

**INSTITUTO FEDERAL DE EDUCAÇÃO,
CIÊNCIAS E TECNOLOGIA
CAMPUS CAMPINA GRANDE**

PROJETO DE PESQUISA

Um Sistema Embarcado para sensoriamento de emissões gasosas de resíduos sólidos orgânicos

Registro Resumido de Atividades

Etapa : Desenvolvimento de Protótipo e Simulação

Aluno Bolsista: Iury Anderson Fernandes Coelho

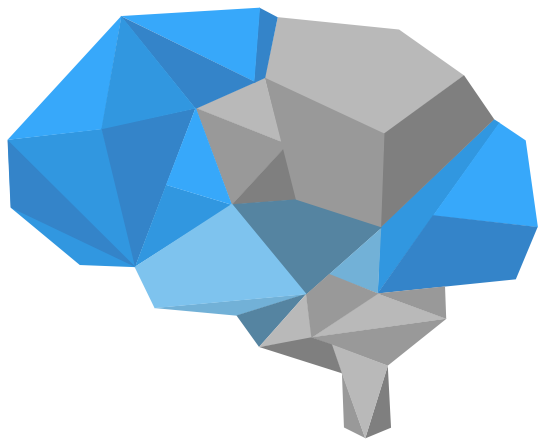
2019

Etapa de Simulação e Algoritmos

Este documento tem o intuito de registrar de maneira resumida as atividades envolvidas para a construção de um protótipo de software capaz de capturar e armazenar dados de sensores conectados a um nodemcu.

Esta documentação aborda os tópicos: entendimento da lógica do sistema, componentes de software necessários para a simulação, implementação de algoritmos, simulação e armazenamento de dados.

Atividades de Desenvolvimento do Protótipo de Captura



- 1 Entendimento da lógica do sistema;
- 2 Componentes do Sistema de Simulação;
- 3 Implementação dos algoritmos em PHP;
- 4 Implementação do algoritmo em C/C++;
- 5 Implementação do algoritmo em SQL;
- 6 Simulação e Armazenamento de dados.

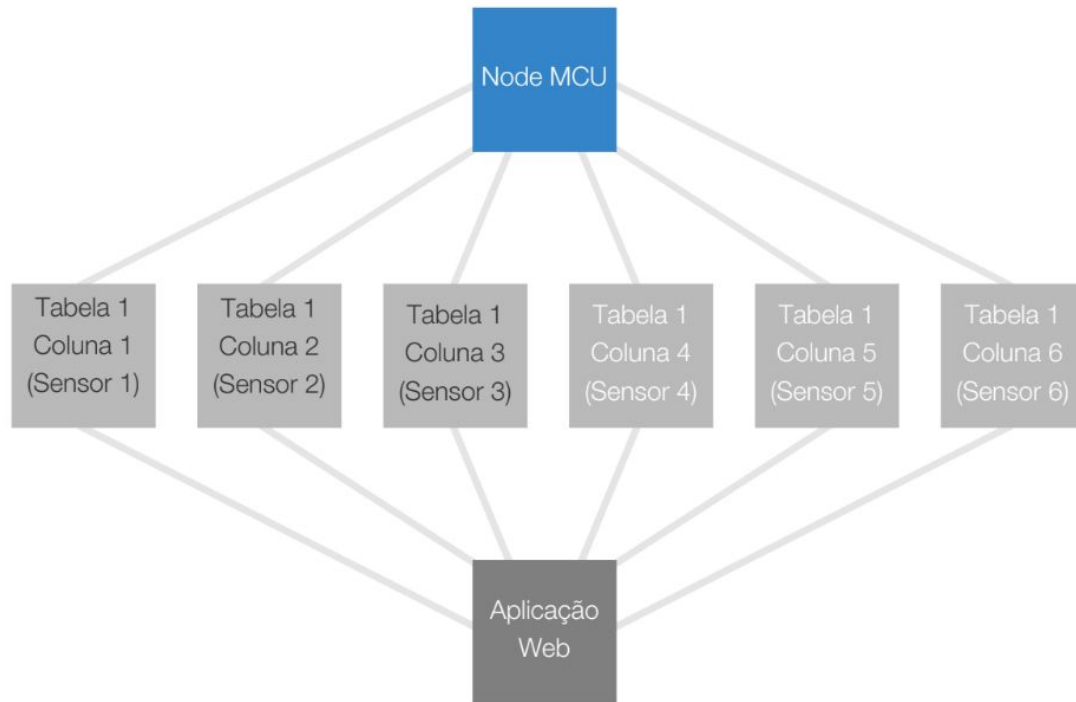
1. Lógica do Sistema

NodeMcu > Ler os valores dos sensores e envia-los para o banco de dados através de comunicação WiFi;

Banco de Dados > Armazena os valores lidos pelo node em tabelas SQL na ferramenta PhpMyAdmin;

Aplicação Web > Captura as informações das tabelas e as exibem no browser para o usuário.

1. Lógica do Sistema



2. Componentes do sistema de simulação

Tabela de softwares instalados		
Software	Versão	Propósito
Arduino IDE	1.8.2	Usado para fazer o upload de programas no nodemcu-esp8266.
Xampp	3.2.2	Integrar as ferramentas necessárias para o funcionamento de um servidor local.
PHP	5.2.0	Trabalhar diretamente com o backend no lado do servidor, e gerar conteúdo dinâmico localmente ou pela internet
MySQL	5.0.12	Sistematizar o gerenciamento de banco de dados
PhpMyAdmin	4.9.5	Manipulação e criação de tabelas de dados no SGBD.

2. Componentes do sistema de simulação

— — —

Tabela de algoritmos produzidos

Algoritmos em php	Algoritmo em C/C++	Algoritmo em SQL
index.php	nodemcu_wifi_mysql.ino	dbdados.sql
conexao.php		
salvar.php		

3. Implementação de Algoritmos em PHP

index.php

O algoritmo index.php é responsável pela primeira entrega de códigos ao servidor. Ele é o algoritmo 'main' e faz a ponte entre todas os outros códigos da aplicação para o servidor e o browser de internet.

```
82
83 include "conexao.php";
84
85 if ($_SERVER['REQUEST_METHOD'] == "POST") {
86     //echo "<h1> Recebeu a data : " . $_POST['data'] . "</h1>";
87     $dataPesquisa = $_POST['data'];
88     $dataArray = explode("/", $dataPesquisa);
89
90     $dataPesquisa = $dataArray[1] . "-" . $dataArray[0];
91
92     echo "data da pesquisa: " . $dataPesquisa;
93
94
95
96
97
98     $sql = "SELECT * FROM tbdados WHERE data_hora LIKE '%" . $dataPesquisa . "%'";
99
100 }else {
101     //echo "<h1> Não recebeu nada, vai mostrar mes atual</h1>";
102     $dataAtual = date('Y-m');
103     // echo "Mes Atual:" . $dataAtual ;
104     $sql = "SELECT * FROM tbdados WHERE data_hora LIKE '%" . $dataAtual . "%'";
105 }
106
107 $stmt = $PDO->prepare($sql);
108 $stmt ->execute();
109 echo "<table border='1'>";
110 echo "<tr>
111     <th> Sensor1</th>
112     <th> Sensor2</th>
113     <th> Sensor3</th>
114     <th>Data/Hora</th>
115 </tr>";
```


3. Implementação de Algoritmos em PHP

conexao.php

O algoritmo `conexao.php` é responsável por fazer a conexão entre os algoritmo da aplicação e o banco de dados a fim de que se possa salvar as informações nas tabelas do SGBD.

```
conexao.php
1 <?php
2     try {
3         $HOST = "localhost";
4         $BANCO = "bdnodemcu";
5         $USUARIO = "root";
6         $SENHA = "";
7
8         $PDO = new PDO("mysql:host=" . $HOST . ";dbname=" . $BANCO . ";charset=utf8", $USUARIO, $
          SENHA);
9
10    } catch (PDOException $erro){
11
12        echo "Erro de conexao, detalhe : " . $erro->getMessage();
13
14    }
15 }
16
17 ?>
```

3. Implementação de Algoritmos em PHP

salvar.php

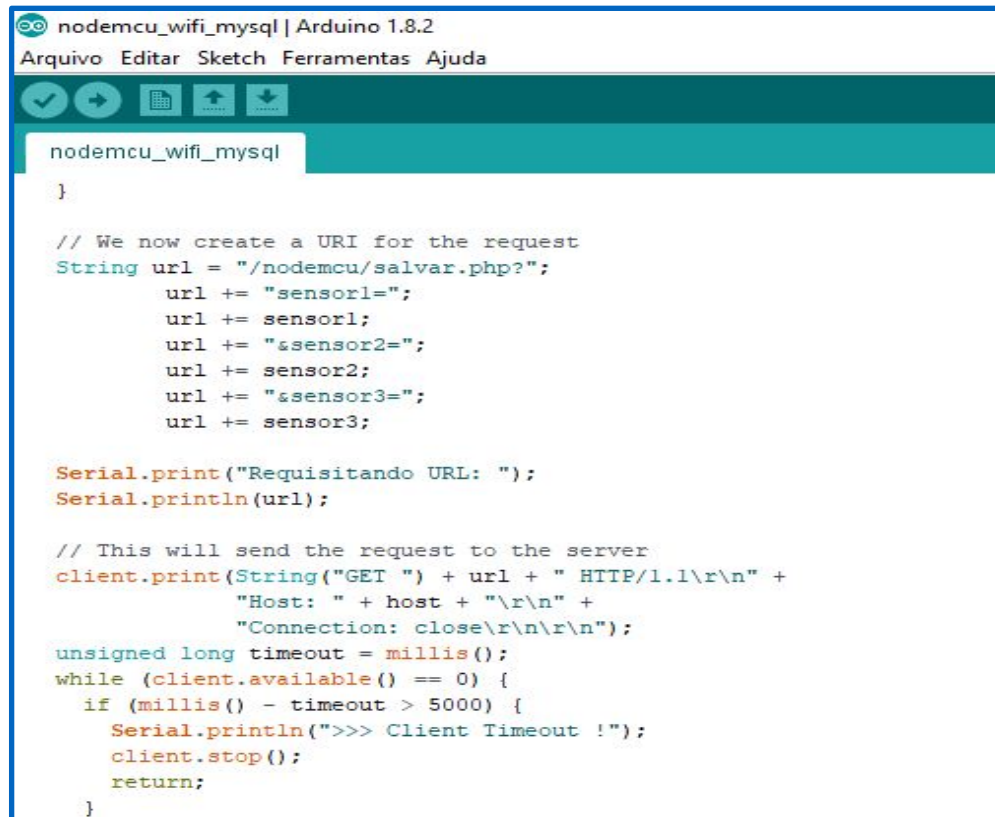
O algoritmo salvar.php é responsável pela inserção dos dados nas respectivas tabelas no banco de dados.

```
salvar.php
1
2
3 <?php
4 include "conexao.php";
5
6
7 $sensor1 = $_GET['sensor1'];
8 $sensor2 = $_GET['sensor2'];
9 $sensor3 = $_GET['sensor3'];
10
11
12 $sql = "INSERT INTO tbdados (sensor1, sensor2, sensor3) VALUES (:sensor1, :sensor2, :sensor3)"
13 ;
14
15 $stmt = $PDO->prepare($sql);
16
17 $stmt->bindParam(':sensor1', $sensor1);
18 $stmt->bindParam(':sensor2', $sensor2);
19 $stmt->bindParam(':sensor3', $sensor3);
20
21 if ($stmt->execute())
22 {
23     echo "salvo_com_sucesso";
24 }
25
26 else
27 {
28     echo "erro_ao_salvar";
29 }
30
31 ?>
32
```

4. Implementação de Algoritmos em C/C++

nodemcu_wifi_mysql.ino

Esse algoritmo é responsável pela transferência de dados do hardware nodemcu para o servidor através do protocolo de comunicação TCP/IP. Os valores dos sensores foram simulados a partir das variáveis sensor 1, sensor 2 e sensor 3 a fim de testar a conexão, captura e exibição dos dados.



```
nodemcu_wifi_mysql | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda

nodemcu_wifi_mysql

}

// We now create a URI for the request
String url = "/nodemcu/salvar.php?";
url += "sensor1=";
url += sensor1;
url += "&sensor2=";
url += sensor2;
url += "&sensor3=";
url += sensor3;

Serial.print("Requisitando URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
             "Host: " + host + "\r\n" +
             "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}
```

5. Implementação do Algoritmo em SQL

tbdados.sql

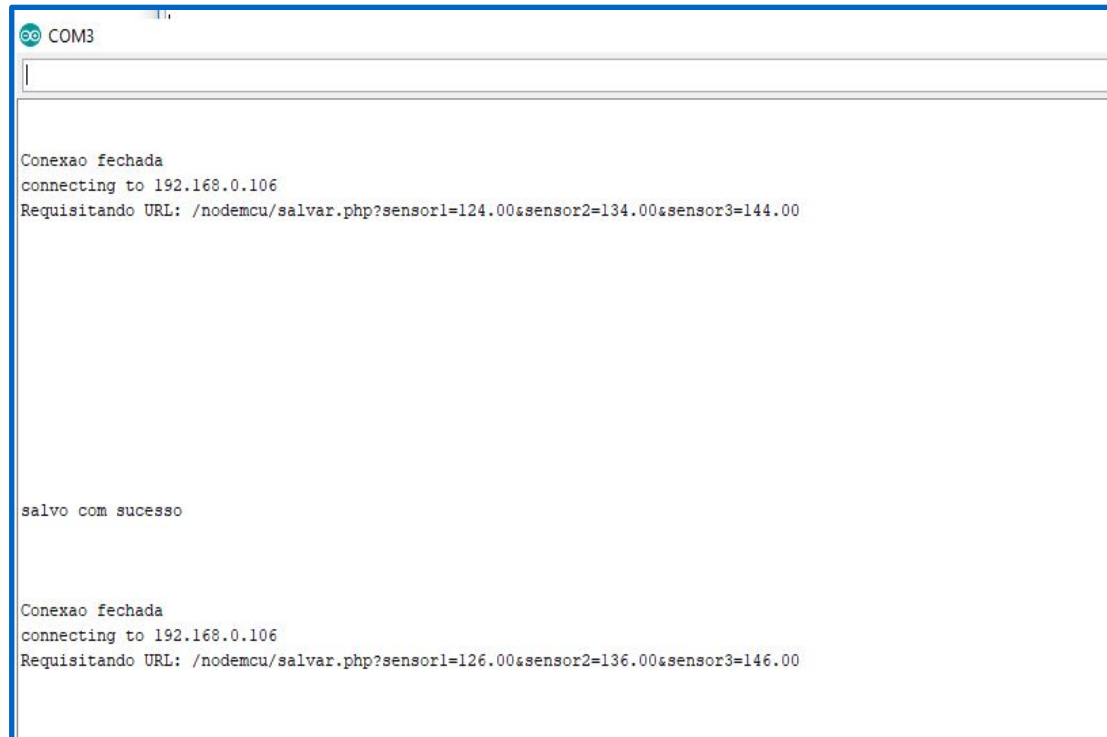
Esse algoritmo é responsável por construir a tabela que irá receber os dados enviados pelos sensores.

```
-- phpMyAdmin SQL Dump
1  -- phpMyAdmin SQL Dump
2  -- version 4.6.5.2
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Versão do servidor: 10.1.21-MariaDB
7  -- PHP Version: 5.6.30
8
9
10 CREATE TABLE `tbdados` (
11   `sensor1` varchar(10) NOT NULL,
12   `sensor2` varchar(10) NOT NULL,
13   `sensor3` varchar(10) NOT NULL,
14   `data_hora` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
15   `id` int(11) NOT NULL
16 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17
```

6. Simulação e Armazenamento de dados

— — —

A partir da tela de monitoramento serial da IDE do Arduino podemos constatar que os dados foram ou não salvos com sucesso na tabela de dados do SGBD através de uma conexão TCP/IP.



```
COM3

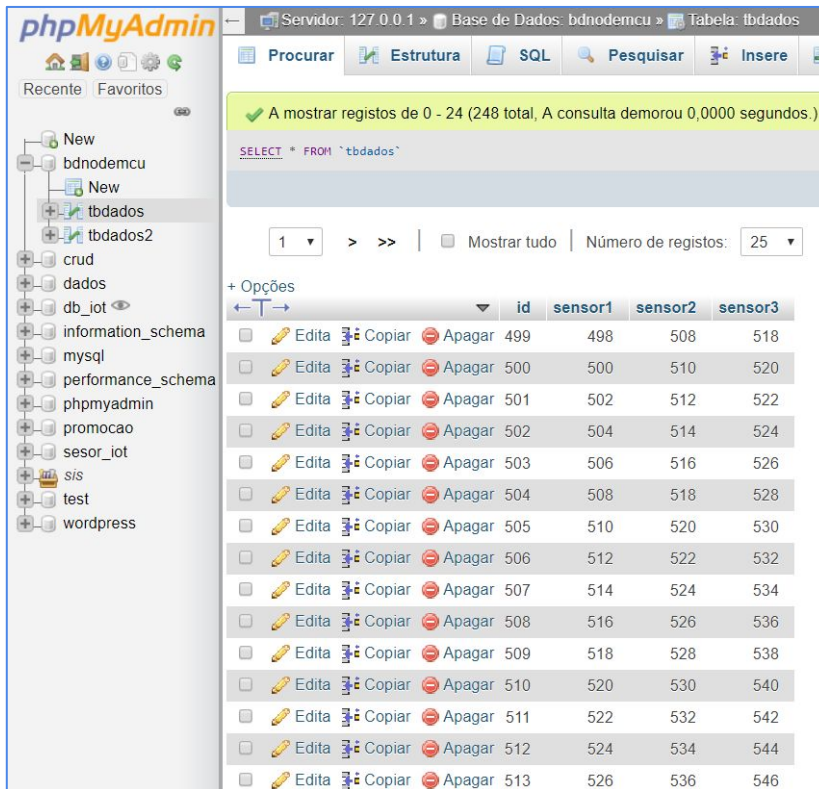
Conexao fechada
connecting to 192.168.0.106
Requisitando URL: /nodemcu/salvar.php?sensor1=124.00&sensor2=134.00&sensor3=144.00

salvo com sucesso

Conexao fechada
connecting to 192.168.0.106
Requisitando URL: /nodemcu/salvar.php?sensor1=126.00&sensor2=136.00&sensor3=146.00
```

5. Simulação e Armazenamento de dados

> Ao receber os dados de simulação dos sensores os produzidos pelo algoritmo nodemcu_wifi_mysql.ino foi possível popular a tabela de dados do SGBD implementada com a partir do tbdados.sql



The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure, including a database named 'bdnodemcu' with a table 'tbdados'. The main area shows the 'tbdados' table with 25 records. The table has columns: id, sensor1, sensor2, and sensor3. The records are displayed in a table format with options to edit or delete each row.

Server: 127.0.0.1 » Base de Dados: bdnodemcu » Tabela: tbdados

Procurar Estrutura SQL Pesquisar Inserir

Recente Favoritos

New bdnodemcu New tbdados tbdados2 crud dados db_iot information_schema mysql performance_schema phpmyadmin promocao sesor_iot sis test wordpress

✓ A mostrar registos de 0 - 24 (248 total, A consulta demorou 0,0000 segundos.)

SELECT * FROM `tbdados`

1 > >> | ☐ Mostrar tudo | Número de registos: 25 ▼

+ Opções

		id	sensor1	sensor2	sensor3
<input type="checkbox"/>	Editar Apagar	499	498	508	518
<input type="checkbox"/>	Editar Apagar	500	500	510	520
<input type="checkbox"/>	Editar Apagar	501	502	512	522
<input type="checkbox"/>	Editar Apagar	502	504	514	524
<input type="checkbox"/>	Editar Apagar	503	506	516	526
<input type="checkbox"/>	Editar Apagar	504	508	518	528
<input type="checkbox"/>	Editar Apagar	505	510	520	530
<input type="checkbox"/>	Editar Apagar	506	512	522	532
<input type="checkbox"/>	Editar Apagar	507	514	524	534
<input type="checkbox"/>	Editar Apagar	508	516	526	536
<input type="checkbox"/>	Editar Apagar	509	518	528	538
<input type="checkbox"/>	Editar Apagar	510	520	530	540
<input type="checkbox"/>	Editar Apagar	511	522	532	542
<input type="checkbox"/>	Editar Apagar	512	524	534	544
<input type="checkbox"/>	Editar Apagar	513	526	536	546

Resultados Alcançados

— — —

- > Configuração o NodeMcu com a rede local;
- > Captura e envio de dados remotamente por comunicação wifi;
- > Exibição dos parâmetros captados em tabelas do SGBD.