicação serial do dispositivo;

  - **(SwA4)**: O algoritmo de reconstrução deverá ser invariante frente o sentido de rotação escolhido pelo usuário.

- **Requisito (S11):** Para cumprir com esse requisito de sistema, que estipula que o resultado do processo de reconstrução deve ser o desenho da geometria do cômodo medido com anotações de distância em cada uma das arestas, foram traçados os seguintes requisitos de *software*:

  - **(SwA5)**: Ao final da execução do *script*, deve ser mostrado ao usuário uma representação gráfica do ambiente medido contendo sua geometria;
  - **(SwA6)**: A representação gráfica final do ambiente deve conter anotações de tamanho de cada parede individual.

Testes

Verificação simples

- SwA1: Deve ser averiguado no diagrama representativo do algoritmo de reconstrução que nenhum dado de entrada advém do usuário após a inicialização do algoritmo;
- SwA3: No diagrama representativo do algoritmo de reconstrução, deve ser averiguado que nenhum dado de entrada, além daqueles advindos da comunicação serial, é utilizado durante o processo de reconstrução.

Análise de documentação

Não foram elencados testes de análise de documentação para essa área. Isso decorre do fato de que o *firmware* é inteiramente produzido dentro do projeto, não havendo documentação externa para ser analisada. Mesmo havendo uso de bibliotecas de terceiros, elas são utilizadas somente ao desempenhar sub-funções do algoritmo, não implementando nenhum de suas partes por inteiro e tendo portanto caráter utilitário.

Teste de medida/funcionalidade

Teste de chegada de dados

Esse teste consiste em, com o dispositivo em estado funcional, averiguar-se a correta implementação da comunicação serial para envio dos dados entre o *hardware* e o *script* de reconstrução. Esse deve ser executado em uma unidade externa que não o *hardware* e deve mostrar ao usuário a mensagem obtida de forma individual, cada vez que um dado é enviado. Esse teste é utilizado para comprovar a adequação do requisito **SwA2**.

Teste do algoritmo de reconstrução

Esse teste consiste em garantir a capacidade do algoritmo de reconstrução de utilizar os dados sequenciais de distância e ângulo a cada aresta do cômodo para reconstruir sua geometria. Ele não é um teste integrado ao *hardware*, uma vez que essa integração é testada a nível de sistema na seção 6.2. Suas entradas são dados anotados manualmente que representam casos de uso específicos do dispositivo, isto é, cômodos de diferentes geometrias com graus variantes de complexidade. Esses dados falsos são criados já tendo em mente a estrutura da mensagem de dados que chega ao algoritmo por meio da comunicação serial e deve representar fidedignamente o comportamento esperado dos sensores do dispositivo. À parte do processo de *parsing*[b] e consumo de dados, o resto do algoritmo deverá se manter inalterado, garantindo que os resultados sejam representativos do funcionamento do dispositivo final junto dele.

As geometrias dos cômodos testados são de grande importância para o levantamento de dados úteis aos resultados. As principais características que devem estar presentes são a mudança de convexidade do cômodo (presença de quinas internas) em diversas quantidades e sequências. Por parte do comportamento do usuário, alguns desses que afetam diretamente os valores dos sensores devem ser também considerados. Eles são: rotações não usuais do equipamento (rotações de $\pm 90°$ sendo desempenhadas como rotações de $270°$ no outro sentido) e presença de rotações completas ($\pm N \cdot 360°$) em pontos avulsos. Foram traçados cinco geometrias diferentes com base nas seguintes características de interesse aos testes: (i) sem presença de quinas internas, (ii) presença de uma quina interna, (iii) presença de duas ou mais quinas internas em sequência, e (iv) presença de duas ou mais quinas internas sem estarem em sequência. Para garantir que o algoritmo possua robustez frente a flutuações nos valores do giroscópio, que são aqueles usados como critério de decisão, adiciona-se ruído às medidas de mesma ordem de magnitude que aqueles mensurados e obtidos no **Teste de medida do giroscópio**. Como relatado em seção 2.1, todos os ângulos internos presentes nos cômodos são de $90°$ ou $270°$ (medida do ângulo interno). Além disso, todos os testes devem ser conduzidos em ambos sentidos de medidas (anti-horário e horário), testando, assim, **SwA4**. Nos casos onde existem quinas internas, deve ser testado conformações onde o primeiro ponto de medida é justamente a quina interna. Esse teste é utilizado para comprovar a adequação dos requisitos **SwA5** e **SwA6**.

Execução

O algoritmo de reconstrução foi desenvolvido em linguagem `Python`, versão 3.10, tendo como abordagem escolhida a orientação a objeto (*Object Oriented Programming*). Para isso, separou-se as funções do algoritmo em partes modulares para serem transformadas em objetos. A orquestração desses objetos é realizada por meio de um *script* principal,

[b] Do inglês, *parsing* consiste no processo de separar um conjunto de dados em seus componentes menores.

doravante chamado de *main*. Em um primeiro grau de abstração, dividiu-se as funções do algoritmo em comunicação serial e reconstrução da geometria. Essas funções foram delegadas às classes `SerialDataBroker` e `Solver`.

A comunicação serial é uma tarefa simples, não necessitando outros objetos internos para auxiliá-la no processo. O processo é baseado na biblioteca `pyserial` e, para seu funcionamento básico, é necessário prover um identificador da porta USB e uma *baud-rate*[c]. Esse primeiro advém do sistema operacional do usuário, sendo geralmente `/dev/ttyUSB0` para sistemas GNU-Linux e `COM0` para Windows. A *baud-rate* é definida de acordo com o dispositivo, sendo escolhido 9600 como o valor. O processo de preparação do `SerialDataBroker` passa por esperar uma mensagem pré-determinada que identifica que findou-se o último processo de inicialização antes do dispositivo estar pronto para enviar dados. Após isso, o `Broker` permanece constantemente esperando pela chegada de um dado. Para isso, ele utiliza do método `.readline()` do objeto que realiza a interface de comunicação serial dentro da biblioteca `pyserial`. Assim, o que delimita o final de uma mensagem de dados é a presença de uma nova linha na mensagem (usualmente, o caractere `\n`). Dentro do *firmware*, o envio de um dado é sempre realizado pela função `Serial.println()`, que já adiciona a quebra de linha ao final da mensagem.

Já a reconstrução da geometria, por ser um processo mais complexo, utiliza outros objetos mais simples para implementar ou auxiliar em partes de sua rotina. O primeiro desses objetos é o `AngleWheel`, que é responsável por dinamicamente transformar e armazenar os dados de variação angular do giroscópio em valores úteis à reconstrução. Por fim, esse objeto utiliza uma estrutura simples de lista recorrente também implementada através de uma classe (`LoopedList`) para manter registro da última direção onde foi traçada uma aresta da geometria. Esse registro é utilizado como forma de, dinamicamente, determinar a próxima direção com base no valor de variação angular do dispositivo. Seu comportamento recorrente vem do fato de que um incremento de seu índice, além do número de elementos contidos, retorna o objeto contido no início da lista, tornando-a recorrente e circular.

A ideia por trás da implementação escolhida é utilizar o valor de variação angular de maneira a conseguir aferir se a parede medida está a $+90°$ ou $-90°$ em relação à parede anteriormente medida. Como o sinal do giroscópio é continuamente integrado, ou seja, o valor atual é a soma de todas as variações anteriores, é necessário processar o sinal para utilizá-lo na comparação com o valor anterior. Para mitigar o uso indevido do dispositivo, o tratamento do valor de ângulo também checa se o usuário realmente realizou uma rotação de $\pm 90°$. Caso a rotação tenha sido menor que $\pm 45°$ ou estiver entre valores de $180° \pm 45°$, o algoritmo não utiliza o conjunto de dados enviado. Para facilitar todas essas checagens

[c] Do inglês, *baud-rate* é uma medida de velocidade de transferência de dados por um canal de comunicação. Um valor exemplo de 9600 de *baud-rate* estipula uma taxa máxima de transferência de 9600 bits por segundo.

por valores inválidos, existe um processo de tratamento desses valores para normalizá-los para valores dentro do intervalo $[-360°, 360°]$. Ele é bastante simples e utiliza do operador módulo, também conhecido como divisão sem resto. Sua operação pode ser entendida como:

$$\alpha_{norm} = \begin{cases} \alpha - 360 \cdot (\alpha // 360), & \text{se } \alpha \geq 0. \\ \alpha + 360 \cdot (\alpha // -360), & \text{se } \alpha < 0. \end{cases} \tag{5.1}$$

onde α é o valor de diferença entre os valores de ângulo atual e o anterior (em graus) e $//$ representa o operador módulo. Para casos onde $\alpha \in (-360°, 360°)$, o operador módulo retornará 0 e não haverá normalização. Para os intervalos $(-\infty, -360°)$ e $(360°, \infty)$ o operador módulo retornará a quantidade de rotações extras realizadas e remover essa quantidade de α, assim, mapeando-o de volta para o seu ângulo equivalente dentro de $[-360°, 360°]$.

A seguir, passa-se ao processo de escolha entre $-90°$, $+90°$ ou inválido como passo prévio à escolha de uma direção final para traçar a próxima parede do cômodo. O cálculo toma como valor de entrada a diferença entre o ângulo atual medido e o anterior já normalizado, ou seja, com essa diferença passando pelo processo de normalização descrito em Equação 5.1. Em primeira vista, o critério de decisão pode ser compreendido como uma checagem simples do valor de diferença, sendo colocado um intervalo do tipo

$$\alpha_{\text{final}} = \begin{cases} -90°, & \text{se } \alpha \in -90° \pm 45° \\ +90°, & \text{se } \alpha \in +90° \pm 45° \\ -1, & \text{caso contrário.} \end{cases} \tag{5.2}$$

onde -1 é o indicador de valor inválido e o valor de $45°$ é escolhido para manter a simetria entre as regiões válidas e inválidas do intervalo de uma rotação, podendo ser alterado e entendido como um parâmetro ajustável da rigorosidade do critério de escolha.

Apesar de válida, essa escolha não contabiliza rotações não usuais do usuário, isto é, quando uma rotação válida é realizada como seu ângulo complementar. Esse é o caso de $90°$ sendo realizado como $-270°$ e $-90°$ como $270°$. Dessa forma, é necessário expandir a Equação 5.2 de forma que ela se torne:

$$\alpha_{\text{final}} = \begin{cases} -90°, & \text{se } \alpha \in -90° \pm 45° \text{ ou } \alpha \in 270° \pm 45° \\ +90°, & \text{se } \alpha \in +90° \pm 45° \text{ ou } \alpha \in -270° \pm 45° \\ -1, & \text{caso contrário.} \end{cases} \tag{5.3}$$

O seu comportamento pode ser entendido de maneira visual através do gráfico dessa função, como podemos ver na Figura 16.

Esse processo como um todo é desempenhado pela classe `AngleWheel` e serve de base para o cálculo das coordenadas das arestas do cômodo. Uma vez resolvido se a

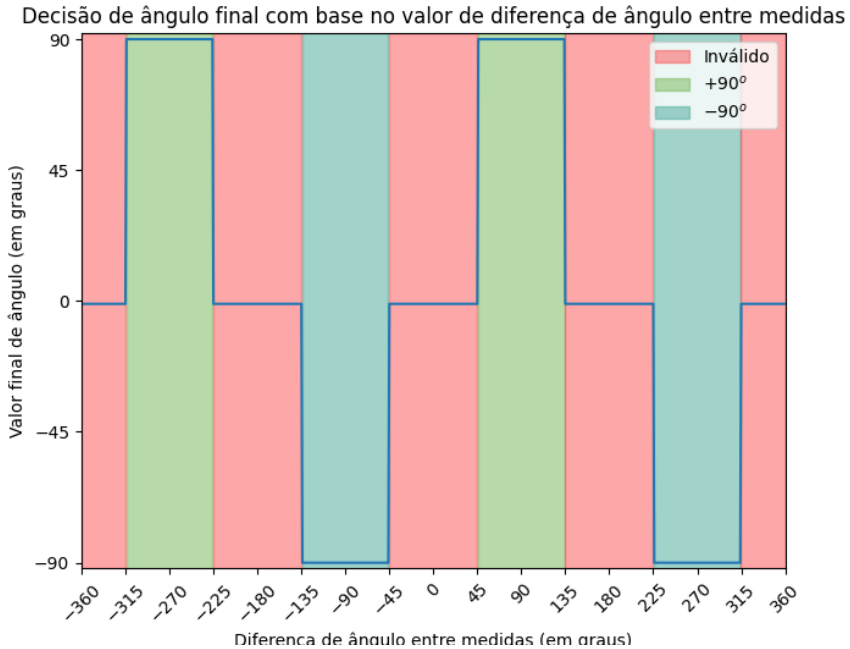

Figura 16 – Plot da função descrita na Equação 5.3 para o intervalo $\alpha \in [-360°, 360°]$.

diferença entre o ângulo de entrada e o último registrado indica $+90°$, $-90°$ ou um valor inválido, é necessário utilizar dessa informação para traçar a geometria do cômodo. Para isso, notou-se que a escolha da direção na qual deve-se traçar o próximo ponto depende da direção na qual foi traçada a última parede do cômodo e do valor de ângulo atual. Dessa maneira, é necessário que o *Solver* armazene a direção da última parede traçada, sendo as direções possíveis `x+`, `y+`, `x-` e `y-`. A classe `AngleWheel` é a responsável por esse armazenamento. Essas quatro possibilidades advém da restrição do problema a ângulos de $90°$ ou $270°$. Removendo essa restrição, faz-se necessário não fixar direções possíveis, mas sim utilizar da leitura do ângulo para determinar a direção. Durante o desenvolvimento, foi notada uma maneira simples de determinar a próxima direção a ser desenhada a parede. Ao serem dispostos na ordem `[y+,x-,y-,x+]`, sempre que a diferença angular entre a última parede desenhada e atual for $+90°$, deve-se pegar o próximo valor da lista como a direção no plano cartesiano a ser traçada a parede atual. No caso de uma diferença de $-90°$, o anterior. Para isso, considera-se a lista como recorrente, isto é, um incremento do seu último ponto (`x+`) leva para o seu primeiro ponto (`y+`). Essa propriedade advém justamente da disposição das coordenadas no eixo cartesiano, podendo ser compreendida facilmente ao observar o comportamento dos vetores unitários dessa base quando aplicadas rotações de $+90°$ e $-90°$. A nível de código, ao implementarmos essa propriedade de *loop* da lista através da classe `LoopedList`, basta, a cada entrada de dado, decidir se incrementamos ou decrementamos a lista para obter a direção e armazenar o resultado para o próximo ponto. As direções possíveis já determinam também qual operação deve ser aplicada às coordenadas anteriores para obter o novo vértice da geometria. A direção é determinada

por `x` ou `y` e se o valor de distância medido deve ser adicionado ou subtraído nessa direção é determinado pelo sinal + ou −. O algoritmo é finalizado pelo usuário por meio da sua terminação através de um `KeyboardInterrupt` (comumente, o pressionamento das teclas `Ctrl + C` no terminal). O conjunto de coordenadas gerado é, então, utilizado dentro de um *plot* utilizando da biblioteca `matplotlib`, sendo esse o resultado final.

Solução

Para tornar clara a estrutura do *software*, utilizou-se de uma forma adaptada da documentação C4 (Contexto, Contêineres, Componentes, Código) [Brown 2019]. Ela utiliza esses quatro níveis de abstração diferentes, sendo comumente o último (Código) omitido, assim como foi feito neste trabalho, em decorrência de seu tamanho e grau de complexidade. A nível de contexto dentro da documentação C4, Figura 17, fica demonstrado os agentes do processo, utilizando do esquema de cores cinza-azul para representar, respectivamente, partes não-integrantes e integrantes do *software* apresentado. Utiliza-se de setas para representar a relação entre as partes em cada grau de abstração e demarcações pontilhadas para determinar partes do sistema de *software*. Ao descer um nível de abstração, as caixas azuis do nível de abstração anterior são discretizadas com maiores detalhes. Logo, a nível de contêineres (Figura 18), demonstra-se o funcionamento das partes menores integrantes dos agentes dentro do contexto, e assim por diante, quando se adentra o nível de componentes (Figura 19). Uma vez que essa metodologia de documentação é pensada para arquiteturas de *software*, alterações foram feitas em sua concepção para melhor refletir o intuito de demonstração do algoritmo em si. Dessa forma, mesmo que alguns componentes e funcionalidades a nível de *script* sejam englobados por uma única classe, ainda assim, representou-se tais funcionalidades individualmente na Figura 19. Esse é o caso para as funcionalidades **Normalização do ângulo** e **Obter ângulo final**, ambos desempenhados pela classe `AngleWheel`, que reúne todas informações relativas ao processamento de ângulos no *script*. Justamente para manter um nível de abstração que permitisse representar de maneira clara o processo desempenhado pelo algoritmo, que se optou por tal abordagem. Com isso, a comunicação dos processos necessários para reproduzir a solução desempenhada pelo algoritmo não fica atrelada aos padrões de design do código.

O *script* contendo o algoritmo de reconstrução, assim como *scripts* de teste, podem ser encontrados na página do GitHub do projeto[d]. O **Teste do algoritmo de reconstrução** pode ser executado diretamente pelo arquivo `testing.py`, que executará automaticamente o algoritmo de reconstrução digital em todos os arquivos de teste dentro da pasta `/comodos_teste`, gerando um conjunto de *plots* e imagens de saída representando a planta baixa daquele arquivo de teste. Ao total, temos 32 arquivos, representando as

[d] https://github.com/Vfgandara/pfcef_software

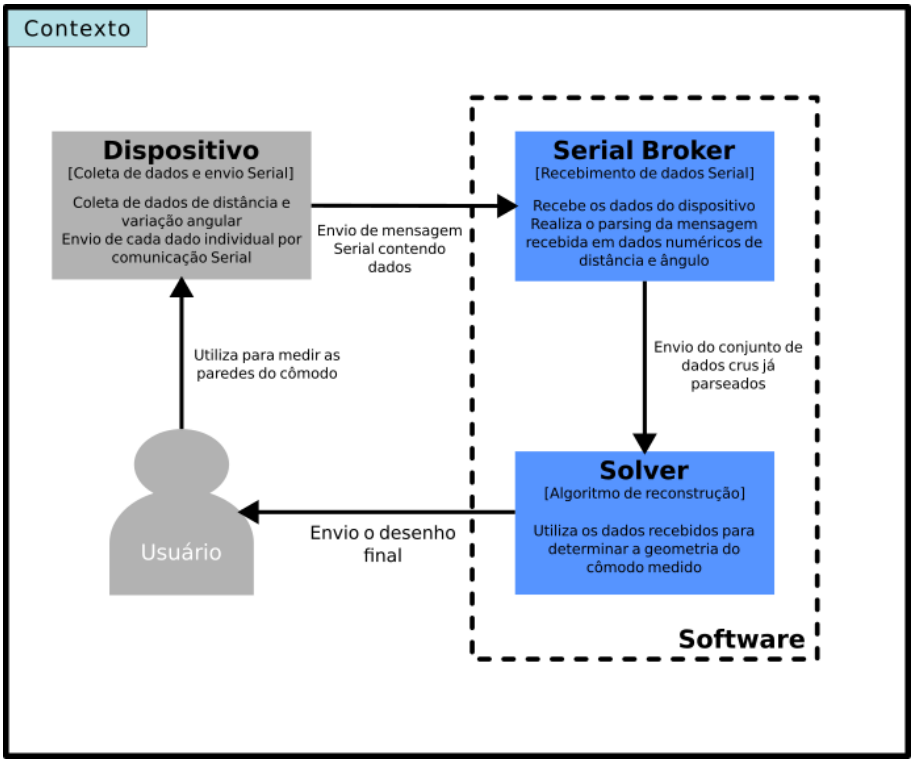

Figura 17 – Esquemático do funcionamento do *software* a nível de contexto, mostrando agentes, dispositivos e algoritmos. Esquema de ilustração adaptado da metodologia C4 de documentação [Brown 2019].

diferentes conformações de geometria e também comportamentos possíveis do usuário dentro delas, como discorridos na seção 5.2. Todos os arquivos testados geraram a planta baixa correta, mesmo com a adição de erro às medidas do giroscópio e as diversas formas de uso não esperado por parte do usuário que foram embarcadas nos dados. Isso demonstra a robustez do método de reconstrução e sua conformação com os requisitos **SwA4-6**.

Os requisitos **SwA1** e **SwA3** podem ser verificados por meio da documentação produzida. No diagrama da Figura 19, assim como aqueles de maior nível de abstração, podemos ver que o usuário interage com o algoritmo de reconstrução somente através do dispositivo durante a tomada de medidas. O único outro momento em que há alguma interação direta do usuário com o *script* é quando do início e fim de sua execução, os quais devem ser realizados pelo usuário. Como **SwA1** especifica que não deve haver entradas por parte do usuário durante a tomada de medidas, esse requisito é tido como cumprido. Novamente, através da Figura 19 e da explicação do funcionamento do algoritmo no seção 5.2, é possível notar confirmar que o requisito **SwA3** está cumprido, uma vez que não é utilizado nenhum tipo de dado externo àqueles advindos da comunicação serial. Exemplos de três geometrias diferentes de cômodos podem ser vistas nas figuras Figura 20, Figura 21, Figura 22 e Figura 23, juntos do resultado final da sua reconstrução pelo algoritmo. Para fins de melhor ilustrar os resultados, adicionou-se ruído somente aos valores de ângulo, com valores de média e desvio padrão retirados da Tabela 3, uma vez

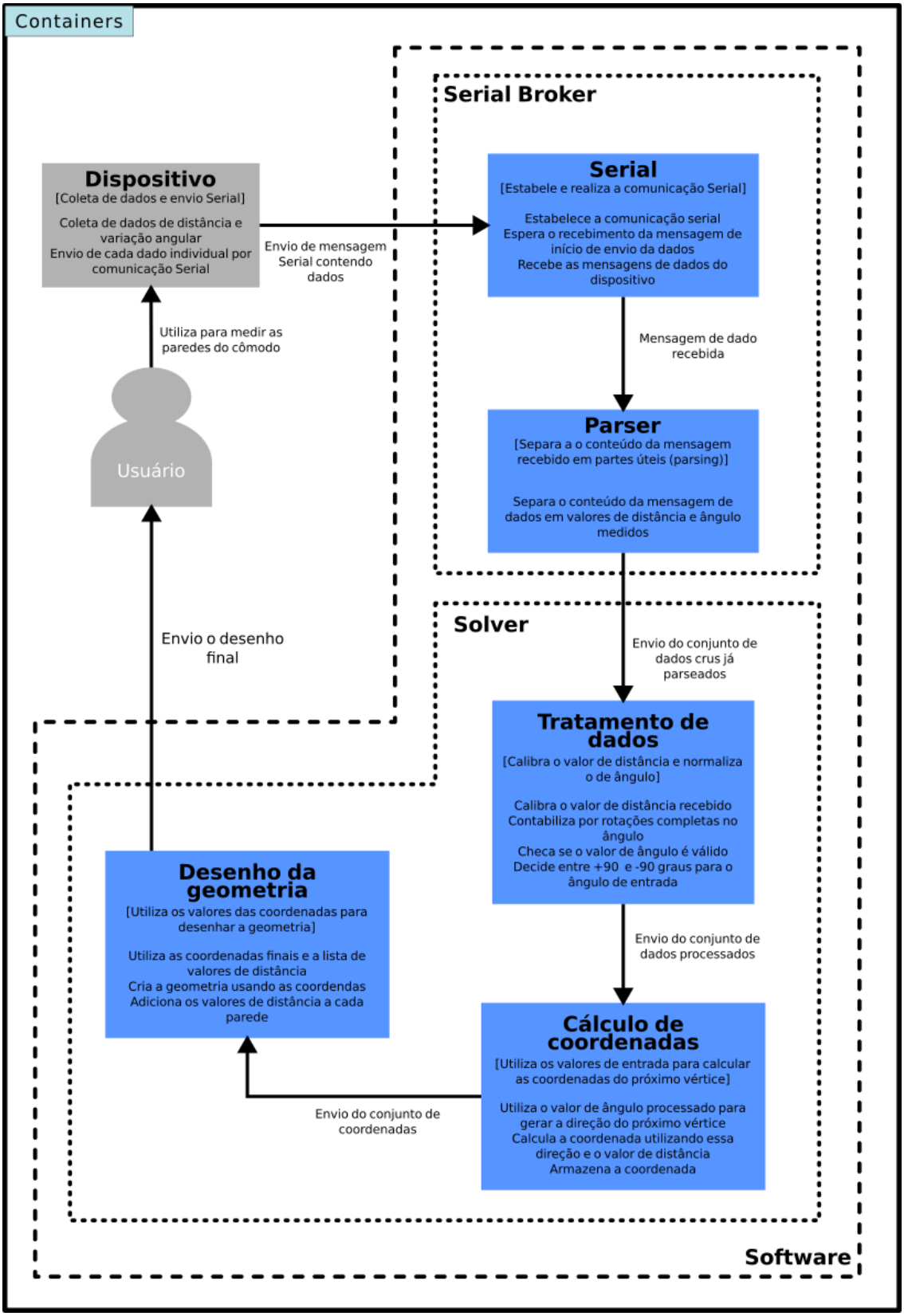

Figura 18 – Esquemático do funcionamento do *software* a nível de contêineres. É um nível de abstração abaixo da Figura 17, ilustrando o funcionamento interno das partes integrantes do software. Esquema de ilustração adaptado da metodologia C4 de documentação [Brown 2019].

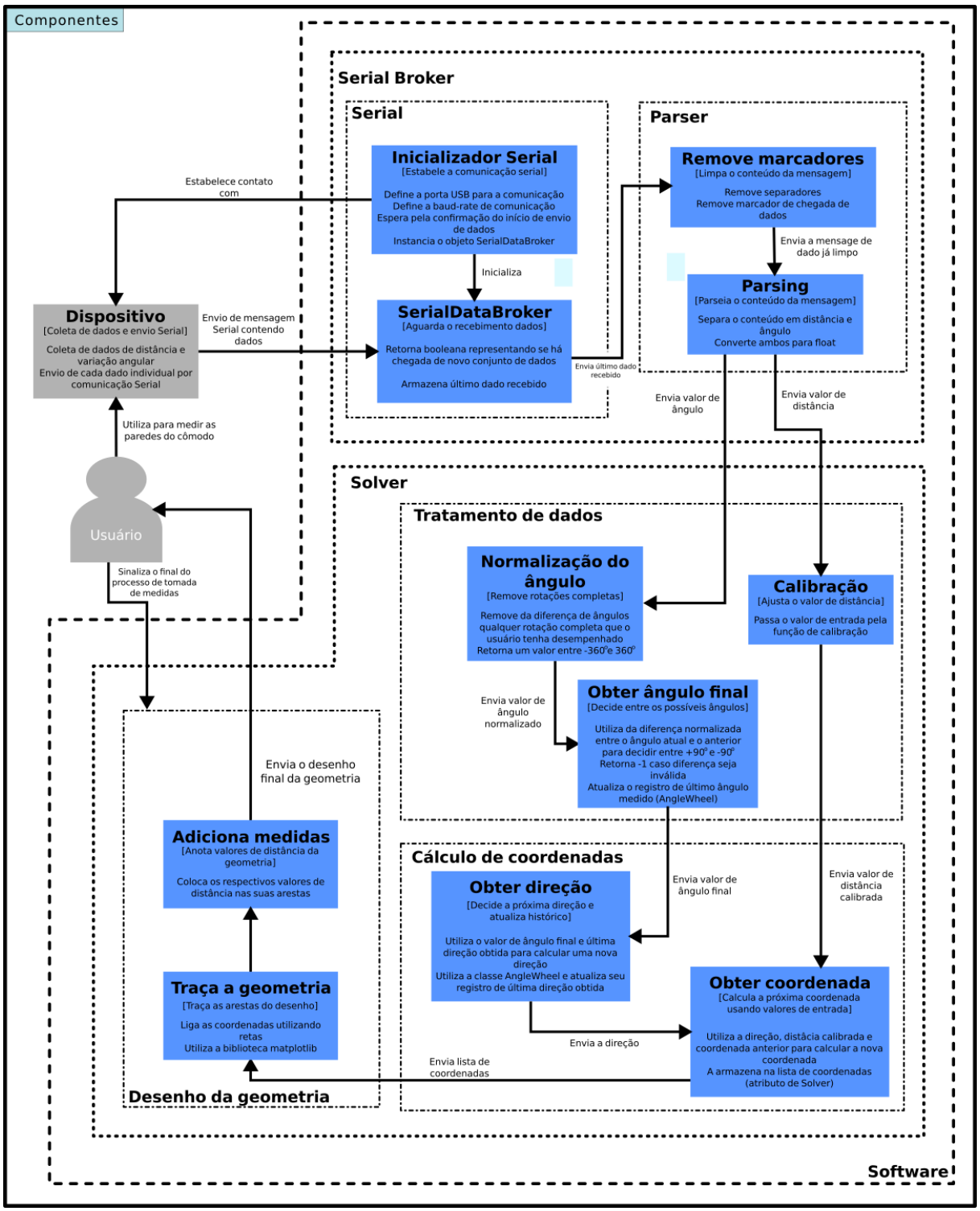

Figura 19 – Esquemático do funcionamento do *software* a nível de componentes. É um nível de abstração abaixo da Figura 18, ilustrando o funcionamento de cada componente, mostrando suas rotinas internas. Esquema de ilustração adaptado da metodologia C4 de documentação [Brown 2019].

que a adição de ruído ao sensor de distância leva a pequenas distorções na geometria. Os dados utilizados, ponto a ponto, estão dispostos nas colunas d e α das figuras, representando os valores de distância e ângulo, respectivamente. O dispositivo se encontra ilustrado como o retângulo azul com ponta vermelha e a ordem de medidas segue a lógica de que a próxima medida é sempre aquela para a qual o dispositivo aponta. Na Figura 23, a estrela amarela representa o ponto no qual uma rotação de $-90°$ é realizada como uma rotação de $+270°$.

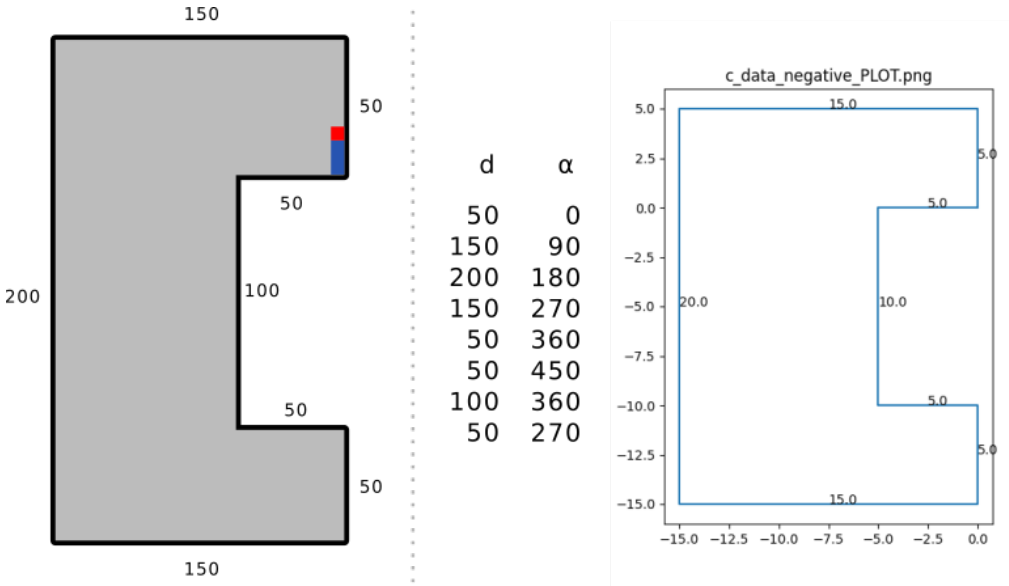

Figura 20 – Ilustração de uma geometria em formato de `C`, seus dados de entrada e a planta baixa construída pelo algoritmo.

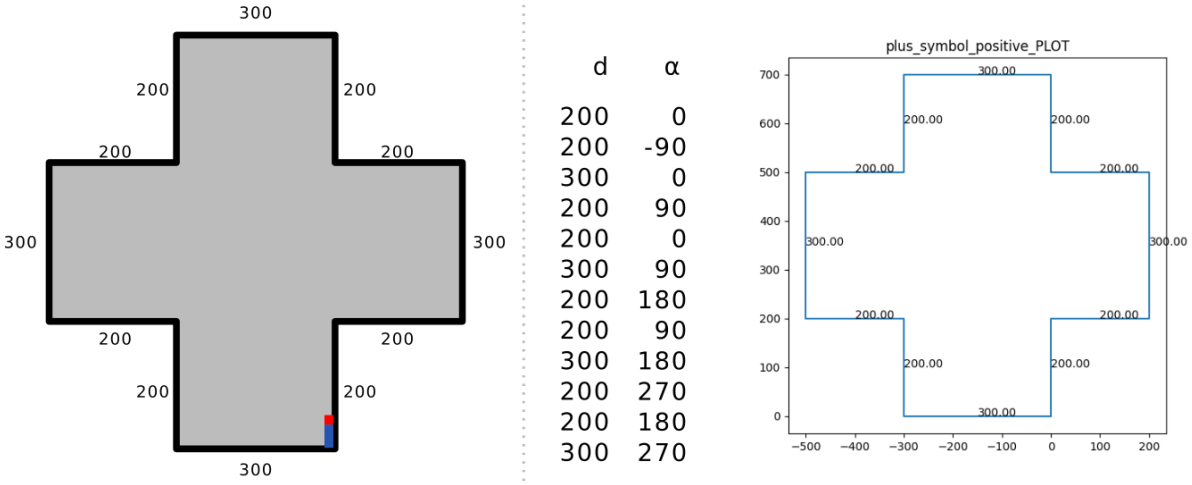

Figura 21 – Ilustração de uma geometria em formato de `+`, seus dados de entrada e a planta baixa construída pelo algoritmo.

Por fim, o requisito **SwA2**, atestável através do **Teste de chegada de dados** também pode ser confirmado. O *script* `main.py` na na página do GitHub do projeto[e]

[e] https://github.com/Vfgandara/pfcef_software

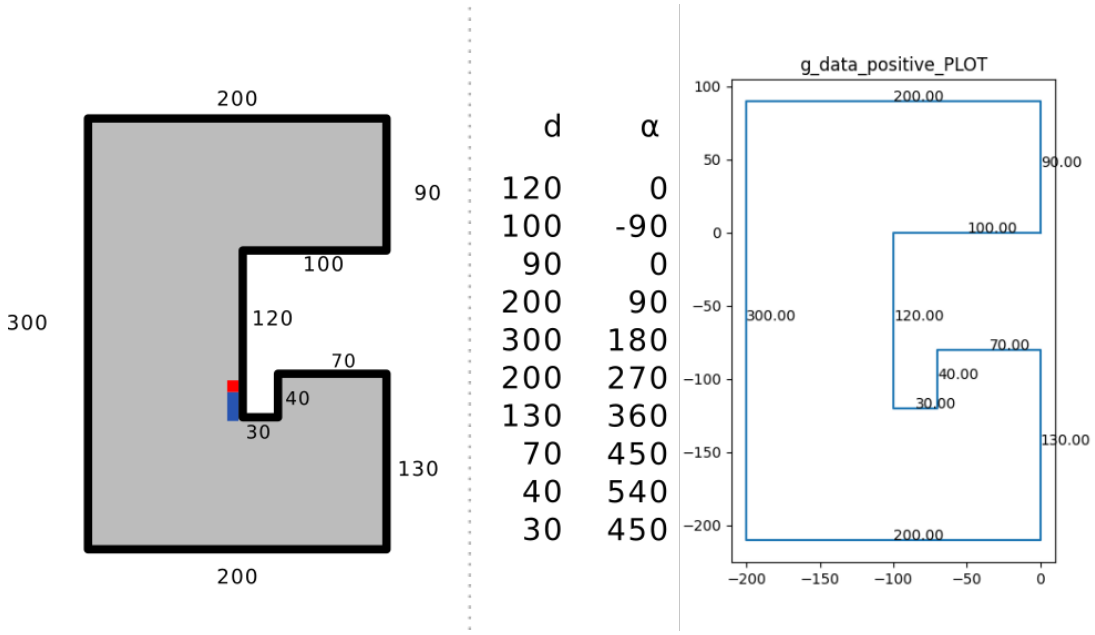

Figura 22 – Ilustração de uma geometria em formato de `G`, seus dados de entrada e a planta baixa construída pelo algoritmo.

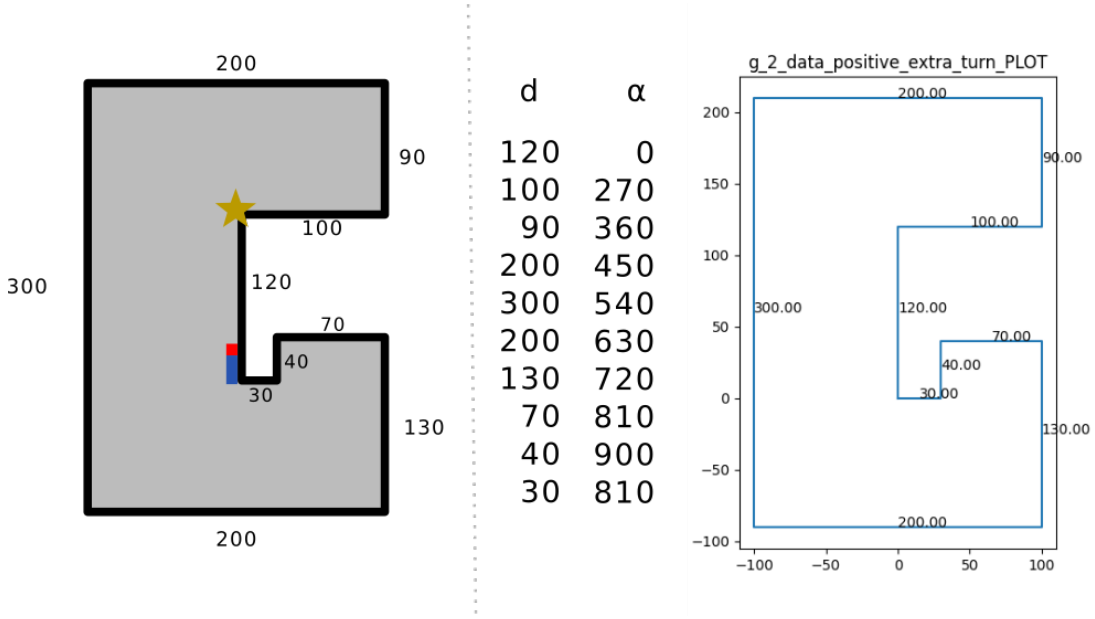

Figura 23 – Ilustração de uma geometria em formato de `G` idêntico à Figura 23, porém com seu segundo ponto de medida tendo sua rotação desempenhada como $+270°$ ao invés de $-90°$.

concentra todas as funções necessárias para a execução do algoritmo, já em sincronia com o dispositivo. Ao ser executado, é possível ser averiguado o correto recebimento dos dados e que ele realiza o controle de fluxo do algoritmo. Assim, têm-se o cumprimento dos requisitos do algoritmo e, como último passo para a verificação da adequação do dispositivo como solução, é necessário que o sistema como um todo seja testado.

6 Protótipo

6.1 Sistema

Uma vez finalizado o desenvolvimento das partes individuais e atestada sua conformidade aos seus requisitos, têm-se um sistema composto de partes menores interagentes. Devido aos testes individuais de cada parte do sistema, é esperado que cada parte cumpra com sua função de maneira autocontida. Cabe, então, aferir que as interações entre suas partes funcionam de maneira a cumprir com a as funções e finalidades esperadas dele. Visto que esta monografia possui caráter de pesquisa de viabilidade da solução, a análise do sistema não se restringe a somente aferir se o mesmo resolve o problema proposto, mas também entender como ele o faz e quais seus pontos de melhoria. No caso de alguma não adequação do sistema, mesmo havendo a adequação de suas partes aos seus requisitos individuais, é necessário não somente levantar como a interação das partes do sistema gera isso, mas também como isso se relaciona à validação da solução proposta. Por fim, um breve plano de melhorias futuras deve ser traçado para delinear o futuro do desenvolvimento do projeto.

O sistema criado como solução ao problema proposto pode ser entendido como um dispositivo de *hardware* projetado para ser utilizado junto de um *software* para desempenhar o processo de digitalização da planta baixa de um cômodo. A Figura 24 possui quatro fotos do protótipo final produzido. O dispositivo conta com projeto de eletrônica e peça mecânica próprias, onde ambos possuem características e componentes cuja análise é fundamental para responder a questão da viabilidade da solução. A escolha dos sensores passíveis de serem utilizados na digitalização para remover completamente as entradas do usuário é justamente um dos cernes dessa análise, juntamente com sua disposição espacial e como isso pode ser utilizado dentro do algoritmo. A escolha da utilização de um sensor de variação angular fornece, junto ao sensor de distância já presente em qualquer dispositivo atual, a base para toda a implementação do algoritmo e, portanto, da solução. Uma vez conhecidos os erros associados às medidas de cada um desses sensores e garantido seu funcionamento, a característica mais relevante da eletrônica como um todo para com o sistema se torna sua capacidade de prover os dados de variação angular de maneira suficiente ao algoritmo. Isso é, para além da capacidade dela de simplesmente prover os dados, a análise de sua adequação passa por verificar que os valores de saída desempenham um comportamento que vai ao encontro daquele o qual o algoritmo espera para ser capaz de realizar corretamente sua execução. As demais partes do *hardware* possuem caráter mais secundário nessa avaliação, desempenhando papéis de aproximar o uso do dispositivo da realidade, assim fornecendo maior qualidade a essa parte da análise de sistema atrelado

à eletrônica e peça mecânica.

A peça mecânica produzida por impressão 3D utilizando PLA se demonstrou capaz de cumprir com seus requisitos. A nível de sistema, seu papel dentro da análise da solução se limita ao seu impacto no uso do dispositivo por parte do usuário. Isso está fortemente atrelado a como a geometria da peça leva o usuário a realizar movimentos com o dispositivo. Idealmente, se espera que a geometria empregada, que já é bastante próxima àquelas presentes em dispositivos comerciais, force o usuário a padrões de movimento que o giroscópio seja capaz de lidar sem gerar erros no algoritmo. Outras interações não deverão ser ignoradas, mas uma vez que seus impactos se relacionam mais, ou somente, à capacidade de uso do dispositivo em sua conformação atual e não à solução em si, quaisquer conclusões advindas delas deverão ser traçadas como melhorias futuras. Esse é o caso para características suas que já estão presentes em dispositivos atuais, como sua geometria retangular e a localização do botão, ou que são de caráter funcional, como a presença do cabo USB e o seu local de saída.

O *firmware* do dispositivo escrito em `C++` e carregado dentro do Arduino Nano é o elo principal entre o *hardware* e o algoritmo de digitalização. Dessa forma, seus requisitos individuais já se assemelham bastante àquilo que é esperado a nível de sistema. Como visto na seção 5.1, suas funções de leitura dos sensores de distância e variação angular, leitura e filtragem do botão de gatilho e envio de dados cumprem com os seus requisitos. Portanto, a nível de sistema, sua parte dentro da análise de viabilidade fica contida à observação da manutenção de seu comportamento apresentado durante os testes.

A digitalização do ambiente por parte do algoritmo é o resultado final e mais importante da análise de viabilidade. Entre o *hardware* do dispositivo e o algoritmo, o último é a parte mais fácil de ser modificada. Em decorrência, a maior parte da análise de viabilidade passa por ele, analisando não somente sua conformidade, mas também o potencial de mudanças nele resultarem em uma melhor solução. O núcleo do algoritmo na parte de decisão da geometria passa pelas decisões que são tomadas por ele utilizando dos valores de ângulo. O processo de normalização, escolha do ângulo e da direção finais e o cálculo das coordenadas, assim como a ordem em que eles ocorrem, deve ser analisado a nível de sistema como um conjunto de passos que dá à solução sua capacidade de lidar com os comportamentos do usuário. Qualquer resultado negativo nos testes de sistema deverão levar os questionamentos acerca de melhorias primeiramente para essa parte do sistema, justamente por ele ser o principal objeto de análise desta monografia. Sua análise não deverá se conter somente a discorrer sobre suas capacidades ou falta de adequação ao comportamento do usuário e deve também incluir adições e modificações no algoritmo.

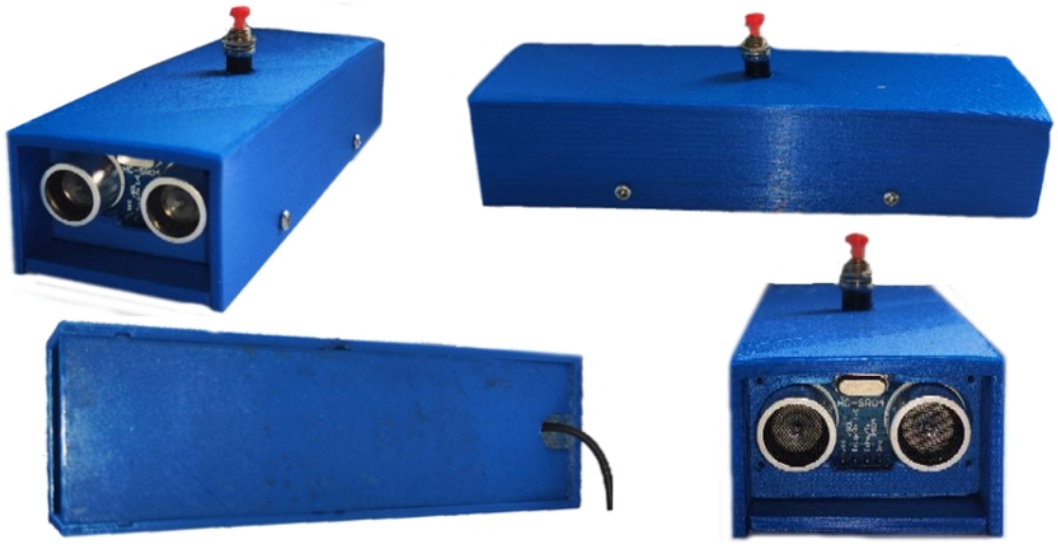

Figura 24 – Fotos do *hardware* do protótipo final produzido.

6.2 Teste de sistema

Apesar das partes individuais que compõem o sistema possuírem testes próprios que comprovam sua adequação aos seus requisitos, o sistema como um todo deve ser testado. Os testes a nível de sistema possuem intuito de averiguar capacidades do dispositivo como um sistema único. Logo, eles se assemelham ao máximo ao uso esperado do dispositivo por parte do usuário. Seu caráter é comprobatório da capacidade do sistema em solucionar o problema proposto e, portanto, o critério de sua validação é a obtenção do resultado final. Aspectos menores que não invalidem o resultado final e que demonstrem pontos de melhoria deverão ser levantados. Tais aspectos somente invalidarão a solução se impactarem negativamente o resultado final a ponto dele não representar mais o cômodo medido.

O teste de sistema consistirá no uso do dispositivo junto do algoritmo de reconstrução para a digitalização de três geometrias diferentes. Durante esses testes, deverão ser atestados pelo menos duas ocorrências dos seguintes comportamentos por parte do usuário por geometria: rotações completas, rotações não-usuais, registro duplicado de um mesmo dado e rotações de 180°. Os vértices onde cada um dos comportamentos foram desempenhados deverá ser anotado. Ao final, a digitalização obtida deverá ser comparada com a geometria do cômodo testado e verificado se são equivalentes. Discrepâncias no valor de distância medido para com os valores reais não serão considerados invalidantes do teste desde que eles não acarretem em uma digitalização inválida da geometria, como, por exemplo, com arestas se cruzando ou a não conexão do ponto final e inicial por uma margem de erro grande frente o número de pontos na geometria. Como temos um erro associado à medida de distância individual de cada parede e esse valor é utilizado para traçar a geometria, o número de paredes do cômodo diretamente impacta a propagação do erro de distância. Cada uma das geometrias é definida por um cômodo a ser medido. Visando

assemelhar os testes ao máximo ao ambiente de uso real, optou-se por realizar os testes justamente em um cômodo real. Assim, fica garantido que a locomoção e outros padrões de movimento do usuário serão contabilizados nos testes, indo ao encontro da necessidade central deles de analisar a robustez do sensor de variação angular e do algoritmo frente tais comportamentos.

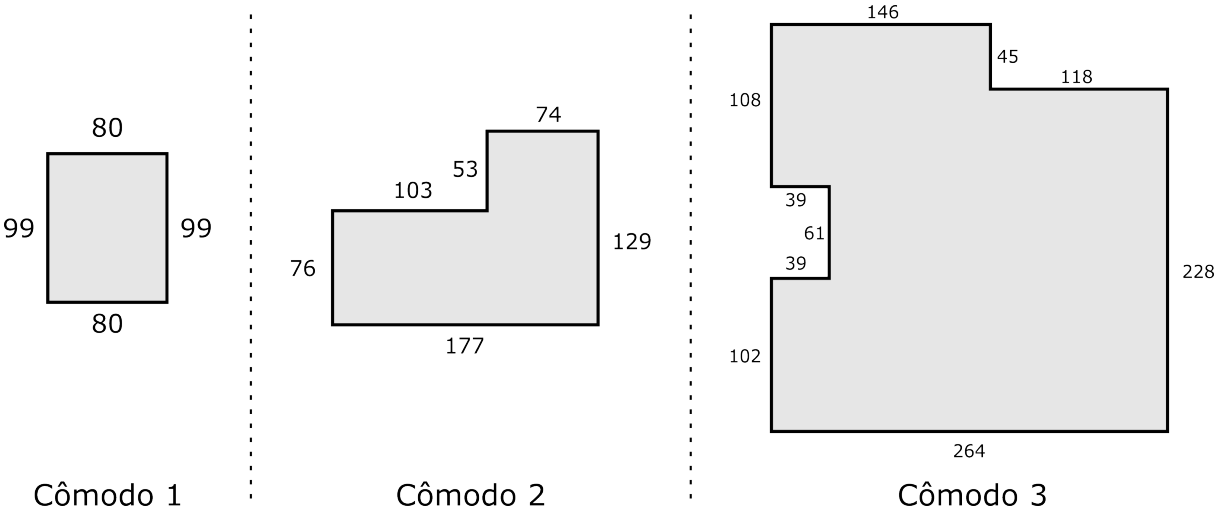

Figura 25 – Ilustração das plantas baixas dos três cômodos de teste. As unidades de medidas estão em cm e os cômodos em proporção entre si.

6.3 Resultados

O resultado ideal dos testes de sistema consiste na reconstrução da planta baixa real criada pelo algoritmo de reconstrução após sua execução no ambiente de testes. Cada uma delas deverá ser comparada com os resultados ideais esperados, ilustrados na Figura 25. A principal análise de interesse é aquela do impacto dos padrões do usuário na geometria reconstruída. Apesar de não negligenciável, o impacto dos erros do sensor de distância é de caráter secundário, a menos que eles alterem de maneira discrepante a própria geometria, resultando em cruzamentos de arestas ou o não fechamento da geometria por uma margem grande a ponto de gerar distorções.

Pela natureza do processo de tomada de medidas em um cômodo de dimensões reais, muitos movimentos inesperados, inerentes ao processo, serão desempenhados durante o manuseio do dispositivo. Além de comportamentos não usuais, porém esperados, como a retomada de um dado recém tomado e rotações de $\pm 90°$ sendo desempenhadas como rotações de $\mp 270°$, haverá também movimentações por parte do usuário decorrentes do seu andar pelo cômodo e do seu manuseio do dispositivo. Os resultados apresentados a seguir, uma vez que foram conduzidos em cômodos reais, embarcam em si esses comportamentos e, portanto, levantam dados acerca da robustez da solução proposta frente o comportamento do usuário. Esse fator é central à análise da viabilidade da solução como um todo e também

do uso do giroscópio apresentado como um possível componente apto a desempenhar a tomada dos dados de variação angular para essa tarefa. Essa análise pode ser tida como uma análise de robustez, pois ela trata do impacto que perturbações não usuais, porém naturais, ao uso do dispositivo têm no resultado final do sistema apresentado como possível solução.

Para o cômodo número 1 foram realizadas três repetições do processo de tomada de medidas. A Figura 26 ilustra esse processo para cada repetição. As flechas simples indicam uma rotação normal de acordo com o padrão esperado pelo usuário. Flechas circulares embaixo do número identificador do vértice indicam uma repetição indevida da medida naquele ponto (equivalente ao pressionar de novo do botão). Quando essa flecha circular se encontra abaixo de um outra flecha normal, isso indica uma rotação completa de ±360° além da rotação normal esperada. Por fim, uma identificação de ±270° abaixo de uma seta normal indica que uma rotação de 90° foi realizada como seu equivalente na direção contrária. Como ilustra a Figura 27, a geometria do cômodo foi corretamente reconstruída para todas as conformações. O sensor de distância, em condições reais de uso, apresenta, em média, erros maiores do que aqueles antes aferidos. No entanto, isso não acarretou em mudanças palpáveis no resultado final da digitalização, sendo ela considerada como suficiente.

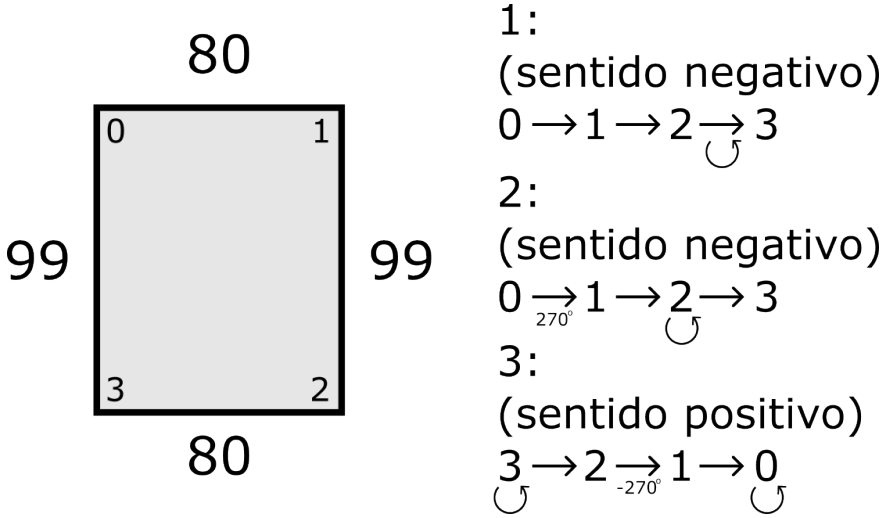

Figura 26 – Repetições de medidas para o cômodo 1. Flechas retilíneas simples indicam uma rotação esperada. Flechas circulares embaixo do número identificador do vértice indicam uma repetição da medida naquele ponto. Uma flecha circular abaixo de uma flecha retilínea indica uma rotação completa de ±360° além da rotação normal esperada. Identificações de ±270° abaixo de uma seta normal indicam uma rotação válida porém em sentido contrário ao usual. Valores em unidade de cm.

O segundo cômodo apresenta uma quina interna. Discernir entre esse tipo de vértice e vértices padrões é a principal capacidade necessária do algoritmo. Para ele, três repetições de medidas foram feitas, cada uma sendo descrita na Figura 28. Os resultados de cada

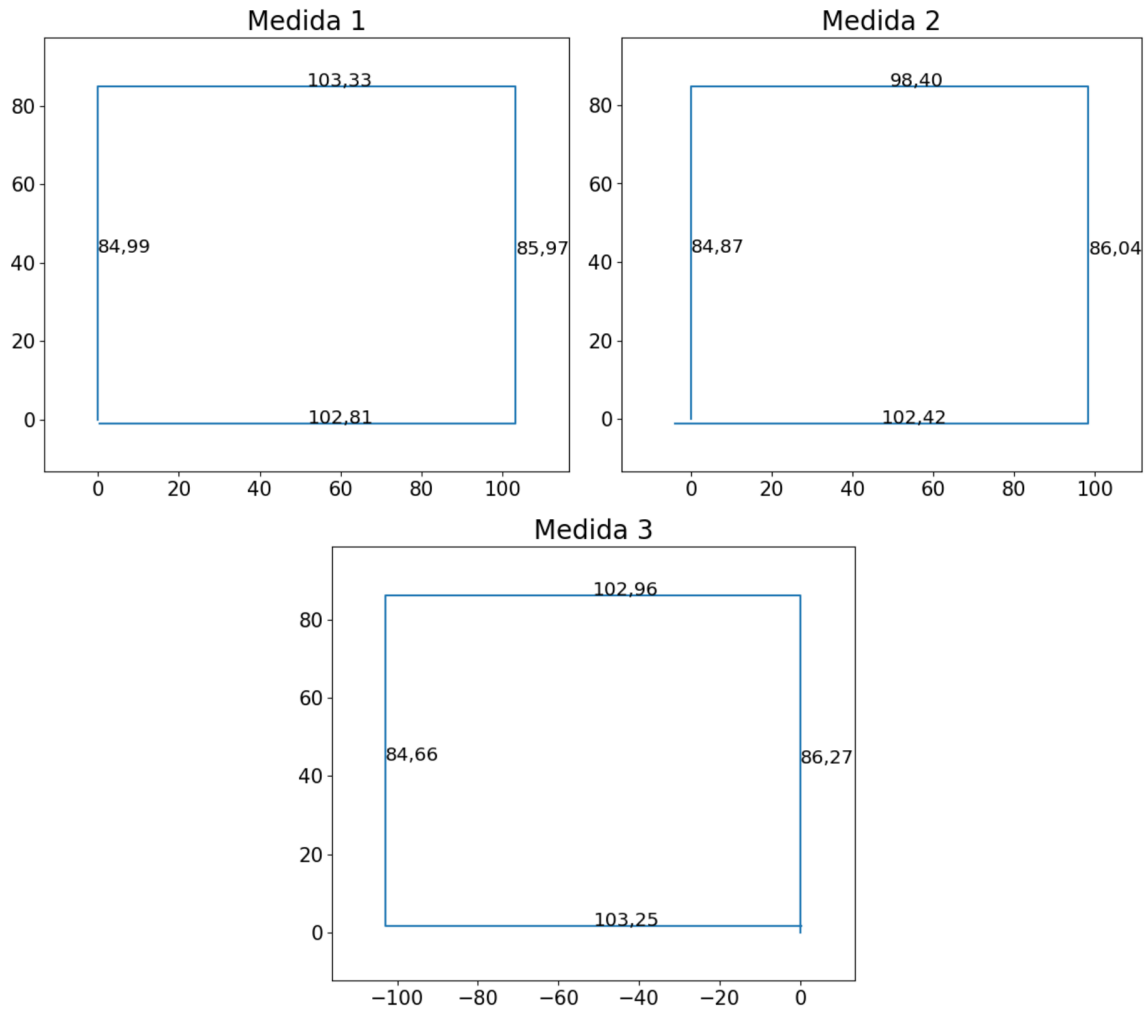

Figura 27 – Resultados da digitalização para o cômodo 1. Todas unidades estão em cm.

uma estão dispostos na Figura 29. Como pode ser visto, a terceira medida apresenta o maior. No entanto, ela demonstra um comportamento bastante importante da solução proposta: independência entre a escolha da direção e da medida de distância a cada ponto. Como pode ser visto nesse exemplo, a primeira medida (ponto 5) deveria retornar cerca de 177 cm como medida de distância, no entanto o valor retornado foi 143,56 cm. No entanto, o impacto disso na geometria foi somente o de encurtar o tamanho dessa parede e propagar esse erro às demais medidas. Isso pode ser aferido justamente pelo fato de que a soma das demais paredes nessa mesma dimensão resulta em 182,02 cm, que é próximo do valor alvo e, portanto, se a medida do ponto 5 fosse mais acurada, a última coordenada estaria muito mais próxima da inicial. Com isso, é possível aferir que o processo de reconstrução da geometria é composto de dois processos independentes: uso da variação angular para determinação dos ângulos internos e uso dos valores de distância para determinação do tamanho de cada parede. Ainda mais importante, isso permite afirmar com maior certeza que o processo de reconstrução da geometria pela parte dos ângulos foi correta para todas as medidas dessa geometria. Mesmo nessa terceira medida com um erro grosseiro em

uma de suas paredes, a geometria em si do cômodo está correta, somente com valores de distância deturpados.

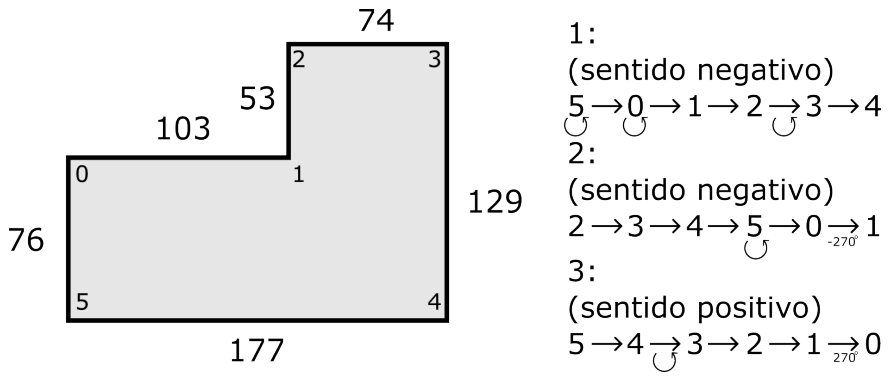

Figura 28 – Repetições de medidas para o cômodo 2. O uso das setas segue o procedimento explicado anteriormente. Valores em unidade de cm.

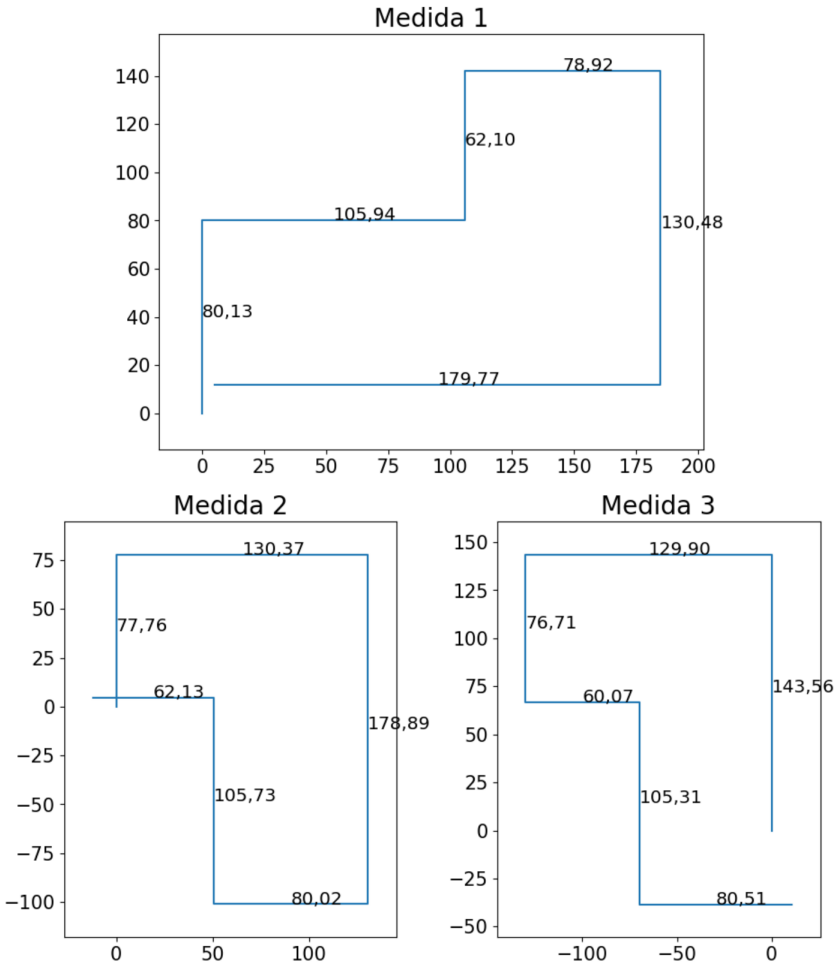

Figura 29 – Resultados da digitalização para o cômodo 2. Valores em unidade de cm.

Um outro resultado menor, passível de ser aferido, é a questão da rotação do resultado em função do seu ponto inicial. A orientação inicial do dispositivo determina qual será a rotação final da planta baixa. Isso advém de dois pressupostos por parte do algoritmo

de reconstrução. O primeiro deles é que o primeiro ponto de medida é a coordenada (0,0) e que a primeira direção a ser traçada após esse ponto é sempre no sentido positivo do eixo y. Alterando esse último pressuposto é o equivalente de realizar uma rotação no cômodo, sendo uma possível implementação simples dessa funcionalidade. Uma vez que não há razão por trás da preferência por uma rotação final ou outra, diferenças entre as medidas 2 e 3 e a medida 1 acerca desse fator não são consideradas como erros no algoritmo.

O terceiro cômodo medido é o mais complexo, e, portanto, demonstrativo, entre os três. Ele possui três quinas internas, sendo duas delas consecutivas (pontos 6 e 7 da Figura 30). Para ele foram feitas duas medidas diferentes, cada uma em um sentido de rotação. Os resultados de cada medida podem ser vistos na Figura 31. Assim como nos demais cômodos, é possível verificar que o sistema cumpre com o seu objetivo de reconstruir a geometria do cômodo. Novamente, o erro associado a cada medida individual de uma aresta acaba sendo propagado para todas as subsequentes arestas do mesmo eixo, o que significa que o erro final tende a escalar com o número de pontos na geometria. No Capítulo 7 isso será abordado com maior profundidade.

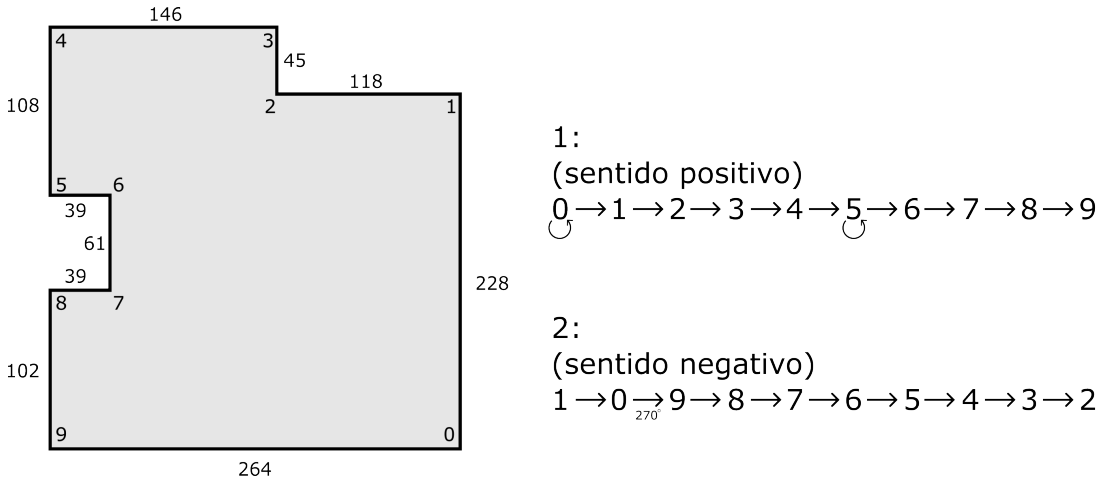

Figura 30 – Repetições de medidas para o cômodo 3. O uso das setas segue o procedimento explicado anteriormente. Valores em unidade de cm.

Todas digitalizações apresentadas demonstraram sua conformação às expectativas tidas para com o sistema. Apesar da presença de erros de medida por parte do sensor de distância, torna-se evidente a correta reconstrução do ambiente para os três cômodos apresentados em todas as suas repetições de tomada de medidas. O erro associado ao sensor de distância se demonstrou maior durante os testes do que aqueles obtidos anteriormente na Tabela 2. No entanto, isso não afeta significativamente o resultado final, assim demonstrando que os processos de decisão com base no ângulo e a tomada do valor de distância são independentes. Esse aspecto reforça a premissa de que a adição do sensor de variação angular é factível em sensores comerciais, não sendo necessário demais reformulações.

O principal resultado obtido foi a capacidade do sensor de variação angular, junto

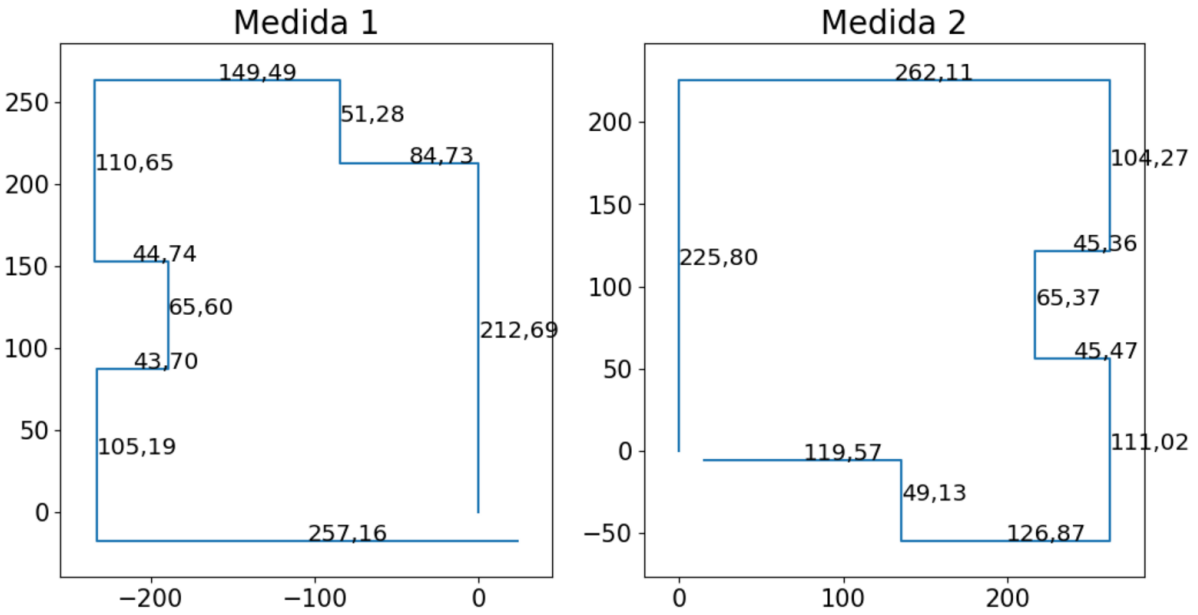

Figura 31 – Resultados da digitalização para o cômodo 3. Valores em unidade de cm.

do algoritmo, de discernir corretamente entre os ângulos possíveis. Ela é o cerne da análise de viabilidade a ser realizada, uma vez que as demais capacidades do dispositivo são de caráter auxiliar a esse. Tal resultado indica que o sensor de variação angular utilizado, o `MPU-6050`, corretamente contabiliza as movimentações do usuário durante o processo de medição. Ele demonstra que tais movimentações são corretamente separadas daquelas de interesse ou que seu efeito não é suficientemente grande para que o algoritmo não seja capaz de lidar com eles.

7 Discussão e Conclusões

7.1 Análise de viabilidade

O principal ponto a ser aferido dentro da análise de viabilidade da solução é sua capacidade de discernir e decidir quais ângulos compõem a geometria do cômodo com base na movimentação do usuário e, por consequência, do dispositivo, durante a tomada de medidas. As demais capacidades do protótipo, que estão somente relacionadas a esse processo, serão também tratadas nessa análise. Devido ao seu caráter não central, quaisquer necessidades de alterações no projeto deverão ser relegadas a uma seção dedicada a melhorias futuras (seção 7.2).

A viabilidade da solução de realizar a correta reconstrução da geometria do cômodo está primeiramente relacionada à sua capacidade de coletar dados de variação angular e, então, tratá-los. Como ilustrado na Figura 29, quando há falta de acurácia do sensor de distância, a decisão acerca do ângulo permanece inalterada. Esse erro é propagado às demais paredes pertencentes ao mesmo eixo, no entanto, o impacto disso no resultado final da digitalização fica restringido somente à alteração dos tamanhos das paredes. Apesar disso alterar a geometria final, o algoritmo segue tendo acertado suas decisões em relação aos ângulos internos do cômodo. Como mencionado na seção 6.3, isso indica a independência entre a decisão acerca do ângulo e a tomada do valor de distâncias. Assim, não se faz necessário que a análise se aprofunde nessa questão do sensor de distância, também visto que dispositivos comerciais apresentam grande qualidade nesse quesito.

Um dos pontos centrais do teste de sistema é que, uma vez que são executados em ambientes reais, os seus resultados demonstram diretamente a capacidade da solução em lidar com a movimentação do usuário durante o seu uso. Como as plantas baixas para todas as repetições de todos os cômodos apresentados na seção 6.3 foram feitas corretamente com relação aos seus ângulos, é possível concluir que a união do sensor de variação angular com o algoritmo proposto desempenha corretamente sua função. Uma vez que esses dois fatores e seu funcionamento conjunto são a essência da questão da viabilidade, têm-se a solução como viável. Em especial, o sensor de variação angular utilizado, o `MPU-6050`, desempenha um papel importante nisso. Como é um *motion-tracking device* (dispositivo de rastreamento de movimento), ele faz uso de três eixos de rotação e três eixos de acelerômetro. Por mais que somente um desses eixos de rotação seja de interesse ao processo de digitalização, isso resulta numa independência maior entre esses eixos. Por ser um dispositivo do tipo MEMS, ou seja, possuir uma estrutura vibracional que é utilizada para quantificar a movimentação em cada um de seus eixos, a presença de mais eixos implica que essa estrutura vibracional é capaz de separar de maneira adequada as movimentações em cada um deles. Em outras

palavras, movimentos do usuário para além das rotações de interesse passam por uma separação e, portanto, podem ser melhor removidas do processo como um todo uma vez que afetam menos ou não afetam o valor de variação angular no eixo de interesse.

Essa capacidade do sensor de variação angular fornece ao algoritmo aquilo necessário para que os tratamentos desses dados resultem em uma correta identificação dos ângulos internos do cômodo. Por meio da normalização do ângulo e o processo de escolha de um ângulo final de acordo com a Equação 5.3 e ilustrados na Figura 16, o algoritmo reconstrói de forma correta o cômodo medido sem ser necessária nenhuma entrada de dados por parte do usuário durante o processo. Ele também é capaz de lidar com dados indicativos do mau uso do equipamento, como repetições indevidas da tomada de um mesmo dado, rotações completas no seu próprio eixo e rotações válidas, porém em sentidos não usuais. Apesar de serem funcionalidades individuais, todas elas dependem do correto funcionamento do sensor de variação angular e também aferem seu correto funcionamento e viabilidade. Essas funcionalidades embarcadas ao algoritmo dão a ele maior robustez para lidar com movimentos naturais do usuário e, por conseguinte, são partes fundamentais para o bom resultado do algoritmo.

Por fim, outros fatores não descritos na análise são considerados como secundários. Dessa maneira, entende-se que o impacto individual de cada um não é suficientemente relacionado à solução como um todo. Em quase sua totalidade, são componentes, funções ou fatores necessários à produção de um dispositivo funcional para testes; contudo, não influenciam diretamente a solução proposta. Por essa mesma razão, na seção seguinte, eles também são omitidos a menos que de alguma forma sua melhoria impacte a solução.

7.2 Melhorias futuras

Uma vez finalizado o desenvolvimento proposto ao projeto, possíveis melhorias foram traçadas como passos naturais de um futuro prosseguimento da solução, visando principalmente sua inserção em dispositivos já existentes. Melhorias acerca do funcionamento do algoritmo, inserindo novas capacidades e removendo limitações impostas também foram traçadas.

Primeiramente, visando melhorar a acurácia e precisão dos tamanhos das paredes medidos e, em decorrência, melhorar o próprio processo de digitalização, é recomendado a troca do sensor de distância. Sensores baseados em lasers são amplamente utilizados no mercado e apresentam melhores métricas de erro. Logo, ao integrar a solução proposta a um dispositivo já existente, isso ocorrerá naturalmente. A diminuição do erro associado torna mais factível a implementação de correções automáticas de erro, se isso ainda for necessário. Um exemplo seria utilizar a diferença entre as coordenadas dos pontos finais e iniciais para obter uma medida de erro em cada eixo e, então, distribuí-lo igualmente

entre as medidas de distância das paredes de cada eixo. Essa ideia, no entanto, exige não somente um sensor de distância preciso, como também um estudo acerca de seus efeitos, uma vez que ele potencialmente pode deturpar medidas precisas em detrimento de uma única medida não precisa ou não ser um comportamento desejável.

Acerca do algoritmo de digitalização, a adição de alguma forma incipiente de interface de usuário seria de bastante ganho ao projeto. Nela, poderia ser utilizado alguma ferramenta de desenho mais propícia que o `matplotlib`, demonstrando sem interrupções[a] o avanço passo-a-passo do algoritmo assim que um novo dado chega. Ainda mais, isso tornaria a interface com o usuário, tanto antes como durante a tomada de medidas, mais informativa. Com ela, o usuário poderia alterar a rotação do cômodo[b] e realizar correções, se necessário, o que atualmente não é possível sem reiniciar o processo de tomada de medidas. Isso abre também a possibilidade de modos diferentes de uso, como, por exemplo, utilizar a média entre medidas repetidas como o valor final ou somente a última medida da série de repetições, alterando, assim, como o algoritmo trata dados repetidos.

Outra possibilidade de melhoria acerca do algoritmo de reconstrução, dessa vez alterando o cerne de seu funcionamento, é a remoção de suas restrições acerca dos ângulos internos possíveis (isto é, $\pm 90°$). Alterando como é feita a decisão com relação à direção e, por conseguinte, o cálculo das coordenadas, é possível dar suporte a qualquer valor de ângulo medido. Isso, contudo, exige um desprendimento grande de desenvolvimento para modificar o comportamento atual, além de exigir um estudo robusto sobre a precisão e acurácia do sensor de variação angular, uma vez que sua margem de erro aceitável seria ordens de magnitude menor do que a utilizada. Uma solução intermediária, entre a total remoção de restrições de ângulo e a forma atual do algoritmo, seria deixar ao usuário final a escolha dos valores de ângulo possíveis. Utilizando da mesma lógica que gerou o comportamento da função de decisão, Equação 5.3, e o comportamento visto na Figura 16, é possível ordenar todos os ângulos inseridos pelo usuário, adicionar uma zona de valores considerados inválidas entre cada um dos valores e, dinamicamente, calcular o intervalo de decisão no entorno de cada ângulo. Efetivamente, isso seria o equivalente de desenvolver um segundo algoritmo menor que desempenha a automação da criação de uma função análoga à Equação 5.3 para os ângulos de interesse do usuário.

[a] Atualmente, o *script* de reconstrução pode desempenhar essa digitalização ponto-a-ponto durante o processo de tomada de dados, mas devido à maneira como o `matplotlib` funciona, isso implica em que, toda vez que uma nova aresta é desenhada, o usuário precise fechar a janela do desenho.

[b] Como foi visto em seção 6.3, o algoritmo pode realizar isso facilmente alterando sua direção inicial dentro da classe `AngleWheel`.

7.3 Considerações Finais

O presente trabalho trata da execução de um projeto de engenharia dentro das áreas de metrologia, instrumentação física e microcontroladores. Ele discorre sobre a viabilidade da solução proposta para o problema da automação do processo de digitalização de plantas baixas. Tal viabilidade é analisada de forma a traçar quais mudanças de *hardware* e quais algoritmos seriam necessários para serem integrados aos dispositivos digitais de medição já presentes atualmente no mercado. Para isso, um protótipo demonstrativo foi projetado e produzido utilizando de uma metodologia de desenvolvimento guiada por testes e uma abordagem focada em engenharia de sistemas.

A solução proposta, como um todo, trata tanto de questões de *hardware*, ao propor adição de componentes, como também de *software*, ao desenvolver o algoritmo responsável pela digitalização a partir dos dados coletados pelo dispositivo. Durante esse desenvolvimento, cada parte individual, assim como seus componentes, possuíram requisitos individuais para sua conformação, os quais foram desdobrados de requisitos a nível de sistema. Esses, por sua vez, são criados a partir de uma definição de uma solução demonstrativa traçada a partir das demandas trazidas por um profissional da área de design de interiores. Os requisitos de cada parte foram traçados visando garantir aquilo necessário ao desempenho de sua função dentro do sistema e, para aferir isso, testes de verificação, documentação e medida/funcionalidade foram conduzidos. Uma vez atestado o cumprimento de cada um desses requisitos de *hardware* e *software*, foram elencados testes de sistema para, assim, com um protótipo demonstrativo já produzido e funcional, testar sua viabilidade como solução ao problema proposto.

Através dos resultados obtidos nos testes de sistema, (seção 6.3) foi realizada a análise da viabilidade da solução. Nela, é concluído que o algoritmo proposto e o uso de um sensor de variação angular, da forma como ambos foram desenvolvidos durante esta monografia, apresentam indicativos suficientes acerca de sua viabilidade. Os resultados apresentados, uma vez que foram conduzidos em ambientes reais, corroboram essa conclusão ao demonstrarem, de maneira clara, que a solução é capaz de gerar resultados adequados. As demais partes do protótipo que foram desenvolvidas junto à solução possuem caráter auxiliar na comprovação desse resultado ao aproximarem ainda mais o protótipo da realidade e permitirem que seu uso seja mais fidedigno. Como todas estão presentes em dispositivos comerciais e não são focos do presente estudo, considera-se que elas não são impactantes à análise de viabilidade. Dessa forma, baseado nos resultados apresentados, conclui-se que a solução proposta é capaz de resolver adequadamente o problema caracterizado e apresenta indícios suficientes de ser viável sua implementação em dispositivos comerciais.

Referências

ADA INSTRUMENTS. *COSMO MICRO 25 Operating Manual*. [S.l.], 2012. Disponível em: <https://adainstruments.com/instructions/Manual%20ADA%20COSMO%20MICRO%2025%20ENG.pdf>. Citado na página 13.

ARDUINO. *Arduino IDE (versão 2.0.0)*. Disponível em: <https://www.arduino.cc/en/software>. Citado na página 49.

Jr. W. H. Bangs. 1864. 45,372. Disponível em: <https://patentimages.storage.googleapis.com/1c/1a/0d/394ec1d821ba98/US45372.pdf>. Citado na página 11.

BOSCH. *Medidor Laser GLM 50 C*. Disponível em: <https://www.youtube.com/watch?v=fdzoj-ga4VU>. Citado 2 vezes nas páginas 13 e 23.

BROWN, S. *The C4 model for visualising software architecture*. 2019. Disponível em: <https://c4model.com/>. Citado 5 vezes nas páginas 8, 58, 59, 60 e 61.

Jean-Claude Dumas. 1981. US4254478A. Disponível em: <https://patents.google.com/patent/US4254478>. Citado na página 11.

ELECFREAKS. *Ultrasonic Ranging Module HC - SR04 Datasheet (PDF)*. [S.l.]. Disponível em: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. Citado 2 vezes nas páginas 30 e 34.

INSTRUMENTS, A. *ADA PHOTO PLAN*. Disponível em: <https://play.google.com/store/apps/details?id=ru.ada.adaphotoplan&gl=US>. Citado na página 13.

INVENSENSE. *MPU-6050 Datasheet (PDF)*. [S.l.], 2012. Revision 3.3. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/517744/ETC1/MPU-6050.html>. Citado 3 vezes nas páginas 30, 31 e 36.

KLEIN, H. A. *The Science of Measurement*: A historical survey. [S.l.]: Dover Publications, 2011. Citado na página 11.

POSTAL, G. *Furnarius Rufus PCB Milling Machine*. Disponível em: <https://ohwr.org/project/fr_pcb_mm/wikis/home>. Citado na página 33.

RIEGEL, J. *FreeCAD (versão 0.19.3)*. Disponível em: <https://wiki.freecad.org/About_FreeCAD>. Citado na página 39.

SMEATON, W. A. The Foundation of the Metric System in France in the 1790s : The Importance of Etienne Lenoir's Platinum Measuring Instruments. *Platinum Metals Rev.*, 2000. Disponível em: <https://technology.matthey.com/article/44/3/125-134/>. Citado na página 11.

Jr. Gustave H. Soule. 1977. US4031360A. Disponível em: <https://patents.google.com/patent/US4031360>. Citado na página 11.

STATOLOGY.ORG. *Confidence Interval for the Difference Between Means*. 2020. Disponível em: <https://www.statology.org/confidence-interval-difference-between-means/#:~:text=Aconfidenceinterval(C.I.)for,acertainlevelofconfidence>. Citado na página 45.

Apêndices

APÊNDICE A – Critérios de decisão de design

Hardware

Capacidade máxima de medida

A capacidade de medida da solução demonstrativa poderá ser menor do que aquela ideal, tendo seu valor máximo de medida 2 m. Tal mudança é pensada a partir da presunção de que o sensor utilizado, apesar de uma parte vital do projeto, é um item cujas capacidades são atreladas unicamente a si próprio. Isto é, a sua capacidade de medida em metros não é definida por outras partes do projeto e, por si só, não impacta a viabilidade do método de automação a nível de prova de conceito.

Capacidade de medida em diversas superfícies

Para uma prova de conceito não é necessário dar suporte a diversas superfícies, visto que o objeto de interesse é a reconstrução do ambiente. Além do mais, no item A.1.1 vemos que o sensor de distância poderá ser diferente daquele do produto final, tornando-se provável que esse grupo de materiais suportados seja alterado quando dessa troca.

Invariabilidade frente à presença de poeira

Assim como A.1.2, essa demanda está relacionada a A.1.1. A alteração do sensor implicaria na alteração dessa capacidade e, portanto, para uma prova de conceito, manter tal necessidade seria contraprodutivo e não geraria resultados aferíveis.

Geometria

A geometria do protótipo deverá ser o mais semelhante possível àquela do dispositivo final pois ela é parte essencial daquilo que o algoritmo de reconstrução pode assumir sobre o processo.

Alimentação

A fonte de alimentação do dispositivo não impacta diretamente na viabilidade do processo de reconstrução. Mantém-se somente a necessidade de não ser alimentado por rede doméstica e que a alimentação não torne o dispositivo não portátil.

Dimensionamento

Para o protótipo, os valores máximos do dimensionamento podem ser menos restritos. A nível de uso a única demanda que dialoga com a prova de conceito é a necessidade de portabilidade.

Envólucro

A geometria do dispositivo está diretamente atrelada à projeção de seu envólucro externo. Dessa maneira, ainda se faz necessário ao protótipo a projeção e manufatura da mesma, no entanto, com a remoção da necessidade de haver um acesso fácil às baterias.

Laser de alinhamento

Não será adicionado ao protótipo, pois possui caráter de auxílio da experiência de usuário e, portanto, não contribui para a conclusão da viabilidade da solução proposta.

Indicadores de nivelamento

Para o correto uso do dispositivo é necessário que haja o nivelamento dele em relação ao assoalho. Desde que essa condição seja suprida, ela poderá ser feita de qualquer maneira.

Botão de gatilho de medida

É necessário que haja delimitação entre medidas, no entanto, para a prova de conceito, ela poderá ser feita por qualquer método, não necessariamente usando um botão.

Indicação de início e finalização do processo de medida

Tal funcionalidade tem caráter puramente de uso e não é estritamente necessária para a prova de conceito. A necessidade estrita é que o usuário consiga saber que a medida terminou e o uso de um pequeno intervalo de tempo poderá ser suficiente.

Display eletrônico

Possui caráter puramente de experiência de usuário e não é vital nem ao correto uso do dispositivo nem à validação do mesmo.

Envio de dados

O envio de dados, apesar de uma parte importante da visão de produto final, a nível de protótipo não necessita de ser feito sem cabeamento. A demanda estrita é que os dados sejam enviados ao software.

Software

Remoção de entradas pelo usuário

O objetivo do projeto é a minimização das entradas externas de informação por parte do usuário. No entanto, a completa remoção não é garantida. O estudo da viabilidade da total remoção será parte do projeto e têm-se isso como objetivo não primário, sendo ele possível consequência da minimização.

Algoritmo em tempo real e integração a outros softwares

Não é uma necessidade que comprove as capacidades da prova de conceito. O algoritmo em tempo real poderá ser uma capacidade futura do produto e poderá demonstrar um estágio superior do algoritmo, mas não é necessário para a prova de conceito. A integração com outros *softwares* é um problema inteiramente de Ciência da Computação e também não traz mérito à viabilidade do conceito.

Compatibilidade para com dispositivos portáteis

Assim como a integração com outros *softwares* de modelagem e desenho técnico, essa necessidade não comprova o método de reconstrução.

Capacidades de correção no *software*

Visto que não há validação do projeto associada a essa implementação, ela é descartada.