**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS**

**FATEC PROFESSOR Jessen Vidal**

**GABRIEL DORNELAS DOS SANTOS**

**WEBSENSOR - MONITORAMENTO DE DADOS PROVIDOS DE SENSOR**

São José dos Campos

2017

**GABRIEL DORNELAS DOS SANTOS**

**WEBSENSOR - MONITORAMENTO DE DADOS PROVIDOS DE SENSOR**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**Orientador: Professor Leônidas Melo.**

São José dos Campos

2017

**Dados Internacionais de Catalogação na Publicação (CIP)**

**Divisão de Informação e Documentação**

Dos Santos, Gabriel Dornelas

Websensor - monitoramento de dados providos de sensor.

São José dos Campos, 2017.

33f. (número total de folhas do TG)

Trabalho de Graduação – Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

FATEC de São José dos Campos: Professor Jessen Vidal, 2017.

Orientador: Professor Leônidas Melo.

1. Sensor. 2. NodeMcu. 3. MQTT. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Título

**REFERÊNCIA BIBLIOGRÁFICA**

Dos Santos, Gabriel Dornelas. **Websensor - monitoramento de dados providos de sensor.** 2017. 33f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

**CESSÃO DE DIREITOS**

NOME DO AUTOR: GABRIEL DORNELAS DOS SANTOS

TÍTULO DO TRABALHO: Websensor - monitoramento de dados providos de sensor

TIPO DO TRABALHO/ANO: Trabalho de Graduação/2017.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  GABRIEL DORNELAS DOS SANTOS  Avenida José Pereira de Andrade, 216  12328-290, Jacareí - São Paulo |  |

**GABRIEL DORNELAS DOS SANTOS**

**WEBSENSOR - MONITORAMENTO DE DADOS PROVIDOS DE SENSOR**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Professor, Leônidas Melo – FATEC**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Professor, Antônio Graça – FATEC**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Professor, Diogo Branquinho – FATEC**

**\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_**

**DATA DA APROVAÇÃO**

**RESUMO**

Este projeto foi desenvolvido para apresentar a possibilidade de monitoramento e visualização de dados providos de sensores, como os de plantas industriais, utilizando a tecnologia de internet das coisas. Houve a iniciativa de um projeto onde uma planta industrial didática será desenvolvida e com ela um software para monitorar os sensores atrelados na mesma. Foi criada uma solução para o software deste problema, um protótipo de sistema para realizar o monitoramento de temperatura e umidade ambiente de maneira inteligente e online, onde o usuário pode visualizar os dados em tempo real pelo aplicativo de monitoria. Desenvolvido em auxilio da metodologia lean, com encontros semanais com o usuário final, para receber feedbacks, criando assim, melhorias no software. Os resultados obtidos foram satisfatórios considerando todas variáveis que acercavam o projeto. Em conclusão pode-se afirmar que o sistema Websensor sanou o problema proposto de monitorar os dados de um sensor de maneira inteligente, tendo a opção de salvar os dados para possíveis futuras análises.

**Palavras-Chave**: Sensor; NodeMcu; MQTT; Arduino; Lean; Python.

**ABSTRACT**

This project was developed to present the possibility of monitoring and visualizing data provided by sensors, such as industrial plants, using the technology of internet of things. There was the initiative of a project where a didactic industrial plant will be developed with a software to monitor the sensors attached to it. A solution was created for the software of this problem, a prototype system to perform the monitoring of ambient temperature and humidity in an intelligent and online way, where the user can view the data in real time by the monitoring application. Developed in support of the lean methodology, with weekly meetings with the end user, to receive feedbacks, thus creating improvements in the software. The results were satisfactory considering all variables that approached the project. In conclusion, it can be stated that the Websensor system solved the proposed problem of monitoring the data of a sensor in an intelligent way, having the option to save the data for possible analyzes.

**Keywords:** Sensor; NodeMcu; MQTT; Arduino; Lean; Python.

**LISTA DE FIGURAS**

[Figura 1 - Layout. 13](#_Toc500283941)

[Figura 2 Ambiente Arduino 14](#_Toc500283942)

[Figura 3 Placa ESP8266 15](#_Toc500283943)

[Figura 4 Layout MQTT 16](#_Toc500283944)

[Figura 5 Sensor DHT11 17](#_Toc500283945)

[Figura 6 Método Lean 20](#_Toc500283946)

[Figura 7 Diagrama de caso e uso 22](#_Toc500283947)

[Figura 8 Diagrama de Fluxo 23](#_Toc500283948)

[Figura 9 Configuração do sensor 24](#_Toc500283949)

[Figura 10 Configuração do servidor 25](#_Toc500283950)

[Figura 11 Configuração de rede 25](#_Toc500283951)

[Figura 12 Conexão da rede wi-fi 26](#_Toc500283952)

[Figura 13 Reconectar com servidor 26](#_Toc500283953)

[Figura 14 Medir dados 26](#_Toc500283954)

[Figura 15 Publicar dados no servidor 26](#_Toc500283955)

[Figura 16 Inicialização 27](#_Toc500283956)

[Figura 17 Execução principal 27](#_Toc500283957)

[Figura 18 Opção "is numeric" 29](#_Toc500283958)

[Figura 19 Tópico 30](#_Toc500283959)

[Figura 20 Dados temperatura 30](#_Toc500283960)

[Figura 21 Dados umidade 31](#_Toc500283961)

[Figura 22 Weather 31](#_Toc500283962)

**LISTA DE SÍMBOLOS**

Ω, ohm, unidade de medida a resistência elétrica.

V, volts, unidade de medida a diferença de potencial elétrico.

A, ampère, unidade de medida a corrente elétrica.

ºC, graus Celsius, unidade de medida a temperatura.

SUMÁRIO

[1. INTRODUÇÃO 11](#_Toc500277889)

[1.1. Objetivo Geral 11](#_Toc500277890)

[1.2. Objetivos Específicos 12](#_Toc500277891)

[1.3. Proposta Metodológica 12](#_Toc500277892)

[2. REVISÃO DA LITERATURA 13](#_Toc500277893)

[2.1. Tecnologias Utilizadas 13](#_Toc500277896)

[2.1.1. Linguagem ARDUINO 13](#_Toc500277899)

[2.1.2. NodeMcu ESP8266 14](#_Toc500277904)

[2.1.3. MQTT 15](#_Toc500277910)

[2.1.4. Sensor DHT11 16](#_Toc500277917)

[2.2. Soluções Existentes 17](#_Toc500277921)

[2.3. Levantamento de Requisitos 18](#_Toc500277922)

[2.3.1. Definição dos Stakeholders 18](#_Toc500277923)

[2.3.2. Metodologia Lean 18](#_Toc500277924)

[2.3.3. Requisitos Funcionais 19](#_Toc500277925)

[2.3.3.1. Requisito funcional 1 19](#_Toc500277926)

[2.3.3.2. Requisito funcional 2 19](#_Toc500277927)

[2.3.3.3. Requisito funcional 3 20](#_Toc500277928)

[2.3.3.4. Requisito funcional 4 20](#_Toc500277929)

[2.3.4 Requisitos Não Funcionais 20](#_Toc500277938)

[2.3.4.1. Requisito não funcional 1 20](#_Toc500277940)

[2.3.4.2. Requisito não funcional 2 20](#_Toc500277941)

[2.3.4.3. Requisito não funcional 3 20](#_Toc500277942)

[2.3.4.4. Requisito não funcional 4 20](#_Toc500277943)

[3. DESENVOLVIMENTO DO TRABALHO 21](#_Toc500277944)

[3.1. Modelo de Dados 21](#_Toc500277945)

[3.2. Arquitetura 21](#_Toc500277946)

[3.3. Websensor 23](#_Toc500277956)

[3.4. Instalação física do sistema 27](#_Toc500277963)

[3.5. Deploy 28](#_Toc500277964)

[4. RESULTADOS, DISCUSSÕES E CONCLUSÕES 29](#_Toc500277965)

[5. REFERÊNCIAS BIBLIOGRÁFICAS 30](#_Toc500277966)

[6. AGRADECIMENTOS 31](#_Toc500277967)

# INTRODUÇÃO

De acordo com Egídio Bega (2006), autor do livro Instrumentação Industrial, a instrumentação é definida como a ciência que estuda, desenvolve e aplica instrumentos de medição e controle de processos. A instrumentação é utilizada para se referir à área de trabalho dos técnicos e engenheiros que lidam com processos industriais (técnicos de operação, instrumentação, engenheiros de processamento, de controle e de automação, entre outros).

A empresa SCIA (2017) que existe no mercado de engenharia há mais de 40 anos afirma que as plantas industriaisnormalmente são estruturas altamente complexas projetadas em função dos principais processos, equipamentos e máquinas associados à sua função. Estas estruturas têm geometrias complexas e uma preocupação com segurança, como no caso de plantas de geração de energia e instalações que lidam com materiais perigosos (indústria química, etc.). Os engenheiros que atuam em projetos de plantas industriais precisam ser capazes de lidar com grandes estruturas integrando-as com seus softwares.

Segundo o orientador deste projeto, Leônidas Melo, na FATEC de São José dos Campos existe a iniciativa de um projeto onde uma planta industrial didática será desenvolvida e com ela deverá existir um software para monitorar os sensores atrelados na mesma. Tendo isto dito propõem-se que seja feito um sistema para receber as informações dos dados de seus sensores. O mesmo irá utilizar da interface comunicativa de modelo MQTT. As ligações entre os sensores e o programa serão feitas através da placa NodeMcu.

## Objetivo Geral

O objetivo geral deste projeto é apresentar a possibilidade de monitoramento e visualização de dados providos de sensores, como os de plantas industriais, utilizando a tecnologia IoT.

## Objetivos Específicos

Para a consecução deste objetivo foram estabelecidos os objetivos específicos:

* A placa deverá ser capaz de utilizar sensores gerais (sensor de exemplo: DHT11);
* Retornar valores tratados dos sensores e gráficos;
* Informações simples e objetivas.

## Proposta Metodológica

Para a conclusão deste trabalho será desenvolvido um programa que irá se comunicar com a placa NodeMcu. Este programa irá receber os dados coletados pelos sensores e os tratará para entendimento do usuário, como umidade, temperatura, etc. Os dados serão apresentados em uma tela simples para o usuário e um breve gráfico poderá ser consultado.

|  |
| --- |
| Figura - Layout.  D:\tg\Imagens\layout.png |

# REVISÃO DA LITERATURA

Neste capítulo serão apresentadas as tecnologias utilizadas na solução do problema, uma pesquisa mercadológica das soluções existentes e um levantamento de requisitos.



## Tecnologias Utilizadas

Esta seção apresenta as principais tecnologias utilizadas na solução proposta para o problema.



### Linguagem ARDUINO

A linguagem nativa das placas Arduino UNO e afins são baseadas em C/C++. É uma linguagem livre, sem necessidade de compra de sistema ou ambiente para seu funcionamento. Ele conecta-se com a biblioteca AVR, que por sua vez é um projeto de software livre cujo objetivo é fornecer uma biblioteca C de alta qualidade para uso em microprocessadores das placas Arduino e afins, permitindo o uso de qualquer de suas funções, facilitando assim o desenvolvimento do software que irá gerenciar os sensores conectados a placa NodeMcu.

Figura Ambiente Arduino





### NodeMcu ESP8266

Hardware escolhido para o desenvolvimento deste projeto por conter um módulo de conexão Wi-Fi embutido em sua placa, abrindo a possibilidade para tornar o sistema capaz de publicar informações online.

Figura Placa ESP8266





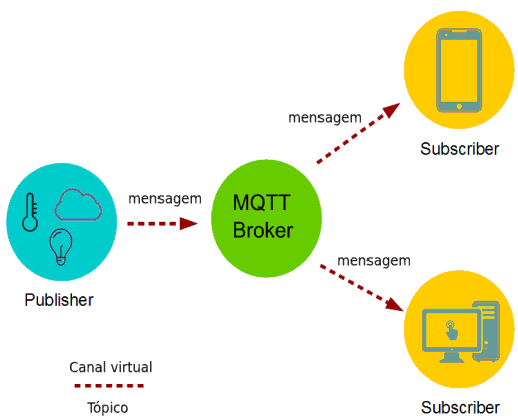
### MQTT

Conforme Francesco Azzola (2017), o protocolo de mensagens MQTT (Message Queue Telemetry Transport) está presente no dia a dia da Internet das Coisas (IoT) e o seu principal uso é fazer as máquinas conversarem, também conhecido como Machine-to-Machine (M2M).

É projetado para um baixo consumo de banda de rede e requisitos de hardware sendo extremamente simples e leve. O MQTT foi desenvolvido pela IBM e Eurotech e é projetado para enviar dados através de redes intermitentes ou com baixa banda de dados, para isto o protocolo é desenvolvido em cima de vários conceitos que garantem uma alta taxa de entrega das mensagens, baseado no TCP/IP e ambos, cliente e broker, que atua como intermediário na comunicação com o servidor, necessitam da pilha TCP/IP para o seu funcionamento. Utiliza o paradigma publish/subscribe (pub/sub) para a troca de mensagens.

Existem vários brokers MQTT disponíveis, pagos e gratuitos, neste projeto será utilizado um gratuito, disponibilizado pela organização Eclipse no endereço iot.eclipse.org:1883, sensorDornelas/temperatura e sensorDornelas/umidade.

Figura Layout MQTT



Fonte: adaptado de Francesco Azzola (2017)

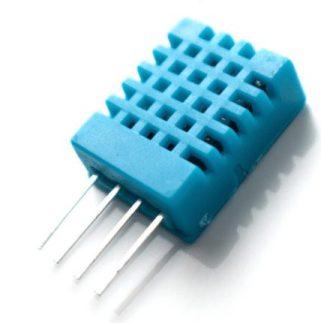


### Sensor DHT11

De acordo com Allan Mota (2017) DHT11, o sensor definido para cumprir o objetivo deste projeto, contém as seguintes características:

* Tensão de alimentação de 3V a 5V;
* 2.5mA de corrente máxima durante a conversão;
* Bom para medir umidade entre 20% e 80%, com aproximadamente 5% de precisão;
* Bom para medir temperaturas entre 0 e 50°C, com ±2°C de precisão;
* Taxa de amostragem de até 1Hz (1 leitura por segundo);
* Dimensões: 15.5mm x 12mm x 5.5mm;
* 4 pinos com 0.1″ de espaçamento entre eles;
* Muito baixo custo.

Figura Sensor DHT11





## Soluções Existentes

De acordo com a National Instruments (2017), o LabVIEW é uma linguagem de programação gráfica originária da National Instruments. Atualmente existem ambientes de desenvolvimento integrados para os Sistemas Operacionais Windows, Linux e Solaris. Os principais campos de aplicação do LabVIEW são a realização de medições e a automação. A programação é feita de acordo com o modelo de fluxo de dados, o que oferece a esta linguagem vantagens para a aquisição de dados e para a sua manipulação. Os programas em LabVIEW são chamados de instrumentos virtuais. São compostos pelo painel frontal, que contém a interface, e pelo diagrama de blocos, que contém o código gráfico do programa. O programa não é processado por um interpretador, mas sim compilado. Deste modo a sua performance é comparável à exibida pelas linguagens de programação de alto nível. O programador liga instrumentos virtuais com linhas (arames) de ligação e define, deste modo, o fluxo de dados. Cada instrumento virtual pode possuir entradas e/ou saídas. A execução de um instrumento virtual começa quando todas as entradas estão disponíveis; os resultados do processamento são então colocados nas saídas assim que a execução do subprograma tenha terminado. Desta forma, a ordem pela qual as tarefas são executadas é definida em função dos dados. Uma ordem pré-definida (por exemplo, "da esquerda para a direita") não existe. Uma importante consequência destas regras é a facilidade com que podem ser criados processos paralelos no LabVIEW. Os sub instrumentos virtuais sem interdependência dos respectivos dados são processados em paralelo. O painel frontal do LabVIEW é um meio confortável para construir programas com uma boa interface gráfica. O programador não necessita de escrever qualquer linha de código. A apresentação gráfica dos processos aumenta a facilidade de leitura e de utilização. Uma grande vantagem em relação às linguagens baseadas em texto é a facilidade com que se cria componentes que se executam paralelamente. Em projetos de grande dimensão é muito importante planejar a sua estrutura desde o início.

## Levantamento de Requisitos

Esta seção apresenta o levantamento de requisitos da solução proposta para o desenvolvimento.

## Definição dos Stakeholders

* Desenvolvedor: Gabriel Dornelas, estudante de Análise e Desenvolvimento de Sistemas - FATEC SJC.
* Orientador: Leônidas Melo, professor em FATEC SJC.
* Especialista consultado: Diogo Branquinho, dono da empresa TecSus.
* Usuário final: professor Leônidas Melo entre outros professores da FATEC SJC que irão utilizar sensores e precisem de monitoramento de seus dados.

## Metodologia Lean

De acordo com a equipe Runrun.it (2017) termo lean deve ser entendido como “enxuto”. Ou seja, trata-se de um método que institui o uso de nada além do que os recursos necessários para a realização de um determinado trabalho, etapa ou processo, evitando desperdícios.

Um dos grandes méritos da cultura lean é ajudar a colocar novos produtos no mercado. O método está apoiado em três importantes pilares:

Enxugue o modelo de negócio com o Canvas **→** Inicialmente não há nada além de hipóteses que precisam ser comprovadas. Assim sendo, em vez de consolidar um longo relatório de plano de negócios, utiliza-se uma ferramenta chamada Canvas para montar o seu business model, trata-se de um diagrama que mostra como se cria valor para os stakeholders.

Testar as possibilidades com o Customer Development **→** Após estruturar com o Canvas, serão testadas as hipóteses com uma abordagem chamada de “desenvolvimento com clientes”, ou customer development. Informações com os stakeholders foram trocadas e suas opiniões sobre todo e qualquer elemento do modelo de negócios coletadas, definindo assim as características do projeto.

Adotar um desenvolvimento ágil **→** Prática que acompanha o desenvolvimento com o usuário final. No desenvolvimento ágil, não há perda de tempo ou de recursos, pois o produto é desenvolvido de forma iterativa e incremental até seu desfecho.

Figura Método Lean



fonte: adaptado de equipe Runrun.it (2017)

## Requisitos Funcionais

Este tópico tem como objetivo apresentar todos os requisitos funcionais levantados nas reuniões de encontro com o usuário final do sistema, o cliente do sistema.

## Requisito funcional 1

O sistema deve apresentar os dados providos de sensores, no caso, de temperatura e umidade.

## Requisito funcional 2

O software deve funcionar estritamente de modo online.

## Requisito funcional 3

O programa deve apresentar um gráfico simples dos valores apresentados.

## Requisito funcional 4

Um relatório dos dados transmitidos pode ser salvo para possíveis análises.



### Requisitos Não Funcionais

Este tópico tem como objetivo apresentar todos os requisitos não funcionais levantados pelo desenvolvedor nas reuniões de encontro com o cliente do sistema.



## Requisito não funcional 1

O software deverá usar o sistema MQTT para disponibilizar os dados online, por ele ter uma ampla capacidade de apresentação de dados.

## Requisito não funcional 2

O sistema deve ser desenvolvido em linguagem Arduino para que a comunicação entre a placa de microcontrolador, computador e servidor seja compreendida.

## Requisito não funcional 3

O programa deverá operar com sensor de temperatura DHT11 ou DHT22 para que a leitura dos dados seja realizada corretamente.

## Requisito não funcional 4

O sensor de temperatura deve utilizar um resistor de 4,7KΩ para prevenção de cargas elétricas.

# DESENVOLVIMENTO DO TRABALHO

Este capítulo apresenta o processo de desenvolvimento da solução proposta.

## Modelo de Dados

O sistema trabalha com o monitoramento e visualização dos dados obtidos pelo sensor de temperatura e umidade. Isso demonstra que se o cliente não precisar, não existirá necessidade de armazenamento e persistência das informações coletadas pelo sensor.

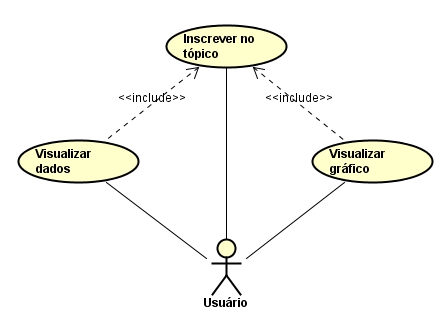
Os dados são mantidos apenas enquanto o software estiver em execução para a demonstração visual do gráfico. Demonstrando que o software não conterá nenhum tipo de dicionário de dados.

## Arquitetura

Este capitulo visa apresentar diagramas de caso e uso e fluxograma.

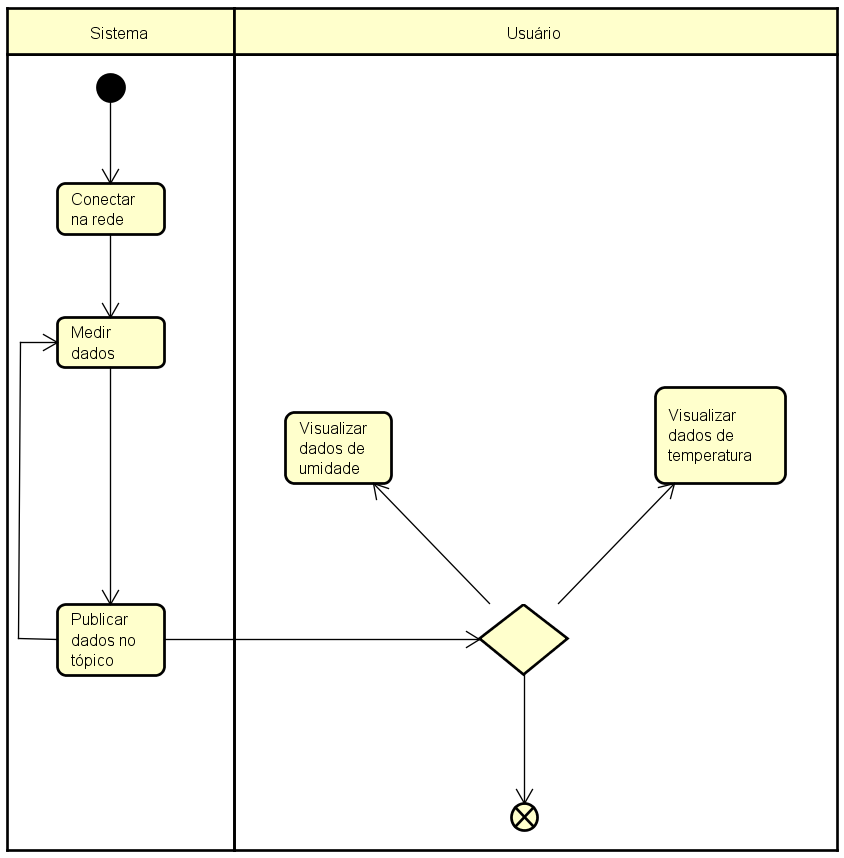
* + 1. **Diagrama de caso e uso**

Figura Diagrama de caso e uso



* + 1. **Diagrama de Fluxo**

Figura Diagrama de Fluxo





## Websensor

Este capitulo demonstrará os códigos desenvolvidos para as funções do sistema e sua utilização, assim como a maneira como eles são disponibilizados para o usuário e o formato em que são recebidos e formatados para visualização.

#### Configuração

Antes de se iniciar o sistema deve-se configurar as seguintes definições:

Incluí-se a biblioteca para uso do sensor de temperatura/umidade, a definição do pino a ser utilizado na placa NodeMcu, realiza-se a definição do modelo do sensor a ser utilizado, é feita a função para utilizar o sensor que utiliza a variável onde será armazenado dado de umidade e a variável onde será armazenado dado de temperatura. A biblioteca é incluída para uso do servidor MQTT, definido o endereço do servidor e o nome do cliente para acessar o servidor para criar o servidor e nomear os tópicos de temperatura e umidade. Para uso do Wi-Fi é incluída a biblioteca, onde é definido o nome da rede e a senha que serão utilizadas e o objeto de conexão é criado para ser utilizado previamente.

Figura Configuração do sensor

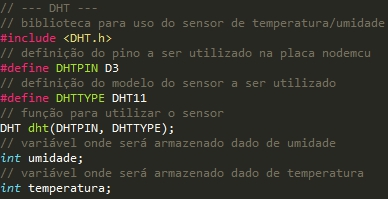


Figura Configuração do servidor

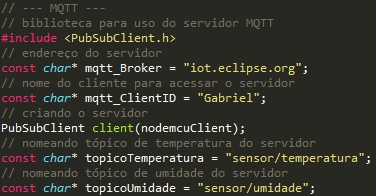
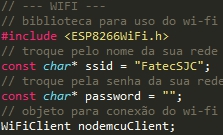


Figura Configuração de rede



#### Funções

Nesta sessão serão apresentadas as funções que o software irá utilizar:

Função para estabelecer a conexão do microcontrolador com a rede Wi-Fi onde é fornecido o nome e a senha definida anteriormente;

Função utilizada para certificar que o cliente está conectado com o servidor;

Função que mede os dados do sensor e os armazena para serem publicados no servidor;

Função que disponibiliza os dados lidos pelo sensor no servidor.

Figura Conexão da rede wi-fi

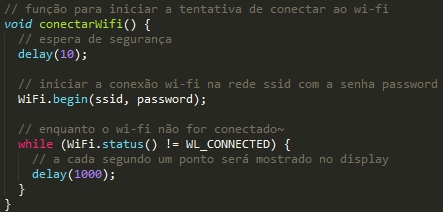


Figura Reconectar com servidor

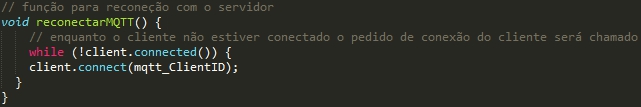


Figura Medir dados

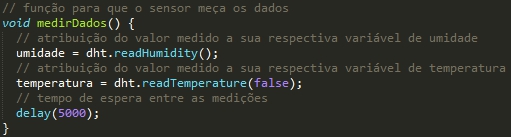
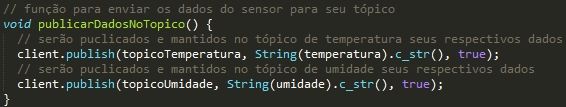


Figura Publicar dados no servidor



#### Inicialização e execução

Esta sessão irá apresentar o primeiro script a ser executado e a atividade principal do sistema:

Inicializando o sistema, é feita a conexão com a rede Wi-Fi e depois a comunicação com o servidor é estabelecida, o sistema entra em funcionamento, sempre que a comunicação com o servidor não estiver estabelecida o software irá tentar se reconectar, caso a conexão esteja estabelecida os dados serão medidos pelo sensor e postados no tópico do servidor.

Figura Inicialização

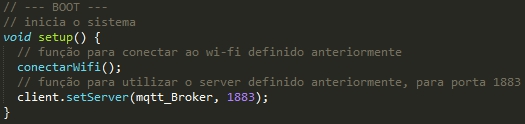
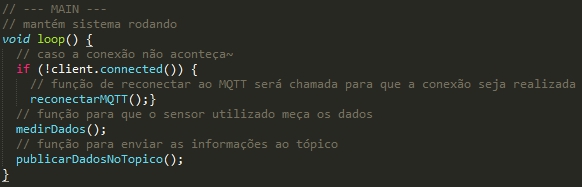


Figura Execução principal





## Instalação física do sistema

Primeiramente os materiais são separados:

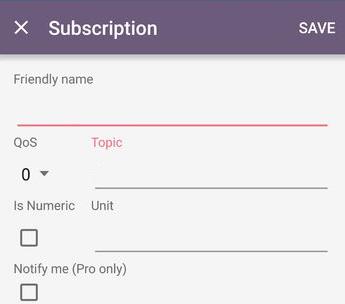
* NodeMcu ESP8266;
* Sensor DHT11 ou 22;
* Resistor 4,7KΩ;
* Cabo de comunicação mini USB;
* Cabos diversos para ligações;
* Placa protoboard.

O passo a passo a ser apresentado deve seguir a organização da figura 1 layout mostrada neste documento. O microcontrolador e o sensor devem ser colocados no protoboard de maneira a haver espaço entre eles para que seja possível um encaixe livre do resistor e os cabos de ligação. Após o encaixe do microcontrolador e do sensor, o resistor deve ser conectado nas hastes 1 e 2 do sensor. A haste 1 do sensor deve ser conectada ao pino 3V3 da placa NodeMcu, a haste 2 deve ser conectada ao pino D3 da placa NodeMcu, a haste 3 não será utilizada e a haste 4 deve ser ligada ao pino terra da placa NodeMcu. A placa NodeMcu se receberá as suas instruções (códigos) com a interface Arduino pelo cabo de comunicação mini USB que será conectado a uma porta livre do computador responsável por transferir a estrutura do software para o microcontrolador. Assim que o procedimento de compilação e transferência de instruções terminar a placa poderá ser alimentada com qualquer tipo de fonte mini USB de especificação 5V e 1A. O hardware deverá ficar sempre conectado a uma rede de internet para que ele publique os dados no servidor e os clientes possam visualizar e salvar essas informações.

## Deploy

Em acordo com o objetivo apresentado neste documento de monitorar e visualizar dados de sensores de maneira inteligente, o sistema, para ser executado precisará das configurações de uma rede Wi-Fi disponível para a publicação no tópico dos respectivos dados medidos pelo sensor. O cliente ao se inscrever no tópico disponibilizado no Server iot.eclipse.org, porta 1883, tópicos sensorDornelas/temperatura e sensorDornelas/umidade, utilizando o aplicativo IoT MQTT Dashboard disponível para download na Play Store e App Store, poderá marcar a opção de “Is numeric” para visualizar os dados transmitidos pelo sensor em forma gráfica.

Figura Opção "is numeric"



Caso o cliente deseje salvar os dados de um tópico para realizar uma análise ou mesmo manter um histórico, foi desenvolvido um programa em conjunto com o projeto que salva um relatório dos dados coletados no processo de leitura. Basta executar o programa enquanto desejar salvar os dados que o sensor esta enviando para o tópico, ele se manterá em funcionamento, salvando assim os dados em um arquivo de texto.

# RESULTADOS, DISCUSSÕES e conclusões

De acordo com a solução apresentada neste projeto, obtivemos os seguintes resultados de dados de medição de temperatura e umidade:

Figura Tópico

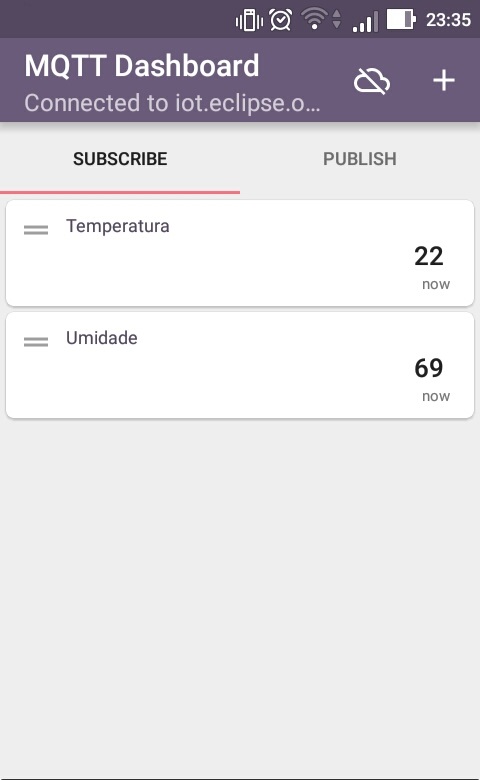


Figura Dados temperatura

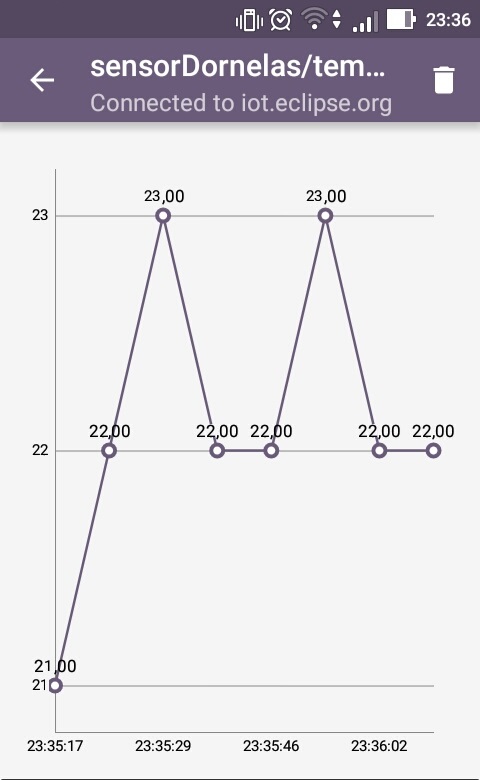
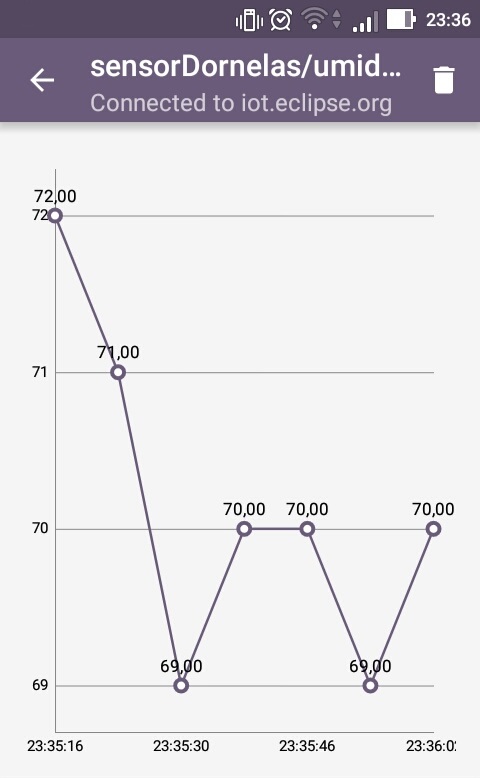


Figura Dados umidade



De acordo com a meteorologia oficial da empresa especializada weather.com dos respectivos dias eram:

Figura Weather



Comparando os valores apresentados pelo sistema com a meteorologia oficial pode-se concluir que o projeto responde realmente ao que foi proposto, apresentando medições próximas ao dado real (obedecendo as capacidades do sensor DHT11), provendo assim uma monitoria eficaz dos dados apresentados, cumprindo os requisitos descritos neste documento.

Comparando superficialmente a solução proposta fornecida por este projeto com as outras soluções existentes citadas neste documento é possível afirmar que elas poderiam cumprir todos requisitos apresentados, mas exigem um custo monetário muito elevado mostrada a simplicidade do problema.

Pode-se dizer que a vantagem do sistema de uma das soluções existentes sejam as análises inteligentes e automáticas, fora a precisão com baixa taxa de incerteza dos dados. Já sua desvantagem seria que pequenas mudanças nas configurações ocasionariam grandes mudanças no software, podendo acarretar a necessidade de reestruturações além do alto custo monetário de investimento para o monitoramento de uma variável provida de um sensor, considerando que a solução apresentada utiliza de um investimento de baixo custo para sua fabricação e instalação.

# REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINING. **ARDUINING - DIAGRAMS.**Disponível em: <https://arduining.com/testpage/>. Acesso em: 05 dez. 2017.

EMBARCADOS. **MQTT - Protocolos para IoT.**Disponível em: <https://www.embarcados.com.br/mqtt-protocolos-para-iot/>. Acesso em: 05 dez. 2017

ENTREPRENEUR. **How to Start a Freight Brokerage Business.**Disponível em: <https://www.entrepreneur.com/article/37928>. Acesso em: 05 dez. 2017.

<https://www.survivingwithandroid.com/2016/10/mqtt-protocol-tutorial.html>

WEATHER. **The Weather channel.**Disponível em: <https://weather.com/pt-BR/clima/hoje/l/46c941c6d8cd0500493ccebd6e836af6fdac9cce9bc0a6640893b3ccdaef0e59>. Acesso em: 05 dez. 2017.

MOTA, Allan. **Sensor DHT11 e DHT22 e o Uso de Bibliotecas.**Disponível em: <https://portal.vidadesilicio.com.br/sensores-dht11-dht22-biblioteca-arduino/>. Acesso em: 05 dez. 2017.

INSTRUMENTS, National. **LabVIEW.**Disponível em: <http://www.ni.com/pt-br/shop/labview.html>. Acesso em: 5 dez. 2017.

# AGRADECIMENTOS

Agradeço ao meu orientador pelo apoio e disponibilização de materiais para iniciação do projeto, aos demais professores e profissionais consultados, pelos conhecimentos transmitidos, aos meus familiares e namorada por me manterem encorajado para a criação deste sistema e também aos meus colegas de classe por me auxiliarem nas correções da documentação e codificação do projeto.