Projeto de Alta Complexidade

Projeto de Bonde

Giuliana Nicanor
Juliana Bauer
Maria Eduarda Hanoff

Inicialmente já foi vista a necessidade do redesign apresentar alterações estéticas para adequar-se à estética futurista, menos lúdica e mais realista.

Após observarmos os projetos do semestre anterior, também identificamos a necessidade do aumento do modelo, para comportar os componentes.

Revisão das Análises

Funcional & Estrutural

Foram observados os elementos que compõem um bonde em vida real e a forma como geralmente são traduzidos para brinquedos, observando seus detalhamentos.

Revisão das Análises

Funcional & Estrutural

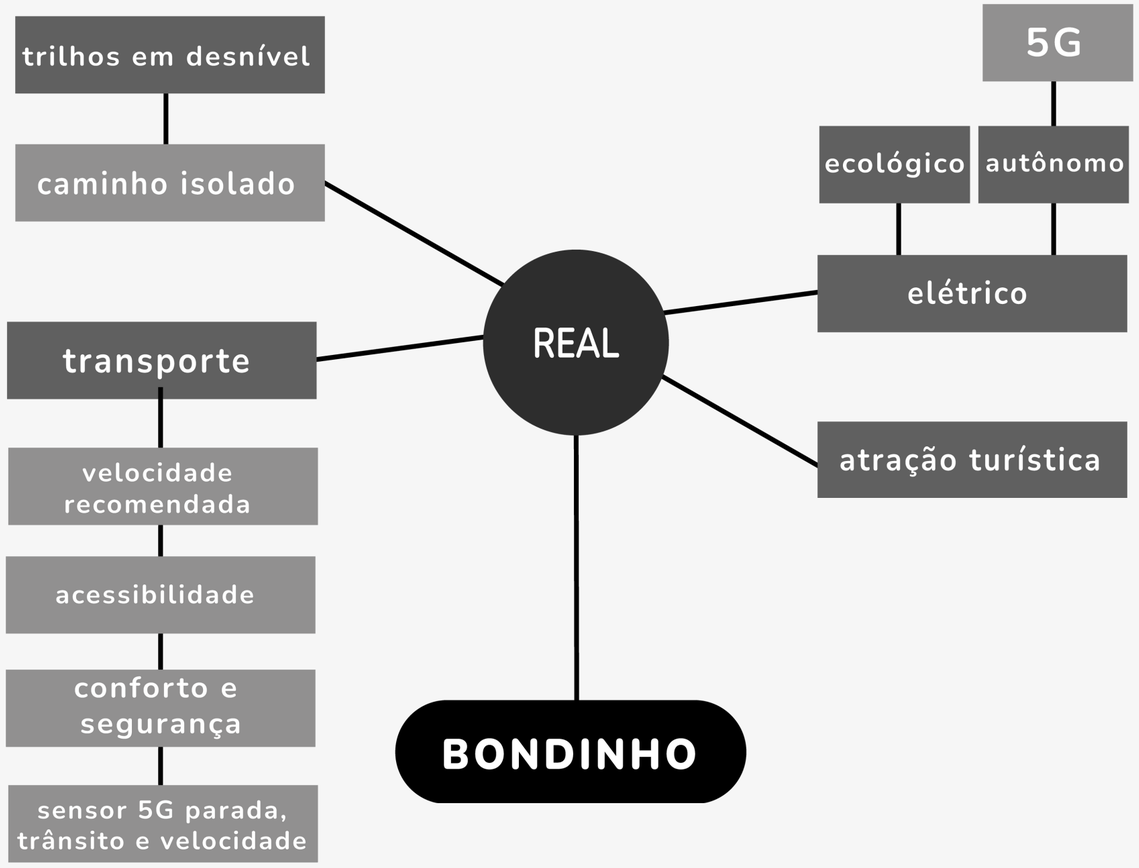
Portas em ambos os lados ao longo de seu comprimento;

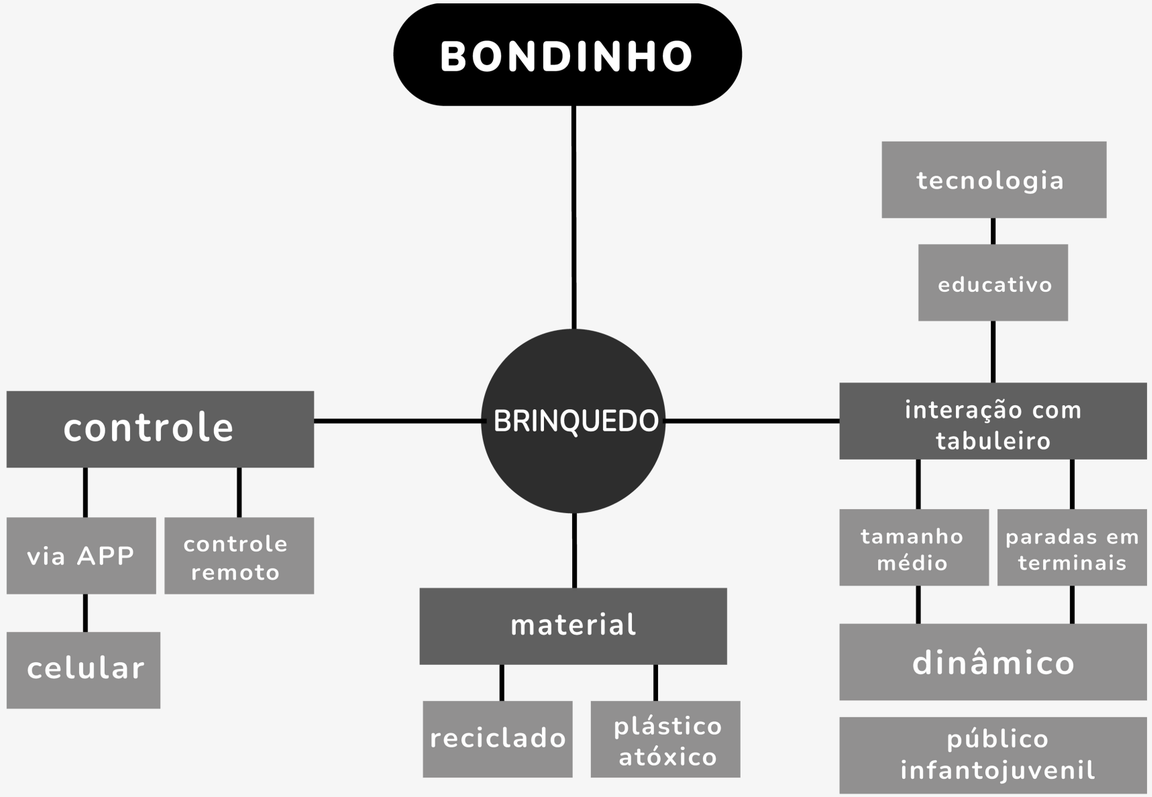
Visor indicando sua direção ou bairro;

Faróis;

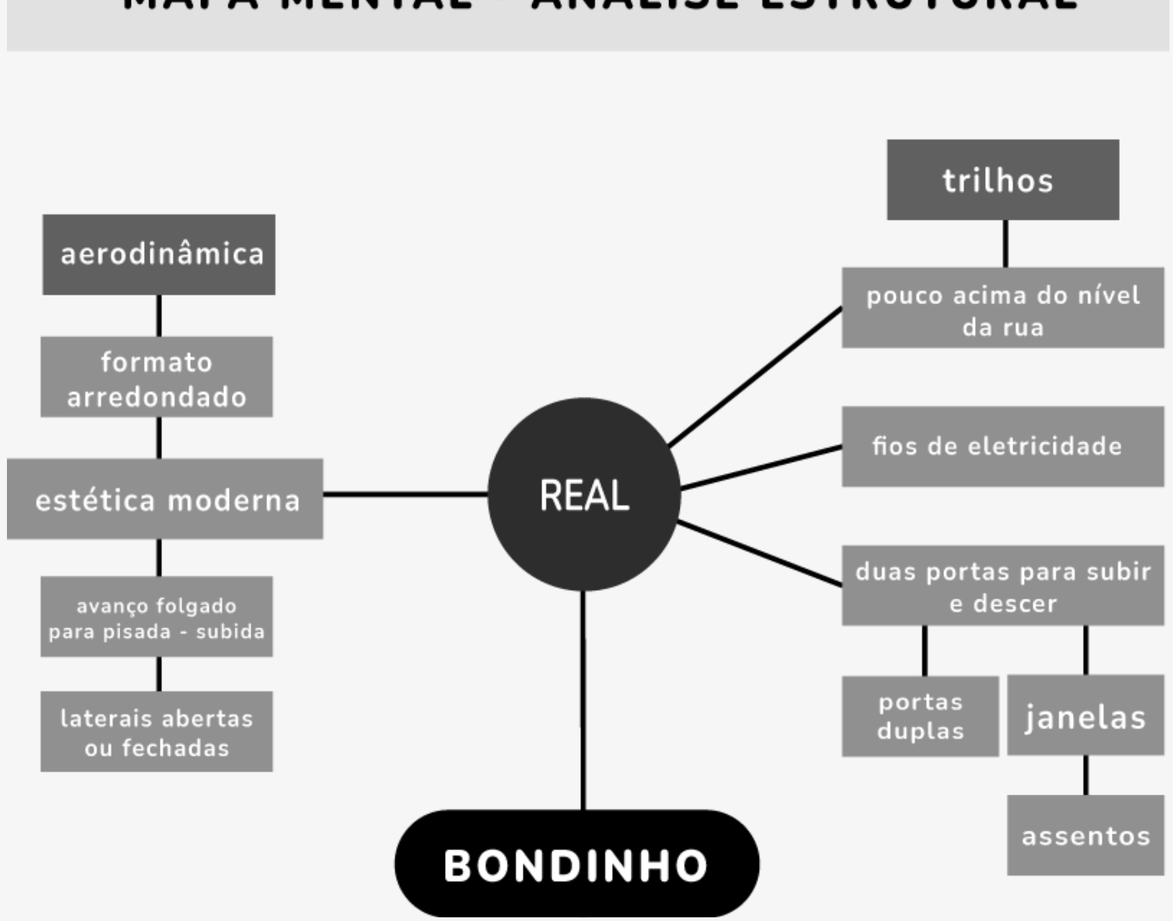
Grandes janelas e vidros, deixando o automóvel com aspecto limpo e tecnológico.

MAPA MENTAL - ANÁLISE FUNCIONAL





MAPA MENTAL - ANÁLISE ESTRUTURAL



Benchmarking

Aplicativo

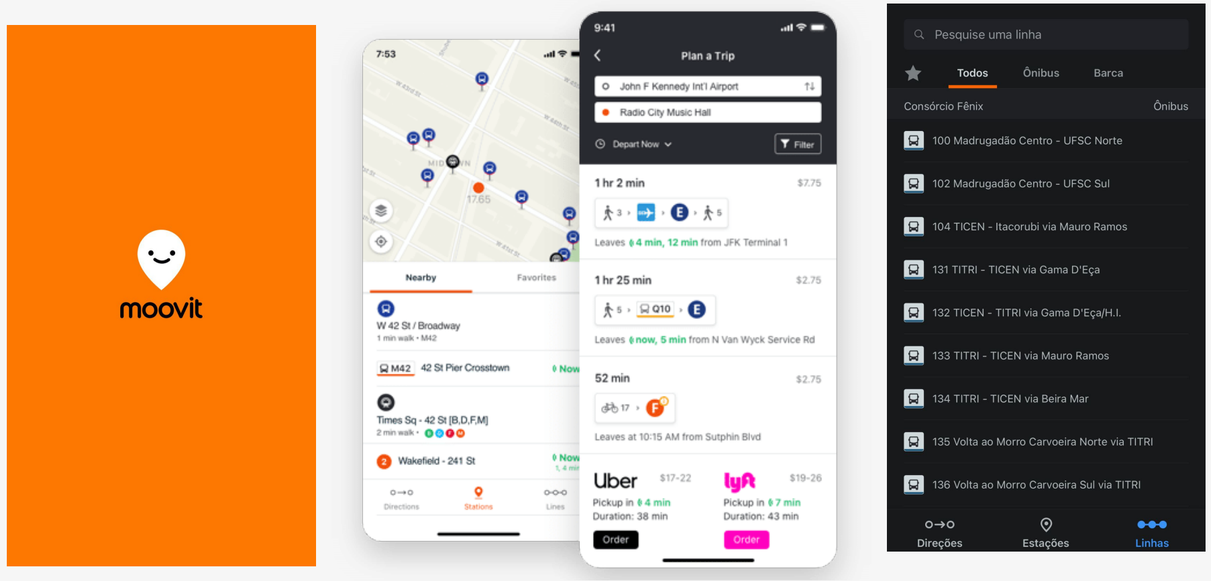
Foi realizada uma pesquisa a respeito de aplicativos de transporte e controle, pontuando suas forças e fraquezas.

1 Moovit

Opção de procurar por linhas, as estações existentes ou direções (ponto inicial até a onde o usuário deseja chegar;

Diferenciação de cada linha de ônibus com os números e nomes referentes a cada linha existente;

Da as opções de todos os tipos de transporte possíveis como também de "favoritar" os preferidos de acordo com o usuário;



Fraquezas

Apenas busca pelas linhas de ônibus, sem ligação direta com pagamento e/ou tickets

Forças

Fácil acesso à busca e procura do seu destino;

Estações fáceis de localizar através do mapa e pelo GPS do celular;

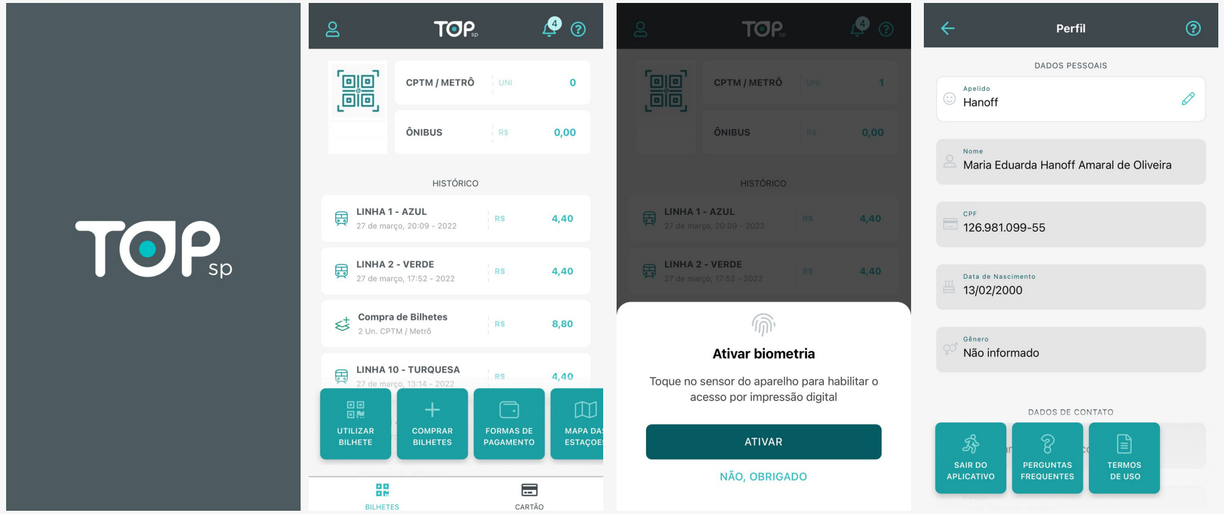
Mostra diversas rotas e linhas para o mesmo destino, facilitando a escolha;

Através das rotas é adicionado também o tempo estimado até o destino

2 TOP (metro SP)

Botões com finalidades diferentes tem mais destaque pois são telas diferentes;

Medida de proteção contra fraudes e compras sem intenção, com biometria para logar;



Fraquezas

Em metrô é difícil o acesso à internet e sinal, e como a maioria dos aplicativos ele não possui função offline

Forças

Possível abrir o mapa das estações e linhas;

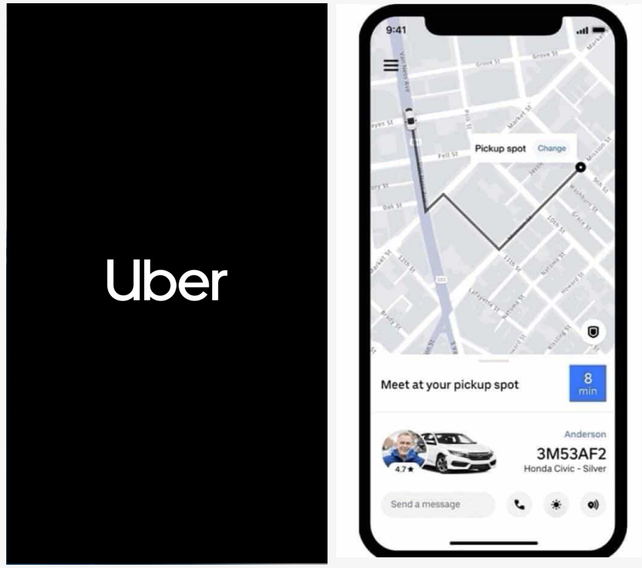
Compra rápida e prática de bilhetes do metrô e ônibus;

Pagamento via leitura por QR Code na estação de metrô, facilitando o controle de quantidade de tickets e mais rápido no fluxo de passageiros

3 Uber

Mostra a rota como também o tempo necessário para o transporte chegue ao ponto final, mostrando a semelhança com outros apps do mesmo nicho;

identificações diferentes para cada transporte e que seja mais fácil de localiza-lo



Fraquezas

App exclusivamente para transporte, sem variedade de serviços.

Forças

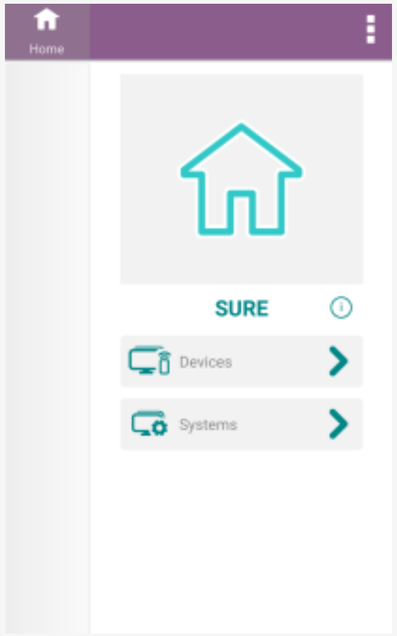
Localização em tempo real;

Estímulo de interações por meio de pop-ups, como avaliação de motoristas;

Tempo estimado até a chegada ao destino.

4 Sure Universal Remote

Botões laterais que diferenciam qual o aparelho o usuário quer controlar e ao centro inferior, onde ficam os comandos gerais mais referentes a um controle remoto de TV (pois o comando da esquerda está selecionado o da "LG Smart TV")



Fraquezas

Tela com muita informação;

Algumas informações são desnecessárias quando já escolhida determinada função, podendo ser escondidas em um menu, por exemplo;

Densidade informacional, com pouca hierarquia informacional

Forças

Fácil acesso e usabilidade;

Botões com ícones simples e de geral conhecimento do público alvo;

Ajustes simples de controle (ligar, desligar, menu, enter, etc...)

Requisitos

Requisitos de projeto- aplicativo

Categoria	Classificação	Requisito	Objetivos	Tipo
Estética	Cores	Paleta de cores já presente no aplicativo	Aparência moderna	Obrigatório
	Formas	Simples, componentes com cantos arredondados	Aparência moderna	Obrigatório
Estrutura	Navegação	Facilitada com conteúdo educativo	Aplicativo mais intuitivo e de fácil uso	Obrigatório

Categoria	Classificação	Requisito	Objetivos	Tipo
Função	Abas	Informativo sobre o projeto; Lista de componentes; Programação; Trajeto	Seguir os padrões do aplicativo	Desejável
	Conteúdo	Didático e completo a respeito do projeto	Acessível para o usuário	Obrigatório

Requisitos

Requisitos de projeto - bonde

Foi feita uma adaptação da matriz realizada no P3, eliminando, adaptando e adicionando ideias

Em preto: conceitos mantidos
Em vermelho: conceitos eliminados
Em verde: conceitos adicionados

Categoria	Classificação	Requisito	Objetivos	Fonte	Tipo
Estrutura	Material	Produção no PRONTO 3D, da UFSC	PLA de impressão 3D	Análise sincrônica e análise estrutural	Obrigatório
	Forma	Geometria orgânica, com continuidade nas formas	Formas arredondadas que remetem à estética futurista	Análise sincrônica e mapa mental	Obrigatório
	Tamanho	Condizente com o tabuleiro	Escala aprox. 16X6X8cm Escala aproximada de 25X10X10	Análise sincrônica e estrutural	Obrigatório
	Componentes	Janelas amplas; Portas móveis	Simulação de acesso facilitado ao bonde e ensino de acessibilidade	Análise sincrônica, lista de verificação e mapa mental	Desejável
	Peso	Manejo grosseiro com pega de produtos médios	Produto leve, em plástico, de pequeno ou médio porte	Análise sincrônica, lista de verificação e mapa mental	Obrigatório
	Interior	Assentos com bom espaço; Espaço para componentes Encaixes para os componentes Fechamento com ímã	Assentos em escala com altura adequada, Painel estilo conversível Encaixes para organização interna Ímã para abertura e fechamento- acesso	Análise sincrônica e estrutura	Desejável

Categoria	Classificação	Requisito	Objetivos	Fonte	Tipo
Função	Controles	Movimento e acionamento das luzes via aplicativo	Uso pelo aplicativo	Diagrama, análise sincrônica, lista de verificação, mapa mental	Desejável
	Funções Automáticas	Paradas, curvas e emissão de áudio	Ensino de tecnologia e formulação de produto inteligente	Diagrama, análise sincrônica, lista de verificação, mapa mental	Desejável
	Ensino de Energia Limpa Ensino Tecnológico	Utilização ou representação de painéis solares Pleno funcionamento do bonde	Informações adicionais pelo aplicativo	Diagrama, lista de verificação	Desejável
	Sensores	De movimento e feedback sonoro	Identificação de paradas do bonde e retorno do trajeto	Diagrama, análise sincrônica, infográfico	Desejável
	Manuseio	Manejo grosseiro com pegas de produtos médios	Pequenas partes acopladas, aumentando o armazenamento e contato físico da criança	Diagrama, análise sincrônica, lista de verificação, mapa mental	Obrigatório
	Paradas pela Cidade/Tabuleiro	Simulação de paradas pelo Centro Histórico e UFSC Pelos projetos do tabuleiro	Movimento e paradas através de sensores adicionais	Infográfico, lista de verificação	Obrigatório

Categoria	Classificação	Requisito	Objetivos	Fonte	Tipo
Estética	Cores	Branco, prata, preto, cinza e detalhes coloridos	Aparência moderna e inovadora; Futurista e robotizada	- Análise sincrônica, Mapa mental, Análise estrutural	Obrigatório
	Formas	Cantos arredondados e encaixes Maior presença de reflexos e texturas	Estética mais futurista e orgânica, também contribuindo com a funcionalidade	- Análise sincrônica, Mapa mental, Análise estrutural e funcional	Obrigatório

Conceitos

1

Futurista

2

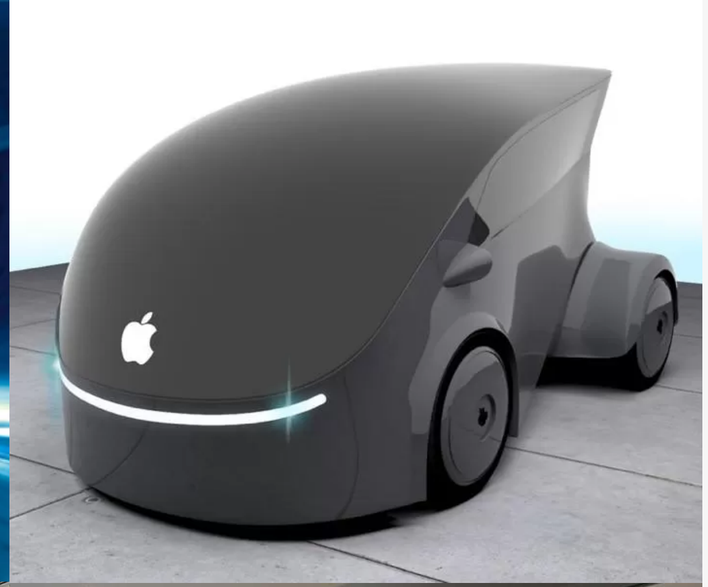
Interativo

3

Funcional

Futurista

- Formas orgânicas
- Minimalismo
- Elegância
- Funcionalidade

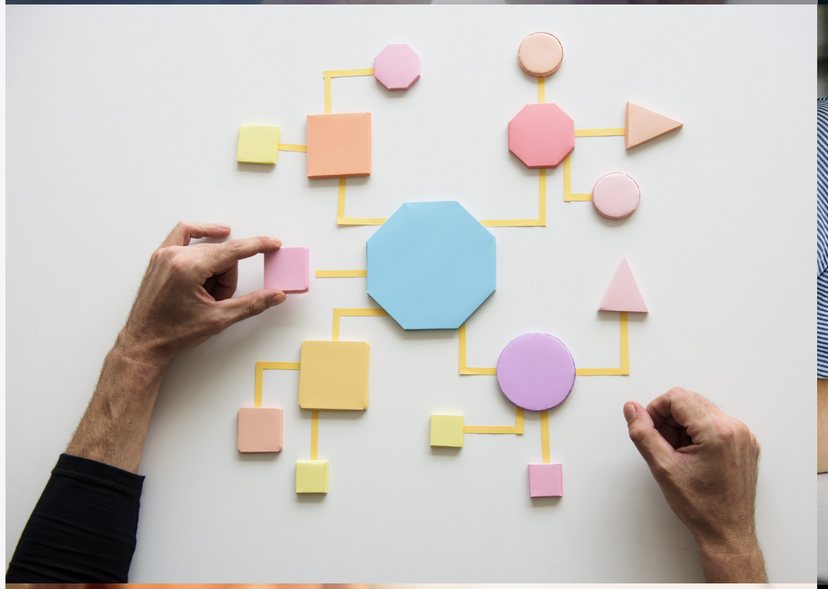


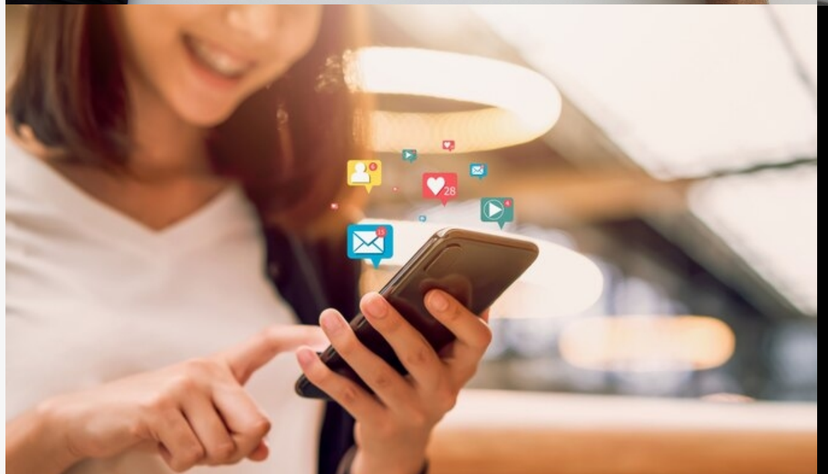
Interativo

Acessível

Fluxo Interativo

Organização de ideias



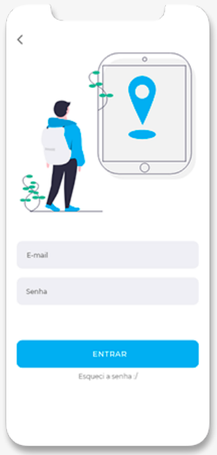


Funcional

Organização de dados

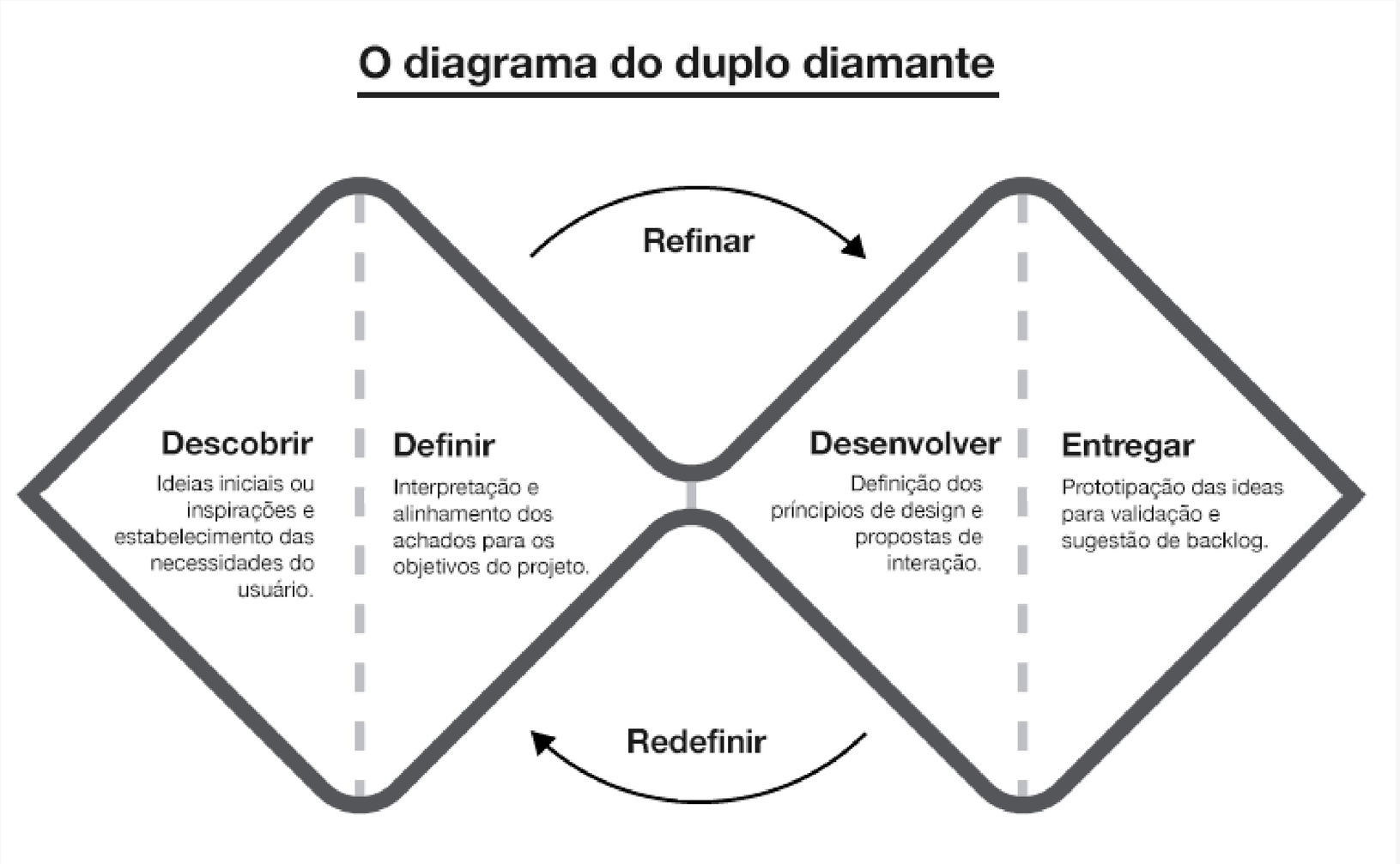
Facilitador de atividades

Fácil adaptação do usuário



Criação

Técnicas criativas



Definição das Funções

Bonde pelo tabuleiro

1. Exibição do destino no Display LCD;
2. Mover-se pelo tabuleiro;
3. Emitir sons designados para cada parada;
4. Parar em locais pré-determinados (estação e totens);
5. Acender e apagar faróis.

Desafios

Modelo físico

Adaptar o trajeto e posicionamento dos totens no tabuleiro;

Realizar um redesign mais futurístico e realista, condizente com o tabuleiro;

Definir os sensores de parada e como seriam acionados no tabuleiro

Redefinir todas as medidas do projeto, observando a necessidade de maior espaço para os componentes;

Diferenciar-se do monotrilho já presente no tabuleiro

Desafios

Aplicativo

Criar fluxos interativos e acessíveis

Integrar novas ideias à interface pré montada

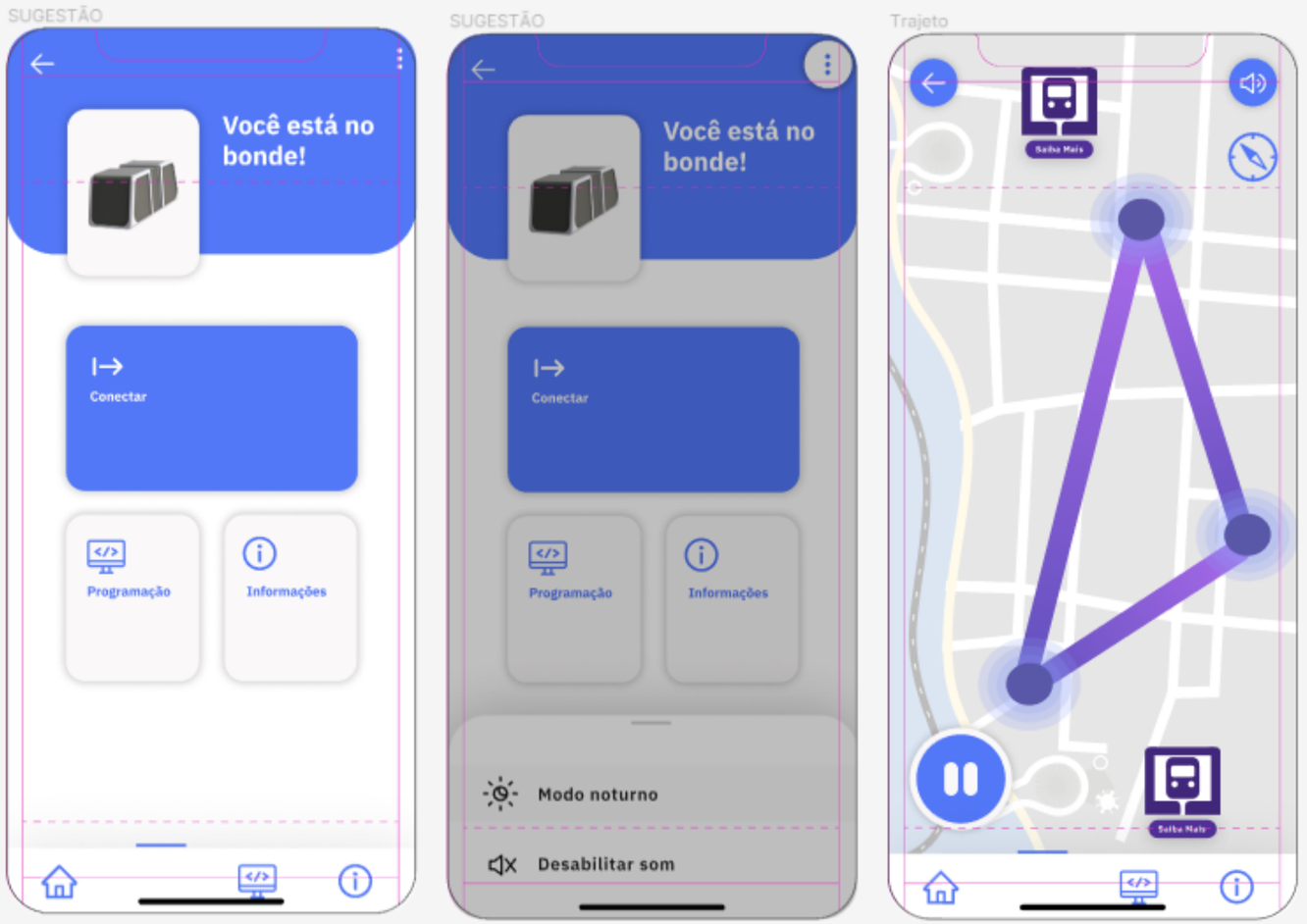
Minimizar ideias e simplificar conteúdos

Ilustrar ícone que se diferencie do monotrilho, facilmente reconhecível

Desafios

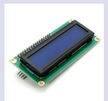
Aplicativo

Aplicativo desconsiderando a zona de toque dos celulares, deixando muitos ícones não clicáveis



Componentes

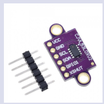
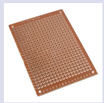
Verificação das funções x componentes

	Power Bank (x1)	Alimentação de bateria
	Roda 40mm (x2)	Locomoção
	DISPLAY I2C 16x2 (x1)	Visor indicando o destino no tabuleiro
	Rodizio com Esfera Metálica (x2)	Auxílio na movimentação do bonde

	Módulo Seguidor De Linha Infravermelho Tcrt5000 - 3 Canais	Direcionamento para locomoção no tabuleiro
	Motor DC 3-6V (2x)	Funcionamento das rodas
	Módulo Sensor de Cor TCS3200 e Reconhecimento TCS230	Distinção de cada parada no tabuleiro
	Ponte H L298n	Conexão para funcionamento dos dois motores DC

Componentes

Verificação das funções x componentes

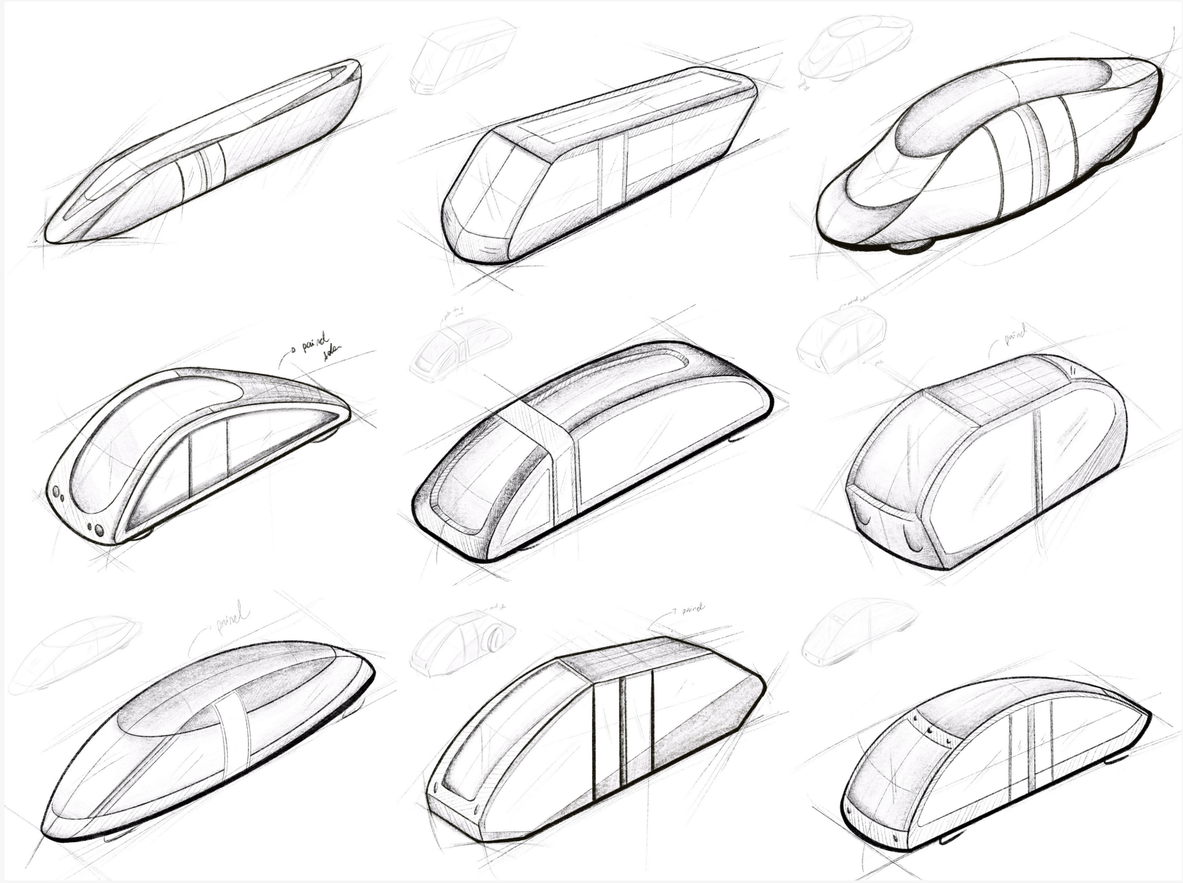
	MiniDF Player	Reprodução do áudio nas paradas
	Micro SD	Armazenamento do áudio
	Adaptador USB	Usado para conectar o power bank com os outros componentes
	Barra de Pinos 1x40 Fêmea 180 Graus (2x)	Ajuda na conexão dos componentes
	Sensor de Distância Laser - VL53L0X	Evitar colisões frontais
	Ímã Neodímio Ø 4x1,5 mm N35 50 unidades	Fechamento do bonde
	Placa Fenolite Perfurada tamanho 7x9cm	Conexão dos componentes
	Jumpers Fêmea-Fêmea x40 Unidades	Conexão dos componentes entre si e com o ESP32
	Mini Alto Falante 0.5W 8R 40mm	Reprodução do áudio

Geração de Alternativas

Sketch

Na primeira versão projetual, a geração foi marcada por formas mais orgânicas e arredondadas, que posteriormente se pareceu semelhante ao monotrilho.

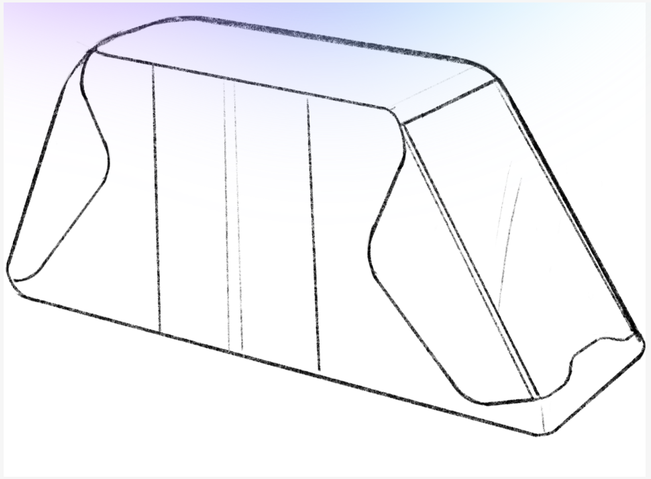
Com isso, foi preciso achar um equilíbrio entre a forma arredondada, porém não muito acentuada.

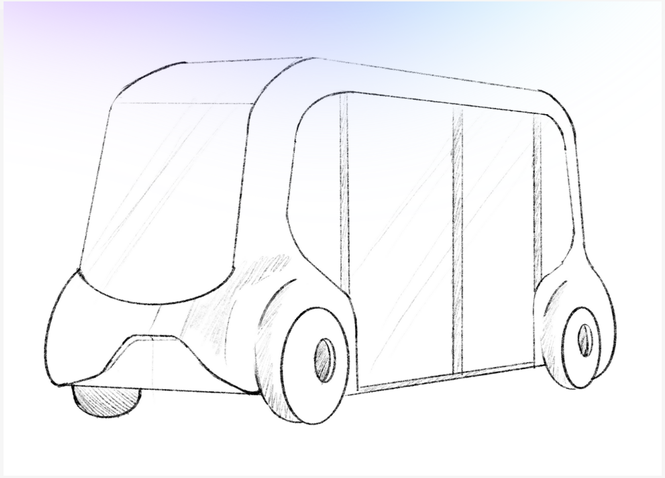


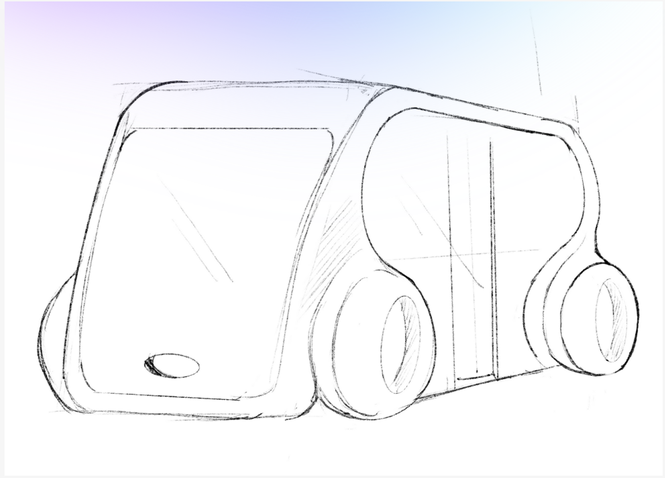
Versões criadas em Projeto de Produto 3

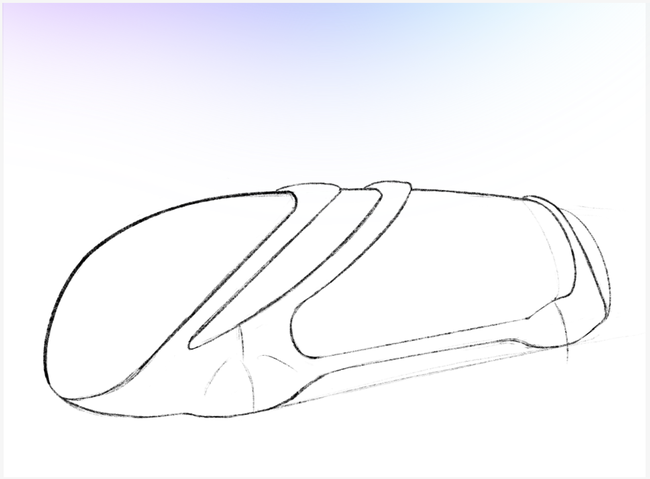
Geração de Alternativas

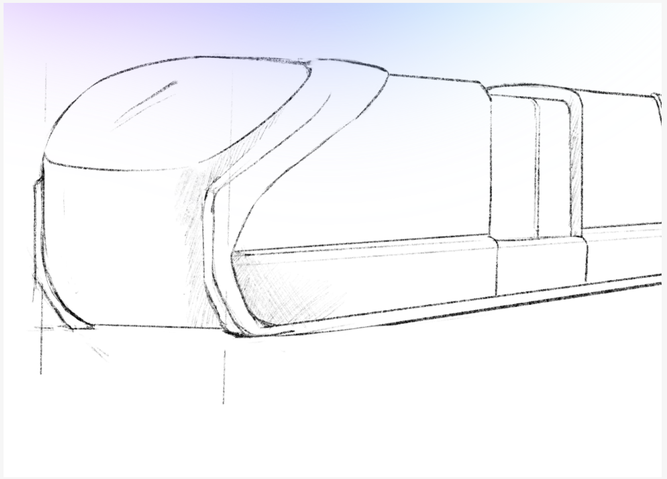
Sketch

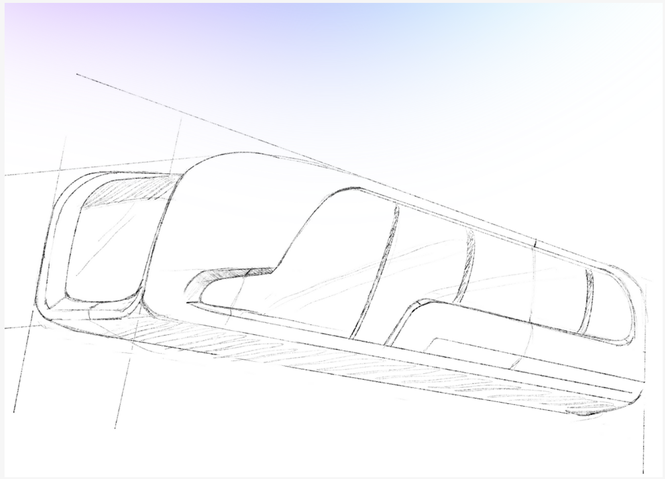












Geração de Alternativas

Modelo físico de baixa fidelidade

A fim de melhor compreender as proporções do projeto, realizamos um modelo de baixa fidelidade em papelão

Foram utilizadas como medidas base, as medidas do bonde de P3, ampliando-as tendo em vista a necessidade de maior espaço para os componentes.

Inspirações e referências

Soluções

Melhoramentos do P3

Proporções

Aumento de altura, largura e comprimento

Levando em consideração:

O tamanho dos outros projetos do tabuleiro, de forma a ficar proporcional;

A altura do trilho do monotrilho.

Estética

Formas mais arredondadas e orgânicas;

Maior presença de vidro e superfícies "reflexivas" ou brilhantes;

Funcionamento

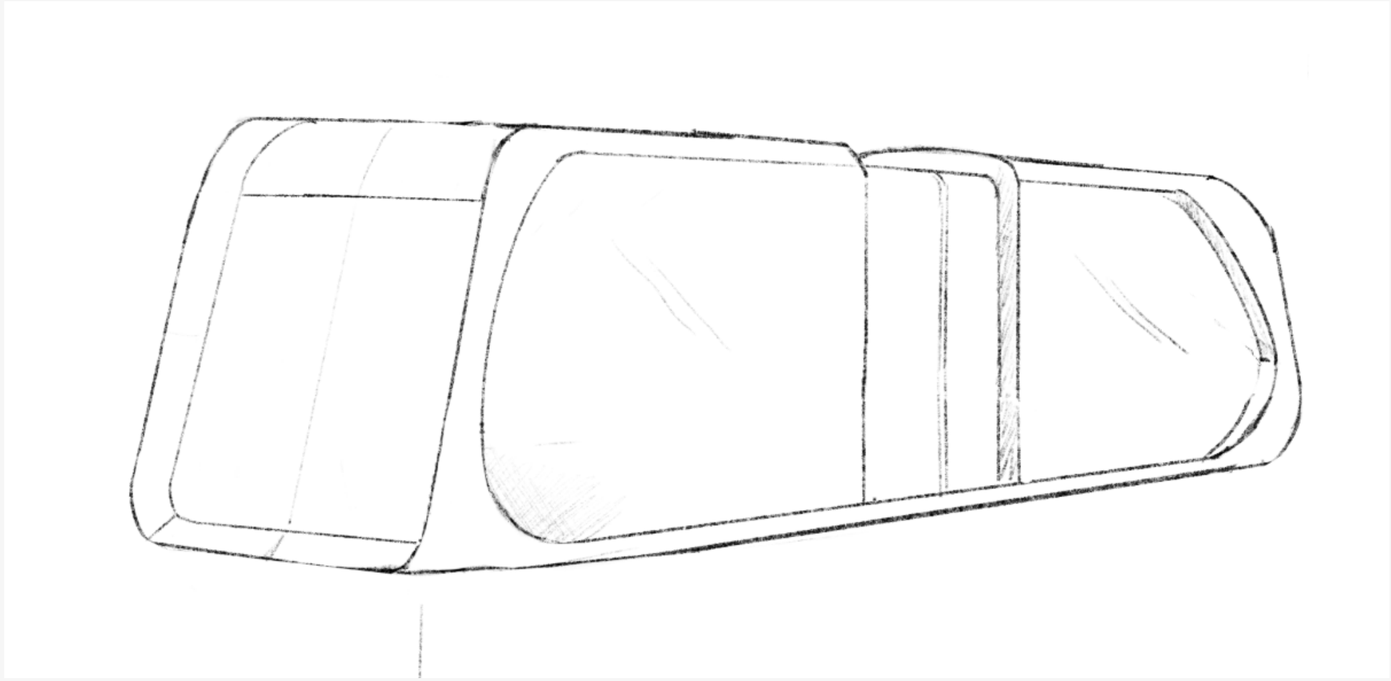
Sensor de cor posicionado na parte inferior do bonde, onde fara a leitura das quatros fitas coloridas coladas no tabuleiro na frente de cada parada;

Sensor de laser a fim de evitar colisões com outros componentes do tabuleiro;

Uso de um power bank para alimentação, ao invés de pilhas.

Soluções

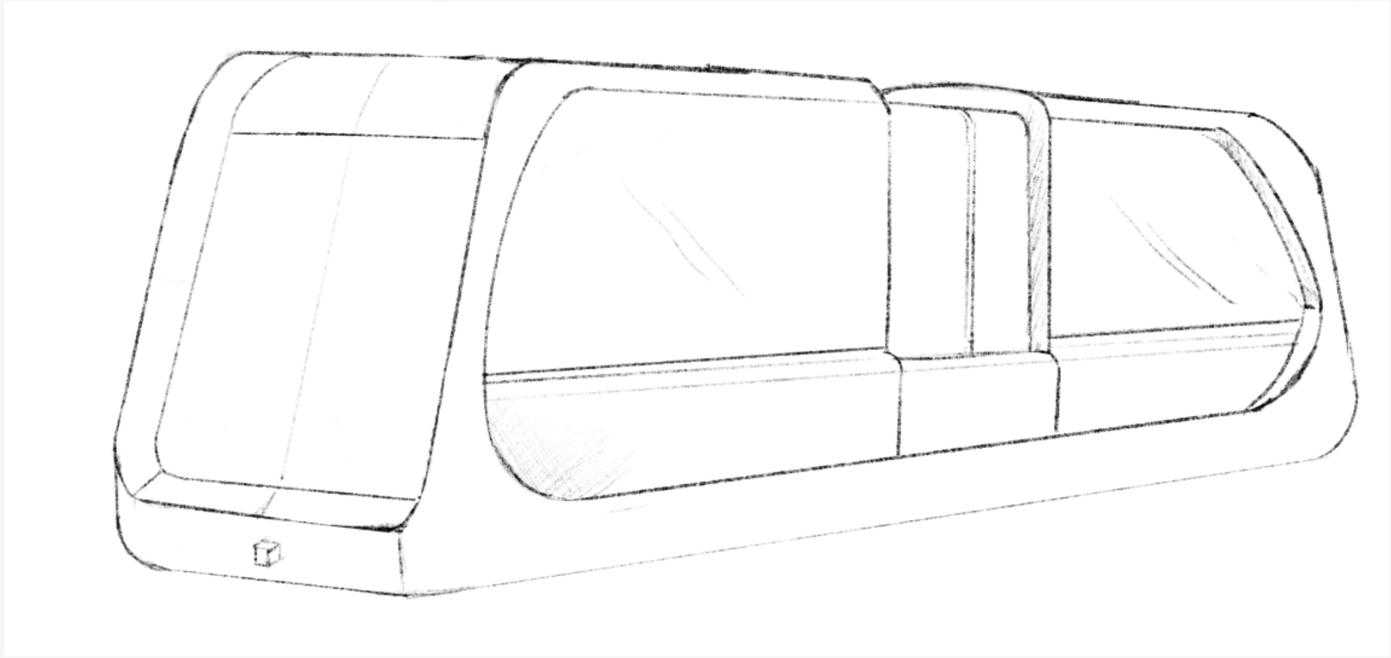
Escolhas definidas



Mistura da estética dos sketches anteriores, buscando os detalhes mais interessantes de cada um e juntando em apenas uma solução.

Soluções

Escolhas definidas



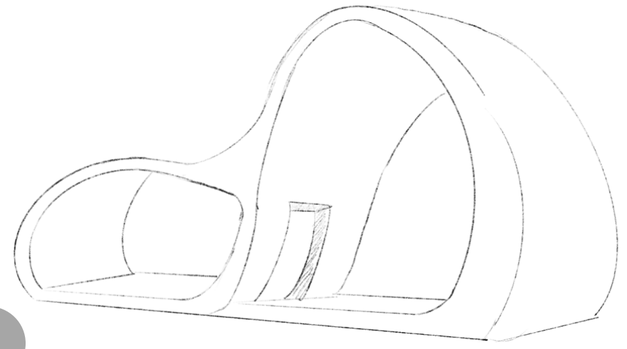
Mesma versão, porém com um rebaixo para a possibilidade de esconder o motor e as rodas sob o chão.

Soluções

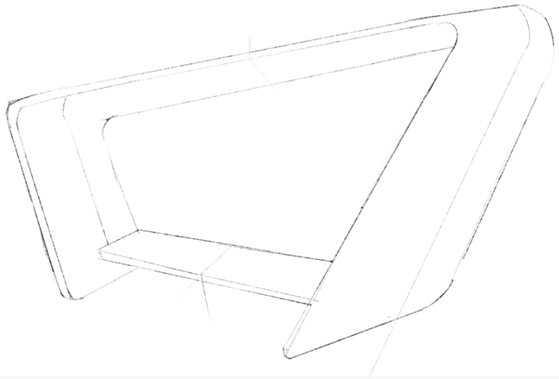
Parada do bonde

Ideias para a criação da parada do bonde, totalizando em 4 paradas em todo o tabuleiro.

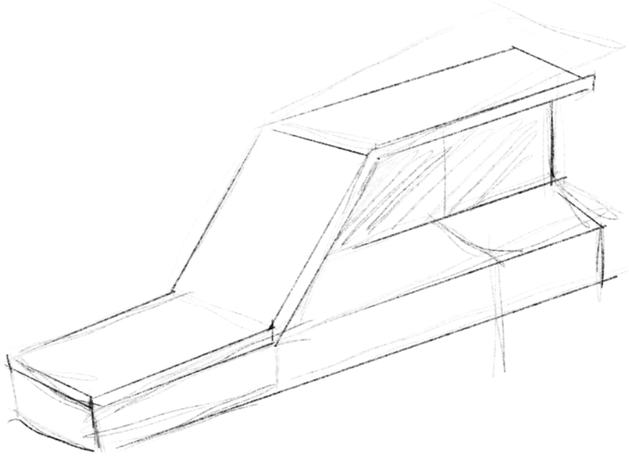
1



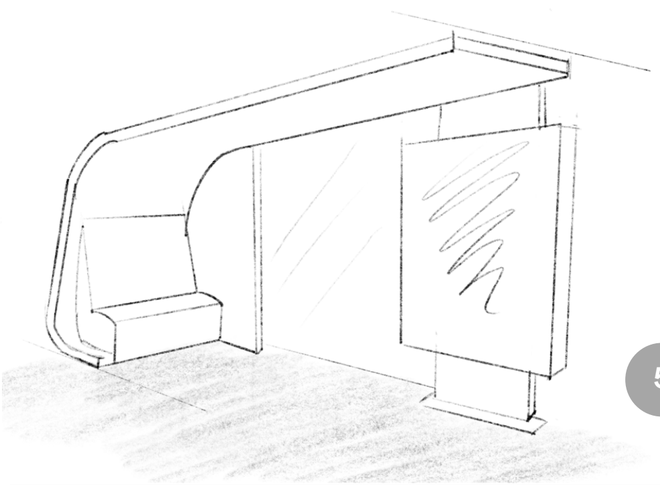
4



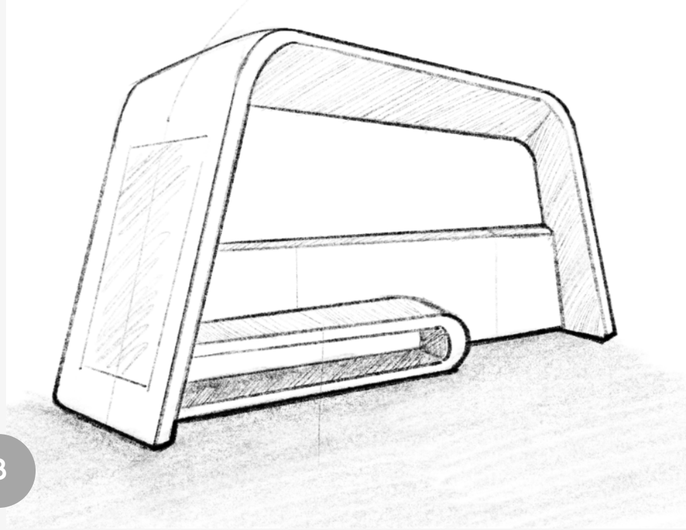
2



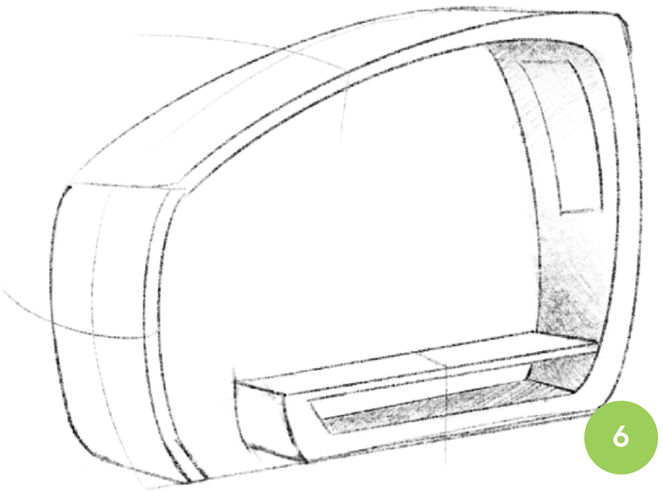
5



3



6



6 *Solução definida.

Máquina de Estados

Soluções de interação

Estado 1 (Normal):

1.1 Seguir linha ( );
1.2 Letreiro - nome com a próxima parada ( );

Estado 2 (Totem/parada):

2.1 Parar de acordo com o totem via cor ( );
2.1 Som/voz - avisando a parada e respectivo ponto ( );
2.1 Acender setas - da direta frente e trás ( );
2.2 Apagar setas da direita e aguardar 4seg ( );
2.3 Volta a andar e seguir linha ( );
2.3 Acender setas da esquerda;
2.4 Apagar setas da esquerda;
2.4 Letreiro - nome com a próxima parada ( );

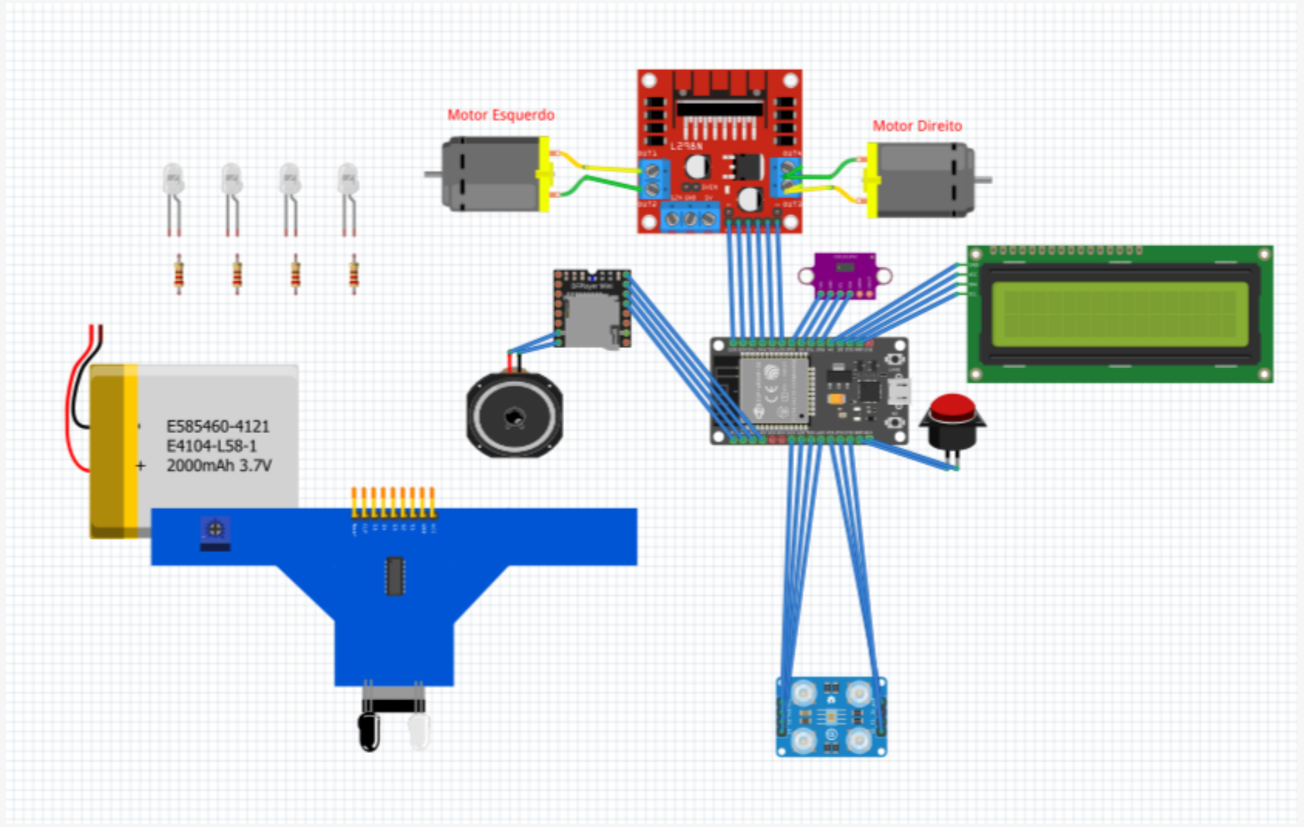
Estado 3 (Estação):

3.1 Parar de acordo com o sensor de cor na estação ( );
3.1 Acender setas da esquerda ( );
3.1 Som/voz - avisando a parada na estação ( );
3.2 Apagar setas da esquerda e aguardar 6seg ( );
3.3 Volta a andar e seguir linha ( );
3.3 Acender setas da esquerda ( );
3.4 Apagar setas da esquerda ( );
3.4 Letreiro - nome com a próxima parada ( );

Depois retorna ao primeiro estado.

Conexão dos Componentes

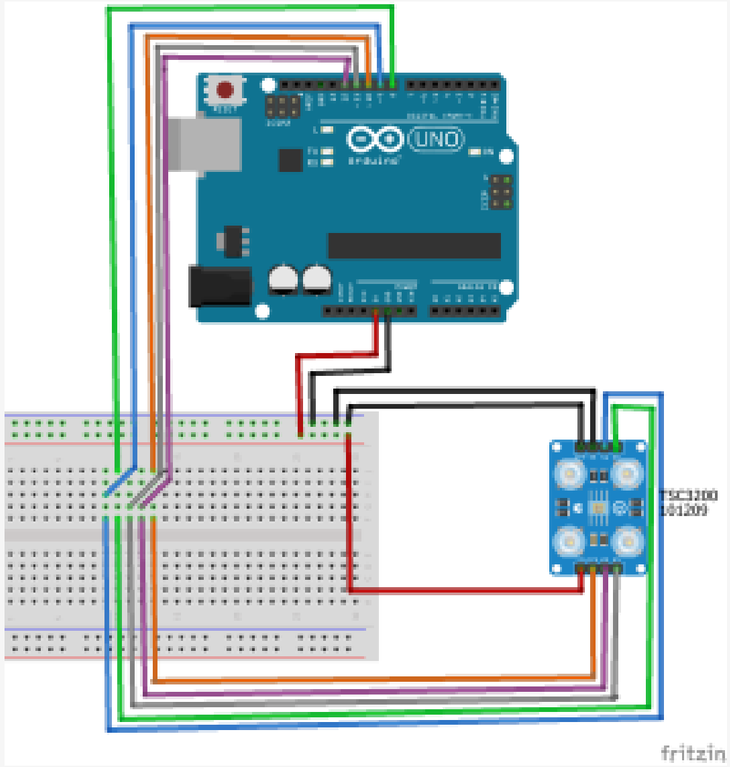
Soluções de hardware



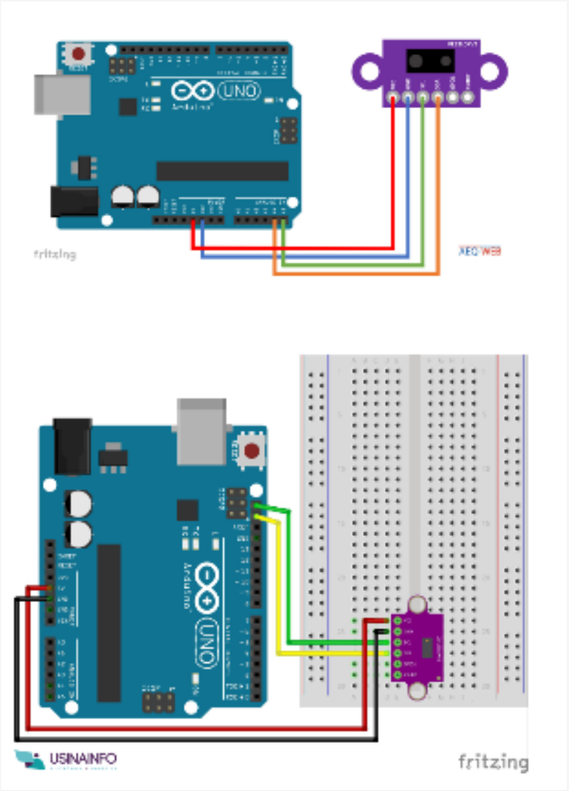
Conexão dos Componentes

Soluções de hardware

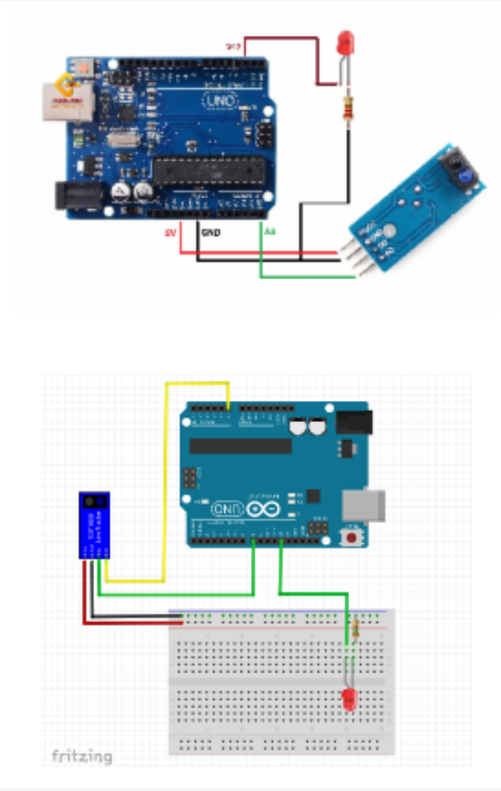
Sensor de cor



Sensor de distância



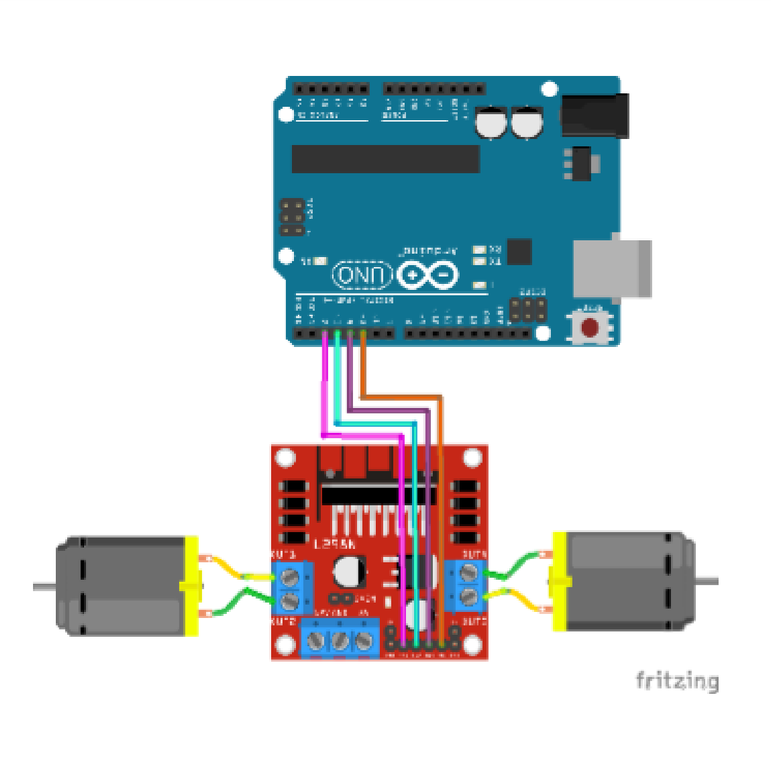
Módulo seguidor de linha



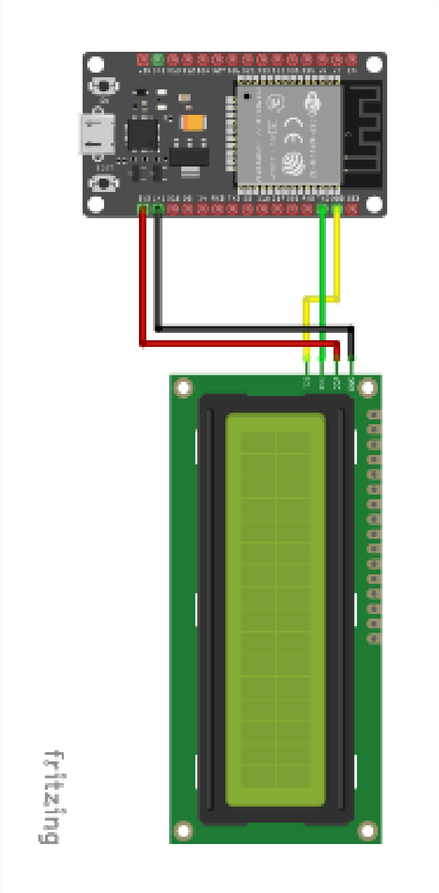
Conexão dos Componentes

Soluções de hardware

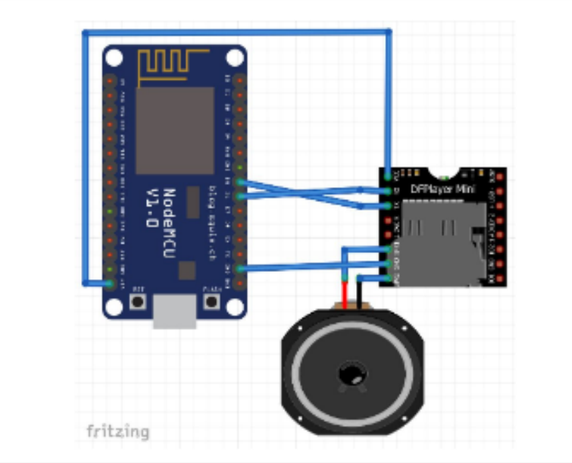
Ponte H

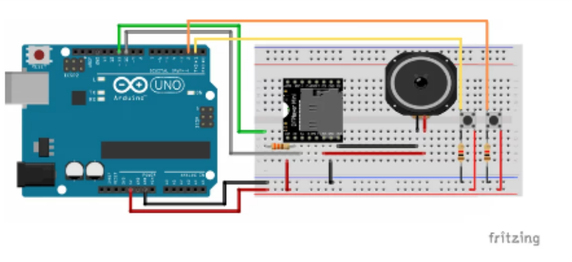


Display I2C



Mini DF Player





Código dos Componentes

Foram compilados exemplos de códigos dos componentes usados, compreendendo sua lógica de funcionamento, facilitando a futura etapa de programação.

Código dos Componentes

Sensor de cor

```
//Pins de ligação ao Arduino
const int s0 = 8;
const int s1 = 9;
const int s2 = 12;
const int s3 = 11;
const int out = 10;

//Variaveis das cores
int ver = 0;
int verde = 0;
int azul = 0;

void setup()
{
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  Serial.begin(9600);
  digitalWrite(s0, HIGH);
  digitalWrite(s1, LOW);
}
```

```
void loop()
{
  //Detecta a cor
  cor();
  //Mostra valores no serial monitor
  Serial.print("Vermelho :");
  Serial.print(ver, DEC);
  Serial.print(" Verde : ");
  Serial.print(verde, DEC);
  Serial.print(" Azul : ");
  Serial.print(azul, DEC);
  Serial.println();

  //Verifica se a cor vermelha foi detetada
  if (ver < azul && ver < verde && ver < 100)
  {
    Serial.println("Vermelho");
  }

  //Verifica se a cor azul foi detetada
  else if (azul < ver && azul < verde && azul < 1000)
  {
    Serial.println("Azul");
  }
```

```
  //Verifica se a cor verde foi detetada
  else if (verde < ver && verde < azul)
  {
    Serial.println("Verde");

  }
  Serial.println();

  //Espera para a proxima leitura e reiniciar o processo
  delay(50);
}
void cor()
{
  //Rotina para leitura das cores RGB
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  //Regista o valor da cor vermelha na variavel ver
  ver = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
  //Regista o valor da cor azul na variavel azul
  azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH)
  digitalWrite(s2, HIGH);
  //Regista o valor da cor verde na variavel verde
  verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH
}
```

Código dos Componentes

Sensor de distância

```
#include "Adafruit_VL53L0X.h"

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup()
{
Serial.begin(115200);

// wait until serial port opens for native USB devices
while (! Serial)
{
delay(1);
}

Serial.println("Adafruit VL53L0X test");
if (!lox.begin())
{
Serial.println(F("Failed to boot VL53L0X"));
while(1);
}
// power
Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
}

void loop()
{
VL53L0X_RangingMeasurementData_t measure;

Serial.print("Reading a measurement... ");
lox.rangingTest(&measure, false); // pass in 'true' to get debug data prin
```

```
if (measure.RangeStatus != 4)
{ // phase failures have incorrect data
Serial.print("Distance (mm): "); Serial.println(measure.RangeMilliMeter);
}
else
{
Serial.println(" out of range ");
}

delay(100);
}
```

Código dos Componentes

Módulo seguidor de linha

```
#define M1AA 10 //Definição do pino que controla o motor A
#define M1AB 11 //Definição do pino que controla o motor A
#define M2BA 5  //Definição do pino que controla o motor B
#define M2BB 6  //Definição do pino que controla o motor B

#define pinS1 7 //Definindo o pino 7 como pino do primeiro sensor
#define pinS2 8 //Definindo o pino 8 como pino do segundo sensor

bool Sensor1 = 0; //Declarando a variavel "Sensor1" e atribuindo a ela o valor "0"
bool Sensor2 = 0; //Declarando a variavel "Sensor2" e atribuindo a ela o valor "0"

int velocidade = 125; //Declarando a variavel "velocidade" e atribuindo a ela o valor "125"

void setup(){
  pinMode(M1AA, OUTPUT); // Define o pino de controle do motor A como saída
  pinMode(M1AB, OUTPUT); // Define o pino de controle do motor A como saída
  pinMode(M2BA, OUTPUT); // Define o pino de controle do motor B como saída
  pinMode(M2BB, OUTPUT); // Define o pino de controle do motor B como saída


  digitalWrite(M1AB, LOW);  // Setamos a direção inicial do motor 1 como 0, isto fará com que o motor gire para frente
  digitalWrite(M2BB, LOW);  // Setamos a direção inicial do motor 2 como 0, isto fará com que o motor gire para frente

  pinMode(pinS1, INPUT); // Define o pino do sensor 1 como entrada
  pinMode(pinS2, INPUT); // Define o pino do sensor 2 como entrada


}

void loop(){
  Sensor1 = digitalRead(pinS1); // A variavel "Sensor1" recebe o valor digital lido pelo sensor 1
  Sensor2 = digitalRead(pinS2); // A variavel "Sensor2" recebe o valor digital lido pelo sensor 2

 if((Sensor1 == 0) && (Sensor2 == 0)) // Os dois lados detectaram branco
 {
   analogWrite(M1AA, velocidade); //O motor A recebe velocidade 125
   analogWrite(M2BA, velocidade); //O motor B recebe velocidade 125
   delay(25);
 }
 if((Sensor1 == 1) && (Sensor2 == 0)) // O primeiro sensor detectou preto e o segundo branco
 {
   analogWrite(M1AA, 0); //O motor A recebe velocidade 0
   analogWrite(M2BA, 140); //O motor B recebe velocidade 140, girando assim o carrinho
   delay(25); //Espera de 25 milissegundos
 }
 if((Sensor1 == 0) && (Sensor2 == 1)) // O primeiro sensor detectou branco e o segundo preto
 {
   analogWrite(M1AA, 140); //O motor A recebe velocidade 140, ficando assim ligado
   analogWrite(M2BA, 0); //O motor B recebe velocidade 0, desligando-o e fazendo assim o carrinho virar no outro sentido
   delay(25); //Espera de 25 milissegundos
 }
}
```

Código dos Componentes

Ponte H

```

//declaracao dos pinos utilizados para controlar a velocidade de rotacao
const int PINO_ENA = 6;
const int PINO_ENB = 5;

//declaracao dos pinos utilizados para controlar o sentido do motor
const int PINO_IN1 = 4;
const int PINO_IN2 = 3;
const int PINO_IN3 = 8;
const int PINO_IN4 = 7;

int i = 0; //declaracao da variavel para as rampas

const int TEMPO_ESPERA = 1000; //declaracao do intervalo de 1 segundo entre os sentidos de rotacao do motor

const int TEMPO_RAMPA = 30; //declaracao do intervalo de 30 ms para as rampas de aceleracao e desaceleracao

void setup() {

  //configuração dos pinos como saida
  pinMode(PINO_ENA, OUTPUT);
  pinMode(PINO_ENB, OUTPUT);
  pinMode(PINO_IN1, OUTPUT);
  pinMode(PINO_IN2, OUTPUT);
  pinMode(PINO_IN3, OUTPUT);
  pinMode(PINO_IN4, OUTPUT);
```

```

  //inicia o codigo com os motores parados
  digitalWrite(PINO_IN1, LOW);
  digitalWrite(PINO_IN2, LOW);
  digitalWrite(PINO_IN3, LOW);
  digitalWrite(PINO_IN4, LOW);
  digitalWrite(PINO_ENA, LOW);
  digitalWrite(PINO_ENB, LOW);

}

void loop() {

  //configura os motores para o sentido horario
  digitalWrite(PINO_IN1, LOW);
  digitalWrite(PINO_IN2, HIGH);
  digitalWrite(PINO_IN3, LOW);
  digitalWrite(PINO_IN4, HIGH);

  //rampa de aceleracao
  for (i = 0; i < 256; i=i+10){
    analogWrite(PINO_ENA, i);
    analogWrite(PINO_ENB, i);
    delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
  }

  //rampa de desaceleracao
  for (i = 255; i >= 0; i=i-10){
    analogWrite(PINO_ENA, i);
    analogWrite(PINO_ENB, i);
    delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
  }
```

```

  //configura os motores para o sentido anti-horario
  digitalWrite(PINO_IN1, HIGH);
  digitalWrite(PINO_IN2, LOW);
  digitalWrite(PINO_IN3, HIGH);
  digitalWrite(PINO_IN4, LOW);

  //rampa de aceleracao
  for (i = 0; i < 256; i=i+10){
    analogWrite(PINO_ENA, i);
    analogWrite(PINO_ENB, i);
    delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
  }

  //rampa de desaceleracao
  for (i = 255; i >= 0; i=i-10){
    analogWrite(PINO_ENA, i);
    analogWrite(PINO_ENB, i);
    delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
  }

  delay(TEMPO_ESPERA); //intervalo de um segundo

}
```

Código dos Componentes

Display I2C

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

//Inicializa o display no endereco 0x27
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
 lcd.init();
}

void loop()
{
  lcd.setBacklight(HIGH);
  lcd.setCursor(0,0);
  lcd.print("Arduino e Cia !!");
  lcd.setCursor(0,1);
  lcd.print("LCD e modulo I2C");
  delay(1000);
  lcd.setBacklight(LOW);
  delay(1000);
}
```

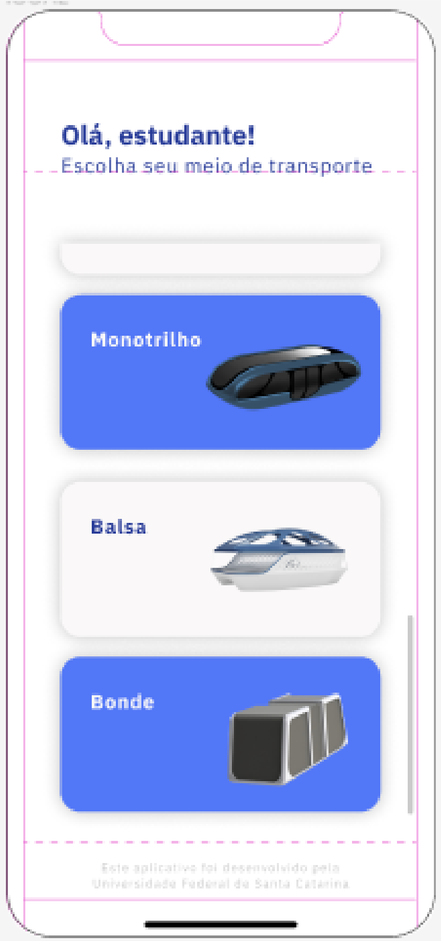
Código dos Componentes

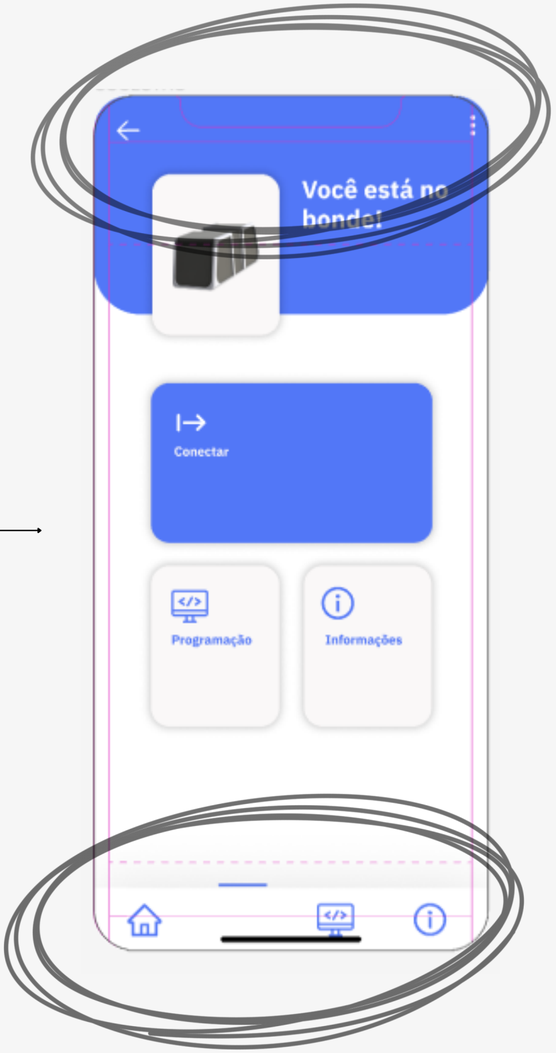
Mini DF Player

```
void setup() {
  pinMode(button, INPUT); //declaramos Button como INPUT.
  pinMode(button2, INPUT); //declaramos Button como INPUT.


  Serial.begin(9600);
  playerMP3Serial.begin(9600);

  Serial.println();
  Serial.println(F("Iniciando DFPlayer ... (Espere 3~5 segundos)"));

  if (!playerMP3.begin(playerMP3Serial)) {  // COMUNICAÇÃO REALIZADA VIA SOFTWARE SERIAL
    Serial.println(F("Falha:"));
    Serial.println(F("1.conexões!"));
    Serial.println(F("2.cheque o cartão SD!"));
    while(true){
      delay(0);
    }
  }

  Serial.println(F("DFPlayer iniciado!"));

  playerMP3.volume(volumeMP3);

  #ifdef DEBUG
    Serial.println("o Setup acabou");
  #endif

}

void loop() {

    if (digitalRead(button) == HIGH){
    playerMP3.playFolder(2, 1);
    Serial.println("Tocando pasta 02, musica 001");
    delay(1000);
    }
    if (digitalRead(button2) == HIGH){
    playerMP3.playFolder(2, 2);
    Serial.println("Tocando pasta 02, musica 001");
    delay(1000);
    }
}
```

Soluções

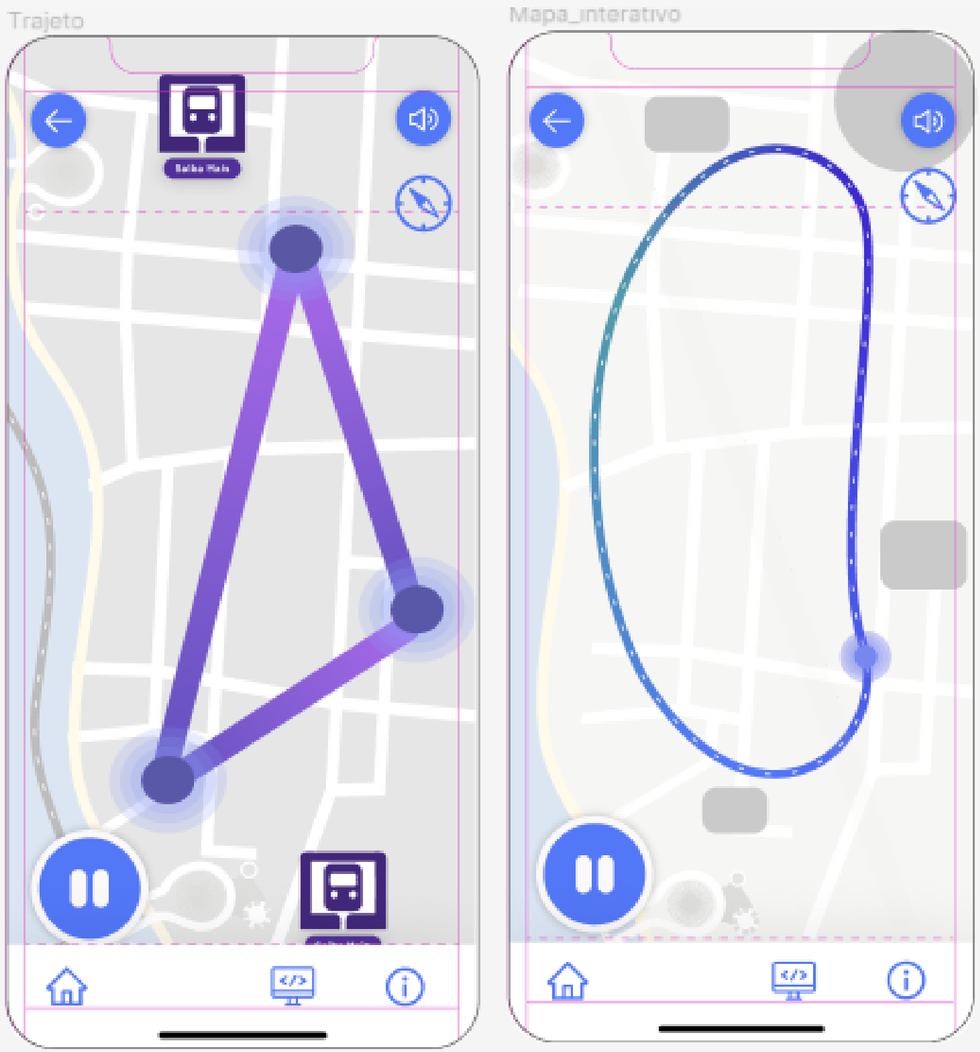
Otimização da navegação, Interação e usabilidade

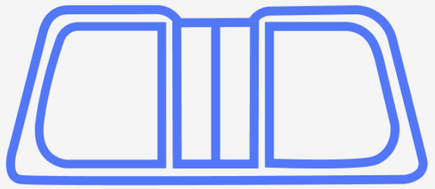




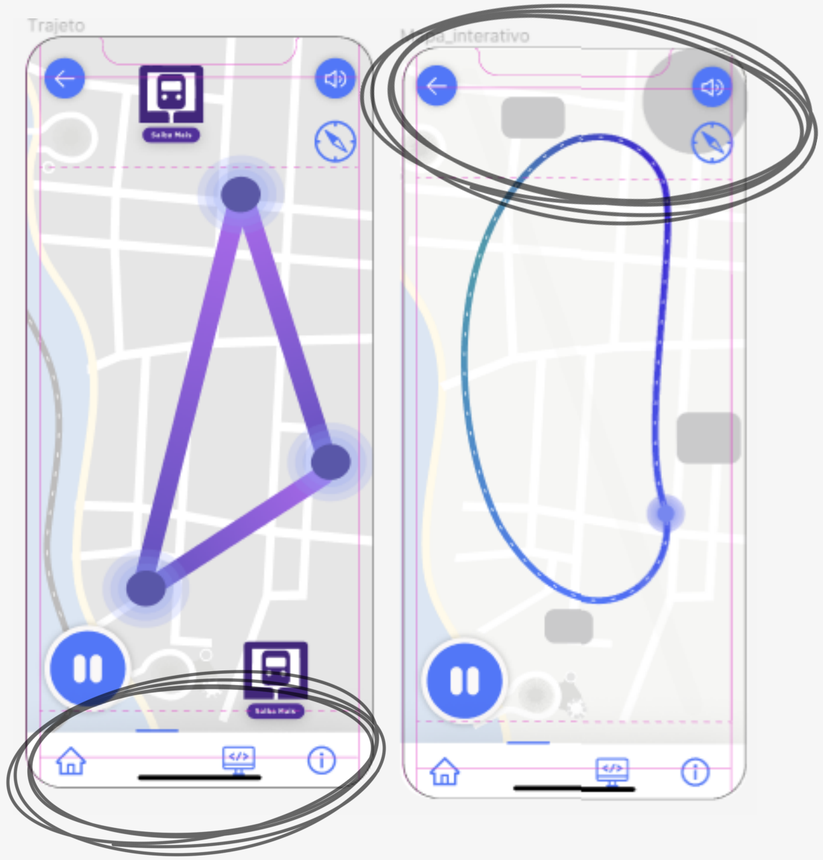
Soluções

Otimização da navegação, Interação e usabilidade





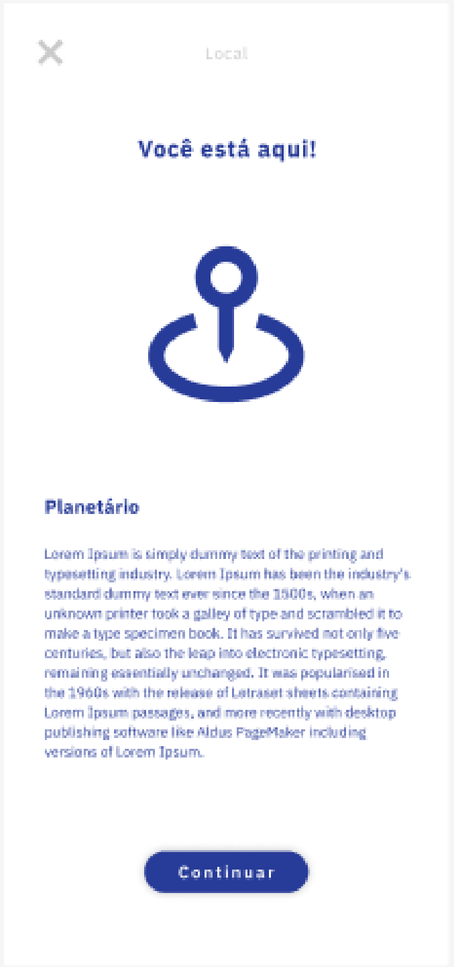
Proposta inicial de icone

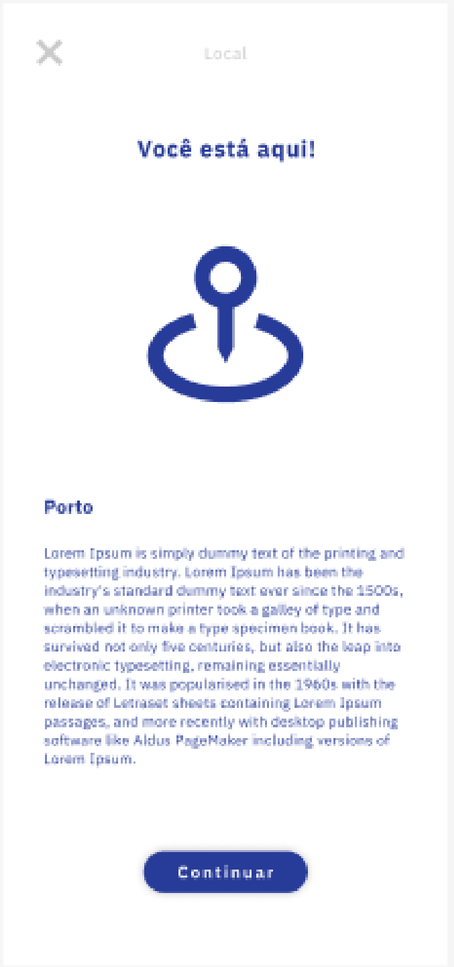


Soluções

Otimização da navegação, Interação e usabilidade

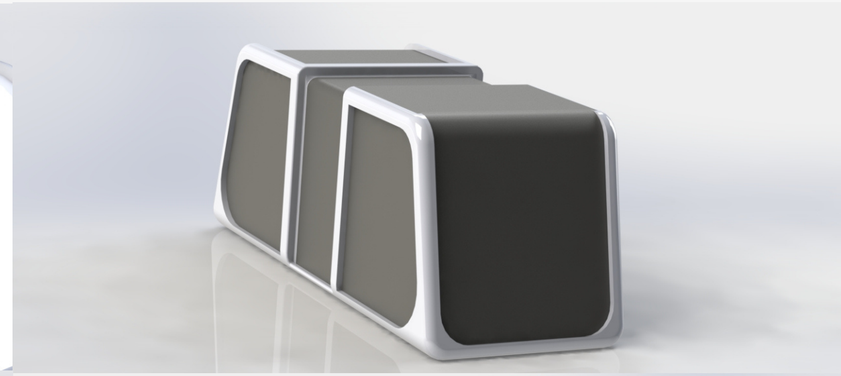
Informações sobre as paradas no tabuleiro, redirecionando para telas informativas dos projetos do tabuleiro

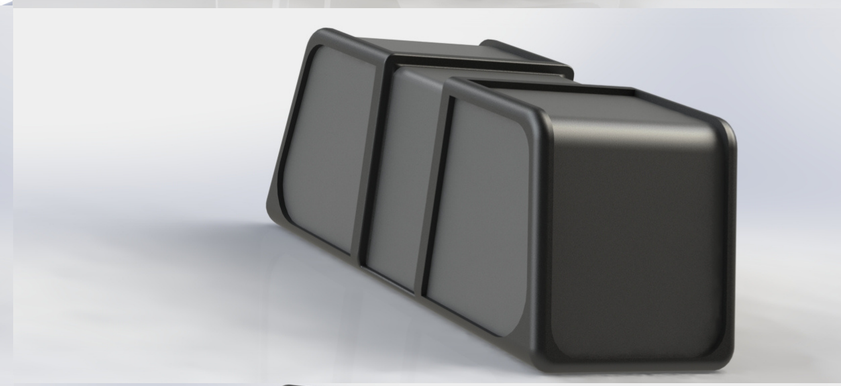


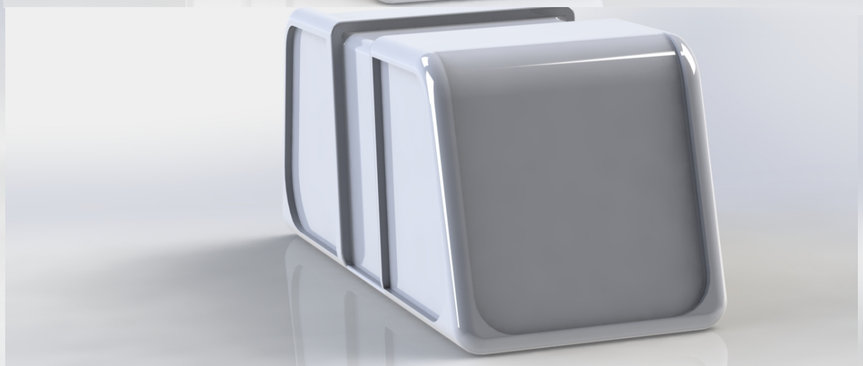


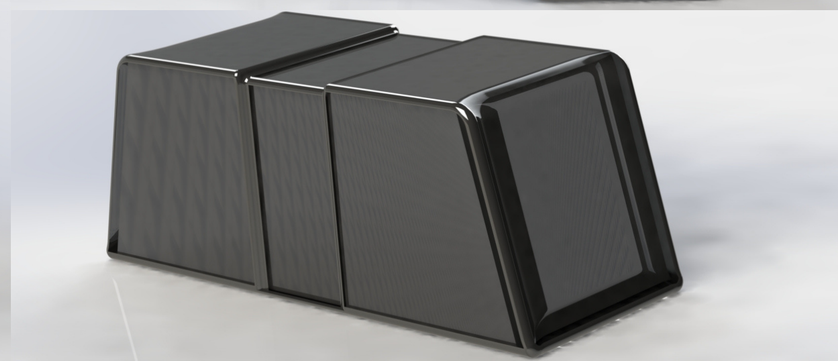
Soluções

Otimização da alternativa escolhida



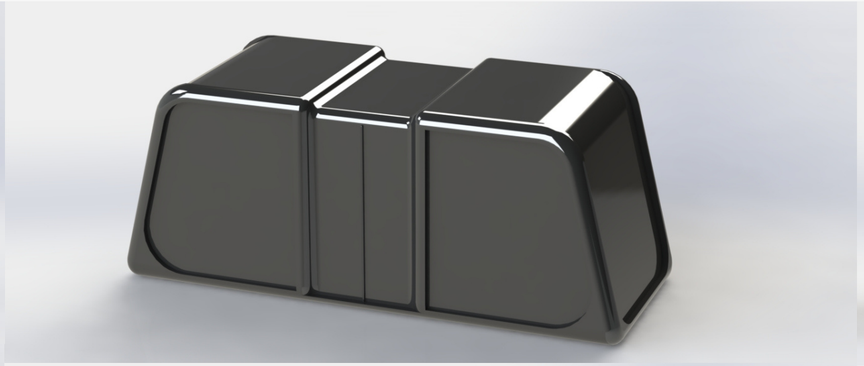
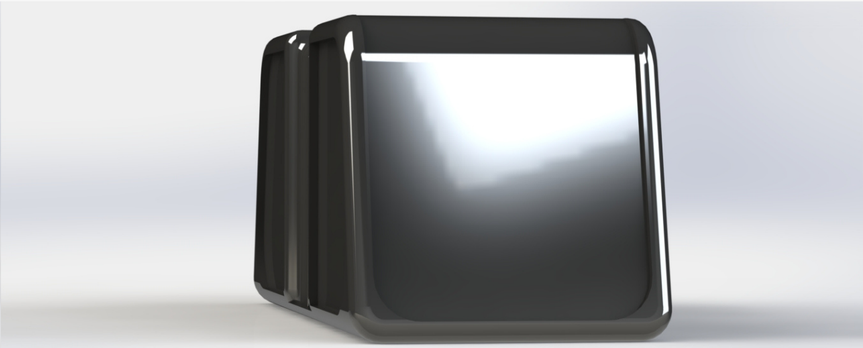
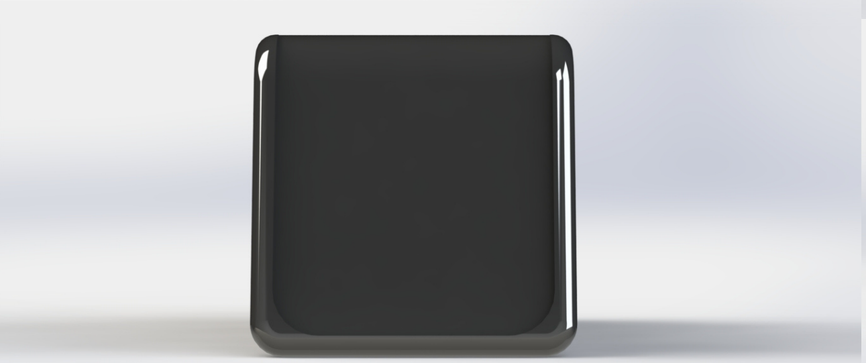
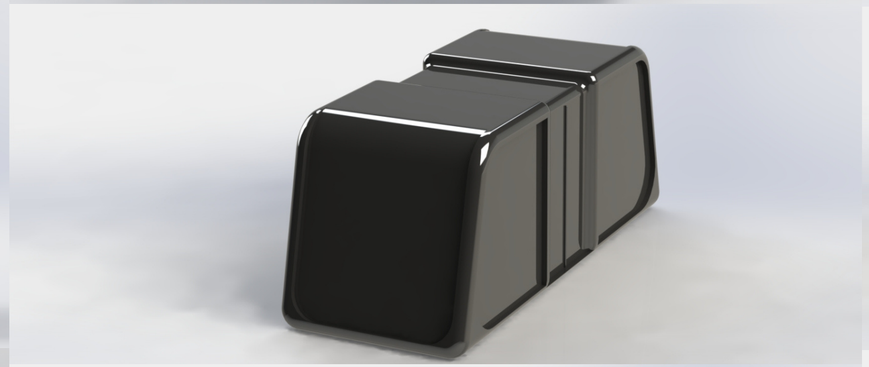
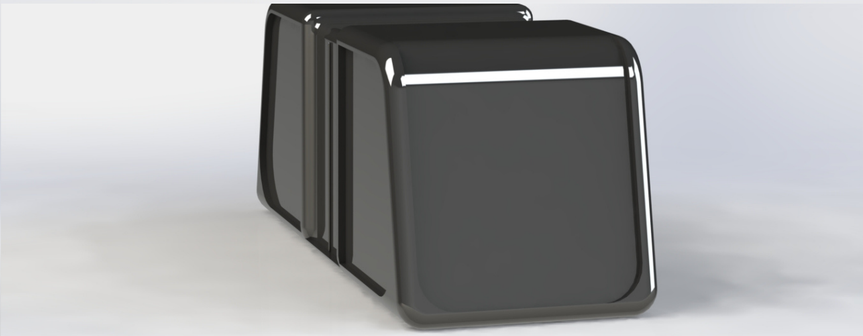


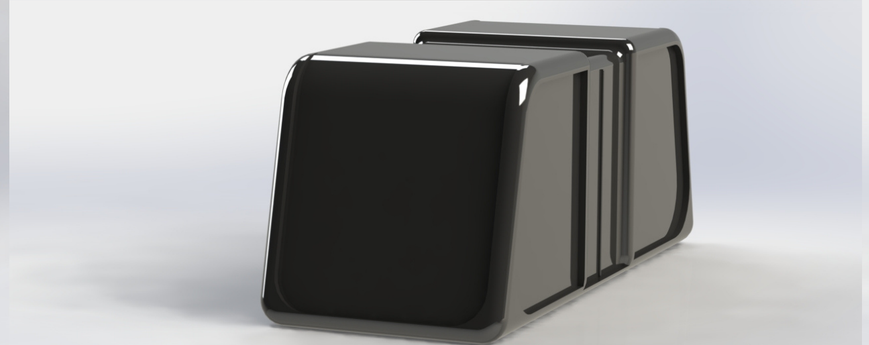
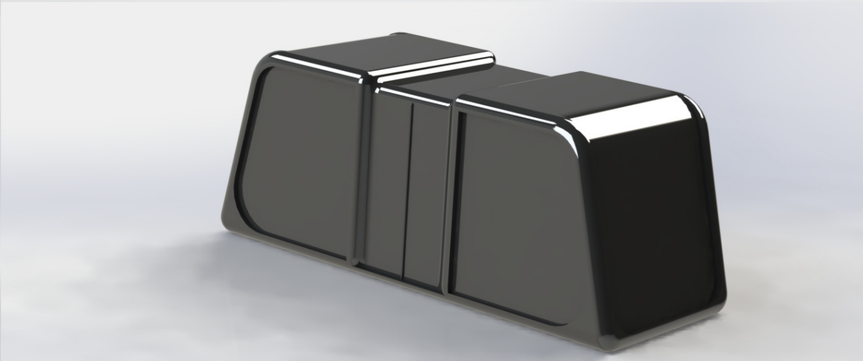




Prototipação

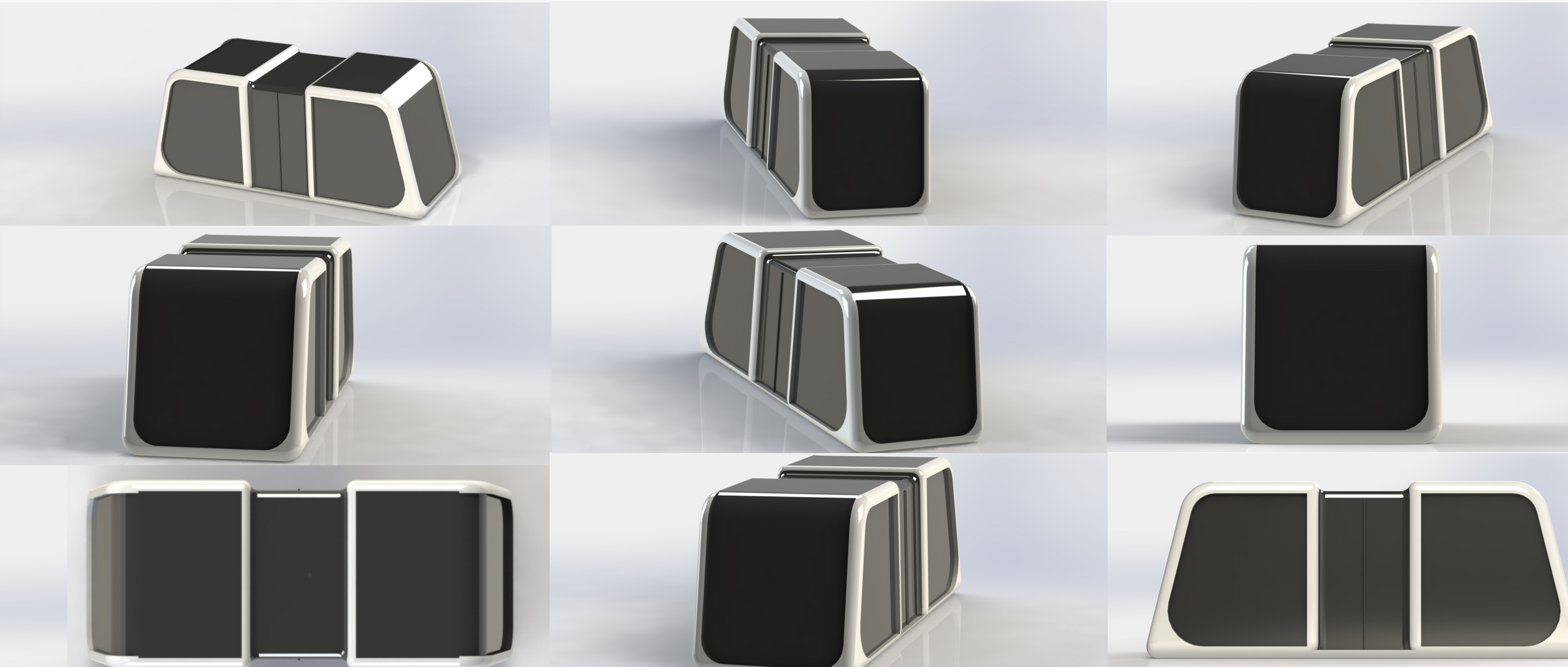
Modelagem e construção do modelo. Testes e ajustes

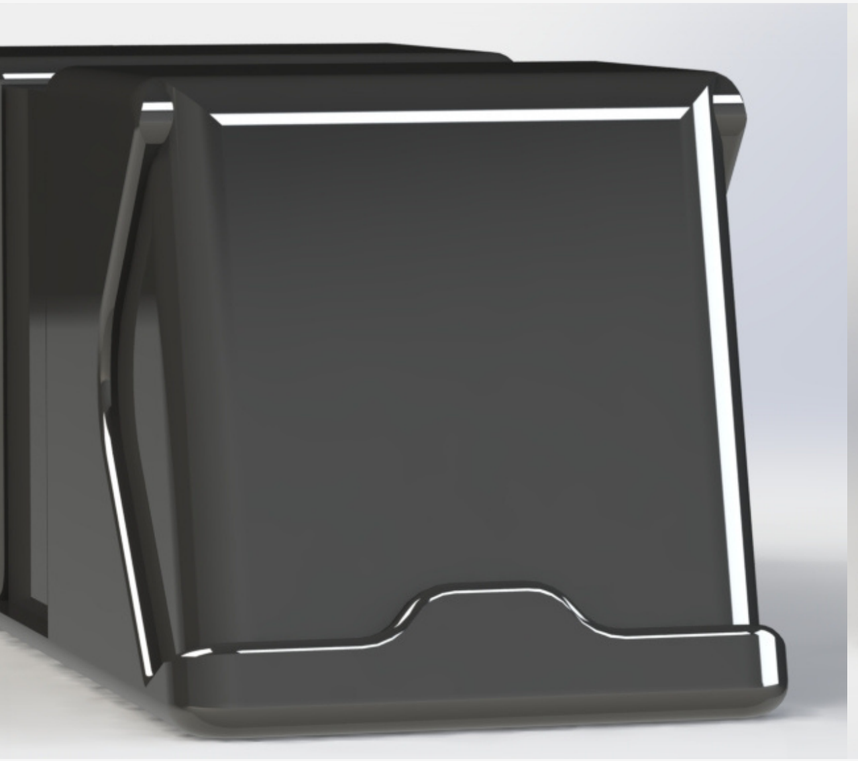
Prototipação

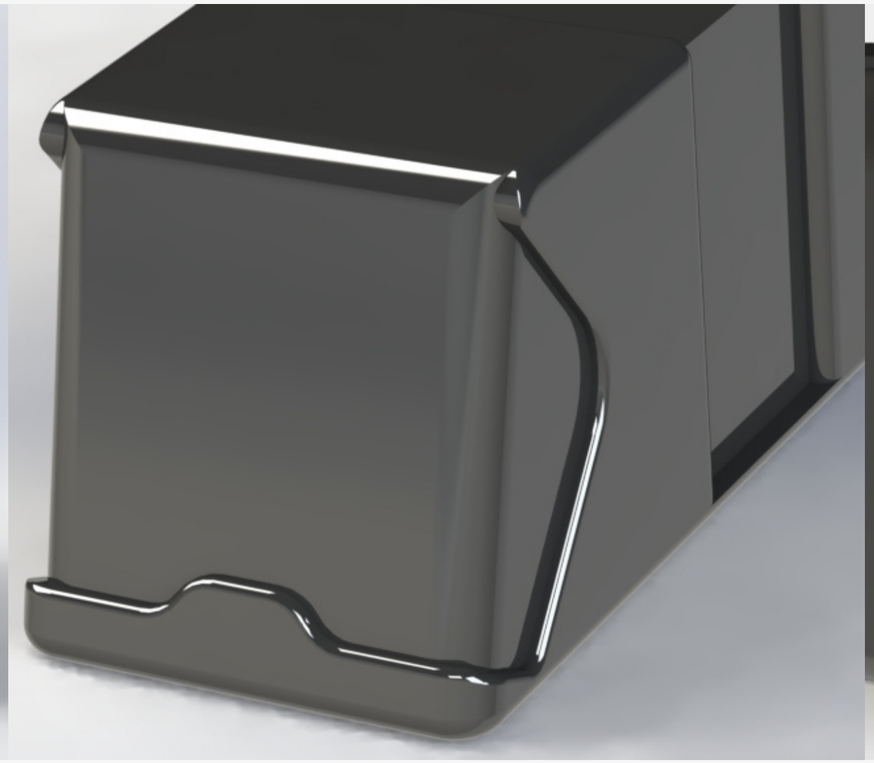
Modelagem e construção do modelo. Testes e ajustes

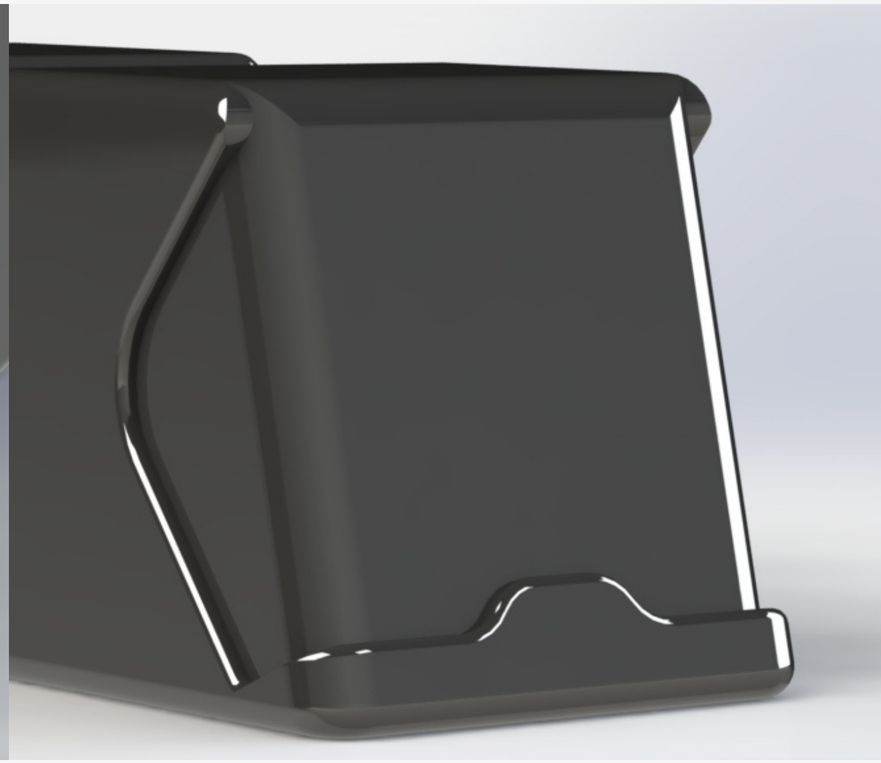


Prototipação

Modelagem e construção do modelo. Testes e ajustes



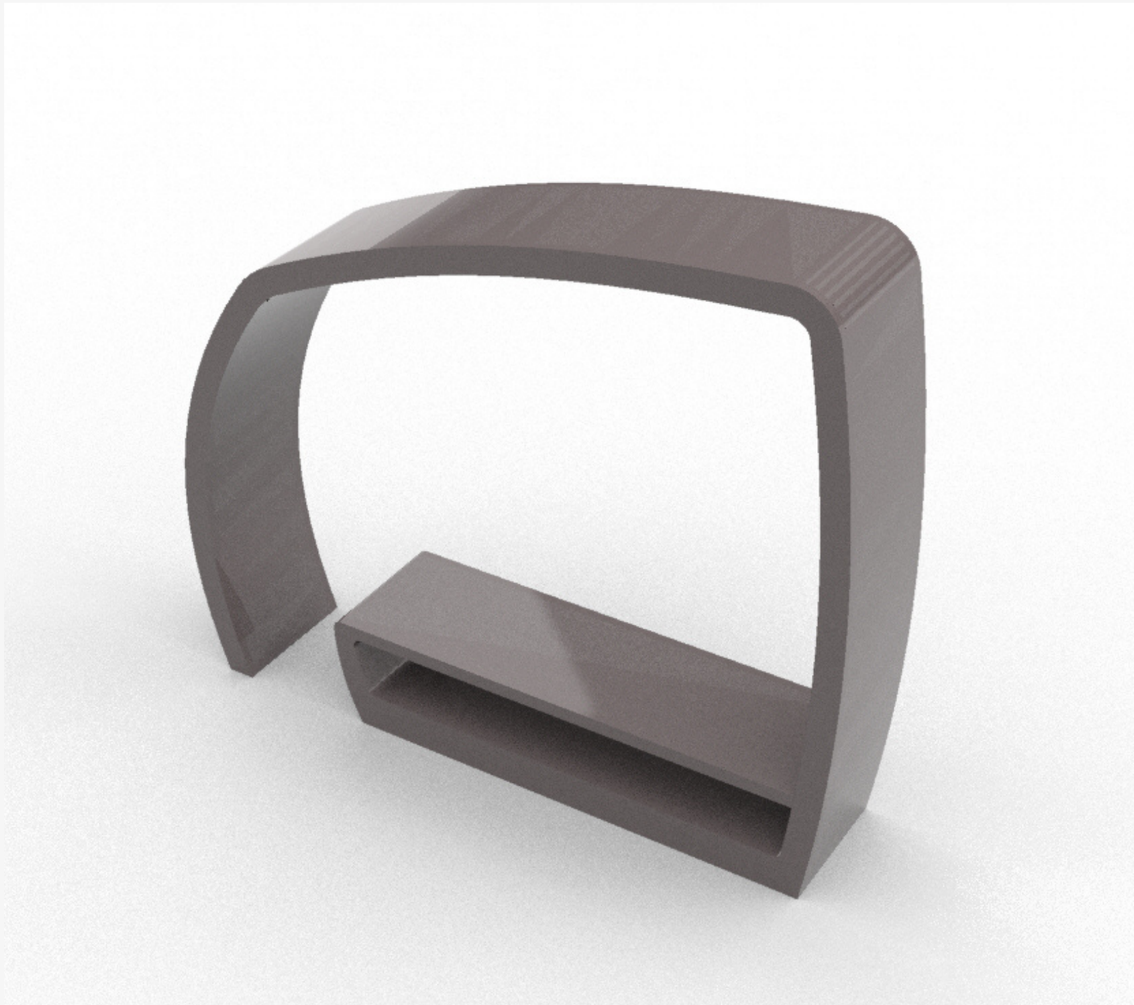


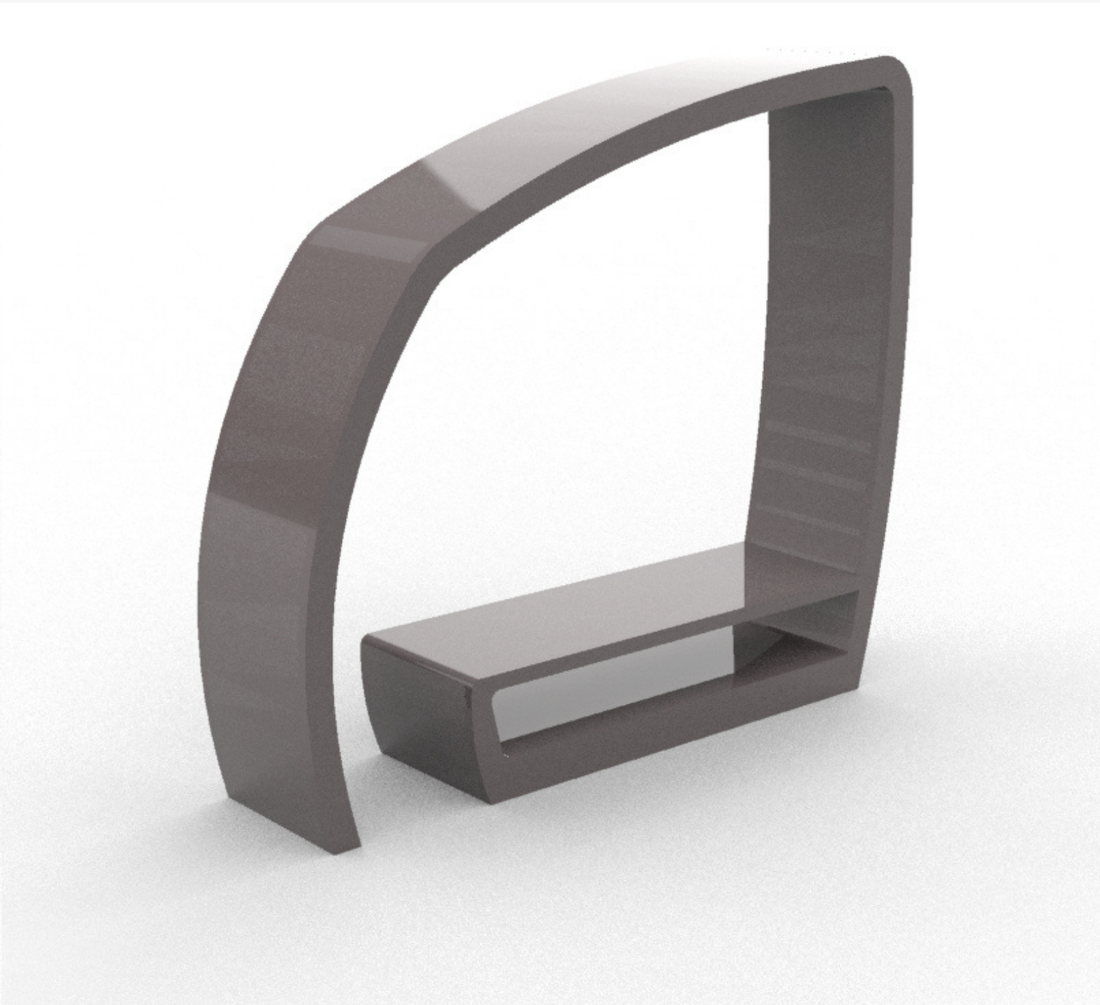


Prototipação

Parada do bonde

8cm x 7cm x 5cm





Obrigada!

Projeto de Produto 4