

Pontifícia Universidade Católica de Minas Gerais (PUC Minas)  
ICEI - Instituto de Ciência Exatas e Informática  
Engenharia de Computação  
Sistemas Embarcados

# **Controle do consumo de água**

Alunos(as): Caio José Rocha Silva, João Pedro Oliveira Diorio e  
Yago Assis M. Faria

Professores(as): Mario Guimaraes Buratto

Data: Junho de 2023

**Abstract.** *The present document aims to understand a little bit more about the implementation planning of the water consumption analysis and control project as a whole.*

**Keywords:** *Hardware; Software; Firmware; Conscious Consumption.*

**Resumo.** *O presente documento visa compreender um pouco mais sobre a planejamento de implementação do projeto de análise e controle do consumo de água como um todo.*

**Palavras-chave:** *Hardware; Software; Firmware; Consumo Consciente.*

## 1. Introdução

O projeto em questão visa controlar e analisar o consumo de água em um espaço, seja residencial ou industrial. Esse projeto é ideal para clientes e usuários que desejam se informar melhor sobre o consumo de água do espaço escolhido, podendo assim detectar vazamentos de água (caso haja um consumo em horário não comercial, ou determinado não comum), e analisar o horário que há mais consumo. Para isso, o cliente/usuário precisaria de ter: o dispositivo Hardware a ser implementado pela equipe; e um login no site desenvolvido (MyConsumption.io).

## 2. Informações Gerais

A ideia principal foi a de implementar um pequeno dispositivo de Hardware na caixa d'água na qual, assim que haja vazão de água, é feito um registro dessa vazão para informar o cliente do evento. Os dados são enviados para uma plataforma, e o site de análise consulta e requer os dados dessa plataforma a cada determinado tempo.

O público alvo do projeto são cidadãos que gostariam de reduzir o consumo de água, ou simplesmente controlar e analisar seu consumo em um determinado período de tempo (por exemplo, quantos litros o cliente gasta ao lavar a louça, registrando o tempo de início e o tempo final da ação para analisar na ferramenta). Além disso, outro público alvo seriam grandes empresas ou o setor industrial em geral, para manter o controle de consumo.

O objetivo do projeto seria de monitorar o consumo diário de água de um cliente, além de levantar dados de consumo, traçar um perfil do consumidor e disponibilizar relatórios de consumo.

### **3. Desenvolvimento durante o semestre**

Com o prazo de seis meses para realizar o projeto as tarefas foram divididas da seguinte forma:

#### **3.1. Desenvolvimento do Hardware (25/02/2023 - 15/03/2023)**

- Pesquisar e adquirir os componentes eletrônicos necessários.
- Realizar a montagem e conexão dos componentes no protoboard.
- Programar o NodeMCU-ESP8266 para realizar a medição do consumo de água e enviar os dados para o broker TagoIO.
- Realizar testes de comunicação entre o hardware e o broker para garantir a transmissão correta dos dados.

#### **3.2. Desenvolvimento do Firmware (15/03/2023 - 05/04/2023)**

- Desenvolver o código do firmware para o NodeMCU-ESP8266.
- Implementar as requisições POST para enviar os dados coletados para o broker TagoIO.
- Programar o NodeMCU-ESP8266 para realizar a medição do consumo de água e enviar os dados para o broker TagoIO.
- Realizar testes unitários para verificar se as requisições estão sendo feitas corretamente e se os dados são armazenados corretamente no broker.

#### **3.3. Desenvolvimento do Software (05/04/2023 - 25/04/2023)**

- Criar o servidor utilizando Node.js e Express.js
- Desenvolver as rotas e os controladores necessários para interagir com o broker TagoIO.
- Implementar a lógica de consulta e tratamento dos dados do consumo de água.
- Realizar testes unitários para verificar se os dados são corretamente consultados e tratados no servidor.

#### **3.4. Desenvolvimento do Front-End (25/04/2023 - 10/05/2023)**

- Criar as páginas do site utilizando HTML, CSS e JavaScript.
- Implementar a interface de seleção de propriedades, período de dias e tipo de agrupamento.
- Integrar as chamadas de API para buscar os dados do consumo de água do servidor.
- Realizar testes de interface para verificar se os dados são corretamente exibidos no site..

#### **3.5. Testes Integrados (10/05/2023 - 08/06/2023)**

- Realizar testes de integração entre o hardware, firmware, software e front-end.
- Simular diferentes cenários de consumo de água e verificar se os dados são coletados, transmitidos, armazenados e exibidos corretamente.
- Registrar e corrigir quaisquer erros ou falhas encontrados durante os testes.
- Realizar testes de desempenho para avaliar a capacidade do sistema em lidar com um grande volume de dados de consumo de água.

Para saber mais sobre os testes eles estão

## 4. Objetivo do Hardware

Atuando como principal dispositivo de IOT, nosso hardware vem com objetivo primário de possibilitar a coleta de dados de consumo através de módulos/dispositivos de medição de fluxo de água ou em linhas gerais o consumo hidrico. Como objetivo secundário o Hardware atuará como despachante de dados, realizando o envio dos dados coletados para um servidor, possibilitando que as outras camadas do projeto possam atuar encima dos dados coletados e disponibilizados ma rede.

### 4.1. Iteração com usuário

Como primeira forma de interação o usuário poderá realizar a instalação do sensores medidores de fluxo/vazão e conectarão o sistema ao wi-fi que desejar, com essas duas premissas realiza basta que ele comece a realizar seus consumos de água corriqueiros e assim seus dados serão enviados para os servirdores que farão a organização dos seus dados na plataforma web.

### 4.2. Arquitetura e Diagrama em blocos

O projeto será foi dividido em 3 grandes partes e feito na seguinte ordem: Hardware, Middleware e Software. Nas próximas sub-seções, seguirão explicações mais detalhadas sobre cada parte do projeto.

O fluxo de acordo com a Figura 2: Haverá coleta de dados do consumo de água pelo ESP32, os dados são enviados para nosso broker tagoIO e são armazenados no próprio database da tagIO. Depois há um script que coleta os dados a partir do horário que o usuário escolher, o script faz a coleta de dados no tagIO depois joga no database do site e mostra para o usuário os dados de acordo com o request.

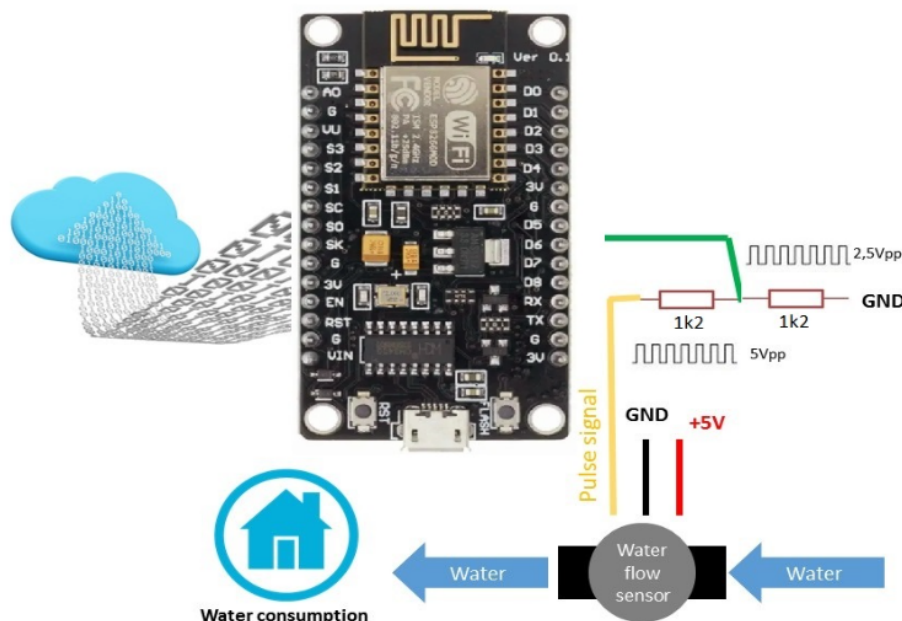


Figura 1. Diagrama Hardware implementado

## 5. Hardware

Tendo como foco principal a medição do consumo hídrico dentro de uma residência, foi realizado um levantamento de componentes eletrônicos que compuseram o sistema criado para medição, cálculo e tratamento dados.

Como componentes selecionados para o sistemas utilizaremos o NodeMCU-ESP8266 como microcontrolador principal que utiliza o protocolo TCP/Ip e o processador Tensilica LX106 160MHz, logo tendo predominantemente suas características voltadas para a criação de aplicações de IOT, para medição do consumo utilizaremos um sensor de fluxo/vazão de água YF-S201 (Efeito Hall), e como Broker, utilizaremos a plataforma TagoIO que fornece a hospedagem de dados por 30 dias.

Visando integrar os sistemas, o código do ESP-8266 realizará algumas requisições do tipo POST para o Broke, e as mesmas permanecerão armazenadas em um bucket que será acessado por uma aplicação criada para integrar os dados coletados ao nosso banco de dados pessoal.

A montagem do hardware será realizada em um protoboard com auxilio de fios rígidos, na qual o ESP-8266 será alimentado por uma fonte de 5V e pela lógica criada ele operará em light mode, realizando requisições para o bucket da TagoIO apenas ao ocorrer medições acima de 0,3L/min.

A fórmula básica utilizada para o cálculo do fluxo foi encontrada no datasheet do módulo YF-S201 no Fluxo em L/min = Frequência dos pulsos (Hz) / 7.5, utilizando coleta de dados baseado na interrupção de uma borda de subida (`attachInterrupt(rising)`), Importante ressaltar que o sensor possui características lineares.

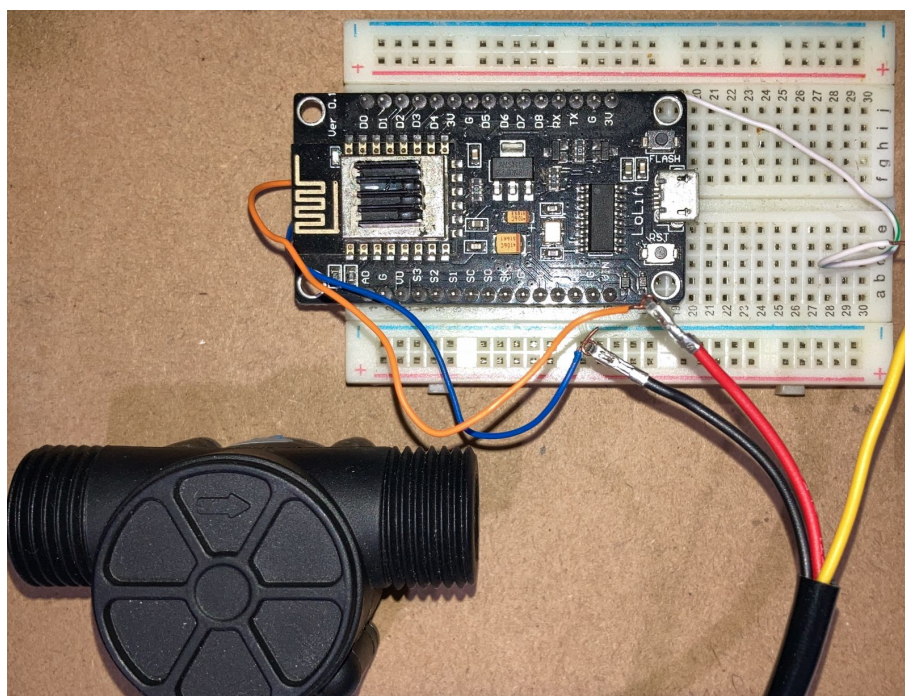


Figura 2. Montagem do Hardware

## 6. Firmware

Essa é a parte responsável por conectar o broker "TagoIO" ao servidor que o site utiliza.

A TagoIO oferece as ferramentas para sua empresa gerenciar dispositivos, armazenar dados, executar análises e integrar serviços. Ele combina tudo com um aplicativo fácil de usar e um sistema de gerenciamento de usuários. Essa plataforma permitiu que os dados do Hardware sejam enviados para a nuvem e para que o servidor possa consumi-los, ou seja, um broker.

O consumo desses dados foi feito em Node.js - a ser citado na **seção 7: Software**, na qual informa quais pacotes foram utilizados - por meio de uma request na API do broker. Nela devemos passar a autorização e o horário dos dados (que foi setado como chave primária), uma start date e uma end date ambas no formato ano/mês/dia hora:min:seg, Exemplo: 2022/06/05 22:30:00. Os dados retornados são no formato JSON.

Após feita as requests o servidor faz um tratamento nesse JSON e armazena os dados dentro de um banco de dados SQLite. Esse banco de dados é o responsável por alimentar o site por meio das rotas request, response feita na **seção 7: Software**.

## 7. Software

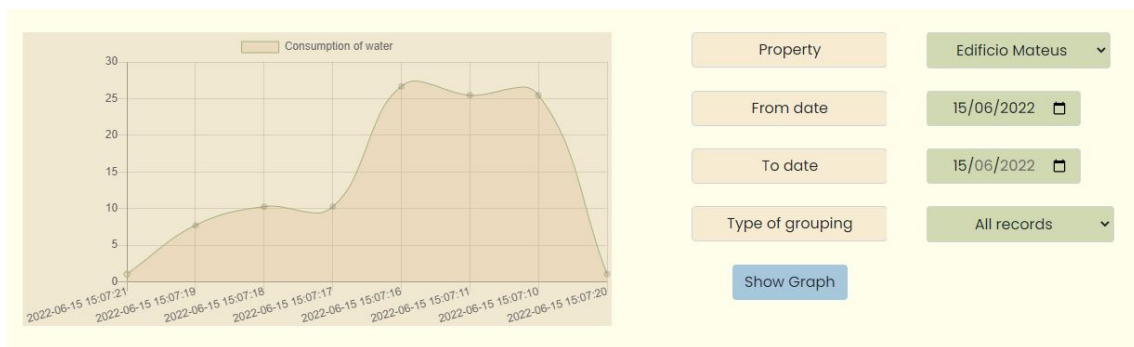
O site foi implementado com HTML, CSS, JS, além de utilizar Node.js (tecnologia usada para executar código JavaScript fora do navegador - para realizar as consultas necessárias no banco de dados) e Express.js (framework para o Node.js que permite otimizar a construção de aplicações web). Ademais, foram utilizados alguns pacotes NPM - para facilitar a implementação do Back-End do site - como: body-parser, sqlite, nodemon, jquery, jsdom, path, cron, ejs, entre outros.

A primeira coisa que o site faz (pelo Back-End) é carregar as propriedades (Edifícios Residenciais, salas, entre outros) do usuário, para mostrar na opção de selecionar. Assim que o site é carregado, as propriedades são carregadas, também, na interface.

Enquanto isso no Back-End, foi criada uma função utilizando um pacote NPM (node-cron) que funciona como um agendador de tarefas, na qual a cada determinado tempo, insere os dados que estão na plataforma Tago.IO - citada na **seção 6: Firmware** - no banco de dados.

Já na interface, o usuário irá selecionar a propriedade na qual ele quer ver os dados de consumo, além de selecionar um período de dias e um tipo de agrupamento. Existem dois agrupamentos possíveis: todos os registros, e diário. O agrupamento de "todos os registros", como já diz o nome, mostra todos os registros de um determinado período de tempo. Esse agrupamento é indicado para usar com apenas um dia (uma vez que mostrará todos os registros desse dia - que podem ser muitos -, onde o cliente poderá observar qual foi o período de maior gasto), conforme a **Figura 3**.

Já o agrupamento "diário", agrupa todos os registros de um dia, em um "dado", somando todo o consumo desse dia. Esse tipo de agrupamento é recomendado para usar com uma seleção maior de dias, na qual assim o cliente conseguirá observar em qual dia seu consumo foi maior, não focando num horário, conforme a **Figura 4**.



**Figura 3. Todos os registros do dia 15/06**



**Figura 4. Soma dos registros de cada dia, durante os dias 06/06 e 12/06**

## 8. Referencias bibliográficas

- NPMJS
- Node Js doc
- TagoIO API doc
- Stack Overflow
- Repositório github