

Centro Universitário Senac

Gustavo Orlando Araújo Vergani

Jackson Henrique Ferreira

João Victor Farias Teixeira

Lucas da Silva Macedo

Maria Mercedes da Silva Rodrigues

PROJETO INTEGRADOR 2:

MONITORAMENTO DE VARIAÇÃO DE NÍVEL DE ÁGUA

São Paulo

2023

Gustavo Orlando Araújo Vergani

Jackson Henrique Ferreira

João Victor Farias Teixeira

Lucas da Silva Macedo

Maria Mercedes da Silva Rodrigues

PROJETO INTEGRADOR 2:

MONITORAMENTO DE VARIAÇÃO DE NÍVEL DE ÁGUA

Projeto Integrador apresentado ao curso de Análise e Desenvolvimento de Sistemas, do Centro Universitário Senac – Santo Amaro, como requisito para obtenção de aprovação em disciplina.

Orientadores: Alexandre Igosheff e Evandro Teruel.

São Paulo

2023

ÍNDICE DE ILUSTRAÇÕES

<i>Figura 1:Montagem no dia da apresentação</i>	10
Figura 2:Sistema Montado	11
Figura 3: Sistema Elétrico 2	11
Figura 4: Sistema Elétrico	11
Figura 5: Sistema Elétrico 3	12
Figura 6: Erro	13
Figura 7: Void Setup	14
Figura 8: Void Loop	15
Figura 9: CRUD	16

LISTA DE ABREVIATURAS E SIGLAS

IoT – Internet of Things (Internet das Coisas).

LED – Light Emitting Diode (Diodo Emissor de Luz).

SQL – Structured Query Language (Linguagem de Consulta Estruturada).

BD – Banco de Dados.

CRUD – Create, Read, Update, Delete (Criar, Ler, Atualizar, Deletar).

WEB – World Electronic Base.

WI-FI – Wireless Fidelity.

SUMÁRIO

INTRODUÇÃO	6
CRONOGRAMA ATUALIZADO	7
HARDWARE E PROGRAMAÇÃO	7
CONSTRUÇÃO DO SISTEMA	10
MONTAGEM DO SISTEMA.....	10
STREAMING DE DADOS	13
CÓDIGO FONTE	14
APLICAÇÃO WEB.....	16
CONCLUSÃO	17
REFERÊNCIAS.....	18

INTRODUÇÃO

Este trabalho tem o objetivo de apresentar o protótipo funcional do projeto de IoT, referente à segunda apresentação da disciplina de Projeto Integrador. Nós desenvolvemos um protótipo de sensor de nível para detectar se um reservatório está com água entre cheio e vazio.

O NodeMCU ESP-8266 foi o microcontrolador escolhido para realizar a automatização do controle hídrico. Ele atuará em conjunto com os componentes definidos pelo grupo. Pela sua capacidade de conexão com o WI-FI, encaminharemos as medições obtidas diretamente para uma base de dados. Essas informações serão exibidas através de uma aplicação Web.

CRONOGRAMA ATUALIZADO

26/10 – Realização de testes do Protótipo.

27/10 – Compra do reservatório.

29/10 – Realização de testes do Protótipo.

31/10 – Elaboração do Streaming de Dados.

07/11 – Segunda Apresentação.

HARDWARE E PROGRAMAÇÃO

Listagem das especificações técnicas de cada produto utilizado até o momento.

Sensor Nível de Água - Boia Horizontal

Modelo	CS-C0058
Marca	OEM
Tensão Máxima de Contato	220V DC/AC
Classificação Máxima de Contato	10W
Corrente de Comutação Máxima	0.5A
Tensão de Ruptura Máxima	100V DC/AC
Corrente Máxima	1.0A
Resistência Máxima de Contato	100m Ohms
Faixa de Temperatura	-10 ~ +60°
Material do Flutuador	P.P
Comprimento do Cabo	36 cm
Composição	Plástico, Metal
Tamanho	56mm Largura x 24mm de Comprimento x 24mm de Altura
Peso	10 g

ESP8266 NodeMCU

Módulo NodeMcu Lua ESP-12E
Versão do módulo: V2
Memória flash: 4 MB
Tensão de operação: <ul style="list-style-type: none">• Pinos Digitais: 3,3 V• Pino Analógico: 1,0 V
Wireless padrão 802.11 b/g/n
Antena embutida
Conector micro-usb para programação e alimentação
Modos de operação: STA/AP/STA+AP
Suporta 5 conexões TCP/IP
Portas GPIO: 13
GPIO com funções de PWM, I2C, SPI, etc.
Resolução do PWM: 10 bits (valores de 0 a 1023)
01x Conversor analógico digital (ADC)
Distância entre pinos: 2,54 mm
Dimensões: 49 x 26 x 7 mm (sem considerar os pinos)

Protoboard

Furos	400
Material	Plástico ABS
Resistência de Isolamento	100MO min
Tensão Máxima	500v AC por minuto
Faixa de Temperatura	- 20 a 80°C
Dimensões	8,3 x 5,5 x 1,0 cm
Para terminais e condutores de 0,3 a 0,8 mm (20 a 29 AWG)	

Jumpers

Tipo	Macho/Macho
Comprimento	10cm

LEDs

Cores	Verde, Amarelo e Vermelho.
--------------	----------------------------

Resistor

Valor	220 Ohms
Tolerância	5%
Potência	1/4W

Recipiente

Litragem	3 litros e 600 mililitros
-----------------	---------------------------

CONSTRUÇÃO DO SISTEMA

O fio GND é conectado no polo negativo da protoboard, o qual se conecta aos polos negativos da Boia, e dos LEDs juntamente com seus resistores.

Temos os pinos de entrada dos sensores, sendo eles: o pino D1 e o pino D2, que estarão ligados ao polo positivo da protoboard. O pino de distribuição 3v também estará conectado ao polo positivo da protoboard.

Como saída, temos o pino D3, D4 e D5, os quais após receber o estado do sensor, irão acender de acordo com o nível de água presente.

MONTAGEM DO SISTEMA

Figura 1: Montagem no dia da apresentação

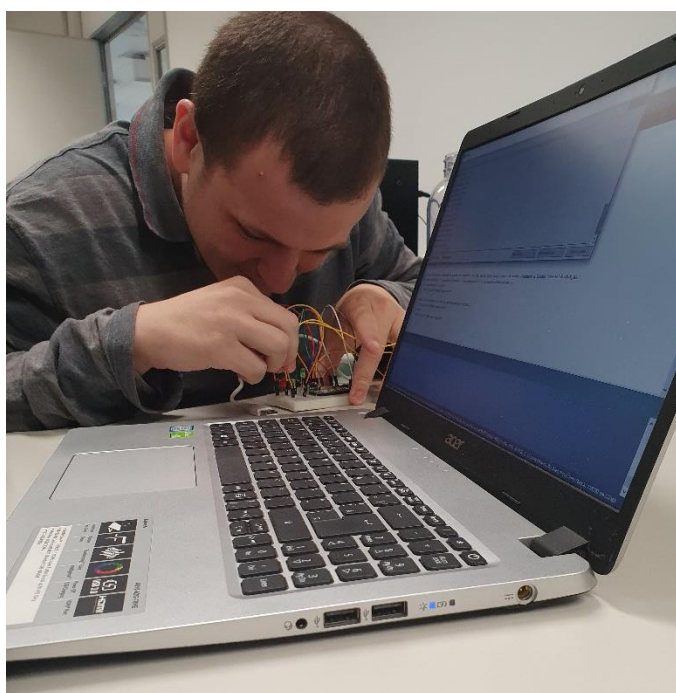


Figura 2: Sistema Montado

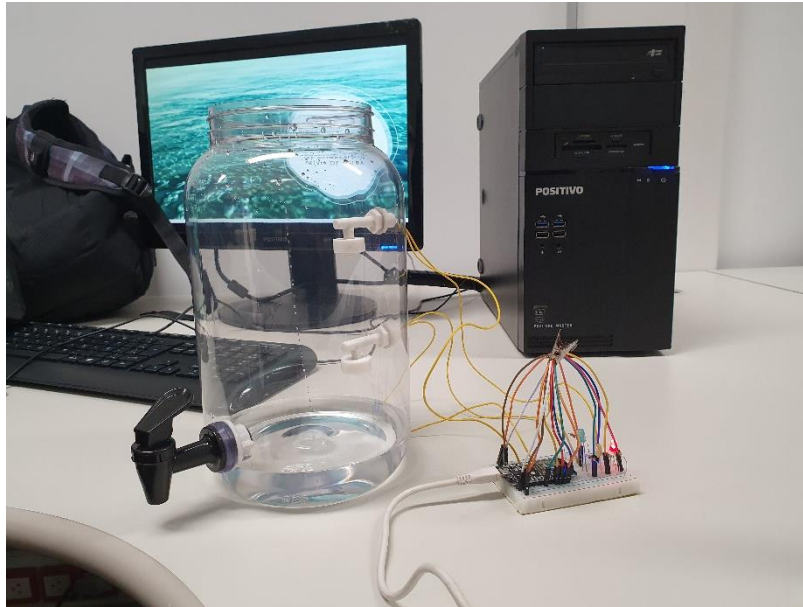


Figura 4: Sistema Eléctrico

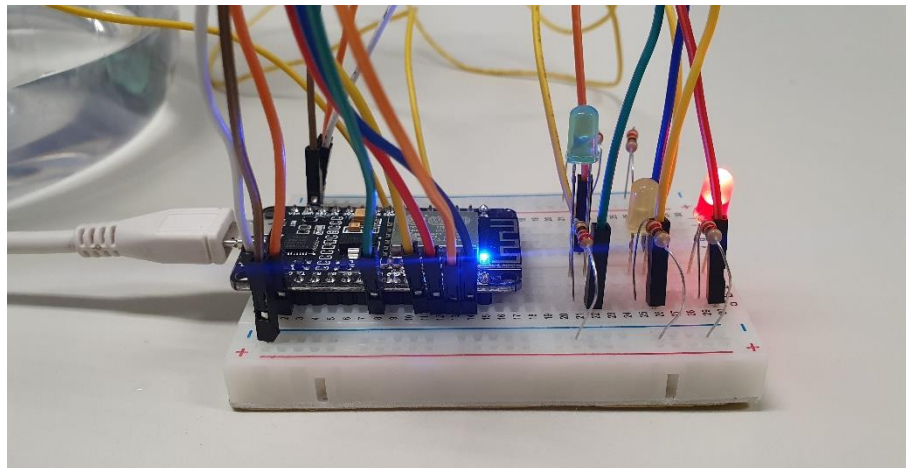


Figura 3: Sistema Eléctrico 2

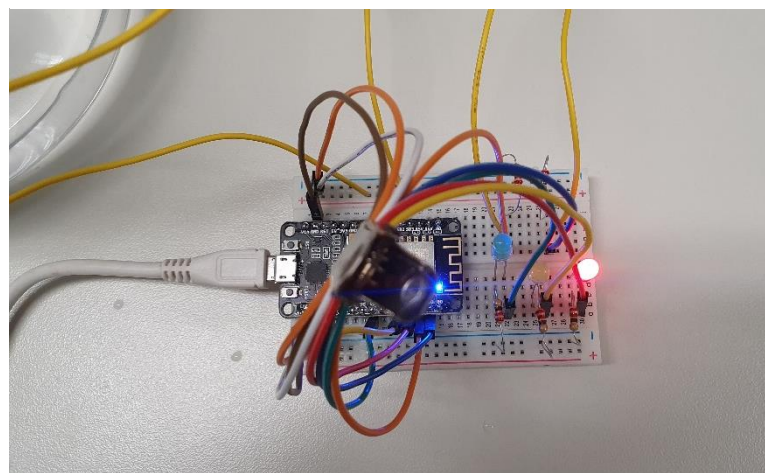
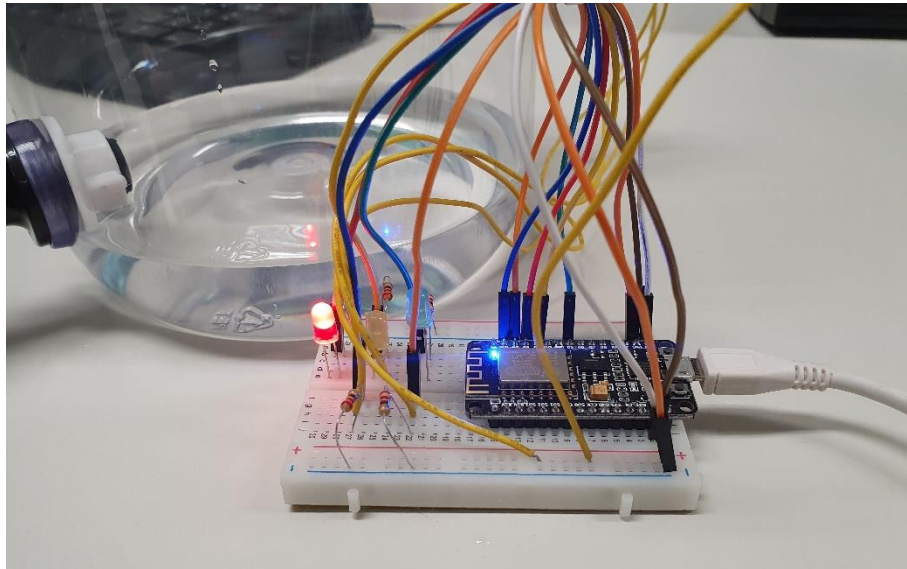


Figura 5: Sistema Elétrico 3



STREAMING DE DADOS

A proposta inicial de streaming de dados era conectar o ESP-8266 diretamente ao Banco de Dados utilizando a biblioteca MySQL Connector Arduino. A biblioteca apresentou um erro ao se conectar com a placa. O erro acontece pela má declaração de algumas funções de um arquivo dentro da biblioteca. A seguinte imagem exibe o erro:

```
C:\Users\joao\Documents\Arduino\libraries\MySQL_Connector_Arduino\src\MySQL_Encrypt_Sha1.
cpp: In member function 'virtual size_t Encrypt_SHA1::write(uint8_t)':

C:\Users\joao\Documents\Arduino\libraries\MySQL_Connector_Arduino\src\MySQL_Encrypt_Sha1.
cpp:89:1: error: no return statement in function returning non-void [-Werror=return-type]

 89 | }
    | ^

C:\Users\joao\Documents\Arduino\libraries\MySQL_Connector_Arduino\src\MySQL_Encrypt_Sha1.
cpp: In member function 'virtual size_t Encrypt_SHA1::write(uint8_t*, int)':

C:\Users\joao\Documents\Arduino\libraries\MySQL_Connector_Arduino\src\MySQL_Encrypt_Sha1.
cpp:95:1: error: no return statement in function returning non-void [-Werror=return-type]

 95 | }
    | ^

cc1plus.exe: some warnings being treated as errors
exit status 1

Erro compilando para a placa NodeMCU 1.0 (ESP-12E Module)
```

Figura 6: Erro

Decidimos, então, utilizar a biblioteca MySQL_MariaDB_Generic. Todas as credenciais do BD (endereço do servidor, o nome de usuário e a senha, e o nome do BD) estarão declaradas no código fonte. A partir desses dados, a placa, conectada a uma rede WI-FI, consegue se comunicar com o BD

CÓDIGO FONTE

Apresentação do código fonte em linguagem C, desenvolvido para a automação do sistema projetado.

As primeiras seis linhas de código referem-se à definição dos elementos da pinagem. Em seguida, as variáveis para guardar as condições dos sensores são declaradas no tipo booleana. Posteriormente, podemos visualizar na imagem abaixo a inicialização do setup, com as configurações dos pinos de entrada e saída.

```
#define sensorTopo D1
#define sensorBaixo D2
#define LED_1 D3
#define LED_2 D4
#define LED_3 D5

bool sensorTopoAtivo = false;
bool sensorBaixoAtivo = false;
bool ultimoSensorTopoAtivo = false;
bool ultimoSensorBaixoAtivo = false;

void setup() {
    Serial.begin(9600);
    |
    pinMode(sensorTopo, INPUT);
    pinMode(sensorBaixo, INPUT);
    pinMode(LED_1, OUTPUT);
    pinMode(LED_2, OUTPUT);
    pinMode(LED_3, OUTPUT);

    Serial.println("Estado das boias");
    Serial.println();
}
```

Figura 7: Void Setup

Dentro do loop, nós temos atribuições de valores às variáveis que guardam o estado dos sensores, e a configuração dos leds de acordo com o estado dos sensores; a condição verifica se houve alteração no estado dos sensores; por fim, há variáveis para guardar última checagem dos sensores:

```
void loop() {
  sensorTopoAtivo = digitalRead(sensorTopo) == HIGH;
  sensorBaixoAtivo = digitalRead(sensorBaixo) == HIGH;

  digitalWrite(LED_1, !sensorTopoAtivo && !sensorBaixoAtivo);
  digitalWrite(LED_2, sensorBaixoAtivo && !sensorTopoAtivo);
  digitalWrite(LED_3, sensorTopoAtivo && sensorBaixoAtivo);
  digitalWrite(LED_3, sensorTopoAtivo);

  delay(10);

  if (sensorTopoAtivo != ultimoSensorTopoAtivo || sensorBaixoAtivo != ultimoSensorBaixoAtivo) {
    if (sensorTopoAtivo == HIGH) {
      Serial.println("Sensor topo ativo");
    }
    else if (sensorTopoAtivo == LOW && sensorBaixoAtivo == HIGH) {
      Serial.println("Sensor baixo ativo");
    } else {
      Serial.println("Sensores inativos");
    }
  }

  ultimoSensorTopoAtivo = sensorTopoAtivo;
  ultimoSensorBaixoAtivo = sensorBaixoAtivo;

  delay(1000);
}
```

Figura 8: Void Loop

APLICAÇÃO WEB

Arquivo “Java Web”, nas linguagens HTML, CSS e Javascript, que se relacionará com o BD. As informações armazenadas serão exibidas na aplicação web que será desenvolvida. A imagem 5 representa um esboço do front-end.



Figura 9: CRUD

CONCLUSÃO

Este trabalho apresentou o desenvolvimento do sistema a partir do planejamento feito nas datas anteriores. Conseguimos concluir todos os objetivos definidos no trabalho anterior. Todos os elementos que compõem este trabalho estão sendo elaborados com êxito. Estamos muito satisfeitos com os resultados da nossa dedicação e compromisso em atender todas as especificações do projeto. Estamos incluindo os conhecimentos de todas as áreas estudadas desde o início do curso, até o momento, o que leva ao aperfeiçoamento das nossas habilidades e aprendizados. O próximo passo é adicionar um display, para complementar o sistema.

REFERÊNCIAS

DAVIS, Stephen. C++ para Leigos. 7ª Edição. Editora Alta Books. Publicado em: 10 de junho de 2016.

OLIVEIRA, C; ZANEETI, H; NABARRO, C; GONÇALVES, J. Aprenda Arduino: Uma abordagem prática. 1ª Edição. Katzen Editora. Publicado em 2018.

RANGEL, Gabriel. Ebook Internet das Coisas para iniciantes. Eletrônica Ômega. Disponível em: <https://blog.arduinoomega.com/ebooks/Eletronica-Omega-Ebook-IoT-Para-Iniciantes.pdf>