

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE MOGI MIRIM
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ADRIANO SOUZA GUIMARÃES
ALAN RODRIGO ALAMINO
ANDERSON ROBERTO THULER
CARLOS EDUARDO SECOLIN
LEONI TEIXEIRA DE ARAÚJO
FELIPE CRISPIM DE MATOS
MARCO OLIVEIRA

SISTEMA DE VOTAÇÃO POR TOTEM

MOGI MIRIM – SP
2019

LISTA DE FIGURAS

Figura 1 - Dispositivo SBC.....	7
Figura 2 - Configurando módulo.	8
Figura 3 - Botões Push Button.....	8
Figura 4 - Botões conectados no dispositivo.	9
Figura 5 - Janela do dispositivo.	11
Figura 6 - Adicionar smartphone.	12
Figura 7 - Localizando Raspberry.....	13
Figura 8 – Aba Programming.....	14
Figura 9 - Mapa de São Paulo - grafites por região.	15
Figura 10 - Galeria de imagens.	16
Figura 11 - Caixa de comentários e cutidas.	16
Figura 12 - Banco de dados.	17
Figura 13 - Protótipo de tela.	17

RESUMO

O Totem é um protótipo desenvolvido com equipamentos de hardware e software habitualmente usados no dia-a-dia das pessoas, e que tem a função de oferecer auxílio na localização em shopping centers, verificação de conteúdos relacionados, controle de avisos internos, interação dos usuários com os horários dos cinemas e visualização de trailers dos filmes em cartaz. Através deste trabalho a ideia é apresentar as artes urbanas em uma espécie de totem para coleta de dados. As funcionalidades de comunicação entre os dispositivos foram previamente simuladas através do software Cisco Packet Tracer utilizando a transmissão de dados via Bluetooth. O protótipo do Totem foi desenvolvido na plataforma .Net usando a tecnologia WPF (Windows Presentation Foundation) e a linguagem de programação C#. Acoplado a uma tela Touchscreen, possibilita a interatividade do usuário com o protótipo de uma forma simples, rápida e clara. Neste trabalho desenvolve-se um totem com a motivação de difundir a tecnologia no cotidiano das pessoas e incluir um meio de mineração de dados sobre o tema abordado.

ABSTRACT

Totem is a prototype developed with hardware and software commonly used in people's daily life and has the function of offering assistance in locating shopping malls, verifying related contents, controlling internal warnings, interacting with users to schedule tickets for cinemas and visualization of trailers of the films in poster. This work has the idea of present urban arts in a kind of totem for data collection. The communication feature between devices are simulated by Cisco Packet Tracer using data transmission via Bluetooth. The prototype on the .Net platform using the WPF (Windows Presentation Foundation) technology and the C # programming language. Coupled with a Touchscreen, it enables user interactivity with the prototype in a simple, fast and clear way. In this work a totem is developed with the motivation to spread the technology in the daily life of the people and to include a means of mining of data on the subject addressed.

SUMÁRIO

1. INTRODUÇÃO	6
2. MÉTODO DE DESENVOLVIMENTO	7
2.1. Adicionando e configurando raspberry	7
2.2. Adicionando e configurando <i>Smartphone</i>	11
3. PROTÓTIPOS DE TELA.....	16
4. CONSIDERAÇÕES FINAIS	18

1. INTRODUÇÃO

Este trabalho tem o propósito de fazer uma ligação entre o projeto inicialmente desenvolvido na disciplina de laboratório de redes que tem por finalidade o uso dos dados de opiniões públicas relacionadas a artes urbanas na cidade de São Paulo, como também, a exibição dos mesmos dentro do mapa da cidade utilizando a ferramenta Power BI da Microsoft. Como forma de melhorar a captura dos dados e obter um resultado mais próximo da realidade, serão utilizadas tecnologias do tipo Totem que poderão ser instalados em frente de cada arte na cidade. Os dispositivos terão carregados em seu sistema uma imagem da obra na qual se encontra o totem e dois botões para votação, “Curtiu” e “Não Curtiu”. Os dados ficarão registrados no Raspberry ficarão disponíveis para captura através de um dispositivo móvel através da tecnologia Bluetooth.

Para desenvolvimento do protótipo de telas, foi utilizado o Visual Studio com funcionalidades que o sistema poderá ter como intuito de facilitar a captura dos dados. No próximo Capítulo, serão apresentados detalhes de como foi feito o esquema de registro de dados com ajuda do sistema Cisco Packet Tracer.

2. MÉTODO DE DESENVOLVIMENTO

Com auxílio do sistema Cisco Packet Tracer, foi possível simular como um sistema básico para captura de dados pode ser desenvolvido.

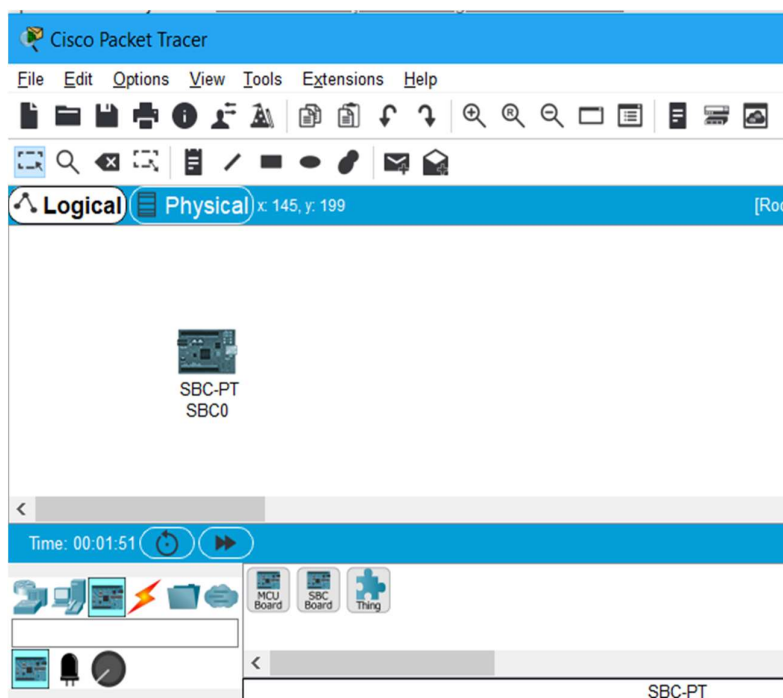
Com apenas a inclusão de um Raspberry, dois botões, um totem e um smartphone, o Packet Tracer possibilitou o desenvolvimento deste sistema e como será feita sua ligação mesmo sem a aquisição dos dispositivos fisicamente, tudo isto apenas configurando os itens na tela do computador.

Abaixo serão exibidos os passos que foram feitos, quais itens adicionados e como foram feitas as configurações deste dispositivo.

2.1. Adicionando e configurando raspberry

Para começar, adicionar o modelo disponibilizado pela Cisco SBC, para isso selecionar a opção “Components”, depois “Boards” e por fim adicionar o dispositivo “SBC” na Figura 1:

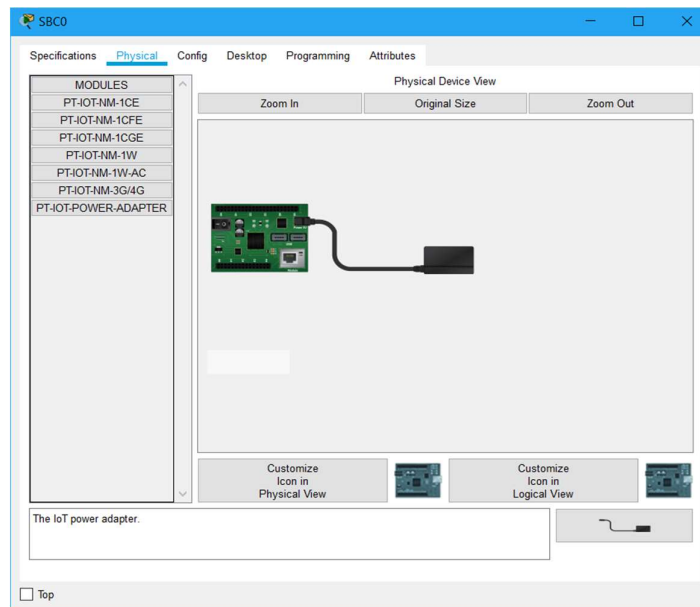
Figura 1 - Dispositivo SBC.



Fonte: Autores.

Clicar sobre o mesmo e certificar que o módulo está configurado como a Figura 2 abaixo:

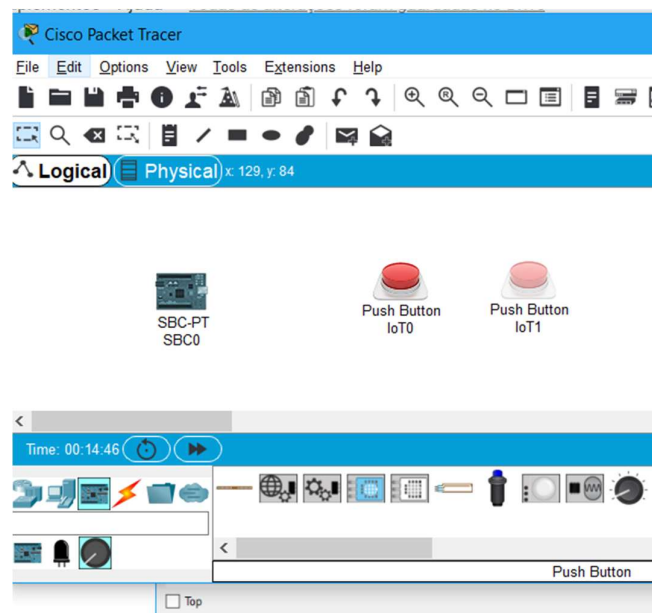
Figura 2 - Configurando módulo.



Fonte: Autores.

Após incluir o Raspberry, é necessário adicionar os botões “Push Button”, Figura 3, que serão responsáveis por realizar a votação, serão a forma de input dos dados de opinião pública.

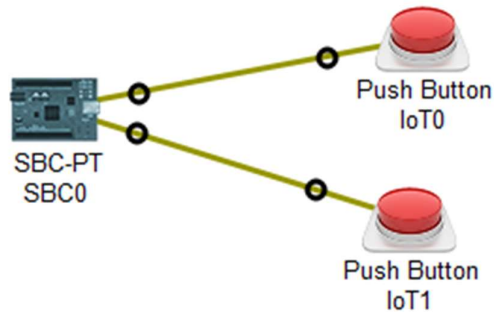
Figura 3 - Botões *Push Button*.



Fonte: Autores.

Nesta etapa, os botões devem ser ligados ao Raspberry com o cabo “IoT Custom Cable”. A Figura 4 abaixo, exibe os botões conectados no dispositivo.

Figura 4 - Botões conectados no dispositivo.



Fonte: Autores.

Com os botões já conectados, deve-se então configurar o Raspberry para realizar as contagens para cada botão pressionado e computar a votação.

Para configurar, clicar sobre o Raspberry, acessar a aba *Programming*, clicar em New JavaScript e inserir o seguinte código:

```
var dstService = "{00000000-0000-0000-0000-000000000001}";  
var dstMac = "FFFF.FFFF.FFFF";
```

```
//Pin na uqal os botões estão conectados
```

```
var botLike = 2;
```

```
var botunlike = 0;
```

```
//Variáveis que registrarão os votos
```

```
var like = 0;
```

```
var unlike = 0;
```

```
// Variáveis para o serviço
```

```
var service;
```

```
function setup(){
```

```
pinMode(botLike, INPUT);
```

```
pinMode(botunlike,INPUT);
```

```
//Inicia o serviço Bluetooth
```

```

Bluetooth.init();
Bluetooth.onDevicePair = function(mac) {
  Serial.println(mac + ' pareado. ');
};

Bluetooth.onDeviceUnpair = function(mac) {
  Serial.println(mac + ' despareado. ');
};

Bluetooth.onDeviceConnect = function(mac) {
  Serial.println(mac + ' conectado. ');
  service.send(dstMac, dstService, "Curtiram: " + like + " - Não Curtiram: " + unlike);
};

Bluetooth.onDeviceDisconnect = function(mac) {
  Serial.println(mac + ' desconectado. ');
};

// Criar Serviço
service = new BluetoothService();

// Iniciar Serviço
service.start(dstService);
Serial.println("Servidor online...");
}

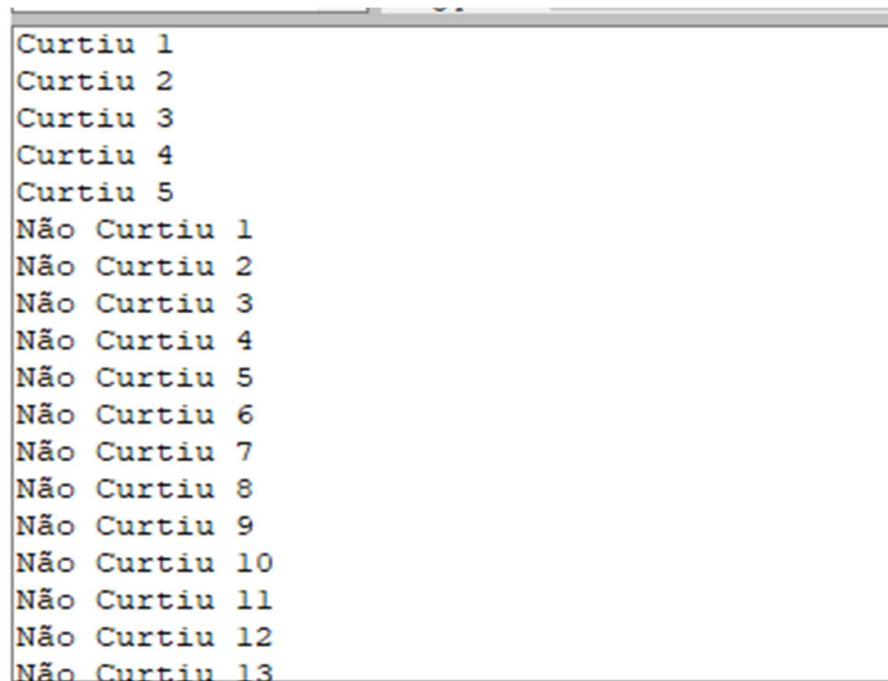
//Função para registrar os votos
function loop () {
  switch (digitalRead(botLike)) {
    case HIGH:
      like += 1;
      console.log('Curtiu ' + like)
      break;
    }
  switch (digitalRead(botunlike)) {
    case HIGH:
      unlike += 1;
      console.log('Não Curtiu ' + unlike)
      break;

```

```
}  
}
```

Após a inserção do código e testar se o mesmo está funcionando, no dispositivo Raspberry, dentro da aba “*Programming*” clicar em Run. A partir deste momento, cada pressionada no botão “Curtir” ou “Não Curtir” estará sendo registrado e é exibido na janela do dispositivo, conforme a Figura 5:

Figura 5 - Janela do dispositivo.



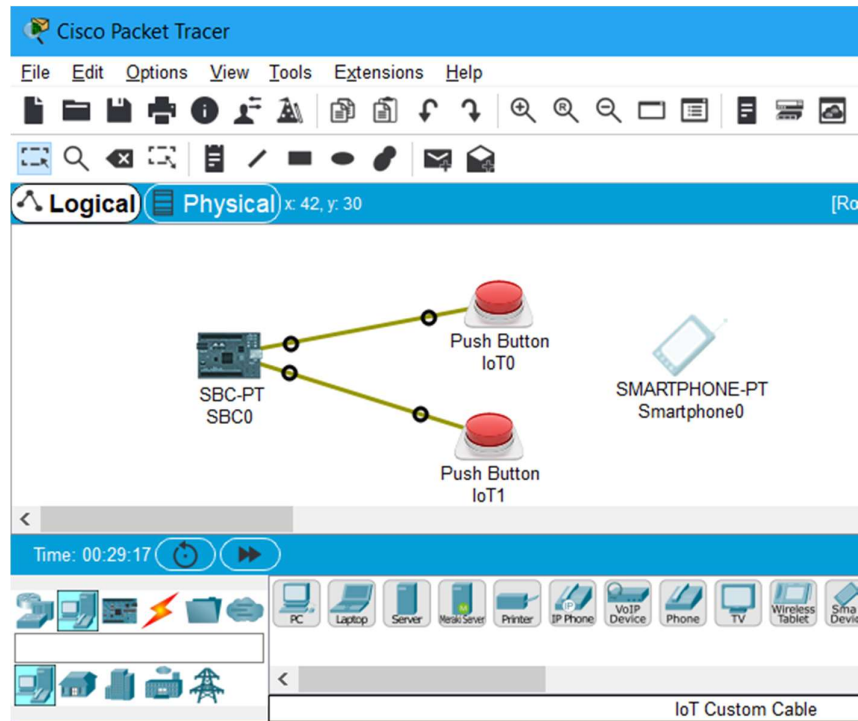
```
Curtiu 1  
Curtiu 2  
Curtiu 3  
Curtiu 4  
Curtiu 5  
Não Curtiu 1  
Não Curtiu 2  
Não Curtiu 3  
Não Curtiu 4  
Não Curtiu 5  
Não Curtiu 6  
Não Curtiu 7  
Não Curtiu 8  
Não Curtiu 9  
Não Curtiu 10  
Não Curtiu 11  
Não Curtiu 12  
Não Curtiu 13
```

Fonte: Autores.

2.2. Adicionando e configurando *Smartphone*

Feito os passos acima, é hora de adicionar o *Smartphone*, nele deverá ser configurado o que deve ser feito quando conectado no Raspberry, para isso adicionar o dispositivo *Smartphone*, conforme exibe a Figura 6:

Figura 6 - Adicionar smartphone.



Fonte: Autores.

Após adicionado, clicar sobre o dispositivo, acessar a aba “*Programming*” e adicionar o seguinte código em um novo arquivo JavaScript:

```
var dstService = "{00000000-0000-0000-0000-000000000001}";
var dstMac = "FFFF.FFFF.FFFF";
var service;
function setup() {
    Bluetooth.init();
    Bluetooth.onDevicePair = function(mac)
        Serial.println(mac + ' pareado.');
```



```
};
Bluetooth.onDeviceUnpair = function(mac) {
    Serial.println(mac + ' despareado.')
```



```
};
Bluetooth.onDeviceConnect = function(mac) {
    Serial.println(mac + ' conectado.');
```



```
};
```

```

Bluetooth.onDeviceDisconnect = function(mac) {
    Serial.println(mac + ' desconectado.');
```

```

};

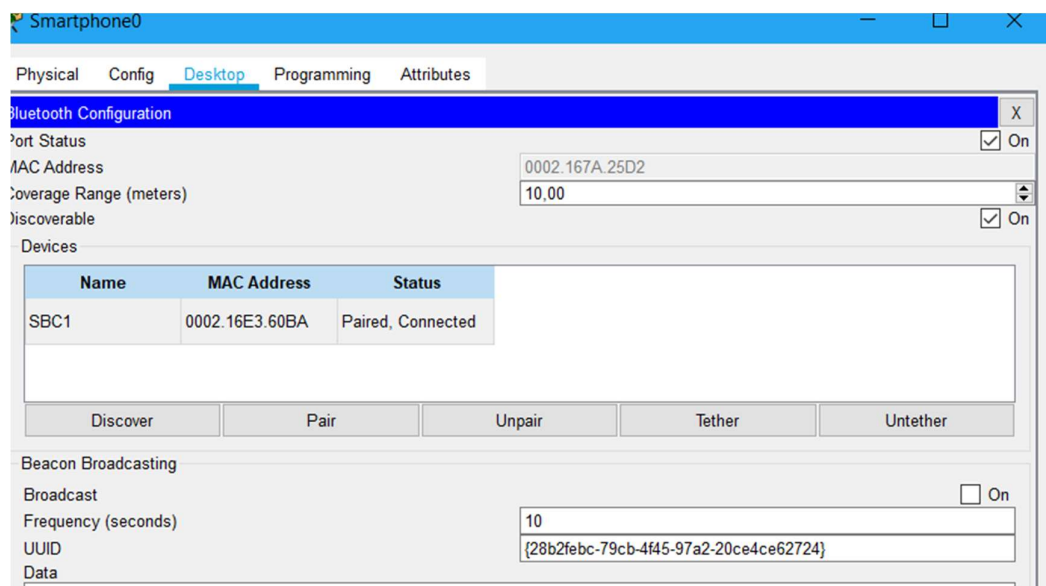
// Criar Serviço
    service = new BluetoothService();
    // Quando recebe dados
    service.onReceive = function(srcMac, srcService, dstMac, dstService, data) {
        Serial.println(data);
    };

// Iniciar Serviço
    service.start(dstService);
}

```

Após a configuração do código, clicar em *Run*, dentro da aba *programming* e logo em seguida, acessar a aba *Desktop*, procurar por *Bluetooth*, clicar em *Discover* para localizar o Raspberry e clicar em *parear* após encontrado, conforme a Figura 7:

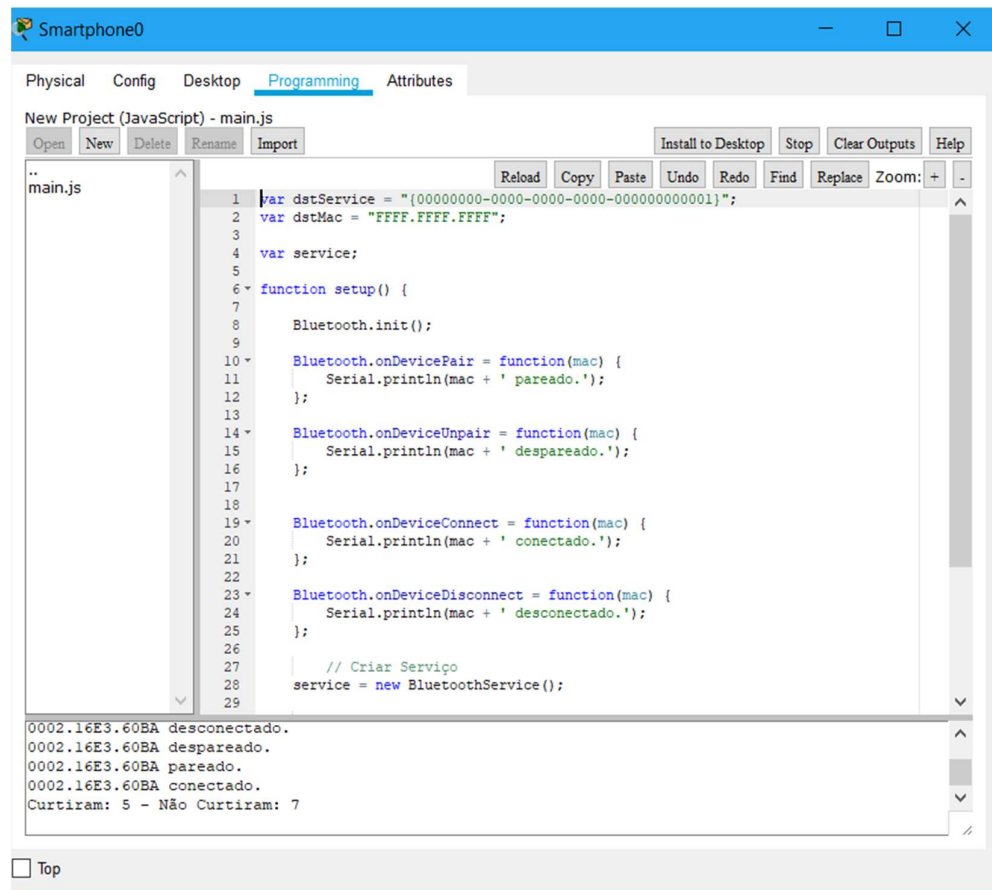
Figura 7 - Localizando Raspberry.



Fonte: Autores.

Após o sistema pairar, acessar a aba *Programming* novamente e teremos a quantidade de curtidas que aquele Totem tem registrado no console do dispositivo, conforme a Figura 8.

Figura 8 – Aba *Programming*.



The screenshot shows the 'Smartphone0' application window with the 'Programming' tab selected. The code editor displays a JavaScript file named 'main.js' with the following content:

```
1 var dstService = "{00000000-0000-0000-0000-000000000001}";
2 var dstMac = "FFFF.FFFF.FFFF";
3
4 var service;
5
6 function setup() {
7     Bluetooth.init();
8
9     Bluetooth.onDevicePair = function(mac) {
10         Serial.println(mac + ' pareado.');
```

The console output at the bottom shows the following messages:

```
0002.16E3.60BA desconectado.
0002.16E3.60BA despereado.
0002.16E3.60BA pareado.
0002.16E3.60BA conectado.
Curtiram: 5 - Não Curtiram: 7
```

Fonte: Autores.

Com os dados finais, é possível montar a relação das contagens da opinião por bairro de cada arte e associar ao projeto do *Microsoft Power BI* para ser exibido no mapa, conforme a Figura 9:

Figura 9 - Mapa de São Paulo - grafites por região.



Fonte: Autores.

3. PROTÓTIPOS DE TELA

Protótipo do totem com galeria de imagens de uma região específica da cidade de São Paulo, conforme a Figura 10:

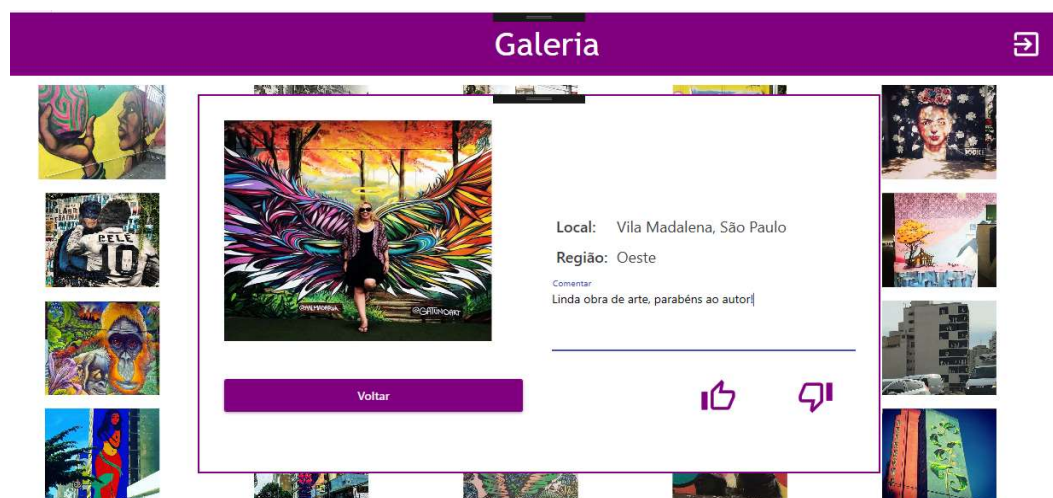
Figura 10 - Galeria de imagens.



Fonte: Autores.

Ao selecionar uma arte, abre-se uma caixa para inserção de comentários e curtidas, conforme a Figura 11:

Figura 11 - Caixa de comentários e curtidas.



Fonte: Autores.

Depois de todo o processo os dados são armazenados, conforme a Figura 12:

Figura 12 - Banco de dados.

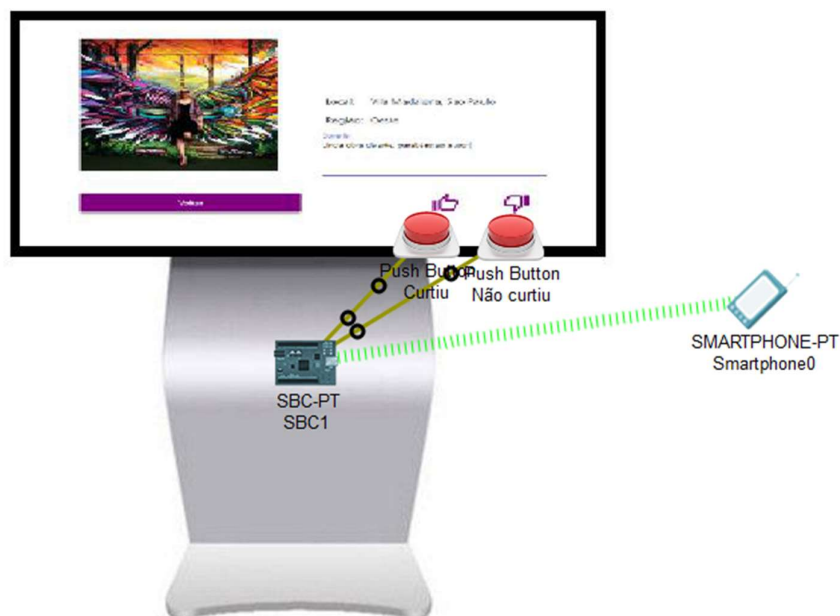
The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The central pane displays a SQL query: `SELECT Comentarios.comentario, Imagens.bairro, Imagens.cidade, Imagens.regiao, Imagens.like_positivo, Imagens.like_negativo FROM Imagens LEFT OUTER JOIN Comentarios ON Comentarios.id_imagem = Imagens.id WHERE Comentarios.comentario <> ''`. The bottom pane shows the results of the query in a table format.

	comentario	bairro	cidade	regiao	like_positivo	like_negativo
1	Linda Arte, Ótimo trabalho!!!	Vila Madalena	São Paulo	Oeste	3	2
2	Imagem estranha, não gostei!	Vila Madalena	São Paulo	Oeste	3	2
3	Não curti.	Vila Madalena	São Paulo	Oeste	3	2
4	Incrível!!! Parabéns	Vila Madalena	São Paulo	Oeste	1	NULL
5	Bonita imagem, parece foto de verdade!	Campo Limpo	São Paulo	Sul	1	NULL
6	Linda obra de arte, parabéns ao autor!!!	Vila Madalena	São Paulo	Oeste	1	NULL

Fonte: Autores.

Modelo de tela incluído no sistema Cisco Packet Tracer, conforme a Figura 13:

Figura 13 - Protótipo de tela.



Fonte: Autores.

4. CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo possibilitou uma análise de como dados podem ser levantados de forma simples através de uma metodologia acessível e viável para sua coleta comparado com outros meios. Nosso estudo utiliza o protótipo de um totem com dois botões "Curtir" e "Não Curtir", uma tela interativa, um *Raspberry* para armazenar dados provenientes da votação e um smartphone para recebê-los através do protocolo Bluetooth, a proposta inicial é captar a aceitação das pessoas por essas obras através destes botões essa solução se mostra muito eficiente pois para a pessoa fazer essa votação ela precisa estar presente no local da obra e tê-la visto, tendo em mãos esses dados podemos popular tabelas que posteriormente podem ser incorporadas a gráficos na ferramenta Power BI da Microsoft, esse estudo pode ser utilizado para fins de sociais, de como cada região da cidade reage aquele tipo de arte. Isso resolveria o problema proposto inicialmente que era de coletar esses dados de uma outra forma que não fosse através de pesquisas feitas obra a obra em sites, blogs e redes sociais.