

PROJETO ALTISSIMA COMPLEXIDADE

RELATÓRIO

RE-DESIGN CAMINHÃO
DE COLETA DE RESÍDUOS

CIDADE INTELIGENTE

UNIVERSIDADE FEDERAL DE SANTA CATARIA 2022

MORGANA GRZYBOWSKI, PEDRO LERINA, ARTHUR MARPE

SUMÁRIO

01	Introdução	02	Pesquisa	03	Painel Semantico Estações de tratamento
04	Definição e refinamento da alternativa	05	Render Cooperativa	06	Materialização Cooperativa
07	Render Cooperativa	08	Materialização Cooperativa	09	Braço robótico no Biodigestor
10	Objetivo da nova proposta de Redesign	11	Refinamento Requisitos	12	Painel Semântico Redesign
14	Sketches e Alternativas	15	Matriz de Decisão	16	Alternativa Escolhida
17	Refinamento	18	Render Caminhão	19	Modelagem Caminhão
17	Materialização Caminhão	18	Alteração dos Componentes	19	Definição e Escolha Portas ESP32
20	Conexões dos Componentes	21	Definição Ciclos de Programação	22	Programação
23	solda e Montagem dos Componentes	24	Projeto Final	25	Conclusão

INTRODUÇÃO AO MÓDULO DE PROJETO 4

Nesse novo relatório, será realizada a continuidade do projeto do módulo 3, que teve como resultado o desenvolvimento de um caminhão robótico para um tabuleiro que representará a cidade de Florianópolis, de maneira automatizada e inteligente. Esse caminhão será o responsável pela coleta de lixo na cidade e foi programado para andar em uma linha central que remeterá o mapa da cidade com seus principais pontos de coleta.

Nosso objetivo então, é desenvolver e projetar o re-design desse caminhão, bem como complementar suas funções e trajeto, contribuindo de maneira significativa, inovadora e interativa para o projeto já em andamento.

Para isso, foi realizada a continuação da pesquisa teórica do módulo 3, sendo complementado informações sobre gestão de resíduos, biodigestor, cooperativa de reciclagem, e como essas se dão presentes em Santa Catarina e Florianópolis. Além disso, foram realizadas novas alternativas para o re-design do caminhão, bem como o desenvolvimento e materialização do biodigestor, dos pontos de coleta e da cooperativa de reciclagem.

PESQUISA GESTÃO DE RESÍDUOS

A pesquisa relacionada a gestão de resíduos foi focada em dois pontos de parada já definidos no projeto do kit robótico, considerando a união de dois processos, o biodigestor e compostagem, por terem uso de materiais orgânicos, apresentando resultados semelhantes sobre a redução da emissão de gases, quanto a substituição dos aterros sanitários, assim possibilitando a gestão em um único ponto do mapa. A pesquisa dá ênfase no entendimento geral sobre o funcionamento dos processos, assim como a compreensão sobre o uso dos sistemas no mundo e no Estado de Santa Catarina.

3.1 Biodigestor

Apesar do biogás ser uma descoberta antiga (por volta de 1600), A Granja do Torto, moradia de alguns presidentes localizada em Brasília, recebeu um dos primeiros biodigestores do Brasil por volta de 1979, mesmo que eles tenham sido um fracasso, pois não supriram a necessidade dos produtores.



Estes, são compartimentos onde ocorre a decomposição de materiais orgânicos, como resíduos da produção vegetal, animal, industrial e humana, e assim são produzidos biogás e biofertilizantes. Para isso, existe a chamada digestão anaeróbica, que por sua vez é realizada por bactérias. Por existirem diferentes tipos de biodigestores e possuírem diversos benefícios, muitas pessoas ainda têm dúvidas sobre como utilizá-los e como utilizá-los para otimizar a gestão de resíduos e gerar energia no campo.

Desse modo, a fabricação de biogás tem impacto direto na preservação do meio ambiente. O secretário Airton Spies explica que a captação do gás metano na decomposição dos dejetos de animais reduz em 21 vezes o efeito estufa. “O metano é um gás 21 vezes mais nocivo para a geração do efeito estufa do que o gás carbônico, que é liberado na geração de energia após a biodigestão. Por isso, a Lei do Biogás contribui significativamente para a redução do aquecimento global, mantendo o equilíbrio e reduzindo as mudanças climáticas”.

3.1.1 Em Santa Catarina

Em 2018, o estado de Santa Catarina criou um incentivo para converter resíduos orgânicos da produção animal em energia. O principal objetivo do projeto de lei é orientar sobre os resíduos (esterco e urina) produzidos pela pecuária e os resíduos orgânicos produzidos em áreas urbanas para produção de energia por meio de biodigestores. Além de tratar do passivo ambiental - poluição por dejetos animais - a política visa proporcionar às populações rurais uma nova fonte de renda e diversificar a matriz energética catarinense. Por meio do Programa SC Rural da Secretaria Estadual de Agricultura e Pesca, com financiamento do Banco Mundial, foram contratados os serviços de consultoria de 3 empresas para diagnosticar o potencial de geração de biogás em Santa Catarina e identificar os maiores locais de geração de resíduos. As regiões de Concórdia e Braço do norte, por exemplo, estão à frente na produção de dejetos suínos, enquanto Biguaçu tem grande fonte de matéria-prima a partir dos resíduos sólidos. A prospecção também detectou os potenciais segmentos e grupos de consumidores deste tipo de energia, estudando as redes de distribuição, geração e aproveitamento do biometano.

3.1.2 Cenário em Florianópolis

O distrito de Ratonas, tema do sétimo capítulo da série Florianópolis do Futuro, que apresenta o novo Plano Diretor da Capital, é um dos mais carentes e também o mais rural de Florianópolis. A paisagem do campo, graças à agropecuária, ao cultivo de hortaliças e a pequenas agroindústrias, é preservada. Os 3,7 mil habitantes são, em maioria, nativos que vivem da pesca e da agricultura de subsistência. Mais de 80% do território apresenta algum tipo de restrição, seja pelos morros, pelas áreas alagáveis, pelo manguezal do Rio Ratonas ou pelos 30% de mata densa. Ratonas não está na mira da especulação imobiliária. Com a aprovação do plano de 2014, as duas principais preocupações dos moradores são a aprovação do plano ambiental - com prazo até janeiro para ser concluído - e a revisão no tamanho mínimo dos lotes, para evitar o parcelamento do solo. Desse modo, a região de ratones, ganha um destaque sobre um local de boa utilização dessa energia.

3.1.3 Usinas e Projetos

O BioEnergie Park Güstrow na Alemanha produz biogás com qualidade de gás natural a partir de Recursos Naturais Renováveis. O projeto tem um potencial térmico de 50 kWth sendo em 2014 o maior biodigestor do mundo.

O projeto completo ocupa uma área de 50 hectares e emprega 50 pessoas devidamente qualificadas diretamente além de todas as pessoas que trabalham com a agricultura na região.



Em Santiago, capital do Chile, a empresa La Farfana, a maior estação privada de tratamento de efluentes do Chile, responsável por tratar 60% do esgoto gerado em Santiago ou o equivalente a 778 mil m³/dia gera em torno de 60 mil m³ de biogás por dia. No início o biogás costumava ser utilizado como fonte de calor no aquecimento de caldeiras de água, que serviam ao abastecimento de processos da própria ETE. Porém, a forte dependência do gás natural importado da Argentina, com abastecimento incerto e preços elevados motivou a construção de uma central para captação e purificação do biogás da estação, com o objetivo de transformá-lo em gás natural e comercializá-lo. O gás natural possui 70% teor de metano, enquanto que no biogás este teor é de aproximadamente 55%; portanto, para converter o biogás em gás natural é necessário remover suas impurezas, “liberando espaço” para novas moléculas de metano.



Está se tornando popular esse bio digestor pelo simples fato A empresa israelense HOME BIOGAS desenvolve digestores que convertem resíduos orgânicos em biogás e fertilizante orgânico líquido. Desde 2011, já foram instalados milhares desses equipamentos em mais de 70 países. E agora, chegou também ao Brasil oferecendo essa forma altamente sustentável de tratar o lixo orgânico.

O sistema produz até 3 horas de gás de cozinha todos os dias (equivalente a meio botijão de gás de cozinha por mês) usando apenas restos de comida e dejetos animais (pets e animais de criação), com vida útil de 15 anos.

Depois de iniciado o processo, as bactérias dentro do sistema funcionarão 24 horas por dia, 7 dias por semana, para decompor os resíduos e produzir biogás.



3.2. Compostagem

A compostagem é um processo que envolve a decomposição da matéria orgânica por microrganismos, que ocorre naturalmente e pode ser acelerado por intervenção, por meio de oxidação biológica por microrganismos para quebrar os compostos constituintes do material que liberam dióxido de carbono e vapor de água. Embora a maioria dos autores a considere um processo aeróbio, a compostagem também é referida como um processo biológico, que envolve a quebra aeróbica ou anaeróbica de resíduos biodegradáveis para produzir um produto.

O objetivo da compostagem é converter a matéria orgânica que não pode ser incorporada ao solo em algo que possa ser misturado ao solo ou usado para reduzir e estabilizar a matéria orgânica que vai para aterros sanitários.

3.2.1 Tipos de compostagem

A compostagem pode ser feita de diferentes formas: em grandes instalações centralizadas com recolha seletiva de matéria orgânica; no que diz respeito a agricultura ou pecuária; em pequenas unidades de cariz doméstico (compostagem doméstica) ou grandes unidades de usinas municipais.

Geralmente leva vários meses para obter material de compostagem satisfatório, embora alguns digestores de biogás com temperatura controlada e movimento constante de material de compostagem possam fornecer "composto acabado" em poucas semanas. Existem muitos sistemas de preparação de composto, mas geralmente podem ser divididos em três tipos: sistemas de leiras revolvidas, estáticas e sistemas fechados ou biorreatores (in-vessel). (SOARES, 2006).

Leiras Revolvidas

O sistema de leiras revolvidas funciona a partir das misturas de reíduos, que são dispostas em pilhas, sendo a aeração fornecida pelo revestimento dos resíduos e pela convecção e difusão do ar na massa do composto. Uma variante, além do revolvimento, utiliza a insuflação de ar sob pressão nas pilhas. Em alguns casos, a leira é montada sobre tubos perfurados que injetam ar na massa do composto. Esta alternativa é chamada de Leiras Revolvidas Aeradas.

Durante o processo as leiras são revolvidas no mínimo três vezes por semana. Já na etapa seguinte, o processo de maturação, o processo de aeração é menor, sendo revolvida entre 20 a 25 dias. O processo final tem duração em média de 2 a 3 meses. (SOARES, 2006).



leiras estáticas aeradas

Neste sistema, uma vez que a mistura de resíduos é colocada sobre a tubulação de aeração, ela permanece estática até o final da fase da bioestabilização

Nos Estados Unidos, onde o sistema é bastante utilizado, os sopradores variam de acordo com as características e volume dos resíduos, sendo ligados e desligados de maneira intermitente, durante a fase de bioestabilização. A aeração é dimensionada de acordo com três objetivos: satisfazer as demandas de oxigênio da biodegradação, remover o excesso de umidade e remover o excesso de calor para manter a temperatura em torno de 60°C. Um dos processos que auxilia o controle de odor é passar o ar aspirado em pilhas de composto já maturado, devido sua capacidade de reter moléculas orgânicas que causam o mau odor.



Sistema de reatores biológicos

A compostagem realizada em reatores biológicos, oferece a possibilidade de maior controle sobre todos os parâmetros importantes para o processo de compostagem, sendo denominada como o processo de compostagem acelerada. De acordo com as características dos resíduos e do tipo de equipamento, o tempo pode variar de 7 a 20 dias, o que faz com que o sistema demande menor espaço para sua implantação. A aeração é feita sob pressão e como o sistema é fechado, também se torna mais fácil monitorar a taxa de aeração através de medidores.

PAINEL SEMÂNTICO ESTAÇÕES DE TRATAMENTO

A partir da compreensão dos levantamentos sobre Cooperativa, Compostagem e Biodigestão, foram gerados painéis semânticos para visualizar referências estéticas sobre os temas. Seguindo a referência conceitual Futurista, foram coletadas imagens arquitetônicas que atribuissem no processo de criação de alternativas dos temas. O primeiro bloco define as referências sobre Cooperativa, seguido da Compostagem em conjunto ao Biodigestor.

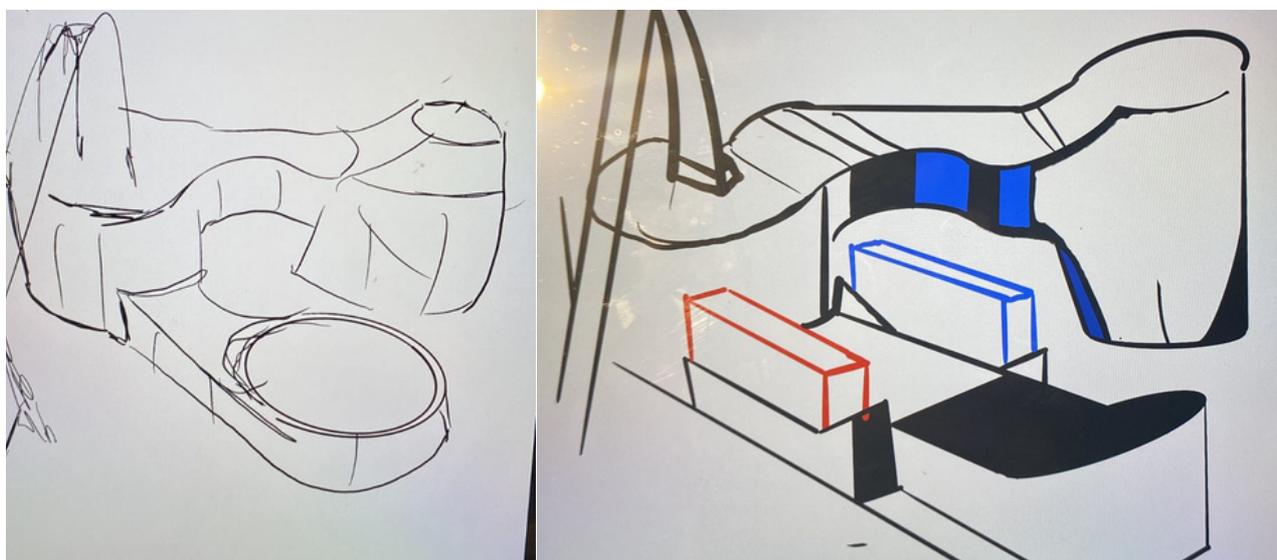


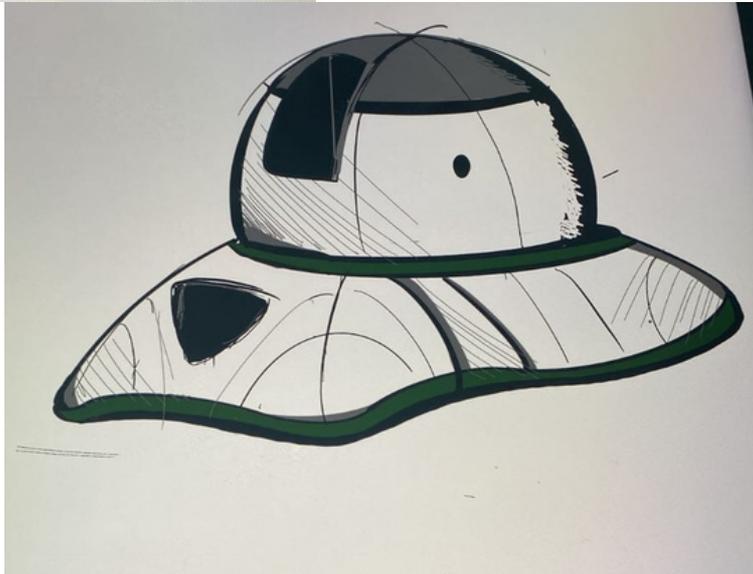
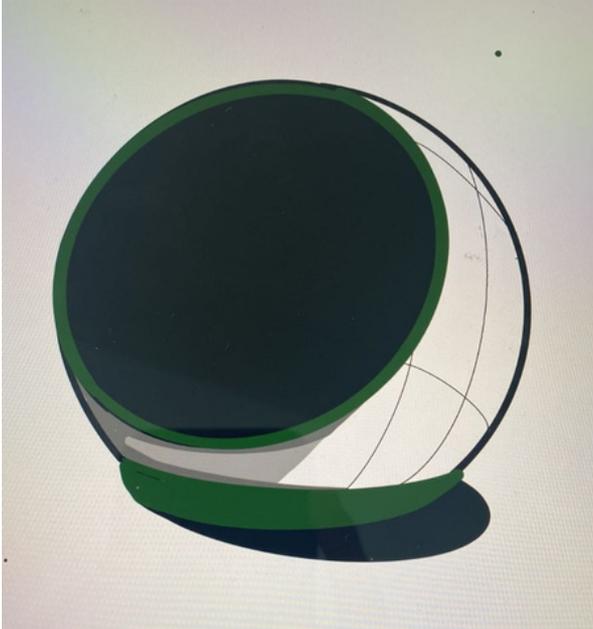
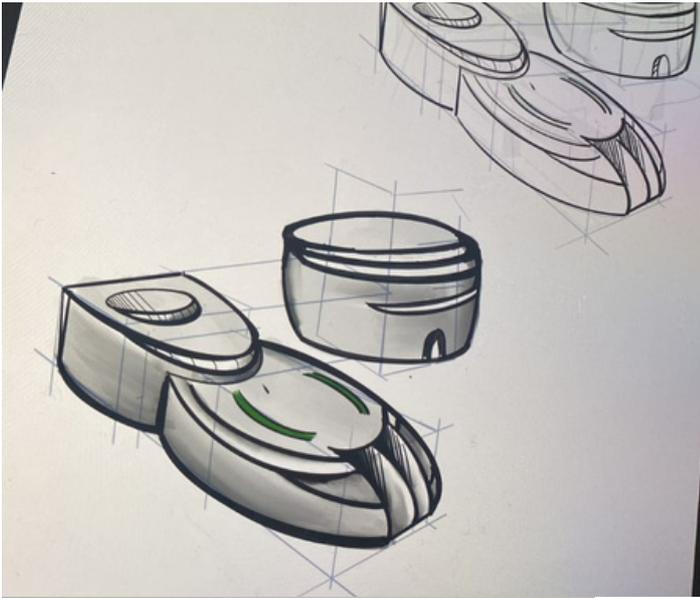


DEFINIÇÃO E REFINAMENTO DA ALTERNATIVA

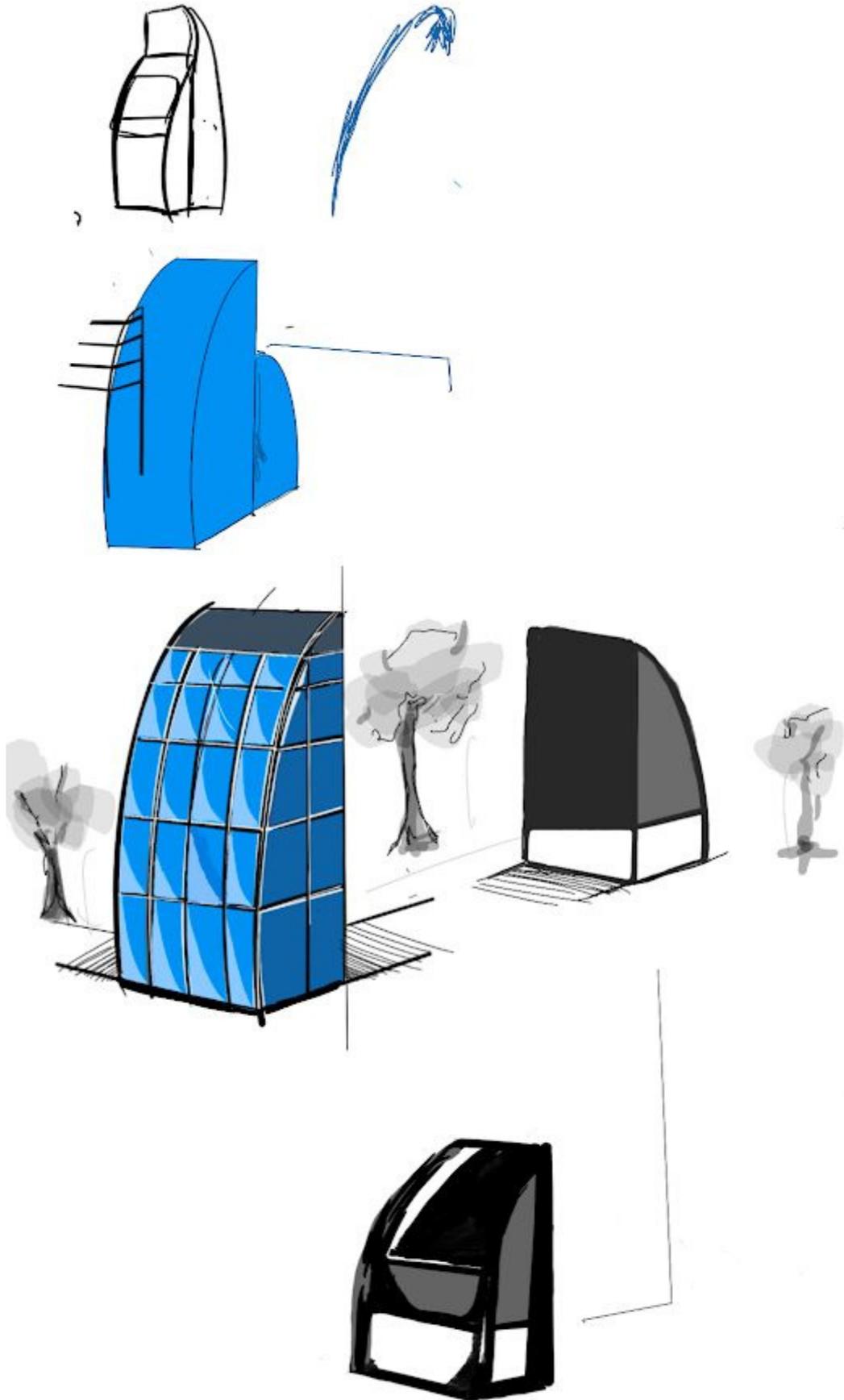
A partir das análises anteriores, foi iniciada a geração das alternativas. Foi definida uma alternativa para cada uma das estações de tratamento, sendo refinadas e adequadas de acordo com o referencial estético, como os métodos de materialização posteriormente definidos para cada uma das estações.

Biodigestor



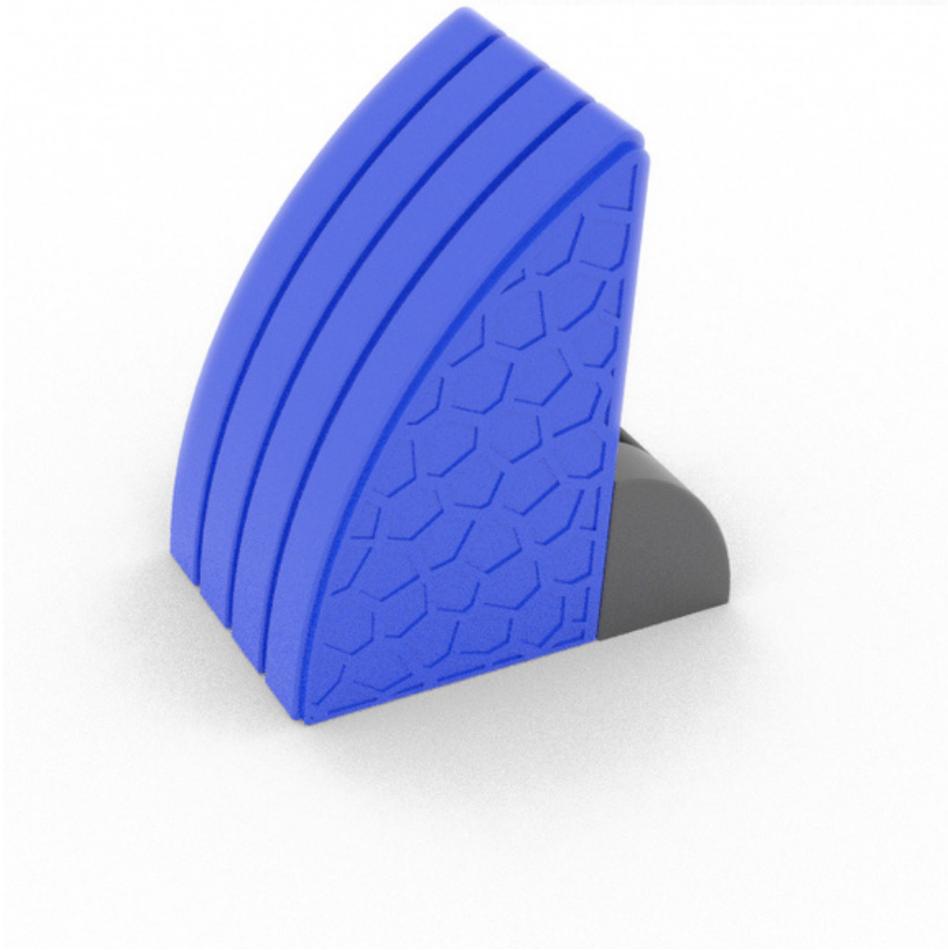
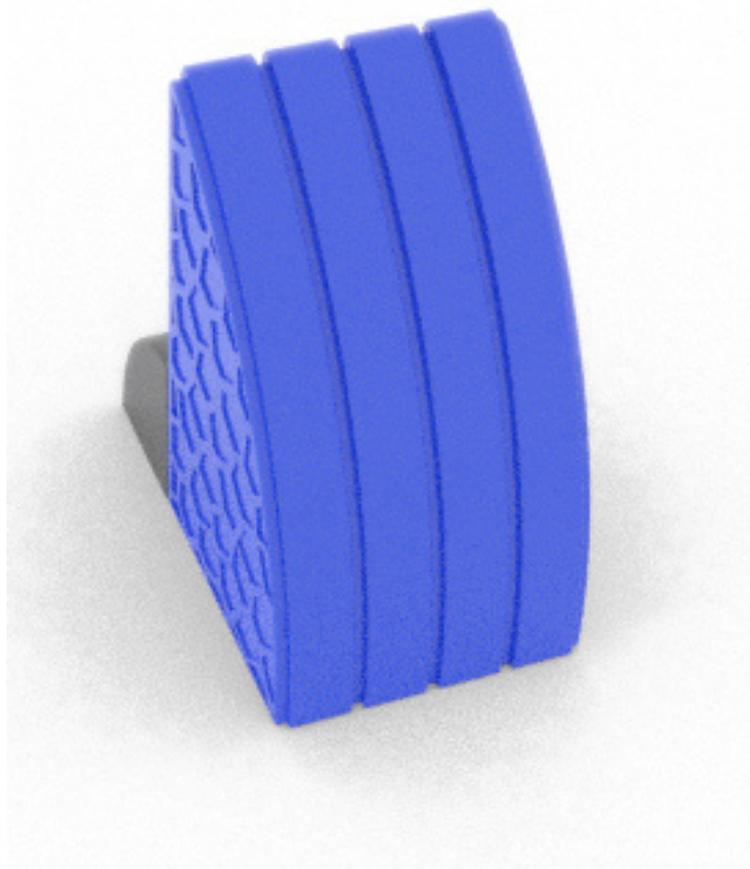
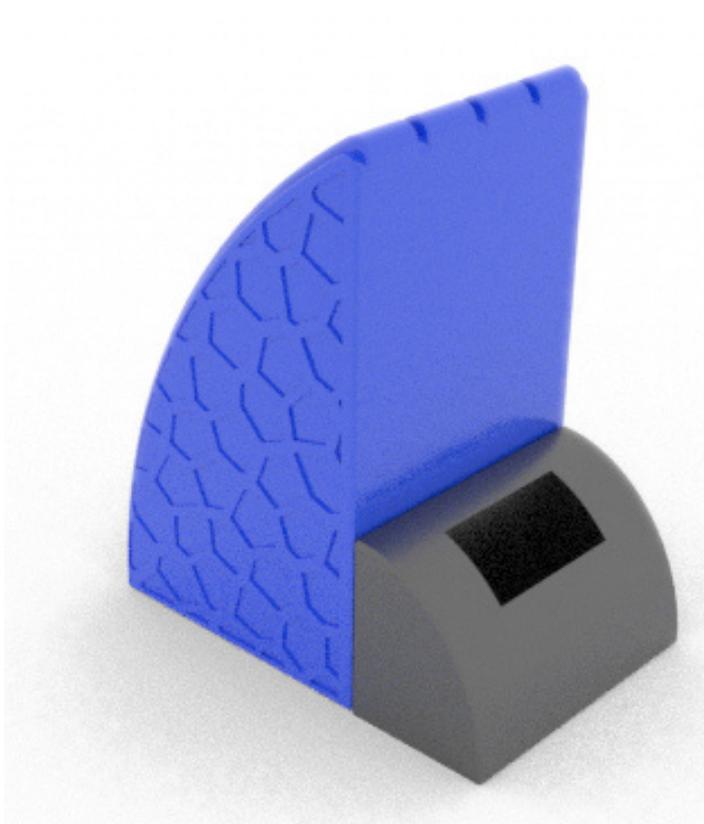


Cooperativa



RENDER COOPERATIVA

Desenvolvimento inspirado nas referencias encontradas para a Cooperativa, com design curvo e geométrico.

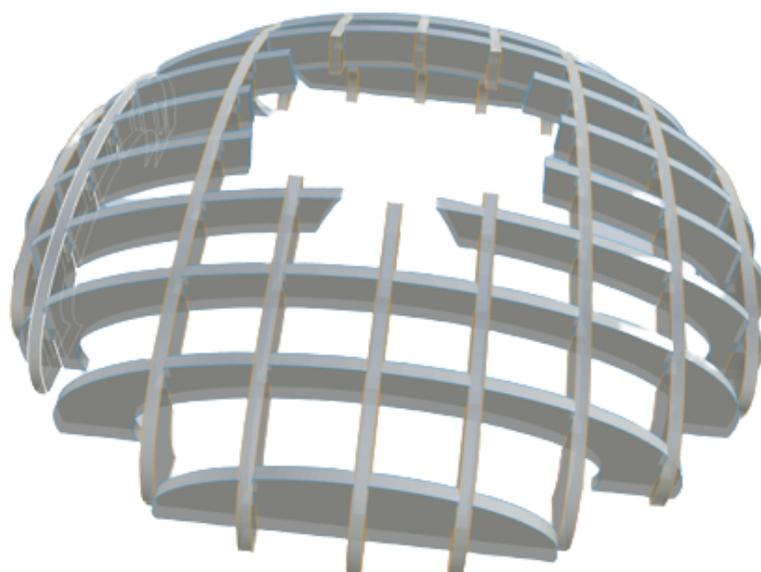
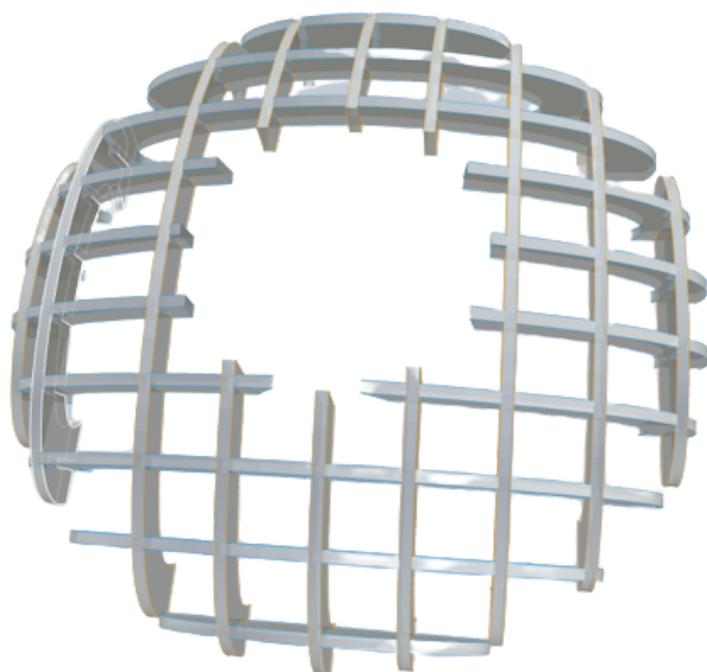


MATERIALIZAÇÃO COOPERATIVA

A Cooperativa foi feita a corte a laser por interlocking, construído no software Slicer.



RENDER BIODIGESTOR

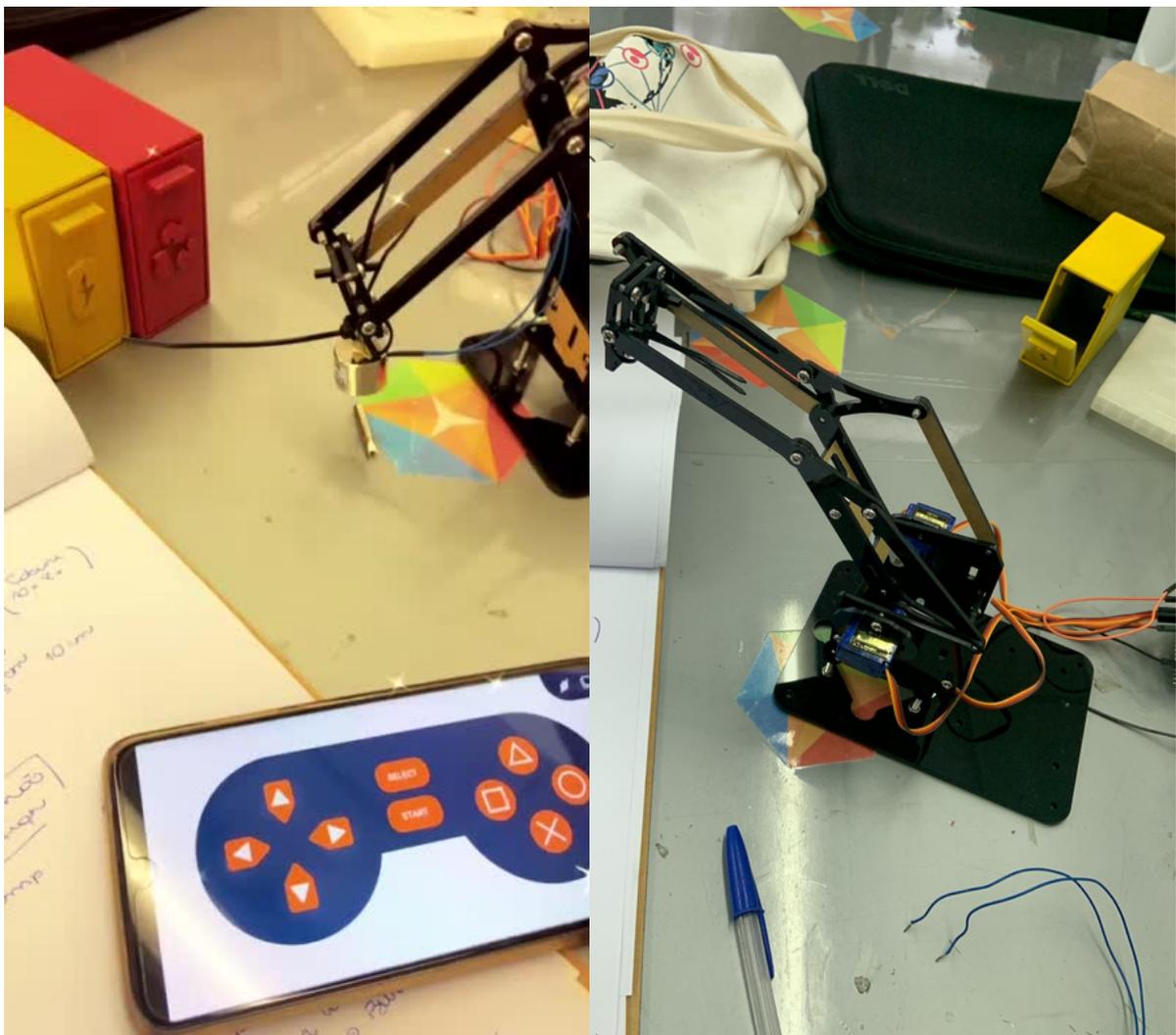


MATERIALIZAÇÃO BIODIGESTOR



BRAÇO ROBÓTICO NO BIODIGESTOR

Durante o projeto, foi implementado um braço robótico desenvolvido por um estudante, a fim de demonstrar o mecanismo dependente das ações dos usuários. Foi inserido juntamente ao Biodigestor para retirada dos resíduos dentro da caçamba de materiais orgânicos, através de um eletroímã, ao se aproximar dos cubos. O usuário movimenta o braço através dos comandos de um JoyStick virtual.



OBJETIVO DE NOVA PROPOSTA DE REDESIGN CAMINHÃO

A partir de discussões entre as equipes, foi estabelecido uma nova análise da proposta do projeto. Observou-se a necessidade da alteração dos conceitos pré estabelecidos no módulo de projeto 3. No módulo foram definidos os conceitos Futurista, Ludico e Educativo. Com a alteração optou-se por se distanciar do conceito Lúdico, concentrando-se numa representação futurista do caminhão sendo uma representação real sobre as cidades inteligentes. Para tal objetivo, foram realizadas pesquisas de referências estéticas, programação e materialização.

REFINAMENTO REQUISITOS

Os requisitos foram modificados de acordo com as características e aos atributos considerados nessa nova etapa, se alinhando com as definições descritas anteriormente, para que a geração das alternativas atenda a proposta final.

Categorias	Requisitos	Objetivo	Obrigatório	Desejável	Fonte	
Estrutural	Capacidade de carga	Alertar por sensor a capacidade máxima de lixo através de luminosidade	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Análise Sincrônica	
	Viés social	Aproximar o ensino de robótica e tecnologia	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Pesquisa	
		Tecnologia de coleta "autônoma"				
	Transporte de carga	Reservatórios separados através de cores da coleta seletiva;				Análise de Verificação
		Carregar certa quantidade de peso;				
	Sistema simples	Carregar recipientes de resíduos até as estações de tratamento: Coleta (casas) e Descarte (Compostagem+Biodigestor+Cooperativa)	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Acessibilidade remota	Wireframes que trazem estética e funções de fácil compreensão	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Publico Alvo		
Compacto	Direção remota pelo aplicativo onde passará sobre o trajeto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Relatório kit robótico		
Funcional	Coleta Seletiva	Dimensão do brinquedo (medidas), seguindo a escala proposta	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Análise Sincrônica	
		Mostrar etapas da coleta			Pesquisa / Relatório kit robótico	
		coleta todos os materiais ao mesmo tempo	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Comando Interativo	Separação de resíduos por recicláveis, orgânicos e rejeitos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Observação	
	Pontos culturais	Uso de controle/aplicativo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Pesquisa	
	Trajeto Estipulado	Remeter a localidade da região através de 3 referências	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Análise Estrutural	
Estético	Design Futurista	3xParadas de coleta e descarte.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Pesquisa/Materialização	
		1xCaminho - Percorrer todo o tabuleio (ilha)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Pesquisa/Materialização	
		Formato Geométrico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Geração de Alternativas	
	Transmitir solidez e estabilidade	Cores neutras (branco e preto) e fluorescentes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Conceitos	
	Simbolos intuitivos	Referencia semantica do conceito futuro e no tema Cidade Inteligente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Pesquisa	
	Linhas marcadas mostrando continuidade	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Geração de Alternativas		
	Uso de ícones e cores referentes ao tipo de material	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Análise Estrutural		
	Avisos visuais por iluminação	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

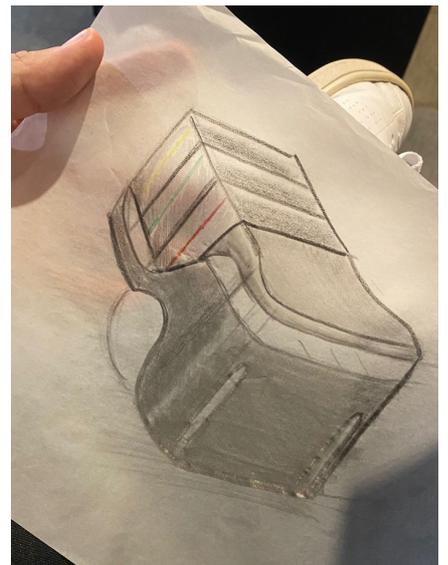
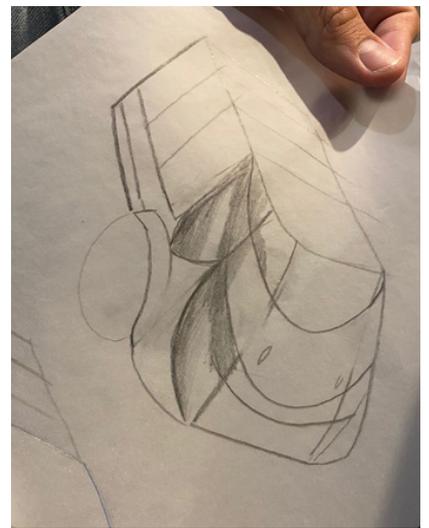
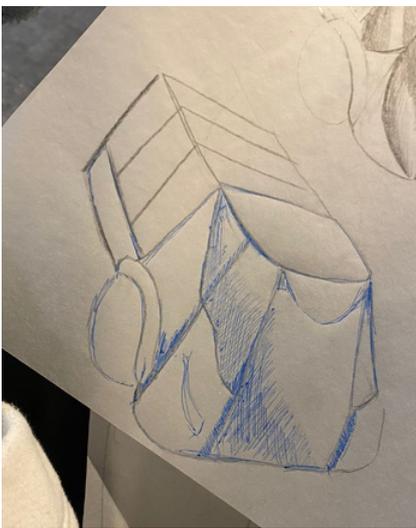
PAINEL SEMANTICO, REDESIGN CAMINHÃO

Foram selecionadas referências estéticas do caminhão a fim de aproximar as alternativas do conceito.



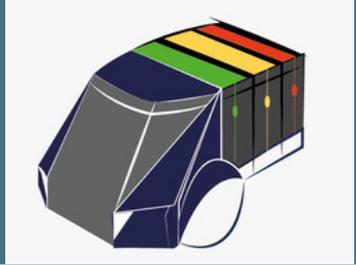
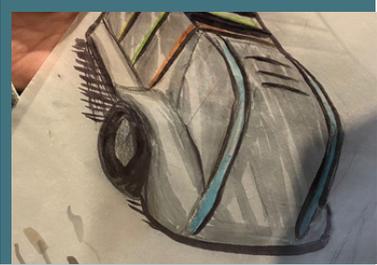
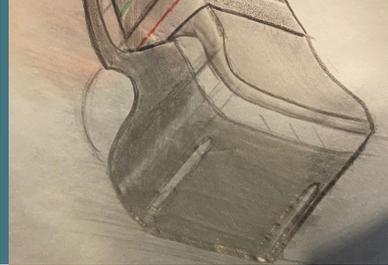
SKETCHES E ALTERNATIVAS

Com os painéis, foi iniciado a criação de sketches para conceitualização e visualização sobre o processo de criação das alternativas propostas para o projeto.

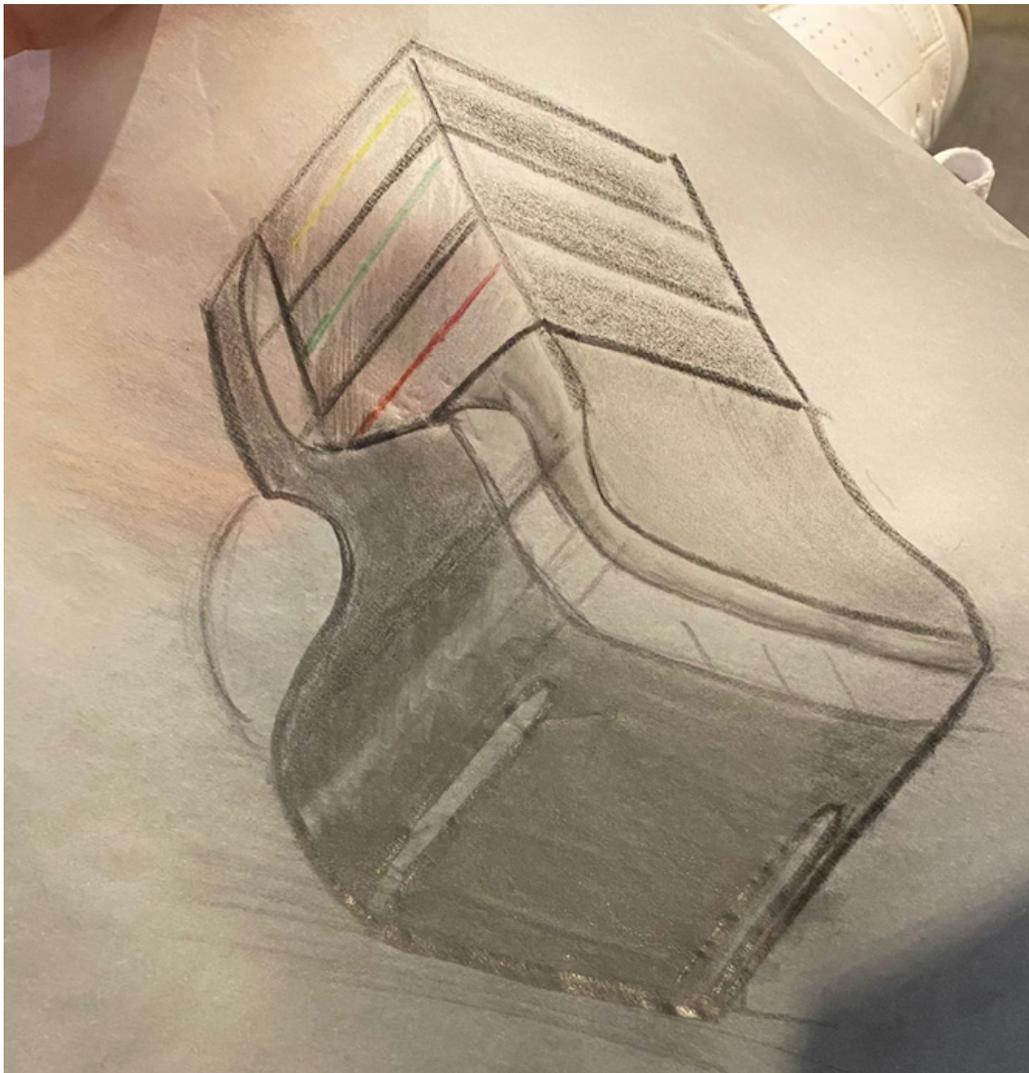


MATRIZ DE DECISÃO

Critérios analisados com indicador de 0 a 3, sendo 0 o indicador mais fraco e 3 o indicador mais forte, sendo 8 critérios em cor verde escuro obrigatórios e com peso 2, e 2 critérios em cor verde claro desejáveis com peso 1.

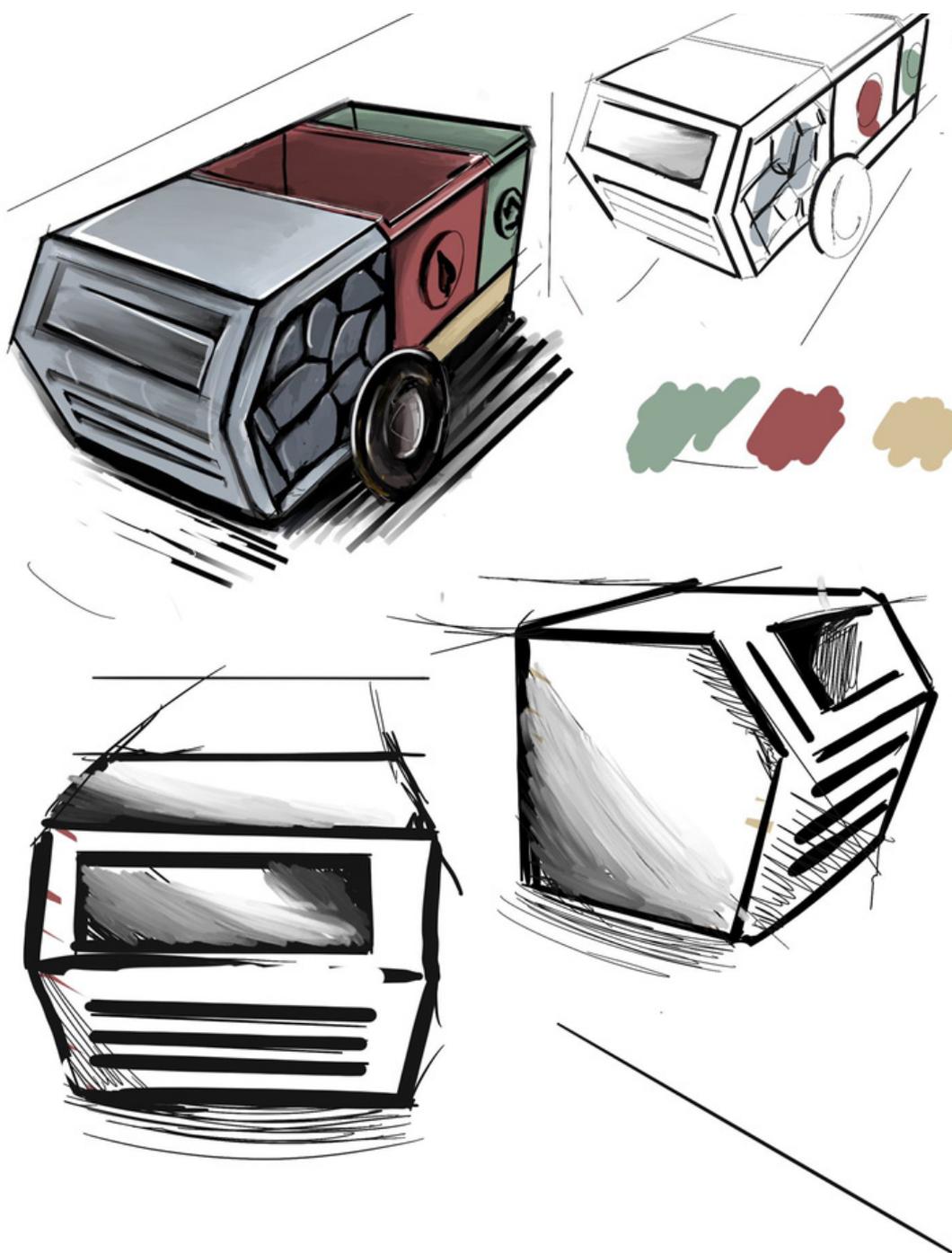
Requisito de projeto			
Massa e volume / Capacidade armazenamento	2	3	3
Sistema simples	3	2	3
Transmitir solidez e estabilidade	2	3	3
Carregar resíduos até as estações	2	2	3
Controlado por comandos à distancia (app)	3	3	3
Design Futurista	1	3	3
Interatividade	3	3	3
Educativo	3	3	3
Formas Orgânicas	3	2	3
Minimalista	3	2	3
TOTAL	34	35	38

ALTERNATIVA ESCOLHIDA

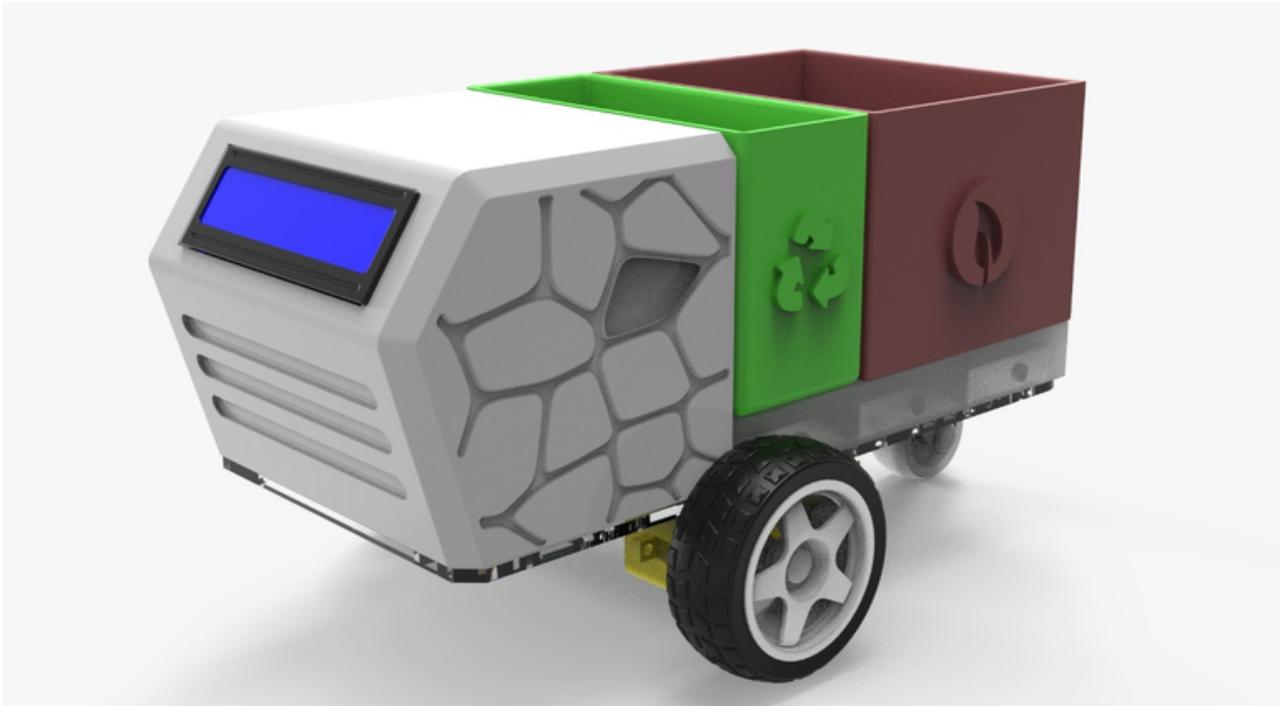


REFINAMENTO

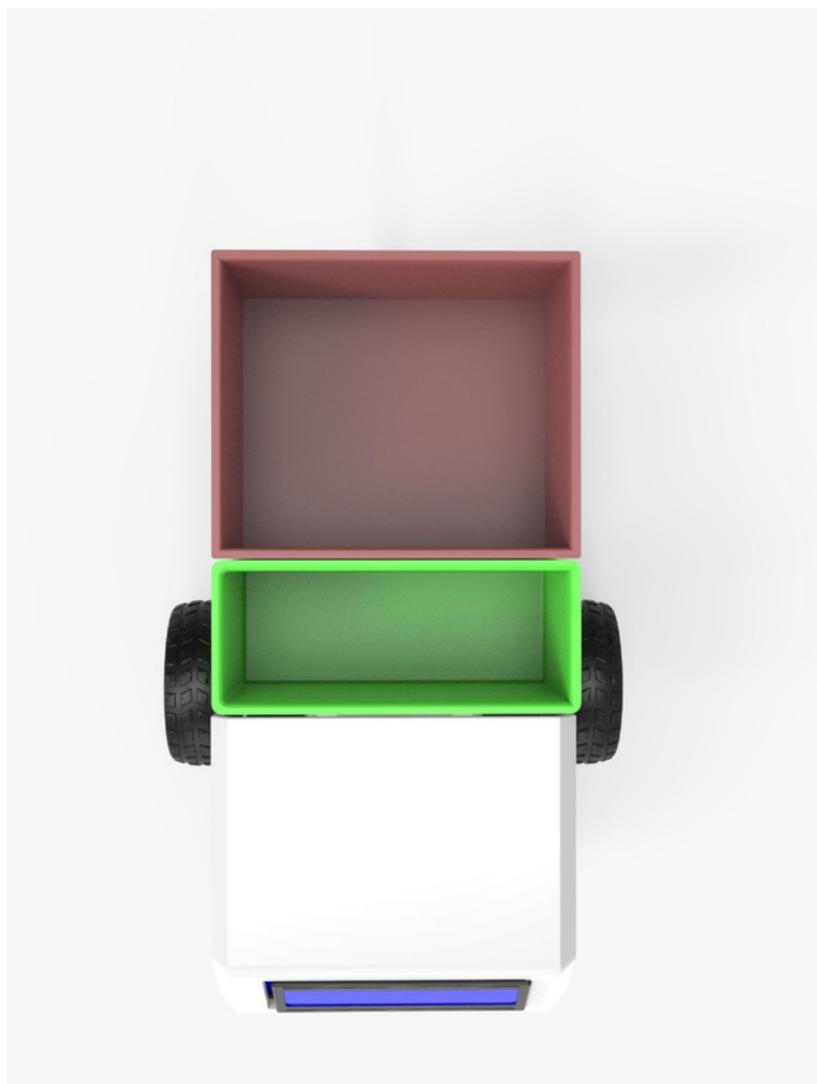
O Refinamento da cabine trouxe ao produto uma linha futurista, casando com a identidade do tabuleiro. As curvas foram trocadas por linhas retas, trazendo um visão mais direta e rebuscada.



RENDER CAMINHÃO

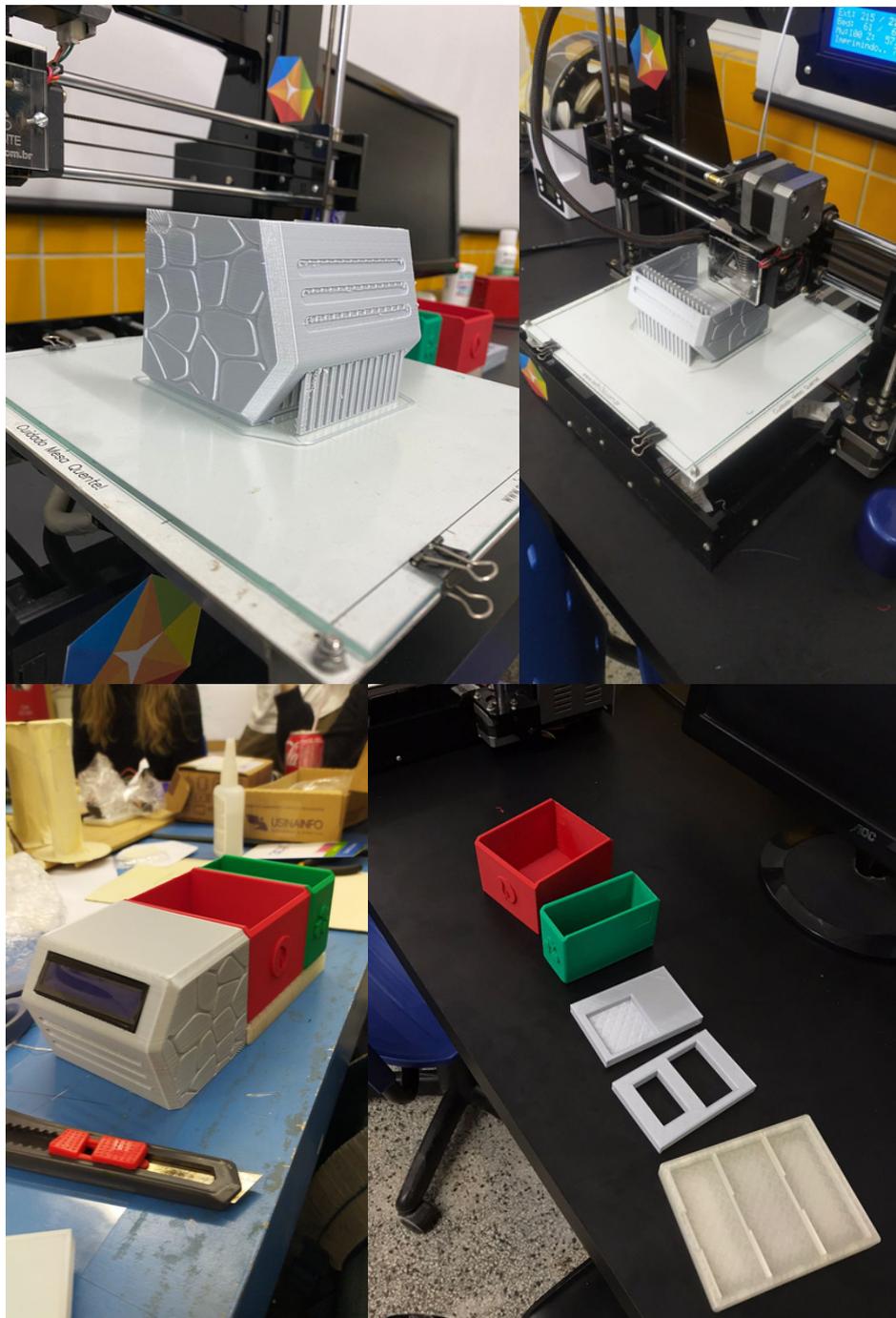


MODELAGEM CAMINHÃO



MATERIALIZAÇÃO CAMINHÃO

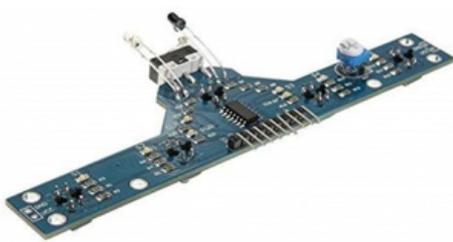
Para a materialização do caminhão, optou-se por trabalhar com a impressão 3D das estruturas superiores como a cabine, as caçambas, os resíduos e a base das caçambas. O chassi de acrílico foi substituído por um de MDF com as dimensões redefinidas seguindo o conjunto superior do caminhão.



ALTERAÇÕES DOS COMPONENTES

Houveram algumas alterações dos componentes definidos no módulo anterior. O sensor de linha IR5 foi substituído por dois sensores de linha tcr5000, pois ambos executam a mesma função, de forma que se encaixa nas estruturas do chassi. Outra alteração foi à inclusão do do display LCD I2C que informa aos usuários os pontos de chegada nas estações e pontos de coleta. Além disso, os LEDs RGB foram substituídos por LEDs comuns, pois o numero de portas do ESP32 não comportaria a quantidade de conexões de cada um dos LEDs.

Sensor de Linha IR5



sensores de linha tcr5000



LED RGB



LED



Display I2C

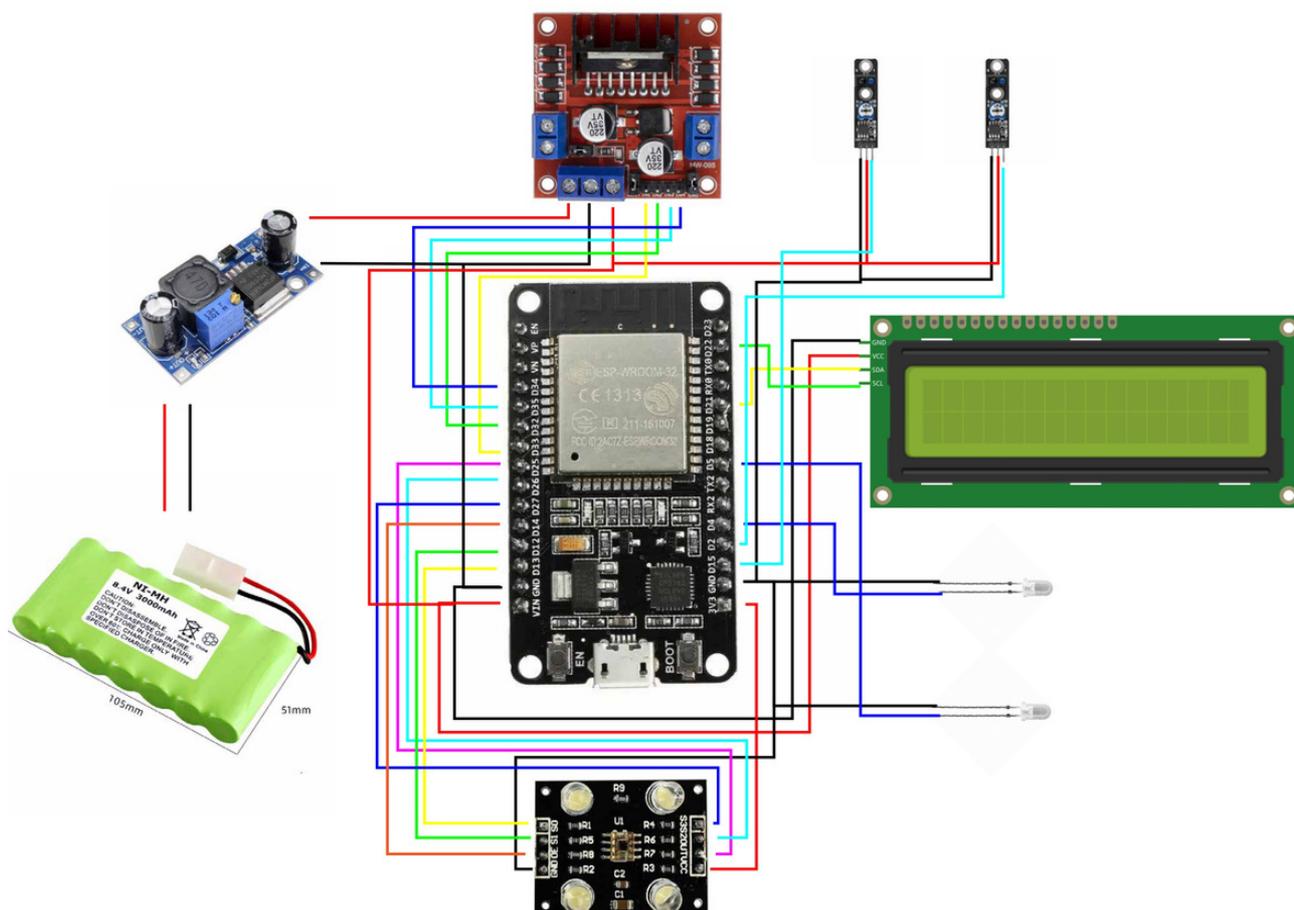


DEFINIÇÃO ESCOLHA DE PORTAS ESP32

O ESP32 possui algumas portas específicas para a utilização de alguns sensores. Para entender as conexões, foi estabelecido um esquema para facilitar o processo de conexões dos componentes.

MOTORES	LCD	SENSOR COR	SENSOR LINHA	LEDS
ENA 26	SDA 21	SO 19	S1 35	LED1 33
IN1 27	SCL 22	S1 23	S2 34	LED 2 32
IN2 14		S2 5		LED 3 15
ENB 25		S3 18		
IN3 12		OUT 4		
IN4 13				

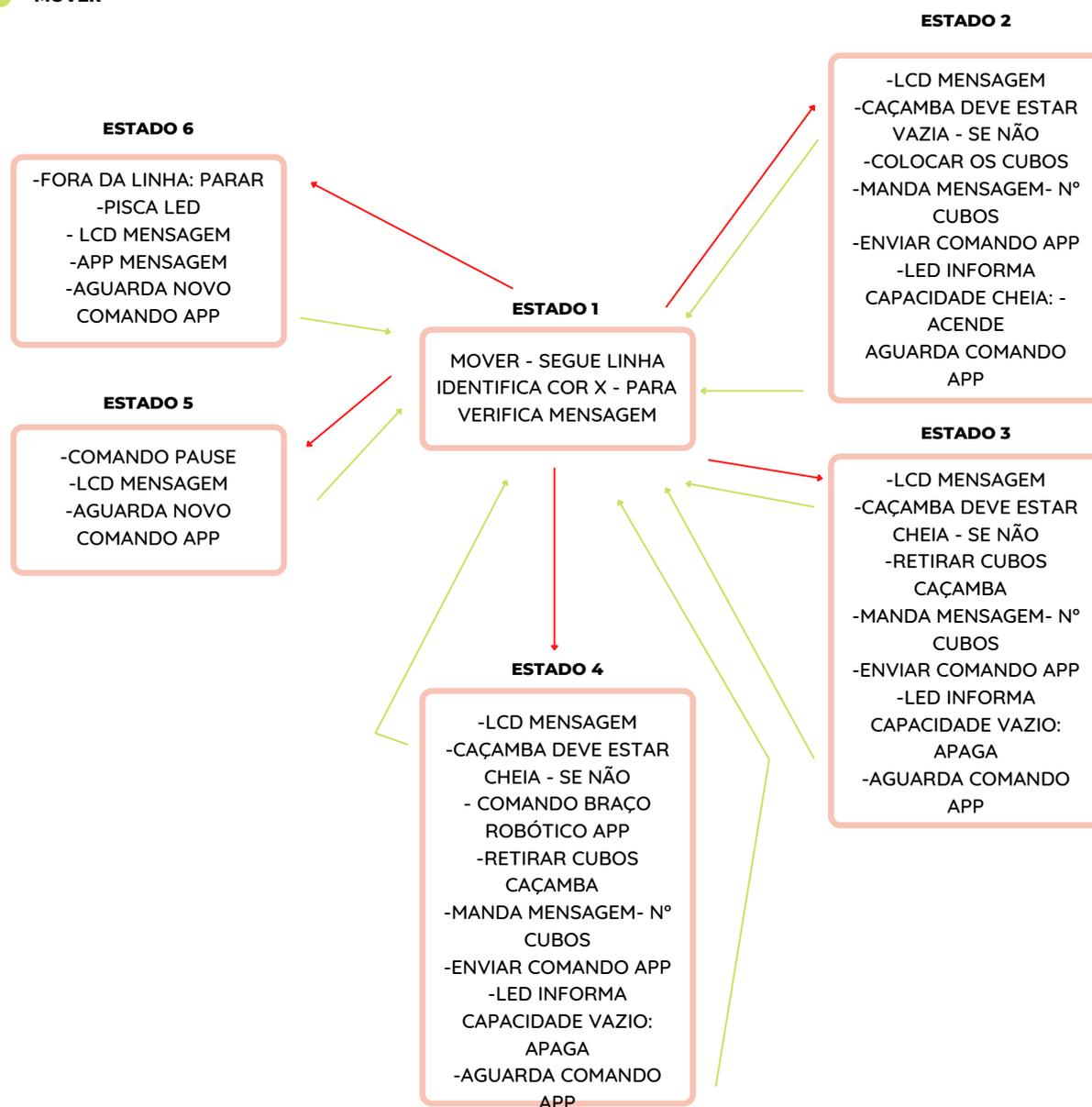
CONEXÃO DE COMPONENTES



DEFINIÇÃO CICLOS DE PROGRAMAÇÃO

A programação do robô é baseada na definição do trajeto determinado pela equipe, do qual o caminhão irá percorrer no tabuleiro. O esquema a seguir, demonstra todas as etapas estabelecidas para maior compreensão do processo de programação. Os ciclos foram divididos em 6 estados que se interagem de acordo com as situações em que o robô e o usuário se deparam.

- PARADO
- MOVER



PROGRAMAÇÃO

A programação foi feita inteiramente pelo aplicativo Arduino, onde cada código foi programado separadamente, testado e depois compilado em um só arquivo.

Abaixo, apresentamos o código separado para cada componente, sendo aqui primeiramente apresentado, o prototipo com todas as funções utilizadas no código:

```
#include <dummy.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Prototipo de funcoes
void seguirlinha();
void parar();
void detectacor();
void motor_set_up();
void avaliador();
void piscaled();
void piscaled2();
void acendeled();
void apagaled();
void lcdeme();
void lcdcol();
void lcdbio();
void lcdcoop();
void lcdpause();
void lcdinicio();
void mover(int vel_esq, int vel_dir);
void testemotor();
```

Código do motor das rodas: Com as funcoes de mover e parar já inclusas

```
// Motor A
int motor1Pin1 = 27;
int motor1Pin2 = 14;
int enable1Pin = 26;

//motor B
int motor2Pin1 = 12;
int motor2Pin2 = 13;
int enable2Pin = 25;

// Setting PWM properties
const int freq = 30000;
const int pwmChannel_1 = 0;
const int pwmChannel_2 = 1;
const int resolution = 8;
int dutyCycle1 = 200;
int dutyCycle2 = 200;

void mover(int vel_esq, int vel_dir){

    int dutyCycle_1;
    dutyCycle_1 = map( vel_dir, 0,100,0,255);
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    ledcWrite(pwmChannel_1, dutyCycle_1);

    int dutyCycle_2;
    dutyCycle_2 = map( vel_esq, 0,100,0,255);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    ledcWrite(pwmChannel_2, dutyCycle_2);
}

void motor_set_up(){
    // sets the pins as outputs:
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(enable1Pin, OUTPUT);

    // configure LED PWM functionalitites
    ledcSetup(pwmChannel_1, freq, resolution);

    // attach the channel to the GPIO to be controlled
    ledcAttachPin(enable1Pin, pwmChannel_1);

    // sets the pins as outputs:
    pinMode(motor2Pin1, OUTPUT);
    pinMode(motor2Pin2, OUTPUT);
    pinMode(enable2Pin, OUTPUT);

    // configure LED PWM functionalitites
    ledcSetup(pwmChannel_2, freq, resolution);

    // attach the channel to the GPIO to be controlled
    ledcAttachPin(enable2Pin, pwmChannel_2);
}

void parar() {
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);

    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, LOW);
}
```

Código - função: seguir linha

```
//Seguir linha 3 sensores
const int sensorS1 = 35;
const int sensorS2 = 34;
#define sensorS3 = 18;
```

(Parte do código presente na função SETUP)

```
void seguirlinha(){
  if(digitalRead(sensorS1)) //sensor direito está na
  {
    mover(70,90);
    Serial.print("\nsensor de s1: ");
    Serial.print(digitalRead(sensorS1));
  }
  if(digitalRead(sensorS2)) //sensor esquerdo está na
  {
    mover(90,70);
    Serial.print("\nsensor de s2: ");
    Serial.print(digitalRead(sensorS2));
  }
  if (!digitalRead(sensorS1)&&digitalRead(sensorS2))
  {
    mover(80,80);
    Serial.print("\n sensor de s1: ");
    Serial.print(digitalRead(sensorS1));
    Serial.print("  sensor de s2: ");
    Serial.print(digitalRead(sensorS2));

  }
  delay(1000);
}
```

Código - função: sensor de cor - detecta a cor e avalia a cor.

```
pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
// pinMode(pinLED, OUTPUT);
pinMode(out, INPUT);

digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);

Serial.begin(9600);
}
```

(Parte do código presente na função SETUP)

```
//Sensor de cor
#define pinS0 13
#define pinS1 = 12;
#define pinS2 = 26;
#define pinS3 = 27;
#define pinLED = 14;
#define pinOut = 25;*/
const int S0 = 23; //23
const int S1 = 2; //21
const int S2 = 5;
const int S3 = 18;
const int out = 4;

//Valores de leitura do sensor de cor
unsigned int valorVermelho = 0;
unsigned int valorVerde = 0;
unsigned int valorAzul = 0;
unsigned int valorBranco = 0;

void detectacor(){

//Vermelho
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);
valorVermelho = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

//Sem filtro
digitalWrite(S2, HIGH);
valorBranco = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

//Azul
digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
valorAzul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

//Verde
digitalWrite(S2, HIGH);
valorVerde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

Serial.print(" \n Azul: ");
Serial.print(digitalRead(valorAzul));
Serial.print(" Verde: ");
Serial.print(digitalRead(valorVerde));
Serial.print(" Vermelho: ");
Serial.print(digitalRead(valorVermelho));

} //Fim do void detectaCor
```

```

//Valores de variacao RGB para avaliar a cor
void avaliador() {
  // detectador(); //?????????

  //para parar cor azul (cooperativa)
if ((valorAzul<255)&&(valorVermelho<150)&&(valorVerde<150))
  estado = 2;
  //para parar cor verde (coleta residuos)
if ((valorAzul<40)&&(valorVermelho<120)&&(valorVerde<255))
  estado = 1;
  //para parar cor vermelho (biodigestor)
if ((valorAzul<50)&&(valorVermelho<255)&&(valorVerde<50))
  estado = 3;
}

```

Código - função: SETUP

```

void setup() {
  // put your setup code here, to run once:
  motor_set_up();

  lcd.init();           // initialize the lcd
  lcd.backlight();

  // Configura o pino d4 e d5 como saída para acender e apagar led
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

  //configurar sensores
  pinMode(sensorS1, INPUT);
  pinMode(sensorS2, INPUT);
  // pinMode(sensorS3, INPUT);

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  // pinMode(pinLED, OUTPUT);
  pinMode(out, INPUT);

  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

  Serial.begin(9600);
}

```

Código - função: LED

```
// Configura o pino d4 e d5 como saída para acender e apagar led  
pinMode(4, OUTPUT);  
pinMode(5, OUTPUT);
```

 (Parte do código presente na função SETUP)

```
void acendeled() {  
  
    // Configura o pino 4 e 5 como HIGH  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
}  
  
void apagaled() {  
    //Configura o pino 4 e 5 como LOW  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
}  
  
void piscaled() {  
    //Configura o pino 4 e 5 pra piscar  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
    delay(1000);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    delay(1000);  
}  
  
void piscaled2() {  
    //Configura o pino 4 e 5 pra piscar quando pausado  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
    delay(1000);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    delay(1000);  
}
```

Código - função: LCD

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,21,22); // set the LCD address to 0x27 for a 16 chars and 2 line display
```

```
lcd.init(); // initialize the lcd  
lcd.backlight();
```

(Parte do código presente na função SETUP)

```
void lcdbio(){  
  //Limpa a tela  
  lcd.clear();  
  //Posiciona o cursor na coluna 3, linha 0;  
  lcd.setCursor(3, 0);  
  //Envia o texto entre aspas para o LCD  
  lcd.print("VOCE ESTA NO");  
  lcd.setCursor(3, 1);  
  lcd.print("BIODIGESTOR");  
  delay(5000);  
  //Limpa a tela  
  lcd.clear();  
  //Posiciona o cursor na coluna 3, linha 0;  
  lcd.setCursor(2, 0);  
  //Envia o texto entre aspas para o LCD  
  lcd.print("ACIONE O BRAÇO");  
  lcd.setCursor(4, 1);  
  lcd.print("ROBOTICO");  
  delay(5000);  
}
```

```
void lcdcol(){  
  //Limpa a tela  
  lcd.clear();  
  //Posiciona o cursor na coluna 3, linha 0;  
  lcd.setCursor(3, 0);  
  //Envia o texto entre aspas para o LCD  
  lcd.print("COLETE OS");  
  lcd.setCursor(3, 1);  
  lcd.print("RESIDUOS");  
  delay(5000);  
}
```

```
void lcdpause(){
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("PAUSADO");
  delay(5000);
}
```

```
void lcdeme(){
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("ERRO!!!");
  delay(5000);
}
```

```
void lcdinicio()
{
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("ECO EH TUDO");
  delay(5000);
}
```

Código - CASE - função loop - o que o ESP32 vai ler para rodar a programação

```
void loop() {  
  
  switch (estado) {  
  case 0:  
    //Seguir linha  
    seguirlinha();  
    detectacor();  
    avaliador();  
    break;  
  case 1:  
    //Coleta de residuos  
    parar();  
    acendeled();  
    detectacor();  
    avaliador();  
    lcdcol();  
    break;  
  case 2:  
    //Cooperativa  
    parar();  
    detectacor();  
    avaliador();  
    lcdcoop();  
    apagaled();  
    break;  
  case 3:  
    //Biodigestor  
    parar();  
    detectacor();  
    avaliador();  
    lcdbio();  
    apagaled();  
    break;  
  case 4:  
    //Estado pausado/stop  
    parar();  
    piscaled2();  
    lcdpause();  
    break;  
  case 5:  
    //Emergencia  
    parar();  
    piscaled();  
    lcdeme();  
    break;  
  
  }  
}
```

Nosso case é categorizado por possíveis estados em que o caminhão se encontrará. Desde quando está em movimento, seguindo a linha, até quando está em suas paradas obrigatórias ou quando é pausado.

Dentro de cada case, as funções correspondentes são chamadas para que ele execute os comandos atribuídos à ele

Código completo, na ordem utilizado na programação

```
#include <dummy.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,21,22); // set the LCD address to 0x27 for a 16 chars and 2 line display
//variaveis
int estado=4;

//Sensor de cor
/*#define pinS0 13
#define pinS1 = 12;
#define pinS2 = 26;
#define pinS3 = 27;
#define pinLED = 14;
#define pinOut = 25;*/
const int S0 = 23; //23
const int S1 = 2; //21
const int S2 = 5;
const int S3 = 18;
const int out = 4;

//Seguir linha 3 sensores
const int sensorS1 = 35;
const int sensorS2 = 34;
/*#define sensorS3 = 18;

//Valores de leitura do sensor de cor
unsigned int valorVermelho = 0;
unsigned int valorVerde = 0;
unsigned int valorAzul = 0;
unsigned int valorBranco = 0;

// Motor A
int motor1Pin1 = 27;
int motor1Pin2 = 14;
int enable1Pin = 26;

//motor B
int motor2Pin1 = 12;
int motor2Pin2 = 13;
int enable2Pin = 25;

// Setting PWM properties
const int freq = 30000;
const int pwmChannel_1 = 0;
const int pwmChannel_2 = 1;
const int resolution = 8;
int dutyCycle1 = 200;
int dutyCycle2 = 200;

//Prototipo de funcoes
void seguirlinha();
void parar();
void detectacor();
void motor_set_up();
void avaliador();
void piscaled();
void piscaled2();
void acendeled();
void apagaled();
void lcdeme();
void lcdcol();
void lcdbio();
void lcdcoop();
void lcdpause();
void lcdinicio();
void mover(int vel_esq, int vel_dir);
void testemotor();

void setup() {
  // put your setup code here, to run once:
  motor_set_up();

  lcd.init(); // initialize the lcd
  lcd.backlight();

  // Configura o pino d4 e d5 como saída para acender e apagar led
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

  //configurar sensores
  pinMode(sensorS1, INPUT);
  pinMode(sensorS2, INPUT);
  // pinMode(sensorS3, INPUT);

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  // pinMode(pinLED, OUTPUT);
  pinMode(out, INPUT);

  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

  Serial.begin(9600);
}
```

```

void loop() {

switch (estado) {
case 0:
//Seguir linha
seguirlinha();
detectacor();
avaliacor();
break;
case 1:
//Coleta de residuos
parar();
acendeled();
detectacor();
avaliacor();
lcdcol();
break;
case 2:
//Cooperativa
parar();
detectacor();
avaliacor();
lcdcoop();
apagaled();
break;
case 3:
//Biodigestor
parar();
detectacor();
avaliacor();
lcdbio();
apagaled();
break;
case 4:
//Estado pausado/stop
parar();
piscaled2();
lcdpause();
break;
case 5:
//Emergencia
parar();
piscaled();
lcdeme();
break;

}

}

void testemotor(){
//TESTE MOTOR E SENSOR DE LINHA
mover(90,70);
delay(2000);

mover(70,90);
delay(2000);

mover(80,80);
delay(2000);
Serial.println("sensor de linha1: ");
Serial.print(digitalRead(sensorS1));
Serial.println("sensor de linha2: ");
Serial.print(digitalRead(sensorS2));
parar();
delay(2000);
}

void seguirlinha(){
if(digitalRead(sensorS1)) //sensor direito está na linha preta?
{
mover(70,90);
Serial.print("\nsensor de s1: ");
Serial.print(digitalRead(sensorS1));
}
if(digitalRead(sensorS2)) //sensor esquerdo está na linha preta?
{
mover(90,70);
Serial.print("\nsensor de s2: ");
Serial.print(digitalRead(sensorS2));
}
if (!digitalRead(sensorS1)&&digitalRead(sensorS2))
{
mover(80,80);
Serial.print("\n sensor de s1: ");
Serial.print(digitalRead(sensorS1));
Serial.print(" sensor de s2: ");
Serial.print(digitalRead(sensorS2));
}
delay(1000);
}
}

```

```

void parar() {

    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);

    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, LOW);
}

void acendeled() {

    // Configura o pino 4 e 5 como HIGH
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);
}

void apagaled() {
    //Configura o pino 4 e 5 como LOW
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
}

void piscaled() {
    //Configura o pino 4 e 5 pra piscar
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(1000);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
}

void piscaled2() {
    //Configura o pino 4 e 5 pra piscar quando pausado
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    delay(1000);
}

void lcdcoop(){

    //Limpa a tela
    lcd.clear();
    //Posiciona o cursor na coluna 3, linha 0;
    lcd.setCursor(3, 0);
    //Envia o texto entre aspas para o LCD
    lcd.print("VOCE ESTA NA");
    lcd.setCursor(3, 1);
    lcd.print("COOPERATIVA");
    delay(5000);
}

void lcdbio(){
    //Limpa a tela
    lcd.clear();
    //Posiciona o cursor na coluna 3, linha 0;
    lcd.setCursor(3, 0);
    //Envia o texto entre aspas para o LCD
    lcd.print("VOCE ESTA NO");
    lcd.setCursor(3, 1);
    lcd.print("BIODIGESTOR");
    delay(5000);
    //Limpa a tela
    lcd.clear();
    //Posiciona o cursor na coluna 3, linha 0;
    lcd.setCursor(2, 0);
    //Envia o texto entre aspas para o LCD
    lcd.print("ACIONE O BRAÇO");
    lcd.setCursor(4, 1);
    lcd.print("ROBOTICO");
    delay(5000);
}

```

```

void lcdcol(){
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("COLETE OS");
  lcd.setCursor(3, 1);
  lcd.print("RESIDUOS");
  delay(5000);
}

void lcdpause(){
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("PAUSADO");
  delay(5000);
}

void lcdeme(){
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("ERRO!!!");
  delay(5000);
}

void lcdinicio()
{
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("ECO EH TUDO");
  delay(5000);
}

void detectacor(){

  //Vermelho
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  valorVermelho = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  //Sem filtro
  digitalWrite(S2, HIGH);
  valorBranco = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  //Azul
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  valorAzul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  //Verde
  digitalWrite(S2, HIGH);
  valorVerde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  Serial.print(" \n Azul: ");
  Serial.print(digitalRead(valorAzul));
  Serial.print(" Verde: ");
  Serial.print(digitalRead(valorVerde));
  Serial.print(" Vermelho: ");
  Serial.print(digitalRead(valorVermelho));

} //Fim do void detectaCor

```

```

//Valores de variacao RGB para avaliar a cor
void avaliador() {
  // detectador(); //?????????

  //para parar cor azul (cooperativa)
  if ((valorAzul<255)&&(valorVermelho<150)&&(valorVerde<150))
    estado = 2;
  //para parar cor verde (coleta residuos)
  if ((valorAzul<40)&&(valorVermelho<120)&&(valorVerde<255))
    estado = 1;
  //para parar cor vermelho (biodegestor)
  if ((valorAzul<50)&&(valorVermelho<255)&&(valorVerde<50))
    estado = 3;
}

void mover(int vel_esq, int vel_dir){

  int dutyCycle_1;
  dutyCycle_1 = map( vel_dir, 0,100,0,255);
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  ledcWrite(pwmChannel_1, dutyCycle_1);

  int dutyCycle_2;
  dutyCycle_2 = map( vel_esq, 0,100,0,255);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  ledcWrite(pwmChannel_2, dutyCycle_2);
}

void motor_set_up(){
  // sets the pins as outputs:
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);

  // configure LED PWM functionalitites
  ledcSetup(pwmChannel_1, freq, resolution);

  // attach the channel to the GPIO to be controlled
  ledcAttachPin(enable1Pin, pwmChannel_1);

  // sets the pins as outputs:
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(enable2Pin, OUTPUT);

  // configure LED PWM functionalitites
  ledcSetup(pwmChannel_2, freq, resolution);

  // attach the channel to the GPIO to be controlled
  ledcAttachPin(enable2Pin, pwmChannel_2);
}

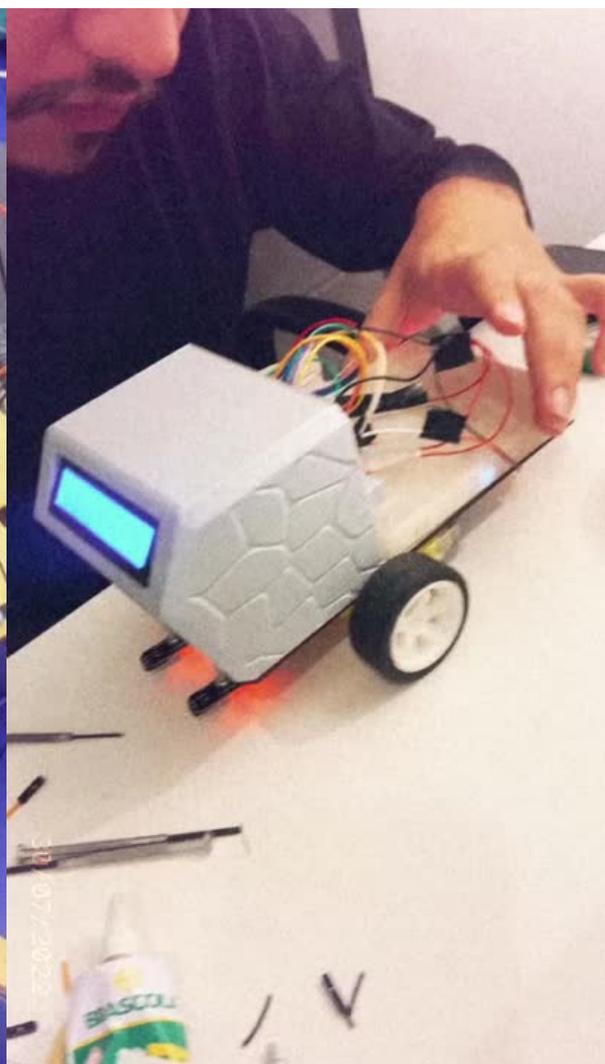
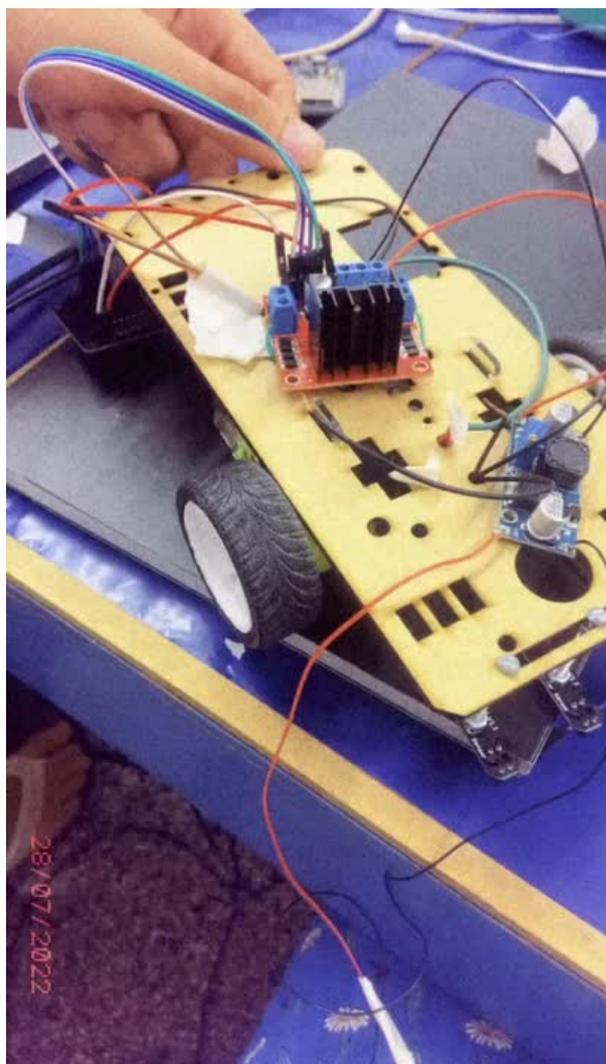
```

Para testes, desabilitávamos o case do loop e habilitávamos as funções que gostaríamos de testar, o que facilitou o processo de teste.

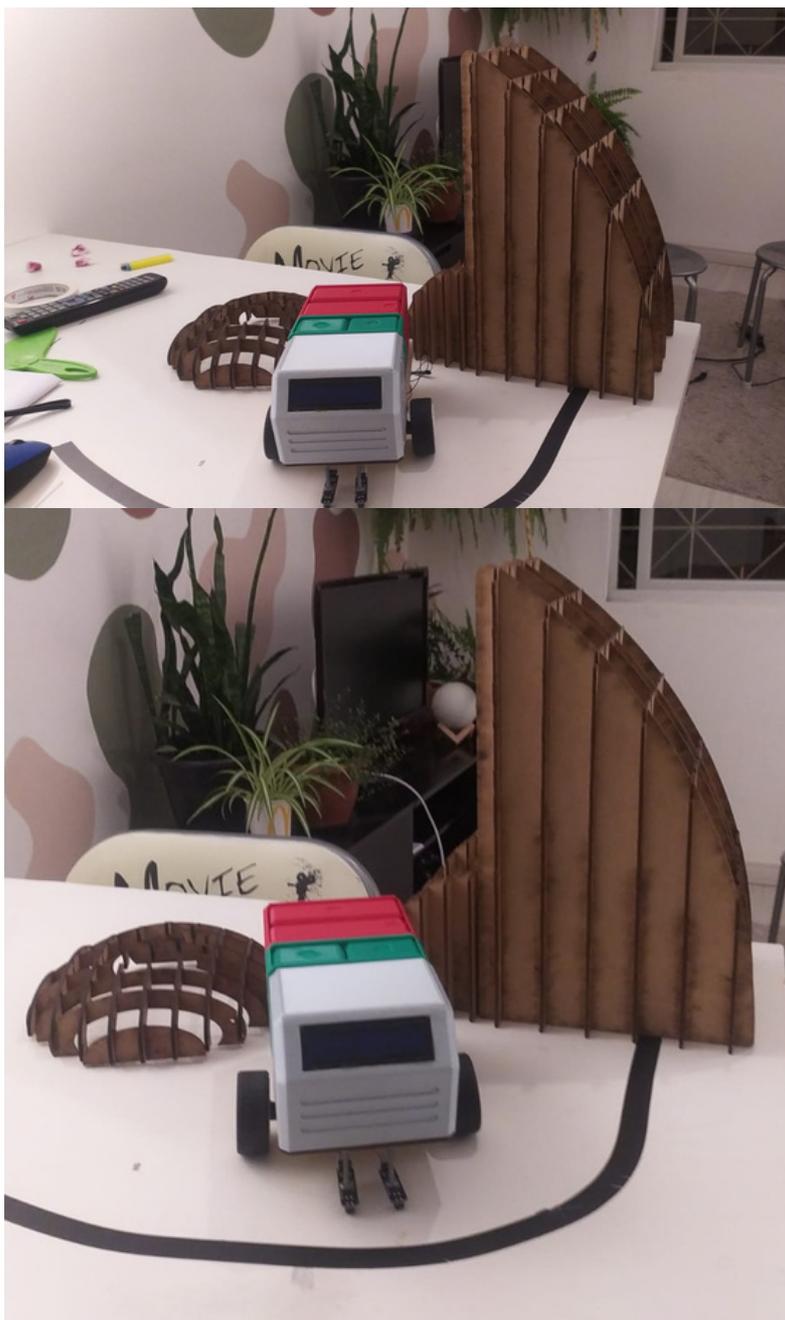
SOLDA E MONTAGEM DOS COMPONENTES



SOLDA E MONTAGEM DOS COMPONENTES

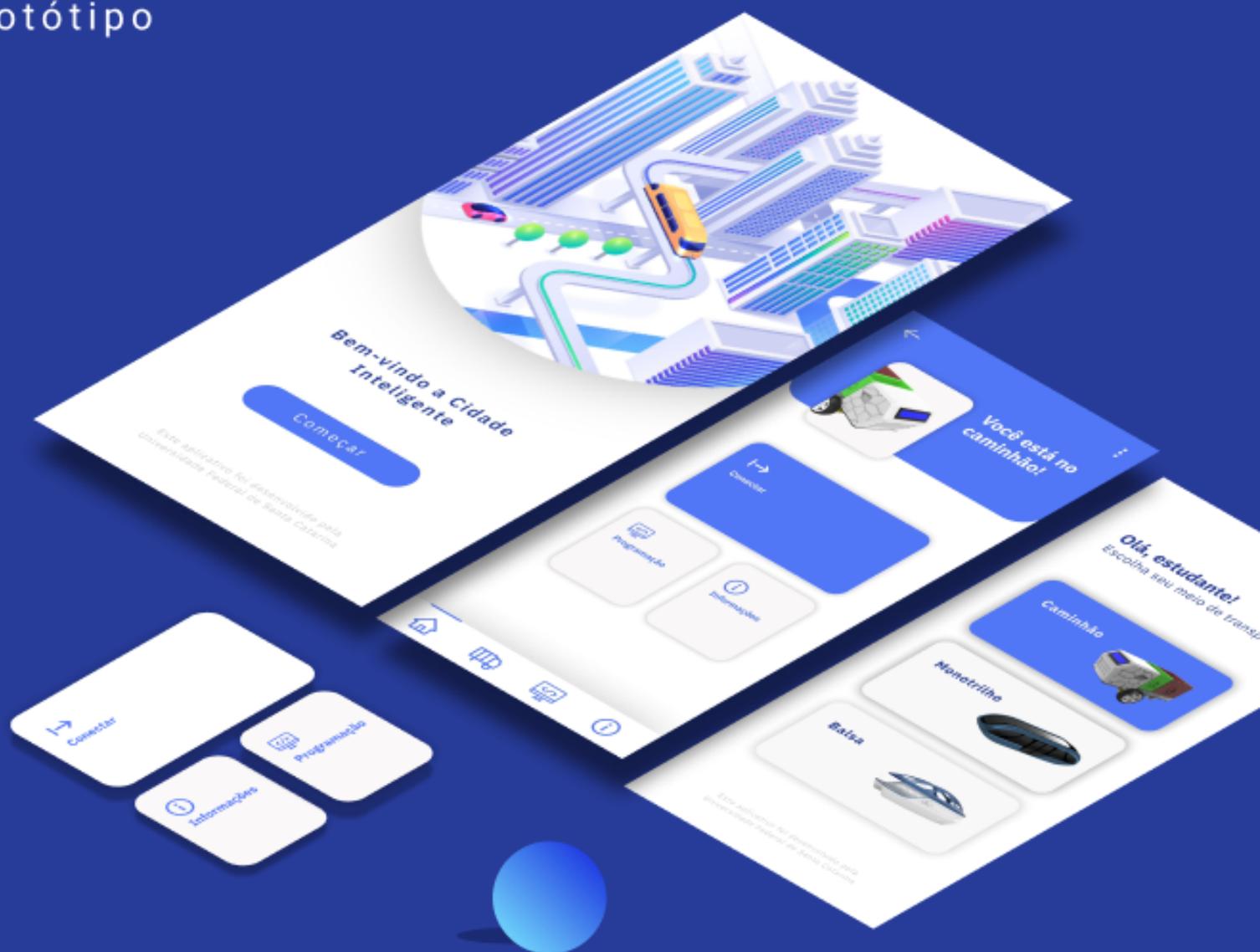


PROJETO FINAL



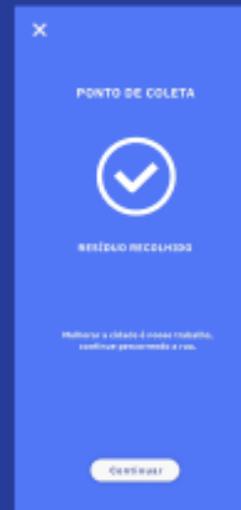
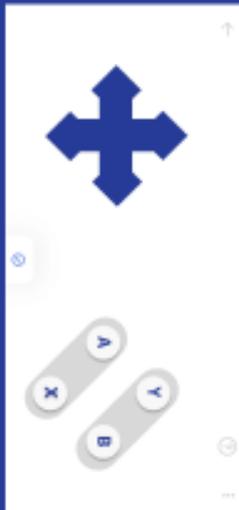
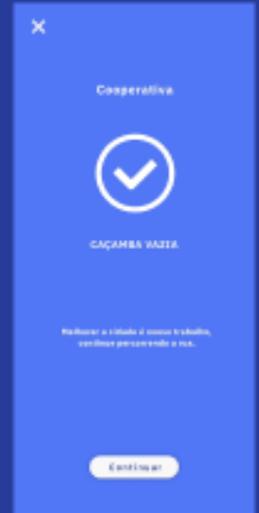
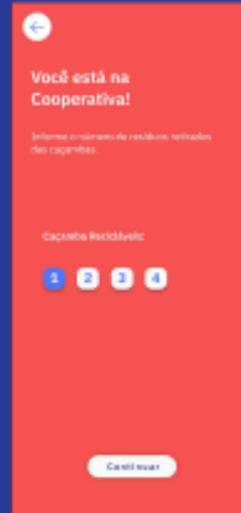
APLICATIVO

Protótipo



APLICATIVO

Conectar

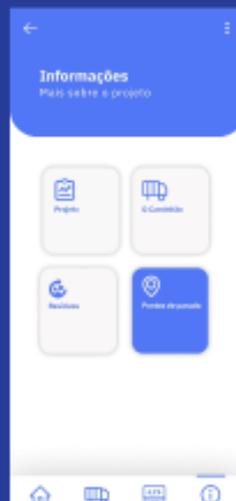
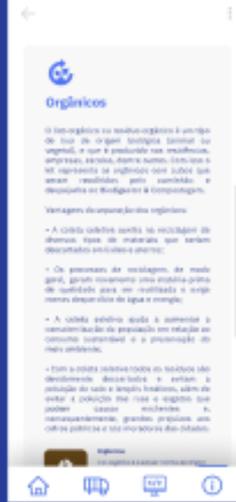
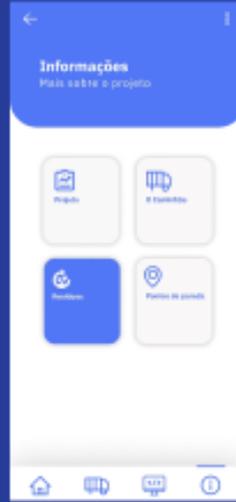
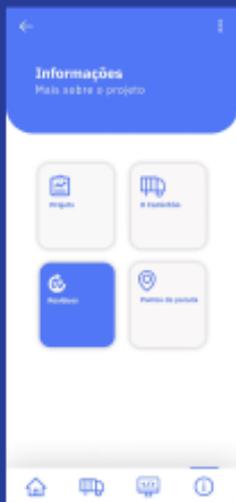
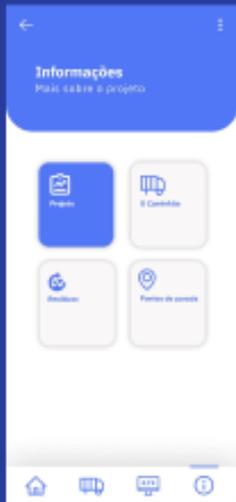


APLICATIVO

Programação



Informações



CONCLUSÃO

Apesar das dificuldades encontradas durante o percurso do projeto, concluímos com um saldo positivo, principalmente relacionado ao conhecimento e a novos repertórios adquiridos nas diversas disciplinas que compõe o módulo de Projeto 4. Devido ao curto prazo, serão necessários ajustes futuros nas conexões dos componentes e uma nova proposta de Design que contemple a identidade do projeto de acordo com os critérios discutidos durante a apresentação.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE LIMPEZA PÚBLICA E RESÍDUOS ESPECIAIS (ABRELPE). **Panorama dos resíduos sólidos no Brasil 2017**. 15. ed. São Paulo: Abrelpe, 2017. 74 p.

EIGENHEER, Emilio Maciel. **Lixo: A limpeza urbana através dos tempos**. Ed: S. Lobo. Rio de Janeiro, 2009. Acessível em <<http://www.lixoeducacao.uerj.br/imagens/pdf/ahistoriadolixo.pdf>>

GOMES, R. C. S. P. P. **Cidades sustentáveis, o conceito europeu**. Dissertação de mestrado. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Lisboa, 2009.

GONÇALVES, M.A.; TANAKA, A.K.; AMEDOMAR, A. **A destinação final dos resíduos sólidos urbanos: alternativas para a cidade de São Paulo através de casos de sucesso**. Future Studies Research Journal, v. 5, n. 1, p. 96-129, 2013

GUEDES, Anibal lopes, GUEDES Fernanda Lopes, CASTRO, Tatiana Brocardo. **Perspectivas do uso da robótica educativa na educação infantil e no Ensino Fundamental**. Universidade Federal da Fronteira Sul, Erechim RS; Instituto Federal Sul-Rio-Grandense, RS; Instituto Educacional do Rio Grande do Sul, RS. 2013 <<http://www.br-ie.org/pub/index.php/wie/article/view/2628/2282>>

OLIVEIRA, Andréa de L.; TURRA, Alexander. **Solid waste management in coastal cities: where are the gaps? case study of the north coast of são paulo**, brazil. Revista de Gestão Costeira Integrada, [S.L.], v. 15, n. 4, p. 453-465, dez. 2015. Associação Portuguesa dos Recursos Hídricos (APRH). <http://dx.doi.org/10.5894/rgci544>.

SAKUMA, Aurélio Tsuguio. **Cidades inteligentes: conceito, modelo e estratégias de desenvolvimento**. Universidade Estadual Paulista "Julio de Mesquita Filho", Bauru SP. Setembro de 2014. <<http://www.intercom.org.br/sis/2014/resumos/R9-0714-1.pdf>>