

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JULIANA MARTINS FERREIRA
LUCAS ELIAS CALDART DA SILVEIRA

BABYSWING

OFICINA DE INTEGRAÇÃO 2 – RELATÓRIO FINAL

CURITIBA

2021

JULIANA MARTINS FERREIRA
LUCAS ELIAS CALDART DA SILVEIRA

BABYSWING

Relatório Final da disciplina Oficina de Integração 2, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Dr. César Manuel Vargas Benítez
Prof. Dr. Heitor S. Lopes

CURITIBA

2021

Este trabalho é dedicado a todas as vítimas da COVID-19, dentro e fora do Brasil.

RESUMO

. BABYSWING. 37 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2021.

O projeto descrito neste documento trata-se de um protótipo de uma cadeirinha de balanço para bebês juntamente com uma babá eletrônica. Suas funções são a captação e transmissão em tempo real da imagem e do som do bebê, o controle do balanço a distância e a notificação em caso de detecção de choro, todo controlado através de um aplicativo Android. Para o desenvolvimento de tal, foram realizados os projetos da parte mecânica, do hardware e do software, e posteriormente foi feita a integração de todas as partes resultando no protótipo final.

Palavras-chave: BabySwing, cadeira de balanço, bebê, choro, aplicativo Android, babá eletrônica

ABSTRACT

. BABYSWING. 37 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2021.

The project described in this document refers to the development of a prototype baby rocking chair combined with a baby monitor. It should be capable of capturing and sharing a video and audio live stream of the baby, controlling the rocking motion of the chair and notifying the parents in case of the baby crying. These functions are to be provided through an Android application. Its development went through stages of mechanical, hardware and software design, eventually combining these parts in the final prototype.

Keywords: BabySwing, rocking chair, baby, crying, Android application, baby monitor

LISTA DE FIGURAS

| | | |
|-----------|--|----|
| FIGURA 1 | – Evolução do desenvolvimento visual (RHEIN, 2006) | 13 |
| FIGURA 2 | – GPIO do Raspberry Pi 4 (LTD., 2021) | 14 |
| FIGURA 3 | – Diagrama de blocos. | 17 |
| FIGURA 4 | – Disposição das 4 rodinhas. | 18 |
| FIGURA 5 | – Disposição das barras de metal. | 18 |
| FIGURA 6 | – Posicionamento da mola. | 19 |
| FIGURA 7 | – Posicionamento do servomotor. | 19 |
| FIGURA 8 | – Diagrama de blocos Hardware. | 20 |
| FIGURA 9 | – Ligações Elétricas. | 20 |
| FIGURA 10 | – Esquemático do circuito. | 21 |
| FIGURA 11 | – Diagrama de classes. | 22 |
| FIGURA 12 | – Diagrama caso de uso. | 23 |
| FIGURA 13 | – Statechart do software. | 24 |
| FIGURA 14 | – Diagrama de sequencia - Transmissão. | 24 |
| FIGURA 15 | – Diagrama de Sequência - Movimento. | 25 |
| FIGURA 16 | – Tela Inicial do aplicativo. | 29 |
| FIGURA 17 | – Notificação em caso de desconexão. | 30 |
| FIGURA 18 | – Tela do controle do movimento de balanço. | 31 |
| FIGURA 19 | – Tela do <i>stream</i> | 32 |
| FIGURA 20 | – Cronograma. | 34 |

LISTA DE TABELAS

| | | | |
|----------|---|--------------------------|----|
| TABELA 1 | – | Lista de materiais | 33 |
|----------|---|--------------------------|----|

SUMÁRIO

| | | |
|----------|---------------------------------------|-----------|
| 1 | INTRODUÇÃO | 11 |
| 1.1 | MOTIVAÇÃO | 11 |
| 1.2 | OBJETIVOS | 11 |
| 1.2.1 | Objetivo geral | 11 |
| 1.2.2 | Objetivos específicos | 11 |
| 1.2.2.1 | Requisitos Funcionais | 12 |
| 1.2.2.2 | Requisitos Não Funcionais | 13 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | RASPBERRY PI 4 | 14 |
| 2.2 | SERVOMOTOR MG996R | 14 |
| 2.3 | CONEXÃO SSH | 15 |
| 2.4 | SERVIDOR FLASK | 15 |
| 2.5 | <i>STREAMING</i> | 16 |
| 2.6 | <i>TENSORFLOW</i> | 16 |
| 3 | METODOLOGIA | 17 |
| 3.1 | VISÃO GERAL | 17 |
| 3.2 | PROJETO MECÂNICO | 17 |
| 3.3 | PROJETO DE <i>HARDWARE</i> | 20 |
| 3.4 | PROJETO DE <i>SOFTWARE</i> | 22 |
| 3.5 | INTEGRAÇÃO | 25 |
| 4 | EXPERIMENTOS E RESULTADOS | 27 |
| 5 | CRONOGRAMA E CUSTOS DO PROJETO | 33 |
| 5.1 | CRONOGRAMA | 33 |
| 5.2 | CUSTOS | 33 |
| 6 | CONCLUSÕES | 35 |
| 6.1 | CONCLUSÕES | 35 |
| 6.2 | TRABALHOS FUTUROS | 35 |
| | REFERÊNCIAS | 37 |

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Pais e mães precisam estar sempre atentos às necessidades do bebê o tempo todo. Além disso, a pandemia do COVID-19 trouxe um novo desafio, no qual pais tiveram que se dividir entre a tarefa de trabalhar em home office e cuidar de seus filhos ao mesmo tempo.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O projeto visa unir dois produtos que os auxiliam muito na jornada de cuidar das necessidades de um bebê para torná-la um pouco mais fácil de ser enfrentada. O robô que balança o berço é a união de uma babá eletrônica e de uma cadeira de descanso que balança automaticamente. Além de controlar o movimento de balanço, os responsáveis poderão ver, conversar e escutar o bebê através de um aplicativo Android.

1.2.2 OBJETIVOS ESPECÍFICOS

Com o auxílio de um aplicativo Android os responsáveis terão acesso a uma câmera que estará apontada para o bebê, podendo verificar em tempo real se está tudo bem, se o bebê está dormindo, entre outras situações que sempre preocupam os pais que acabaram de ter um filho. Eles também poderão controlar o tipo de balanço desejado: movimentos extremamente curtos, que imitam a vibração de um carro, movimentos médios, que imitam o balanço de um colo, e movimentos longos e lentos, como o balançar de um carrinho. A cadeira também terá um microfone para identificar quando o bebê começar a chorar e, nesse caso, os pais receberão uma notificação pelo aplicativo.

1.2.2.1 REQUISITOS FUNCIONAIS

- RF1: A cadeira deve ter a capacidade de balançar com o ângulo de 10 graus a 20 graus, acompanhando a curvatura da cadeira de repouso escolhida. E deve ter três intensidades de movimento a serem escolhidas pelo usuário.
- RF2: Os movimentos poderão ser: extremamente curtos, medianos ou longos e lentos.
- RF3: A intensidade do movimento deve ser alterada pelo aplicativo.
- RF4: O microfone instalado deve ser utilizado para detectar sons altos e agudos que se assemelhem ao choro estridente de um bebê.
- RF5: Em caso de detecção do som do requisito RF3, deve ser enviada uma notificação pelo aplicativo.
- RF6: O microfone também será utilizado na transmissão de áudio em tempo real do bebê para os pais.
- RF7: O alto-falante será utilizado para transmitir a voz dos pais ao bebê.
- RF8: O aplicativo deve captar a voz dos pais para transmissão à cadeira.
- RF9: A microfonia na interação entre o microfone e o alto-falante deve ser mínima.
- RF10: A câmera apontada para o bebê fornecerá o vídeo em tempo real para o aplicativo utilizado pelos pais.
- RF11: O aplicativo testará periodicamente a conexão com a cadeirinha, gerando uma notificação em caso de falha, como em caso de interrupção da alimentação elétrica.

Observação: Optou-se por não colocar uma tela na cadeirinha de forma que o bebê pudesse visualizar os pais, uma vez que a visão do bebê demora a se desenvolver, como mostra a Figura 1. Portanto, seria um custo a mais para o projeto que não traria benefícios reais ao produto.

MARCOS DA EVOLUÇÃO DO DESENVOLVIMENTO VISUAL

| Idade | Desenvolvimento visual |
|--------------------------------------|---|
| 30 a 34 semanas de idade gestacional | Reação pupilar à luz, fechar as pálpebras diante de luz intensa e reflexo de fixação. |
| Ao nascimento | Busca da fonte luminosa, fixação visual presente, mas breve, tentativas de seguir objeto em trajetória horizontal. |
| 0 a 1º mês: | Reação palpebral e à luz Reação à alto contraste Visão de vultos Fixação fugaz e de seguimento (rudimentar) Visão central ruim |
| 1º ao 2º mês | Visão de seguimento visual mais suave Presta atenção ao rosto da mãe Reage à padrões complexos Inicia movimento de coordenação binocular |
| 2º ao 3º mês | Fixa, converge e acomoda Discrimina rostos – responde com sorrisos Reage a cores: amarela, azul, vermelha Inicia estimulação de visão central pelas mãos em linha média (simetria) |
| 3º ao 4º mês | Mantém contato visual Observa movimento das próprias mãos Coordenação Olho/Mão Explora visualmente os pés |
| 5º ao 6º mês | Alcance dirigido visualmente Agarra objetos, examina-os visualmente Coordenação Olho/Objeto |
| 6º ao 7º mês | Transfere o olhar de um objeto para o outro Momento de exploração e descoberta (pega, joga, sacode, puxa) Combina a experiência motora |

Figura 1: Evolução do desenvolvimento visual (RHEIN, 2006)

1.2.2.2 REQUISITOS NÃO FUNCIONAIS

- RN1: Todo o conjunto será alimentado através de tomada ligada à rede 127V.
- RN2: A comunicação entre o conjunto e o aplicativo deverá ser via Wi-fi, com o Raspberry Pi funcionando como ponto de acesso para uma rede local.
- RN3: O aplicativo será desenvolvido para Android.
- RN4: A integração entre os diferentes componentes do projeto será provida através de um Raspberry Pi.
- RN5: O aplicativo deve ser user friendly para o público alvo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 RASPBERRY PI 4

O Raspberry Pi 4 Model B é um computador de placa única pequeno, versátil e de baixo custo, sendo muito utilizado no aprendizado e em aplicações relacionadas a área de computação. O modelo fornece conectividade wi-fi integrada e entradas USB que podem ser utilizadas para a conexão da câmera, do microfone e do alto-falante. Os pinos de entrada e saída de uso geral podem ser utilizados para o controle de periféricos, como o servo motor que será utilizado nesse projeto.

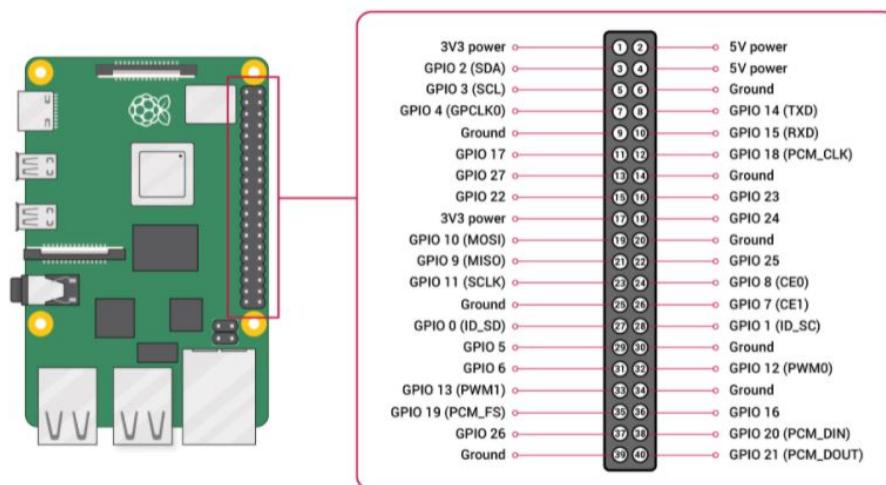


Figura 2: GPIO do Raspberry Pi 4 (LTD., 2021)

2.2 SERVOMOTOR MG996R

O servomotor utilizado é um motor DC realimentado em malha fechada controlado através da modulação por largura de pulso (Pulse-width modulation - PWM) (WIKIPEDIA, 2021b). Sua corrente de operação é de 500mA a 900mA, e sua tensão de alimentação é 4.8 a 7.2V, sendo que utilizaremos 7.2V para que possamos ter o maior torque que ele pode fornecer

(13kg.cm). Suas dimensões aproximadas são 40.7x19.7x42.9mm e pesa aproximadamente 55g.

2.3 CONEXÃO SSH

Para iniciar, interromper e trocar as rotinas de comando do servomotor responsável pelo balanço da cadeira, foi utilizado o protocolo Secure Shell (SSH), um protocolo de rede criptografado que fornece um canal seguro para operar serviços através de redes desprotegidas (WIKIPEDIA, 2021a). Apesar de qualquer serviço de rede poder ser operado através do SSH, uma de suas aplicações mais comuns, que foi utilizada neste trabalho, é a execução de linha de comando remota. O SSH funciona no modelo cliente-servidor, no qual uma aplicação fornece um serviço que os clientes podem fazer uso. O Raspberry Pi provê um servidor SSH previamente configurado pronto para receber conexões através da porta 22, a qual é comumente utilizada para conexão SSH. Para esse projeto, a implementação do SSH para o cliente (aplicativo Android), foi realizada a partir dos métodos providos pela biblioteca JSch. Intermediado pela biblioteca, um objeto de sessão entre o cliente e o servidor é criado e utilizado para o envio de comandos que executam scripts python armazenados no Raspberry Pi. Após enviado o comando, a sessão é encerrada e o aplicativo volta ao estado de espera de uma nova interação por parte do usuário.

2.4 SERVIDOR FLASK

A funcionalidade de verificação da manutenção da conexão entre o Raspberry Pi e o aplicativo foi feita com requisições para uma aplicação web implementada com Flask. O Flask é um framework de desenvolvimento web concebido com o objetivo de simplicidade, sem funções tais como ORM (object relational mapping, ou mapeamento relacional de objetos) ou autenticação, sendo por isso classificado como um microframework. Frameworks web tipicamente provêm serviços facilitadores para desenvolvimento web através de bibliotecas para, por exemplo, gerenciamento de sessões e acesso a banco de dados. No caso do Flask, um dos poucos serviços disponíveis sem a adição de extensões é a resposta a requisições com métodos HTTP. Os métodos HTTP são definidos de forma a identificar com um verbo o objetivo da requisição enviada. O método GET, por exemplo, responde a uma requisição simples enviada com uma URL base, devolvendo dados que representem de alguma forma um estado atual do servidor. Nos aplicativos Android, a biblioteca Volley do Google fornece os métodos necessários para realizar essas requisições. O aplicativo faz então requisições periódicas ao servidor Flask instalado no Raspberry Pi, e gera uma notificação para o usuário caso a requisição

não obtenha resposta, o que configura a falha de conexão entre o aplicativo e o Raspberry Pi (PALLETTS, 2010).

2.5 *STREAMING*

A visualização do vídeo em tempo real do bebê ocorre através do acesso a uma URL onde o streaming fica disponível. O vídeo é fornecido por um pequeno módulo de câmera para Raspberry Pi conectado através de uma porta para interface serial CSI (camera serial interface), própria para câmeras, enquanto o microfone que recebe os sons do ambiente é ligado em uma das entradas USB da placa. A comunicação entre a câmera, o microfone e o software é intermediada pela biblioteca Picam, que é capaz de acessar esses periféricos e gerar um fluxo de vídeos no formato m3u8 (HLS, ou HTTP Live Streaming) para streaming. Por limitações do formato HLS, a latência mínima da transmissão é de 3 segundos, podendo ser até maior dependendo de fatores como a qualidade do vídeo e da conexão de rede. Da mesma forma que na funcionalidade de verificação da conexão, é necessário que um servidor web responda as requisições com os arquivos de vídeo que, sequencialmente, formam a o vídeo em tempo real. No aplicativo Android, um tocador de mídia é implementado através da biblioteca ExoPlayer, mantida pelo Google. Esse tocador de mídia recebe como um de seus parâmetros a URL onde está exposto o fluxo de vídeo e acessa esses arquivos montando, também sequencialmente, o vídeo em tempo real para exibição.

2.6 *TENSORFLOW*

Para a detecção do choro do bebê, pretendia-se utilizar a biblioteca TensorFlow, interfaceada através da biblioteca Keras para comparar o áudio em tempo real com um modelo de rede neural. As redes neurais são sistemas para computação com funcionamento baseado nos cérebros de animais. Elas são formadas por camadas de unidades de processamento conhecidos como neurônios, sendo conectadas entre si de forma análoga às sinapses, formando um grafo. A rede neural recebe uma entrada de dados que é sucessivamente processado pelas camadas de neurônios até que fique disponível na camada de saída cujo resultado pode ser classificado de acordo com o modelo fornecido. A criação de um modelo de rede neural é chamada de treinamento, e para isso é fornecido um conjunto de dados (dataset) com cada objeto adequadamente rotulado.

3 METODOLOGIA

3.1 VISÃO GERAL

Optou-se por alimentar o conjunto através de uma tomada ligada a rede 127V, uma vez que o projeto foi desenvolvido para uso exclusivo domiciliar. Não se pode deixar um bebê sem a supervisão de um adulto, o projeto visa apenas possibilitar que os responsáveis fiquem em cômodos diferente do mesmo domicílio.

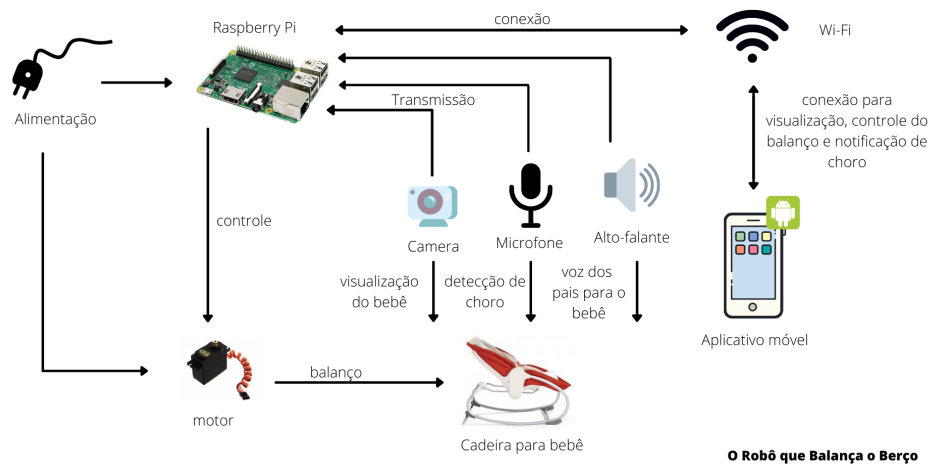


Figura 3: Diagrama de blocos.

3.2 PROJETO MECÂNICO

A estrutura mecânica foi desenvolvida com tábuas de Pinus. Optou-se pela utilização de 4 rodinhas (Figura 4) junto com uma alça para auxiliar na locomoção da cadeirinha.

Duas barras de aço foram fixadas na cadeirinha (Figura 5). A barra frontal servirá para converter o movimento gerado pelo servo motor no balanço da cadeirinha. A barra posterior será conectada a mola, fornecendo estabilidade para o movimento.



Figura 4: Disposição das 4 rodinhas.



Figura 5: Disposição das barras de metal.

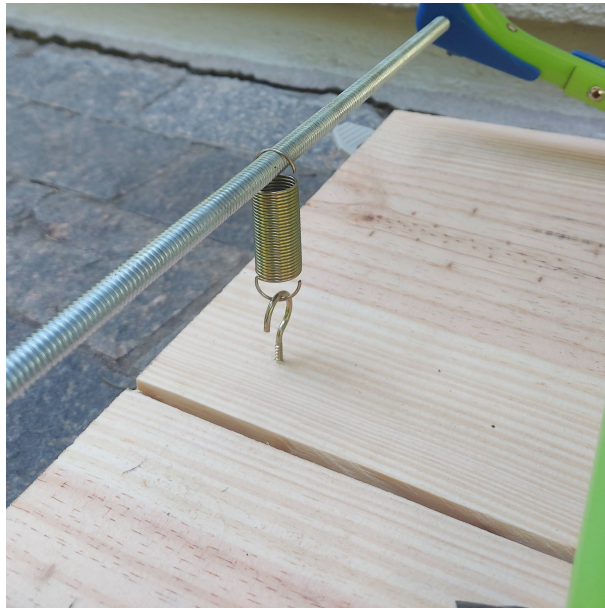


Figura 6: Posicionamento da mola.

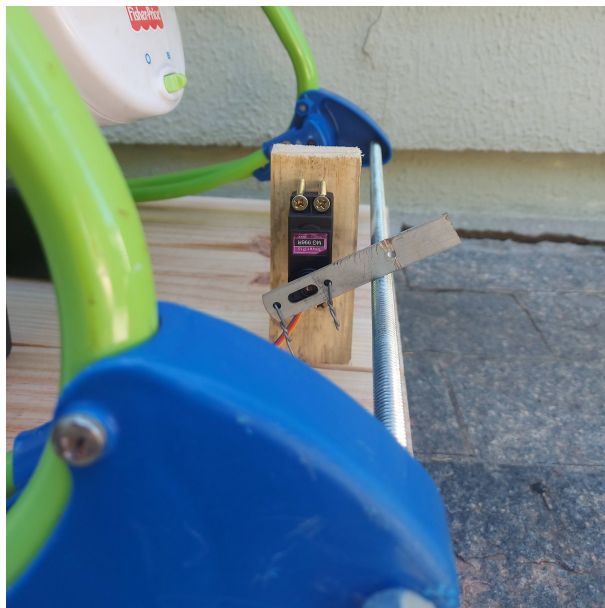


Figura 7: Posicionamento do servomotor.

3.3 PROJETO DE *HARDWARE*

No projeto hardware temos um alto falante e um microfone ligados ambos no Raspberry Pi através de portas USB, e uma câmera conecta através da porta CSI (*camera serial interface*) do Raspberry Pi.

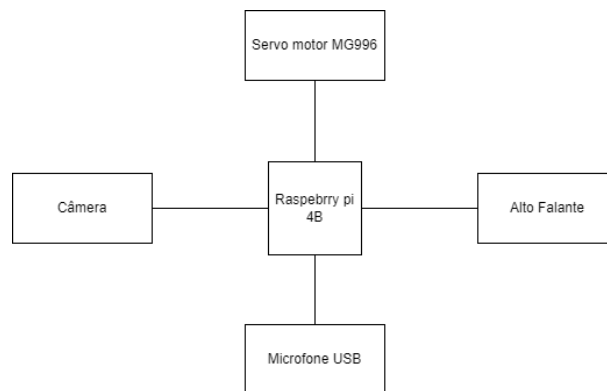


Figura 8: Diagrama de blocos Hardware.

Para que o servomotor tenha o maior torque, ele precisa ser alimentado com 7V, já o Raspberry Pi precisa ser alimentado 5V. Para isso, foram usadas duas fontes comerciais independentes conectadas em um único plug de tomada e com as referências devidamente conectadas, como mostra a Figura 9. A escolha de se usar duas fontes foi feita também para evitar a possível interferência do servo com o Raspberry Pi.

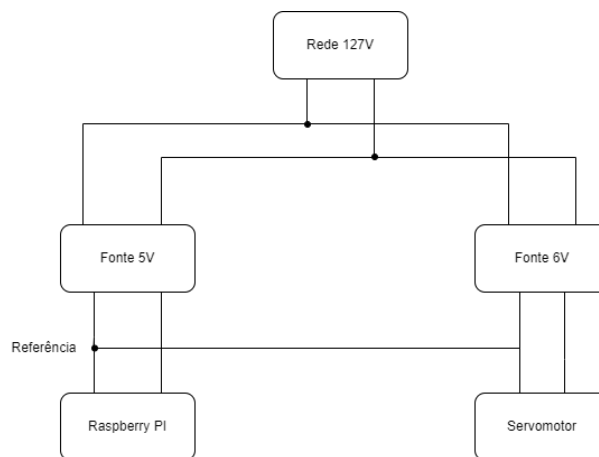


Figura 9: Ligações Elétricas.

E a Figura 10 mostra o esquemático do nosso projeto.

3.4 PROJETO DE SOFTWARE

A equipe optou pelo desenvolvimento de um aplicativo Android uma vez que é o sistema operacional que temos aceso. O aplicativo foi desenvolvido com o auxílio do Android Studio utilizando a linguagem Kotlin. Inicialmente, no plano de projeto estava previsto o desenvolvimento em Java, contudo após a discussão inicial foi sugerido a troca para Kotlin. A Figura 11 mostra o diagrama de classes.

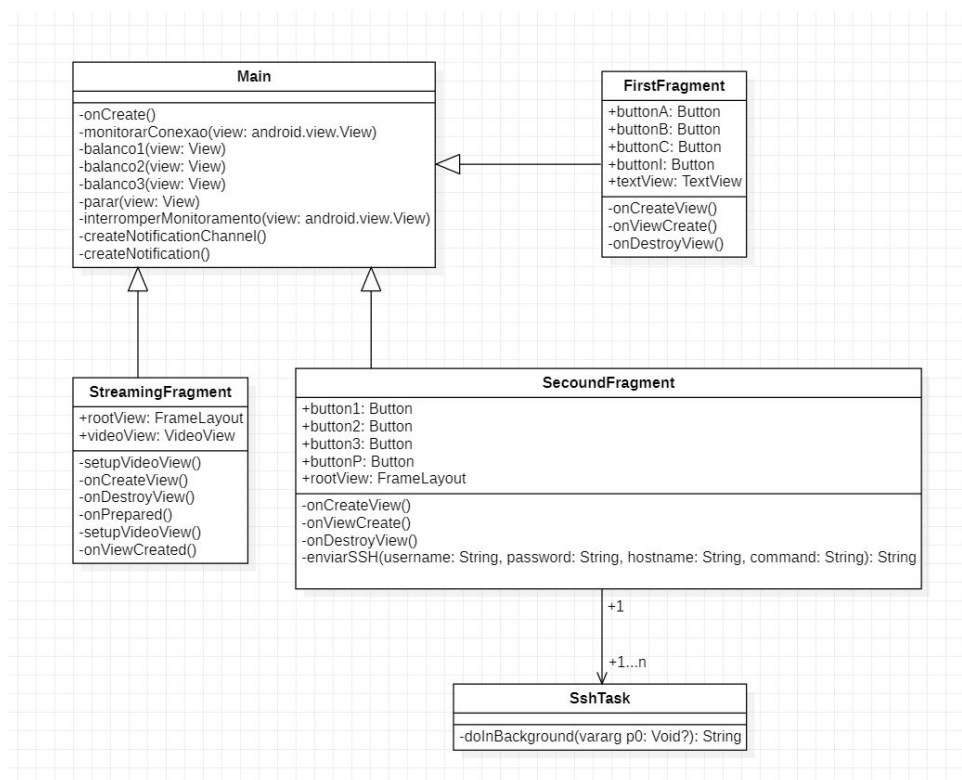


Figura 11: Diagrama de classes.

A Figura 12 ilustra os casos de usos existentes para o software bem como seus atores. E a Figura 13 ilustra o statechart do software desenvolvido pela equipe.

Já a Figura 14 representa o Diagrama de Sequencia do processo de transmissão do vídeo e áudio do bebê. E por fim, a Figura 15 representa o diagrama de Sequencia do processo do balanço da cadeirinha.



Figura 12: Diagrama caso de uso.

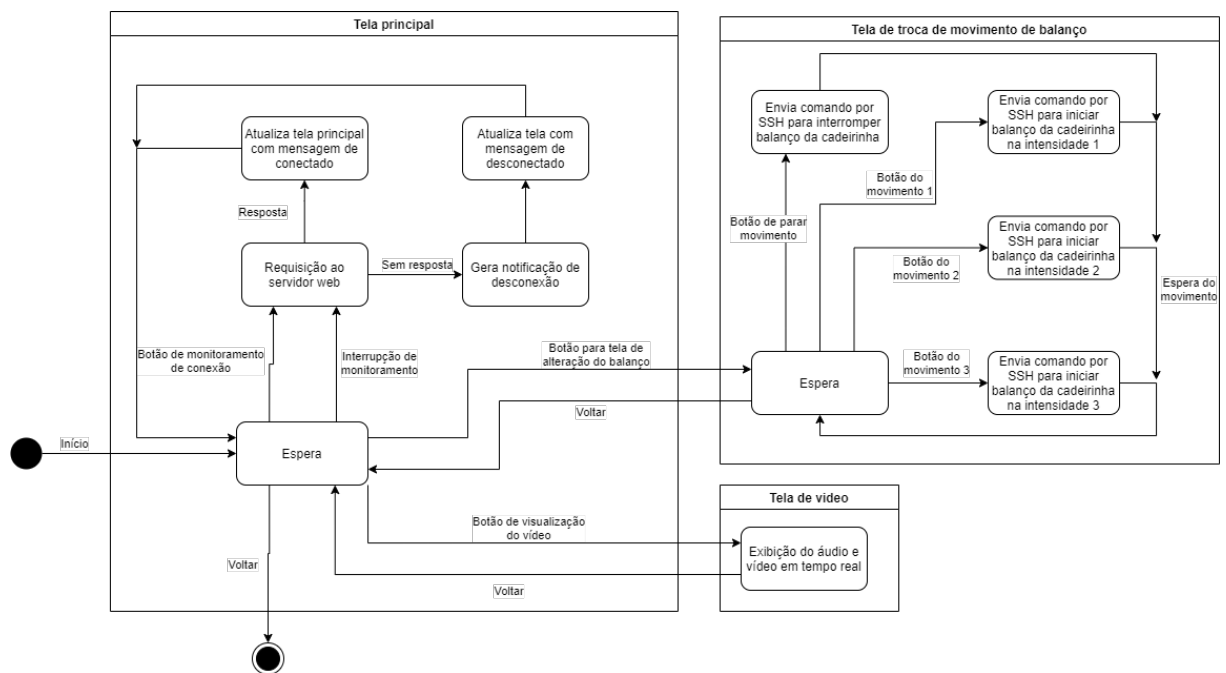


Figura 13: Statechart do software.

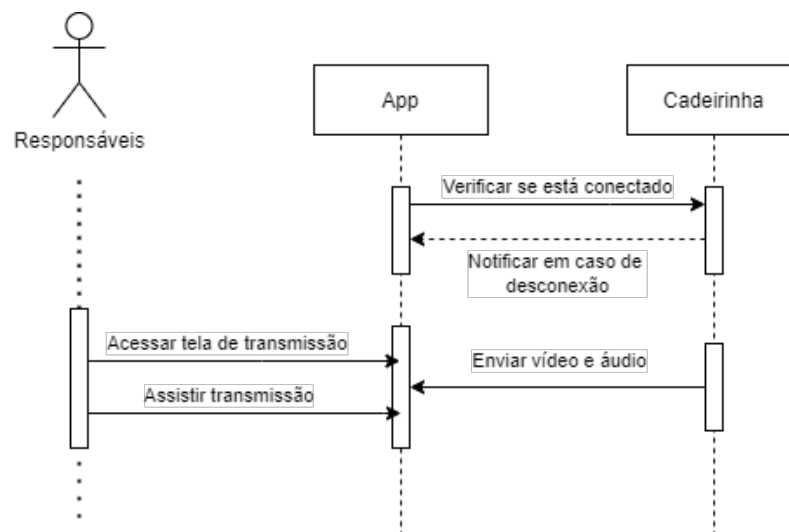


Figura 14: Diagrama de sequencia - Transmissão.

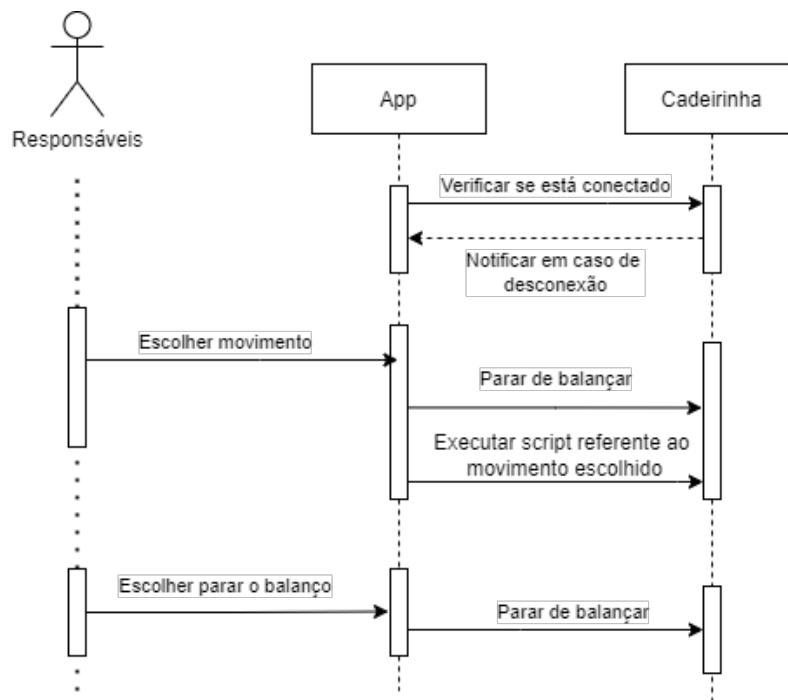


Figura 15: Diagrama de Sequência - Movimento.

3.5 INTEGRAÇÃO

As integrações entre a parte mecânica, software e hardware foram sendo realizadas durante a própria implementação do projeto, uma vez que todas essas partes estão fortemente interligadas. Para testar, por exemplo, se o aplicativo está exibindo corretamente o *stream* de vídeo e áudio do bebê é necessário que esteja integrado com o Raspberry Pi e que ele esteja funcionando. Portanto, durante o tempo previsto para essa etapa a equipe concentrou seus esforços na implementação dos requisitos funcionais relacionados ao aplicativo e sua consequentemente integração com a parte mecânica e hardware.

Durante o desenvolvimento do projeto diversas disciplinas, citadas abaixo, já cursadas pela equipe foram essenciais.

- Introdução à Lógica para Computação: a lógica forma a base de todo sistema decisório e de todo controle de fluxo em programas.
- Fundamentos de Programação 1: toda a programação aprendida ao longo do curso se apoia no contato tido inicialmente durante essa disciplina.
- Física Teórica A: compreensão dos esforços envolvidos na estrutura e movimentação do conjunto mecânico.

- Técnicas de Programação: aprofundamento em programação, especialmente se tratando do paradigma de orientação de objetos, que será utilizado para a programação do aplicativo.
- Introdução a Prática de Laboratório em Eletricidade e Eletrônica: base da operação e ligação de componentes envolvendo eletricidade.
- Eletricidade: conceitos básicos sobre a interação entre componentes envolvendo eletricidade.
- Oficina de Integração 1: experiência necessária para integração e controle com componentes eletrônicos e mecânicos.
- Análise e Projeto de Sistemas: conceitos de análise e projeto de sistemas e sua aplicação no desenvolvimento de programas.
- Comunicação de Dados: conhecimentos sobre a conexão de sistemas para transmissão de dados através de redes (wi-fi).
- Desenho Técnico Aplicado: Desenho, montagem, conexão e utilização de placas de circuito.

4 EXPERIMENTOS E RESULTADOS

Durante todo o desenvolvimento de projeto foram realizados testes a fim de verificar o correto funcionamento de cada parte que era acrescentada ao conjunto.

Primeiramente, o servo, a câmera, o microfone e o alto falante, assim que recebidos foram testados individualmente. Para o servo foi feito um programa Python simples que apenas ia para posição 0° , esperava 1 segundo, ia para a posição 45° , esperava novamente 1 segundo, e, finalmente mudava para a posição 90° . Nessa etapa, um tempo foi gasto para identificar que uma vez que foram utilizadas duas fontes independentes, era preciso que o servomotor e Raspberry Pi tivessem a mesma referência de alimentação para o seu correto funcionamento. Os membros já haviam trabalhado com micro servomotores anteriormente e microcontroladores, mas por se tratar de projetos menores, o servomotor foi alimentado pela mesma fonte do microcontrolador, não sendo necessária essa conexão. Portanto, esse não era um conceito pre-adquirido. O microfone e o alto falante foram testados rapidamente conectando-os na USB do computador. A câmera foi testada inicialmente tirando fotos e em seguida diretamente na própria implementação da transmissão do vídeo.

A falta de conhecimentos Linux dificultou a instalação e utilização dos pacotes necessários para a captura da imagem e do som, e a posterior criação da *stream*. A recepção da stream no aplicativo Android foi especialmente complicada, sendo as referências encontradas para essa implementação razoavelmente antiga, com métodos depreciados e fora de uso, um dos motivos que levou o grupo a trocar o desenvolvimento do aplicativo de Java para Kotlin. A exibição correta da stream no aplicativo também necessitou da inclusão no código fonte de declaração permitindo a utilização de comunicação HTTP sem a camada de segurança.

Para a detecção do choro, foi utilizada o conjunto de dados "ESC-50: dados para classificação de sons ambiente", um conjunto contendo 50 tipos de sons com 40 amostras cada, validado no artigo de Piczak (2015) que deu origem, sendo um dos tipos de sons o de bebês chorando. O modelo foi gerado a partir dessa base com sucesso, mas não foi possível aplicá-lo ao projeto devido a uma dificuldade no pré-processamento do áudio para segmentá-lo e fornecê-

lo ao modelo de predição da rede neural.

A montagem mecânica foi realizada com o auxílio de um parente de um dos membros do grupo, que tinha as ferramentas necessárias para realizar os cortes na madeira e fixar os componentes.

A Figura 16 mostra a tela Inicial do aplicativo, no caso em que houve uma desconexão com a cadeira.

A Figura 17 mostra a notificação gerada pelo aplicativo, no caso em que houve a desconexão com a cadeira.

A Figura 18 mostra a tela em que é possível escolher o movimento de balanço ou a interrupção do movimento.

A Figura 19 mostra a tela de exibição do vídeo e do áudio em tempo real.

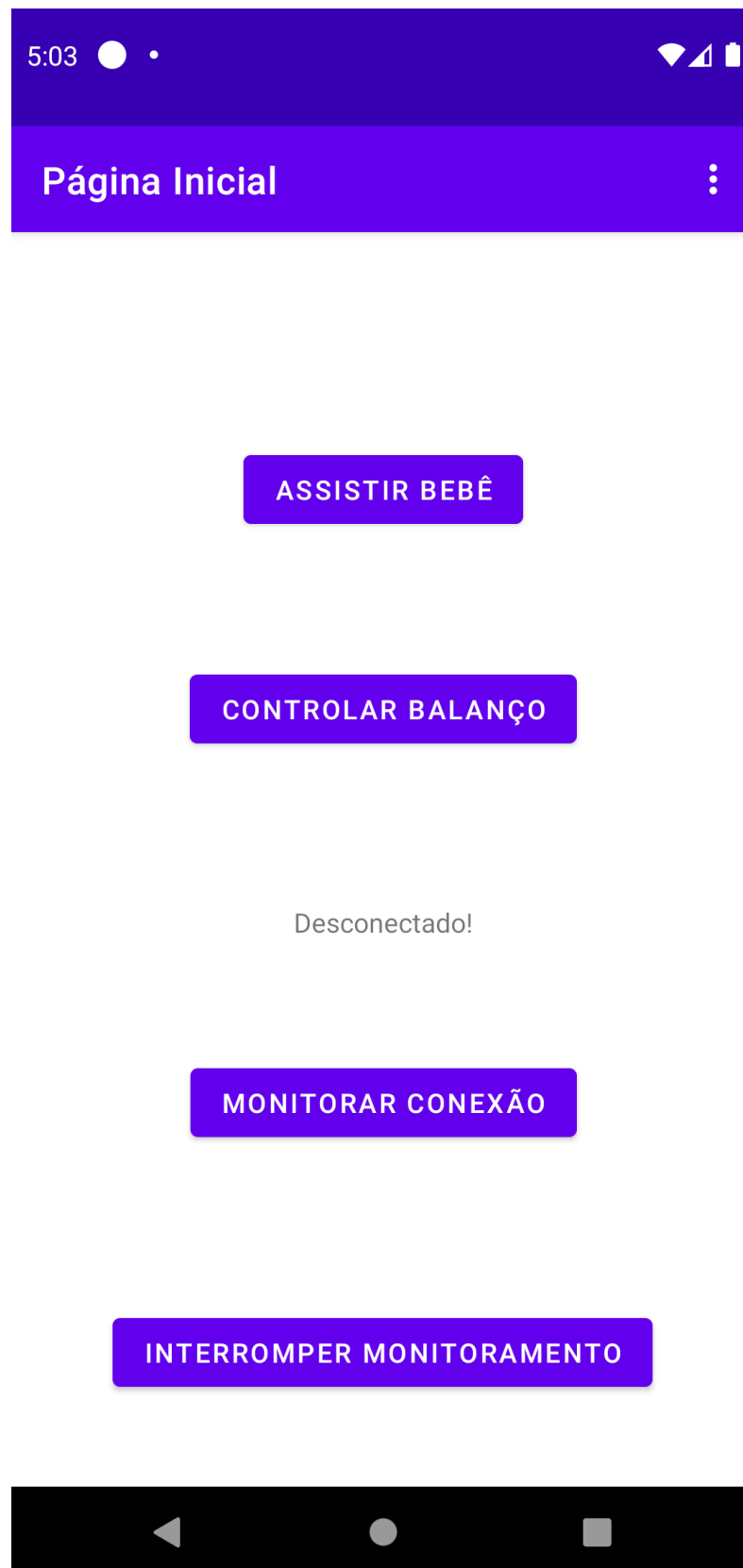


Figura 16: Tela Inicial do aplicativo.

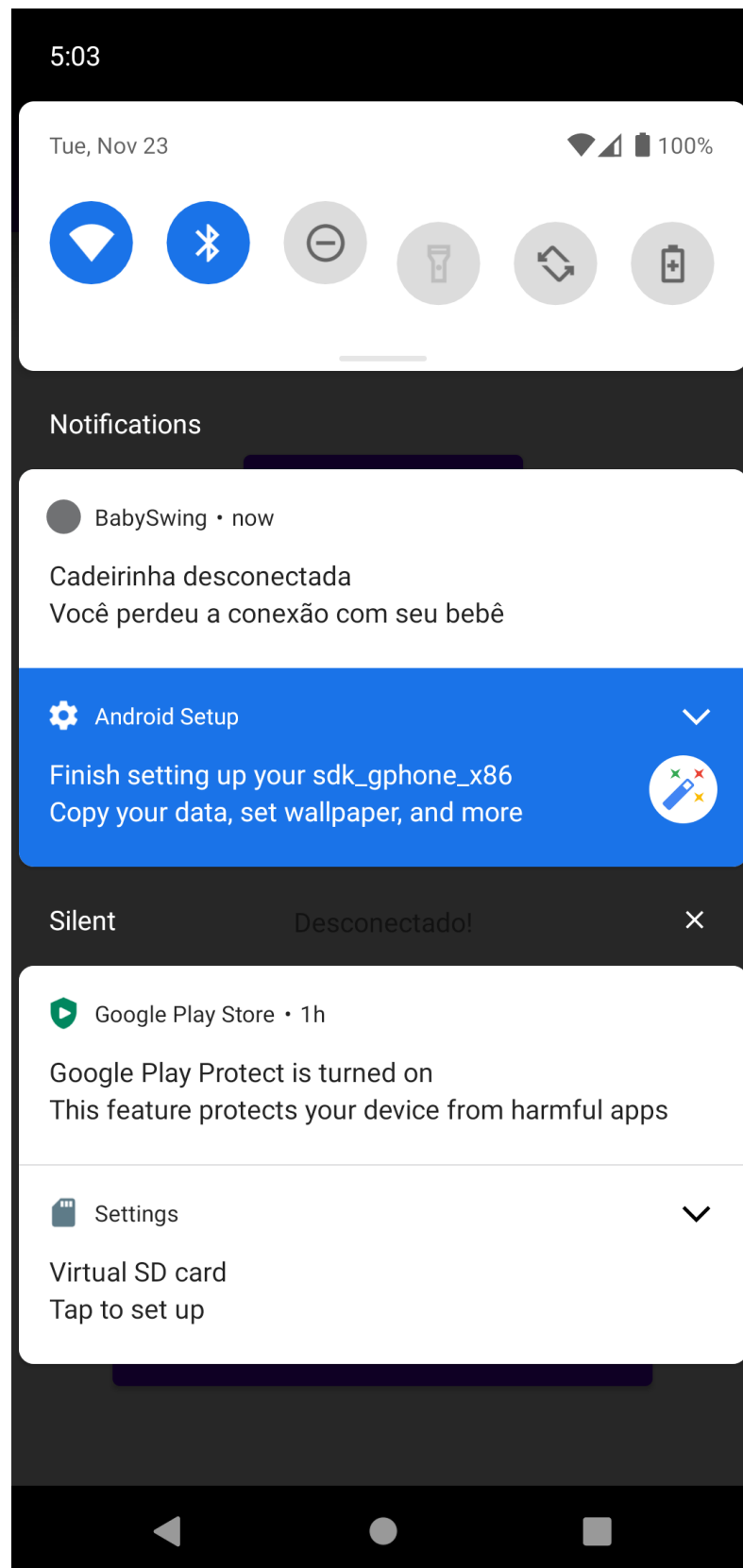


Figura 17: Notificação em caso de desconexão.



Figura 18: Tela do controle do movimento de balanço.

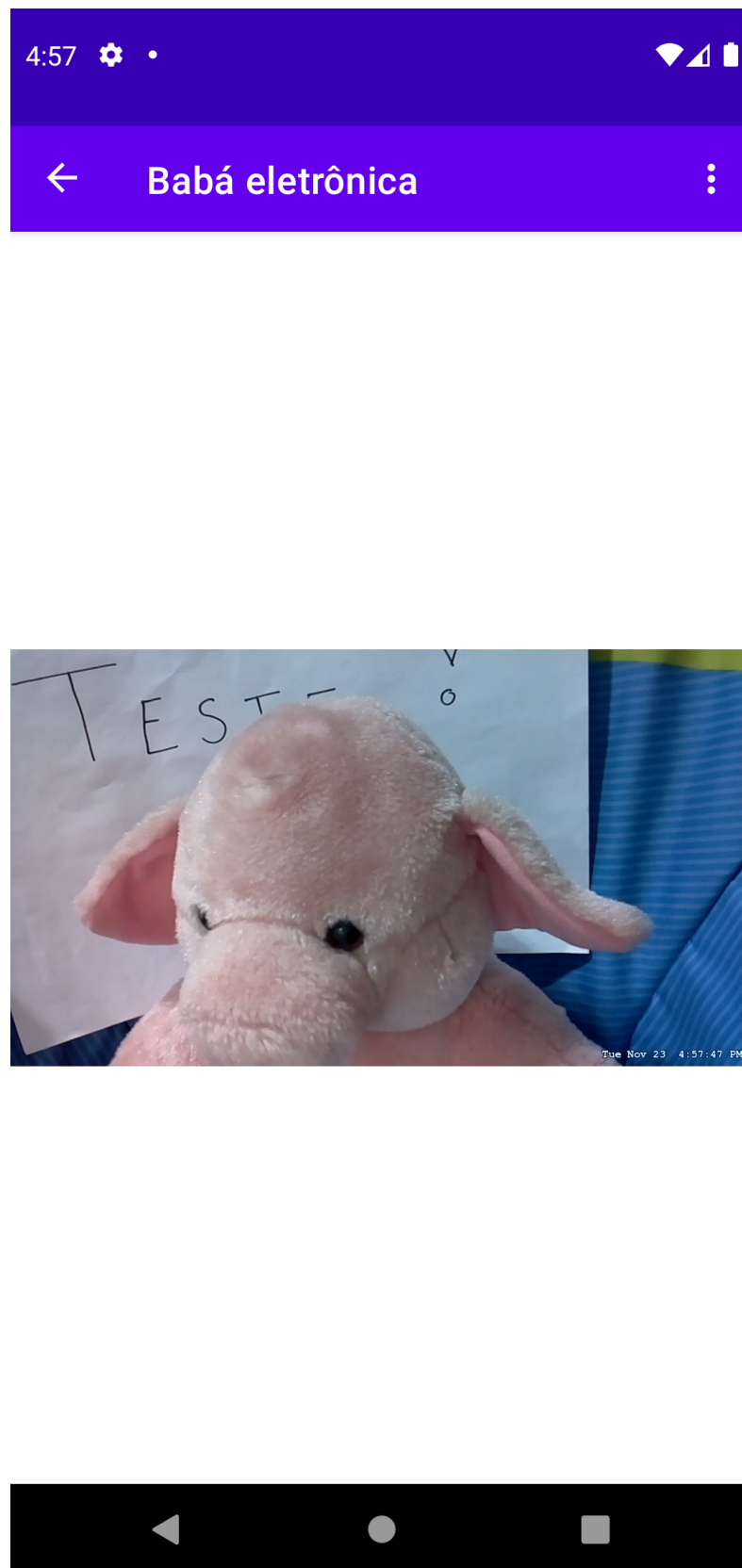


Figura 19: Tela do *stream*

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

A figura 20 apresenta o cronograma seguido pela equipe no desenvolvimento do trabalho.

O cronograma também pode ser visualizado através do link https://docs.google.com/spreadsheets/d/1Ce3Ae8nfBja1zeRbaumBCh0KL9d2sVq_6ehsr5jk/edit?usp=sharing

5.2 CUSTOS

A tabela 1 apresenta os custos que a equipe deve para o desenvolvimento do projeto. Os gastos foram divididos igualmente entre os dois membros da equipe.

Tabela 1: Lista de materiais

| Material | Quantidade | Preço |
|--------------------------------|------------|---------|
| Raspberry Pi 4 | 1 | 890.00 |
| Servo motor MG996R | 1 | 60.00 |
| Cadeira de bebê | 1 | 150.00 |
| Cartão micro sd | 1 | 60.00 |
| Madeira para suporte em pinus | 1 | 100.00 |
| Mola para auxílio no movimento | 1 | 10.00 |
| Microfone | 1 | 29.90 |
| Alto-falante | 1 | 19.90 |
| Câmera | 1 | 50.00 |
| Barra de aço 9.5mm | 1 | 20.00 |
| Cabo Flat 1m 15 vias | 1 | 48.00 |
| Fonte 5V | 1 | 50.00 |
| Fonte 7V | 1 | 50.00 |
| Frete | 1 | 24.00 |
| | Total | 1561.80 |

6 CONCLUSÕES

6.1 CONCLUSÕES

Ao decorrer do projeto a equipe encontrou várias dificuldades a serem superadas. As dificuldades mais importantes e que demandaram mais tempo foram em relação ao aplicativo, uma vez que nenhum estudante da dupla havia anteriormente trabalhado com aplicativo Android. Dessa forma, muitas vezes no decorrer do projeto a equipe se deparava com um erro que pode ser considerado simples, porém a equipe perdia horas em busca da solução uma vez que não tinha conhecimento de um conceito próprio de aplicativos Android. Sendo o maior empecilho do grupo o tempo extremamente limitado gerado pela pandemia do COVID-19, e consequentemente, o semestre reduzido.

Entretanto, justamente essas dificuldades encontradas foram o que levou a equipe a aprender muito durante o desenvolvimento do projeto. Tanto sobre a parte mecânica, hardware e software, quanto sobre o desenvolvimento de projetos de forma geral, o trabalho em equipe e a importância de um bom planejamento e um bom cronograma antes de começar a implementação do projeto.

Apesar da equipe não ter tido o tempo hábil para a implementação do requisito funcional que se referia a transmissão da voz dos pais ao bebê. A equipe considera que o resultado final do trabalho foi bem positivo levando em conta o semestre reduzido e todo conhecimento adquirido. Pois até mesmo do requisito não atendido a equipe consolidou e se aprofundou nos conhecimentos de sockets aprendidos anteriormente na disciplina de Comunicação de dados.

6.2 TRABALHOS FUTUROS

Como trabalho futuro, melhorias poderiam ser aplicadas sobre o projeto atual. Como, por exemplo, a implementação da comunicação pais→bebê utilizando os auto falantes instalados na cadeirinha e o microfone do próprio celular dos responsáveis. Essa aplicação foi

inicialmente proposta, mas devido ao curto período de tempo não foi desenvolvida a tempo da apresentação deste relatório e projeto final pela equipe.

Ainda utilizando os auto falantes, seria possível a implementação de sons da natureza, sons relaxantes para o bebê relaxar e dormir. Ruídos brancos que os responsáveis poderiam escolher com o uso do aplicativo. Ou ainda, sons padrões, mas que os pais também pudessem gravar outros sons no lugar, adaptando-se melhor a individualidade cada criança.

REFERÊNCIAS

LTD., R. P. **Raspberry Pi Documentation**. 2021. <https://www.raspberrypi.com/documentation/computers/os.html>. Acessado em 25 de novembro de 2021.

PALLETS. **Flask**. 2010. <https://flask.palletsprojects.com/en/2.0.x/>. Acessado em 26 de novembro de 2021.

PICZAK, K. J. Esc: Dataset for environmental sound classification. **MM '15: Proceedings of the 23rd ACM international conference on Multimedia**, 2015.

RHEIN, L. **Pediatria e Visão. Cert - Editora de Revistas Técnicas Ltda. São Paulo**. 2006.

WIKIPEDIA. **Secure Shell**. 2021. https://en.wikipedia.org/wiki/Secure_shell. Acessado em 26 de novembro de 2021.

WIKIPEDIA. **Servomotor**. 2021. <https://pt.wikipedia.org/wiki/Servomotor>. Acessado em 25 de novembro de 2021.