

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENG. DE CONTROLE E AUTOMAÇÃO

NATASHA GOMES DE ANDRADE

**AUTOMAÇÃO DE LEITOS
HOSPITALARES VOLTADA AO
PACIENTE**

Porto Alegre
2019

NATASHA GOMES DE ANDRADE

**AUTOMAÇÃO DE LEITOS
HOSPITALARES VOLTADA AO
PACIENTE**

Trabalho de Conclusão de Curso (TCC-CCA)
apresentado à COMGRAD-CCA da Universidade
Federal do Rio Grande do Sul como parte dos re-
quisitos para a obtenção do título de *Bacharela em
Eng. de Controle e Automação* .

ORIENTADOR: Prof. Dr. Alcy Rodolfo dos Santos Carrara

Porto Alegre
2019

NATASHA GOMES DE ANDRADE

**AUTOMAÇÃO DE LEITOS
HOSPITALARES VOLTADA AO
PACIENTE**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Alcy Rodolfo dos Santos Carrara, UFRGS
Doutor pela McMaster University – Ontario, Canadá

Banca Examinadora:

Prof. Me. Alceu Heinke Frigeri, UFRGS
Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Alcy Rodolfo dos Santos Carrara, UFRGS
Doutor pela McMaster University – Ontario, Canadá

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

Prof. Dr. Marcelo Götz
Coordenador de curso
Eng. de Controle e Automação

Porto Alegre, julho de 2019.

AGRADECIMENTOS

A meus pais por todo apoio dado durante o período da faculdade, não teria conseguido nem embarcar no primeiro voo sem vocês.

A meus irmãos e toda minha família que mesmo distantes se fazem presentes.

Ao Rodrigo pela paciência e pelo companheirismo no TCC e na vida. Muito obrigada por toda assistência prestada, desde encontrar um ponto-e-vírgula (a mais – não a menos – nunca pensei que fosse passar por isso) até o ombro emprestado para chorar. Agradeço também por ter trazido sua família para minha vida que tem sido tão acolhedora e incentivadora.

Ao Godofredo, cujo único prejuízo dado foi comer o livro pego emprestado na biblioteca da Informática, mas manteve intacto o modelo da cama.

A meus colegas e amigos que a faculdade me trouxe por terem compartilhado esse momento comigo. Em especial, ao Carara, Cláuser, Vini e Jonas que viveram do começo ao fim todas as emoções ao meu lado – talvez nem tanto no meio, mas entramos e nos formaremos juntos!

Aos bons professores que contribuíram para minha formação e não só demonstraram superioridade.

À UFRGS por me oferecer a oportunidade de cursar a melhor faculdade de engenharia de controle e automação do país.

Ao governo do PT por ter tido a consciência de investir em educação pública de qualidade.

RESUMO

Esse trabalho apresenta a implementação de um aplicativo de *smartphone* que proporciona ao paciente internado num leito de hospital o controle de algumas funções como, por exemplo, luminosidade. Esse aplicativo lê um código QR associado à pulseira de identificação do hospital o usuário e, por um banco de dados, libera o quarto designado. O trabalho implementa esse sistema com o sinal vindo do aplicativo passando por um banco de dados. Esse banco de dados é lido pelo sistema supervisório que comanda o controlador do ambiente. Um modelo de escala 1:10 de uma cama hospitalar foi feito para demonstrar fisicamente o conceito.

Palavras-chave: Automação, Tecnologia, Código QR, Aplicativo, Leitos hospitalares, Sistema supervisório.

ABSTRACT

This paper demonstrates the implementation of a smartphone app which provides the control of some actions such as luminosity to patients accommodated in a hospital room. This app reads a QR code in the identification bracelet of the hospital and, through a database, allows access to the room associated functions. The study implements this system with the signal coming from the app passing by a database. The last is read by the SCADA which commands the ambient controller. A 1:10 scaled model of a hospital bed was made to demonstrate physically the concept.

Keywords: Automation, SCADA, Technology, Hospital beds, QR Code, Apps.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS	10
1 INTRODUÇÃO	11
1.1 Contexto e motivação	11
1.2 Objetivos	12
1.3 Descrição da solução	12
1.3.1 Interface com usuário	12
1.3.2 Método de identificação	13
1.3.3 Acesso e armazenamento de dados	14
1.3.4 Comunicação com dispositivos controlados	14
1.3.5 Conjunto final	15
1.4 Estrutura do texto	16
2 REVISÃO DA LITERATURA	17
2.1 Código QR	17
2.2 Tecnologia em hospitais	18
2.3 Banco de dados	18
2.4 Protocolos de comunicação	19
2.4.1 Protocolos TCP e IP	19
2.4.2 KNX	20
2.4.3 ModBus	21
2.4.4 Zigbee	22
2.5 Automação de ambientes	22
2.6 Sistemas supervisórios	23
2.7 Controladores Lógicos Programáveis	23
3 METODOLOGIA	25
3.1 Criação de banco de dados e tabelas	25
3.2 Desenvolvimento do sistema supervisório	26
3.3 Desenvolvimento do aplicativo	27
3.4 Comunicação	30
3.4.1 SCADA e banco de dados	30
3.4.2 SCADA e CLP	31
3.4.3 Aplicativo e banco de dados	31

4	RESULTADOS	33
4.1	Teste das luzes	33
4.2	Teste da cama	34
4.3	Teste do quarto	35
5	CONCLUSÕES	36
5.1	Considerações finais	36
5.2	Trabalhos futuros	37
	BIBLIOGRAFIA	38
	APÊNDICE A CÓDIGOS UTILIZADOS NO TRABALHO	41
A.1	<i>Android Studio</i>	41
A.1.1	<i>Main Activity</i>	41
A.1.2	<i>Resources Activity</i>	42
A.1.3	Classe <i>GetData</i>	48
A.1.4	Classe <i>DadosPaciente</i>	49
A.1.5	Classe <i>EnviaDados</i>	50
A.1.6	Classe <i>ConnectionHelper</i>	50
A.2	<i>BluePlant</i>	51
A.2.1	Tela principal	51
A.2.2	Tela do quarto	51
A.2.3	<i>Header</i>	53
A.2.4	<i>Script</i> de envio para banco de dados	53
A.2.5	<i>Script</i> de leitura do banco de dados	54
A.3	<i>MasterTool Cama</i>	54
	APÊNDICE B PROJETO MECÂNICO	56
B.1	Modelagem 3D da cama hospitalar	56
B.2	Resultados	60
	ANEXO A REVISÃO BIBLIOGRÁFICA PARA MODELAGEM DA CAMA	63
A.1	Elementos de máquinas	63
A.1.1	Parafusos e porcas	63
A.1.2	Mancais de rolamento	64
A.2	Mecanismos	64
A.2.1	Mecanismo de quatro barras	64
A.2.2	Fuso de avanço e castanha	66
A.3	Prototipagem rápida	66

LISTA DE ILUSTRAÇÕES

1	Esquema da estrutura proposta do projeto	16
2	Exemplo de identificação de padrões de posição em um código QR .	17
3	Arquitetura do Modbus	21
4	Modelo relacional das tabelas do banco de dados TCC	25
5	Tela inicial do sistema supervisorio - versão de edição	27
6	Tela inicial do sistema supervisorio - versão do usuário	28
7	Tela de controle dos componentes do quarto	28
8	Tela inicial do aplicativo	29
9	Tela para alteração dos parâmetros do quarto	30
10	Configuração da porta de comunicação serial do CLP	31
11	Teste da cama	34
12	Modelo Fawler CM-010 GDLZ - Cama hospitalar utilizada como guia no projeto	56
13	Primeira versão do modelo 3D do protótipo da cama	57
14	Fecho prisioneiro	58
15	Modelo da cama atualizado após alterações	58
16	Mecanismo de quatro barras projetado	59
17	Configuração das chapas inclinadas aos ângulos mencionados	59
18	Peças da cama feitas pela impressora 3D com rolamentos acoplados .	61
19	Correção da envergadura da cama	61
20	Sensor de fim de curso paliativo e pino original	61
21	Ângulos das chapas	62
22	Elementos constituintes de uma rosca	63
23	Rolamento 608 ZZ/C3	64
24	Mecanismos de quatro barras	65
25	Gráfico auxiliar para equacionamento do mecanismo de quatro barras	65
26	Exemplos de mecanismos de fuso de avanço e castanha	66
27	Princípio do processo FDM	67

LISTA DE TABELAS

1	Colunas na tabela “paciente”	26
2	Colunas na tabela “leito”	26
3	Equivalência entre variáveis e pontos de comunicação	32

LISTA DE ABREVIATURAS

UFRGS	Universidade Federal do Rio Grande do Sul
QR	Quick Response
SCADA	Sistema de Supervisão e Aquisição de Dados
ANS	Agência Nacional de Saúde Suplementar
SQL	Structured Query Language
RFID	Identificação por radio frequência
CLP	Controlador Lógico Programável
PLC	Programmable Logic Controller
E/S	Entrada/Saída
PNAD C	Pesquisa Nacional por Amostra de Domicílios Contínua
IBGE	Instituto Brasileiro de Geografia e Estatística
SGBD	Sistema Gerenciador de Banco de Dados
KNX	Konnex
EIB	European Installation Bus
TCP	Transmission Control Protocol
IP	Internet Protocol
TCB	Transmission Control Block
FFD	Dispositivos de Função Completa
RFD	Dispositivos de Função Reduzida
PAN	Rede de Área Pessoal
FDM	Modelagem por Fusão e Deposição
LIFEE	Laboratório de Inovação e Fabricação Digital da Escola de Engenharia
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
ST	Structured Text

1 INTRODUÇÃO

Com o objetivo de concluir o curso de engenharia de controle e automação pela Universidade Federal do Rio Grande do Sul (UFRGS), esse trabalho implementa um aplicativo de *smartphone* que proporciona ao paciente internado num leito de hospital o controle de algumas funções, sendo elas luminosidade e posição da cama. Esse aplicativo lerá num código de resposta rápida (*Quick Response*, em inglês) a identificação do usuário e liberará o quarto designado.

O trabalho implementa esse sistema utilizando um banco de dados, simulando o do hospital, como caminho de comunicação entre o aplicativo de celular e o SCADA (Sistema de Supervisão e Aquisição de Dados). O SCADA então se comunicará com o controlador do ambiente e instruirá novos comandos. Esse sistema supervisorio também poderá ser acessado por funcionários do hospital tanto para monitoramento das funções do quarto quando ocupado quanto para manipulá-las quando desocupado.

As restrições impostas ao trabalho são as decorrentes do ambiente a ser desenvolvido ser hospitalar. Dessa forma, é preciso que o projeto seja economicamente viável e não interfira de maneira inconveniente nos equipamentos instalados. Também foi proposto que a interface seja intuitiva ao usuário.

1.1 Contexto e motivação

De acordo com a Lei brasileira, apenas uma parcela da população tem direito a acompanhante durante sua internação, sendo elas:

- Crianças e adolescentes de até 18 anos (Lei 8.069 (BRASIL, 1990));
- Mulheres grávidas e no pós-parto (Lei 11.108 (BRASIL, 2005));
- Pessoas com deficiência ou em situação de dependência (Lei 13.146 (BRASIL, 2015));
- Idosos (Lei 10.741 (BRASIL, 2003)).

Em tempo, essas leis só se aplicam a pacientes que não ponham a saúde de seus acompanhantes em risco. Portanto, grande parte dos internados não possui direito assegurado a acompanhantes, ficando a critério do hospital ou unidade de saúde em que se encontra e da disponibilidade de seus entes.

A Agência Nacional de Saúde Suplementar (ANS, 2013) divulgou um estudo indicando que o tempo médio de internação hospitalar em 2011 foi de 4,5 dias por paciente. Dada a natureza de curto prazo da internação, não é viável que o controle dessas características seja feito por um algoritmo inteligente e automático, visto que as preferências

mudam de usuário para usuário. Essa inviabilidade se dá, pois o tempo que demanda para ou o algoritmo aprender as preferências ou o usuário especificá-las é relativamente alto. Dessa forma, decidiu-se implementar um algoritmo mais simples e intuitivo para o paciente facilmente manipulá-lo.

De acordo com dados divulgados na Pesquisa Nacional por Amostra de Domicílios Contínua de 2016 (PNAD C) do IBGE (Instituto Brasileiro de Geografia e Estatística) 77,1% da população com 10 ou mais anos possuíam um aparelho de celular para uso pessoal no ano da pesquisa, sendo o RS o estado com segundo maior porcentagem de usuários, com 85%. Desse resultado, 78,9% possuíam acesso a internet por meio do aparelho. Não foram divulgados dados a respeito de quantos desses aparelhos são *smartphones*. (IBGE, 2018)

1.2 Objetivos

O objetivo geral desse projeto é proporcionar maior conforto ao paciente internado em leitos hospitalares ou em unidades de saúde, unificando o controle de sistemas de iluminação e de conforto numa só interface. Além disso, de forma a validar o resultado, projetou-se uma cama hospitalar em escala reduzida. A revisão bibliográfica para o projeto mecânico encontra-se no Anexo A e o desenvolvimento no Apêndice B.

Como objetivos específicos, podem ser citados os seguintes:

- Escolha e criação de uma interface com o usuário com um método de identificação de dados;
- escolha e criação de uma forma de armazenamento e acesso a dados;
- escolha dos atuadores e sensores do projeto;
- escolha da forma de comunicação com dispositivos controlados.

1.3 Descrição da solução

Nessa seção serão discutidas as possíveis soluções analisadas para atingir o objetivo proposto.

1.3.1 Interface com usuário

Foi definido nos objetivos do trabalho que se fizesse uma interface com o usuário que, de maneira intuitiva, se pudesse controlar funções do quarto. As seguintes soluções foram analisadas:

- Controle com fio;
- controle sem fio;
- *smartphone* do usuário.

Controles tem como vantagem não precisar conceder acesso à rede do banco de dados do hospital a dispositivos estranhos, pois os dispositivos são disponibilizados pelo próprio hospital. Em contrapartida, o próprio fornecimento desses dispositivos pode se tornar algo oneroso, tendo que repô-los se necessário. Outra desvantagem é que sempre que for se

alterar, acrescentar ou cessar alguma função, o *design* do aparelho físico terá que ser alterado para comportá-las. Além disso, no caso dos controles com fio, a fiação também terá que ser alterada quando uma atualização for feita. Os controles sem fio, por sua vez, possui uma complicação a respeito do padrão de programação e comunicação que nem sempre é universal. Teria que ser feita uma análise de modelos possíveis, com suas próprias vantagens e desvantagens.

Por outro lado, observando os dados do (IBGE, 2018), descritos no início do texto, entende-se que *smartphones* são uma ferramenta comum ao ser humano. Dessa forma, cada um já possuiria seu próprio dispositivo e precisaria apenas instalar um aplicativo disponibilizado pelo hospital. Assim, o hospital teria apenas que fornecer o aplicativo, que pode ser facilmente alterado para comportar novas atualizações. Dessa forma, optou-se pelo desenvolvimento de um aplicativo de celular como interface com o usuário.

1.3.2 Método de identificação

A interface precisa de alguma forma identificar o paciente e o quarto em que o mesmo se encontra para liberar acesso ao controle do mesmo. Com essa consideração, foram analisadas as seguintes soluções:

- RFID;
- NFC;
- leitura de texto;
- códigos de barra convencional;
- códigos QR.

O RFID foi descartado no momento em que se escolheu pelo *smartphone* como forma de interface. Esse seria uma boa solução se fossem utilizados controle pois só seria necessário uma etiqueta na pulseira de identificação do paciente e um leitor no controle.

Considerando casos em que o tamanho do código de barras importa para caber na pulseira do paciente, como, por exemplo, casos em que um paciente possua alguma doença cujos sintomas incluem perda excessiva de massa corporal, a pulseira terá um tamanho reduzido. Assim, o código QR leva vantagem em relação ao código de barras, pois ainda terá que dividir espaço com a identificação textual do paciente, forma mais fácil de identificação por funcionários do hospital.

Levanta-se então a possibilidade de aproveitar o texto com identificação já escritos na pulseira para apenas desenvolver um sistema de leitura pelo celular. A desvantagem desse método se encontra em diferentes padrões de escrita utilizados pelos hospitais, sendo necessário convencionar um tipo de letra. Além disso, é possível perder informações ao se corromper a pulseira física. Códigos QR, como visto no Capítulo 2, possuem sistemas de recuperação de informações.

Por fim, há uma maior oferta de *smartphones* que possuem câmera, para leitura do código de barras, do que dos que possuem um módulo NFC (*Near Field Communication*) atualmente. Dessa forma, concluiu-se que o código QR ainda é uma opção mais acessível. As tendências indicam que num futuro próximo as opções em radiofrequência se tornarão mais atraentes às que utilizam ótica. Considerando esse caso, nota-se que o aplicativo pode ser facilmente alterado para adaptar o novo recurso de identificação, porém ainda foi optado para esse projeto desenvolver um leitor de código QR no aplicativo para identificar o usuário.

1.3.3 Acesso e armazenamento de dados

Identificando o usuário, é necessário obter informações de qual quarto ele está e quais são as funções disponíveis para esse quarto, além de poder informar quais as funções que ele quer alterar quando ele assim o quiser. Assim, foram analisadas possíveis soluções para o acesso e armazenamento desses dados.

Uma solução possível para armazenar e acessar dados seria a criação de comunicação via arquivos de texto. Porém, notou-se uma organização reduzida e conseqüente dificuldade de encontrar os dados desejados. Concluiu-se que um recurso com uma interface já preparada para as aplicações seria melhor aproveitada, como um banco de dados, o qual é mais aprofundado na Seção 2.3.

Esse recurso um banco de dados também foi escolhido como integrante do trabalho devido à sua alta probabilidade de já estar implementado num hospital. Assim, só é necessário conectar os outros dispositivos a ele, sem necessidade de desenvolvê-lo. Nesse trabalho, no entanto, foi criado um para simulação para que os resultados pudessem ser analisados. Optou-se pelo banco de dados *Oracle* pela grande participação no mercado, sem ser feita uma avaliação mais profunda, pois considerou-se que a única relevância na escolha é ser um banco de dados de entidade relacional.

1.3.4 Comunicação com dispositivos controlados

Uma outra preocupação do trabalho é a forma de enviar e receber sinais dos dispositivos controlados para realizar o controle dos mesmo. Optou-se por utilizar um protocolo de comunicação por ser uma forma mais robusta de implementar esse sistema. Isso se dá pois dessa forma não é necessário incluir fiação elétrica a mais para cada dispositivo a ser controlado, tendo a necessidade de se alterar sempre que atualizar o sistema. Inclusive, uma das possibilidades é utilizar protocolos sem fio que eliminam de vez o uso dos mesmos.

Posteriormente, na Seção 2.4, serão apresentados em maiores detalhes alguns protocolos que fizeram parte da análise para escolha de fazer a comunicação entre supervisorio e dispositivos. Na seção atual esses protocolos serão discutidos, analisados e comparados para se optar pelo mais indicado no desenvolvimento.

Primeiramente cogitou-se utilizar protocolos característicos da domótica, vista na Seção 2.5, na implementação do sistema pela natureza similar dos dispositivos a serem controlados. Dois dos protocolos apresentados se encaixam nessa definição e disputaram o cargo: o KNX e o Zigbee.

A vantagem do KNX, como se explica na Seção 2.4, se encontra nos diferentes meios de comunicação possíveis, podendo economizar na implementação física num hospital, e em possuir um sistema descentralizado. Porém, seus custos são muito altos. Extrapolando o valor de uma única lâmpada com módulo KNX para um hospital com dezenas ou centenas de quartos com mais de uma lâmpada cada, o custo final da implementação se torna provavelmente injustificável ao hospital.

Analisando uma outra solução mais barata, o Zigbee, nota-se que apesar de não ser independente como o KNX – necessita de um coordenador na rede –, possui a vantagem da rede se rearranjar em malhas para encontrar um caminho mais eficiente entre dispositivos. Seu baixo consumo de energia, grande versatilidade e até mesmo baixa taxa de dados o tornam um protocolo indicado para aplicações domésticas. Porém, o ambiente hospitalar pode conter restrições ao uso de sinais sem fio devido a interferências e considerou-se ser um recurso pouco robusto.

Para aumentar a robustez, um protocolo de uso mais industrial, mas que (MTUNGWA; PRETORIUS; CHOWDHURY, 2018) provaram ser eficaz no ambiente hospital, o Modbus, foi analisado. Apesar de ser mais trabalhoso de se implementar em algumas funções domésticas, possui uma extensa bibliografia de muitos anos e é de livre uso. Outros protocolos da área como Profibus e Fieldbus são mais caros e suas vantagens não justificam o custo.

Necessita-se agora de um elemento de controle central para emitir e receber sinais dos dispositivos a serem controlados. Como soluções, foram analisadas a utilização de um CLP, cujas funcionalidades serão descritas na Seção 2.7, e a utilização de um computador dedicado.

Por se considerar inseguro, em questões de exposição a ameaças externas, o computador precisaria ser de uso industrial, sem conexão com redes externas. Comparando-se um computador industrial com um CLP, percebe-se que a padronização do último garante uma maior flexibilidade em trocas de aparelho, sendo necessário apenas atualizar o programa para alterar os endereços do mesmo utilizados. Isso se dá pois a comissão eletrotécnica internacional regula as linguagens permitidas em controladores programáveis, o que não acontece em computadores que podem ter diferentes sistemas operacionais.

Portanto, com o auxílio de um CLP, também utilizado por (WU; NIE; PEI, 2011) em hospitais, optou-se por essa solução. Para diminuir custos e simplificar o problema, é proposto que se utilize um CLP para todos os quartos. Apesar da desvantagem em centralizar o sistema, considerou-se que a robustez do protocolo é atrativa o suficiente para sua aplicação.

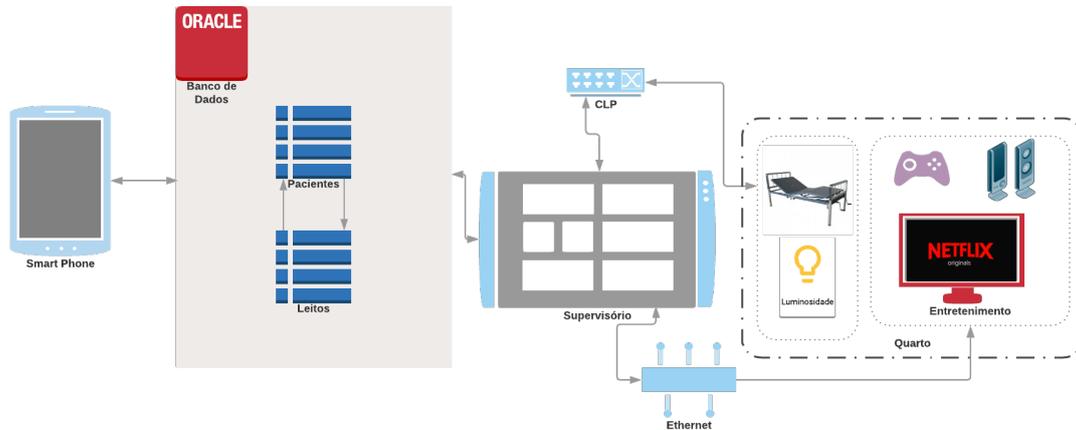
1.3.5 Conjunto final

Para atingir o objetivo proposto, o trabalho foi organizado nas etapas dispostas a seguir:

- Escolha e criação de um banco de dados genérico para hospital;
- Criação das tabelas de pacientes e de ambientes dentro do banco de dados;
- Criação de um aplicativo de *smartphone* com leitor de código QR;
- Escolha e criação de um sistema supervisório genérico de ambientes;
- Escolha e implementação da comunicação entre aplicativo e bancos de dados;
- Escolha e implementação da comunicação entre banco de dados e supervisório;
- Projeto e prototipagem do modelo da cama hospitalar para demonstração física;
- Realização de testes.

Na Figura 1 está ilustrada a estrutura idealizada para o projeto. É importante ressaltar que a parte de entretenimento não será abordada, mas está disposta na figura como uma sugestão de trabalhos futuros que incluem o conforto do paciente. Foi também idealizado o controle de climatização do ambiente, porém as normas técnicas não dão margem para alterações de temperatura. (ABNT, 2005)

Figura 1: Esquema da estrutura proposta do projeto



Fonte: Elaborada pela autora na plataforma *LucidChart*

1.4 Estrutura do texto

Esse relatório está dividido em cinco capítulos, dois apêndices e um anexo. No Capítulo 1 encontra-se a motivação por trás do trabalho, seus objetivos e discussões sobre a solução adotada. No Capítulo 2 faz-se uma revisão na bibliografia de elementos utilizados para seu desenvolvimento, divididos em seções para cada um deles de forma a familiarizar o leitor com os conceitos abordados. A Seção 2.2, em particular, apresenta um estado da arte da tecnologia hoje encontrada em hospitais.

No Capítulo 3, por sua vez, encontra-se as etapas seguidas para o desenvolvimento do projeto, com uso das ferramentas adequadas. Essas ferramentas estão também justificadas e comparadas a outras não utilizadas ao longo do capítulo. Esse capítulo aborda os passos destacados na seção anterior.

O Capítulo 4 apresenta os resultados obtidos dos testes realizados e o Capítulo 5 as conclusões constatadas a partir dos mesmos, além de sugestões de trabalhos futuros. Por fim, o Apêndice A exhibe os códigos desenvolvidos no processo de execução do projeto e, como dito anteriormente, o Anexo A e o Apêndice B descrevem a revisão bibliográfica e desenvolvimento do protótipo 3D da cama hospitalar, respectivamente.

2 REVISÃO DA LITERATURA

Nesse capítulo é feito o estudo dos itens que constituem a base desse trabalho. É organizado em seções que correspondem a cada um deles.

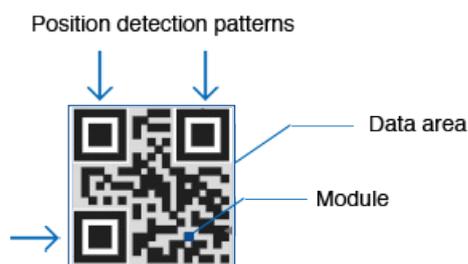
2.1 Código QR

Código QR é um código de barras bidimensional criado por DENSO WAVE cuja patente é liberada para qualquer finalidade. Nesse trabalho, é utilizado como identificador do paciente na pulseira de identificação. Assim, ao utilizar o aplicativo de celular para ler o código, se habilitado, o usuário terá acesso às funções desenvolvidas nesse projeto.

Algumas vantagens descritas do código QR, segundo seu criador (DENSO_WAVE, 2019), são as seguintes:

- Enquanto o código de barras convencional é capaz de armazenar no máximo aproximadamente 20 dígitos, o QR pode conter até 7089 caracteres, incluindo letras, números, caracteres japoneses, entre outros;
- É capaz de armazenar a mesma quantidade de dados em aproximadamente um décimo do espaço utilizado pelo código de barras convencional;
- Tem um mecanismo de correção de erros que permite restaurar dados mesmo com uma certa quantidade de impurezas;
- É lido em qualquer direção por sua detecção de posição por padrões, visto na Figura 2.

Figura 2: Exemplo de identificação de padrões de posição em um código QR



Fonte: (DENSO_WAVE, 2019)

2.2 Tecnologia em hospitais

A história da medicina é a história de humanos tentando entender melhor e tratar as várias doenças e lesões que os atingem. De simplesmente oferecer conforto e empatia, cuidados médicos evoluíram com meios de diagnóstico e tratamento cada vez mais efetivos. (STREET, 2012)

Desde o estetoscópio, segundo (STREET, 2012) provavelmente o primeiro equipamento de diagnóstico, até os atuais robôs cirurgiões a medicina vai evoluindo em conjunto com a tecnologia para ser capaz de oferecer o menor risco com maior conforto e rapidez.

Conceitos comuns na engenharia são necessários para identificar possíveis problemas, como pressão arterial, análise de ondas de eletrocardiograma e eletroencefalograma, temperatura, entre outros.

Em (AMBRÓSIO; KANETO, 2014) são citados recentes avanços em biologia computacional, como mudança nas pesquisas clínicas relacionadas ao câncer ocorridas após a introdução das técnicas de análise de microarrays e espectrometria em massa, previsão de insuficiência cardíaca, diagnóstico e terapia baseados na modelação e análise proteômica, aplicação de um algoritmo evolutivo para previsão de estrutura terciária de proteínas, entre outros.

Em (BONELLA, 2014) é ressaltada a importância de se considerar políticas amplas que sejam as mais úteis entre as disponíveis, se colocando diante de cálculos de custo-benefício e de custo-eficácia. Aplicando esse pensamento a esse trabalho, pode-se inferir que um profissional de saúde ser chamado toda vez que o paciente esteja precisando de auxílio em tarefas facilmente solucionadas por meio de automação enquanto poderia ser alocado em tarefas mais emergenciais é pouco eficaz.

(FLORENTINO et al., 2008) mostram uma aplicação de RFID num ambiente hospitalar. O texto propõe a utilização dessa tecnologia para diminuir o erro humano ao se cadastrar a identificação do paciente ao exame do mesmo. A proposta consiste em utilizar cartões inteligentes com informações do paciente, como número de identificação, nome, tipo sanguíneo, portabilidade de diabetes e hipertensão, entre outros, além da identificação do próprio cartão. Também propõe etiquetas de radio frequência nas pulseiras dos pacientes e em tubos de coleta para automatizar a leitura dos dados.

Os hospitais Moinhos de Vento e Mãe de Deus de Porto Alegre foram contatados para fins de pesquisa sobre tecnologia em hospitais da região, porém não foi obtida a resposta do primeiro e o segundo não autorizou divulgação dos dados. Não houve outras tentativas de contato com outros hospitais, da região ou não, para esse fim.

2.3 Banco de dados

Banco de dados e sistemas de banco de dados são um componente essencial da vida na sociedade moderna, sendo encontrado diariamente em diversas atividades, como ao se realizar transações bancárias, fazer reservas em hotéis ou voos ou acessando o catálogo de uma biblioteca virtual para procurar uma referência bibliográfica. Essas aplicações são exemplos de aplicações de banco de dados tradicionais, em que a maior parte da informação armazenada e acessada é textual ou numérica. Nos últimos anos, os avanços na tecnologia levaram a interessantes novas aplicações dos sistemas de banco de dados. A nova tecnologia de mídia tornou possível armazenar imagens, clipes de áudio e *streams* de vídeo digitalmente. Esses tipos de arquivo são chamados de banco de dados de multimídia. (ELMASRI; NAVATHE, 2011)

Um banco de dados é uma coleção de dados relacionados. Com dados, quer-se dizer fatos conhecidos que podem ser registrados e possuem significado implícito. (ELMASRI; NAVATHE, 2011) Sistemas de banco de dados são projetados para gerir grandes volumes de informações. O gerenciamento de informações implica a definição das estruturas de armazenamento das informações e a definição dos mecanismos para a manipulação dessas informações. Ainda, um sistema de banco de dados deve garantir a segurança das informações armazenadas contra eventuais problemas com o sistema, além de impedir tentativas de acesso não autorizadas. Se os dados são compartilhados por diversos usuários, o sistema deve evitar a ocorrência de resultados anômalos. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999)

Um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999)

Os recursos que um SGBD oferece a um usuário são: (GARCIA-MOLINA; ULLMAN; WIDOM, 2001)

- Armazenamento persistente: do mesmo modo que um sistema de arquivos, um SGBD admite o armazenamento de quantidades muito grandes de dados que existem independentemente de quaisquer processos que estejam utilizando esses dados. Porém, o SGBD vai muito além do sistema de arquivos ao proporcionar flexibilidade, como estruturas de dados que permitem o acesso eficiente a quantidades muito grandes de dados.
- Interface de programação: Um SGBD permite ao usuário acessar e modificar dados através de uma linguagem de consulta poderosa. Mais uma vez, a vantagem de um SGBD sobre um sistema de arquivos é a flexibilidade para manipular dados armazenados de formas muito mais complexas que a leitura e gravação de arquivos.
- Gerenciamento de transações: Um SGBD admite o acesso concorrente a dados, isto é, o acesso simultâneo por muitos processos distintos (chamados “transações”) ao mesmo tempo. Para evitar algumas das consequências indesejáveis do acesso simultâneo, o SGBD admite o isolamento, a aparência de que as transações são executadas uma de cada vez, e a atomicidade, o requisito de que as transações sejam executadas completamente ou não sejam executadas de forma alguma. Um SGBD também admite a resiliência, a capacidade de se recuperar de muitos tipos de falhas ou erros.

2.4 Protocolos de comunicação

Nessa seção serão apresentados os protocolos analisados para serem empregados no trabalho. A análise será feita posteriormente no próximo capítulo.

2.4.1 Protocolos TCP e IP

IP (*Internet Protocol*, em inglês) e TCP (*Transmission Control Protocol*, em inglês) são conjuntos de protocolos da *internet*, sendo o primeiro aplicado à camada de rede e o segundo à camada de transporte. (TANENBAUM; WETHERALL, 2011)

2.4.1.1 IP

O protocolo IP é o elemento que mantém a *internet* unida, tendo sido desde o início projetado com o objetivo de interligar as redes. A camada de transporte, ao receber os fluxos de dados, os divide para serem enviados como pacotes IP. Apesar desses pacotes poderem ter um tamanho de até 64kB, na prática, geralmente tem no máximo 1500 bytes e assim cabem num único quadro *Ethernet*. Os roteadores IP, por sua vez, encaminham cada pacote pela *internet*, de um roteador até o seguinte, até que o destino final seja atingido. Ao chegar ao destino, a camada de rede entrega os dados de volta à camada de transporte e, quando todos os fragmentos chegam, são remontados pela camada de rede no datagrama original. (TANENBAUM; WETHERALL, 2011)

2.4.1.2 TCP

Segundo (COMER; STEVENS, 1999), o protocolo de controle de transmissão é o mais complexo do conjunto de protocolos da *Internet*, proporcionando serviço de *stream* confiável, com controle de fluxo fim-a-fim entre duas máquinas de velocidades de processamento arbitrárias que usem o mecanismo IP não-confiável para comunicação. O protocolo foi projetado especificamente para oferecer esse controle (TANENBAUM; WETHERALL, 2011) e “cuidadosamente construído para funcionar corretamente mesmo que os datagramas sofram retardos, sejam duplicados, perdidos, entregues fora de ordem ou entregues com dados danificados ou truncados”. (COMER; STEVENS, 1999)

O TCP usa três processos separados para tratar entrada, saída e temporização de eventos. Por meio de uma estrutura de dados compartilhada por todos os processos, o TCP coordena as atividades de transmissão, recepção e retransmissão referentes a cada conexão TCP. Essa estrutura é conhecida como TCB (*transmission control block*, em inglês), sendo mantida pelo TCP uma para cada conexão ativa. “O TCB contém todas as informações referentes à conexão TCP, incluindo os endereços e números de portas dos pontos terminais da conexão, a estimativa corrente de tempo de ida e volta, dados que tenham sido enviados ou recebidos, se há necessidade de mensagem de confirmação ou retransmissão e quaisquer estatísticas que o TCP colete sobre o uso da conexão.” (COMER; STEVENS, 1999)

2.4.2 KNX

Konnex, mais conhecido como KNX, é um sistema de comunicação para controle predial, originário de EIB (European Installation Bus), que utiliza tecnologia da informação para conectar dispositivos, tais como sensores, atuadores, controladores, terminais de operação e monitores. Sua tecnologia é própria para ser usada em instalações elétricas para implementar funções e processos automatizados em prédios. (MERZ; HANSE-MANN; HÜBNER, 2009)

A missão da *KNX association*, associação responsável pelo protocolo, é “desenvolver e promover o padrão KNX para que possa ser reconhecido como base para criação de casas inteligentes e soluções prediais, sendo seguro, conectável e utilizado globalmente.” (KNX ASSOCIATION, 2019)

Os benefícios do protocolo se encontram na intercompatibilidade entre dispositivos de diferentes fabricantes, versatilidade para aplicações em diversos tipos de ambientes, diferentes meios de comunicação, como IP/Ethernet, cabo de pares trançados, rede elétrica e radiofrequência. O sistema é descentralizado, não sendo necessária uma unidade de controle central. Isso se dá ao fato de que cada dispositivo possui um microprocessador.

Dessa forma, a utilidade de um controle central se faz destacar apenas quando se deseja realizar tarefas específicas. A vantagem do sistema descentralizado se encontra no controle de erros, uma vez que ao ocorrer falha num dispositivo, apenas as tarefas dependentes dele serão afetadas, mantendo o funcionamento das outras. Podendo ainda ser realizado algum procedimento para que os outros possam o suprir. (TORRES, 2018)

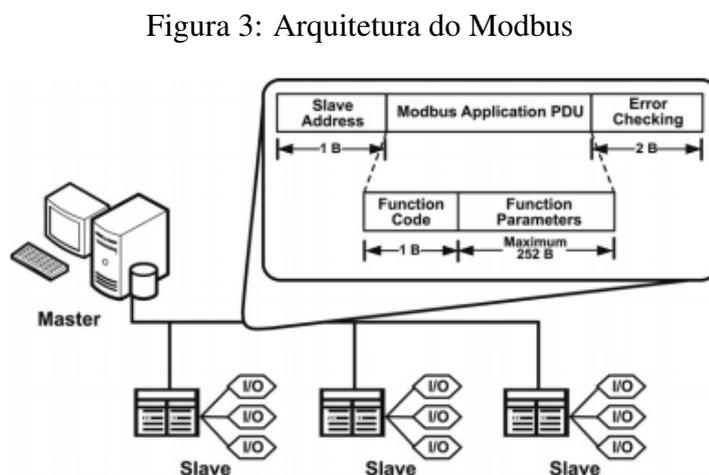
Como pontos negativos pode-se destacar que esse protocolo também é conhecido pelo seu preço exorbitante; entrando em contato com uma loja especializada do Reino Unido, encontrou-se valores entre £45 e £120 para uma lâmpada – sem o interruptor. Sem considerar os impostos, os valores chegam a R\$570.

2.4.3 ModBus

Originalmente formulado em 1979, o protocolo Modbus é um dos mais antigos, e mais amplamente utilizados, protocolos de controle industrial. O protocolo define a estrutura de mensagem e regras de comunicação usadas por, mais frequentemente, sistemas de controle industriais para transmitir sinais entre instrumentação ou dispositivos de controle e o controlador principal ou sistema de banco de dados. Modbus é um protocolo aberto, ou seja, é de livre uso para desenvolvedores criarem seu equipamento sem precisar pagar *royalties* e isso tem contribuído para sua popularidade. (HUIITSING et al., 2008)

O protocolo possui duas principais variações: Modbus Serial e Modbus TCP. No Serial, mensagens são trocadas entre um mestre e até 247 escravos (dispositivos de campo) por cabos seriais. O mais novo protocolo Modbus TCP provê conectividade entre várias redes Modbus conectadas por IP. (HUIITSING et al., 2008)

Modbus empenna um mecanismo simples de pedido/resposta entre um centro de controle e dispositivos de campo. Por exemplo, um centro de controle (mestre) pode enviar uma mensagem de leitura para um sensor (escravo) para obter o valor de parâmetro do processo. Alternativamente, pode mandar uma mensagem de escrita para um atuador (escravo) para realizar a ação de controle desejada. Há dois modos de mensagens enviadas pelo mestre: *unicast*, no qual o mestre envia para um endereço de um escravo específico e espera resposta, e *broadcast*, no qual o mestre envia a mensagem a todos os escravos e nenhum responde. As mensagens trocadas possuem 3 partes, como visto na Figura 3: o endereço do escravo, a unidade de dados do protocolo, e um campo de checagem de erro. (HUIITSING et al., 2008)



Fonte: (HUIITSING et al., 2008)

O campo de endereço tem o endereço do escravo com 1 byte de tamanho, sendo de 1 a 247, 0 para modo *broadcast* e de 248 a 255 reservados para outras funções. O campo de unidade de dados contém duas partes: código da instrução de 1 byte e seus parâmetros de até 252 bytes, caso necessário. Por fim, há uma verificação de erro para garantia de que a mensagem foi entregue sem perdas de informação. (MODBUS.ORG, V1.1b3)

2.4.4 Zigbee

De acordo com (ZIGBEE_ALLIANCE, 2019), o protocolo Zigbee é uma linguagem sem fio que dispositivos do dia-a-dia utilizam para se comunicarem uns com os outros. É um protocolo de curto alcance, de baixo custo e baixo consumo de energia. (LIU et al., 2017) citam exemplos desses dispositivos como termostatos, interruptores de luz, controles remotos e dispositivos inteligentes.

IEEE 802.15.4 especifica as camadas física e de controle para protocolos sem fio de baixa taxa. Baseando-se nessas camadas, a especificação do Zigbee estabelece um *framework* para as camadas de rede e aplicação. Há um total de 27 canais de rádio disponíveis em diferentes áreas do mundo: canal 0 (868 MHz), canais 1 a 10 (928 MHz) e canais 11 a 26 (2,4 GHz). A rede suporta 65536 endereços diferentes, o que significa que numa única rede pode conter esse número de dispositivos, podendo ser estendida. (LIU et al., 2017)

Os nós da rede podem ser classificados como coordenadores, roteadores ou dispositivos finais. Coordenadores e roteadores são dispositivos de função completa (FFD) e dispositivos finais são dispositivos de função reduzida (RFD). Só existe um coordenador na rede de área pessoal (PAN), responsável por inicializá-la. (LIU et al., 2017) Na inicialização, todos os canais da frequência de operação são escaneados até esse PAN ID único ser encontrado. O coordenador opera em estado ativo para efetuar o controle da rede e costuma ser alimentado diretamente reduzindo o risco de falha no nó centralizador da rede.

Os roteadores são utilizados para dar maior robustez à rede. Possuem tabelas de roteamento e, por serem FFD, são capazes de encontrar o menor caminho para se chegar ao destino. Caso um roteador não possua o endereço de destino requisitado na tabela, ele efetua um *broadcast* de uma requisição de rota e recebe do destino a rota mais eficaz, atualizando a tabela. Assim, ele tem o poder de se autorregenerar caso ocorra algum defeito em outros roteadores da rede. Dispositivos finais dão o endereço de sensores ou atuadores na rede. Eles se comunicam apenas com o roteador e passam a maior parte do tempo inativos.

2.5 Automação de ambientes

Há registros da automação sendo utilizada desde a Idade Antiga, sem uma maior precisão de época, sendo definida como qualquer processo que auxilie o ser humano em tarefas, sejam elas de natureza industrial, comercial, doméstica ou do campo. Como exemplo, cita-se a Roda d'Água na automatização de processos de moagem, serrarias, ferrarias e trituração de grãos em geral. (TEZA, 2002)

A automação de ambientes tem início na automação residencial, que por sua vez é um sub-campo da automação predial. Como esse trabalho trata de sistemas que podem ser encontrados em casa adaptados ao uso hospitalar, essa seção utiliza bibliografia de automação residencial para revisar as soluções utilizadas.

A integração é um dos principais temas no desenvolvimento de sistemas de automação

para residências inteligentes. Por exemplo, sensores utilizados pelo equipamento de segurança patrimonial podem ser aproveitados para também controlar a iluminação do ambiente. Outro exemplo se aplica a sensores de temperatura que atuam permanentemente como sensores de incêndio. (BOLZANI, 2004)

A informação fornecida por qualquer sensor (termostato, detector infravermelho, sensor de luminosidade etc.) é colocada à disposição através de uma rede interna à habitação ou prédio (ou, nesse caso, leito) e, portanto, está disponível para ser manipulada a qualquer momento. (PRUDENTE, 2011)

Uma vantagem de sistemas inteligentes para iluminação também é a economia de energia, pois a intensidade da luz é controlada conforme a necessidade e as lâmpadas não precisam operar com seus brilhos máximos, como acontece normalmente. (BOLZANI, 2004)

Para os sistemas multimídias, os integradores de funções se mostram uma opção interessante. “São equipamentos que controlam uma infinidade de portas seriais RS-232/422/485, portas infravermelho e ainda possui relés de saída para controle adicional. Eles unificam muitas operações de controle de dispositivos inteligentes interconectados através de um barramento de dados proprietário ou mesmo via *Ethernet* possibilitando inúmeras configurações e arranjos.” (BOLZANI, 2004)

2.6 Sistemas supervisórios

O termo SCADA na área da automação refere-se a sistemas de supervisão, controle e aquisição de dados composto por um ou mais computadores monitorando e controlando um processo. O seu principal objetivo é propiciar uma interface de alto nível do operador com o processo informando-o “em tempo real” de todos os eventos de importância da planta. (VIANNA, 2008)

Hoje os principais sistemas de supervisão oferecem três funções básicas: funções de supervisão que inclui todas as funções de monitoramento do processo; funções de operação para ligar e desligar equipamentos e sequências de equipamentos, operação de malhas PID, mudança de modo de operação de equipamentos, etc.; funções de controle. (VIANNA, 2008)

2.7 Controladores Lógicos Programáveis

“Nos últimos anos, muitas tarefas de automação nos prédios são resolvidas com o uso de microcontroladores programáveis. Estes podem ser ligados em uma rede de barramento de automação predial com o uso de módulos especiais de interfaceamento. A esses microcontroladores é possível ligar sensores do tipo industrial, o que aumenta notavelmente a performance da automação.” (PRUDENTE, 2011)

Também conhecidos como CLPs (ou *PLCs*, em inglês), os controladores lógicos programáveis podem ser definidos como controladores baseados em microcomputador que utilizam instruções armazenadas em uma memória programável para implementar lógica, sequenciamento, temporização, contagem e funções aritméticas por meio de módulos de entrada/saída (E/S) digitais ou analógicos para controle de máquinas e processos. (GRO-OVER, 2011)

“Os CLPs são talhados inicialmente para operar em ambiente industrial, onde ocorrem distúrbios elétricos, alta temperatura, vibrações. Nas últimas décadas foram desenvolvidos pequenos CLPs, chamados de microCLP, para executar pequenas tarefas, tais como

comando de portões elétricos, irrigação de jardins, gestão da iluminação, pequenos elevadores, entre outras.” (PRUDENTE, 2011)

Os componentes básicos do CLP são os seguintes: processador, unidade de memória, fonte de energia, módulo de E/S e dispositivo de programação. (GROOVER, 2011)

3 METODOLOGIA

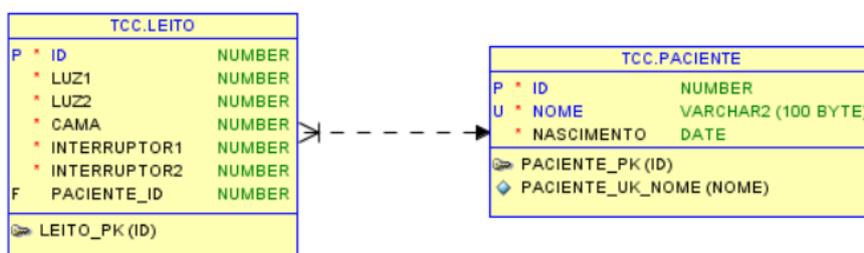
Nesse capítulo será descrito o desenvolvimento do projeto. Este também será dividido em seções que distribuem o trabalho feito em blocos definidos na subseção 1.3.5.

3.1 Criação de banco de dados e tabelas

Nesse trabalho foi utilizado o ambiente de desenvolvimento de banco de dados *SQL Developer* para criação de um banco de dados *Oracle*. Essa ferramenta foi escolhida pela interface intuitiva com o usuário de forma a facilitar a criação de um sistema de banco de dados. É uma ferramenta gratuita, porém acredita-se que o hospital já possua uma ferramenta própria, sendo necessário apenas a análise de compatibilidade do modelo do banco de dados.

Um banco chamado TCC foi criado com duas tabelas, chamadas “leito” e “paciente”, como pode ser visto na Figura 4. O conjunto simula uma parte banco de dados que possivelmente se encontra no hospital e deve ser combinado com o mesmo para obter sucesso na operação.

Figura 4: Modelo relacional das tabelas do banco de dados TCC



Fonte: Elaborada pela autora utilizando o *software SQL Developer*

Na tabela “paciente” há as variáveis de cada paciente: identificador, nome e data de nascimento. Os detalhes de cada coluna, como tipos de dados e possibilidade de possuir um valor nulo são vistos na Tabela 1. Segundo esse modelo, o hospital deve inserir o identificador do paciente como nessa tabela no campo equivalente da tabela “leito”.

Na tabela “leito” há as variáveis de cada quarto: identificação do quarto, estado da lâmpada 1, estado da lâmpada 2, estado do interruptor digital 1, estado do interruptor digital 2, posição da cama e uma variável que pode ser nula, com o identificador do paciente. A Tabela 2 mostra os detalhes de cada coluna.

Tabela 1: Colunas na tabela “paciente”

Coluna	Tipo de dado	Nulo
ID	Número	Não
NOME	Número	Não
NASCIMENTO	Data	Não

Fonte: Elaborada pela autora

Tabela 2: Colunas na tabela “leito”

Coluna	Tipo de dado	Nulo
ID	Número	Não
LUZ1	Número	Não
LUZ2	Número	Não
CAMA	Número	Não
INTERRUPTOR1	Número	Não
INTERRUPTOR2	Número	Não
PACIENTE_ID	Número	Sim

Fonte: Elaborada pela autora

3.2 Desenvolvimento do sistema supervisório

A ferramenta Elipse E3 foi cogitada a ser utilizada, mas a empresa não respondeu positivamente ao pedido de licença de uso. Já a proprietária do *BluePlant*, Altus, estava disposta a ceder uma versão mais completa caso a versão livre disponível não fosse suficiente para suprir as demandas do projeto. Além disso, também é a fabricante do CLP utilizado. Por isso, nesse trabalho foi utilizado o *software BluePlant* para desenvolver o SCADA. Foram criadas duas telas no sistema: uma tela principal inicial com ilustração de todos os quartos e seus respectivos pacientes, como visto na Figura 5, e outra de cada quarto específico, com seus componentes, em detalhes mais adiante no texto.

A Figura 5 mostra a versão de edição da tela inicial, por isso há os nomes das variáveis globais do sistema, chamadas de *Tags*, entre colchetes nos textos a serem exibidos. Essas variáveis são responsáveis por armazenar valores que podem ser lidos e alterados por qualquer parte do sistema, como em tarefas (códigos cíclicos), códigos ativados por usuários na tela gráfica ou pelo *driver* de comunicação, por exemplo.

Na tela gráfica que aparece ao usuário, vista na Figura 6, as janelas que representam quartos mudam de cor conforme presença de paciente internado. Essa condição é verificada de acordo com a *tag* de dados do paciente ligado ao quarto ser nula ou não, considera-se que quando é nula não há paciente. Acima da tela pode ser visto o cabeçalho do sistema, onde se encontra botões para navegação, nome do projeto, data e hora e campo para inserir usuário. Esse cabeçalho pode ser editado, mas manteve-se as configurações padrões dele por não pertencer ao objetivo do trabalho.

Ao selecionar um dos quartos, com paciente ou não, uma nova tela é exibida. Essa tela contém o número do quarto, os componentes a serem manipulados e um botão para voltar à tela inicial. A Figura 7a mostra um exemplo de configuração que pode ser encontrado. Como pode ser visto, há duas lâmpadas, dois interruptores e uma cama na imagem. As lâmpadas reproduzem as lâmpadas físicas do quarto, os interruptores representam o in-

Figura 5: Tela inicial do sistema supervisor - versão de edição



Fonte: Elaborada pela autora utilizando o *software BluePlant*

terruptor digital, sendo independente do interruptor físico, e a cama possui quatro estados possíveis: (i) encostos da cabeça e das pernas abaixados, (ii) encosto da cabeça abaixado e das pernas levantado, (iii), encosto da cabeça levantado e das pernas abaixado e (iv) encostos da cabeça e das pernas levantados.

Ao clicar no interruptor, ele e a lâmpada associada se alteram conforme o comando. Já ao se clicar na cama, ela muda de cor para a variante de alteração onde pode ser clicada até o usuário encontrar a posição desejada. O usuário então tem duas opções: confirmar sua seleção ou cancelar a alteração, por meio de dois botões, como pode ser visualizado na Figura 7b. Qualquer um dos botões retorna a cor da cama para a cor original, assim como clicar no botão "voltar". O botão "confirmar" mantém a posição que o usuário escolheu enquanto os outros dois botões mencionados ("cancelar" e "voltar") retornam à posição anterior ao início das mudanças na posição.

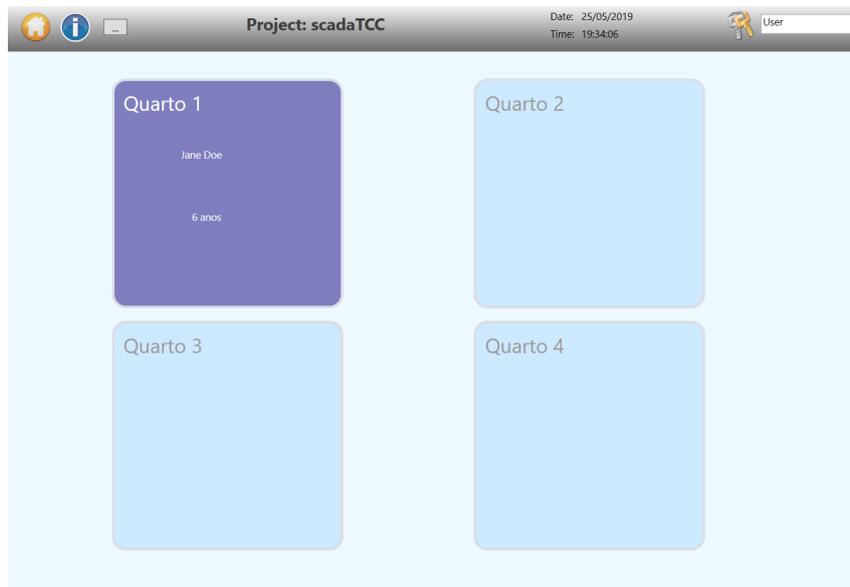
Todos os componentes presentes na tela do quarto estão associados a pelo menos uma *tag* que se modifica em conjunto com a parte gráfica. O valor dessa *tag* é utilizado para atualizar os registros adequados do banco de dados pela comunicação estabelecida com o mesmo, posteriormente no texto detalhada.

3.3 Desenvolvimento do aplicativo

O aplicativo desse trabalho foi desenvolvido utilizando a IDE (*Integrated Development Environment*) *Android Studio*. Essa ferramenta foi escolhida por ter uma grande comunidade *online*, característica importante para uma primeira experiência autônoma. A IDE cria aplicativos para o sistema Android; não foram desenvolvidas soluções para iOS. Como o SCADA, o aplicativo também foi desenvolvido de forma a ter duas telas, chamadas atividades.

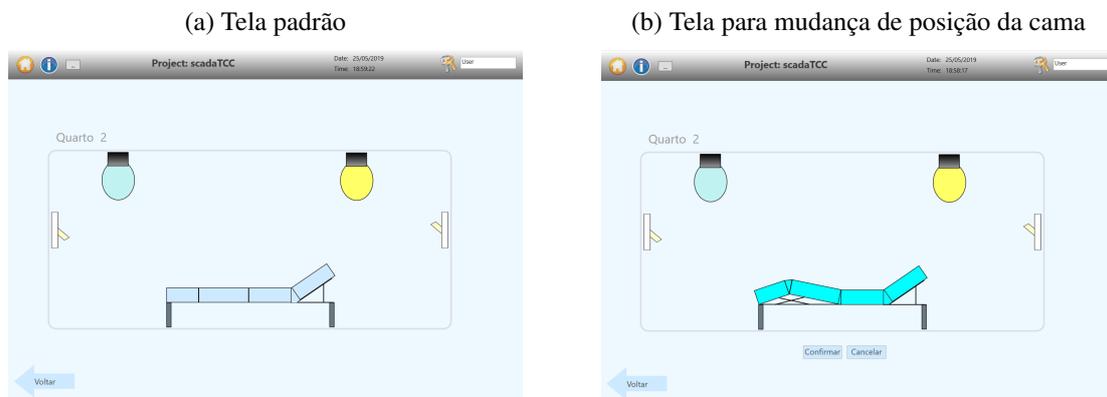
A primeira consiste na tela de *login*, onde o usuário pode clicar num botão e ativar a câmera para ler o código QR. Ela é apresentada na Figura 8a. Para realizar testes no laboratório da UFRGS foram adicionados uma caixa de texto e um botão para ser possível alterar o endereço IP do banco de dados, pois este, instalado no computador da aluna,

Figura 6: Tela inicial do sistema supervisorio - versão do usuário



Fonte: Elaborada pela autora utilizando o *software BluePlant*

Figura 7: Tela de controle dos componentes do quarto



Fonte: Elaborada pela autora utilizando o *software BluePlant*

muda ao ser realocado na rede da UFRGS. Essa tela de testes pode ser vista na Figura 8b.

A tela seguinte, após ter sido feito o *login*, é a de controle dos componentes do quarto, semelhante à do SCADA. Como pode ser visto na Figura 9, ela possui o identificador do quarto, o nome e a data de nascimento do paciente internado, dois interruptores para controlar a luz e quatro botões para controlar a posição da cama.

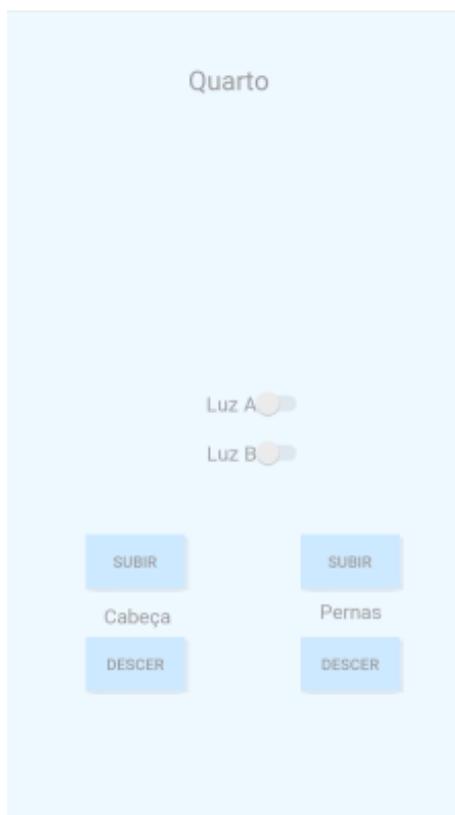
Quando o paciente tenta clicar nos botões de alterar a posição da cama, mas ela já chegou no fim de curso, uma mensagem é exibida ao usuário alertando que já atingiu o máximo ou mínimo. A abordagem usada nas funções pelo aplicativo é diferente da do SCADA porque não é um sistema supervisorio, ou seja, não precisa se observar os estados do quarto. A intenção do aplicativo é apenas o acionamento das funções e só botões bastam. Além disso, considerou-se essa solução mais intuitiva ao paciente.

Figura 8: Tela inicial do aplicativo



Fonte: Elaborada pela autora utilizando a IDE *Arduino Studio*

Figura 9: Tela para alteração dos parâmetros do quarto



Fonte: Elaborada pela autora utilizando a IDE *Arduino Studio*

3.4 Comunicação

Nessa seção, será descrito detalhadamente o desenvolvimento da comunicação entre os sistemas já expostos.

3.4.1 SCADA e banco de dados

Para a comunicação entre o *software BluePlant* e o banco de dados, foi utilizado um *driver* já implementado para a ferramenta se comunicar com o banco de dados *Oracle*. Ademais, uma configuração precisou ser feita na aba “*Datasets*” do *BluePlant* onde se criou a comunicação com o banco de dados em si e as *queries*, que são “perguntas” feitas ao banco para obter acesso aos dados, necessárias para a comunicação.

Uma comunicação com o banco de dados “TCC” e duas *queries* foram criadas; uma para receber ciclicamente os dados das tabelas e outra para atualizá-las de acordo com a requisição do usuário. É necessário receber os dados da tabela visto que eles podem ser alterados pelo aplicativo por outros usuários ou um paciente pode ser cadastrado ou descastrado de um quarto. No caso de não haver paciente no leito, a variável *ID_PACIENTE* da tabela “leitos” e todas as variáveis consequentes da tabela “pacientes” são nulas e uma implementação de caso especial foi necessário para tratá-las da melhor forma.

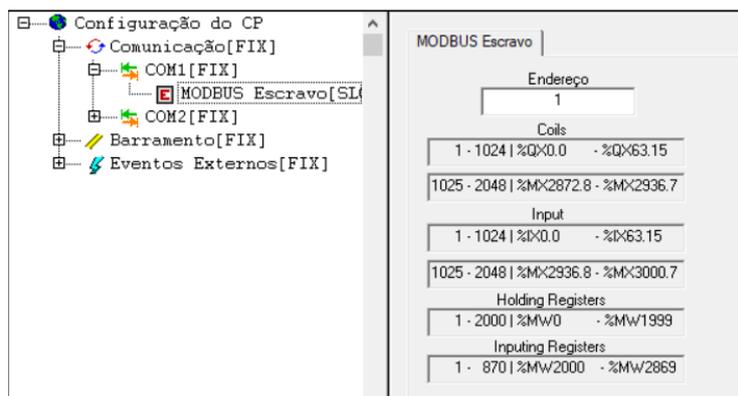
Ao se mudar algum parâmetro de um quarto, uma variável de *trigger* é ativada para a *query* de atualizar a tabela ser acionada. Então, a comunicação é feita de duas formas: em ciclos de 1 segundo para ler o banco de dados e atualizar as variáveis do sistema supervisorio e a cada vez que uma variável do sistema for alterada para atualizar o banco de dados com seu novo valor.

3.4.2 SCADA e CLP

Essa etapa do projeto consiste em estabelecer uma comunicação entre o SCADA e o CLP de forma que o ambiente exibido no supervisor seja fiel ao real. Primeiramente foram escolhidas as variáveis a serem utilizadas pelo CLP. No início pensou-se apenas em testar se as simulações das luzes estavam de acordo, feitas com os indicadores de saída digital do CLP. Para os interruptores utilizou-se os endereços de entrada do CLP, para as luzes os endereços de saída e para a posição da cama *Holding Register*, registrador passível de leitura e escrita, para armazenar a posição da cama.

Foi utilizado a ferramenta *MasterTool* para configuração e programação do código do controlador. Nela foi criado um código simples que apenas lia os valores de entrada nos interruptores do CLP e as armazenavam no registrador respectivo e inseriam nas variáveis de saída valores lidos de outros registradores. Além disso, a porta de comunicação serial RS232 do controlador (COM1) foi configurada como Modbus escravo, como pode ser vista na Figura 10.

Figura 10: Configuração da porta de comunicação serial do CLP



Fonte: Elaborada pela autora utilizando o *software MasterTool*

No *BluePlant* foi criado um novo canal de comunicação configurado como Modbus Serial mestre. A partir desse canal foi criado um nó e deste pontos de comunicação foram criados. Cada ponto atribuiu uma *tag* do supervisor a um endereço Modbus e configurada como leitura ou escrita. A Tabela indica a trajetória de cada *tag* do *BluePlant* para variável do *MasterTool*, passando pelo protocolo. Nela já está incluída a variável de posição da cama, um número de 0 a 3 que o código no CLP trata com a lógica de movimentação criada.

3.4.3 Aplicativo e banco de dados

Para a comunicação entre o aplicativo e o banco de dados foi utilizada a interface de programação JDBC (*Java Database Connectivity*, em inglês). Essa interface utiliza *drivers* já prontos para realizar a conexão entre os sistemas, sendo necessário apenas adicionar o pacote ao projeto e utilizar suas classes de forma adequada. Para tanto, foi criada uma *thread* para atualizar os dados do aplicativo de acordo com o banco de forma a ser executada em paralelo com a aplicação principal. Assim, o aplicativo não desperdiça tempo de leitura dos botões para a cada ciclo, definido como 1 segundo, ler as variáveis do banco de dados e atualizá-las no aplicativo.

Outro ponto é que o paciente ao ser descadastrado do quarto, precisa ler novamente o código QR para verificar se possui direito ao controle das variáveis, pois o aplicativo

Tabela 3: Equivalência entre variáveis e pontos de comunicação

<i>Tag no BluePlant</i>	Endereço Modbus	Endereço no CLP	Variável no CLP
Quarto(0).lamp1	000017	%QX1.0	LUZ1_1BP
Quarto(0).lamp2	000018	%QX1.1	LUZ1_2BP
Quarto(1).lamp1	000019	%QX1.2	LUZ2_1BP
Quarto(1).lamp2	000020	%QX1.3	LUZ2_2BP
Quarto(2).lamp1	000021	%QX1.4	LUZ3_1BP
Quarto(2).lamp2	000022	%QX1.5	LUZ3_2BP
Quarto(3).lamp1	000023	%QX1.6	LUZ4_1BP
Quarto(3).lamp2	000024	%QX1.7	LUZ4_2BP
Quarto(0).switchanalog1	100001	%IX0.0	INT1_1BP
Quarto(0).switchanalog2	100002	%IX0.1	INT1_2BP
Quarto(1).switchanalog1	100003	%IX0.2	INT2_1BP
Quarto(1).switchanalog2	100004	%IX0.3	INT2_2BP
Quarto(2).switchanalog1	100005	%IX0.4	INT3_1BP
Quarto(2).switchanalog2	100006	%IX0.5	INT3_2BP
Quarto(3).switchanalog1	100007	%IX0.6	INT4_1BP
Quarto(3).switchanalog2	100008	%IX0.7	INT4_2BP
posicao(3)	400001	%MW0	CAMABP

Fonte: Elaborada pela autora

retorna à tela inicial. Além disso, de forma semelhante ao SCADA, ao ser alterada uma variável o programa envia uma *query* para atualizar o banco de dados. Estima-se, portanto, que o máximo de tempo entre ser mudado um parâmetro no aplicativo e o SCADA perceber essa alteração é de 1 segundo, se desconsiderar o tempo para ativar a conexão.

4 RESULTADOS

Na parte digital, o transporte do sinal do aplicativo até o supervisório foi relativamente bem sucedido. Houve algumas falhas inicialmente, porém corrigidas com certa facilidade. Um erro frequente, por exemplo, foi esquecer de desativar o *firewall* do computador onde o banco de dados está alojado, fazendo com que a conexão entre o aplicativo e o banco de dados fosse bloqueada.

Já o projeto a partir do *BluePlant* foi mais complicado. Ainda na comunicação com o CLP – inicialmente considerada a parte mais tranquila do trabalho, já que o supervisório foi feito para se comunicar com o controlador, além de serem da mesma empresa – houve problemas na configuração do canal que tomaram muito mais tempo que o previsto para serem solucionados. A dificuldade maior foi ter que encontrar a fonte do erro sem um indicador claro.

Enquanto os problemas de comunicação não foram solucionados, não houve testes com a cama, por receio de quebrá-la. Ao resolvê-los, notou-se algumas outras falhas na lógica do supervisório e foram corrigidas na hora. Na hora de testar com a cama, observou-se que as saídas de tensão inicialmente consideradas para acionarem os motores e alimentarem os sensores compartilham o mesmo endereço no CLP que as saídas luminosas. Com isso, não há saídas suficientes (8 luzes, 2 sensores e 2 pares de alimentação para 2 motores) para todo o projeto. Optou-se por dividir a demonstração do projeto em duas separadas: a das luzes e a da cama.

4.1 Teste das luzes

O controlador utilizado (Altus Duo 351) com interface homem-máquina integrada (maleta de treinamento) possui 8 entradas digitais com visualização de estado e possibilidade de acionamento manual e 8 saídas digitais também com visualizador de estado, ou seja, quando ligada uma luz é acesa. Essas E/S foram utilizadas para simulação dos interruptores e luzes do quarto do paciente.

O primeiro teste realizado demonstrou sucesso; não é relevante a origem do acionamento do interruptor – se do aplicativo de celular, direto do supervisório ou do controlador – a luz alternava entre acesa e apagada sempre que o interruptor se alterasse. Esse mecanismo é similar ao conhecido como “chave hotel”, onde há dois interruptores acionando uma lâmpada de forma dependente apenas do estado atual da mesma.

É importante ressaltar um possível problema originado da lógica utilizada. O algoritmo responsável por determinar o valor da lâmpada a partir das entradas dos valores dos interruptores digitais e reais está localizado no sistema supervisório, sendo as entradas e saídas do CLP acessadas diretamente. Caso ocorra uma falha de comunicação entre o CLP e o SCADA nem mesmo o interruptor real acionará a lâmpada. Uma alteração nos

códigos deve ser feita de forma a migrar a lógica utilizada para controle das luzes em função dos interruptores digitais e físicos do supervisor para o CLP, de forma a perder apenas a possibilidade de acionamento via digital, mas não a real caso o ambiente digital perca funcionalidade.

4.2 Teste da cama

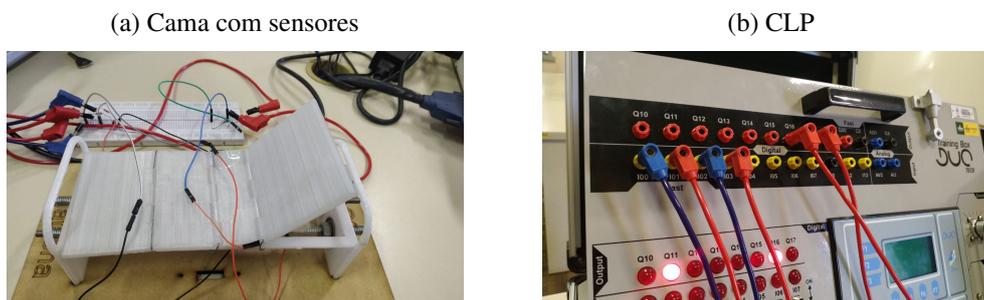
Para o teste com a cama, 4 das 8 entradas digitais utilizadas anteriormente foram reutilizadas de forma a obter o sinal do sensor de fim de curso por contato elétrico criado. Uma distinção, porém, é vista no acionamento das mesmas – enquanto nas luzes era feito manualmente (com um contato elétrico), agora é feito por meio de fios condutores.

Além disso, 6 das 8 saídas digitais foram reutilizadas, 4 para acionar os 2 motores necessários e 2 para alimentar os sensores de fim de curso. Analogamente às entradas, as saídas também foram manuseadas de forma diferente – fornecendo tensão elétrica em vez de apenas exibir visualmente o estado.

A partir do sucesso da simulação feita no próprio *software* do CLP (*MasterTool*), o sistema físico foi acoplado ao controlador e analisou-se o resultado. Logo no primeiro teste foi notado um erro estúpido do projeto; o dimensionamento dos motores foi negligenciado. Os mini motores CC de 9V empregados na estrutura não foram suficientes para vencer o torque mecânico presente e, devido às falhas anteriores, não havia mais tempo hábil para substituição da máquina.

Para não travar o trabalho, foi feita um novo teste incluindo um motor simulado. Luzes indicavam a tensão que deveria ser aplicada aos terminais dos motores e o eixo era girado manualmente de acordo com a instrução visual. Foi constatado que o sensor funciona, pois ao chegar nos extremos de cada guia a luz indicadora se apagava, simbolizando parada do motor. O código em linguagem ST implementado para teste da cama pode ser visto no Apêndice A.3. Na Figura 11 é possível visualizar a configuração do CLP conectado aos sensores da cama.

Figura 11: Teste da cama



Fonte: Elaborada pela autora

Os cabos com terminais do tipo “banana” são conectados em série a uma resistência e inseridos na *proto board*. De lá, os fios dos sensores são conectados a cama. Note que no arranjo mostrado na Figura 11b a luz Q11 acesa indica que o motor deve levar a chapa da cabeça da cama à posição horizontal. A saída Q16 está alimentando o sensor e ao fechar contato, a entrada I00 será acionada, apagando todas as luzes, indicando que não há mais alimentação no sistema físico.

4.3 Teste do quarto

De forma a avaliar o desempenho do algoritmo de forma geral, foi realizado um teste extra envolvendo o modelo da cama e um par de luzes e interruptores simulando um quarto completo. Há entradas e saídas suficientes para o teste já que o número de luzes e interruptores diminui em 75%. Como esperado, visto os testes anteriores, o sistema se comportou idealmente, com a ressalva dos motores.

5 CONCLUSÕES

Esse capítulo apresenta as considerações finais e trabalhos futuros possíveis propostos para o projeto.

5.1 Considerações finais

A conclusão geral do projeto é que a arquitetura de comunicações funciona. O trabalho concluiu de forma satisfatória o objetivo que foi posto como permitir que pacientes internados possam alterar funções do quarto por meio de um aplicativo desde que haja acesso à rede. Isto pois o projeto parte do pressuposto que o hospital tenha os outros recursos necessários para a realização da operação. Dessa forma, pacientes não necessitam de tanta assistência ou mobilidade para exercer tais funções.

O projeto demonstrou a aplicação em diversas configurações de entrada e saída que podem ser adaptadas a novas funções. A medição do tempo médio de resposta entre o comando na sua origem inicial e o seu resultado no destino não foi feito. Porém, seria possível de ser feito com sensibilidade considerada satisfatória sincronizando os horários do *smartphone* e do computador onde se instalou o banco de dados com o NTP (protocolo de tempo para redes).

Os testes realizados demonstraram que a aplicação na cama do leito com posições previamente definidas teve resultado aceitável, apesar da falha em se dimensionar os motores de acordo com o projeto. Embora tenha-se mostrado visualmente uma simulação do acionamento do motor, não foi suficiente para caracterizar êxito de projeto. Ainda assim, os sensores, que acreditava-se serem incapazes de cumprir sua função, teve bom resultado.

Embora tenha sido realizado a baixo custo pela aluna, nota-se que o sistema real terá um custo maior devido às licenças e equipamentos necessários. Além disso, o hospital potencialmente terá que reformar suas instalações elétricas de modo a comportar o sistema implementado.

Apesar do revés sofrido ao não conseguir entrar em contato com hospitais da região, nota-se que a aplicação agrega valor ao conforto do paciente, tendo atingido seu objetivo. O valor exato, no entanto, não foi possível obter, pois não há informações sobre tendência de reações dos pacientes ao novo produto ou necessidade do mesmo.

Houve uma satisfação maior por parte da aluna também pelo primeiro contato com banco de dados e a linguagem de programação SQL, cada vez mais utilizados no mercado de trabalho. O contato mais profundo com sistemas supervisórios também foi uma vantagem a ser destacada.

5.2 Trabalhos futuros

Como visto na seção anterior, há espaço para melhorias em várias áreas do projeto. Ainda na parte de protocolos recomenda-se a continuidade do estudo a respeito dos protocolos utilizáveis para encontrar um que se encaixe em todas as aplicações do hospital de forma que não interfira em equipamentos e que seja economicamente viável. Para melhorar o conforto do usuário pode-se implementar novas aplicações, como estabelecer comunicação com uma *SmartTV*.

Por um lado mais acadêmico, é possível focar mais no controle de posição da cama e necessário realizar o dimensionamento justo dos motores. A vantagem dessa abordagem se daria na maior precisão de posicionamento pelo usuário. Considera-se, todavia, que essa funcionalidade não garantiria aumento de nível de conforto relevante ao usuário em relação ao atual.

Outro ponto a ser aprimorado é a rede em que o banco de dados se encontra. No projeto foi utilizado o computador pessoal da aluna para aloca-lo e era necessário que o supervisor e o *smartphone* de origem do aplicativo estivessem conectados à mesma rede para acessar o seu IP. Sugere-se alocar um servidor para esse propósito. Alternativamente, outra solução proposta é a utilização de controles RFID. Assim, não é necessário garantir acesso à rede do banco de dados a um dispositivo estranho pois o controle já terá esse acesso. Fica a critério do hospital escolher a melhor opção de acordo com suas preferências.

Sugere-se também a arquitetura de sistemas adaptados a necessidades específicas de alguns pacientes. Por exemplo, um sistema de segurança para pacientes idosos ou com risco de queda. Soluções mais dedicadas a diferentes situações trazem maior satisfação ao usuário que se sente melhor atendido.

Por fim, para haver uma motivação concreta desse trabalho é necessário um estudo de campo com possíveis usuários em hospitais. Dessa forma, obtêm-se informações suficientes para atender de maneira mais eficiente aos problemas enfrentados e é possível alcançar uma melhor gestão da qualidade do projeto.

BIBLIOGRAFIA

ABNT. **NBR-7256: Tratamento de ar em estabelecimentos assistenciais de saúde (EAS) - Requisitos para projeto e execução das instalações.** Rio de Janeiro, 2005. p. 22.

AMBRÓSIO, P. E.; KANETO, C. M. *Biologia Computacional e Engenharia Biomédica.* In: TECNOLOGIAS, Técnicas e Tendências em Engenharia Biomédica. Bauru, SP: [s.n.], 2014. p. 114–129.

ANS. **Média de permanência geral.** Brasil, 2013. Disponível em: <http://www.ans.gov.br/images/stories/prestadores/E-EFI-05.pdf>. Acesso em: 18 mar. 2019.

BOLZANI, C. A. M. **Residências inteligentes: um curso de domótica.** São Paulo: Livraria da Física, 2004. p. 332. ISBN 85-88325-25-X.

BONELLA, A. E. *Saúde Pública, Utilidade e Equidade.* In: TECNOLOGIAS, Técnicas e Tendências em Engenharia Biomédica. Bauru, SP: [s.n.], 2014. p. 18–26.

BRASIL. Lei n 10.741, de 1 de outubro de 2003. **Estatuto do Idoso**, ago. 2003. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/2003/L10741.htm. Acesso em: 18 mar. 2019.

BRASIL. Lei n 11.108, de 7 de abril de 2005. **Altera a Lei no 8.080, de 19 de setembro de 1990, para garantir às parturientes o direito à presença de acompanhante durante o trabalho de parto, parto e pós-parto imediato, no âmbito do Sistema Único de Saúde - SUS.**, abr. 2005. Disponível em: http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2005/Lei/L11108.htm. Acesso em: 18 mar. 2019.

BRASIL. Lei n 13.146, de 6 de julho de 2015. **Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência)**, jul. 2015. Disponível em: http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Lei/L13146.htm. Acesso em: 18 mar. 2019.

BRASIL. Lei n 8.069, de 13 de julho de 1990. **Estatuto da Criança e do Adolescente**, jul. 1990. Disponível em: http://www.planalto.gov.br/ccivil_03/LEIS/L8069.htm. Acesso em: 18 mar. 2019.

CENTER_MEDICAL. **Cama Fawler Manual c/ Leito em Chapa - CM-010 GDLZ.** Portuguese. Disponível em: <https://www.centermedical.com.br/cama-fawler-manual-com-chapa-cm-010c-gdlz/p>. Acesso em: 08 mai. 2019.

COMER, D. E.; STEVENS, D. L. **Interligação em redes TCP/IP, volume 2.** Rio de Janeiro: Campus, 1999. p. 592. ISBN 85-352-0395-8.

DENSO_WAVE. **QR Code**. English. Disponível em: <https://www.qrcode.com/en/>. Acesso em: 18 mar. 2019.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. São Paulo: Pearson Addison Wesley, 2011. p. 788. ISBN 978-85-7936-085-5.

FLORENTINO, G. H. P. et al. Hospital Automation System RFID-Based: Technology Embedded In Smart Devices (Cards, Tags and Bracelets). **30th Annual International IEEE EMBS Conference**, ago. 2008.

FRANCHESI, A.; ANTONELLO, M. G. **Elementos de Máquinas**. Santa Maria: UFSM, 2014. p. 152.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. **Implementação de sistemas de banco de dados**. Rio de Janeiro: Campus, 2001. p. 697. ISBN 85-352-0749-X.

GROOVER, M. **Automação industrial e sistemas de manufatura**. São Paulo: Pearson Prentice Hall, 2011. p. 581. ISBN 978-85-7605-871-7.

HUITSING, P. et al. Attack taxonomies for the Modbus protocols. **International Journal of Critical Infrastructure Protection**, p. 37–44, dez. 2008. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187454820800005X>. Acesso em: 19 mar. 2019.

IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal**. PNAD Contínua, 2018. Disponível em: <https://www.ibge.gov.br/estatisticas-novoportal/sociais/trabalho/17270-pnad-continua.html?=&t=downloads>. Acesso em: 18 mar. 2019.

JUVINALL, R. C.; MARSHEK, K. M. **Fundamentos do projeto de componentes de máquinas**. Rio de Janeiro: LTC, 2016. p. 562. ISBN 978-85-216-3009-8.

KNX_ASSOCIATION. **Our mission**. English. Disponível em: <https://www.knx.org/knx-en/for-professionals/What-is-KNX/Our-mission/index.php>. Acesso em: 17 abr. 2019.

LIU, X. et al. Design of Special Vehicle Torsion Beam Test Terminal Based on ZigBee and LabVIEW. **2017 IEEE International Conference of Real-Time Computing and Robotics**, jul. 2017.

MERZ, H.; HANSEMANN, T.; HÜBNER, C. **Building automation: communication systems with EIB/KNX, LON, and BACnet**. Alemanha: Springer, 2009. p. 277. ISBN 978-3-540-88828-4.

MODBUS.ORG. **MODBUS APPLICATION PROTOCOL SPECIFICATION**. Massachusetts, EUA, V1.1b3. p. 50.

MTUNGWA, L. A.; PRETORIUS, J.; CHOWDHURY, S. Design of Energy Monitoring System for the Boiler and Chiller Plants in Netcare Private Hospital. **2018 IEEE PES/IAS PowerAfrica**, 2018.

NORTON, R. **Cinemática e dinâmica dos mecanismos**. Porto Alegre: AMGH Editora Ltda., 2010. p. 812. ISBN 978-85-63308-19-1.

ONWUBOLU, G. C. **Mechatronics: principles and applications**. Reino Unido: Elsevier, 2005. p. 645. ISBN 0-7506-6379-0.

PRUDENTE, F. **Automação predial e residencial: uma introdução**. Rio de Janeiro: LTC, 2011. p. 211. ISBN 978-85-216-0617-8.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. São Paulo: Pearson Makron Books, 1999. p. 778. ISBN 978-85-346-1073-5.

SILVA, C. W. **Mechatronics: an integrated approach**. EUA: CRC Press LLC, 2004. p. 1312. ISBN 0-8493-1274-4.

STREET, L. J. **Introduction to biomedical engineering technology**. EUA: CRC, 2012. p. 339. ISBN 978-1-4398-6058-8.

TANENBAUM, A. S.; WETHERALL, D. **Redes de computadores**. São Paulo: Pearson Prentice Hall, 2011. p. 561. ISBN 978-85-7605-924-0.

TEZA, V. R. **Alguns Aspectos sobre Automação Residencial - Domótica**. 2002. f. 108. Dissertação (Mestrado em ciência da computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis.

TORRES, G. G. **Tecnologia Assistiva para Detecção de Quedas: Desenvolvimento de Sensor Vestível Integrado ao Sistema Inteligente**. 2018. f. 71. Dissertação (Mestrado em engenharia elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre.

UNIÃO MUNDIAL. **Fecho Prisioneiro para Janela Guilhotina Ref 530 — Aço — 65mm**. Portuguese. Disponível em: <https://www.uniaomundial.com.br/fecho.php>. Acesso em: 08 mai. 2019.

VIANNA, W. S. **Sistema SCADA superviório**. Campos dos Goytacazes: IFF, 2008. p. 67.

VOLPATO, N. et al. **Prototipagem rápida**. São Paulo: Edgard Blücher, 2007. p. 244. ISBN 85-212-0388-8.

WU, W.; NIE, J.; PEI, P. Application of PLC in Hospital Emergency Calling Device. **2011 International Conference of Information Technology, Computer Engineering and Management Sciences**, 2011.

ZIGBEE_ALLIANCE. **What is zigbee**. English. Disponível em: <https://www.zigbee.org/what-is-zigbee/>. Acesso em: 08 jul. 2019.

APÊNDICE A CÓDIGOS UTILIZADOS NO TRABALHO

A.1 *Android Studio*

A.1.1 *Main Activity*

```

1 package com.example.tcc;
2 import android.app.Activity;
3 import android.content.Intent;
4 import android.support.annotation.Nullable;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.Toast;
11 import com.google.zxing.integration.android.IntentIntegrator;
12 import com.google.zxing.integration.android.IntentResult;
13 public class MainActivity extends AppCompatActivity {
14     Button btnscan, teste, mudarip;
15     EditText ipnumero;
16     int resultado=0;
17     String IP = "192.168.15.5"; //IP padrao
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         btnscan = (Button) findViewById(R.id.btnscan);
23         teste = (Button) findViewById(R.id.teste);
24         mudarip = (Button) findViewById(R.id.mudarip);
25         ipnumero = (EditText) findViewById(R.id.numeroIP);
26         resultado=0;
27         final Activity activity = this;
28         btnscan.setOnClickListener(new View.OnClickListener() {
29             @Override //botao de leitura de QR
30             public void onClick(View v){
31                 IntentIntegrator integrator = new IntentIntegrator(
32                     activity);
33                 integrator.setDesiredBarcodeFormats(IntentIntegrator.
34                     QR_CODE_TYPES);
35                 integrator.setPrompt("Câmera Scan");
36                 integrator.setCameraId(0);
37                 integrator.initiateScan();
38             }
39         });
40         mudarip.setOnClickListener(new View.OnClickListener() {

```

```

39         @Override //botao que permite alterar IP
40         public void onClick(View v) {
41             String teste = ipnumero.getText().toString();
42             if (teste.length()>6) IP = teste;
43             ipnumero.setText("");
44         }
45     });
46 }
47 @Override //compara resultado do QR com ID no banco de dados
48 protected void onActivityResult(int requestCode, int resultCode,
49 @Nullable Intent data) {
50     IntentResult result = IntentIntegrator.parseActivityResult(
51 requestCode, resultCode, data);
52     if(result != null){
53         GetData myData = new GetData();
54         String leitura = result.getContents();
55         resultado = myData.getdata(leitura, IP);
56         if (resultado==1){
57             openActivityResources(leitura);
58         }else if (resultado==0){
59             alert("Usuário não cadastrado");
60         }else if (resultado==-1){
61             alert("Falha na conexão");
62         }else if (resultado==-2){
63             alert("Falha no login");
64         }else alert("Scan Cancelado");
65     }
66     else {
67         super.onActivityResult(requestCode, resultCode, data);
68     }
69 }
70 private void alert (String msg){ //exibe mensagem de alerta
71     Toast.makeText(getApplicationContext(),msg,Toast.LENGTHLONG).
72 show();
73 }
74 public void openActivityResources(String id){
75     Intent intent = new Intent(this, ResourcesActivity.class);
76     intent.putExtra("id", id); //passa id do paciente lido no QR
77     intent.putExtra("IP", IP); //passa IP como parametro
78     startActivity(intent); //abre atividade do quarto
79 }
80 }

```

A.1.2 Resources Activity

```

1 package com.example.tcc;
2 import android.support.v7.app.AppCompatActivity;
3 import android.os.Bundle;
4 import android.util.Log;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.CompoundButton;
8 import android.widget.Switch;
9 import android.widget.TextView;
10 import android.widget.Toast;
11 public class ResourcesActivity extends AppCompatActivity {
12     Switch switch1, switch2;
13     Button subircabeca, descercabeca, subirperna, descperna;
14     TextView quarto, nome, data;

```

```

15  boolean cabecaerguida , pernaerguida , inton1 , inton2 ;
16  String id_quarto ;
17  @Override
18  protected void onCreate(Bundle savedInstanceState) {
19      super.onCreate(savedInstanceState);
20      setContentView(R.layout.activity_resources);
21      Bundle b = getIntent().getExtras();
22      String id = b.getString("id"); //recupera id do paciente
23      final String IP = b.getString("IP"); //recupera ip
24      switch1 = (Switch) findViewById(R.id.switch1);
25      switch2 = (Switch) findViewById(R.id.switch2);
26      subircabeca = (Button) findViewById(R.id.subircabeca);
27      descercabeca = (Button) findViewById(R.id.descercabeca);
28      subirperna = (Button) findViewById(R.id.subirperna);
29      descerperna = (Button) findViewById(R.id.descerperna);
30      quarto = (TextView) findViewById(R.id.quarto);
31      nome = (TextView) findViewById(R.id.nome);
32      data = (TextView) findViewById(R.id.data);
33      //le banco de dados ao abrir tela para pegar id do quarto
34      DadosPaciente quarto_tab = new DadosPaciente();
35      id_quarto = quarto_tab.pegarDados("SELECT LT.ID FROM LEITO LT
JOIN PACIENTE PC ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" +id ,
"ID" , IP);
36      quarto.setText("Quarto " +id_quarto);
37      subircabeca.setOnClickListener(new View.OnClickListener() {
38          @Override //botao para subir a chapa da cabeca
39          public void onClick(View v){
40              if (!cabecaerguida){
41                  if (pernaerguida) atualizaTabela("UPDATE LEITO SET
CAMA=2 WHERE ID =" +id_quarto , IP);
42                  else atualizaTabela("UPDATE LEITO SET CAMA=1 WHERE
ID =" +id_quarto , IP);
43                  cabecaerguida=true ;
44              }
45              else{
46                  alert("Cabeça já está no máximo");
47              }
48          }
49      });
50      descercabeca.setOnClickListener(new View.OnClickListener() {
51          @Override //botao para descer a chapa da cabeca
52          public void onClick(View v){
53              if (cabecaerguida){
54                  if (pernaerguida) atualizaTabela("UPDATE LEITO SET
CAMA=3 WHERE ID =" +id_quarto , IP);
55                  else atualizaTabela("UPDATE LEITO SET CAMA=0 WHERE
ID =" +id_quarto , IP);
56                  cabecaerguida=false ;
57              }
58              else{
59                  alert("Cabeça já está no mínimo");
60              }
61          }
62      });
63      subirperna.setOnClickListener(new View.OnClickListener() {
64          @Override //botao para subir chapa da perna
65          public void onClick(View v){
66              if (!pernaerguida){

```

```

67         if (cabecaerguida) atualizaTabela("UPDATE LEITO SET
CAM A=2 WHERE ID =" + id_quarto , IP);
68         else atualizaTabela("UPDATE LEITO SET CAM A=0 WHERE
ID =3"+id_quarto , IP);
69         pernaerguida=true;
70     }
71     else{
72         alert("Pernas já estão no máximo");
73     }
74 }
75 });
76 descerperna.setOnClickListener(new View.OnClickListener() {
77     @Override //botao para descer chapa da perna
78     public void onClick(View v){
79         if (pernaerguida){
80             if (cabecaerguida) atualizaTabela("UPDATE LEITO SET
CAM A=1 WHERE ID =" + id_quarto , IP);
81             else atualizaTabela("UPDATE LEITO SET CAM A=0 WHERE
ID =" + id_quarto , IP);
82             pernaerguida=false;
83         }
84         else{
85             alert("Pernas já estão no mínimo");
86         }
87     }
88 });
89 switch1.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {
90     @Override //botao switch1
91     public void onCheckedChanged(CompoundButton buttonView ,
boolean isChecked) {
92         if (inton1){
93             atualizaTabela("UPDATE LEITO SET INTERRUPTOR1=0
WHERE ID =" + id_quarto , IP);
94             inton1=false;
95         }
96         else {
97             atualizaTabela("UPDATE LEITO SET INTERRUPTOR1=1
WHERE ID =" + id_quarto , IP);
98             inton1=true;
99         }
100     }
101 });
102 switch2.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {
103     @Override //botao switch2
104     public void onCheckedChanged(CompoundButton buttonView ,
boolean isChecked) {
105         if (inton2){
106             atualizaTabela("UPDATE LEITO SET INTERRUPTOR2=0
WHERE ID =" + id_quarto , IP);
107             inton2=false;
108         }
109         else {
110             atualizaTabela("UPDATE LEITO SET INTERRUPTOR2=1
WHERE ID =" + id_quarto , IP);
111             inton2=true;
112         }

```

```

113     }
114     });
115     AtualizaDados atualizaDados = new AtualizaDados(id, IP);
116     new Thread(atualizaDados).start();
117 }
118 //atualiza tabela sempre que valor eh alterado
119 void atualizaTabela(String query, String IP){
120     EnviaDados tabela = new EnviaDados();
121     tabela.enviaDdos(query, IP);
122 }
123 //thread de leitura de dados
124 class AtualizaDados implements Runnable{
125     String id;
126     String IP;
127     String query;
128     String label;
129     String resultado;
130     DadosPaciente dados = new DadosPaciente();
131     AtualizaDados(String id, String IP){
132         this.id=id;
133         this.IP=IP;
134     }
135     @Override
136     public void run() {
137         for (;;) {
138             query="SELECT LT.PACIENTE_ID FROM LEITO LT RIGHT JOIN
139             PACIENTE PC ON PC.ID = LT.PACIENTE_ID WHERE PC.ID =" + id;
140             label="PACIENTE_ID";
141             resultado=dados.pegaDados(query, label, IP);
142             if (resultado.equals("")){
143                 runOnUiThread(new Runnable() {
144                     @Override
145                     public void run() {
146                         alert("Usuário não encontrado");
147                         finish();
148                     }
149                 });
150                 break;
151             }
152             else if (resultado.equals("Sem conexão")){
153                 runOnUiThread(new Runnable() {
154                     @Override
155                     public void run() {
156                         alert(resultado);
157                         finish();
158                     }
159                 });
160                 break;
161             }
162             else if (!resultado.equals(id)){
163                 runOnUiThread(new Runnable() {
164                     @Override
165                     public void run() {
166                         alert("Usuário não encontrado");
167                         finish();
168                     }
169                 });
170                 break;

```

```

170     }
171     query="SELECT NOME FROM LEITO LT LEFT JOIN PACIENTE PC
ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" + id;
172     label="NOME";
173     resultado=dados.pegDados(query, label, IP);
174     if (resultado.equals("")){
175         runOnUiThread(new Runnable() {
176             @Override
177             public void run() {
178                 alert("Usuário não encontrado");
179                 finish();
180             }
181         });
182         break;
183     }
184     else if (resultado.equals("Sem conexão")){
185         runOnUiThread(new Runnable() {
186             @Override
187             public void run() {
188                 alert(resultado);
189                 finish();
190             }
191         });
192         break;
193     }
194     else runOnUiThread(new Runnable() {
195         @Override
196         public void run() {
197             nome.setText(resultado);
198         }
199     });
200     query="SELECT NASCIMENTO FROM LEITO LT LEFT JOIN
PACIENTE PC ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" + id;
201     label="NASCIMENTO";
202     resultado=dados.pegDados(query, label, IP);
203     if (resultado.equals("")){
204         runOnUiThread(new Runnable() {
205             @Override
206             public void run() {
207                 alert("Usuário não encontrado");
208                 finish();
209             }
210         });
211         break;
212     }
213     else if (resultado.equals("Sem conexão")){
214         runOnUiThread(new Runnable() {
215             @Override
216             public void run() {
217                 alert(resultado);
218                 finish();
219             }
220         });
221     }
222     else runOnUiThread(new Runnable() {
223         @Override
224         public void run() {
225             if(resultado.length() > 10) data.setText(

```

```

226 resultado.substring(0,10));
227     }
228     });
229     query="SELECT CAMA FROM LEITO LT LEFT JOIN PACIENTE PC
230 ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" +id;
231     label="CAMA";
232     resultado=dados.pegDados(query, label, IP);
233     if (resultado.equals("")){
234         runOnUiThread(new Runnable() {
235             @Override
236             public void run() {
237                 alert("Usuário não encontrado");
238                 finish();
239             }
240         });
241     }
242     else if (resultado.equals("Sem conexão")){
243         runOnUiThread(new Runnable() {
244             @Override
245             public void run() {
246                 alert(resultado);
247                 finish();
248             }
249         });
250     }
251     else if (resultado.equals("0")){
252         cabecaerguida=false;
253         pernaerguida=false;
254     }
255     else if (resultado.equals("1")){
256         cabecaerguida=true;
257         pernaerguida=false;
258     }
259     else if (resultado.equals("2")){
260         cabecaerguida=true;
261         pernaerguida=true;
262     }
263     else if (resultado.equals("3")){
264         cabecaerguida=false;
265         pernaerguida=true;
266     }
267     query="SELECT INTERRUPTOR1 FROM LEITO LT LEFT JOIN
268 PACIENTE PC ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" +id;
269     label="INTERRUPTOR1";
270     resultado=dados.pegDados(query, label, IP);
271     if (resultado.equals("")){
272         runOnUiThread(new Runnable() {
273             @Override
274             public void run() {
275                 alert("Usuário não encontrado");
276                 finish();
277             }
278         });
279     }
280     else if (resultado.equals("Sem conexão")){
281         runOnUiThread(new Runnable() {
282             @Override
283             public void run() {

```

```

281         alert(resultado);
282         finish();
283     }
284 });
285     break;
286 }
287     else if (resultado.equals("0")) {
288         inton1=false;
289     }
290     else if (resultado.equals("1")){
291         inton1=true;
292     }
293     query="SELECT INTERRUPTOR2 FROM LEITO LT LEFT JOIN
PACIENTE PC ON PC.ID = LT.PACIENTE_ID WHERE PACIENTE_ID =" +id;
294     label="INTERRUPTOR2";
295     resultado=dados.pegaDados(query, label, IP);
296     if (resultado.equals("")){
297         runOnUiThread(new Runnable() {
298             @Override
299             public void run() {
300                 alert("Usuário não encontrado");
301                 finish();
302             }
303         });
304         break;
305     }
306     else if (resultado.equals("Sem conexão")){
307         runOnUiThread(new Runnable() {
308             @Override
309             public void run() {
310                 alert(resultado);
311                 finish();
312             }
313         });
314         break;
315     }
316     else if (resultado.equals("0")) {
317         inton2=false;
318     }
319     else if (resultado.equals("1")){
320         inton2=true;
321     }
322     try {
323         Thread.sleep(1000);
324     } catch (InterruptedException e) {
325         Log.e("Error na Thread",e.getMessage());
326     }
327 }
328 }
329 }
330 private void alert (String msg){
331     Toast.makeText(getApplicationContext(),msg,Toast.LENGTHLONG).
show();
332 }
333 }

```

A.1.3 Classe *GetData*

```

1 package com.example.tcc;

```

```

2 import android.util.Log;
3 import java.sql.Connection;
4 import java.sql.ResultSet;
5 import java.sql.Statement;
6 public class GetData {
7     Connection connect;
8     int login=0;
9     public int getdata(String qr, String IP) {
10        login=0;
11        connect=null;
12        try {
13            ConnectionHelper connectionHelper = new ConnectionHelper();
14            connect = connectionHelper.connections(IP);
15            if (connect == null) {
16                login = -1;
17            } else {
18                String query = "SELECT * FROM LEITO LT LEFT JOIN
19                PACIENTE PC ON PC.ID = LT.PACIENTE_ID";
20                Statement stmt = connect.createStatement();
21                //envia query
22                ResultSet rs = stmt.executeQuery(query);
23                while (rs.next()) {
24                    String tab = "" + rs.getInt("PACIENTE_ID");
25                    if (tab.equals(qr)) {
26                        login = 1;
27                    }
28                }
29                connect.close(); //encerra nonexao
30            } catch (Exception ex) {
31                Log.e("Error from ", ex.getMessage());
32                login=-2;
33            }
34            return login; //return codigo de estado
35        }
36    }

```

A.1.4 Classe *DadosPaciente*

```

1 package com.example.tcc;
2 import android.util.Log;
3 import java.sql.Connection;
4 import java.sql.ResultSet;
5 import java.sql.Statement;
6 public class DadosPaciente {
7     Connection connect;
8     String tab = "";
9     public String pegaDados(String query, String label, String IP) {
10        connect=null;
11        try {
12            ConnectionHelper connectionHelper = new ConnectionHelper();
13            connect = connectionHelper.connections(IP);
14            if (connect == null) {
15                return "Sem conexão";
16            } else {
17                Statement stmt = connect.createStatement();
18
19                //envia query
20                ResultSet rs = stmt.executeQuery(query);

```

```

21         while (rs.next()) {
22             tab = "" + rs.getString(label);
23         }
24     }
25     connect.close(); //encerra conexao
26 }catch(Exception ex){
27     Log.e("Error from dados", ex.getMessage());
28 }
29 return tab; //retorna dados
30 }
31 }

```

A.1.5 Classe *EnviaDados*

```

1 package com.example.tcc;
2 import android.util.Log;
3 import java.sql.Connection;
4 import java.sql.Statement;
5 public class EnviaDados {
6     Connection connect;
7     String tab = "";
8     public void enviaDdos(String query, String IP) {
9         connect=null;
10        try {
11            ConnectionHelper connectionHelper = new ConnectionHelper();
12            connect = connectionHelper.connections(IP);
13            if (connect == null) {
14                return;
15            } else {
16                Statement stmt = connect.createStatement();
17                //envia query
18                stmt.executeQuery(query);
19            }
20            connect.close(); //encerra conexao
21        }catch(Exception ex){
22            Log.e("Error from dados", ex.getMessage());
23        }
24    }
25 }

```

A.1.6 Classe *ConnectionHelper*

```

1 package com.example.tcc;
2 import android.annotation.SuppressLint;
3 import android.os.StrictMode;
4 import android.util.Log;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.SQLException;
8 public class ConnectionHelper {
9     String IP, DBUserName, DBPassword;
10    @SuppressLint("NewApi")
11    public Connection connections(String IP){
12        DBUserName = "tcc";
13        DBPassword = "tcc";
14        StrictMode.ThreadPolicy policy=new StrictMode.ThreadPolicy.
15        Builder().permitAll().build();
16        StrictMode.setThreadPolicy(policy);
17        java.sql.Connection connection = null;

```

```

17     String ConnectionURL=null;
18     try{ //cria conexao
19         Class.forName("oracle.jdbc.driver.OracleDriver");
20         ConnectionURL = "jdbc:oracle:thin:@"+IP+":1521:XE";
21         connection = DriverManager.getConnection(ConnectionURL ,
22         DBUserName, DBPassword);
23     }catch (SQLException se){
24         Log.e("error from SQL", se.getMessage());
25     }catch (ClassNotFoundException e){
26         Log.e("Error from class", e.getMessage());
27     }catch (Exception ex){
28         Log.e("Error from Exception", ex.getMessage());
29     }
30     return connection;
31 }

```

A.2 BluePlant

A.2.1 Tela principal

```

1 'codigos que implementam acoes ao clique com botao esquerdo do mouse
   quando tem mais de uma acao a ser executada (uma acao soh eh
   possivel ser feito diretamente na configuracao do objeto a ser
   clicado) em objetos presentes na tela principal
2 Public Sub MouseLeftButtonDown1(ByVal sender As Object, ByVal e As
   System.Windows.Input.MouseButtonEventArgs)
3     @Tag.indice=0 ' Seta tag indice para 0, para visualizar os parametros
   do quarto escolhido
4     @Display.Quarto.Open ' Abre quarto escolhido
5 End Sub
6 Public Sub MouseLeftButtonDown2(ByVal sender As Object, ByVal e As
   System.Windows.Input.MouseButtonEventArgs)
7     @Tag.indice=1 ' Seta tag indice para 1, para visualizar os parametros
   do quarto escolhido
8     @Display.Quarto.Open ' Abre quarto escolhido
9 End Sub
10 Public Sub MouseLeftButtonDown3(ByVal sender As Object, ByVal e As
   System.Windows.Input.MouseButtonEventArgs)
11     @Tag.indice=2 ' Seta tag indice para 2, para visualizar os parametros
   do quarto escolhido
12     @Display.Quarto.Open ' Abre quarto escolhido
13 End Sub
14 Public Sub MouseLeftButtonDown4(ByVal sender As Object, ByVal e As
   System.Windows.Input.MouseButtonEventArgs)
15     @Tag.indice=3 ' Seta tag indice para 3, para visualizar os parametros
   do quarto escolhido
16     @Display.Quarto.Open ' Abre quarto escolhido
17 End Sub

```

A.2.2 Tela do quarto

```

1 ' Funcao executada quando abre a tela
2 Public Sub DisplayOpening()
3     @Tag.conf(@Tag.indice)=0 ' configura a tag de alteracao de posicao da
   cama desse quarto como 0
4     @Tag.temp(@Tag.indice)=@Tag.posicao(@Tag.indice) ' configura posicao
   temporaria da cama (utilizada na alteracao da posicao) do quarto
   como a mesma da posicao atual

```

```

5 End Sub
6 ' Funcao executada quando fecha a tela
7 Public Sub DisplayClosing()
8     @Tag.conf(@Tag.indice)=0 ' refaz comandos de quando abre a tela para
9     cancelar a alteracao caso o usuario saia da tela no meio da mudanca
10    @Tag.temp(@Tag.indice)=@Tag.posicao(@Tag.indice)
11 End Sub
12 ' logica para alteracao da posicao da cama do quarto, 8 objetos foram
13    criados: 2 para cada uma das 4 posicoes possiveis da cama
14 ' ao se clicar num objeto estatico ele muda a tag de estado conf para
15    verdadeira, entrando em modo de ateracao
16 Public Sub MouseLeftButtonDown1(ByVal sender As Object, ByVal e As
17    System.Windows.Input.MouseButtonEventArgs)
18    @Tag.conf(@Tag.indice) = 1
19 End Sub
20 ' Ao se clicar num dos 4 objetos temporarios, a visibilidade entre eles
21    se alterna
22 Public Sub MouseLeftButtonDown2(ByVal sender As Object, ByVal e As
23    System.Windows.Input.MouseButtonEventArgs)
24    @Tag.temp(@Tag.indice)=1 ' clicar no objeto 0 faz o 0 sumir e o 1
25    aparecer
26 End Sub
27 Public Sub MouseLeftButtonDown3(ByVal sender As Object, ByVal e As
28    System.Windows.Input.MouseButtonEventArgs)
29    @Tag.temp(@Tag.indice)=2 ' clicar no objeto 0 1 faz o 1 sumir e o 2
30    aparecer
31 End Sub
32 Public Sub MouseLeftButtonDown4(ByVal sender As Object, ByVal e As
33    System.Windows.Input.MouseButtonEventArgs)
34    @Tag.temp(@Tag.indice)=3 ' clicar no objeto 0 2 faz o 2 sumir e o 3
35    aparecer
36 End Sub
37 Public Sub MouseLeftButtonDown5(ByVal sender As Object, ByVal e As
38    System.Windows.Input.MouseButtonEventArgs)
39    @Tag.temp(@Tag.indice)=0 ' clicar no objeto 0 3 faz o 3 sumir e o 0
40    aparecer
41 End Sub
42 ' Ao se clicar no botao confirmar, sai do estado de alteracao e
43    confirma posicao escolhida enviando a informacao para o banco de
44    dados alterando a variavel triggertask
45 Public Sub MouseLeftButtonDown6(ByVal sender As Object, ByVal e As
46    System.Windows.Input.MouseButtonEventArgs)
47    ' Add your code here
48    @Tag.conf(@Tag.indice)=0
49    @Tag.posicao(@Tag.indice)=@Tag.temp(@Tag.indice)
50    @Tag.indice_enviar = @Tag.indice
51    @Tag.triggertask.ToggleValue
52 End Sub
53 ' Ao se clicar no botao cancelar sai do estado de alteracao e volta a
54    posicao temporaria para a posicao original
55 Public Sub MouseLeftButtonDown7(ByVal sender As Object, ByVal e As
56    System.Windows.Input.MouseButtonEventArgs)
57    ' Add your code here
58    @Tag.conf(@Tag.indice)=0
59    @Tag.temp(@Tag.indice)=@Tag.posicao(@Tag.indice)
60 End Sub
61 ' Ao se clicar no interruptor, muda o estado do interruptor e envia
62    para o banco de dados

```

```

44 Public Sub MouseLeftButtonDown8(ByVal sender As Object, ByVal e As
    System.Windows.Input.MouseButtonEventArgs)
45     ' Add your code here
46     @Tag.Quarto(@Tag.indice).switch1.ToggleValue
47     @Tag.indice_enviar = @Tag.indice
48     @Tag.triggertask.ToggleValue
49 End Sub
50 ' Ao se clicar no interruptor, muda o estado do interruptor e envia
    para o banco de dados indicando
51 Public Sub MouseLeftButtonDown9(ByVal sender As Object, ByVal e As
    System.Windows.Input.MouseButtonEventArgs)
52     ' Add your code here
53     @Tag.Quarto(@Tag.indice).switch2.ToggleValue
54     @Tag.indice_enviar = @Tag.indice
55     @Tag.triggertask.ToggleValue
56 End Sub

```

A.2.3 Header

```

1 ' o header esta presente com qualquer tela aberta, entao sempre que o
    sistema esta rodando essa funcao eh executada
2 Public Sub DisplayIsOpen()
3     ' procura alteracao no interruptor vinda do interruptor fisico do clp
    para cada quarto
4     For i as Integer = 0 to 3
5         Dim trigar as Integer = 0
6         if @Tag.Quarto(i).switchanalog1 <> @Tag.Quarto(i).switchanalogant1
            Then
7             @Tag.Quarto(i).switchanalogant1=@Tag.Quarto(i).switchanalog1
8             @Tag.Quarto(i).switch1.ToggleValue
9             trigar = 1
10        End if
11        if @Tag.Quarto(i).switchanalog2 <> @Tag.Quarto(i).switchanalogant2
            Then
12            @Tag.Quarto(i).switchanalogant2=@Tag.Quarto(i).switchanalog2
13            @Tag.Quarto(i).switch2.ToggleValue
14            trigar = 1
15        End if
16        if @Tag.Quarto(i).lamp1 <> @Tag.Quarto(i).switch1 Then
17            @Tag.Quarto(i).lamp1=@Tag.Quarto(i).switch1
18            trigar = 1
19        End if
20        if @Tag.Quarto(i).lamp2 <> @Tag.Quarto(i).switch2 Then
21            @Tag.Quarto(i).lamp2=@Tag.Quarto(i).switch2
22            trigar = 1
23        End if
24        ' ao achar envia para o banco de dados o novo valor da tag switch
25        if trigar <> 0 Then
26            @Tag.indice_enviar = i
27            @Tag.triggertask.ToggleValue
28        End if
29    Next
30 End Sub

```

A.2.4 Script de envio para banco de dados

```

1 ' Exibe mensagem avisando que esta enviando dados
2 @Info.Trace("Iniciando banco - enviar")
3 ' Envia query para atualizar paramentros da tabela leito

```

```

4 @Dataset.Query.UpdateQuery.SqlStatement = "UPDATE LEITO " & _
5   " Set INTERRUPTOR1 = " & @Tag.Quarto(@Tag.indice_enviar).switch1 & _
6   " , INTERRUPTOR2 = " & @Tag.Quarto(@Tag.indice_enviar).switch2 & _
7   " , LUZ1 = " & @Tag.Quarto(@Tag.indice_enviar).lamp1 & _
8   " , LUZ2 = " & @Tag.Quarto(@Tag.indice_enviar).lamp2 & _
9   " , CAMA = " & @Tag.posicao(@Tag.indice_enviar) & _
10  " WHERE ID = " & @Tag.indice_enviar+1
11 @Dataset.Query.UpdateQuery.SelectCommand

```

A.2.5 Script de leitura do banco de dados

```

1 ' Exibe mensagem avisando que esta acessando o banco de dados
2 @Info.Trace("Iniciando banco - receber")
3 'envia query para receber dados da tabela leito em conjunto com a
   paciente, setado pelo id que compartilham, se for nulo (funcao NVL)
   seta um outuro valor definido
4 @Dataset.Query.Queryda.SqlStatement = "SELECT LT.ID AS ID_Q " & _
5   " , LT.CAMA AS CAMA " & _
6   " , NVL(PC.NOME,'0') AS NOME " & _
7   " , NVL(PC.ID,0) AS ID_P " & _
8   " , NVL(PC.NASCIMENTO,sysdate) AS NASCIMENTO " & _
9   " , LT.INTERRUPTOR2 AS INTERRUPTOR2 " & _
10  " , LT.INTERRUPTOR1 AS INTERRUPTOR1 " & _
11  " FROM LEITO LT " & _
12  " LEFT JOIN paciente PC ON PC.ID = LT.PACIENTE_ID"
13 Dim Nomedatabela as DataTable = @Dataset.Query.Queryda.SelectCommand()
14 ' define as tags como o retorno do banco de dados
15 Try
16   For Each r as DataRow in Nomedatabela.Rows
17     @Tag.posicao(r("ID_Q")-1)=r("CAMA")
18     @Tag.Dados(r("ID_Q")-1).Nome=r("NOME")
19     @Tag.Dados(r("ID_Q")-1).Idade= (Date.Today.Subtract(r("NASCIMENTO")
20     ).Days)/365
21     @Tag.Dados(r("ID_Q")-1).ID_=r("ID_P")
22     @Tag.Quarto(r("ID_Q")-1).switch1=r("INTERRUPTOR1")
23     @Tag.Quarto(r("ID_Q")-1).switch2=r("INTERRUPTOR2")
24   Next
25 Catch ex as Exception
26   @Info.trace("erro em task.taskete! ex: " & ex.Message)
27 End Try

```

A.3 MasterTool Cama

```

1 IF CAMABP <> POSICAO AND MOVENDO
   = FALSE THEN
2   MOVENDO := TRUE;
3   CAMA := CAMABP;
4 END_IF
5 IF MOVENDO THEN
6   IF CAMA = 0 THEN
7     IF POSICAO = 1 THEN
8       MOTOR1_1 := FALSE;
9       MOTOR1_2 := TRUE;
10      SAIDA1 := TRUE;
11      IF SENSOR1 = TRUE THEN
12        POSICAO := 0;
13        MOTOR1_2 := FALSE;
14        SAIDA1 := FALSE;
15        MOVENDO := FALSE;
16      END_IF
17    ELSIF POSICAO = 2 THEN
18      MOTOR1_1 := FALSE;
19      MOTOR1_2 := TRUE;
20      SAIDA1 := TRUE;
21      IF SENSOR1 = TRUE THEN
22        POSICAO := 3;
23        MOTOR1_2 := FALSE;
24        SAIDA1 := FALSE;
25        MOVENDO := FALSE;
26      END_IF
27    ELSIF POSICAO = 3 THEN

```

```

28     MOTOR2.1 := FALSE;
29     MOTOR2.2 := TRUE;
30     SAIDA2 := TRUE;
31     IF SENSOR3 = TRUE THEN
32         POSICAO := 0;
33         MOTOR2.2 := FALSE;
34         SAIDA2 := FALSE;
35         MOVENDO := FALSE;
36     END_IF
37 END_IF
38 ELSIF CAMA = 1 THEN
39     IF POSICAO = 0 THEN
40         MOTOR1.1 := TRUE;
41         MOTOR1.2 := FALSE;
42         SAIDA1 := TRUE;
43         IF SENSOR2 = TRUE THEN
44             POSICAO := 1;
45             MOTOR1.1 := FALSE;
46             SAIDA1 := FALSE;
47             MOVENDO := FALSE;
48         END_IF
49     ELSIF POSICAO = 2 THEN
50         MOTOR2.1 := FALSE;
51         MOTOR2.2 := TRUE;
52         SAIDA2 := TRUE;
53         IF SENSOR3 = TRUE THEN
54             POSICAO := 1;
55             MOTOR2.2 := FALSE;
56             SAIDA2 := FALSE;
57             MOVENDO := FALSE;
58         END_IF
59     ELSIF POSICAO = 3 THEN
60         MOTOR2.1 := FALSE;
61         MOTOR2.2 := TRUE;
62         SAIDA2 := TRUE;
63         IF SENSOR3 = TRUE THEN
64             POSICAO := 0;
65             MOTOR2.2 := FALSE;
66             SAIDA2 := FALSE;
67             MOVENDO := FALSE;
68         END_IF
69     END_IF
70 ELSIF CAMA = 2 THEN
71     IF POSICAO = 0 THEN
72         MOTOR1.1 := TRUE;
73         MOTOR1.2 := FALSE;
74         SAIDA1 := TRUE;
75         IF SENSOR2 = TRUE THEN
76             POSICAO := 1;
77             MOTOR1.1 := FALSE;
78             SAIDA1 := FALSE;
79             MOVENDO := FALSE;
80         END_IF
81     ELSIF POSICAO = 1 THEN

```

```

82     MOTOR2.1 := TRUE;
83     MOTOR2.2 := FALSE;
84     SAIDA2 := TRUE;
85     IF SENSOR4 = TRUE THEN
86         POSICAO := 2;
87         MOTOR2.1 := FALSE;
88         SAIDA2 := FALSE;
89         MOVENDO := FALSE;
90     END_IF
91 ELSIF POSICAO = 3 THEN
92     MOTOR1.1 := TRUE;
93     MOTOR1.2 := FALSE;
94     SAIDA1 := TRUE;
95     IF SENSOR2 = TRUE THEN
96         POSICAO := 2;
97         MOTOR1.1 := FALSE;
98         SAIDA1 := FALSE;
99         MOVENDO := FALSE;
100    END_IF
101 END_IF
102 ELSIF CAMA = 3 THEN
103     IF POSICAO = 0 THEN
104         MOTOR2.1 := TRUE;
105         MOTOR2.2 := FALSE;
106         SAIDA2 := TRUE;
107         IF SENSOR4 = TRUE THEN
108             POSICAO := 3;
109             MOTOR2.1 := FALSE;
110             SAIDA2 := FALSE;
111             MOVENDO := FALSE;
112         END_IF
113     ELSIF POSICAO = 1 THEN
114         MOTOR2.1 := TRUE;
115         MOTOR2.2 := FALSE;
116         SAIDA2 := TRUE;
117         IF SENSOR4 = TRUE THEN
118             POSICAO := 2;
119             MOTOR2.1 := FALSE;
120             SAIDA2 := FALSE;
121             MOVENDO := FALSE;
122         END_IF
123     ELSIF POSICAO = 2 THEN
124         MOTOR1.1 := FALSE;
125         MOTOR1.2 := TRUE;
126         SAIDA1 := TRUE;
127         IF SENSOR1 = TRUE THEN
128             POSICAO := 3;
129             MOTOR1.2 := FALSE;
130             SAIDA1 := FALSE;
131             MOVENDO := FALSE;
132         END_IF
133     END_IF
134 END_IF
135 END_IF

```

APÊNDICE B PROJETO MECÂNICO

B.1 Modelagem 3D da cama hospitalar

Para o desenvolvimento do modelo 3D do protótipo da cama hospitalar, foi utilizada como referência o modelo CM-010 GDLZ da marca Fawler. Não foram encontradas especificações na página da fabricante, mas sim na loja (CENTER_MEDICAL, 2019). A cama utilizada como base pode ser vista na Figura 12. O *software* utilizado para o projeto foi o *SolidWorks*.

Figura 12: Modelo Fawler CM-010 GDLZ - Cama hospitalar utilizada como guia no projeto



Fonte: (CENTER_MEDICAL, 2019)

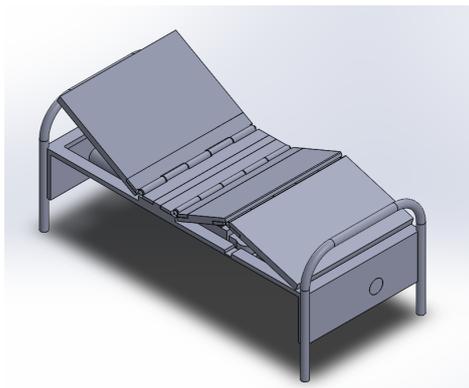
Considerando os limites dimensionais da impressora 3D do LIFEE (Laboratório de Inovação e Fabricação Digital da Escola de Engenharia) disponível para utilização do corpo acadêmico da UFRGS, a escala definida foi de 1:10. Essa escala gerou outros fatores limitantes, como emprego de eixos postos em direções opostas um ao outro para não haver colisão. Além disso, detalhes na cabeceira, calçadeira e braçadeiras foram ignorados a fim de canalizar os esforços para o objetivo do projeto.

Utilizando, então, o modelo de referência citado, se propôs dimensionar as 4 chapas em proporção de 5:2:3:4, sendo o mais da esquerda a chapa apoiadora da região superior do corpo do paciente. Utilizando as medidas da referência, obteve-se 70 mm, 28 mm, 42 mm e 56 mm, respectivamente, totalizando 196mm, 10% da medida do modelo original.

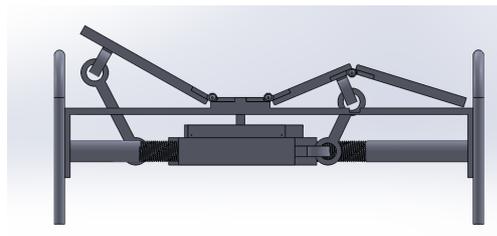
Para levantar as chapas, foi pensado um mecanismo combinado de fuso e castanha com quatro barras. Dessa forma, o motor giraria o fuso que moveria linearmente a castanha - restrita por uma guia para não rotacionar - e esta, por sua vez, moveria uma das quatro barras, ocasionando no movimento desejado da placa. O primeiro modelo criado pode ser visualizado na Figura 13. O mecanismo pode ser melhor entendido na Figura 13b.

Figura 13: Primeira versão do modelo 3D do protótipo da cama

(a) Vista em perspectiva



(b) Vista lateral



Fonte: Elaborada pela autora utilizando o *software SolidWorks*

Porém, a “castanha” criada foi impressa num teste e descobriu-se que a máquina não fornece resolução suficiente para confeccionar a rosca interna. Outro problema visto pelo técnico do laboratório foi o fato de que as peças deveriam ser todas impressas juntas, com o objeto final já montado. Isto demandaria um tempo muito grande de trabalho da máquina e qualquer parada no trabalho, como em decorrência de falha na energia elétrica, ocasionaria numa perda completa e recomeço.

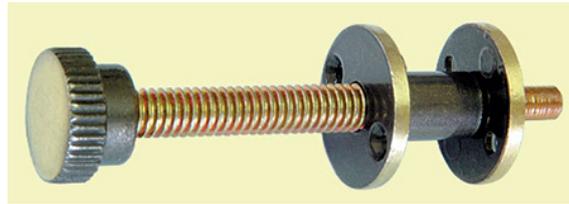
Em decorrência de alterações no projeto para melhorar os problemas descritos, um novo modelo foi feito. Dessa vez, optou-se por comprar o fuso e castanha separados e acoplá-los ao componente. Adiou-se então a impressão de forma a adequar o modelo às peças compradas separadamente, pois estas são mais restritas.

Ao procurar em uma dezena de lojas especializadas na cidade de Porto Alegre, a aluna falhou em encontrar fuso e castanha para aquisição. Mais, houve certo desconhecimento do que seriam essas peças por parte dos fornecedores. Porém, ao ir numa loja especializada em madeiras, encontrou-se uma peça chamada “fecho prisioneiro”, cuja visualização pode ser dada pela Figura 14. Um lado negativo dessa peça é a falta de opções de comprimento, sendo o encontrado incompatível com o projeto. Este foi contornado testando uma barra roscada no lugar do componente macho da peça. Verificando-se que os componentes casavam, o problema foi resolvido.

No fim, para o fuso e as castanhas, a aluna se deparou com as seguintes opções de aquisição:

- Pedido *online* de fuso com castanha;
- Barra roscada com componente fêmea do fecho prisioneiro em loja local;
- Usinagem de fuso e castanha no laboratório da UFRGS.

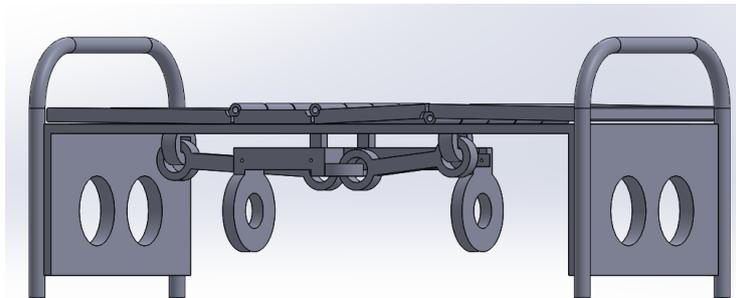
Figura 14: Fecho prisioneiro



Fonte: (UNIÃO_MUNDIAL, 2019)

Dados o preço, o prazo de entrega, a dificuldade técnica e a versatilidade das opções acima, optou-se pela segunda. Foi também adquirido um rolamento para acoplamento da barra na peça impressa. Dessa forma, alterou-se o projeto inicial para enquadrar as peças obtidas e o resultado pode ser visto na Figura 15.

Figura 15: Modelo da cama atualizado após alterações



Fonte: Elaborada pela autora utilizando o *software SolidWorks*

O mecanismo de quatro barras foi calculado de forma a apresentar a chapa deitada horizontalmente quando a castanha estivesse no extremo esquerdo, segundo referência da Figura 16a, inclinada a 60° e 50° , para cabeça e pernas, respectivamente, quando no extremo oposto. A Figura 16b evidencia as barras dispostas no projeto. Além disso, por ser movido por uma castanha, esse mecanismo possui a propriedade de uma das barras ter tamanho variável, descrito nas equações abaixo por dx . Foi definido que esse parâmetro seria igual a 0 no caso da chapa deitada horizontalmente. Observa-se que a Figura 16 apresenta uma configuração não atingível pelo modelo pois a distância entre a chapa e a guia é maior que o pino projetado. Os cálculos apresentados mais adiante são baseados na configuração real.

Por Pitágoras, tem-se que o comprimento da barra b_1 é dado pela equação

$$b_1^2 = b_{11}^2 + b_{12}^2 \quad (1)$$

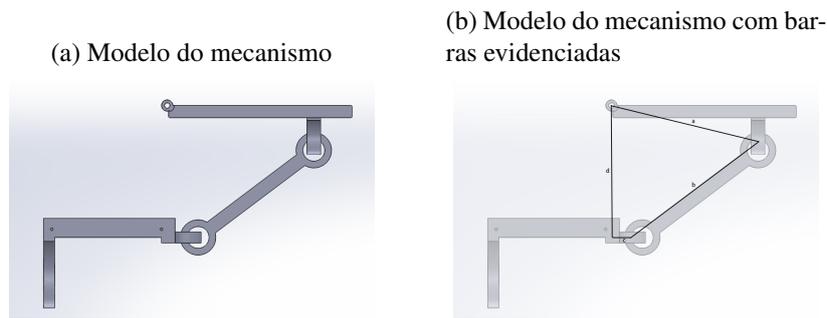
$$b_1 = \sqrt{54,5^2 + 11,5^2} = 55,7mm$$

para a chapa superior e

$$b_1 = \sqrt{34,75^2 + 11,5^2} = 36,6mm$$

para a chapa inferior. Portanto, por questões construtivas, já são possuídos os tamanhos de 3 das 4 barras mostradas na Figura 16a. Sabe-se também o ângulo entre a barra b_3 e b_4 , por serem perpendiculares, logo, sabe-se os parâmetros B_x e B_y . Definindo o ângulo

Figura 16: Mecanismo de quatro barras projetado



Fonte: Elaborada pela autora utilizando o *software SolidWorks*

da chapa apoiadora da cabeça do paciente como 0° , e aplicando-se as equações 2, 3 e 4, obtêm-se o comprimento necessário para o elo b_2 , para a parte superior e inferior da cama. Nota-se que se considera apenas o resultado positivo de b , por se tratar de uma medida de comprimento.

$$A_x = 55,7 \cos(88,1^\circ) = 1,85 \text{ mm}$$

$$A_y = 55,7 \sin(88,1^\circ) = 55,67 \text{ mm}$$

$$b = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2} = \sqrt{(19 - 1,85)^2 + (-5 - 55,67)^2} = 63 \text{ mm}$$

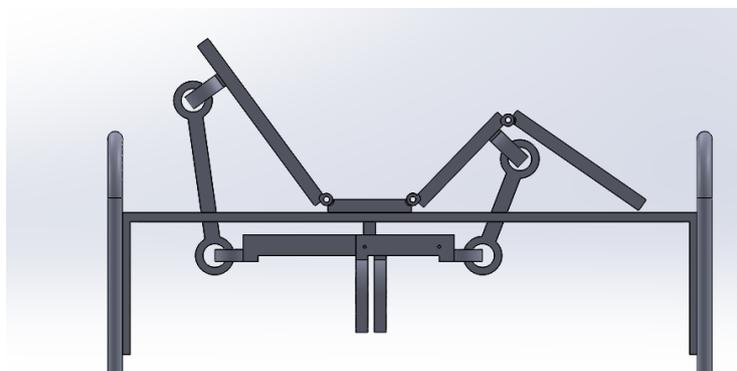
$$A_x = 36,6 \cos(71,7^\circ) = 11,5 \text{ mm}$$

$$A_y = 36,6 \sin(71,7^\circ) = 34,75 \text{ mm}$$

$$b = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2} = \sqrt{(19 - 11,5)^2 + (-5 - 34,75)^2} = 40,5 \text{ mm}$$

Para calcular o dx máximo, ou seja, o tamanho da guia projetada, foi acrescentado um ângulo de 60° para a chapa superior e 50° para a chapa inferior e calculado o novo comprimento da barra B_3 , sendo 5 (da construção anterior) + dx . A Figura 17 apresenta a nova configuração.

Figura 17: Configuração das chapas inclinadas aos ângulos mencionados



Fonte: Elaborada pela autora utilizando o *software SolidWorks*

Sabe-se que $B_x = d$ e $B_y = c$, por serem perpendiculares, então a Equação 5 resulta num $0 = 0$ dispensável. Utilizando novamente as equações 2, 3 e 4, obtêm-se

$$A_x = 55,7 \cos(138,1^\circ) = -41,4 \text{ mm}$$

$$A_y = 55,7 \sin(138,1^\circ) = 37,2 \text{ mm}$$

$$b = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2}$$

$$B_y = \pm \sqrt{b^2 - A_x^2 + 2A_x B_x - B_x^2} + A_y = 34,16 \text{ mm}$$

para a chapa superior e

$$A_x = 36,6 \cos(121,7^\circ) = -19,2 \text{ mm}$$

$$A_y = 36,6 \sin(121,7^\circ) = 31,1 \text{ mm}$$

$$B_y = \pm \sqrt{b^2 - A_x^2 + 2A_x B_x - B_x^2} + A_y = \{19,87 \text{ mm}, 41,33 \text{ mm}\}$$

para a chapa inferior. Os dois resultados apresentados para a chapa inferior refletem a geometria, ou seja, há dois possíveis dx para a guia que façam a chapa ficar no ângulo desejado. Optou-se pelo menor comprimento para facilitar o projeto, já que para chegar no extremo maior se passaria pelo menor.

Com os resultados das equações anteriores, soma-se 5 por ser o tamanho fixo da barra, porém no sentido oposto de dx. No fim, o projeto contém uma guia de 40 mm para o mecanismo que levanta a chapa apoiadora da cabeça do paciente e outra de 25 mm para o das pernas.

Ao entrar em contato com o LIFEE para efetivar a impressão das peças, porém, foi informado que o conselho responsável pelo laboratório decidiu em reunião não utilizar mais a impressora 3D no semestre atual por falta de manutenção. Uma alternativa sugerida pelo professor Renato Ventura foi utilizar a empresa PrintUp3D para executar a prototipagem e assim foi feito.

B.2 Resultados

Após ter sido realizada a impressão das peças, iniciou-se a fase de montagem. Os primeiros passos foram encaixar os elos nas chapas e guias e os rolamentos em seus respectivos furos. A Figura 18 exibe a etapa posterior à execução desses primeiros passos. Como o rolamento utilizado é radial, ou seja, permite forças axiais que o deslocam o eixo no sentido longitudinal, foi necessário utilizar porca para fixação do eixo linearmente.

Observou-se, porém, um erro de projeto em que a espessura do suporte das chapas da cama é muito fina e seu comprimento muito longo. Dessa forma, houve flexão do componente e posterior possível ruptura. Para contornar o problema, foi feita uma base em madeira, cortada a laser no LIFEE. A Figura 19a mostra a envergadura presente e na Figura 19b é possível visualizar o resultado da cama fixada à base referida. Nota-se na imagem furos feitos para encaixar um novo pedaço de madeira cortado de forma a criar mais apoios. Porém, não foi necessário visto que apenas os furos feitos para encaixar os pés da cama foram suficientes para correção do problema.

Retomando à montagem, previamente pausa para construção de medidas paliativas, foi utilizado arame para servir como eixo para as dobradiças entre as chapas. Além disso, foi elaborado um circuito para servir como "sensor de fim de curso" para as guias. Já na impressão, foram criados furos, vide Figura 16a, para inserção de fios na guia. A partir dessa delimitação o pino da guia foi coberto por papel alumínio de forma a conduzir

Figura 18: Peças da cama feitas pela impressora 3D com rolamentos acoplados



Fonte: Elaborada pela autora

Figura 19: Correção da envergadura da cama

(a) Envergadura



(b) Base de MDF



Fonte: Elaborada pela autora

Figura 20: Sensor de fim de curso paliativo e pino original

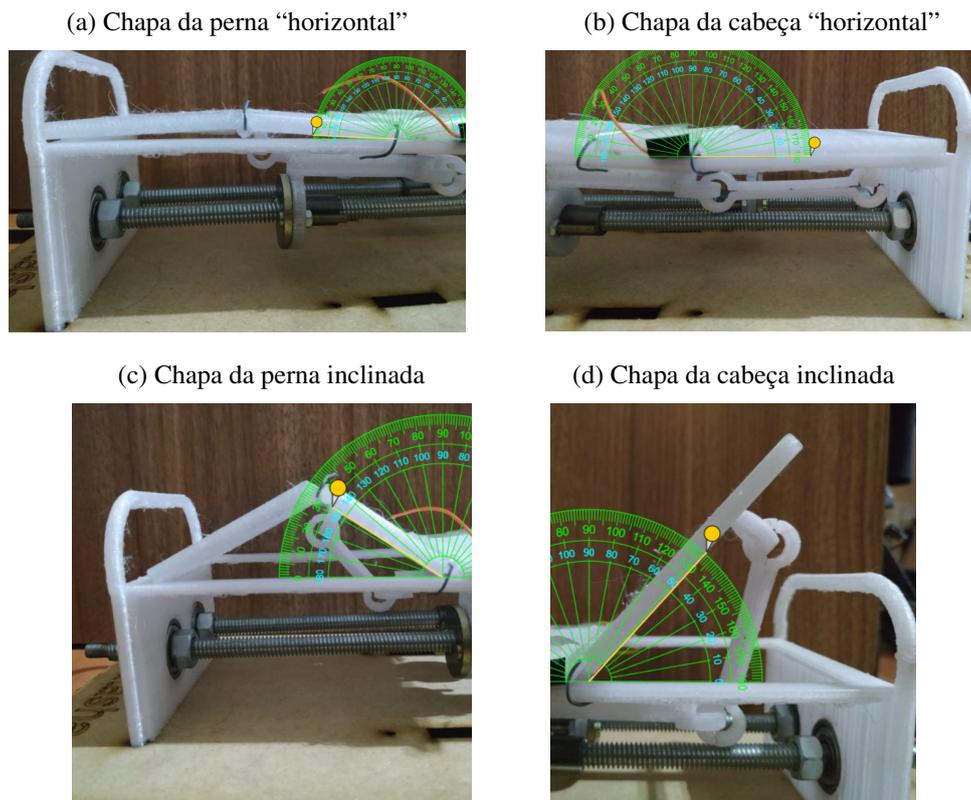


Fonte: Elaborada pela autora

eletricidade ao encostar no fio carregado. A Figura 20 mostra o recurso criado ao lado do pino original.

Em relação aos ângulos projetados, a Figura 21 mostra o resultado individualmente. Devido ao espaço ocupado pela dobradiça entre a chapa e o esqueleto, a Figura 21a exibe um ângulo de aproximadamente $2,5^\circ$. Esse desvio já era esperado, pois no projeto considerou-se menos prejudicial esse ângulo mínimo do que diminuir a espessura do esqueleto nesse ponto. A Figura 21b mostra que o ângulo da chapa da cabeça nessa posição coincide com o esperado, sendo de 0° .

Figura 21: Ângulos das chapas



Fonte: Elaborada pela autora utilizando ferramenta *Online Protractor*

Já o ângulo de inclinação das chapas quando a guia se encontra no extremo oposto ao anterior (projetado para 50° na perna e 60° na cabeça) teve um erro considerável. Como visto nas figuras 21c e 21d, os ângulos resultantes são de aproximadamente 33° e 48° , respectivamente. Esse resultado configura erros de 34% e 20% e mesmo atribuídos às folgas presentes na estrutura foram considerados excessivos.

ANEXO A REVISÃO BIBLIOGRÁFICA PARA MODELAGEM DA CAMA

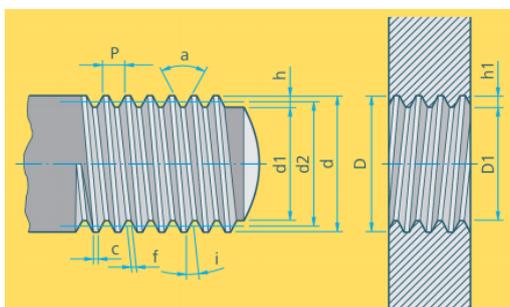
A.1 Elementos de máquinas

Parte do projeto mecânico do protótipo da cama hospitalar, os elementos de máquina são componentes mecânicos empregados em funções de fixação, apoio, transmissão, acoplamento, vedação, etc. Nessa seção será dada uma visão geral dos dois principais elementos, com suas funções e modo de funcionamento.

A.1.1 Parafusos e porcas

Fazendo parte da classe de elementos de fixação roscados, são usados para união de peças. Consistem em uma rosca helicoidal em torno de um cilindro (parafuso) ou de um furo cilíndrico (porca) para, ao ser girado, transformar movimento rotacional em linear e se unir à peça. (FRANCHESE; ANTONELLO, 2014) Os componentes de uma rosca podem ser visualizados na Figura 22, onde P é o passo, i o ângulo da hélice, d o diâmetro externo, c a crista, d_1 o diâmetro interno, D o diâmetro do fundo da porca, d_2 o diâmetro do flanco, D_1 o diâmetro do furo da porca, a o ângulo do filete, h_1 a altura do filete da porca, f o fundo do filete e h a altura do filete do parafuso.

Figura 22: Elementos constituintes de uma rosca



Fonte: (FRANCHESE; ANTONELLO, 2014)

Os tipos de perfis da rosca variam de acordo com as aplicações. Por exemplo, roscas triangulares são as mais comuns e utilizadas em parafusos e porcas. Já roscas trapezoidais são empregadas em instrumentos de comando de máquinas operatrizes e fusos. Há também roscas redondas, quadradas e dente-de-serra. (FRANCHESE; ANTONELLO, 2014)

A.1.2 Mancais de rolamento

Os mancais tem o propósito de fornecer um suporte de apoio a eixos girantes de máquinas. Os mancais mais simples possíveis são os planos sem lubrificação ou mancais deslizantes, como em tempos remotos, onde se acoplava a roda de madeira de carroças diretamente aos eixos também de madeira. Menores atritos e vidas mais longas foram obtidas pela adição de lubrificantes, sendo ainda utilizados hoje em algumas aplicações modernas. Nos mancais de rolamento, o eixo e os componentes mais externos são separados por esferas ou roletes. Portanto, substituem o atrito de deslizamento por atrito de rolamento, diminuindo assim o atrito de partida. (JUVINALL; MARSHEK, 2016) A Figura 23 mostra o rolamento utilizado nesse trabalho da Fersa.

Figura 23: Rolamento 608 ZZ/C3



Fonte: FERSA Bearings

Os rolamentos podem ser classificados de duas maneiras: (i) quanto às forças suportadas: radiais, axiais ou mistas, e (ii) quanto aos tipos de elementos rolantes: roletes ou esferas. (FRANCHESI; ANTONELLO, 2014) O rolamento utilizado nesse trabalho visto na Figura 23 é um rolamento radial de esferas.

A.2 Mecanismos

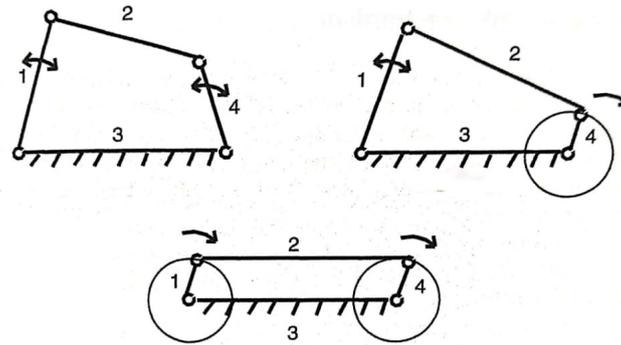
Segundo (NORTON, 2010), “um mecanismo é um dispositivo que transforma um movimento qualquer em um padrão desejado e geralmente desenvolve forças de baixa intensidade e transmite pouca potência”. Nessa seção serão estudados os dois mecanismos utilizados no projeto da cama hospitalar.

A.2.1 Mecanismo de quatro barras

Segundo (NORTON, 2010), simplicidade é uma das marcas de um bom projeto. Ou seja, a menor quantidade de peças que podem realizar um trabalho geralmente fornece a solução mais barata e confiável. O mecanismo de quatro barras é o mais simples para movimentos de um grau de liberdade. Ele consiste em 4 elos conectados por 4 juntas que permitem rotação. (ONWUBOLU, 2005; NORTON, 2010) É o dispositivo mais comum e usual em mecanismos, aparecendo também em outras variações, como biela-manivela e came seguidor. Também é extremamente versátil em termos de tipos de movimento que pode gerar. (NORTON, 2010) A Figura 24 mostra três exemplos de cadeias de quatro barras alterando o comprimento dos elos.

Para fazer a análise algébrica da posição do mecanismo, será utilizado o esquema visto na Figura 25. Os pontos O2, A, B e O4 representam as juntas e os ângulos θ_2 , θ_3 e θ_4 os

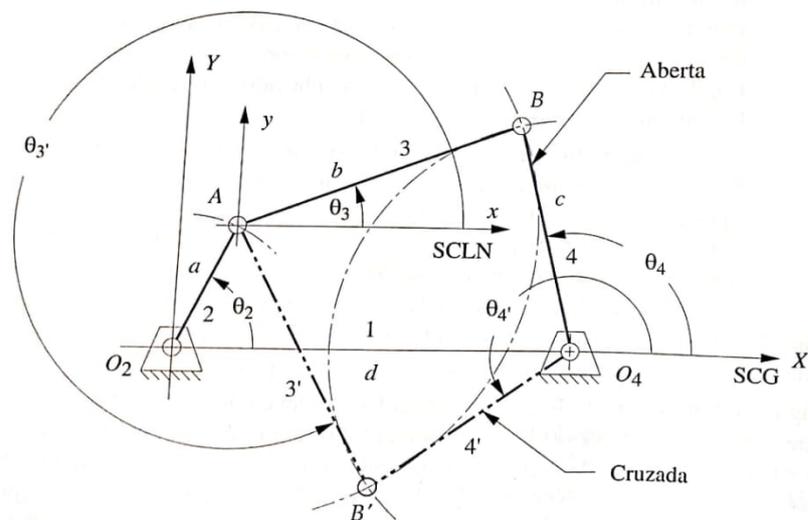
Figura 24: Mecanismos de quatro barras



Fonte: (ONWUBOLU, 2005)

ângulos formados entre os elos. Observa-se que há um elo imaginário disposto sobre o eixo x.

Figura 25: Gráfico auxiliar para equacionamento do mecanismo de quatro barras



Fonte: (NORTON, 2010)

As coordenadas do ponto A são obtidas de:

$$A_x = a \cos \theta_2 \quad (2)$$

$$A_y = a \sin \theta_2 \quad (3)$$

As coordenadas do ponto B são obtidas usando as equações dos círculos sobre A e O_4 , vistos na Figura 25:

$$b^2 = (B_x - A_x)^2 + (B_y - A_y)^2 \quad (4)$$

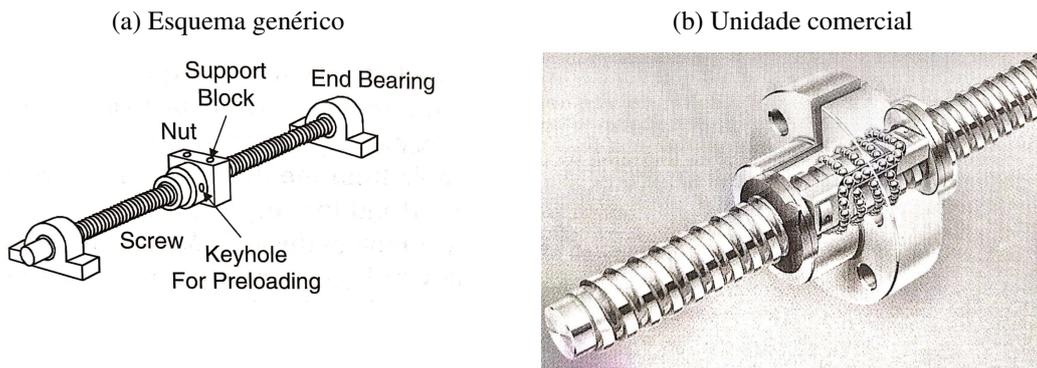
$$c^2 = (B_x - d)^2 + B_y^2 \quad (5)$$

que fornecem um par de equações simultâneas em B_x e B_y .

A.2.2 Fuso de avanço e castanha

Fuso de avanço é um componente de transmissão que converte movimento rotacional em linear. O fuso é rotacionado por um motor e, como resultado, a castanha acoplada se desloca ao longo do eixo do fuso. O bloco de suporte, fixo na castanha, proporciona os meios para acoplar o dispositivo que deve ser movido por esse mecanismo. (SILVA, 2004) Exemplos desse mecanismo pode ser vistos na Figura 26.

Figura 26: Exemplos de mecanismos de fuso de avanço e castanha



Fonte: (SILVA, 2004)

A cinemática desse mecanismo é dada de forma que, para roscas de uma única entrada, uma revolução do motor resulta no deslocamento da castanha a medida de um passo, em condições ideais, como visto na Figura 22. (SILVA, 2004) Na referência utilizada também são encontradas as equações para a dinâmica do mecanismo, porém, por não ser o foco desse trabalho, não serão abordadas.

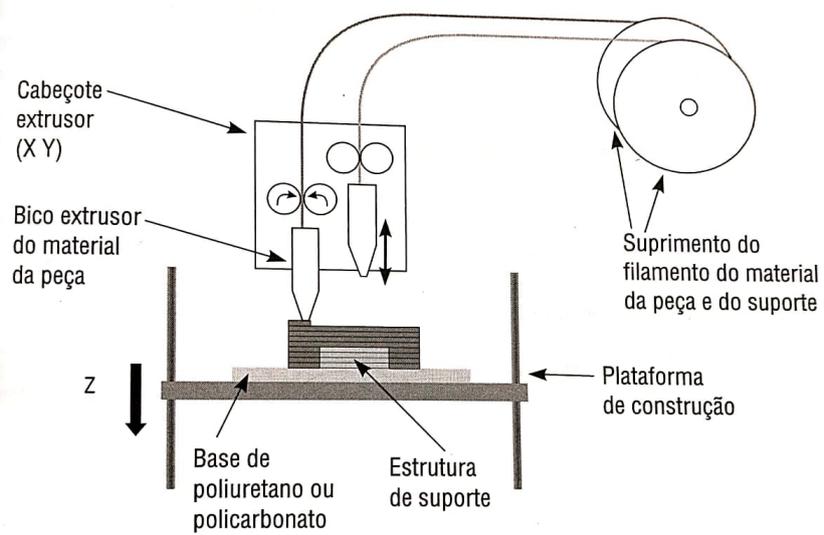
A.3 Prototipagem rápida

O processo utilizado para prototipagem do modelo da cama hospitalar utilizado nesse trabalho será a modelagem por fusão e deposição (FDM). O material utilizado nesse processo encontra-se no estado sólido, extrudado. A cabeça de extrusão, com movimentos no plano horizontal, é posicionada sobre uma mesa com movimento no eixo vertical e recebe continuamente o material na forma de um fio, aquecendo-o até o ponto semilíquido ou pastoso. (VOLPATO et al., 2007)

Ao entrar em contato com o material da superfície da peça, o filamento fino de material extrudado se solidifica e adere à camada anterior. A mesa, então, se desloca no eixo z (vertical) o valor referente à espessura de uma camada e o processo é repetido até que a peça fique pronta. (VOLPATO et al., 2007) A Figura 27 apresenta o esquema de funcionamento dessa tecnologia.

O processo FDM necessita da criação de estrutura de suporte. Esta é utilizada para possibilitar a construção de regiões que não estejam conectadas ao corpo da peça em algum estágio do processo. Regiões desse tipo ficariam suspensas no espaço sem uma estrutura para suportá-las. Esse problema impossibilitaria fisicamente sua construção. Assim, a máquina FDM possui um bico extrusor adicional exclusivo para o material do suporte, que precisam ser removidos depois do processo. (VOLPATO et al., 2007)

Figura 27: Princípio do processo FDM



Fonte: (VOLPATO et al., 2007)