UNIVERSIDADE FEDERAL DO ABC
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO DE ESTÁGIO SUPERVISIONADO III

NATÁLIA SATIE MOTOKUBO HALKER

Orientador : Prof. Dr. Fabrício Olivetti de França

Santo André - SP
2019

NATÁLIA SATIE MOTOKUBO HALKER

PROCESSO DE IMPRESSÃO 3D DE PEÇAS DIGITALIZADAS NO LABORATÓRIO

Relatório de estágio apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Federal do ABC como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Fabrício Olivetti França

Santo André - SP
2019

DEDICATÓRIA

Dedico à todos que colaboraram com o desenvolvimento deste relatório.

AGRADECIMENTOS

Agradeço ao professores:

- Fabrício Olivetti, que orientou na escrita deste relatório de estágio;

- Mario Gazziro, que auxiliou com dicas e alguns equipamentos para a realização do projeto;

Também agradeço ao meu supervisor de estágio, Edgard Gonçalves, por me orientar nas tarefas a serem realizadas no decorrer do estágio.

RESUMO

O relatório tem como objetivo descrever as tarefas realizadas nos laboratórios secos destinados à graduação da UFABC. Tais tarefas foram desenvolvidas com base nos critérios estabelecidos na disciplina de Estágio Supervisionado III do curso de Bacharelado em Ciência da Computação da Univerdade Federal do ABC, totalizando 96 horas de estágio. As atividades realizadas no estágio incluem desenvolvimento de um sistema de digitalização de objetos a partir de um dispositivo ótico capaz de gerar uma nuvem de pontos e um arquivo que a impressora 3D é capaz de ler e imprimir. Além do acompanhamento e preparação de aula nos laboratórios didáticos secos 404-1 e 406-1 pertinentes a matérias da graduação.

Palavras-Chave: Impressão, digitalização, Impressora 3D, Desenvolvimento.

ABSTRACT

This report aims to describe the performed tasks in Dry Laboratories used during UFABC undergraduation programs. These tasks were developed based on established criteria by Supervised Internship from Computer Science Bachelor degree at the Federal University of ABC, totaling 96 hours of intership. The performed activities at the internship include the development of an object scanning system from an optical device capable of generating a cloud of dots and a 3D printer compatible file ready to print. Another reported task is the preparation of didactic materials for laboratories 404-1 and 406-1 and class tutorin.

Keywords: Print, Scan, 3D Printer, Development

Lista de Figuras

Foto Kinect [autoria própria] pag 17

Foto da mesa de impressão com eixos desenhados [autoria própria] pag 18

Foto da parte traseira da impressora [autoria própria] pag 19

Foto display da impressora (ligado via USB) [autoria própria] pag 20

Captura da tela com demonstração do ReconstructMe [8] pag 21

Captura da tela com demonstração do MeshLab [10] pag 22

Exemplo de aplicação de filtros do PyMesh[11] pag 22

Captura da tela com demonstração do Meshmixer [13] pag 23

Módulo de Bluetooth para comunicação [autoria própria] pag 24

Placa de desenvolvimento Arduino UNO [autoria própria] pag 25

Sensor de gás [autoria própria] pag 26

IDE do MIT App Inventor [figura de autoria prórpia,
site: ai2.appinventor.mit.edu] pag 27

Foto do mini robô [autoria própria] pag 28

Captura de tela com o resultado da sobreposição e renderização das
imagens no KScan3D [autoria própria] pag 28

Captura da tela do Skanect com o mini robô sendo digitalizado
[autoria própria] pag 29

Captura da tela do ReconstructMe com o mini robô digitalizado
[autoria própria] pag 31

Captura da tela durante a digitalização de uma pessoa [autoria
própria] pag 32

Digitalização do objeto no MeshLab antes da limpeza [autoria própria] pag 33

Captura da tela Meshmixer antes da aplicação de filtros [autoria própria] pag 34

Captura da tela MeshMixer: objeto com suporte e pronto para impressão [autoria própria] pag 35
Captura da tela do Repetier com objeto pronto para impressão [autoria própria] pag 36
Foto início da impressão [autoria própria] pag 37
Fotos da impressão pronta [autoria própria] pag 37
Esquema de montagem do protótipo [autoria própria] pag 40
Protótipo montado [autoria própria] pag 40
IDE do Arduino [autoria própria] pag 41
IDE App Inventor (a) Interface (b) Programação em blocos [autoria própria] pag 42
Algumas das telas do aplicativo (a) Login (b) Menu (c) Armazenar dado (d) Histórico [autoria própria] pag 44
Teste de funcionamento do protótipo [autoria própria] pag 45

Lista de Tabelas

Tabela dos cursos na UFABC em ambos os campi [1] pag 9
Tabela das áreas administrativas da UFABC [1] pag 11

Sumário

1. Introdução 9
   a) Descrição da Empresa 9
   b) Descrição dos objetivos do estágio 14

2. Atividades Desenvolvidas 15
   a) Planejamento das etapas a serem desenvolvidas 15
   b) Estudo dos instrumentos e softwares 16
   - Kinect 16
   - Impressora 3D 18
   - ReconstructMe 20
   - MeshLab 21
   - PyMesh 22
   - MeshMixer 23
   - Bluetooth 23
   - Arduino 24
   - Sensor de gás MQ - 135 25
   - Android e linguagem de programação Java 26
   c) Digitalização a partir do Kinect 27
   - K-Scan 27
   - Skanect 29
   - ReconstructMe 30
   d) Tratamento dos dados capturados 32
   - MeshLab 32
   - PyMesh 33
   - MeshMixer 34
   e) Impressão do objeto digitalizado 35
   - Slicer 35
   - Repetier 36

f) Estudo de emissão de gases 38
- ABS 38
- PLA 38

g) Montagem do protótipo de detecção de gás 39
- Programação do Arduino 41

h) Aplicativo de celular 42
- Tela de Login 42
- Tela de Cadastro 43
- Tela de Menu 43
- Tela de Armazenamento 44
- Tela de Histórico 44
- Tela de Ajuda 44

i) Testes do protótipo 45

3. Fundamentação 46
a) Relação de disciplinas da UFABC - BCC 46
b) Processamento da Informação 46
c) Programação Estruturada 46
d) Computação Gráfica 47
e) Programação de dispositivos móveis 47

4. Considerações finais 48

Referências 50

ANEXO
- Autorização de utilização de imagem 52
- Código em Python 53
- Código para a placa de desenvolvimento Arduino 54
- Código do aplicativo de celular no App Inventor 55

1. Introdução

Este relatório tem como objetivo apresentar o trabalho desenvolvido durante estágio desenvolvido nos laboratórios secos da UFABC referente a disciplina de Estágio Supervisionado III.

A seguir serão apresentados a empresa e os objetivos do estágio.

a) Descrição da Empresa

A UFABC é uma universidade federal fundada em 2006 composta por alunos docentes, técnicos administrativos, estagiários e funcionários terceirizados. Ao todo são aproximadamente 15 mil pessoas que circulam diariamente pela UFABC[1].

A Universidade realiza atividades voltadas tanto para a comunidade interna (docentes, dicentes, TAs, terceirizados) quanto para a comunidade externa (membros sem relação direta com a universidade mas que vivem no entorno).

São ministrados cursos direcionados para a área de exatas e de humanas. A tabela a seguir[1] indica quais cursos rodam em cada um dos campus existentes da instituição.

Campus	Modalidade	Curso	Matriculados	
			Matutino	Noturno
	BI	Bacharelado em Ciência e Tecnologia	2930	3310
		Bacharelado em Ciências Biológicas	38	61
		Bacharelado em Ciências da Computação	93	151
		Bacharelado em Física	30	45
		Bacharelado em Matemática	16	35
		Bacharelado em Química	64	84
		Engenharia Ambiental e Urbana	154	181

Santo André	Pós BC&T	Engenharia de Energia	101	208
		Engenharia de informação	51	174
		Engenharia de Instrumentação, Automação e Robótica	143	200
		Engenharia de Materiais	113	201
		Licenciatura em Química	28	49
		Licenciatura em Ciências Biológicas	20	36
		Licenciatura em Física	31	50
		Licenciatura em Matemática	37	50
Total SA Tabela cursos em SA e SBC			3.849	4.835
			8.684	
São Bernardo do Campo	BI	Bacharelado em Ciências e Humanidades	984	1269
		Bacharelado em Ciência e Tecnologia	986	1193
	Pós BC&T	Engenharia Aeroespacial	82	194
		Engenharia Biomédica	50	128
		Engenharia de Gestão	145	168
		Bacharelado em Neurociência	49	56
	Pós BC&H	Bacharelado em Ciências Econômicas	72	84
		Bacharelado em Filosofia	15	15
		Bacharelado em Planejamento Territorial	25	38
		Bacharelado em Políticas Públicas	40	74
		Bacharelado em Relações Internacionais	63	66
		Licenciatura em Filosofia	12	9

Total SBC	2.523	3.294
		5.817
Total UFABC	14.501*	

Tabela 1 - Tabela dos cursos na UFABC em ambos os campi [1]

A UFABC conta com uma infraestrutura que está em parte ainda em construção e outra parte já construída e em uso, da parte já construída estão inclusas salas de aula, laboratórios e áreas administrativas. A instituição é dividida em diversos setores, os que são preenchidos por técnicos administrativos estão listados na tabela abaixo[1]:

TÉCNICOS ADMINISTRATIVOS – POR ÁREA						
Área	Total	Nível Médio	Graduação	Especialização	Mestrado	Doutorado
Agência de Inovação	9	1	2	4	2	-
Assessoria de Comunicação e Imprensa	15	-	4	8	3	-
Assessoria de Relações Internacionais	8	-	3	3	2	-
Auditoria Interna	8	-	-	6	2	-
Biblioteca	34	4	12	15	3	-
CCNH	13	-	5	6	2	-
CECS	20	2	9	8	1	-
Cedidos	12	-	3	7	2	-
CGFC	16	2	5	9	-	-
CGSA	26	2	5	17	2	-
CMCC	13	1	3	8	1	-
Coord. Lab. Informática	18	3	11	2	2	-

Coord. Lab. Secos	65	12	31	10	12	-
Coord. Lab. Úmidos	39	7	11	6	12	3
Corregedoria	3	-	-	3	-	-
Editora da UFABC	2	-	2	-	-	-
Gabinete da reitoria	13	1	3	7	2	-
NTE	8	-	2	5	1	-
Núcleo de Tecnologia da Informação	71	13	26	25	7	-
Ouvidoria	2	-	1	1	-	-
Prefeitura Universitária	37	2	8	23	4	-
PROAD	6	1	1	4	-	-
PROAP	52	6	11	26	6	3
Procuradoria Jurídica	3	-	-	2	1	-
PROEC	23	-	7	12	3	1
PROGRAD	82	8	21	36	15	2
PROPES	36	4	12	12	6	2
PROPG	34	3	7	22	2	-
PROPLADI	14	-	-	10	4	-
Reitoria	-	-	-	-	-	-
Secretaria Geral	9	-	5	3	1	-
SUGEPE	35	1	9	21	5	-
Superintendência de obras	10	1	-	6	3	-
Vice reitoria	-	-	-	-	-	-

Total	736	74	218	327	106	11
Superintendência de obras	10	1	-	6	3	-
Vice reitoria	-	-	-	-	-	-
Total	736	74	218	327	106	11

Tabela 2 - Tabela das áreas administrativas da UFABC [1]

Dentre os setores citados acima, a PROGRAD a responsável pela área da graduação, ou seja, responsável por todos os assuntos que envolvem alunos e professores da graduação. Um subsetor da PROGRAD é o CLD (Coordenação dos Laboratórios Didáticos), que é dividido em laboratórios secos, úmidos e de informática. Cada um dos tipos de laboratório acomoda diferentes tipos de matéria da graduação.

A atuação se dá na Universidade como parte da PROGRAD, mais especificamente no subgrupo dos laboratórios secos, que é responsável por parte dos laboratórios da graduação, dentre eles os laboratórios de eletrônica, controle, modelagem, robótica, ótica, entre outros. A atuação ocorre principalmente no laboratório no qual rodam as disciplinas referentes a robótica.

Ultimamente alguns alunos e professores têm procurado soluções que utilizariam a impressora 3D para projetos relacionados à graduação. Alguns já com desenhos prontos para a impressão do modelo, porém outros apenas com ideias de projetos a serem impressos.

Com o objetivo de incluir uma nova ferramenta para alunos e professores nos projetos de graduação, aproveitando equipamentos existentes na universidade, surgiu a ideia de confeccionar um equipamento capaz de digitalizar objetos, pessoas ou estruturas um pouco maiores e imprimi-los numa impressora 3D.

b) Descrição dos objetivos do estágio

O estágio tem como objetivo o desenvolvimento de alguma atividade ou desenvolvimento de um projeto que traga benefícios para a comunidade acadêmica e que ao mesmo tempo aborde conceitos vistos no curso de Ciência da Computação.

A partir dos recursos presentes no Kinect será realizado a digitalização das peças a serem impressas. A leitura da digitalização será feita através do software ReconstructMe e posteriormente serão utilizados filtros do programa MashLab para editar o objeto digitalizado.

Com a intenção de abrir mais o leque de opções de ferramenta para alunos e professores, tanto no âmbito da graduação, pós-graduação e pesquisa, o projeto desenvolvido durante o estágio tem como objetivo o desenvolvimento de um conjunto capaz de realizar a digitalização de objetos, ou até mesmo pessoas, e se necessário a impressão 3D dos mesmos.

Durante o período de estágio pretende-se desenvolver as seguintes atividades:

- Estudar/pesquisar sobre os hardwares e possíveis softwares a serem utilizados;

- Realizar testes de impressão na impressora 3D, aprender a ajustar parâmetros pertinentes a impressora;

- Testar softwares capazes de realizar a digitalização dos objetos a partir do Kinect;

- Realizar testes de digitalização e verificar qual dos softwares apresenta melhor resultado ao aparelho Kinect;

- Após realizar a digitalização, testar softwares de edição de arquivos 3D capazes de corrigir possíveis erros ocorridos durante a etapa de digitalização;

- Aprender a ajustar os parâmetros para as peças digitalizadas e corrigidas no software;

- Verificar quais etapas podem ser automatizadas a partir de um aplicativo e simplifica-las para as demais pessoas que vierem a utilizar os equipamentos.

2. Atividades desenvolvidas

Nesta seção serão apresentadas atividades a serem realizadas durante o período de estágio supervisionado.

a) Planejamento das etapas a serem desenvolvidas

Inicialmente foi feito um planejamento mais amplo das atividades:

- Digitalizar objetos através do sensor Kinect;

- Imprimir o objeto digitalizado na impressora 3D;

- Criar um aplicativo para a automatização do processo.

Esses eram os tópicos que guiariam as atividades a serem desenvolvidas durante o estágio. Porém, conforme o andamento do projeto foram surgindo questões que não haviam sido levantadas anteriormente. A ideia principal se manteve, porém alguns sub-tópicos foram adicionados posteriormente:

- Digitalizar objetos através do sensor Kinect

- Quais softwares utilizar para realizar a digitalização?;
- Qual o hardware necessário para realizar esse tipo de processamento de imagem?;
- Como realizar a captura da nuvem de pontos de modo a obter a melhor imagem possível?;
- Como realizar ajustes das imagens obtidas?;
- Qual software utilizar para ajuste das imagens?;
- Qual software utilizar para preparar o arquivo para a impressão 3D?;

- Quais configurações utilizar na impressora?;
- Qual o tipo de filamento é melhor para esse tipo de impressão?;
- Qual resolução utilizar para otimizar o tempo sem perder qualidade?;
- Quais processos podem ser automatizados?;
- Quais processos seriam possíveis/viáveis de serem controlados através de um aplicativo?

Essas perguntas passaram a ser a base para a realização do estágio, com o objetivo de tentar sanar o maior número de dúvidas possível.

b) Estudo dos Instrumentos e softwares

Os principais instrumentos utilizados são a impressora 3D e o digitalizador Kinect, enquanto que os softwares são o ReconstructMe, Mash Lab, e Mesh Mixer.

- **Kinect**

O Kinect[2][3] é um equipamento que possui aproximadamente 25cm de comprimento e possui 5 recursos principais, que são: câmera RGB, sensor de profundidade (infravermelho), microfone embutido, processador próprio e a capacidade de detectar 48 pontos de articulação.

Trata-se de um dispositivo originalmente desenvolvido para ser conectado ao console do XBox 360 e Xbox One para utilizar em jogos em que são feitas capturas de movimento de jogadores, tornando possível que os mesmos interajam com o jogo de forma mais intuitiva do que quando comparada a interação com um controle.

O produto é um sensor de movimentos desenvolvido pela Microsoft (desenvolvedora do XBox 360 e XBox One) e a empresa Prime Sense que começou a ser comercializado em Novembro de 2010 e teve sua produção descontinuada em

Outubro de 2017. Capaz de capturar imagens 3D por conta do infravermelho e do processador presente nele, o Kinect precisa estar a uma distância mínima de 60 cm do objeto para conseguir detectá-lo devido a ajustes do fabricante, além da capacidade de processamento do mesmo.

Hoje em dia existem empresas buscando a digitalização de objetos 3D como uma opção de tecnologia a ser empregada na confecção de projetos, de modo a proporcionar um diferencial ao produto entregue ao cliente [4]. Sensores com maior resolução de imagem e maior taxa de processamento que o Kinect são capazes de realizar a digitalização com grande precisão, exibindo vários pequenos detalhes e texturas de objetos. No caso do projeto, as imagens podem não ter uma boa resolução ou acabamento devido a qualidade mediana do sensor Kinect, porém, como uma tentativa preliminar, com o objetivo de entender o funcionamento da digitalização 3D, o hardware é o suficiente como objeto de estudo.

Figura 1 - Foto Kinect [autoria própria]

- **Impressora 3D**

A impressora 3D é um equipamento capaz de imprimir peças através da deposição de camadas de material. Atualmente existem impressoras 3D capazes de

imprimir em diversos materiais como por exemplo cimento, para a construção de casas[5] ou até mesmo de materiais/tecidos vivos (ou remédios) para uso na área da saúde[6][7]. Porém o material mais comumente utilizado para impressão são os derivados do plástico (PLA ou ABS).

Para o projeto será utilizada uma impressora 3D e o material de impressão serão derivados de plástico (ABS ou PLA). A impressora 3D existente no laboratório deve ser ligada na rede elétrica (110V) e é composta principalmente por:

- Bico extrusor: Aquece para derreter o material plástico e depositar o material sobre a mesa. Tem altura fixa e se movimenta nos eixos x e y (plano da mesa) (Figura 2);

- Mesa ou bandeja: Também aquece com o intuito de manter a peça fixa durante a impressão, evitando que o material se desprenda da mesa e a impressão passe a ficar comprometida. Além disso ela se movimenta no eixo z (para cima e para baixo) e é fixa nos eixos x e y (Figura 2). A mesa tem dimensão de 21cm por 21cm.

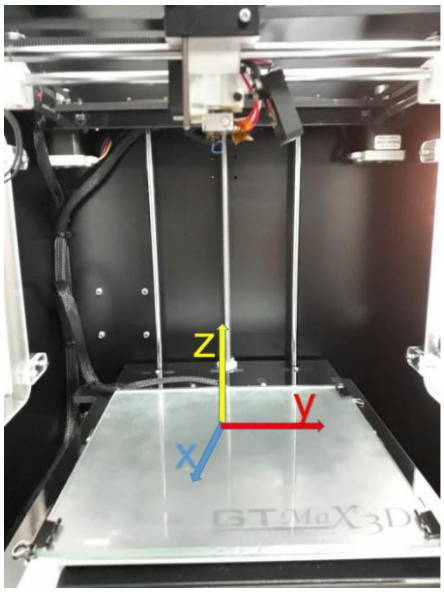


Figura 2 - Foto da mesa de impressão com eixos desenhados [autoria própria]

- Porta material e engrenagens da extrusora: Na parte traseira da impressora o rolo de filamento apoia-se num suporte estando livre para girar e soltar filamento conforme as engrenagens da extrusora puxam o material que será derretido no bico extrusor (Figura 3).

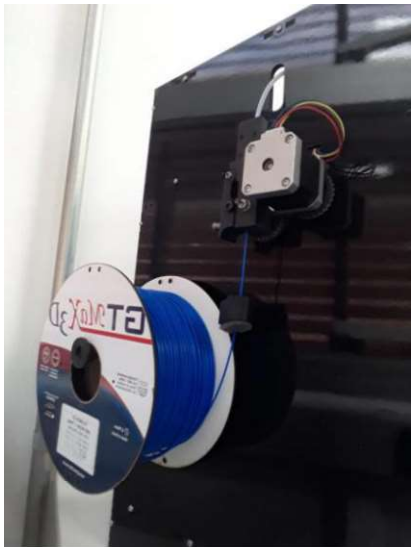


Figura 3 - Foto da parte traseira da impressora [autoria própria]

-Display e Arduino: Algumas funções como o ajuste de calibração da impressora, aquecimento ou resfriamento do bico extrusor ou da mesa podem ser acessadas através do display, controlado por uma placa Arduino, que fica na parte frontal da impressora. Esse display, e em conjunto a placa Arduino, podem ser ligados através de um cabo USB no computador. Porém as funções que dependem do acionamento de motores ou aquecimento de peças só passarão a funcionar no momento que a impressora for ligada na tomada. No caso, a placa Arduino fica disposta no interior da impressora.

Figura 4 - Foto display da impressora (ligado via cabo USB) [autoria própria]

A empresa fabricante da impressora a ser utilizada recomenda alguns programas *open source* para realizar o fatiamento da peça a ser impressa (Slicer) e a comunicação com o Arduino que irá comandar o funcionamento dos motores (Repetier).

- **ReconstructMe**

O ReconstructMe[8] é um sistema de digitalização 3D em tempo real. Utilizado para modelar objetos 3D foi feito para digitalizar desde objetos pequenos a pessoas, ou até mesmo cômodos inteiros.

O software é livre e é responsável pela interface do driver do computador com as diferentes versões do Kinect, capaz de capturar uma nuvem de pontos ao redor do objeto e processá-la em tempo real, sendo possível verificar se o processo de digitalização está sendo executado com sucesso.

Após gerar uma nuvem de pontos ao redor do objeto, ao finalizar a digitalização, o programa gera um arquivo com extensão .ply e mostra na tela o resultado do processo.

Figura 5 - Captura da tela com demonstração do ReconstructMe (TAKE A TOUR) [8]

- **MeshLab**

O MeshLab[9] é um software *open source* capaz de realizar processamento de malhas triangulares 3D. Um programa com várias ferramentas para edição de malhas 3D.

No projeto o programa será utilizado para receber o arquivo .ply, que será criado pelo ReconstructMe. A malha recebe uma limpeza, que serve para retirar partes digitalizadas indesejadas, como um objeto ao fundo por exemplo, além disso podem ser aplicados filtros de acordo com a necessidade. Após essas edições básicas o modelo é convertido para uma extensão .stl, pois a maioria dos softwares que trabalham com impressão 3D lêem esse formato de arquivo.

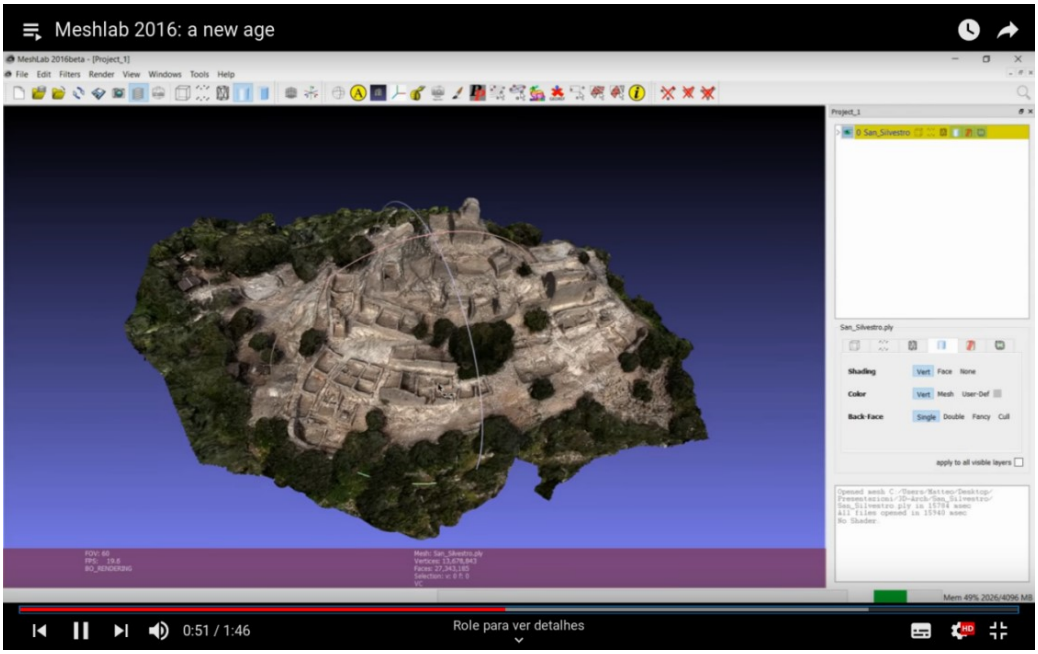


Figura 6 - Captura da tela com demonstração do MeshLab[10]

- **PyMesh**

PyMesh[11] é uma biblioteca do Python que possui diversas funções que permitem o modelamento de objetos 3D. Dentre as possibilidades oferecidas por essa biblioteca, serão utilizadas principalmente funções de limpeza e redimensionamento de malhas.

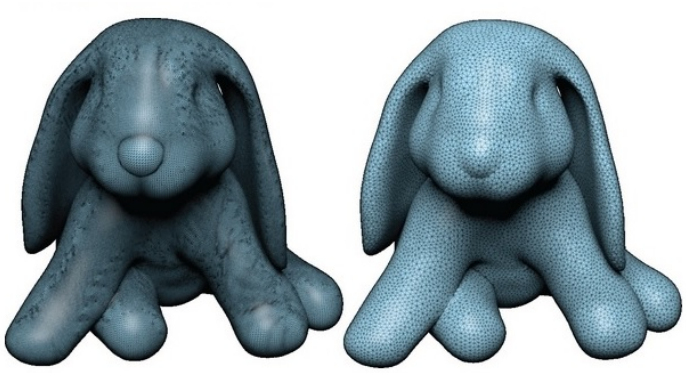


Figura 7 - Exemplo de aplicação de filtros do PyMesh[11]

- **Meshmixer**

O Meshmixer[12] é outros software livre capaz de processar malhas 3D além de possuir várias ferramentas para edição. Possui tantas funções quanto o MeshLab, porém foi escolhido este software de edição por afinidade na utilização e de sentir que ele é um programa mais fácil de se utilizar do que o MeshLab.

Nele serão feitos retoques na malha de modo a já deixá-la preparada para a impressão 3D. Serão ajustados pequenos defeitos que podem ser decorrentes do processo de digitalização, além de dimensionar o tamanho do objeto final com o tamanho da mesa de impressão. Também é o responsável por criar as bases de suporte para partes de peças que iniciam sua impressão suspensas.

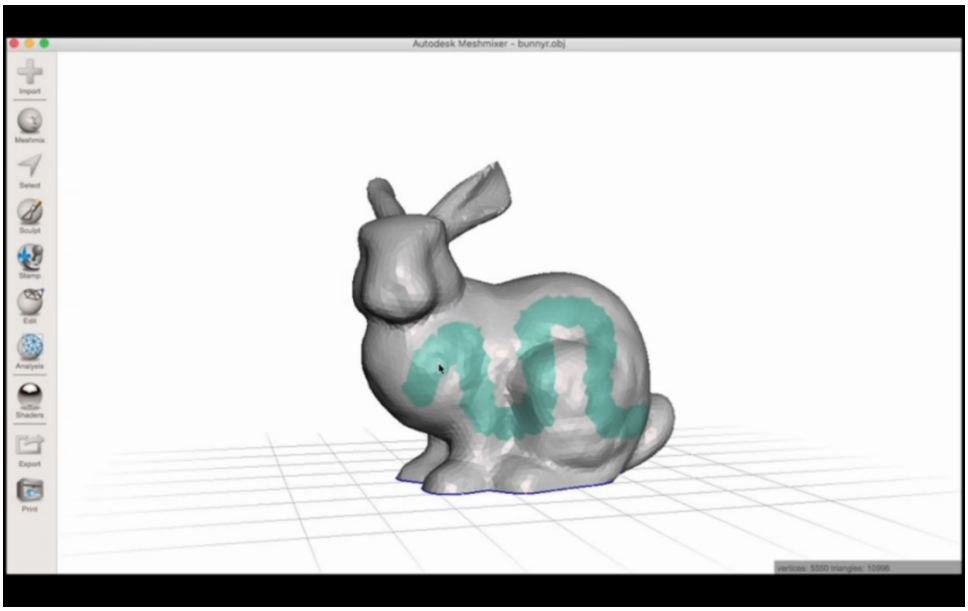


Figura 8 - Captura da tela com demonstração do Meshmixer [13]

- **Bluetooth**

O Bluetooth[14] é uma tecnologia que foi desenvolvida por um conjunto de empresas, cujo objetivo era integrá-las através dos mais diversos dispositivos que existiam no mercado na época. Trata-se de uma tecnologia sem fio que possibilita a comunicação de dispositivos, como celulares, notebooks e tablets, a curto alcance. O

alcance do sinal depende da potência do dispositivo, podendo chegar a 100 metros de distância.

Na conexão entre os dispositivos existe um mestre responsável por comandar a conexão, que pode receber ou enviar informações, e um escravo, que pode apenas receber informações do mestre. Este tipo de conexão é chamada de piconet. Na conecção entre diversos dispositivos um deles é o mestre enquanto os demais que desejarem participar da conecção se tornarão escravos.

Para realizar a comunicação entre dispositivos é utilizada uma técnica que efetua mudanças muito rápidas de frequência, sendo assim para que seja possível a comunicação entre dispositivos, todos devem estar em sintonia com a mudança de frequência.

No projeto será utilizado o módulo de Bluetooth HC-05 mostrado na Figura 9. Este dispositivo será utilizado com a intenção de ser uma extensão da placa Arduino, pois a placa em si não é capaz de realizar a transmissão e a recepção do sinal, este dispositivo será o responsável por realizar a conecção via Bluetooth com o aparelho celular.

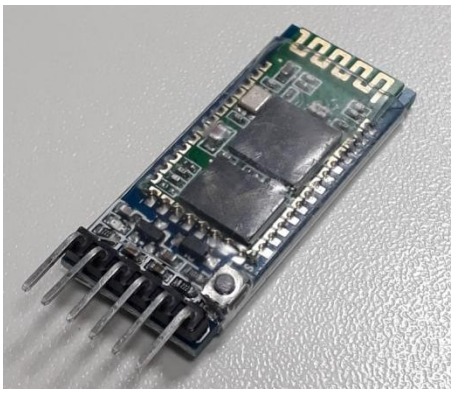


Figura 9 – Módulo de Bluetooth para comunicação[autoria própria]

- **Arduino**

Arduino[14] é composto por driver e softwares próprios projetados para o desenvolvimento de protótipos.

As placas Arduino existentes são fáceis de serem utilizadas, capazes de ler sinais provenientes de botões, sensores, entre outras fontes de entrada. Além disso também são capazes de emitir diferentes formas de saída, como acender um LED ou transmitir sinais via Bluetooth.

Figura 10 – Placa de desenvolvimento Arduino UNO [autoria própria]

Para que a placa de desenvolvimento saiba quais instruções deverá executar é necessário enviar um programa escrito em linguagem própria. O programa é enviado através da IDE do Arduino, onde o programa é escrito, para a placa através de um cabo USB conectado ao computador. O microcontrolador indicado com “A” na Figura 10 é o principal componente da placa, pois é responsável por processar o programa e controlar todas as funções existentes da placa. No projeto será utilizado uma placa Arduino Uno.

- **Sensor de gás MQ-135**

O sensor de gás MQ-135[15] é capaz de detectar alguns gases tóxicos, como óxido nítrico, amônia, benzeno além de dióxido de carbono, fumaça ou álcool.

O valor de medição indicado pelo sensor é uma relação entre duas resistências, a resistência que varia de acordo com a concentração dos gases e a resistência do sensor a 100ppm de NH_3 em ar puro[15].

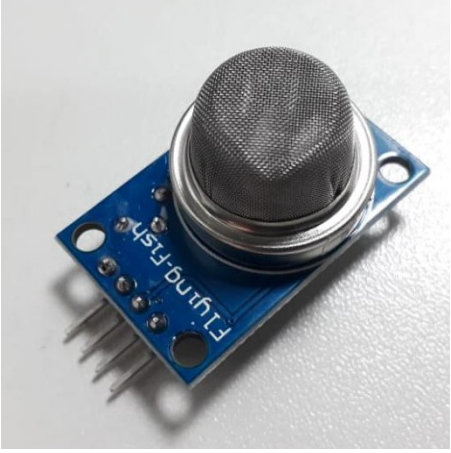


Figura 11 - Sensor de gás [autoria própria]

No projeto foi utilizado o sensor já acoplado a um módulo de modo a facilitar a conecção do sensor com a placa de desenvolvimento Arduino. Neste módulo além do circuito que facilitar a conecção do sensor à placa também existe um trimpot responsável pela regulagem da sensibilidade do sensor.

- **Android e linguagem de programação Java**

O Android [14] é um sistema operacional desenvolvido para aparelhos móveis, tais como celulares e tablets. É resultado do trabalho conjunto da Google e de outras empresas que tem como comprometimento trazer ao mercado produtos melhores e abertos, ou seja, que qualquer pessoa pode ter acesso ao código.

Vários aplicativos são desenvolvidos para o Android através de programas que se utilizam da linguagem de programação Java.

Java é uma tecnologia amplamente utilizada tanto por empresas quanto por pessoas físicas. Desenvolvida pela Oracle, com o objetivo de desenvolver softwares de aplicativos e jogos, é a linguagem de programação utilizada para o desenvolvimento dos aplicativos de Android.

O MIT App Inventor é um software de desenvolvimento de aplicativos para aparelhos celulares. A plataforma possui sua IDE (Figura 12) dividida em duas partes, a montagem do design e a programação. A programação é feita em blocos, ou seja,

arrastando blocos de comando para a tela, este tem como objetivo substituir as linhas de código.

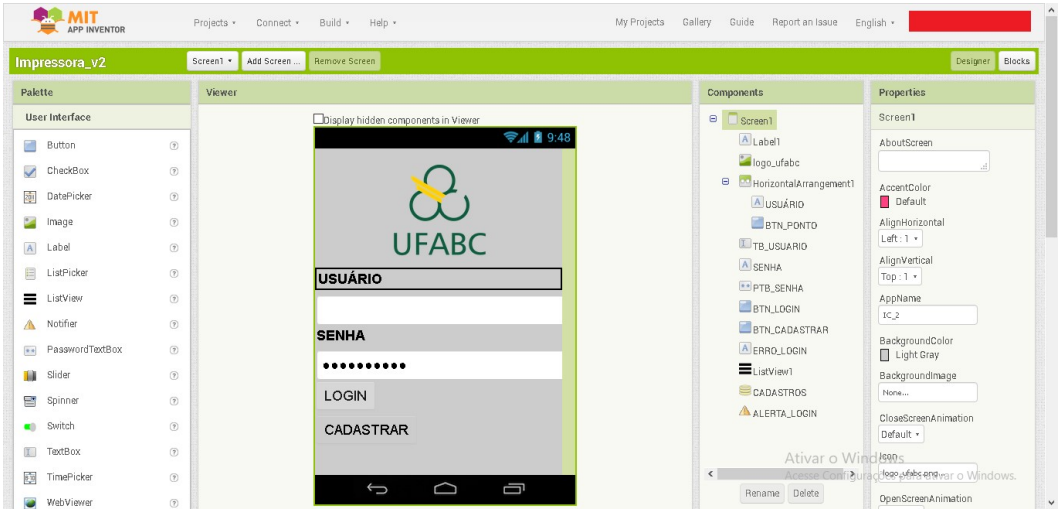


Figura 12 – IDE do MIT App Inventor[figura de autoria própria, site: ai2.appinventor.mit.edu]

O software não precisa ser instalado, para utilizá-lo é necessário ter uma conta no Google e entrar no site do MIT App Inventor (ai2.appinventor.mit.edu) para começar a utilizar o programa.

c) Digitalização a partir do Kinect

Antes de realizar a digitalização, foram testados diversos programas capazes de gerar uma malha de objetos 3D. Foram testados os seguintes programas:

- K-Scan - Programa compatível com o sensor Kinect. Funciona tirando várias fotos do objeto 3D e depois gera um objeto com as imagens capturadas. Problema: Não é muito prático, pois são necessárias muitas fotos do objeto sendo rotacionado 360°de pelo menos uns 2 ou 3 ângulos diferentes. No caso da simulação resultou em 151 fotos.

  Além disso não é garantido que o resultado da junção das imagens terá um bom resultado. Com objetos menos detalhados, como por exemplo um capacete (que possui menos ‘pontas’ e espaços vazios no meio) a chance de conseguir uma boa

digitalização é maior. Já para objetos com maior quantidade de detalhes, como foi testado com o mini robô da Figura 13, por possuir um espaço entre os braços, o mesmo teve boa imagens individuais, porém, quando as imagens foram sobrepostas e renderizadas pelo programa muitas partes (da parte do braço superior) acabaram sem preenchimento.

Figura 13 - Foto do mini robô [autoria própria]

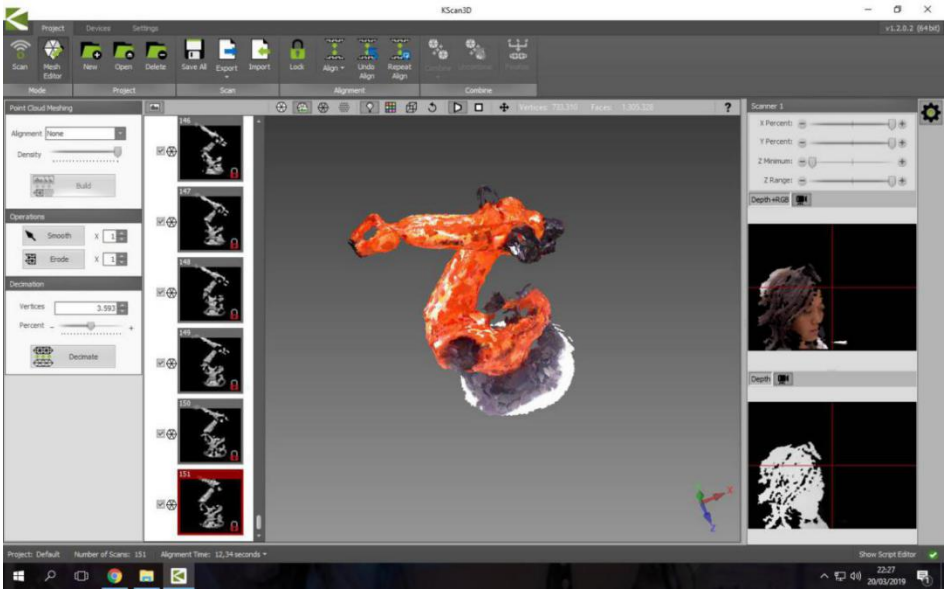


Figura 14 - Captura da tela com o resultado da sobreposição e renderização das imagens no KScan3D [autoria própria]

Por ser pouco prático (por conta das várias fotos que devem ser tiradas) e não ter apresentado um resultado muito satisfatório no teste, foi preferível procurar outro programa que pudesse realizar a digitalização mais facilmente.

- Skanect - A segunda tentativa foi melhor sucedida, pois foi utilizado um programa que realiza a digitalização em tempo real. Ou seja, já digitaliza e monta a malha de pontos obtida no programa.

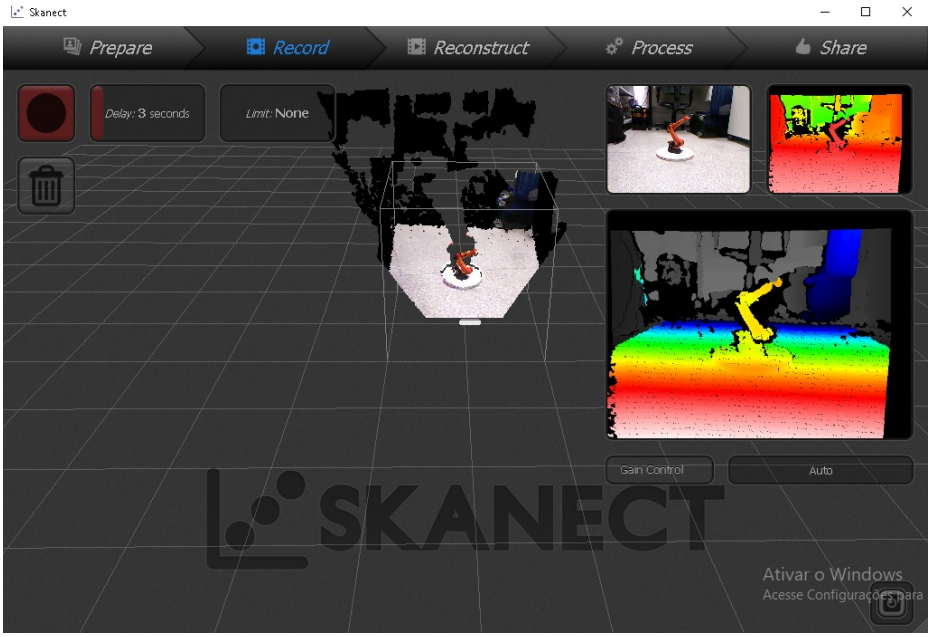


Figura 15 - Captura da tela do Skanect com o mini robô sendo digitalizado [autoria própria]

Um dos primeiros problemas encontrados foi que para realizar a digitalização em tempo real da maneira que este software faria, seria necessária uma placa de vídeo com maior capacidade de processamento. Até o momento estava sendo usado um computador com uma placa de vídeo bem simples, que assim que se conectava ao sensor e mostrava a imagem que estava sendo captada pelo mesmo, já começava a mostrar sinal de dificuldade no processamento da imagem. Quando se iniciava a captura da nuvem de pontos o computador parava de funcionar de vez.

Como forma de tentar resolver esse problema, foi testado realizar a digitalização em um computador com uma placa de vídeo um pouco melhor e com processadores melhores. Dessa forma foi possível realizar a digitalização, porém ainda com alguma dificuldade.

Outro problema desse programa é que o mesmo não nativo no Windows, portanto, após algumas digitalizações de objetos, foi acusado memória de processamento

cheia, impossibilitando a obtenção de novas imagens. Além disso o programa possibilita salvar apenas o conjunto de imagens geradas na versão gratuita do software, sendo possível salvar a malha apenas na versão paga.

Constatando tais problemas, por orientação do Professor Dr. Mario Gazziro, que já vem trabalhando com este tipo de sensor para projetos de pesquisa, fui recomendada a utilizar o programa ReconstructMe, um programa nativo do Windows e que teria melhor compatibilidade e apresentaria menos problemas de memória ou conexões. Além de auxiliar com ideias, o professor também auxiliou com o empréstimo de uma placa de vídeo melhor que já possibilitaria uma digitalização dos objetos sem que o computador apresentasse sinais de lentidão.

- ReconstructMe - Programa recomendado pelo professor Dr. Mario Gazziro, necessita de alguns requisitos mínimos especificados no site [8], que caso não sejam seguidos o programa nem abrirá.

  Após a instalação da placa de vídeo e dos drivers necessários no computador, foi possível fazer o programa do ReconstrucMe rodar.

  Nesta etapa já foram obtidas imagens satisfatórias sem que o computador travasse. Após alguns testes e alguns objetos 3D obtidos foi possível verificar que o programa do ReconstructMe é mais prático e fácil de se utilizar para realizar a captura da nuvem de pontos que formará as imagens. O arquivo é salvo automaticamente numa extensão .ply.

Figura 16 - Captura da tela do ReconstructMe com o mini robô digitalizado [autoria própria]

Um detalhe notado ao realizar testes de digitalização com o Kinect, foi que este sensor não tem uma capacidade muito boa de formação de malhas triangulares, que é o que permite uma boa definição do 3D do objeto. Ou seja, não é capaz de capturar com precisão pequenos detalhes de algumas peças ou até mesmo capturar o 3D de objetos muito pequenos. Isso é compreensível, uma vez que o sensor foi feito com o objetivo de capturar movimentos não necessariamente muito precisos de pessoas em jogos.

Porém, ao serem realizados testes com objetos maiores ou até mesmo pessoas, é perceptível que o resultado da digitalização tem uma distorção proporcionalmente menor. Ou seja, apesar dos pequenos detalhes não aparecerem, já é possível traçar uma malha de pontos razoável e sem muitas falhas.

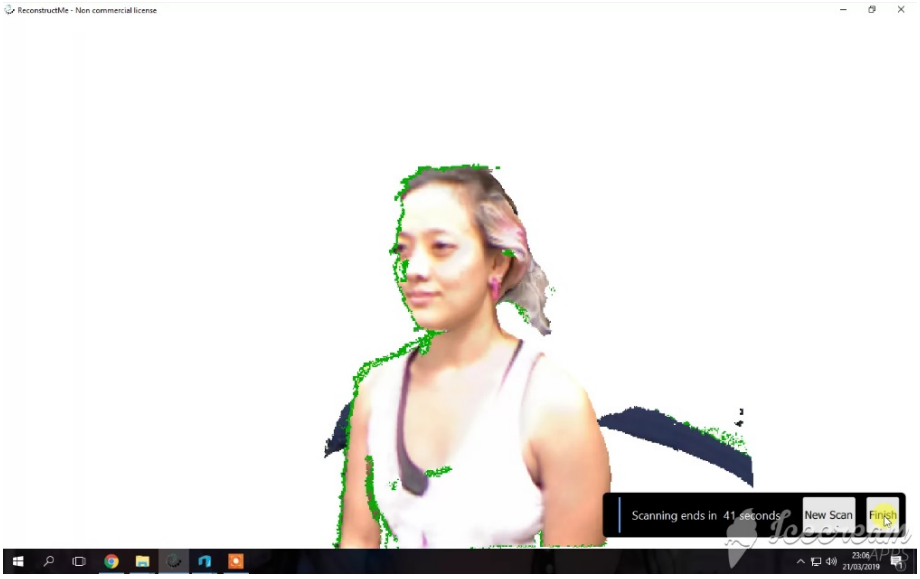


Figura 17 - Captura da tela durante a digitalização de uma pessoa [autoria própria]

d) Tratamento dos dados capturados

- MeshLab

Após a captura das imagens com o sensor Kinect, o programa ReconstructMe gera uma imagem a partir de uma nuvem de pontos para a construção do objeto. O arquivo gerado pelo ReconstructMe possui a extensão .ply, um tipo de extensão incomum e que nem todos os programas de edição 3D são capazes de abrir. Para isso utilizaremos o MeshLab com o objetivo de abrir e converter o arquivo para um tipo mais usual e que pode ser lido pelo programa da impressora 3D.

Além da conversão, o MeshLab também será utilizado para realizar a “limpeza” do objeto, retirando pedaços que foram digitalizados mas que não fazem parte do objeto. Por exemplo, no caso da Figura 14, foi digitalizada uma pessoa, porém, da digitalização bruta percebe-se que existem alguns pontos ou regiões que estão flutuando ou que não fazem parte do objeto alvo. Para a impressão 3D é desejável que apenas o objeto central apareça, dessa forma é realizado apenas a seleção do mesmo.

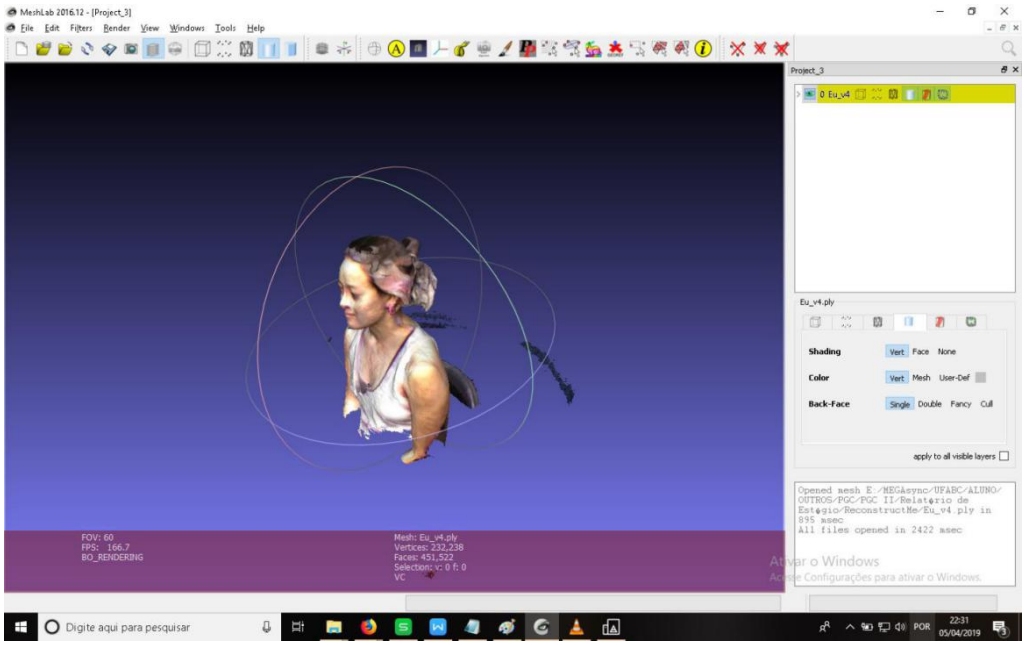


Figura 18 - Digitalização do objeto no MeshLab antes da limpeza [autoria própria]

Após a limpeza os arquivos serão convertidos para a extensão .stl pelo MeshLab, uma extensão que possibilita a edição do objeto além de ser compatível com o programa da impressora 3D.

- PyMesh

Realizada a conversão, uma segunda etapa de limpeza mais sutil do objeto é realizada através de uma biblioteca de modelagem do objetos 3D do Python, o PyMesh. Além de realizar uma nova limpeza, retirando possíveis sujeiras que foram digitalizadas erroneamente e que ainda tenham ficado flutuando ao redor do objeto, a biblioteca do PyMesh possui algumas ferramentas para correção e edição de malhas, dentre elas foram utilizadas algumas que realizam um redimensionamento da malha, fazendo com que o detalhamento ficasse maior ou menor. Aplicando de forma iterativa esses dois redimensionamentos o objeto final acaba mais uniforme.

Um outro exemplo de filtro aplicado que auxilia no momento da impressão é o que retira faces duplicadas, evitando que cause rugosidades inesperadas na peça digitalizada. O código completo em Python encontra-se em anexo deste relatório.

- MeshMixer

Após a conversão, por questão de escolha pessoal, foi escolhido o editor MeshMixer para realizar as edições no objeto. Este programa possui ferramentas avançadas para a finalização de modelos 3D, ou seja, realiza edições já preparando o modelo para a impressão.

Uma das etapas a ser realizada no Meshmixer é a aplicação de um filtro que irá analisar a existência de buracos na estrutura. Caso seja identificado algum buraco na estrutura o programa já possui um corretor capaz de fecha-los. Outro filtro também útil para ser aplicado antes da impressão é o responsável por suavizar quinas, ou seja, torna o objeto mais contínuo, que transforma a peça num formato que facilita a impressão 3D, de forma que a mesma terá menos problemas e falhas no decorrer da impressão.

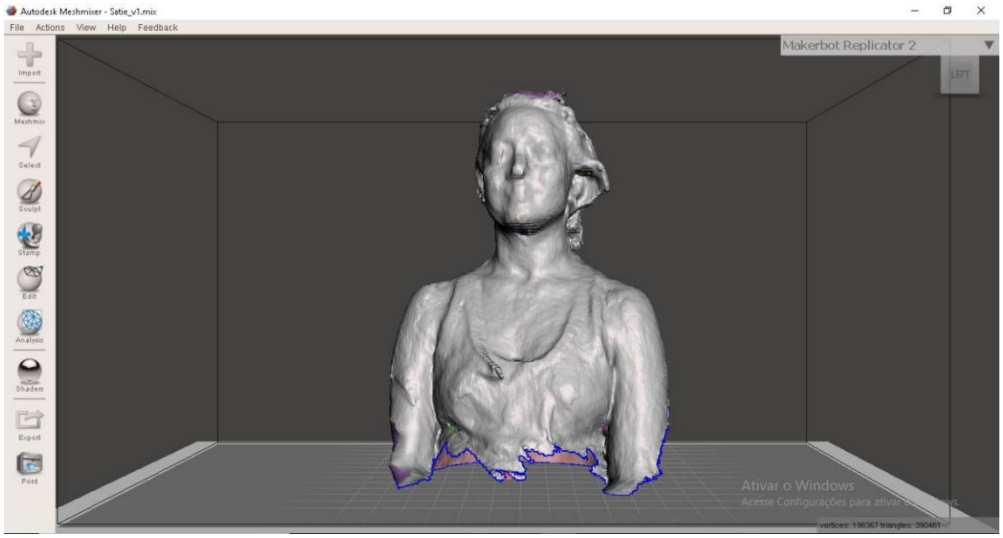


Figura 19 - Captura da tela do Meshmixer antes da aplicação de filtros [autoria própria]

Após a aplicação dos filtros desejados são construídos os suportes do objeto. Os suportes são pequenas hastes que serão adicionadas à impressão do objeto mas que poderão ser destacadas posteriormente. São geradas com o intuito de dar suporte às partes da peça que teriam o início de sua impressão no ar, e sem um apoio as mesmas podem resultar numa impressão ruim.

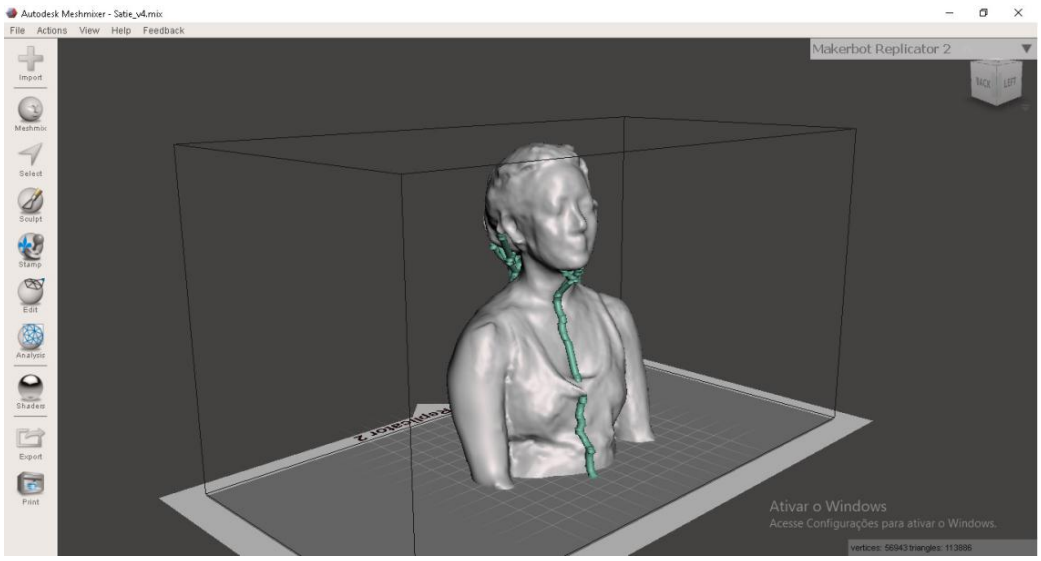


Figura 20 - Captura da tela do MeshMixer: objeto com suporte pronto para impressão [autoria própria]

e) Impressão do objeto digitalizado

Após realizadas as etapas de digitalização, leitura, processamento e filtragem do objeto, é gerado um arquivo legível pelos programas que realizam a comunicação com a impressora.

Os programas recomendados pela empresa fabricante da impressora 3D são *open source*. São eles o Slicer e o Repetier. Eles foram instalados e configurados seguindo as orientações do fabricante da impressora.

- Slicer - Programa responsável pelo fatiamento da peça a ser impressa. Este programa construirá um tipo de mapa a ser seguido pelo bico extrusor, indicando o local onde deverá ser feita a deposição de material e a quantidade de material a ser depositada por camada.

  Os parâmetros de ajuste de quantidade de material são definidos neste mesmo programa. Também é possível realizar a alteração de vários outros parâmetros, como o tipo de material a ser utilizado, o tamanho da base do objeto, a temperatura que deve estar o bico extrusor e a mesa, entre vários outros parâmetros ajustáveis.

- Repetier - O programa simula a mesa da impressora, sendo possível configurar as dimensões da mesma de acordo com seu tamanho. Este programa deve ser conectado a impressora 3D, tal processo é feito conectando o cabo USB no computador e na placa Arduino existente dentro da impressora. Além disso, no programa Repetier existe um botão de conectar. Um aviso de sucesso de conexão aparecerá na tela após clicar no botão, caso contrário deve ser verificado se o cabo está bem encaixado ou se não estão abertos mais de uma janela do Repetier ao mesmo tempo tentando conectar ao módulo Arduino.

  Neste programa é possível utilizar modelos prontos ou importar arquivos .stl para serem impressos. Também é possível dimensionar os objetos a serem impressos, inserir mais um objeto na mesa de impressão, realizar espelhamento ou rotação dos objetos (entre outras aplicações de filtros), entre outras operação.

  O objeto a ser digitalizado foi importado no formato .stl e foram feitos ajustes no objeto de acordo com o tamanho da mesa da impressora.

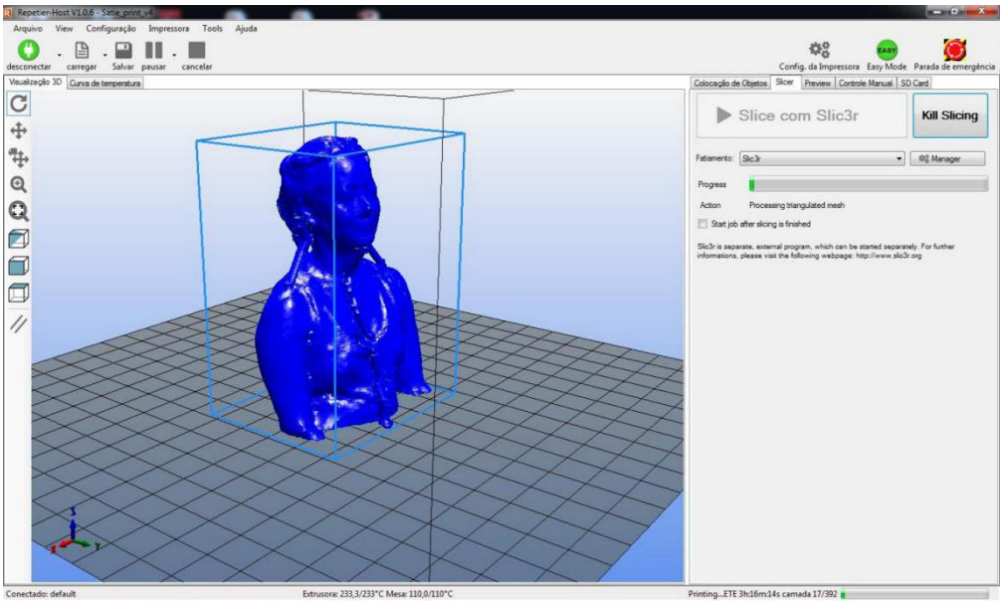


Figura 21 - Captura da tela do Repetier com objeto pronto para impressão [autoria própria]

Após concluídas as alterações realizadas pelos filtros é hora de fatiar a peça. A opção do fatiamento aparecerá em uma das abas do lado direito da tela do

programa. Caso já tenham sido feitas todas as configurações necessárias, tanto para o Repetier quanto para o Slicer, aparecerá um botão indicando “Fatiar com Slicer”. Feito o fatiamento a peça aparece já pronta, indicando como será feita a impressão, podendo ser visualizada camada a camada.

Estando a impressora conectada e o fatiamento realizado com sucesso, basta clicar em iniciar impressão que se dará inicio à impressão do modelo.

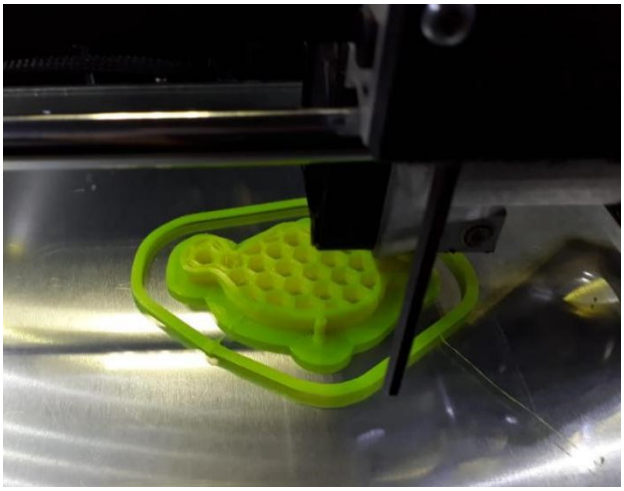


Figura 22 - Foto início da impressão [autoria própria]

Figura 23 - Fotos da impressão pronta [autoria própria]

f) Estudo de emissão de gases

Para a impressão de objetos 3D em materiais plásticos o processo mais comum é o de derretimento de filamento pela extrusora e deposição do material (ainda maleável) em camadas bem finas na mesa da impressora.

No decorrer do processo de derretimento e resolidificação do material, alguns gases são liberados dependendo da composição do plástico. Alguns estudos questionaram a segurança da impressão 3D[16] e realizaram análises térmicas de termoplásticos comumente utilizados para esse fim.

Dos materiais existentes no mercado utilizados para a impressão, estão disponíveis nos laboratórios da presente Instituição o ABS e o PLA. Ambos apresentaram o desprendimento de partículas ultrafinas de componentes orgânicos voláteis (VOC) durante o processo de aquecimento dos termoplásticos, seja em menor ou em maior quantidade.

- ABS

  O ABS (Acrilonitrila butadieno estireno)[17] é um material termoplástico leve que possui boa rigidez, resistente a absorção de impacto e pouco flexível. Um material altamente aplicado na indústria, tem sua composição levemente alterada para cada caso específico, ou seja, dependendo do segmento da indústria que será aplicado, as quantidades de acrilonitrila, butadieno e estireno sofrem leves alterações, de modo a atender as necessidades de cada um dos produtos.

  Ao ser submetido a temperaturas que podem variar de 220ºC a 240ºC o termoplástico começa a ficar maleável, sendo possível moldá-lo. Ao atingir essas temperaturas, o ABS libera um gás que segundo estudos [14] pode ser danoso a saúde se inalado em grandes quantidades.

- PLA

  O PLA (Poliácido Lático)[18] é também um material termoplástico e, diferentemente do ABS, é biodegradável, leve e resistente. Bastante maleável, utilizado para a

confecção de diversos utensílios de cozinha, como talheres, pratos e copos de plástico além de dispositivos médicos entre outras coisas mais.

Caso seja condicionado corretamente, o material pode ser biodegradado completamente, caso contrário, em condições anaeróbias ocorre a emissão de metano. O termoplástico formado principalmente por cadeias de ácido lático começa a ser moldável ao atingir temperturas de aproximadamente 180ºC.

O principal poluente liberado pelo PLA é o metilmetacrilato (cerca de 40 - 60% de todo VOC emitido). No caso do ABS o VOC liberado em maior quantidade é o Estireno (mais de 50%), mas segundo estudos também existe a formação de CO (monóxido de carbono) e HCN (ácido cianídrico).

Como apenas mais recentemente a impressora 3D passou a ser um instrumento mais acessível, ainda não existem muitos dados sobre possíveis efeitos colaterais causados pela inalação de gases liberados pelos termoplásticos. Com o intuito de analisar alguns aspectos da liberação de gases da impressora 3D, decidiu-se desenvolver um protótipo capaz de analisar o gás liberado, tanto em quantidade quanto em qualidade. Para isso será utilizado uma placa de desenvolvimento Arduino Uno, um sensor de gás, entre outros componentes. Como opção foi escolhido utilizar um módulo Bluetooth do Arduino para que os dados gerados através dessa análise sejam transmitidos para o celular e armazenados para comparações futuras.

Para que o celular receba os dados, será desenvolvido um aplicativo para aparelhos Android através do ambiente do MIT App Inventor. O celular Android que estiver com o aplicativo será capaz de receber os dados da placa de desenvolvimento Arduino através de um sinal de Bluetooth.

g) Montagem do protótipo de detecção de gás

Para a montagem do protótipo será utilizado uma placa de desenvolvimento Arduino, juntamente com um sensor de gás e o módulo Bluetooth. Tais componentes serão conectados seguindo o esquema da Figura 24.

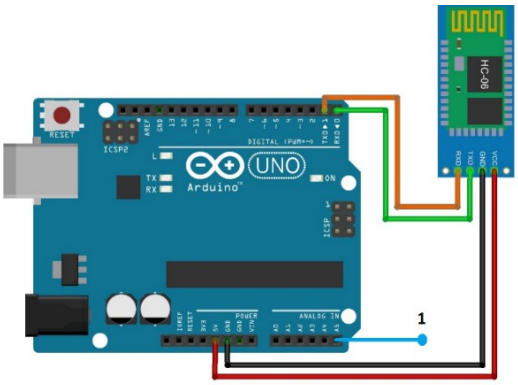


Figura 24 – Esquema de montagem do protótipo [autoria própria]

O Arduino receberá o sinal do sensor de gás através do ponto 1 (Figura 24) e repassará o mesmo para o módulo de Bluetooth nele acoplado. O módulo de Bluetooth acoplado transmitirá esse sinal para um aparelho celular que contenha o sistema operacional Android com o aplicativo instalado.

Para auxiliar na montagem e apoio da estrutura será montado em cima de uma protoboard. Após feitos testes, caso seja necessário, pode se pensar futuramente na confecção de uma placa de circuito impresso ou apenas uma base de madeira para auxiliar nas conecções e apoio da estrutura.

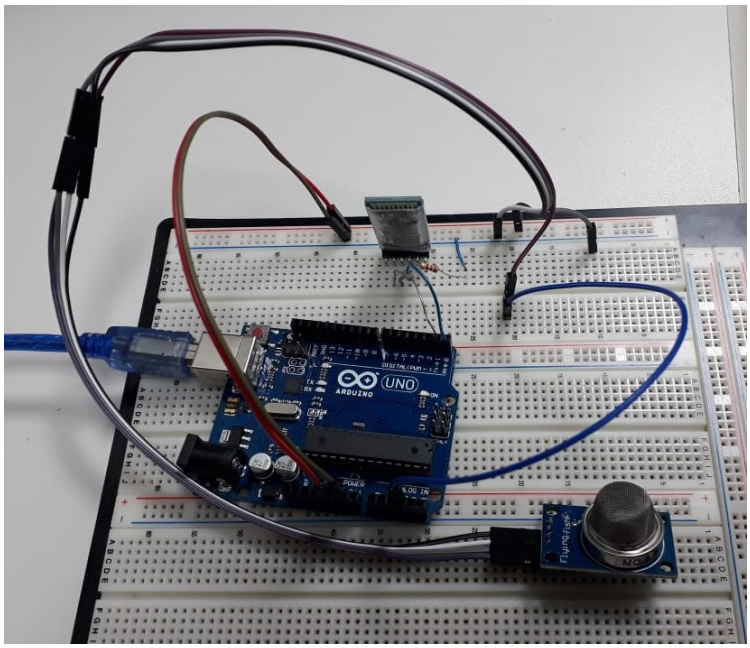


Figura 25 - Protótipo montado [autoria própria]

Com relação ao sensor, de acordo com os sensores disponíveis no laboratório foi escolhido utilizar o sensor MQ-135, sensor destinado a realizar a detecção de gases tóxicos, tais como benzeno, amônia, dioxido de carbono, óxido nítrico além de fumaça e álcool.

Outra opção que pode ser testada posteriormente seria o MQ-7, responsável pela detecção de monóxido de carbono. Este segundo sensor é muito utilizado para aplicações automotivas, dado que o monóxido de carbono pode ser letal para o ser humano em grandes quantidades. Porém, como os filamentos não liberam apenas monóxido de carbono, foi optado por se utilizar o sensor MQ-135.

- **Programação do Arduino**

O código desenvolvido para a placa de desenvolvimento Arduino será o responsável por captar o sinal adquirido pelo sensor de gás e transmitir o mesmo para o aparelho celular Android que tiver o aplicativo instalado através do módulo de Bluetooth. A captação do sinal será realizada constantemente, de acordo com a frequência estabelecida no programa.

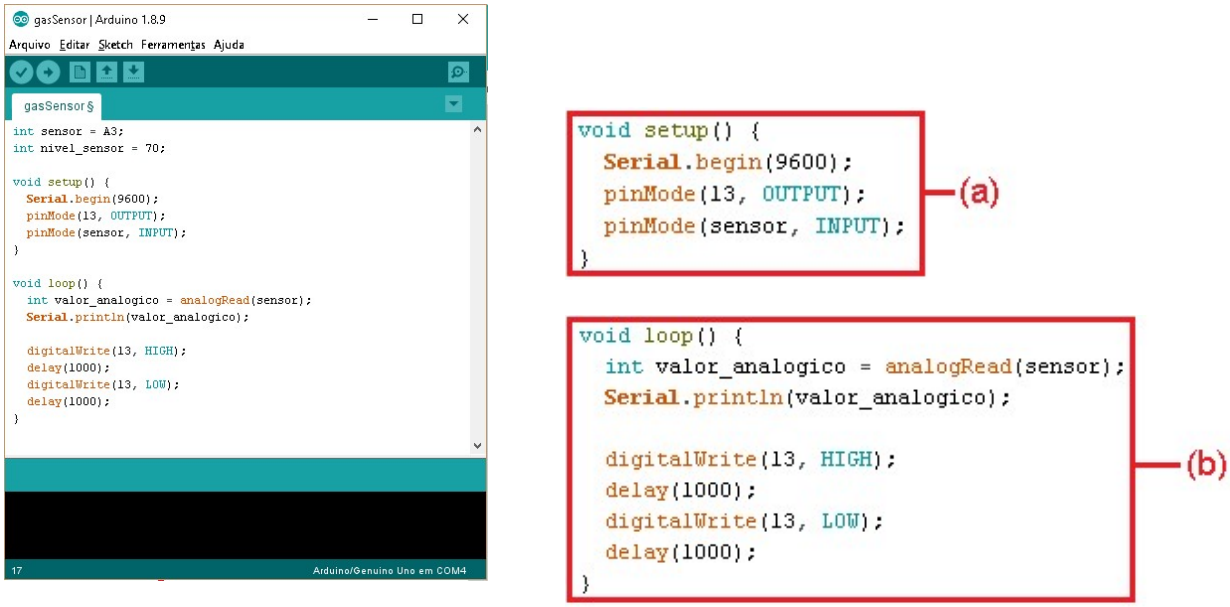


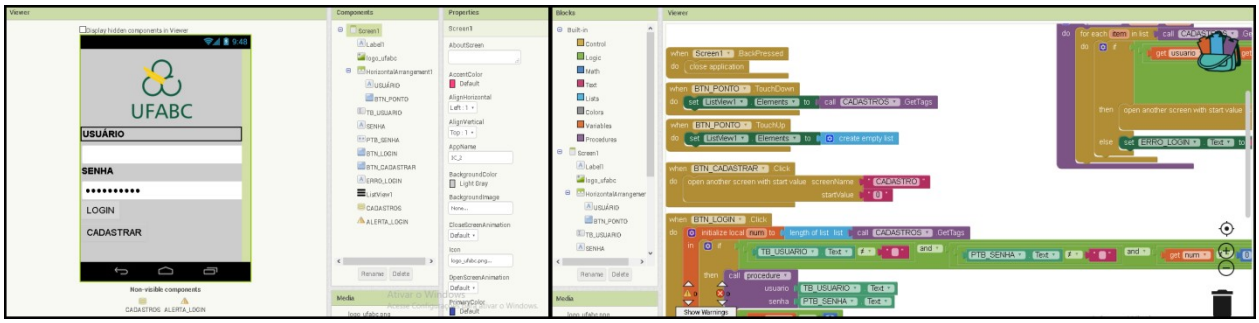
Figura 26 - IDE do Arduino [autoria própria]

O código é feito numa IDE (Ambiente de desenvolvimento de programa) própria e consiste na inicialização dos pinos a serem utilizados (Figura 26(a)). Logo em seguida é feita a programação que irá rodar na placa, realizando a leitura do sensor e enviando o sinal obtido para o módulo Bluetooth (Figura 26(b)).

O código completo utilizado para programar a placa Arduino encontra-se em anexo no final deste relatório.

h) Aplicativo de celular

Para o aplicativo de celular foi utilizado o ambiente de programação MIT App Inventor que possui uma IDE online. A programação é feita em blocos, além de possuir uma interface que mostra como fica a apresentação final da tela do celular.



(a) *(b)*

Figura 27 - IDE App Inventor (a) Interface (b) Programação em blocos [autoria própria]

- <u>Tela Login</u>

  O aplicativo conta com uma tela inicial de login de usuário, para caso seja necessário separar os dados de diferentes impressoras ou caso o usuário deseje separar as medições para diferentes materiais, temperaturas, etc. Caso necessário, é possível realizar um novo cadastro clicando no botão “CADASTRAR” e para acessar uma conta já criada basta colocar o usuário e senha cadastrados anteriormente e clicar no botão “LOGIN”.

- Tela Cadastro

Escolhendo a opção “CADASTRAR” o usuário chega a uma tela no qual é solicitado o nome, data de nascimento e dados para realizar o login, como usuário e senha. Para finalizar o cadastro o usuário deve selecionar o botão “CADASTRAR” na parte inferior da tela, ou selecionar “CANCELAR” caso desista de realizar o cadastro do usuário.

- Tela Menu

Entrando com um usuário e senha cadastrados, chega-se a tela “MENU”. Assim que esta tela é aberta uma janela pop-up é acionada caso o celular não esteja com o Bluetooth ligado, perguntado se o usuário não gostaria de ligar o mesmo, pois para receber o sinal é necessário que o aparelho esteja apto para isso.

Após ligado o Bluetooth, nesta mesma tela de “MENU” encontram-se alguns botões que possibilitam do usuário realizar uma das seguintes ações:

- “Realizar medição”, no qual o usuário é redirecionado para uma tela em que a medição do gás será feita e possivelmente armazenada caso o usuário assim deseje;

- “Histórico”, onde são armazenados os dados coletados, contendo a data e a hora no qual os dados foram armazenados;

- “Ajuda”, uma tela com instruções de como se conectar com o módulo Bluetooth e como utilizar o aplicativo é aberta para caso o usuário tenha dúvidas de como utilizar o mesmo.

- “Excluir cadastro”, caso o usuário deseje excluir o cadastro todos os dados coletados anteriormente nesta conta serão excluídos juntamente com a conta.

Ao selecionar a opção de “Realizar medição” será aberta uma tela indicando a informação referente ao gás juntamente com as opções de “Armazenar dado”, “Realizar nova medição” e “Cancelar”.

- Tela Armazenar

  - “Armazenar dado”, neste caso o dado indicado na tela será armazenado no histórico, juntamente com data e hora que a medição foi realizada.

  - “Realizar nova medição”, para caso a medição realizada esteja muito fora do esperado, seja por uma falha na medição, seja por que no momento da medição a concentração de gás esperada ainda não tenha sido atingida ou por qualquer outro motivo.

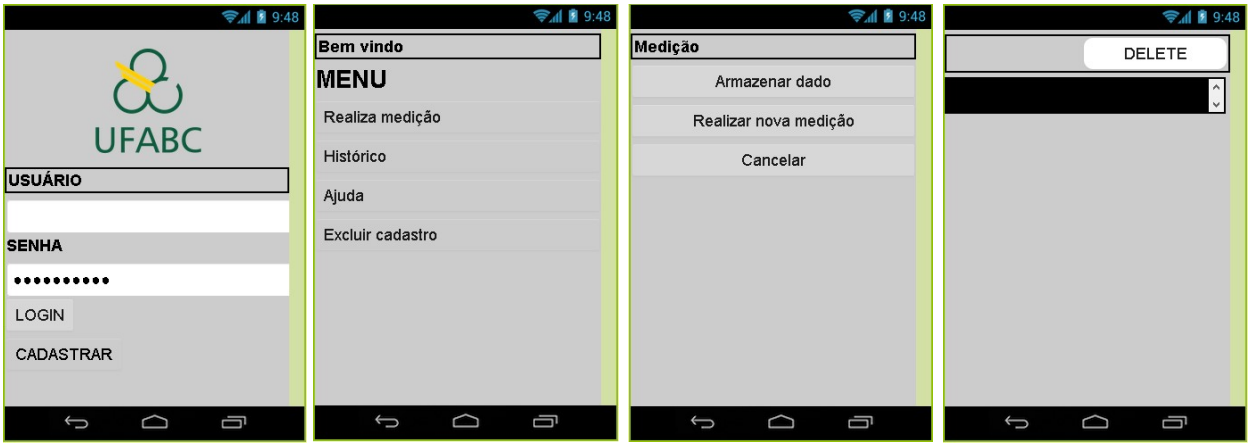
  - “Cancelar”, caso não deseje realizar uma nova medição nem adicionar o dado medido no histórico.

- Tela Histórico

  Caso seja selecionado a opção de verificar o histórico, as medição armazenadas estarão listadas. Caso seja de interesse do usuário é possível apagar as medições realizadas selecionando a mesma e apertando o botão “DELETE”.

- Tela Ajuda

  Será aberto uma tela com instruções de como conectar o aplicativo via Bluetooth.



(a) (b) (c) (d)

Figura 28 – Algumas das telas do aplicativo (a)Login (b)Menu (c)Armazenar dado (d)Histórico [autoria própria]

A programação em blocos do aplicativo se encontra no final do relatório em anexo.

i) Testes do protótipo

Foi realizado um teste do protótipo montado conectado a um aparelho celular, porém, como a impressora estava com alguns problemas para liberar material pela extrusora, o que consequentemente não liberaria o gás a ser analisado, e será retirada para manutenção em breve, preferiu-se testar o funcionamento do protótipo com a fumaça liberada por um ferro de solda.

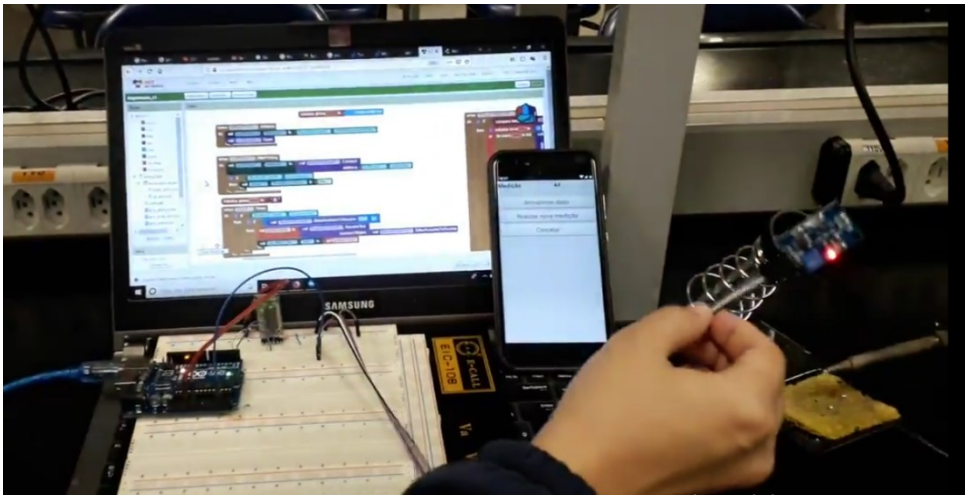


Figura 29 - Teste de funcionamento do protótipo [autoria própria]

A partir deste teste foi possível verificar que quando o sensor ficou mais exposto à fumaça liberada pelo ferro de solda a medição indicada no aparelho celular teve uma alteração mais abrupta.

Para facilitar a visualização do funcionamento do projeto foi feita uma gravação do funcionamento do conjunto ainda ênfase de testes. O link https://www.youtube.com/watch?v=fK9gj_pi0Os dá acesso ao vídeo.

Futuramente, assim que a impressora retornar ao laboratório, serão realizados mais alguns testes de detecção de gases liberados pela impressora. Caso seja interessante o protótipo poderá ser instalado na lateral da impressora ao lado do

computador, que já está conectado à impressora, que fornecerá energia para a placa e para o restante do circuito através de uma das portas USB. Enquanto isso apenas o sensor permanecerá na parte interna da cabine, o mais próximo possível da extrusora, local que o material será liberado e que a concentração dos gases possivelmente será maior.

3. Fundamentação

Nesta etapa são descritas as fundamentações teóricas e as disciplinas que correlacionam tais conceitos acadêmicos.

a) Relação de Disciplinas da UFABC - BCC

Muitas das disciplinas contribuíram para a realização de algumas das atividades realizadas no estágio. Na sequência são apresentadas as matérias que mais contribuíram com a atividade em foco apresentada neste relatório.

Processamento da Informação - lógica de programação
Programação estruturada - Estruturação do código
Computação Gráfica - Processamento de imagens.
Programação de Dispositivos Móveis - Aplicativo de celular

b) Processamento da Informação

A matéria é fundamental para qualquer tipo de programação, pois tem como principal objetivo o desenvolvimento da lógica de programação, desde programas mais simples até programas mais complexos e avançados.

No caso será aplicado no desenvolvimento do aplicativo que visa a automatização do processo de digitalização e impressão de objetos.

c) Programação Estruturada

Dando sequência a matéria de Processamento da Informação, é responsável por organizar em blocos funcionais e estruturados a lógica implementada, que facilita a

compreensão do código. Ou seja, a programação do aplicativo será feita de forma mais organizada, de modo que fique claro para qualquer pessoa, que venha alterar posteriormente o programa, entenda o mesmo e faça as modificações necessárias.

d) Computação Gráfica

Compreende o processamento de imagens realizado entre os processos de digitalização e impressão do objeto.

Utilização de filtros, que nada mais são do que técnicas numéricas aplicadas a cada pixel da imagem, modificando configurações das mesmas com intuito de melhorar a apresentação final da mesma para o usuário.

No caso essas técnicas serão utilizadas não para imagens 2D, mas sim para objetos 3D. Estes serão “lapidados” através de uma filtragem na nuvem de pontos digitalizada para posteriormente gerar o objeto num formato de arquivo legível por softwares de desenho gráfico e modelagem e pela impressora 3D.

e) Programação de Dispositivos Móveis

Desenvolvimento e criação de aplicações para dispositivos móveis além de aprender sobre os recursos mais comuns de serem utilizados neste tipo de programação.

Tais técnicas foram utilizadas para o desenvolvimento do aplicativo responsável por receber e armazenar os dados recebido pela placa de desenvolvimento Arduino por intermédio do módulo Bluetooth.

4. Considerações finais

Inicialmente foram feitas as revisões bibliográficas, revisando softwares e equipamentos que seriam utilizados no decorrer do projeto.

Em seguida foi realizada a digitalização de um objeto 3D através do sensor Kinect juntamente com o software ReconstructMe. Foram realizas etapas de limpeza no objeto digitalizado, para isso foram utilizados o programa MeshLab e um código em python utilizando biblioteca especializada em modelos 3D, o PyMesh. Por ultimo, antes de realizar a impressão foram feitos suportes na peça, preparando-a para a mesma com o programa MeshMixer. Após ter o modelo pronto para impressão, foram ajustados os parâmetros da impressora e depois de algumas horas de deposição de material o objeto digitalizado e tratado estava pronto.

Nesta ultima etapa foi feito um estudo sobre os gases emitidos pelos filamentos no momento em que são aquecidos e se tais gases tem potencial de prejudicar a saúde humana em exposição demasiada.

Para realizar os testes com os gases foi montado um prototipo com a placa de desenvolvimento Arduino, um sensor de gás e um modulo Bluetooth responsável por enviar os dados adquiridos para um celular Android. Para receber o dado no aparelho celular foi desenvolvido um aplicativo capaz armazenar o mesmo, sendo possível realizar futuras consultas desses dados.

Foram realizados alguns testes iniciais com a fumaça emitida pelo derretimento do estanho para verificar o funcionamento do protótipo e do aplicativo para Android. Nesses testes foi possível verificar que a fumaça alterou a medição realizada pelo sensor, porém no momento não foi possível testar a emissão de tais gases de materiais utilizados para impressão por conta de da impressora estar apresentando defeito na extrusão de material.

Ainda assim não é possível dar certeza do efeito desses gases pois a impressora 3D ainda se trata de uma tecnologia que foi difundida recentemente. Além disso o sensor utilizado não e o mais preciso que existe, este serve apenas para ter uma noção da quantidade de gás emitido. Um estudo futuro, seria interessante realizar testes mais precisos, no qual seria necessária a utilização de equipamentos mais específicos capazes de realizar tal análise com precisão.

.

_____

Natália Satie Motokubo Halker

Aluno

_____

Edgard Gonçalves Cardoso

Chefia

_____

Fabrício Olivetti de França

Orientador da UFABC

REFERÊNCIAS

[1] UFABC. **UFABC em números**. [*S. l.*], 1 jul. 2018. Disponível em: http://propladi.ufabc.edu.br/informacoes/ufabc-em-numeros. Acesso em: 13 maio 2019.

[2] XBOX. **Configuração do Kinect no Xbox 360**. [*S. l.*]. Disponível em: https://support.xbox.com/pt-BR/xbox-360/accessories/kinect-sensor-setup. Acesso em: 13 maio 2019.

[3] KINECT. [*S. l.*], 27 abr. 2019. Disponível em: https://pt.wikipedia.org/wiki/Kinect. Acesso em: 13 maio 2019.

[4] SILVA, Fábio Pinto da *et al*. Design & Tecnologia. **A digitalização tridimensional móvel e sua aplicação no design de produto**, [*S. l.*], 2010. 01-2010 pgdesign 2010.

[5] FLORÊNCIO, Eduardo Quintella *et al*. Prospecção tecnológica. **Concreto para uso em impressora 3D e sua utilização na construçãode edificações: um estudo prospectivo**, Salvador, p. 478-589, 16 ago. 2017. Disponível em: https://rigs.ufba.br/index.php/nit/article/view/23203. Acesso em: 14 mar. 2019.

[6] FLORÊNCIO, Eduardo Quintella *et al*. Ciências médicas. **Impressão 3D: inovações no campo da medicina**, Belo Horizonte, 2017. Disponível em: http://revista.fcmmg.br/ojs/index.php/ricm/article/view/14. Acesso em: 15 mar. 2019.

[7] OLIVEIRA, Anila A. *et al*. Pesquisa veterinária brasileira. **Bioimpressão e produção de mini-orgãos com célula tronco**, Pirassununga, Setembro 2017. Disponível em: https://ainfo.cnptia.embrapa.br/digital/bitstream/item/167715/1/Bioimpressao-e-producao-de-mini-orgaos.pdf. Acesso em: 15 mar. 2019.

[8] RECONSTRUCTME. [*S. l.*]. Disponível em: http://reconstructme.net/. Acesso em: 13 maio 2019.

[9] MESHLAB. [*S. l.*]. Disponível em: http://www.meshlab.net/. Acesso em: 13 maio 2019.

[10] MESHLAB 2016: a new age. [*S. l.*], 23 jan. 2017. Disponível em: https://www.youtube.com/watch?v=l8yu_B0miXg&list=PL8B1E816EAE236B4D. Acesso em: 13 maio 2019.

[11] PYMESH. [*S. l.*]. Disponível em: https://pymesh.readthedocs.io/en/latest/. Acesso em: 13 maio 2019.

[12] AUTODESK Meshmixer. [*S. l.*]. Disponível em: http://www.meshmixer.com/. Acesso em: 13 maio 2019.

[13] AUTODESK Meshmixer 3: What's New. [*S. l.*], 20 jan. 2016. Disponível em: https://www.youtube.com/watch?v=PCgP8VLFaz4. Acesso em: 13 maio 2019.

[14] HALKER, Natália Satie Motokubo. **Estudo da Transmissão de Sinais Biomédicos de Forma Remota proveniente de sensor implantável**. 2016. Iniciação

Cinetífica (Engenharia de Instrumentação Automação e Robótica) - Graduação, Santo André, 2016.

[15] MQ-135 GAS SENSOR. [*S. l.*]. Disponível em: https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf. Acesso em: 13 maio 2019.

[16] WOJTYŁA, Szymon; KLAMA, Piotr; BARAN, Tomasz. Is 3D printing safe? Analysis of the thermal treatment of thermoplastics: ABS, PLA, PET, and nylon. **Journal of Occupational and Environmental Hygiene**, [*S. l.*], p. D80-D85, 4 maio 2017. Disponível em: https://www.tandfonline.com/doi/full/10.1080/15459624.2017.1285489. Acesso em: 10 maio 2019.

[17] EQUIPE MAIS POLIMEROS. **Plástico ABS e suas principais características e aplicações**. [*S. l.*], 24 set. 2018. Disponível em: http://www.maispolimeros.com.br/2018/09/24/plastico-abs-e-suas-principais-caracteristicas-e-aplicacoes/. Acesso em: 13 maio 2019.

[18] EQUIPE ECYCLE. **Plástico PLA: alternativa biodegradável e compostável**. [*S. l.*]. Disponível em: https://www.ecycle.com.br/component/content/article/37-tecnologia-a-favor/738-pla-o-plastico-compostavel.html. Acesso em: 13 maio 2019.

ANEXO

Eu, Natália Satie Motokubo Halker, inscrita no CPF sob o nº 390.895.758-39 e no RG nº 47.749.426-2, através desta, autorizo a utilização da minha imagem para fins de pesquisa e educação.

_____

Natália Satie Motokubo Halker

Data: 11/04/2019

ANEXO

Código em Python.

```
import pymesh

def fix_mesh(mesh):
    target_len = 2.5e-3;
    count = 0;
    mesh, __ = pymesh.remove_degenerated_triangles(mesh, 100);
    mesh, __ = pymesh.split_long_edges(mesh, target_len);
    num_vertices = mesh.num_vertices;
    while True:
        mesh, __ = pymesh.collapse_short_edges(mesh, 1e-6);
        mesh, __ = pymesh.collapse_short_edges(mesh, target_len,
                preserve_feature=True);
        mesh, __ = pymesh.remove_obtuse_triangles(mesh, 150.0, 100);
        if mesh.num_vertices == num_vertices:
            break;

        num_vertices = mesh.num_vertices;
        count += 1;
        if count > 10: break;

    mesh = pymesh.resolve_self_intersection(mesh);
    mesh, __ = pymesh.remove_duplicated_faces(mesh);
    mesh = pymesh.compute_outer_hull(mesh);
    mesh, __ = pymesh.remove_duplicated_faces(mesh);
    mesh, __ = pymesh.remove_obtuse_triangles(mesh, 179.0, 5);
    mesh, __ = pymesh.remove_isolated_vertices(mesh);

return mesh;

def main():
    mesh = pymesh.load_mesh("model.stl");
    mesh = fix_mesh(mesh);
    pymesh.save_mesh("final_model.stl", mesh);

if __name__ == "__main__":
    main();
```

ANEXO

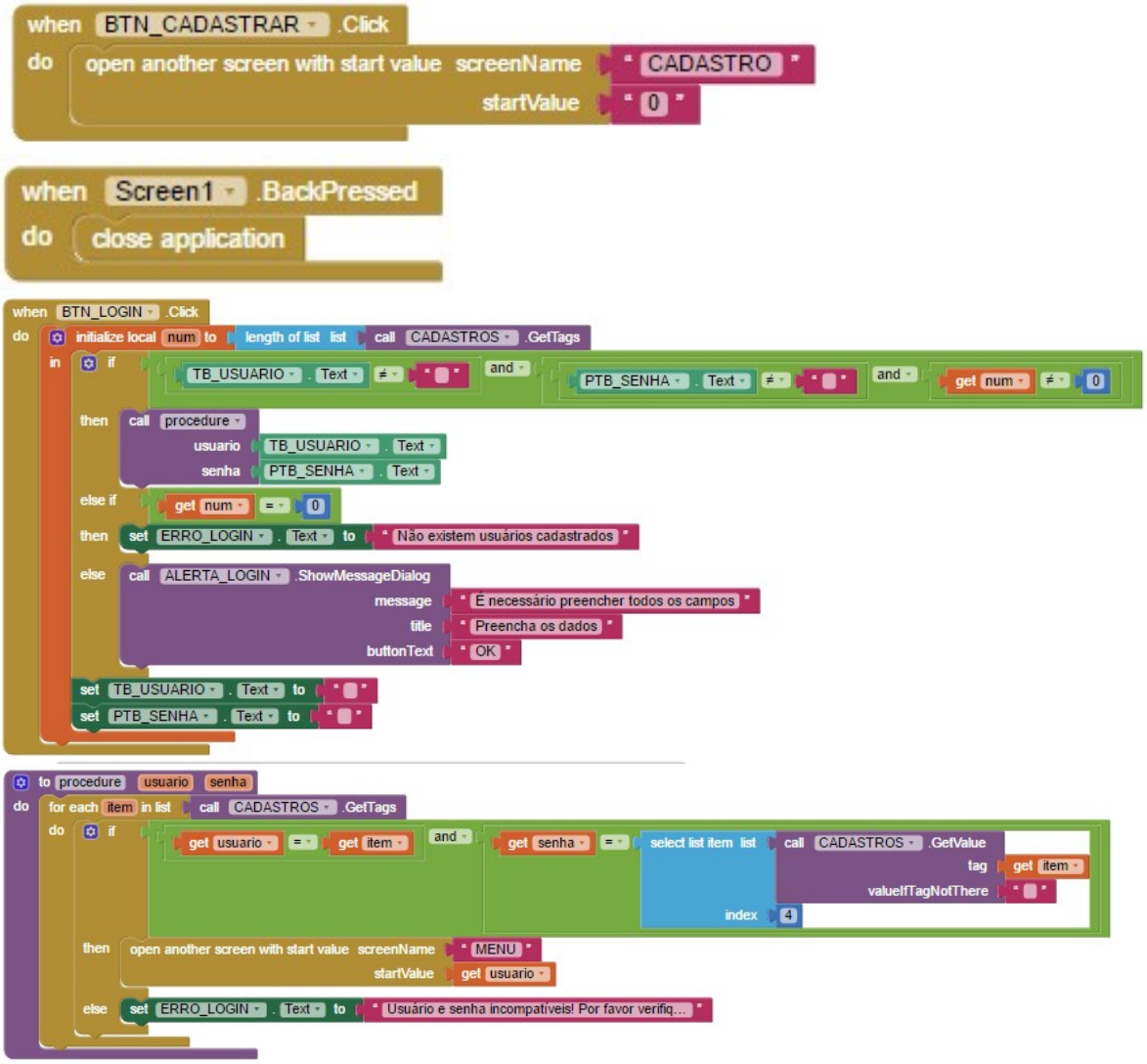
Código para a placa de desenvolvimento Arduino.

```
int sensor = A3;
int nivel_sensor = 70;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  pinMode(sensor, INPUT);

}

void loop() {
  int valor_analogico = analogRead(sensor);
  //Serial.print(" Pino A0 : ");
  Serial.println(valor_analogico);

  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);

}
```

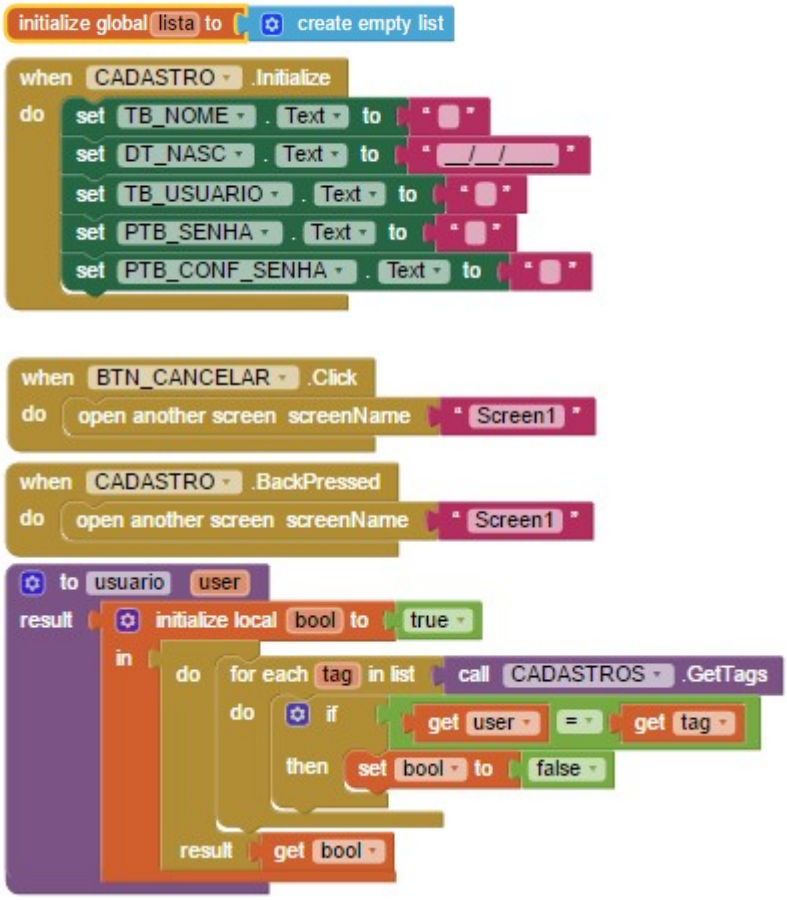
ANEXO

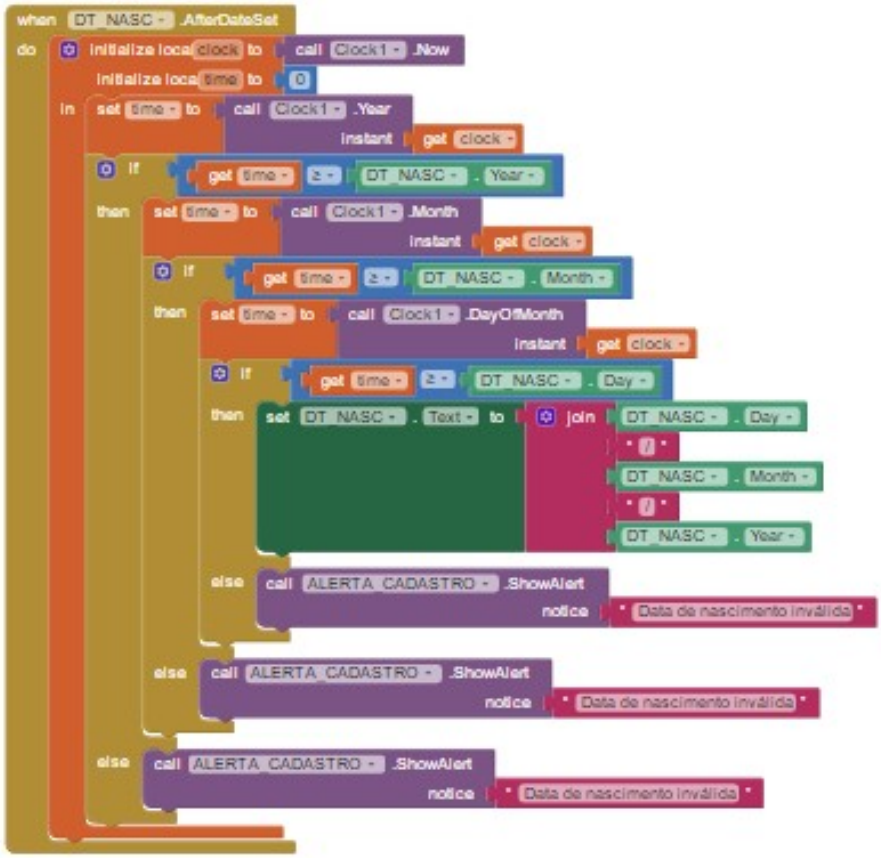
Código para o aplicativo no App Inventor.

Screen 1 (LOGIN)



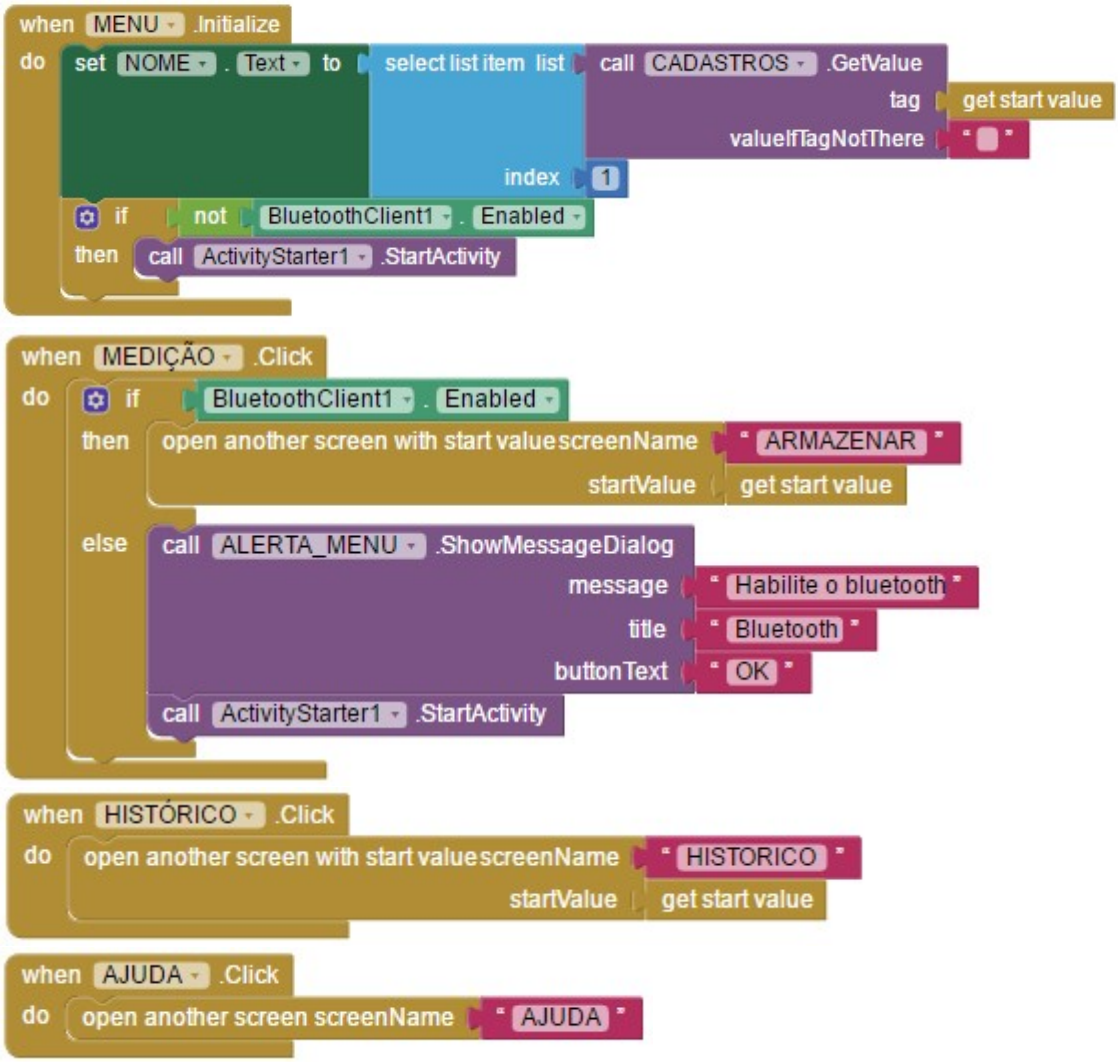
CADASTRO

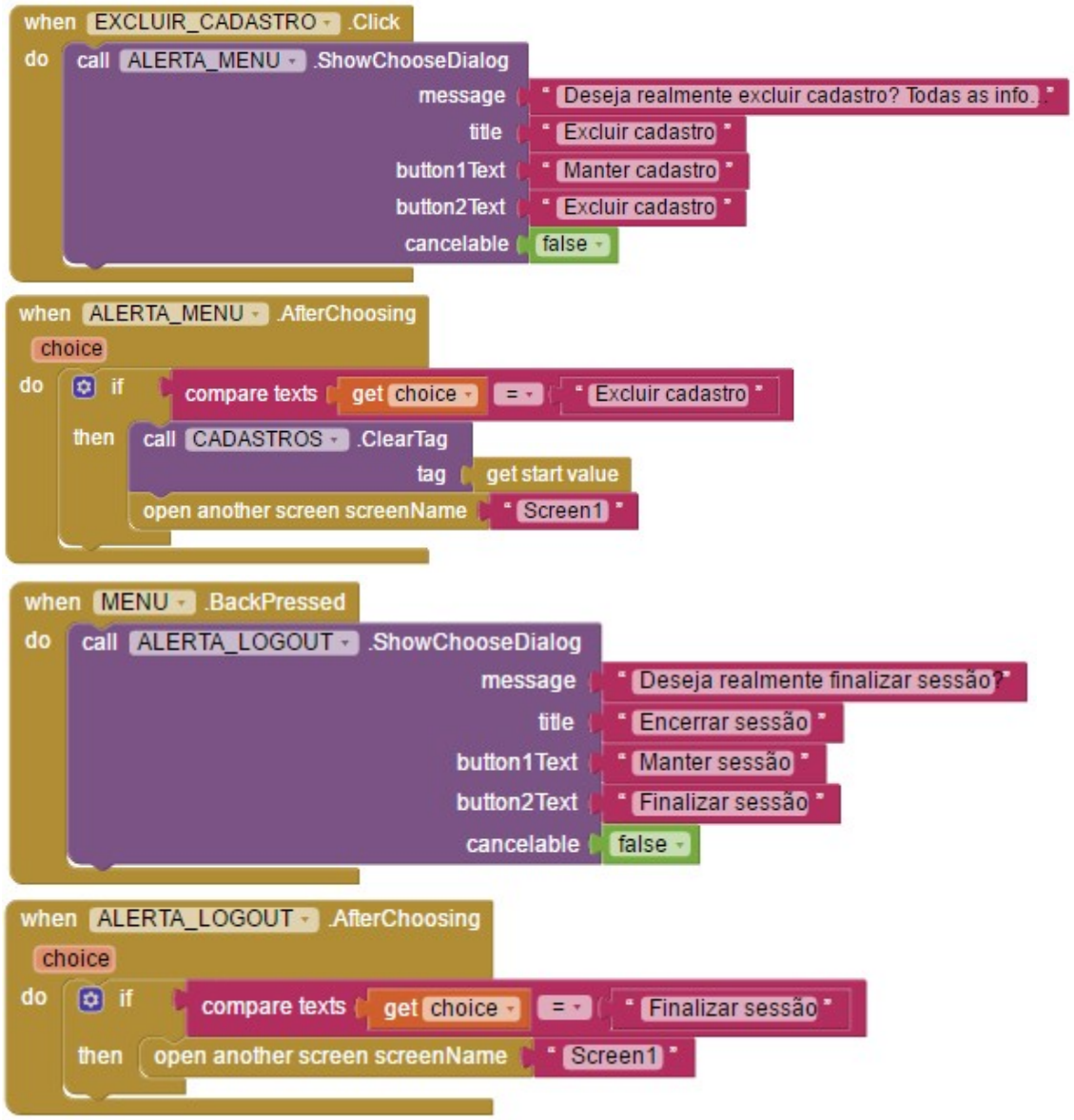

initialize global lista to
create empty list
when CADASTRO .Initialize
do
set TB_NOME . Text to
set DT_NASC . Text to
set TB_USUARIO . Text to
set PTB_SENHA . Text to
set PTB_CONF_SENHA . Text to
when BTN_CANCELAR .Click
do
open another screen screenName
Screen1
when CADASTRO .BackPressed
do
open another screen screenName
Screen1
to usuario user
result
initialize local bool to
true
in
do
for each tag in list
call CADASTROS .GetTags
do
if
get user
=
get tag
then
set bool to
false
result
get bool


when DT_NASC .AfterDateSet
do
initialize local clock to call Clock1 .Now
initialize local time to 0
in set time to call Clock1 .Year
instant get clock
if get time ≥ DT_NASC . Year
then set time to call Clock1 .Month
instant get clock
if get time ≥ DT_NASC . Month
then set time to call Clock1 .DayOfMonth
instant get clock
if get time ≥ DT_NASC . Day
then set DT_NASC . Text to join DT_NASC . Day
"/"
DT_NASC . Month
"/"
DT_NASC . Year
else call ALERTA_CADASTRO .ShowAlert
notice "Data de nascimento inválida"
else call ALERTA_CADASTRO .ShowAlert
notice "Data de nascimento inválida"
else call ALERTA_CADASTRO .ShowAlert
notice "Data de nascimento inválida"

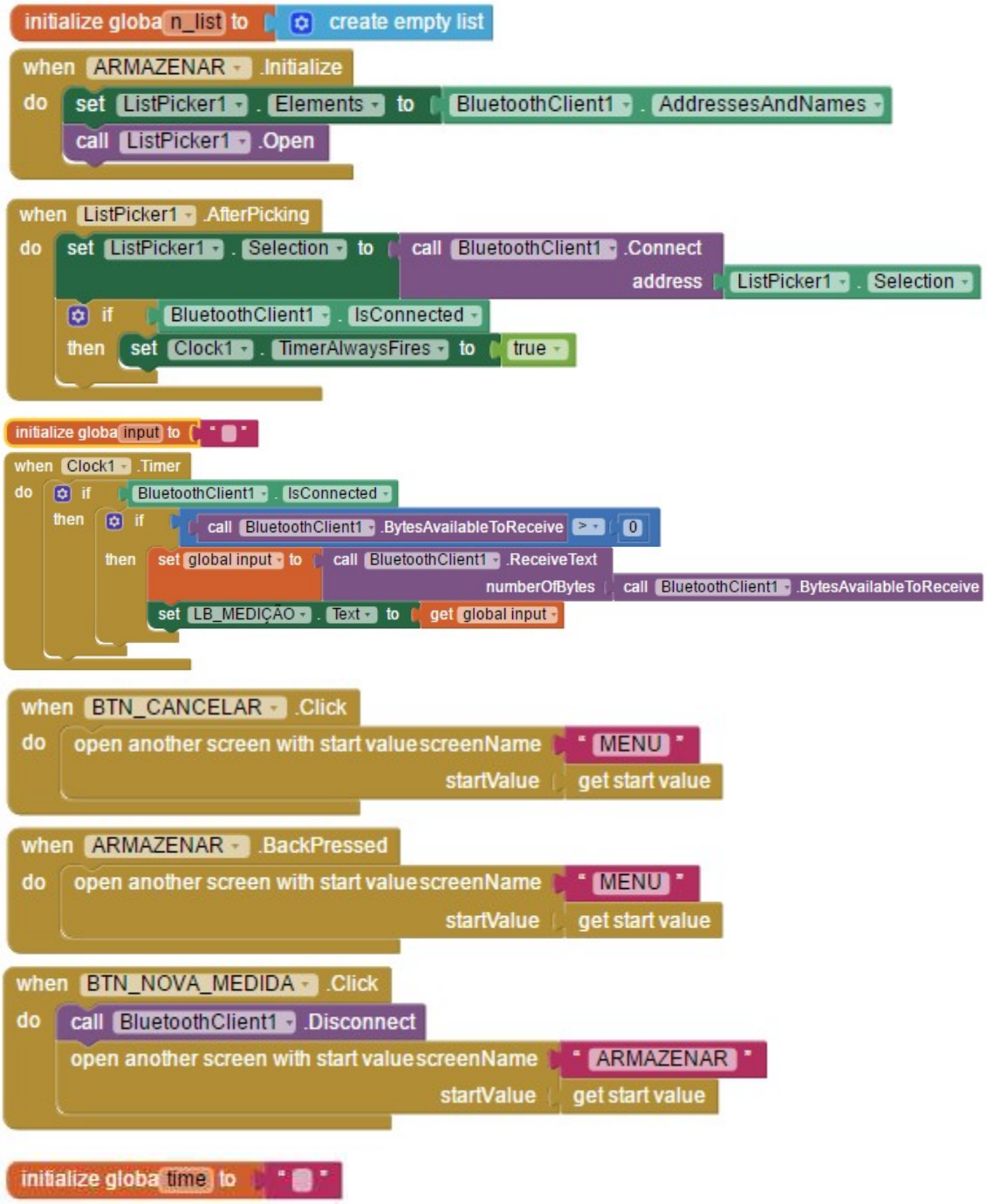
when BTN_CADASTRAR .Click
do if call usuario
user TB_USUARIO . Text
then if TB_NOME . Text ≠ "" and DT_NASC . Text ≠ "__/__/____" and TB_USUARIO . Text ≠ "" and PTB_SENHA . Text ≠ "" and PTB_CONF_SENHA . Text ≠ ""
then if PTB_SENHA . Text = PTB_CONF_SENHA . Text
then add items to list list get global lista
item TB_NOME . Text
item DT_NASC . Text
item TB_USUARIO . Text
item PTB_SENHA . Text
call CADASTROS .StoreValue
tag TB_USUARIO . Text
valueToStore get global lista
open another screen screenName "Screen1"
else call ALERTA_CADASTRO .ShowMessageDialog
message "As senhas inseridas não coincidem. Digite novame..."
title "Senhas incompatíveis"
buttonText "OK"
else call ALERTA_CADASTRO .ShowMessageDialog
message "É necessário preencher todos os campos"
title "Preencha os dados"
buttonText "OK"
else call ALERTA_CADASTRO .ShowMessageDialog
message "Este nome de usuário já está sendo utilizado. Po..."
title "Usuário já cadastrado"
buttonText "OK"
set PTB_SENHA . Text to ""
set PTB_CONF_SENHA . Text to ""

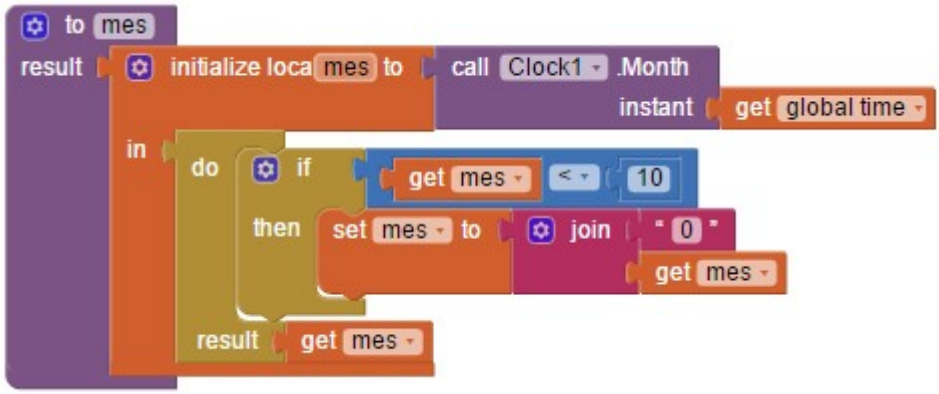
MENU

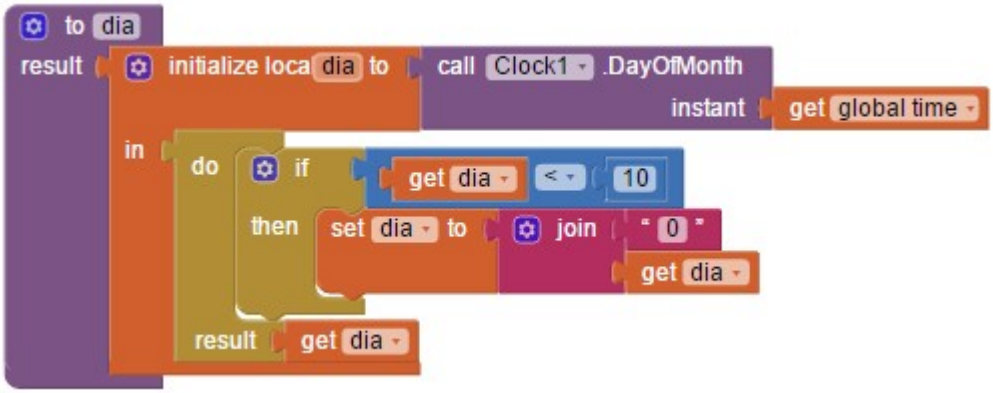

when MENU .Initialize
do set NOME . Text to select list item list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere
index 1
if not BluetoothClient1 . Enabled
then call ActivityStarter1 .StartActivity
when MEDIÇÃO .Click
do if BluetoothClient1 . Enabled
then open another screen with start value screenName " ARMAZENAR "
startValue get start value
else call ALERTA_MENU .ShowMessageDialog
message " Habilite o bluetooth "
title " Bluetooth "
buttonText " OK "
call ActivityStarter1 .StartActivity
when HISTÓRICO .Click
do open another screen with start value screenName " HISTORICO "
startValue get start value
when AJUDA .Click
do open another screen screenName " AJUDA "


when EXCLUIR_CADASTRO .Click
do call ALERTA_MENU .ShowChooseDialog
message " Deseja realmente excluir cadastro? Todas as info... "
title " Excluir cadastro "
button1Text " Manter cadastro "
button2Text " Excluir cadastro "
cancelable false
when ALERTA_MENU .AfterChoosing
choice
do if compare texts get choice = " Excluir cadastro "
then call CADASTROS .ClearTag
tag get start value
open another screen screenName " Screen1 "
when MENU .BackPressed
do call ALERTA_LOGOUT .ShowChooseDialog
message " Deseja realmente finalizar sessão? "
title " Encerrar sessão "
button1Text " Manter sessão "
button2Text " Finalizar sessão "
cancelable false
when ALERTA_LOGOUT .AfterChoosing
choice
do if compare texts get choice = " Finalizar sessão "
then open another screen screenName " Screen1 "

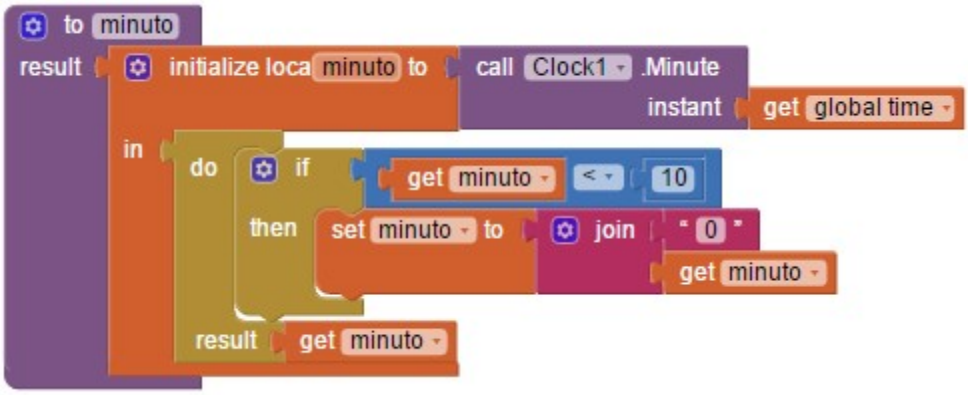
ARMAZENAR


initialize global n_list to create empty list
when ARMAZENAR .Initialize
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
call ListPicker1 .Open
when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
if BluetoothClient1 . IsConnected
then set Clock1 . TimerAlwaysFires to true
initialize global input to " "
when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive > 0
then set global input to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set LB_MEDIÇÃO . Text to get global input
when BTN_CANCELAR .Click
do open another screen with start value screenName " MENU "
startValue get start value
when ARMAZENAR .BackPressed
do open another screen with start value screenName " MENU "
startValue get start value
when BTN_NOVA_MEDIDA .Click
do call BluetoothClient1 .Disconnect
open another screen with start value screenName " ARMAZENAR "
startValue get start value
initialize global time to " "

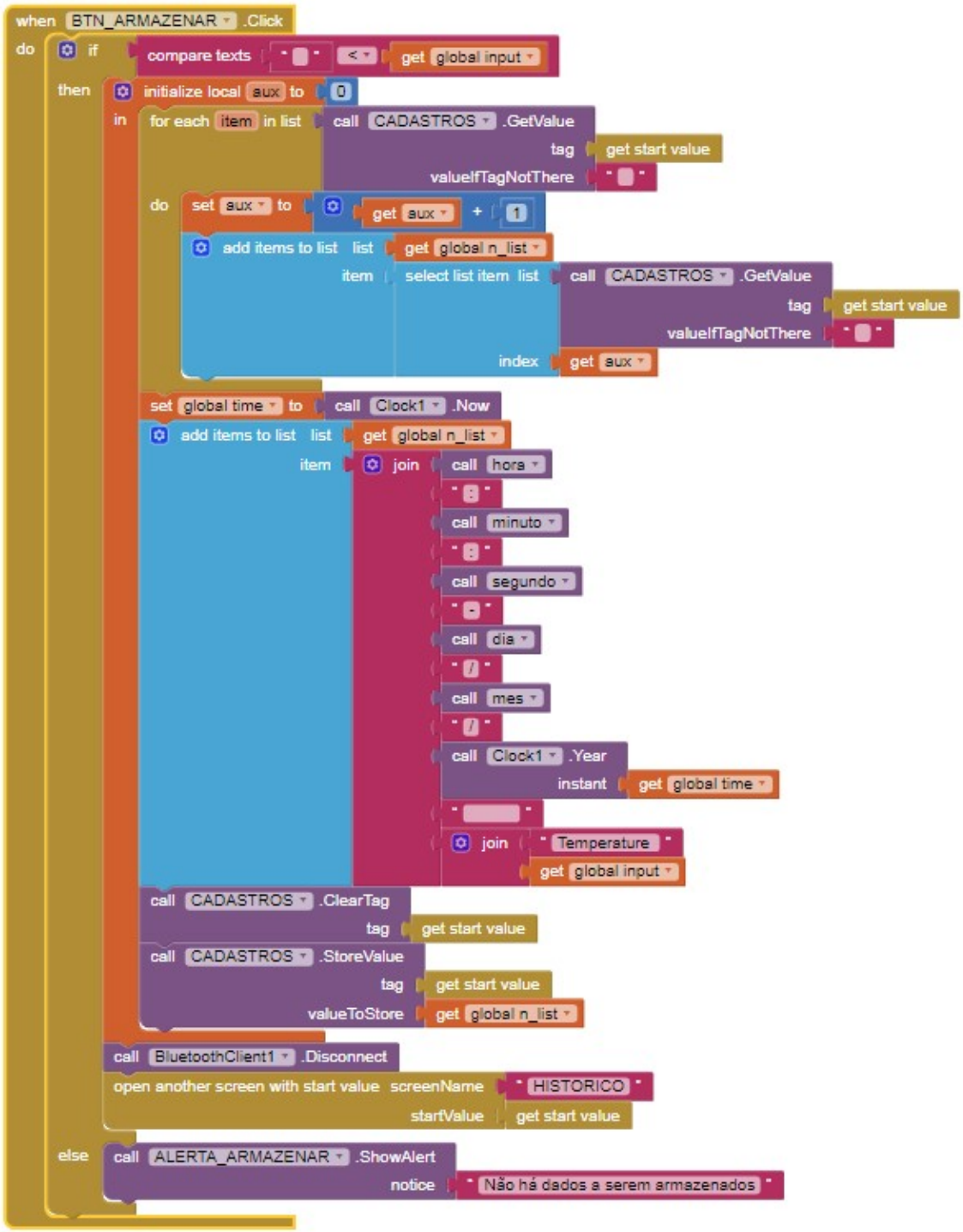

to mes
result
initialize local mes to
call Clock1 .Month
instant
get global time
in
do
if
get mes
<
10
then
set mes to
join
" 0 "
get mes
result
get mes


to dia
result
initialize local dia to
call Clock1 .DayOfMonth
instant
get global time
in
do
if
get dia
<
10
then
set dia to
join
" 0 "
get dia
result
get dia

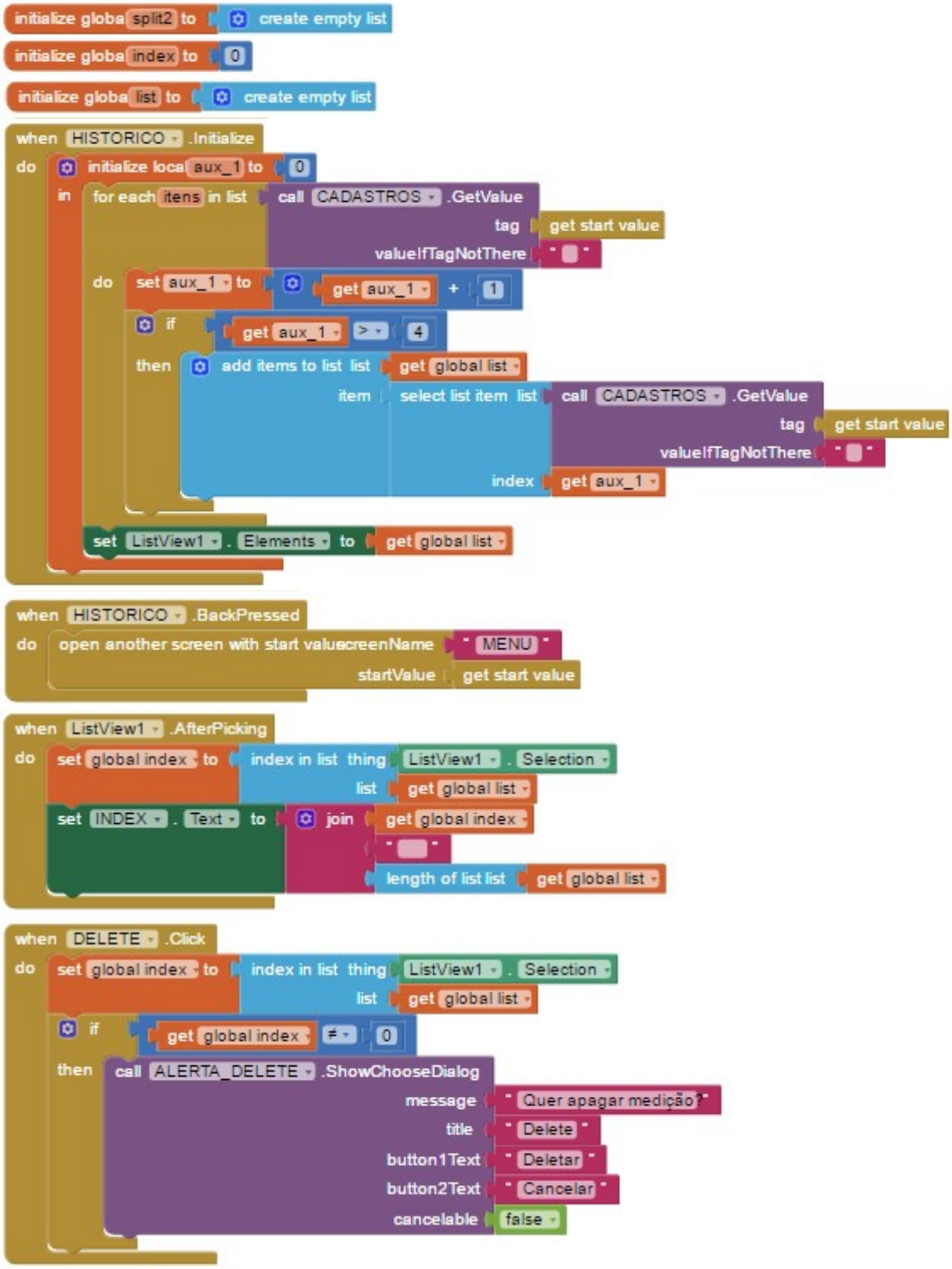
to hora
result
initialize local hora to
call Clock1 .Hour
instant
get global time
in
do
if
get hora
<
10
then
set hora to
join
" 0 "
get hora
set LB_MEDIÇÃO . Text to
get hora
result
get hora

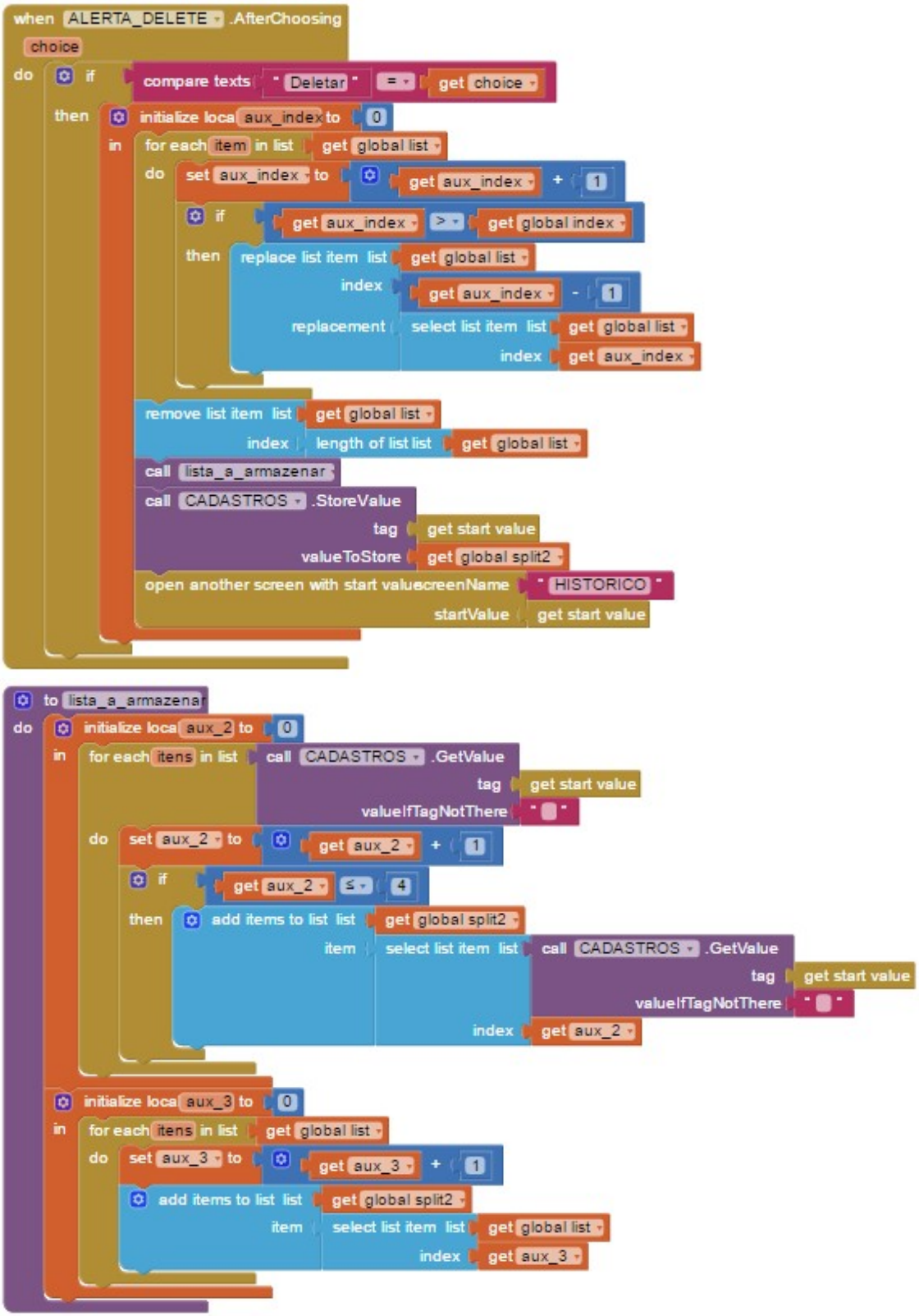

to minuto
result
initialize loca minuto to
call Clock1 .Minute
instant
get global time
in
do
if
get minuto
<
10
then
set minuto to
join
" 0 "
get minuto
result
get minuto

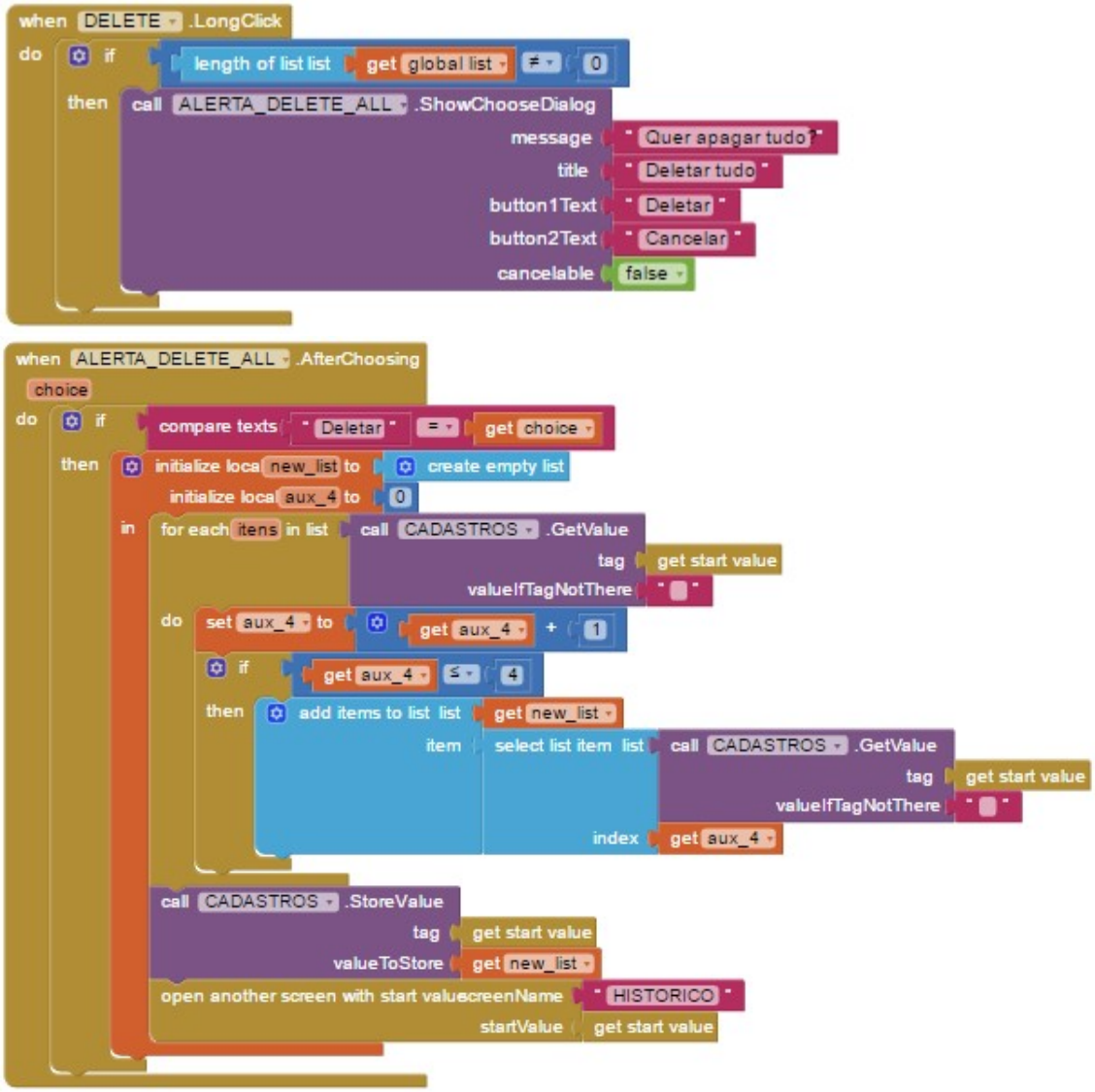
to segundo
result
initialize loca segundo to
call Clock1 .Second
instant
get global time
in
do
if
get segundo
<
10
then
set segundo to
join
" 0 "
get segundo
result
get segundo


when BTN_ARMAZENAR .Click
do
if
compare texts
<
get global input
then
initialize local aux to
0
in
for each item in list
call CADASTROS .GetValue
tag
get start value
valueIfTagNotThere
do
set aux to
get aux
+
1
add items to list list
get global n_list
item
select list item list
call CADASTROS .GetValue
tag
get start value
valueIfTagNotThere
index
get aux
set global time to
call Clock1 .Now
add items to list list
get global n_list
item
join
call hora
call minuto
call segundo
call dia
call mes
call Clock1 .Year
instant
get global time
join
Temperature
get global input
call CADASTROS .ClearTag
tag
get start value
call CADASTROS .StoreValue
tag
get start value
valueToStore
get global n_list
call BluetoothClient1 .Disconnect
open another screen with start value screenName
HISTORICO
startValue
get start value
else
call ALERTA_ARMAZENAR .ShowAlert
notice
Não há dados a serem armazenados

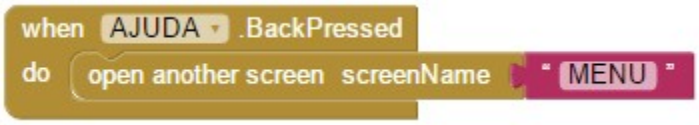
HISTÓRICO


initialize global split2 to create empty list
initialize global index to 0
initialize global list to create empty list
when HISTORICO .Initialize
do initialize local aux_1 to 0
in for each itens in list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere
do set aux_1 to get aux_1 + 1
if get aux_1 > 4
then add items to list list get global list
item select list item list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere
index get aux_1
set ListView1 . Elements to get global list
when HISTORICO .BackPressed
do open another screen with start valuescreenName MENU
startValue get start value
when ListView1 .AfterPicking
do set global index to index in list thing ListView1 . Selection
list get global list
set INDEX . Text to join get global index
length of list list get global list
when DELETE .Click
do set global index to index in list thing ListView1 . Selection
list get global list
if get global index ≠ 0
then call ALERTA_DELETE .ShowChooseDialog
message Quer apagar medição?
title Delete
button1Text Deletar
button2Text Cancelar
cancelable false


when ALERTA_DELETE .AfterChoosing
choice
do if compare texts " Deletar " = get choice
then initialize local aux_index to 0
in for each item in list get global list
do set aux_index to get aux_index + 1
if get aux_index > get global index
then replace list item list get global list
index get aux_index - 1
replacement select list item list get global list
index get aux_index
remove list item list get global list
index length of list list get global list
call lista_a_armazenar
call CADASTROS .StoreValue
tag get start value
valueToStore get global split2
open another screen with start valuescreenName " HISTORICO "
startValue get start value
to lista_a_armazenar
do initialize local aux_2 to 0
in for each itens in list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere " "
do set aux_2 to get aux_2 + 1
if get aux_2 ≤ 4
then add items to list list get global split2
item select list item list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere " "
index get aux_2
initialize local aux_3 to 0
in for each itens in list get global list
do set aux_3 to get aux_3 + 1
add items to list list get global split2
item select list item list get global list
index get aux_3


when DELETE .LongClick
do if length of list list get global list ≠ 0
then call ALERTA_DELETE_ALL .ShowChooseDialog
message "Quer apagar tudo?"
title "Deletar tudo"
button1Text "Deletar"
button2Text "Cancelar"
cancelable false
when ALERTA_DELETE_ALL .AfterChoosing
choice
do if compare texts "Deletar" = get choice
then initialize local new_list to create empty list
initialize local aux_4 to 0
in for each itens in list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere
do set aux_4 to get aux_4 + 1
if get aux_4 ≤ 4
then add items to list list get new_list
item select list item list call CADASTROS .GetValue
tag get start value
valueIfTagNotThere
index get aux_4
call CADASTROS .StoreValue
tag get start value
valueToStore get new_list
open another screen with start valuescreenName "HISTORICO"
startValue get start value

AJUDA


when AJUDA .BackPressed
do open another screen screenName "MENU"