

## Relatório Técnico

**Nº Grupo:** 9

**Nome dos integrantes:** Arthur Rodrigues, Gabriel Pereira, Gleison Almeida, Lucas Gabriel, Pedro Henrique, Samara Farias

**Turma:** 1ADSA

---

**Tema do projeto:** Monitoramento de Lixeiras Subterrâneas

**Sensor:** HC-SR04 (Ultrassônico)

---

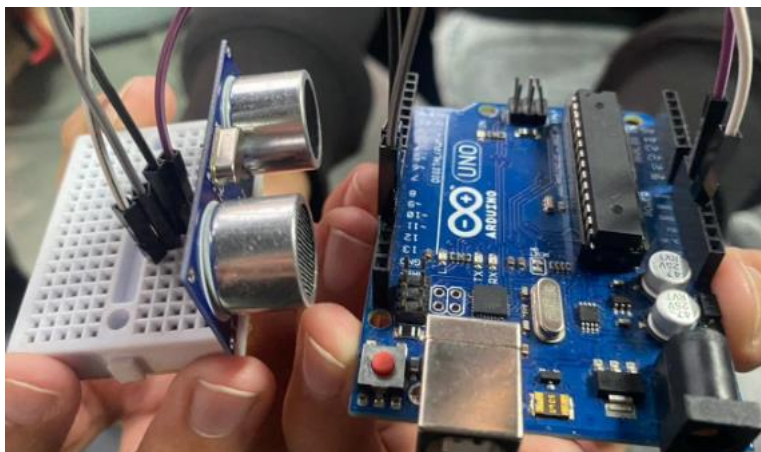
### Introdução

As frotas de caminhões adaptadas para coleta de contêineres em lixeiras subterrâneas gastam anualmente cerca de milhões de reais com rotas longas, complexas e ineficazes, todo o processo de extração do lixo envolve combustível das máquinas, mão de obra especializada e um processo demorado que se não for feito de forma inteligente pode gerar coletas desnecessárias em lixeiras pouco utilizadas.

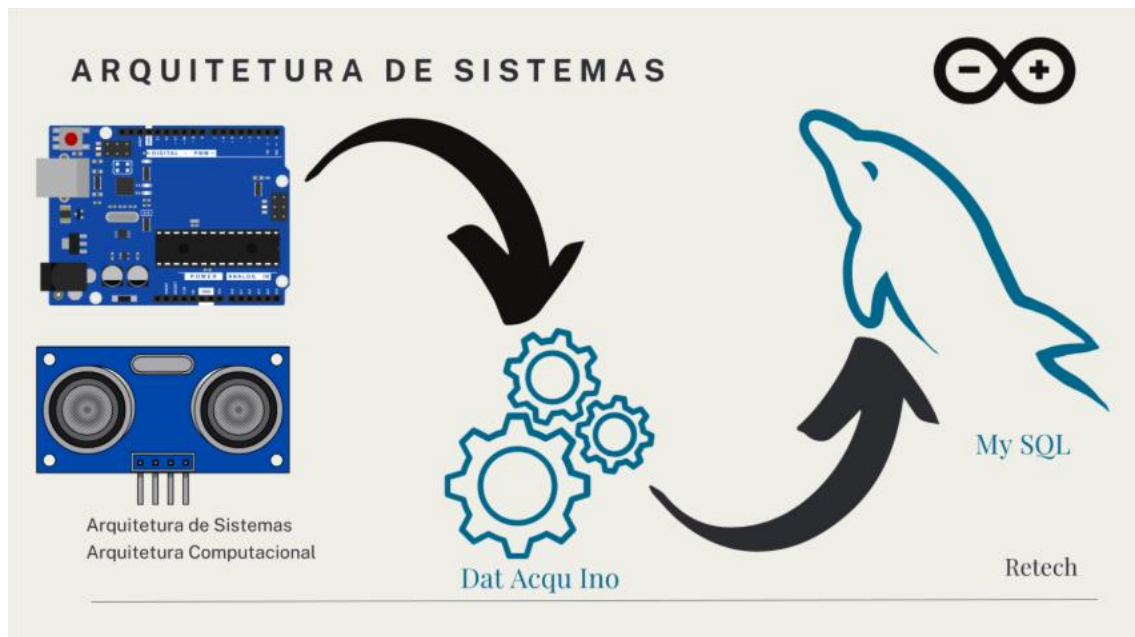
A ReTech visa mitigar os custos operacionais envolvidos nas coletas de lixeiras subterrâneas. Através do monitoramento contínuo, desenvolvemos dashboards personalizadas para o seu negócio em nosso site institucional onde disponibilizamos funções de perfil individual e uma calculadora de prejuízo para seu planejamento.

### Arquitetura de Montagem do Sensor

Nossa organização está utilizando o Arduino uno R3 somado ao sensor HC-SR04 para realizar a coleta dos dados de distância da tampa até o fundo da lixeira o sensor é conectado em um mini protoboard por jumpers do tipo macho-macho a um Arduino responsável por executar o código de coleta dos Arduino.



## Arquitetura do Sistema



O sensor arduino HC-SR04 capta os dados de distância que são coletados pela API Dat Acqui No e os envia posteriormente ao banco de dados MySQL através de inserts.

## Código do Projeto

O código node utilizado para transferência de dados entre sensor e banco de dados é dividido em partições editáveis para utilização personalizável do devido sensor.

A repartição abaixo habilita ou desabilita o código node de realizar inserts no banco de dados, basta alterar o valor da variável de falso para verdadeiro.

```
// habilita ou desabilita a inserção de dados no banco de dados  
const HABILITAR_OPERACAO_INSERTIR = true;
```

O bloco de código abaixo configura as informações do banco de dados a ser conectado com a API. O host é definido como local por ser executado na própria máquina; user que será utilizado para criação do database; password com a senha do usuário, database utilizado para criar as tabelas e a porta serial usada para conectar o sensor.

```
// conexão com o banco de dados MySQL  
let poolBancoDados = mysql.createPool(  
  {  
    host: 'localhost',  
    user: 'ReTech_Arduino',  
    password: 'techtechlixo',  
    database: 'Retech',  
    port: 3306  
  }  
).promise();
```

A partição abaixo processa os dados recebidos pelo sensor e os classifica com o tipo de dados coletado por ele.

```
// processa os dados recebidos do Arduino
arduino.pipe(new serialport.ReadlineParser({ delimiter: '\r\n' })).on('data', async (data) => {
  console.log(data);
  const valores = data.split(';');
  // const sensorDigital = parseInt(valores[0]);
  const sensorDistancia = parseFloat(valores[0]);
```

Por fim, este bloco executa os inserts na tabela do banco de dados configurado posteriormente.

```
// insere os dados no banco de dados (se habilitado)
if (HABILITAR_OPERACAO_INSERIR) {

  // este insert irá inserir os dados na tabela "DadosArduino"
  await poolBancoDados.execute(
    'INSERT INTO DadosArduino (profundidade) VALUES (?)',
    [sensorDistancia]
  );
  console.log("valores inseridos no banco: ", sensorDistancia);
}
```

Dentro do Arduino IDE validamos e executamos o código C++ que da funcionalidade ao sensor que estamos utilizando, o bloco de código a seguir retira os valores inválidos no node e finaliza a operação com “;”.

```
#include <Ultrasonic.h>

const int PINO_TRIGGER = 12;
const int PINO_ECHO = 13;

HC_SR04 sensor(PINO_TRIGGER, PINO_ECHO);

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print(sensor.distance());
  Serial.println(";");

  delay(1000);
}
```

## Resultados Iniciais

Através da utilização da API Dat Acqui Ino do Node JS conseguimos extrair os dados do Arduino e assim gerar diversas saídas resultantes da captação realizada.

- Arduino IDE

Assim que verificamos se o código em C++ para nosso sensor foi feito com a sintaxe correta, podemos executá-lo pela própria IDE para o sensor captar os dados e mostrar em um terminal próprio. Dessa forma a API não tem acesso aos dados para os armazenar no banco, porém, é uma ótima maneira de testar quaisquer mudanças no código do sensor.

```
Message (Enter to send message to 'Arduino Uno' on 'COM5')

7203;
94;
95;
6;
4;
200;
95;
```

- CMD (NPM Start)

Podemos visualizar os dados coletados pelo sensor pelo próprio terminal CMD assim que ativamos a API com o comando NPM Start. Dessa maneira impossibilitamos que os mesmos resultados apareçam na IDE do Arduino, pois a porta USB já está sendo utilizada pelo Dat Acqui Ino.

```
API executada com sucesso na porta 3300
A leitura do arduino foi iniciada na porta COM5 utilizando Baud Rate de 9600
791;
791;
3;
3;
791;
202;
791;
94;
```

- Dados Inseridos no Banco MySQL

O objetivo primordial da API Dat Aqcu Ino, transportar os dados captados pelo sensor Arduino, no nosso caso o HC-SR04 (Ultrassônico), para o banco de dados do nosso projeto, onde os dados serão persistidos e posteriormente utilizados para a criação de dashboards.

	idLixeira	Profundidade	DataMedicao
▶	1	18.00	2025-10-01 09:57:36
	2	16.00	2025-10-01 09:57:37
	3	18.00	2025-10-01 09:57:38
	4	16.00	2025-10-01 09:57:39
	5	18.00	2025-10-01 09:57:40
	6	7.00	2025-10-01 09:57:41
	7	6.00	2025-10-01 09:57:42
	8	2.00	2025-10-01 09:57:43
	9	16.00	2025-10-01 09:57:44
	10	16.00	2025-10-01 09:57:45
	11	15.00	2025-10-01 09:57:46
	12	15.00	2025-10-01 09:57:47

- Gráficos na Web

Primeiro temos os gráficos dinâmicos. Sendo realizados pelo código HTML/CSS/JS, os gráficos são gerados em uma página web interpretada pelo navegador e mostram a variação dos dados captados pelos sensores ao longo do tempo, fazendo uso da API.

