



**Ministério  
da Educação**

Secretaria de Educação Profissional e Tecnológica  
Instituto Federal de Educação, Ciência e Tecnologia de  
Santa Catarina

**CAMPUS FLORIANÓPOLIS**

**DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA**

**BACHARELADO EM ENGENHARIA MECATRÔNICA**

**CRISTIAN FERNANDO RITTER**

**MEDIDOR DE ENERGIA MONOFÁSICO DE BAIXA TENSÃO  
BASEADO EM REDES SEM FIO**

**FLORIANÓPOLIS, OUTUBRO DE 2020**



**INSTITUTO FEDERAL  
SANTA CATARINA**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA**

**CAMPUS FLORIANÓPOLIS**

**DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA**

**BACHARELADO EM ENGENHARIA MECATRÔNICA**

**CRISTIAN FERNANDO RITTER**

**MEDIDOR DE ENERGIA MONOFÁSICO DE BAIXA  
TENSÃO BASEADO EM REDES SEM FIO**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso, do curso de Engenharia Mecatrônica, do Departamento Acadêmico de Metal-Mecânica, do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Engenharia Mecatrônica.

Orientador Prof. Valdir Noll, Dr. Eng.

**FLORIANÓPOLIS, OUTUBRO DE 2020**

Ficha de identificação da obra elaborada pelo autor.

Ritter, Cristian Fernando

**Medidor de Energia Monofásico de Baixa Tensão Baseado em Redes Sem Fio / Cristian Fernando Ritter ;** orientação de Valdir Noll.

- Florianópolis, SC, 2020

86 p.

**Trabalho de Conclusão de Curso (TCC)** - Instituto Federal de Santa Catarina, Campus Florianópolis. Bacharelado em Engenharia Mecatrônica, Departamento Acadêmico de Metal Mecânica.

Inclui Referências.

1. Internet das Coisas. 2. Monitor de Energia. 3. Microcontroladores. I. Noll, Valdir. II. Instituto Federal de Santa Catarina. Departamento Acadêmico de Metal Mecânica. III. Título

**MEDIDOR DE ENERGIA MONOFÁSICO DE BAIXA TENSÃO BASEADO EM REDES SEM FIO**

**CRISTIAN FERNANDO RITTER**

Este trabalho foi julgado adequado para obtenção de título de Bacharel em Engenharia Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Bacharelado em Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 16 de outubro de 2020.

Banca examinadora:

---

Prof. Valdir Noll, Dr.

Orientador

---

Prof. Gregory Chagas da Costa Gomes

Membro Titular

---

Prof. Mario de Noronha Neto, Dr.

Membro Titular



INSTITUTO FEDERAL  
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS

## DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o estudante CRISTIAN FERNANDO RITTER, matrícula nº 1510047271, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado *MEDIDOR DE ENERGIA MONOFÁSICO DE BAIXA TENSÃO BASEADO EM REDES SEM FIO*, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 16 de outubro de 2020.



Prof. Orientador do TCC: Valdir Noll

## RESUMO

RITTER, Cristian Fernando. MEDIDOR DE ENERGIA MONOFÁSICO DE BAIXA TENSÃO BASEADO EM REDES SEM FIO. 85 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Engenharia Mecatrônica, Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Florianópolis, 2020.

Neste trabalho propõe-se o desenvolvimento de um sistema para medição de energia monofásico de baixa tensão baseado em redes de comunicação de dados sem fio que realizará o monitoramento de diversos parâmetros elétricos da rede e seu registro em um serviço de banco de dados. O sistema será desenvolvido em um microcontrolador que enviará os dados para um serviço de Internet das Coisas. Uma interface web permitirá a filtragem e consulta dos dados de qualquer lugar por meio de um acesso à internet. O banco de informações adquirido poderá ser usado para inúmeros fins, que incluem: traçar um padrão de consumo do local, definir estatisticamente a estimativa do consumo, verificar informações relacionadas à conformidade do fornecimento de energia elétrica e possíveis problemas na instalação elétrica, como desperdício de energia ou problemas no coeficiente de fator de potência. Além disso, este sistema pode colaborar na realização de mudanças nos hábitos de consumo que permitam aprimorar a eficiência energética do ambiente monitorado.

**Palavras chave:** Internet das Coisas, monitor de energia, microcontroladores, sensores de corrente e tensão.

## ABSTRACT

RITTER, Cristian Fernando. LOW-VOLTAGE SINGLE-PHASE POWER METER BASED ON WIRELESS NETWORKS. 85 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Engenharia Mecatrônica, Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Florianópolis, 2020.

This work proposes the development of a measuring single-phase low voltage energy system based on a wireless network that will monitor several electrical grid parameters and record them in a database service. This system will be developed in a microcontroller that will use Internet of Things data. Web interface will allow filtering and querying data from anywhere that has internet access. Collected database can be used for a number of purposes, which include: drawing a consumption pattern, prediction, information related to the conformity of the electricity and to investigate electrical installation troubles, such as waste of electricity and problems with the power factor coefficient. In addition, this system can collaborate to make changes in consumption habits allowing energy efficiency improvements.

**Keywords:** Internet of Things, energy monitor, microcontrollers, current and voltage sensors.

## LISTA DE FIGURAS

Figura 1 - Esboço geral do sistema.....	21
Figura 2 - Shunts resistivos. Fonte: AMDS4 (2020) .....	23
Figura 3 - Transdutores de Efeito Hall. Fonte: AMDS4 (2020) .....	24
Figura 4 - Transformadores de Corrente. Fonte: AMDS4 (2020) .....	24
Figura 5 - Bobinas de Rogowski. Fonte: AMDS4 (2020) .....	25
Figura 6 - Circuito integrado ADE7753. Fonte: Analog Devices (2010) .....	26
Figura 7 - Módulo Node MCU com ESP8266. Fonte: FilipeFlop (2020).....	26
Figura 8 - Funcionamento do protocolo MQTT. Fonte: Ayush Gemini (2019) .....	27
Figura 9 - Exemplo de pacote de dados JSON .....	28
Figura 10 – Publicação de mensagem no tópico.....	29
Figura 11 - Exemplo de tabela do Dynamo DB .....	30
Figura 12 - Protótipo em placa de montagem .....	33
Figura 13 – Tela de autenticação de usuários na aplicação web .....	34
Figura 14 – Demonstrativo de consulta de dados na aplicação web.....	34
Figura 15 - Diagrama do circuito elétrico.....	36
Figura 16 - Transformador de Corrente SCT-013-020. Fonte: Nice gear (2020).....	38
Figura 17 - Circuito de alimentação e canal 2 de Tensão .....	38
Figura 18 – Diagrama de utilização de transformadores de corrente. Fonte: Analog Devices (2010).....	40
Figura 19 - Configuração de entrada do Canal 1 de Corrente .....	41
Figura 20 - Display <i>OLED</i> SSD1306. Fonte: ETT (2020) .....	42
Figura 21 - Conexão do display <i>OLED</i> ao microcontrolador .....	42
Figura 22 - Informações do display .....	43
Figura 23 – Projeto de invólucro para o dispositivo .....	43
Figura 24 – Máscara de componentes da placa de circuito impresso.....	44
Figura 25 – Máscara inferior da placa de circuito impresso. ....	44
Figura 26 - Vista 3D da placa .....	45
Figura 27 - Fluxograma do funcionamento do software .....	46
Figura 28 - Descrição da programação do microcontrolador .....	48
Figura 29 - Configuração da rede <i>wi-fi</i> .....	48
Figura 30 - Configuração de identificação de dispositivo .....	48



Figura 31 - Configuração de periodicidade .....	49
Figura 32 - Configuração de certificados.....	49
Figura 33 - Configuração dos registradores do ADE7753.....	49
Figura 34 - Método de escrita nos registradores .....	50
Figura 35 - Método de leitura dos registradores.....	51
Figura 36 - Método de cálculo de média .....	51
Figura 37 - Exemplo de método de retorno de valor .....	52
Figura 38 - Inclusão da AWS SDK no projeto .....	53
Figura 39 - Configuração de região e credenciais no código PHP .....	53
Figura 40 – Solicitação de busca de dados em PHP .....	54
Figura 41 - Estrutura de arquivos da aplicação web .....	54
Figura 42 - Dados analisados.....	56
Figura 43 - Médias e estimativas de consumo .....	57

Todas as figuras que não possuem fontes informadas são de autoria própria.

## LISTA DE EQUAÇÕES

Equação 1 - Potência aparente .....	32
Equação 2 - Fator de potência .....	32
Equação 3 - Potência ativa.....	32
Equação 4 - Potência reativa .....	32
Equação 5 - Equação tensão de pico.....	39
Equação 6 - Calculo da tensão de pico no canal 2 .....	39
Equação 7 - Calculo da tensão diferencial no canal 2.....	39
Equação 8 - Cálculo do fator de multiplicação para calibração .....	40
Equação 9 - Cálculo de valor de corrente no canal 1 .....	41

## LISTA DE SIGLAS

ADC	<i>Analogic to digital converter (conversor analógico-digital)</i>
ABESCO	Associação Brasileira das Empresas de Serviços de Conservação de Energia
AC	<i>Alternate Current (corrente alternada)</i>
AGND	<i>Analogic Ground (ground do circuito analógico)</i>
AP	<i>Access Point (ponto de acesso à rede sem fio)</i>
AVDD	Tensão de alimentação positiva do circuito analógico
AWS	<i>Amazon Web Services</i>
CSS	<i>Cascading Style Sheets</i>
CT	<i>Current Transformer (transformador de corrente)</i>
DC	<i>Direct Current (corrente continua)</i>
DGND	<i>Digital Ground (ground do circuito digital)</i>
DVDD	Tensão de alimentação positiva do circuito digital
EC2	<i>Elastic Compute Cloud</i>
GND	<i>Ground</i>
GPIO	<i>General Purpose Input/Output</i>
HTML	<i>Hyper Text Markup Language</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IOT	<i>Internet of things (internet das coisas)</i>
JSON	<i>Java Script Object Notation</i>
LSB	<i>Least Significant Bit (bit menos significativo)</i>
MQTT	<i>MQ Telemetry Transport</i>
MSB	<i>Most Significant Bit (bit mais significativo)</i>
PHP	<i>Hypertext Preprocessor</i>
PWM	<i>Pulse-Width Modulation</i>
RMS	<i>Root mean square</i>

SLA	<i>Service Level Agreement (termos do acordo de serviço)</i>
SPI	<i>Serial Peripheral Interface</i>
TCP/IP	<i>Internet protocol suite</i>
UTC	<i>Coordinated Universal Time</i>
UTS	<i>Unix Time Stamp</i>

# SUMÁRIO

1	INTRODUÇÃO .....	16
1.1	OBJETIVOS .....	17
1.1.1	OBJETIVO GERAL .....	17
1.1.2	OBJETIVOS ESPECIFICOS .....	17
1.2	JUSTIFICATIVA .....	19
1.3	METODOLOGIA .....	20
1.3.1	ESBOÇO GERAL DO SISTEMA .....	20
1.4	ESTRUTURA DO DOCUMENTO .....	21
2	FUNDAMENTAÇÃO TEÓRICA.....	23
2.1	SENSORES DE CORRENTE E TENSÃO .....	23
2.1.1	SHUNTS RESISTIVOS .....	23
2.1.2	TRANSDUTORES DE EFEITO HALL .....	23
2.1.3	TRANSFORMADOR DE CORRENTE.....	24
2.1.4	BOBINA DE ROGOWSKI .....	24
2.1.5	ISOLAMENTO GALVÂNICO .....	25
2.1.6	TRANFORMADOR E DIVISOR DE TENSÃO .....	25
2.2	CIRCUITO INTEGRADO ADE 7753 .....	25
2.3	MÓDULO NODE MCU .....	26
2.4	PROTOCOLOS DE COMUNICAÇÃO E NOTAÇÃO.....	27
2.4.1	A INTERNET DAS COISAS E O PROTOCOLO MQTT .....	27
2.4.2	SERIAL PERIPHERAL INTERFACE (SPI).....	27
2.4.3	NOTAÇÃO JSON .....	28
2.5	AMAZON WEB SERVICES.....	28
2.5.1	AWS IOT CORE .....	29
2.5.2	DYNAMO DB.....	29

2.5.3	AMAZON EC2 .....	30
2.5.4	PACOTE XAMPP .....	31
2.6	GRANDEZAS ELÉTRICAS .....	31
2.6.1	POTÊNCIA APARENTE .....	31
2.6.2	FATOR DE POTÊNCIA .....	32
2.6.3	POTENCIA ATIVA .....	32
2.6.4	POTÊNCIA REATIVA .....	32
3	DESENVOLVIMENTO .....	33
3.1	ELABORAÇÃO DO PROTÓTIPO .....	33
3.2	DESENVOLVIMENTO DA APLICAÇÃO WEB .....	33
3.3	ESCOLHA DOS SERVIÇOS E COMPONENTES .....	34
3.4	CIRCUITO ELETRÔNICO .....	36
3.4.1	MEDIÇÃO DE ENERGIA COM O ADE7753 .....	37
3.4.2	TRANSFORMADOR DE CORRENTE .....	38
3.4.3	FONTE DE ALIMENTAÇÃO .....	38
3.4.4	CANAL 2 – SONDA DE TENSÃO .....	39
3.4.5	MEDIÇÃO DE CORRENTE .....	40
3.4.6	DISPLAY OLED .....	42
3.5	DESIGN DE PCI E INVÓLUCRO DE POLÍMERO .....	43
3.6	SOFTWARES DESENVOLVIDOS .....	45
3.6.1	CIRCUITO INTEGRADO ADE 7753 .....	46
3.6.2	MÓDULO <i>NodeMCU</i> .....	47
3.6.3	INTERFACE WEB .....	52
4	RESULTADOS ALCANÇADOS .....	55
4.1	INTEGRAÇÃO .....	55
4.2	INTERFACE WEB .....	55

4.3	COMPUTAÇÃO NA NUVEM.....	55
4.4	MEDIÇÕES REALIZADAS.....	55
5	CONCLUSÕES.....	58
5.1	DIFICULDADES ENCONTRADAS.....	60
6	REFERÊNCIAS.....	61
	APÊNDICE A – CÓDIGO FONTE.....	64
	APÊNDICE B – CONFIGURAÇÃO DOS SERVIÇOS AWS.....	64
	Conta - Billing.....	65
	Permissões - Policy.....	65
	Dispositivos - Things.....	67
	DynamoDB - Configurando Ações.....	72
	Instância E2C - Criação.....	78
	Instância E2C - Acesso remoto.....	79
	Instância E2C - Instalação de serviços.....	82
	Instância E2C - Servidor web.....	84

# 1 INTRODUÇÃO

Concorrentemente à popularização da internet e sua disponibilização sobre diversas formas de propagação, muitas delas sem a necessidade de fios, houve também a ampliação da gama de dispositivos com dimensões reduzidas e baixo consumo que podem acessar tais redes e compartilhar dados de sensoriamento e até mesmo de controle para sistemas nas mais diversas áreas, com uma ótima relação de custo benefício. Isso tem incentivado uma cada vez mais evidente tendência de interconexão e monitoração de dispositivos e ambientes, não somente industriais, mas também domésticos.

Diante da necessidade de coleta e análise de grandes quantidades de dados, surgiram diversas empresas que oferecem serviços de tratamento e armazenamento de dados em servidores remotos. Tais serviços dispensam a necessidade de quaisquer servidores ou infraestrutura hospedada localmente para manter um serviço disponível online.

Dispondo do recente amadurecimento das tecnologias citadas acima, propõe-se neste trabalho o desenvolvimento de um sistema para medição de energia monofásica de baixa tensão baseado em rede sem fio que realizará o monitoramento e registro de diversos parâmetros elétricos da rede. A medição dos parâmetros elétricos da rede será feita utilizando um circuito integrado especificamente desenvolvido para este fim. Utilizar-se-á um microcontrolador com interface de rede sem fio integrada para leitura, processamento e envio de informações que serão armazenados em um serviço de armazenamento de dados online. As análises e os resultados obtidos poderão ser consultados diretamente pela internet em uma aplicação web desenvolvida para este fim.

Os dados obtidos poderão ser utilizados para diversos fins, como por exemplo, definir estatisticamente a previsibilidade do consumo, verificar informações relacionadas à conformidade do fornecimento de energia elétrica e possíveis mudanças nos hábitos de consumo que permitam aprimorar a eficiência energética do ambiente monitorado.



## **1.1 OBJETIVOS**

### **1.1.1 OBJETIVO GERAL**

Desenvolvimento de um sistema que permita monitorar a qualidade de fornecimento e o consumo de energia elétrica de uma residência ou outro ambiente. O sistema permitirá a visualização dos dados em uma interface com gráficos de consumo em retrospectão, além de análises baseadas nos hábitos de consumo registrados.

Os principais requisitos de projeto são:

- O hardware de aquisição dos dados deve ser compacto.
- O produto deve oferecer segurança ao usuário e fácil instalação.
- O custo do produto final deve ser um dos itens a ser considerado na escolha dos componentes eletrônicos.
- O acesso do usuário à interface de consulta dos dados e resultados deve ser feita de maneira simples e prática por meio de uma interface de aplicativo ou página da web na internet.
- O dispositivo de medição deve ser capaz de enviar os dados para o banco de dados por meio de redes sem fio.
- O acesso aos dados deve estar protegido de acessos indevidos por meio de autenticação de usuário e senha ou outra tecnologia similar.

### **1.1.2 OBJETIVOS ESPECIFICOS**

Para facilitar a organização deste trabalho e tentar garantir que todos os requisitos citados no objetivo principal sejam atingidos com sucesso, foram elaborados objetivos específicos que servirão como metas sequenciais a serem alcançadas durante o desenvolvimento. Como itens de uma lista inicial de objetivos específicos podem ser alistados:

- Descrição em esboço de funcionamento do sistema.
- Análise dos fluxos de dados e de viabilidade técnica e econômica.
- Pesquisa e definição das tecnologias e componentes a serem utilizadas.
- Estudo e descrição de todas as tecnologias e componentes envolvidos no projeto.
- Elaboração dos diagramas elétricos e eletrônicos.

- Programação dos dispositivos eletrônicos e das plataformas de *back-end*.
- Execução de testes e correções.
- Programação das plataformas de *front-end*.
- Montagem de protótipo.
- Testes, correções e aprimoramentos finais.
- Apresentação dos resultados.

## 1.2 JUSTIFICATIVA

A utilização racional de energia é um ponto de pauta importante à medida que o mundo procura pensar e agir de modo sustentável. Conforme relatório divulgado pela Associação Brasileira das Empresas de Serviços de Conservação de Energia ([ABESCO](#)) e noticiado pelo jornal O Estadão, o desperdício de energia custou mais de R\$60 bilhões para a economia brasileira nos anos de 2015 a 2017. Além do prejuízo econômico, o uso indiscriminado da energia aumenta a geração de gases nocivos para a atmosfera, sendo responsável por uma parcela no efeito.

No Brasil, a maior parte da geração de energia elétrica provém de usinas hidrelétricas, ficando, portanto suscetíveis a mudanças climáticas e a períodos de estiagens, bem como afeta o ambiente quando se necessita ampliar a produção, pois exigem grandes áreas territoriais. Diante da ocorrência de longos períodos sem chuvas, outras formas de geração de energia podem ser utilizadas, porém muitas delas geram maior poluição e possuem maior custo por kWh gerado.

Para tentar analisar e corrigir os hábitos de consumo precisa-se mensurar adequadamente o consumo de energia de modo adequado. Neste aspecto, o medidor de energia elétrica da concessionária de energia não se mostra eficiente, pois não permite a visualização remota nem a criação de um banco de dados retrospectivo de consumo, além de não diferenciar os dados de cada ambiente.

A proposta deste trabalho é desenvolver um sistema capaz de facilitar a medição da energia elétrica em residências ou ambientes similares, permitindo que sejam coletados dados zoneados do consumo. Este sistema será capaz de informar quanto cada equipamento está consumindo, e em quais horários.

Com um mapa de consumo completo do ambiente podem ser tomadas medidas assertivas com o objetivo de substituir equipamentos de baixa eficiência energética e identificar gargalos de consumo. Ampliando-se a utilização de um sistema como esse a nível nacional, certamente poderiam ser minimizados o desperdício de energia e a necessidade de constantes investimentos em infraestrutura de geração e distribuição de energia, além de colaborar para um melhor uso dos recursos naturais.

## **1.3 METODOLOGIA**

Conforme descrito nos objetivos, este projeto é um dispositivo de ‘Internet das Coisas’ e utiliza plataformas de serviços disponibilizados online para evitar a necessidade de hospedar servidores localmente. O sistema é formado por diversos serviços trabalhando em conjunto, sendo cada um responsável por tarefas específicas.

Seu desenvolvimento tem caráter experimental, contemplando o projeto de um protótipo de dispositivo para medição e a programação dos firmwares e códigos fontes das aplicações necessárias. Além disso, será necessária a configuração de diversos serviços para a integração dos componentes em nuvem computacional. O estudo necessário para o desenvolvimento de tais atividades será realizado com base na documentação disponibilizada pelos serviços web, documentações de bibliotecas, bem como também, na informação presente nos manuais de componentes e em experiências de autores que desenvolveram trabalhos semelhantes anteriormente.

Com a finalidade de facilitar o entendimento ao expor a integração esperada entre os sistemas que serão desenvolvidos, será apresentado um esboço geral do sistema proposto.

### **1.3.1 ESBOÇO GERAL DO SISTEMA**

A Figura 1 ilustra o relacionamento entre o conjunto dos serviços de computação em nuvem, dos dispositivos de aquisição de dados e da interface com o usuário.

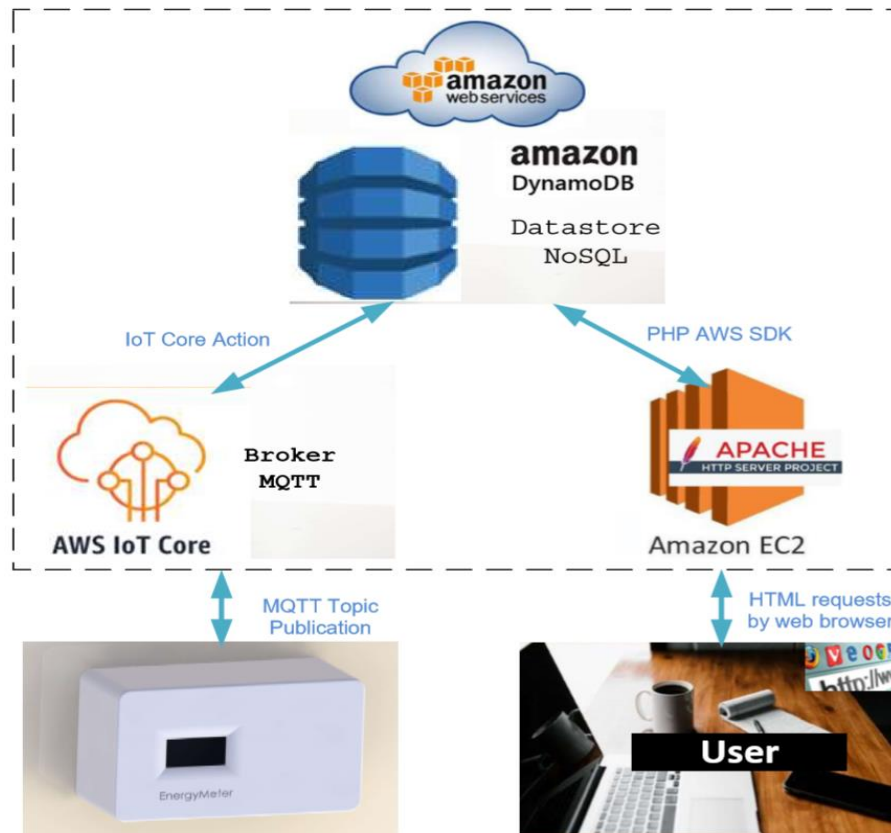


Figura 1 - Esboço geral do sistema.

Será produzido um protótipo de dispositivo eletrônico medidor de energia chamado de *Energy Meter*, que executará o envio das informações adquiridas para um *broker* que interpretará os dados e armazenará em um banco de dados *online*. Estes dados podem ser consultados por meio de requisições realizadas de uma página *web* especificamente desenvolvida para este fim que será disponibilizada por um serviço de hospedagem.

## 1.4 ESTRUTURA DO DOCUMENTO

A documentação deste projeto está dividida em capítulos para facilitar a organização, sendo: Introdução, Fundamentação teórica, Desenvolvimento, Resultados e Conclusão.

Na Introdução estão contidas diversas informações iniciais, que incluem considerações sobre o tema e o trabalho proposto. Nesta seção são definidos também os requisitos de projeto, os objetivos e as razões que justificam o seu desenvolvimento.

A fundamentação teórica apresenta o estudo do conhecimento de base necessário para a execução deste trabalho, com informações sobre todos os componentes, tecnologias, linguagens de programação e formatos de dados utilizados. Todas as funcionalidades

relevantes das plataformas serão explicadas, assim como seu funcionamento no presente projeto. O embasamento do estudo das grandezas físicas mensuradas também estará descrito nesta seção.

No Desenvolvimento estão descritas as razões que influenciaram a escolha dos serviços e componentes escolhidos, e apresentados o circuito eletrônico e seu funcionamento. São descritos os principais trechos de programação e as principais funções.

Na seção de Resultados apresentam-se um resumo e análises dos dados coletados durante os testes do protótipo produzido. Finalizando este capítulo são apresentadas as principais dificuldades encontradas durante o desenvolvimento do trabalho.

A Conclusão apresenta observações relacionadas às expectativas com este projeto, comentários oportunos sobre os objetivos que foram alcançados e sugestões de aprimoramentos e desenvolvimentos para outros trabalhos.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 SENSORES DE CORRENTE E TENSÃO

Serão utilizados sensores de tensão e corrente para a leitura dos parâmetros da rede. O circuito integrado ADE7753 nos permite a utilização de diversos tipos de sensores. Precisamos entender então a diferença entre eles para a seleção do sensor e sua correta utilização.

#### 2.1.1 SHUNTS RESISTIVOS

É composto basicamente por um resistor de baixo valor em série com a carga, dimensionado de maneira que a queda de tensão seja bem pequena. Partindo-se do princípio de que a queda de tensão é proporcional a corrente, pode-se mensurar a corrente utilizando este tipo de sensor.



Figura 2 - Shunts resistivos. Fonte: AMDS4 (2020)

#### 2.1.2 TRANSDUTORES DE EFEITO HALL

São dispositivos capazes de medir o fluxo de corrente com total isolamento galvânico, isto é, sem a necessidade de inserir o dispositivo em série no circuito a ser medido. Podem trabalhar com vários tipos de forma de onda e em uma ampla faixa de frequência. Possuem uma saída em corrente determinada através de uma relação entrada / saída e possuem um resistor de saída para transformar a saída de corrente em um valor de tensão. Pode-se trabalhar com correntes alternadas ou contínuas, mas necessita de fonte de alimentação auxiliar para o circuito hall.



Figura 3 - Transdutores de Efeito Hall. Fonte: AMDS4 (2020)

### 2.1.3 TRANSFORMADOR DE CORRENTE

Um transformador de corrente [1] (conhecido como CT (*Current Transformer*)) é um dispositivo que reproduz no seu circuito secundário, a corrente que circula em um enrolamento primário com sua posição vetorial substancialmente mantida, em uma proporção definida, conhecida e adequada. É de baixo custo e serve apenas para correntes alternadas.

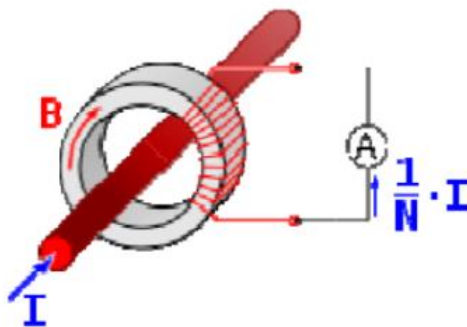


Figura 4 - Transformadores de Corrente. Fonte: AMDS4 (2020)

### 2.1.4 BOBINA DE ROGOWSKI

Na sua forma mais simples, a bobina de Rogowski é um sensor de corrente composto por uma bobina enrolada uniformemente com muitas ( $N$ ) voltas por metro em um núcleo não magnético de área transversal constante. Por não possuir núcleo magnético, possui baixa inércia eletromagnética, permitindo assim a medição de altas frequências com bastante precisão. Serve apenas para correntes alternadas (AC).





Figura 5 - Bobinas de Rogowski. Fonte: AMDS4 (2020)

### 2.1.5 ISOLAMENTO GALVÂNICO

Segundo a fornecedora YANASHI o isolamento galvânico em transdutores se faz necessário quando se deseja que duas grandezas elétricas associadas fiquem separadas eletricamente com o objetivo de atenuar interferências que possam causar danos ou distúrbios aos circuitos ligados na saída do transdutor isolador. O isolamento colabora com a segurança do circuito em casos de tensões altas, permitindo que a leitura do valor de sensoriamento possa ser feita em uma tensão menor respeitando uma razão conhecida.

### 2.1.6 TRANSFORMADOR E DIVISOR DE TENSÃO

Para o sensoriamento de tensões é comumente utilizado um circuito rebaixador composto de dois resistores em série com uma derivação central. Através da relação entre a resistência dos componentes utilizados obtêm-se uma amostra conhecida do valor de tensão de entrada. Pode-se utilizar em conjunto com este circuito um transformador de tensão com fator de redução conhecido para prover o isolamento elétrico.

## 2.2 CIRCUITO INTEGRADO ADE 7753

Este circuito integrado [6] mostrado na Figura 7 é um medidor que permite executar a medição instantânea de potência ativa, reativa, e aparente em circuitos monofásicos de baixa tensão. Permite também monitorar valores RMS de tensão e corrente, frequência, e de eventos específicos como quedas de tensão e índices de fator de potência.

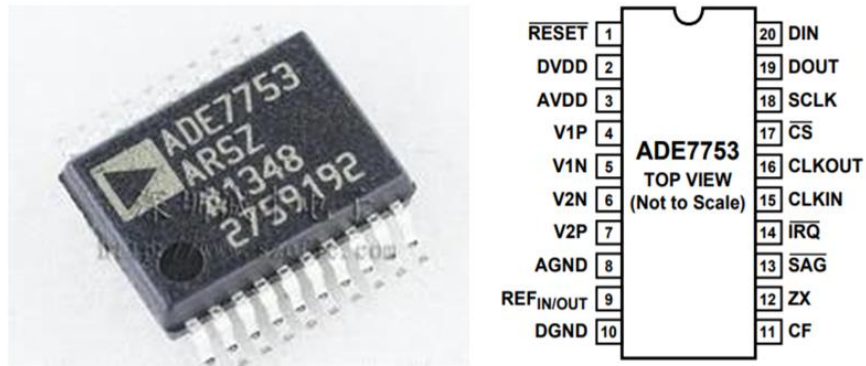


Figura 6 - Circuito integrado ADE7753. Fonte: Analog Devices (2010)

### 2.3 MÓDULO NODE MCU

O módulo de hardware conhecido como *NodeMCU* [3][4] é uma plataforma de desenvolvimento que permite a prototipagem de projetos [IOT](#). Os softwares utilizados são de código aberto (*open source*) e utilizam o microcontrolador Espressif ESP8266<sup>®</sup> que integra serviços de [GPIO](#), [PWM](#), I<sup>2</sup>C e conversores [A/D](#) além de possuir também um módulo de rede sem fio integrado que permite acesso à internet.

A parcela de serviços que será executada localmente está sob a responsabilidade deste dispositivo. O módulo NodeMCU coletará as informações de energia dos registradores do circuito integrado ADE7753 que está conectado aos sensores. Os dados adquiridos são processados e publicados como uma mensagem [JSON](#) em um tópico [MQTT](#) previamente definido.

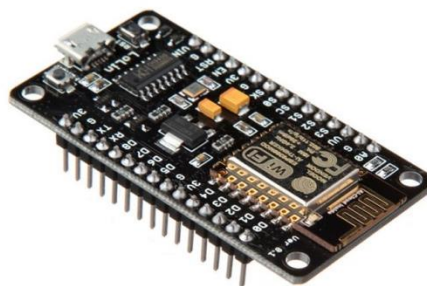


Figura 7 - Módulo Node MCU com ESP8266. Fonte: FilipeFlop (2020)

## 2.4 PROTOCOLOS DE COMUNICAÇÃO E NOTAÇÃO

### 2.4.1 A INTERNET DAS COISAS E O PROTOCOLO MQTT

O protocolo [MQTT](#) é um protocolo com baixo *payload* (leve) de mensagens [TCP/IP](#) para sensores ou dispositivos móveis. Este protocolo utiliza um esquema de comunicação fundamentado no modelo publicador-assinante e é muito utilizado em internet das coisas. Seu fundamento básico é apresentado na Figura 8.

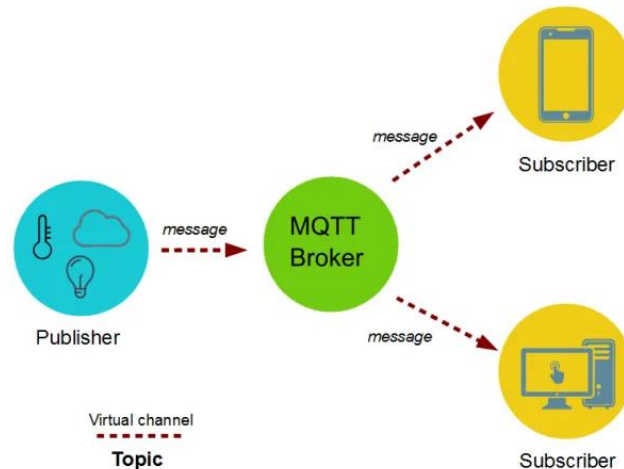


Figura 8 - Funcionamento do protocolo MQTT. Fonte: Ayush Gemini (2019)

São criados canais de comunicação virtuais chamados tópicos. Serviços e dispositivos podem assinar ou publicar em um tópico de um broker específico. O tópico não faz parte da mensagem, é apenas uma referência a um identificador de destino. Um dispositivo publica em um tópico, e o *broker* [MQTT](#) realiza o envio da mensagem recebida a todos os assinantes daquele tópico específico.

A comunicação entre o dispositivo e o *broker* depende do serviço utilizado. Neste trabalho utilizou-se o [AWS IOT Core](#) que se baseia em [TCP/IP](#) e transmite através da internet executando autenticação de segurança por meio do uso de certificados.

### 2.4.2 SERIAL PERIPHERAL INTERFACE ([SPI](#))

Este protocolo permite a comunicação do microcontrolador com diversos outros componentes, formando uma rede. É uma especificação de interface de comunicação série síncrona usada para comunicação de curta distância, principalmente em sistemas embarcados. No caso deste projeto a comunicação [SPI](#) permite que o módulo NodeMCU execute a leitura e escrita dos registradores do circuito integrado ADE7743.

### 2.4.3 NOTAÇÃO JSON

Utilizaremos também um protocolo de notação muito comum para padronizar o transporte de dados entre os serviços. O *Java Script Object Notation* (JSON) é um modelo de formatação leve para troca de dados baseado em um subconjunto da linguagem de programação *Java Script* e utiliza convenções familiares às linguagens *C*, *Java*, *Java Script*, *Perl*, *Python* e muitas outras. Este modelo é constituído basicamente de uma coleção de pares chave/valor. Em várias linguagens, isto é caracterizado como um dicionário ou matriz. Estas são estruturas de dados universais e praticamente todas as linguagens de programação modernas as suportam, sendo interessante que um formato de troca de dados que seja independente de linguagem de programação se baseie nesse tipo de estrutura. A Figura 9 apresenta um exemplo de pacote de dados [JSON](#).

```
{
  "id" : "0",
  "timestamp" : "15734695",
  "tensao" : "220",
  "corrente" : "220"
}
```

Figura 9 - Exemplo de pacote de dados JSON

## 2.5 AMAZON WEB SERVICES

A plataforma de serviços computacionais *online Amazon Web Services* ([AWS](#)) está entre uma das mais adotadas e abrangentes do mundo, oferecendo mais de 175 serviços disponíveis em qualquer lugar com acesso à internet. Milhões de clientes, incluindo empresas de crescimento rápido (startups), grandes empresas e os maiores órgãos governamentais, estão usando a [AWS](#) para reduzirem seus custos, ficarem mais ágeis e inovarem seus processos mais rapidamente.

Neste projeto utilizam-se três serviços: o banco de dados *Dynamo DB*, o núcleo de serviços de internet das coisas *IOT Core* e o serviço de computação em nuvem [EC2](#).

Um guia explicativo sobre a configuração dos serviços AWS utilizados está disponível no Apêndice B.

### 2.5.1 AWS IOT CORE

É um módulo da [AWS](#) que fornece serviços de [IOT](#) para soluções industriais, comerciais e de consumo. O módulo [AWS IOT Core](#) permite que dispositivos conectados interajam de maneira fácil e segura com aplicativos de nuvem e outros dispositivos.

Este serviço serve como um *broker* [MQTT](#) disponibilizando um canal de comunicação entre o dispositivo e a nuvem computacional, permitindo a publicação em um tópico específico, sendo assim responsável por receber as mensagens publicadas e retransmiti-las a todos os clientes.

Neste projeto o tópico utilizado foi definido como: 'aws/rules/Energy\_meter'. A Figura 10 apresenta o formato da mensagem publicada.

```
{
  "id": "0",
  "timestamp": "1587995345",
  "Tensao": "212.07",
  "Freq": "59.87",
  "Pot_At": "133.00",
  "Pot_Ap": "267.26",
  "Corrente": "1.26",
  "FP": "0.50",
  "Pot_Re": "231.45",
  "indice": "1"
}
```

Figura 10 – Publicação de mensagem no tópico

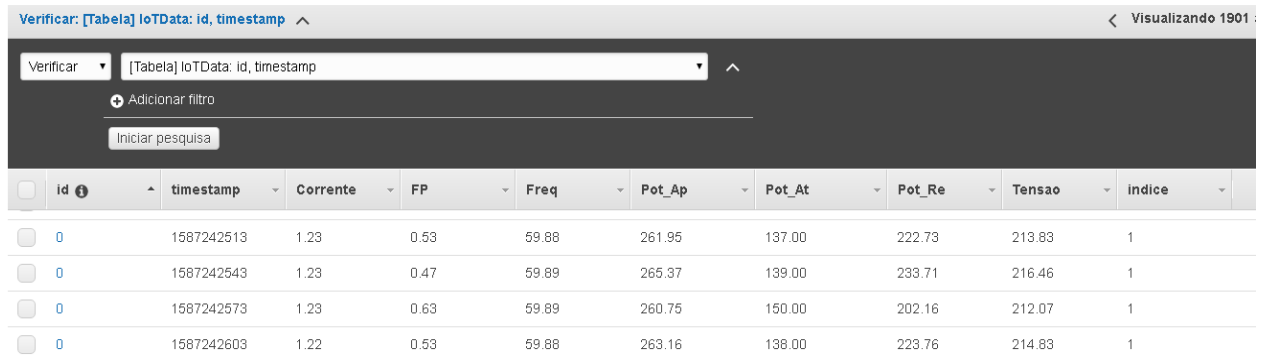
A informação útil (*payload*) é uma matriz em um formato [JSON](#). Os dados estão organizados por chave e valor, permitindo que a mensagem seja interpretada pelo serviço [IOT Core](#). Pode-se configurar uma ação para que o próprio *broker* realize a interpretação dos dados de cada mensagem e encaminhe o conteúdo para o banco de dados Dynamo DB. Para que isso funcione deve-se certificar de que todas as colunas descritas nas chaves existam de maneira idêntica no banco de dados Dynamo DB.

### 2.5.2 DYNAMO DB

É um serviço de banco de dados do tipo valor-chave que oferece ótimo desempenho e confiabilidade dos dados armazenados. [9] Permite acesso aos dados por meio de diversas linguagens computacionais, que incluem as mais utilizadas como *Python*, [PHP](#) e *Java Script*. [10]

Neste projeto utiliza-se esta plataforma para armazenar as informações coletadas pelo dispositivo de medição. Após a recepção de uma mensagem [MQTT](#), seu conteúdo é adicionado ao banco de dados pelo serviço *IOT Core*.

A Figura 8 mostra o formato de armazenamento de uma tabela do *Dynamo DB*. A primeira linha define as chaves e as demais linhas representam os registros de valores.



	id	timestamp	Corrente	FP	Freq	Pot_Ap	Pot_At	Pot_Re	Tensao	indice
<input type="checkbox"/>	0	1587242513	1.23	0.53	59.88	261.95	137.00	222.73	213.83	1
<input type="checkbox"/>	0	1587242543	1.23	0.47	59.89	265.37	139.00	233.71	216.46	1
<input type="checkbox"/>	0	1587242573	1.23	0.63	59.89	260.75	150.00	202.16	212.07	1
<input type="checkbox"/>	0	1587242603	1.22	0.53	59.88	263.16	138.00	223.76	214.83	1

**Figura 11 - Exemplo de tabela do Dynamo DB**

Cada registro é formado por um conjunto de chaves e valores, sendo uma chave de identificação do dispositivo, uma chave de classificação do registro e as demais chaves relacionam os dados adquiridos. A chave *id* é chamada de chave de partição. A chave *timestamp*, é uma chave de classificação e torna cada registro de dados único, por isso não pode haver repetições dos valores desta chave. Isso é garantido pelo fato de que o valor de *timestamp* é gerado no momento do envio do *payload* por um código [UTS](#) que corresponde à quantidade de segundos desde o início do primeiro dia de janeiro 1970, no formato UTC +0.

### 2.5.3 AMAZON EC2

É um serviço que disponibiliza capacidade computacional segura e redimensionável na nuvem computacional. Permite que seja configurado um sistema operacional em uma máquina virtual e que serviços rodando nessa instancia sejam acessados remotamente de qualquer local. É possível liberar portas e acessos conforme a necessidade.

Utiliza-se esta plataforma para hospedar a aplicação desenvolvida, que será gerenciada por um servidor *Apache*. Este servidor será responsável por responder a todas as solicitações [HTTP](#) de consulta de dados. Far-se-á uso de uma instância executando o sistema operacional *Ubuntu Server* e um servidor *Apache*. O acesso será realizado por meio do endereço de [DNS](#) público da instancia [EC2](#) por um navegador, tal como o *Chrome*.

#### **2.5.4 PACOTE XAMPP**

Adicionalmente aos serviços da Amazon, utiliza-se um pacote de softwares de uso livre chamado *XAMPP*. Esta solução é uma distribuição *Apache* fácil de instalar e utilizar que contém os serviços de interpretação *PHP*, *MySQL* e *Perl*. É de uso livre, inclusive para fins comerciais. Este serviço ficará hospedado em um servidor na nuvem e armazenará as páginas web e os scripts *PHP* relacionados à consulta dos dados. Os demais serviços do pacote não serão utilizados.

### **2.6 GRANDEZAS ELÉTRICAS**

Para que possamos trabalhar corretamente com os sensores a serem utilizados, precisamos entender os fundamentos relacionados às grandezas mensuradas. Neste projeto usar-se-á a medição de sete grandezas diferentes, são elas: tensão, frequência, corrente, potência ativa, potência reativa, potência aparente e fator de potência.

Grandezas como tensão, frequência, e corrente são corriqueiras e seus métodos de medição bem compreendidos. O circuito integrado ADE7753 usado realiza a medição da tensão por meio de um canal de entrada que mensura uma fração conhecida da tensão amostrada. Posteriormente são realizados diversos cálculos para obtenção do valor real. A frequência é obtida através da medição do período da onda amostrada, sabendo-se que a frequência é o inverso do período da onda, pode-se calcular facilmente seu valor. A medição de corrente neste caso será realizada por meio de um transformador de corrente, com resistor de amostragem que transforma níveis de corrente dentro de um range específico em níveis de tensão conhecidos com resposta linearizada, permitindo assim a medição de corrente com uma exatidão aceitável.

#### **2.6.1 POTÊNCIA APARENTE**

É a soma vetorial das potências ativa e reativa. É medida em VA. Por meio desta relação é extraído o valor do fator de potência de consumo da rede, que se resume ao ângulo entre os catetos formados pelas potências ativa e reativa. Este parâmetro da rede é muito importante, pois está relacionado à eficiência do consumo excessivamente capacitivo ou indutivo que, em alguns casos, pode ser mensurada e taxada pela empresa de fornecimento de energia. Neste projeto o valor da potência aparente foi calculado por meio da Equação 1.

$$S = VRMS.IRMS$$

#### **Equação 1 - Potência aparente**

### **2.6.2 FATOR DE POTÊNCIA**

O fator de potência (FP) é uma relação entre a energia ativa e a energia aparente do circuito em um determinado período de tempo. No ADE7753 temos dois registradores que integram e armazenam cada um desses tipos de energia. Com duas amostras de períodos distintos podemos calcular o FP. Se utilizássemos um período de 1 segundo entre a coleta das amostras teríamos o apresentado na Equação 2. Este é o método utilizado para a obtenção do fator de potência neste trabalho.

$$FP = \frac{(E_{P1} - E_{P0})}{E_{S1} - E_{S0}} = \frac{P}{S}$$

#### **Equação 2 - Fator de potência**

### **2.6.3 POTENCIA ATIVA**

A potência ativa instantânea é definida como a taxa de energia fluindo de uma fonte de alimentação para a carga a cada instante de tempo. É obtida pelo produto dos valores instantâneos de tensão, corrente e fator de potência. A unidade de medida da potência ativa é o Watt e equivale a um joule/segundo. A potência ativa está relacionada à realização de trabalho como, por exemplo, geração de calor, movimento ou luz. A potência ativa é calculada por meio da Equação 3.

$$P = V.I. \cos \theta = V.I. FP$$

#### **Equação 3 - Potência ativa**

### **2.6.4 POTÊNCIA REATIVA**

É a energia utilizada que é responsável por manter o campo eletromagnético em motores, geradores, transformadores e certos tipos de lâmpadas. Este tipo de energia não gera trabalho, mas tem seu estudo relevante, pois afeta a carga das redes elétricas, especialmente em grandes potências. Sua unidade de medida é o VAr. Neste trabalho seu valor será obtido por meio da Equação 4.

$$Q = V.I. \sin \theta = V.I. \sin(\arccos(FP))$$

#### **Equação 4 - Potência reativa**



## 3 DESENVOLVIMENTO

### 3.1 ELABORAÇÃO DO PROTÓTIPO

Para validar o projeto, o circuito eletrônico foi montado em uma placa de montagem conforme mostra a Figura 12.

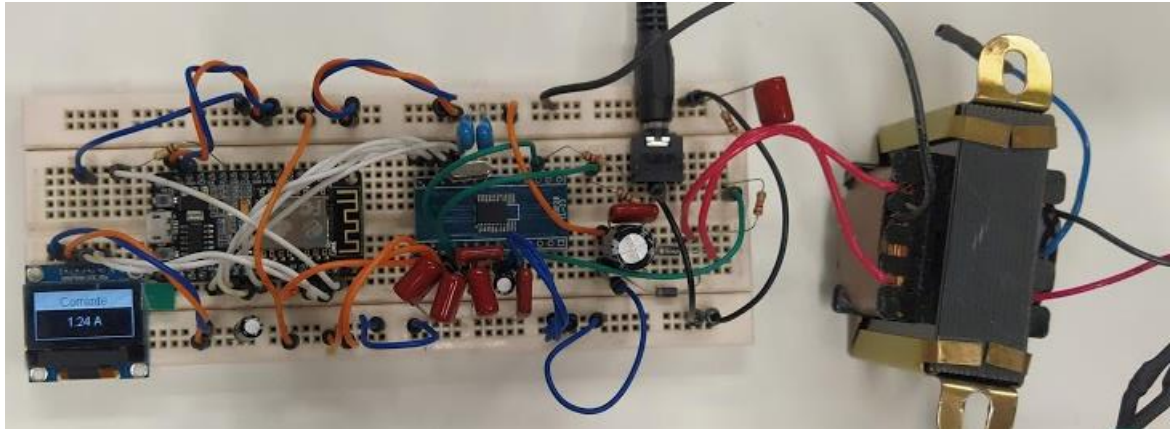


Figura 12 - Protótipo em placa de montagem

### 3.2 DESENVOLVIMENTO DA APLICAÇÃO WEB

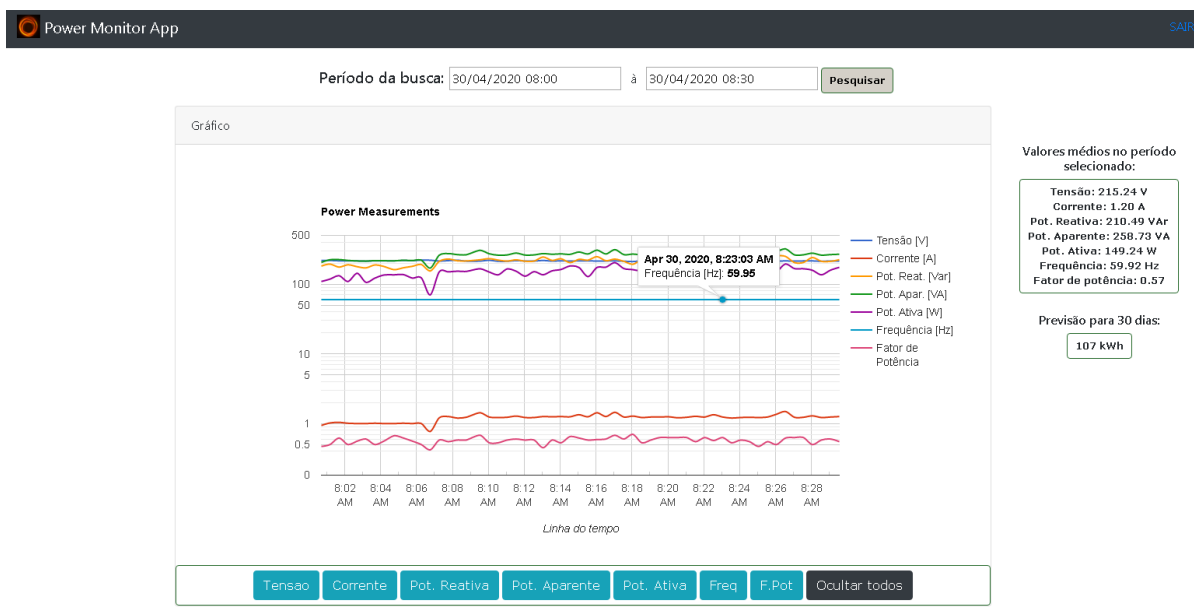
Conforme as informações apresentadas na metodologia, os aspectos necessários de configuração dos serviços WEB foram executados a contento. Os serviços da [AWS](#) foram configurados e testados, a página *web* para interface com o usuário foi desenvolvida e testada. A máquina virtual foi preparada, o servidor Apache instalado e os arquivos da aplicação *web* carregados. Os procedimentos de configuração da AWS estão descritos no apêndice B deste trabalho.

A aplicação *web* é composta de duas páginas: uma tela de autenticação, que é apresentada na Figura 13, e uma tela de dados, mostrada na Figura 14.

Login

**Figura 13 – Tela de autenticação de usuários na aplicação web**

A Figura 14 apresenta um exemplo de pesquisa e apresentação de dados.



**Figura 14 – Demonstrativo de consulta de dados na aplicação web**

### 3.3 ESCOLHA DOS SERVIÇOS E COMPONENTES

A escolha do circuito integrado ADE7753 foi considerada a mais adequada devido à ampla gama de funções desenvolvidas, bem como preço acessível e facilidade para

aquisição no mercado. Existem também, diversos trabalhos que utilizam este *chip*, fornecendo uma ampla diversidade de materiais para consulta.

O microcontrolador ESP8266 foi escolhido para uso neste projeto pelo baixo custo e adequação as necessidades do projeto. Seu tamanho físico reduzido com acesso à rede sem fio incluído o tornam uma boa escolha para utilização em projetos de [IOT](#).

A escolha da plataforma de serviços da *Amazon* foi uma escolha da equipe de projeto. Existem diversos trabalhos que utilizam outras plataformas, principalmente com servidores hospedados localmente, como por exemplo, utilizando o *Mosquitto* [MQTT Broker](#) e a *SQL database*. Os serviços da [AWS](#) se encaixaram neste projeto, pois ainda não existem muitos trabalhos acadêmicos que façam uso desta plataforma, além de ser um serviço comercial cujo aprendizado pode ser útil posteriormente. A configuração dos serviços desta plataforma não é de trivial aprendizado, mas está muito bem documentada. Além da documentação do fabricante, existem também tutoriais em vídeos na plataforma Youtube que também auxiliam no aprendizado desta plataforma. Todos os materiais utilizados estão alistados na bibliografia deste trabalho.

O modelo de sistema apresentado com todos os serviços hospedados em servidores de terceiros, possui diversas vantagens, como por exemplo, não existe a necessidade de dar manutenções regulares, nem ter uma equipe local de suporte, diminuindo as despesas com esses aspectos. A confiabilidade do serviço é aprimorada, sendo que a [AWS](#) promete uma disponibilidade [SLA](#) (*Service Level Agreement*) de 99,99% no serviço de *cloud computing*. As despesas por estes serviços são fixas e definidas pelas cotas de uso. Apesar disso, dependendo da aplicação, pode ser vantajoso manter um serviço hospedado localmente, que não dependa de serviços web contratados. Atualmente o valor de um contrato simples de computação em nuvem com a *AWS* está custando aproximadamente \$100 por mês. Então se não necessitamos de tanta confiabilidade quanto é oferecida pela [AWS](#), pode ser vantajoso testar provedores mais baratos deste serviço.

Trazendo a atenção para as linguagens de programação utilizadas, foram escolhidas linguagens que estão sendo muito usadas no mercado. A programação da ESP8266 foi desenvolvida em C, sendo a linguagem de mais baixo nível que compõe este trabalho. Todas as demais estão relacionadas à programação web, que podem englobar muitas linguagens em um mesmo sistema. A página web criada neste projeto utilizou trechos de código em

[HTML](#), [PHP](#), [CSS](#) e *Java Script*, além de utilizar pacotes de terceiros, como por exemplo, o *BOOTSTRAP* que traz recursos de design importantes para a plataforma *web*. Várias empresas disponibilizam [SDKs](#) (*Software development kit*). para utilização dos desenvolvedores e isto é muito útil, pois permite que trechos de códigos já prontos sejam reutilizados, facilitando o desenvolvimento. [18]

### 3.4 CIRCUITO ELETRÔNICO

Na Figura 15 é apresentado o circuito eletrônico do dispositivo proposto. O arranjo dos componentes foi elaborado através das sugestões dadas pelo fabricante no manual do componente.

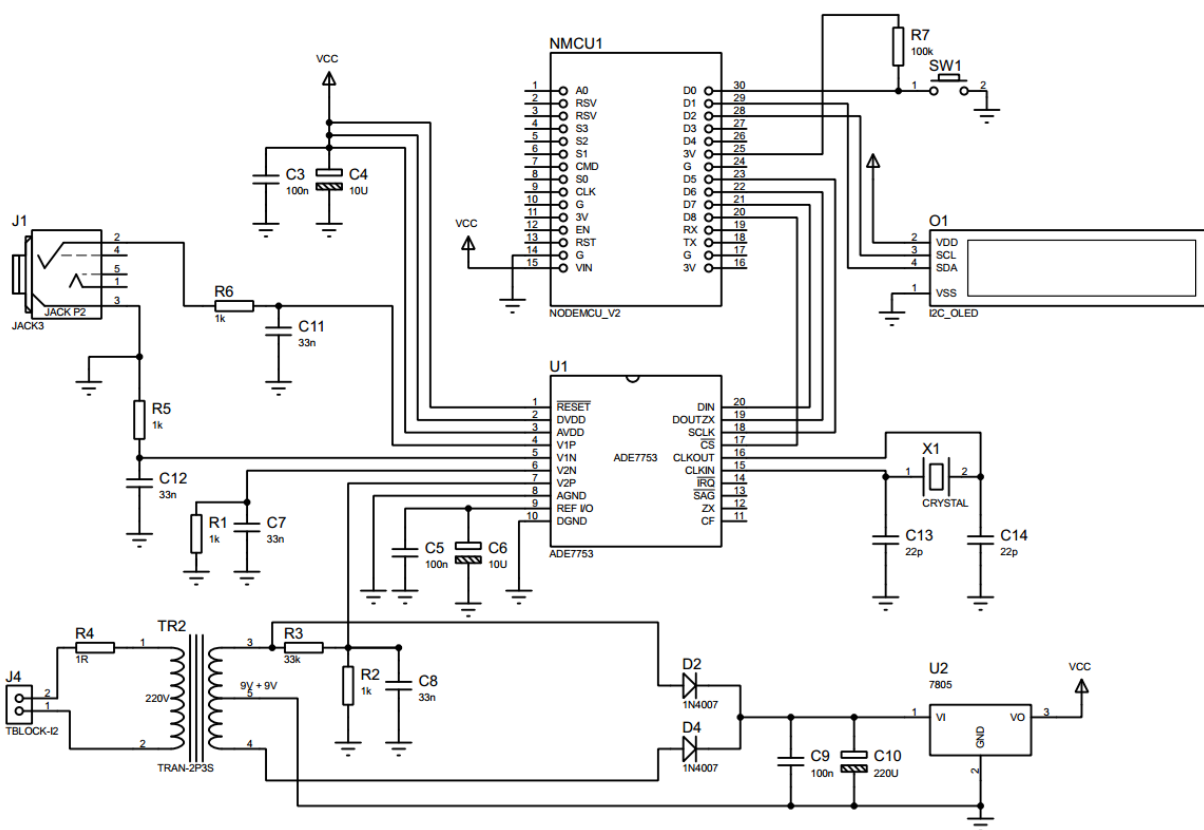


Figura 15 - Diagrama do circuito elétrico

O funcionamento deste circuito está construído sobre dois pilares principais. O primeiro pilar está no circuito integrado ADE7753, que é um medidor de energia monofásico da *Analog Devices*. O segundo pilar é composto pelo microcontrolador ESP8266 que faz parte do módulo *NodeMCU*.

O circuito integrado executa todas as tarefas relacionadas à interpretação dos valores retornados pelos sensores que a ele estão conectados, extraíndo os valores desejados de potências, tensão, corrente, entre outros. Todos estes valores estão armazenados em registradores que podem ser lidos por meio do protocolo de comunicação [SPI](#). Nos próximos itens descreve-se em detalhes cada um desses pilares.

### 3.4.1 MEDIÇÃO DE ENERGIA COM O ADE7753

As conexões elétricas deste circuito integrado foram executadas conforme instruções do fabricante presentes no *datasheet* do produto. A seguir então descritas às informações sobre os principais pinos utilizados e suas funções específicas:

- a) Os primeiro três pinos são respectivamente o pino de reset, a alimentação positiva do circuito digital ([DVDD](#)) e a alimentação positiva do circuito analógico ([AVDD](#)). Sendo o *reset* ativo em nível lógico baixo, manteremos estes três pinos conectados ao [VCC](#).
- b) Pinos 8 e 10 são pinos complementares de alimentação, respectivamente [AGND](#) (*Analogic Ground*) e [DGND](#) (*Digital Ground*) que devem ser conectados ao [GND](#).
- c) O pino 9 é uma referência interna que o fabricante recomenda que seja desacoplada do [AGND](#) utilizando dois capacitores conforme apresentado no diagrama.
- d) Pinos 15 e 16 são os pinos de *CLKIN* e *CLKOUT*, conectados a um cristal oscilador de 3.57MHz que fornece o *clock* de funcionamento do circuito.
- e) Pinos 17 a 20 são os pinos da comunicação serial, são conectados aos pinos de comunicação [SPI](#) do módulo *Node MCU*.
- f) Os pinos 6 e 7 são as entradas analógicas do canal 2 e devem ser utilizadas com um transdutor de tensão. Neste projeto faremos uso de um divisor de tensão resistivo acoplado a um dos enrolamentos do transformador de entrada de rede para obter uma amostra do valor de tensão.
- g) Os pinos 4 e 5 são as entradas analógicas diferenciais do canal 1 e são desenvolvidas para serem utilizadas em conjunto com um sensor de corrente. Estão conectadas a um conector TRS 3.5mm fêmea para utilização com o sensor de corrente.

Os registradores e principais protocolos utilizados para a comunicação entre este circuito integrado e o módulo *Node MCU* serão explanados posteriormente na seção sobre o software.

### 3.4.2 TRANSFORMADOR DE CORRENTE

Para aquisição dos dados de corrente usa-se o [CT](#) modelo SCT-013-020 conforme mostra a Figura 16. Segundo informações do fabricante, este sensor apresenta 3% de erro em fundo de escala, considerado aceitável pela equipe de projeto.

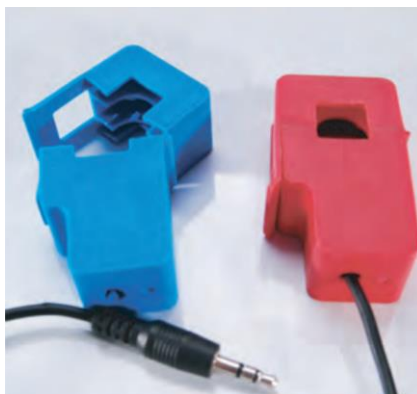


Figura 16 - Transformador de Corrente SCT-013-020. Fonte: Nice gear (2020)

### 3.4.3 FONTE DE ALIMENTAÇÃO

Na Figura 17 pode-se ver o circuito responsável pela alimentação do sistema.

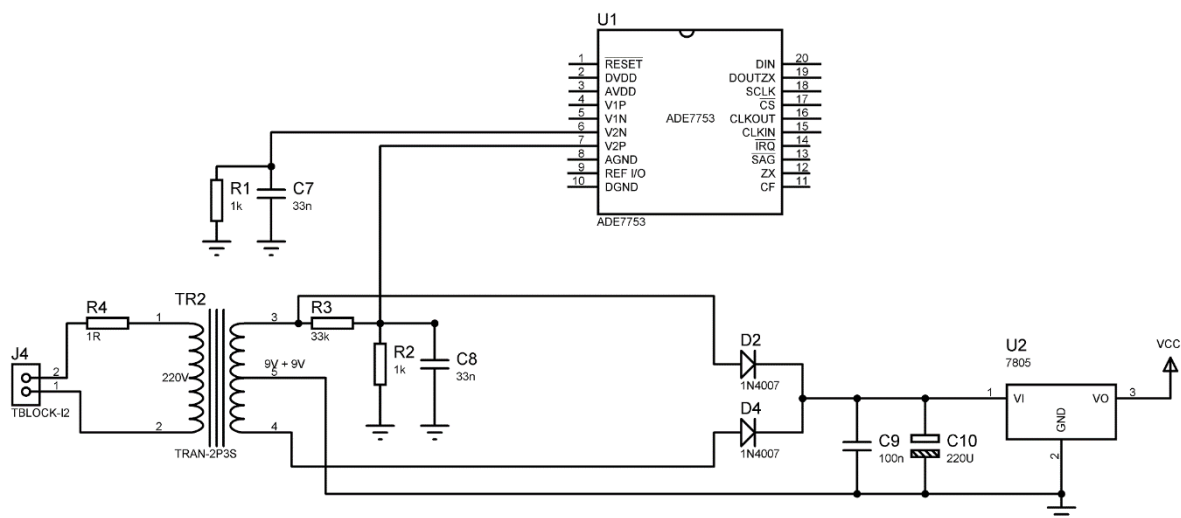


Figura 17 - Circuito de alimentação e canal 2 de Tensão

O conector J4 está conectado a entrada da rede [AC](#) em 220V. O resistor R4 atua como proteção do sistema em caso de curto circuito ou sobre corrente. TR2 é um transformador com derivação central de 9V + 9V que é retificada em um modelo de onda completa com dois diodos. Após a filtragem existe um regulador de tensão que estabiliza a tensão de saída [VCC](#) em 5V.

#### 3.4.4 CANAL 2 – SONDA DE TENSÃO

Na Figura 14 é apresentado também o amostrador de tensão que está conectado ao canal 2 do ADE7753. Enquanto o pino V2N está conectado ao GND por meio de R1 e C7, o pino V2P está conectado ao divisor de tensão formado por R3 e R2. Sendo a tensão eficaz sobre R3 + R2 de 9V, podemos calcular a tensão de pico utilizando a Equação 5.

$$VRMS = 0,707 * Vp$$

##### Equação 5 - Equação tensão de pico

Logo, a tensão de pico-a-pico ( $Vp$ ) sobre R3 +R2 será de:

$$V(R3 + R2)p = \frac{\pm 9V}{0,707} = \pm 12,73$$

##### Equação 6 - Calculo da tensão de pico no canal 2

A derivação central deste divisor de tensão fornece o valor para a amostragem no circuito integrado ADE7753. Conforme informações do fabricante, o valor de amostragem na entrada diferencial do circuito integrado deve estar entre +-0,5V. Para isso precisamos que a tensão sobre R2 esteja dentro deste intervalo.

Verificamos isso utilizando a equação para divisores de tensão. Para R3 = 33k e R2 = 1K temos:

$$VR2p = \frac{(VR3p + VR2p) * R2}{(R3 + R2)} = \frac{\pm 12,73 * 1k}{34k} = \pm 0,374Vp$$

##### Equação 7 - Calculo da tensão diferencial no canal 2

Obtendo-se esse valor de 0,374V para uma tensão de pico de 12,73V no transformador podemos elaborar uma relação para referenciar o valor da tensão da rede conforme as equações abaixo:

$$VRMS(rede) = Vp(rede) * 0,707$$

$$V_p(\text{rede}) = V_p(\text{primario})$$

$$\begin{aligned} V_p(\text{primario}) &= V_p(\text{secundario}) * (\text{fator de reducao do transformador medido indiretamente}) \\ &= V_p(\text{secundario}) * \left(\frac{217.3}{8.11}\right) \end{aligned}$$

$$V_p(\text{secundario}) = VR_{2p} * \left(\frac{12,73}{0,374}\right)$$

Substituindo as equações temos que:

$$VR_{RMS} = VR_{2p} * \left(\frac{12,73}{0,374}\right) * \left(\frac{217.3}{8.11}\right) * 0,707$$

$$VR_{RMS} = VR_{2p} * 644,22$$

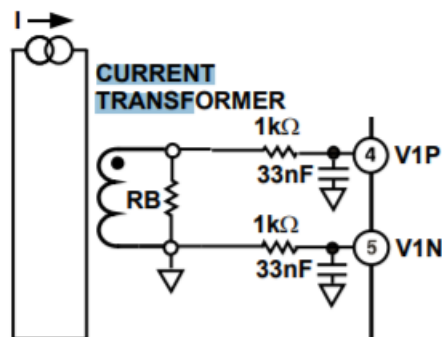
#### **Equação 8 - Cálculo do fator de multiplicação para calibração**

Assim podemos utilizar o valor mensurado pelo canal 2 para estimar a tensão da rede.

### **3.4.5 MEDIÇÃO DE CORRENTE**

O canal 1 será utilizado com um sensor do tipo transformador de corrente modelo SCT-013-020. Conforme as informações do fabricante, este sensor permite a leitura de correntes em uma faixa de 0 a 20A. Este sensor possui internamente um resistor de amostragem que gera um sinal de 0 a 1V na saída com erro de linearidade de 3%.

Instruções do fabricante orientam a instalação de um transformador de corrente conforme o modelo apresentado na Figura 18.

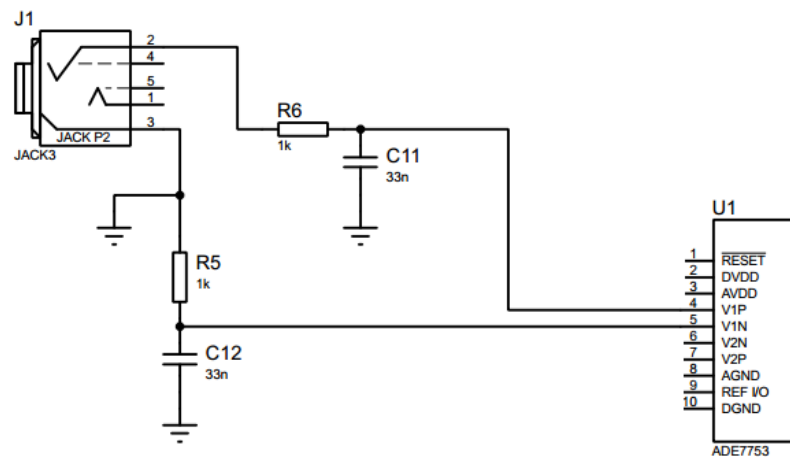


**Figura 18 – Diagrama de utilização de transformadores de corrente. Fonte: Analog Devices (2010)**

Neste projeto intentou-se a alteração do diagrama original pela adição de um divisor resistivo para que a tensão sobre a entrada diferencial não ultrapasse os 0,5V descritos no



manual do fabricante, visto que o sensor utilizado apresenta tensão de 0-1V para um range de corrente entre 0-20A. Como resultado prévio cita-se que esta configuração com divisor de tensão causou a depreciação do funcionamento e impedindo a medição. Para resolver o problema utilizou-se o sistema exatamente como descrito pelo fabricante, ficando a corrente de medição restrita a 0-10A para conformidade do range de entrada do circuito integrado. O diagrama final é apresentado na Figura 19.



**Figura 19 - Configuração de entrada do Canal 1 de Corrente**

Para este canal segue-se o mesmo modelo de cálculo utilizado no canal 1.

Temos um intervalo de saída em tensão sobre o resistor de amostragem interno ao sensor que retorna valores de 0 a 1V para correntes respectivamente de 0 a 20A. Utilizaremos um range máximo de 10A que garante o valor para 0,5V na entrada do canal 1. Logo temos a seguinte equação:

$$I = V * \left(\frac{10}{0,5}\right) = V * 20$$

**Equação 9 - Cálculo de valor de corrente no canal 1**

### 3.4.6 DISPLAY OLED



Figura 20 - Display *OLED* SSD1306. Fonte: ETT (2020)

A Figura 21 apresenta a conexão do display *OLED* (*organic light-emitting diode*) ao microcontrolador ESP8266.

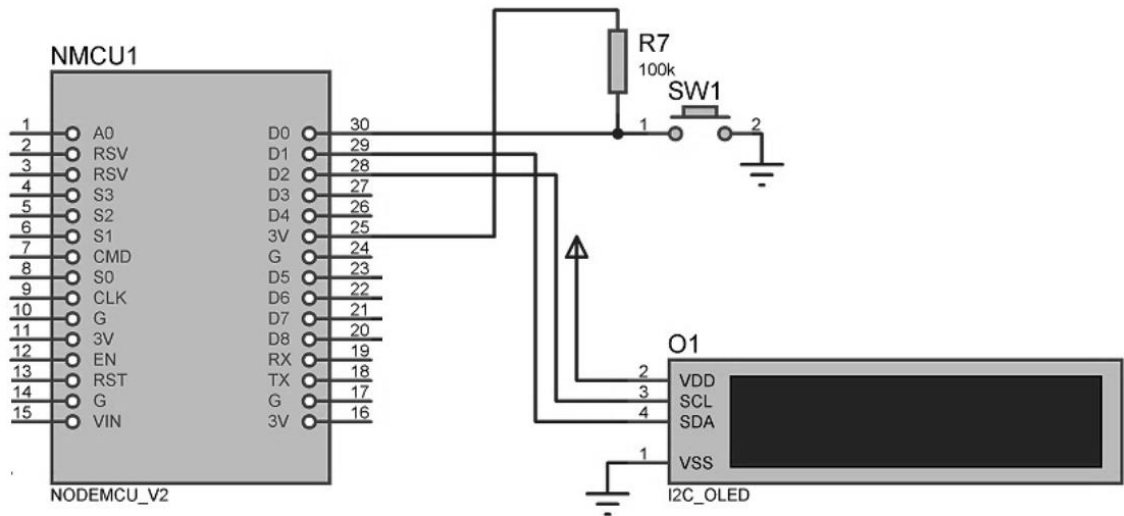


Figura 21 - Conexão do display *OLED* ao microcontrolador

Conectado aos pinos D1 e D2 do módulo *Node MCU* temos os pinos da comunicação [I2C](#) (*Inter-Integrated Circuit*) do display. Esta tela apresentará algumas informações do dispositivo, como o status e estabilidade da conexão com a internet e informações dos dados coletados da rede elétrica. Através da chave SW1 podemos fazer a mudança entre as informações apresentadas na tela. A Figura 22 apresenta a descrição das possíveis informações do *display*.

Conectando...	Informa que o dispositivo está tentando se conectar a internet.	
Conectado!	Informa que o dispositivo se conectou com sucesso a internet e em breve estará pronto para aquisição dos dados.	
Tensão	Corrente	Após a correta inicialização do dispositivo e conexão bem sucedida a uma rede wi-fi podemos verificar os parâmetros atuais de leitura. Para mudança de tela podemos utilizar a chave SW1.
220.00V	10.00A	
Freq	FP	
60.00Hz	1.00	
Pot_At	Pot_Ap	
2200W	2200VA	

Figura 22 - Informações do display

### 3.5 DESIGN DE PCI E INVÓLUCRO DE POLÍMERO

Para a montagem definitiva do protótipo que permitiria sua utilização profissional em um ambiente real, foi projetado um case de polímero conforme apresentado na Figura 23. Neste involuço estarão conectados um plugue elétrico macho para conexão a energia elétrica de alimentação do sistema e um plugue fêmea do tipo P2 para a conexão do sensor de corrente que deve ser conectado ao condutor fase da instalação a ser monitorada.



Figura 23 – Projeto de invólucro para o dispositivo

Foi projetada também uma placa de circuito impresso para alocação definitiva dos diversos componentes. A face superior da placa é apresentada na Figura 24.

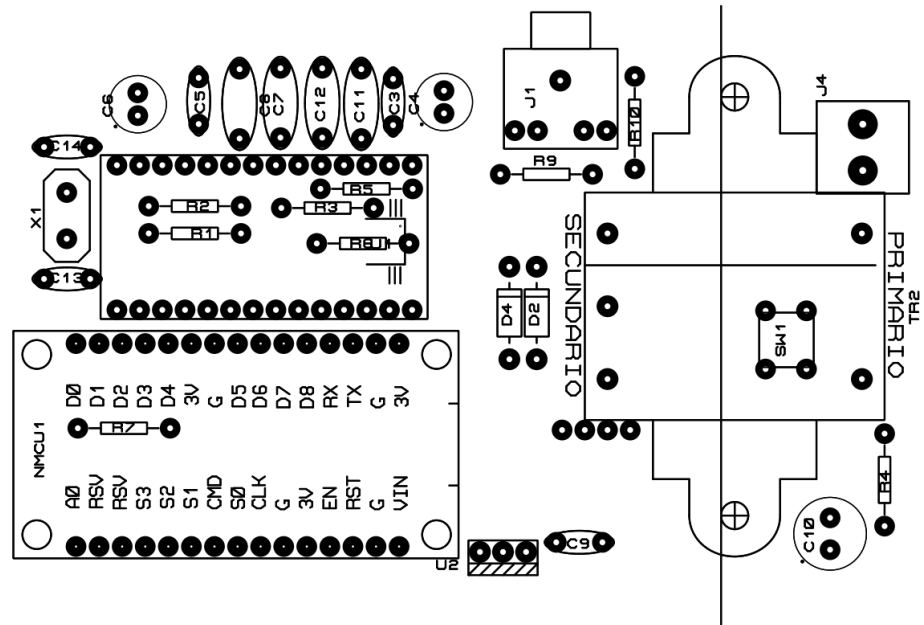


Figura 24 – Máscara de componentes da placa de circuito impresso

A face inferior da placa projetada é apresentada na Figura 25.

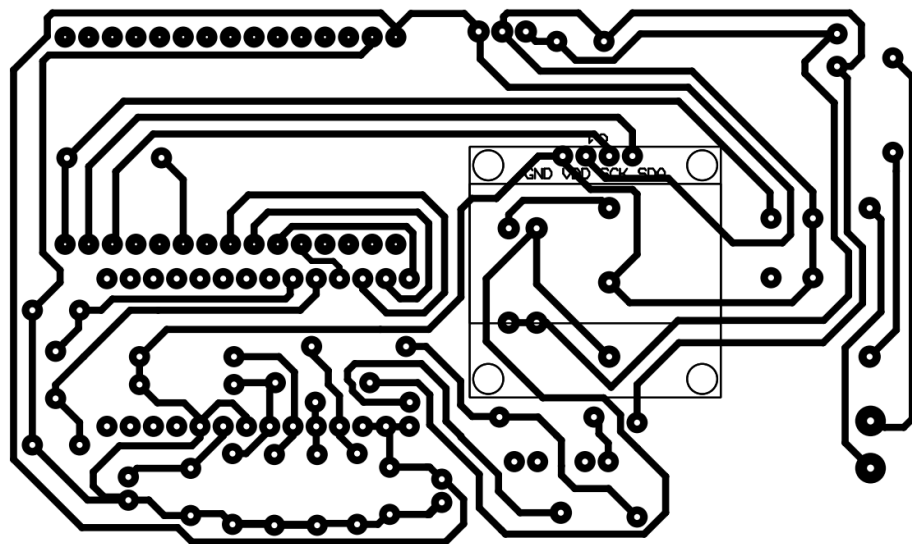
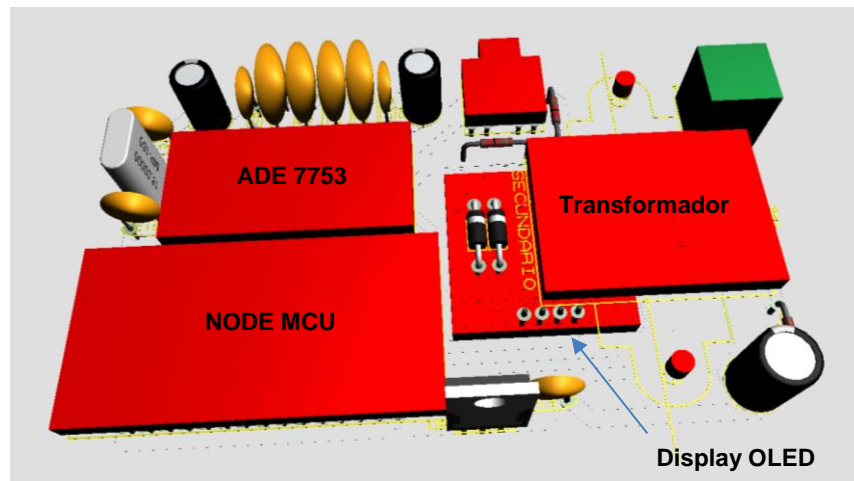


Figura 25 – Máscara inferior da placa de circuito impresso.



**Figura 26 - Vista 3D da placa**

Como pode-se notar na Figura 26, o display de OLED fica localizado na face inferior da placa, enquanto os demais componentes se encontram na face superior. Isso permite o encaixe no invólucro projetado.

Os componentes *NodeMCU* e *ADE7753* estarão soldados sobre suportes do tipo soquete elevados da placa, o que permite que existam componentes soldados na parte inferior da mesma face. Isso permite que o projeto seja executado em uma PCI de face simples e facilita a fabricação do mesmo com a estrutura disponível no instituto.

### **3.6 SOFTWARES DESENVOLVIDOS**

O software do sistema proposto está dividida em três partes principais: programação do módulo *NodeMCU* com o microcontrolador ESP8266 [4][5], a programação da interface *web* e a configuração dos diversos serviços [AWS](#).

A Figura 27 mostra um fluxograma do funcionamento do sistema de software.

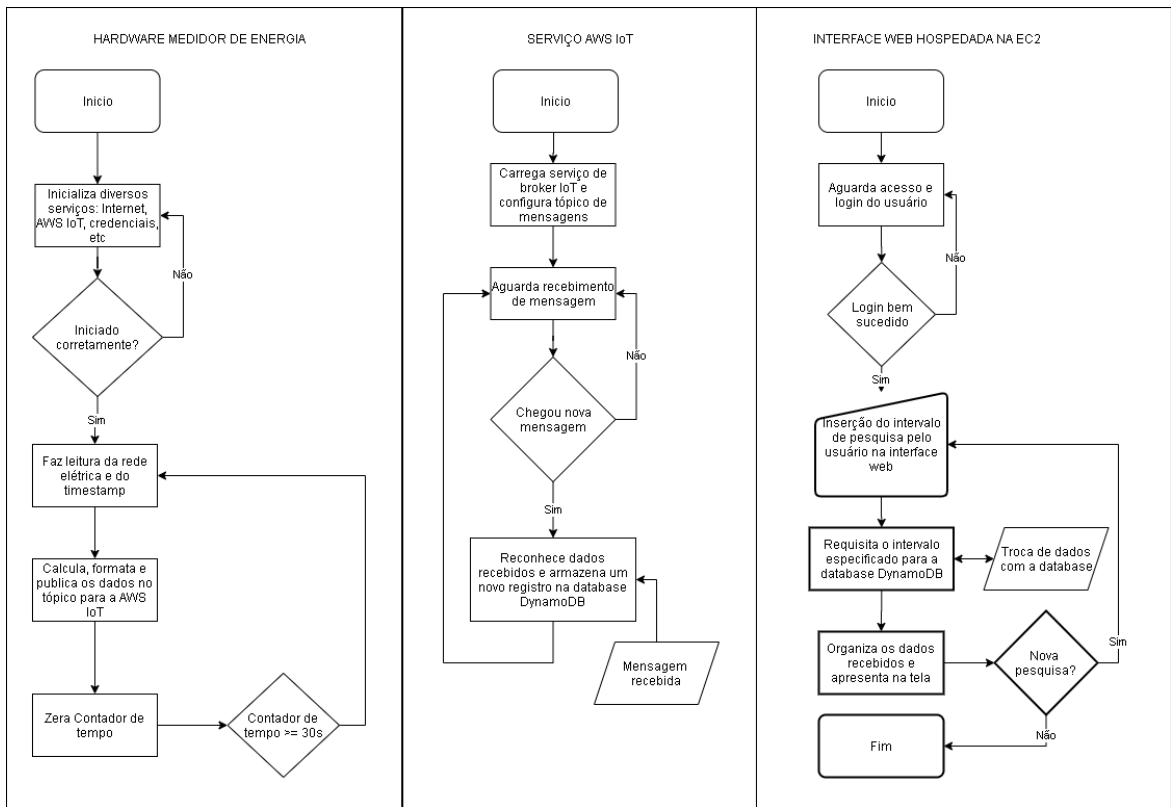


Figura 27 - Fluxograma do funcionamento do software

### 3.6.1 CIRCUITO INTEGRADO ADE 7753

Durante a inicialização sistema, é necessária que seja feita a configuração do modo de operação do ADE7753. Tais informações ficam armazenadas em memória volátil, logo precisa ser refeita a cada reinicialização. Para isso existem endereços de memória específicos chamados de registradores. Os métodos de leitura e gravação estão descritos na próxima seção deste trabalho. Existem também registradores utilizados para leitura dos dados mensurados pelo circuito integrado.

Na Tabela 1 destacamos os principais registradores utilizados neste trabalho e suas respectivas funções.

Nome	Endereço de memória	Função
0x02	AENERGY	Leitura dos valores registrados de energia ativa
0x05	VAENERGY	Leitura dos valores registrados de energia aparente
0x09	MODE	Escrita de diversos bits que configuram a operação do circuito
0x0B	STATUS	Leitura de diversas interrupções que podem ser configuradas, como por exemplo, tensão baixa, sobre tensão, temperatura, entre outros.
0x16	IRMS	Leitura do registrador de medição de corrente
0x17	VRMS	Leitura do registrador de medição de tensão
0x0D	CH1OS	Escrita da configuração de offset do canal 1
0x0E	CH2OS	Escrita da configuração de offset do canal 2
0x1C	LINECYC	Escrita da configuração de quantidade de ciclos de integração para medição de potência

**Tabela 1 – Principais registradores do ADE 7753**

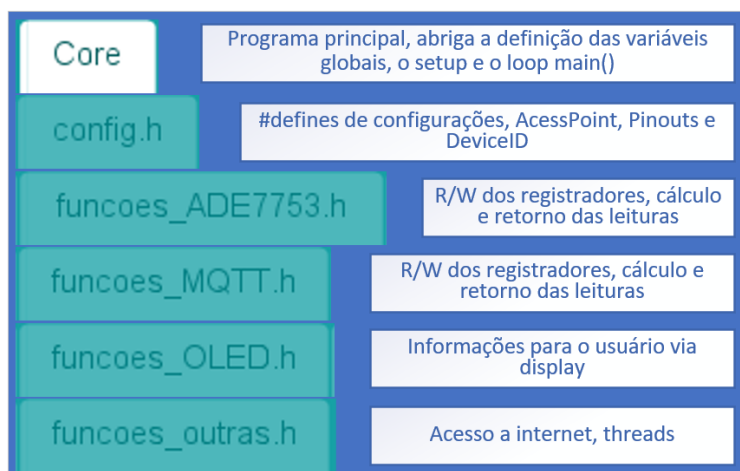
### **3.6.2 MÓDULO *NodeMCU***

Dentre as funções deste módulo atrelado ao microcontrolador, estão incluídas: acesso à internet por meio de uma conexão sem fio, escrita e leitura dos registradores do circuito integrado ADE7753, processamento das informações recebidas e posteriormente o envio de dados para a nuvem computacional. Esses dados são enviados por meio do protocolo [MQTT](#) contendo todos os dados para o tópico do *broker* [AWS](#).

Para programação das diversas funções descritas utiliza-se uma biblioteca chamada ESP8266.h disponível no *GitHub*. [7] Todos os links para os arquivos utilizados estão disponíveis nas referências bibliográficas deste trabalho. [17] Conforme as instruções da

biblioteca, usar-se-á a [IDE Arduino](#) para a programação. Todos os aspectos relacionados à instalação estão descritos no arquivo README da biblioteca.

O código fonte completo e comentado está disponível para consulta no GitHub (<https://github.com/cristianritter/EnergyMeter>). Este foi dividido em seis arquivos para facilitar a organização e compreensão. E estrutura ficou definida como se apresenta na Figura 28.



**Figura 28 - Descrição da programação do microcontrolador**

Os principais trechos do código são descritos a seguir.

A configuração da rede sem fio é feita conforme o código apresentado na Figura 29.

```
Core § config.h § funcoes_ADE7753.h funcoes_MQTT.h funcoes
//WIFI
const char* ssid = ██████████ //WiFi network name
const char* password = ██████████; //WiFi network password
```

**Figura 29 - Configuração da rede wi-fi**

A Figura 30 apresenta a configuração do nome do dispositivo que é utilizada para identificação de uma *thing* na tabela de dados. Podemos ter inúmeros dispositivos gravando dados paralelamente na mesma *database*, contanto que esta função seja implementada no futuro.

```
Core § config.h § funcoes_ADE7753.h funcoes_MQTT.h funcoes_OLED.h funcoes_outras.h
#define deviceid "0" //nome do device é armazenado na coluna id da tabela DynamoDB
```

**Figura 30 - Configuração de identificação de dispositivo**



O tempo entre *uploads* pode ser configurado dependendo da periodicidade desejada. Cada registro possui um valor de cada parâmetro mensurado.

```
Core  config.h$  funcoes_ADE7753.h  funcoes_MQTT.h  funcoes_OLED.h  funcoes_outras.h
#define time_between_uploads 10000 //ms          //tempo entre uploads - uma amostra por vez
```

Figura 31 - Configuração de periodicidade

No arquivo `funcoes_MQTT.h` são definidos os endereços de hosts do dispositivo na AWS e os certificados de segurança necessários para autenticação do dispositivo. Estes certificados são gerados durante a criação de uma *thing* e serão descritos posteriormente.

```
Core  config.h$  funcoes_ADE7753.h  funcoes_MQTT.h$  funcoes_OLED.h  funcoes_outras.h
const char *host = "a6ufkvuua7imp-ats.iot.us-east-2.amazonaws.com"; // AWS IoT Core

|
// : xxxxxxxxxxxx-certificate.pem.crt
const char cert_str[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----

```

Figura 32 - Configuração de certificados

O Circuito integrado ADE7753 possui vários registradores que podem ser lidos ou escritos conforme a necessidade. Antes de estar pronto para ler os parâmetros da rede, precisamos configurar alguns parâmetros de modo de operação, *offsets* e ganhos. Isso é feito por meio da escrita em registradores específicos. A função `configADE()` executa estas configurações por gravar os valores definidos de cada registrador.

```
Core  config.h$  funcoes_ADE7753.h$  funcoes_MQTT.h$
void configADE(){
    write16(MODE, MODE_VALUE_16BITS);
    write16(IRQEN, IRQEN_VALUE_16BITS);
    write16(LINECYC, LINECYC_VALUE_16BITS);
    write16(APOS, APOS_VALUE_16BITS);
    write8(CH2OS, CH2OS_VALUE_8BITS);
    write16(VRMSOS, VRMSOS_VALUE_12BITS);
}
```

Figura 33 - Configuração dos registradores do ADE7753

O método de escrita nos registradores é descrito na função a seguir. Primeiramente habilita-se a comunicação [SPI](#) utilizando a função `enableChip()`. O parâmetro *reg* carrega o

endereço do registrador que é sempre um valor de oito *bits* com o *bit* mais significativo ([MSB](#)) nulo. Para indicar ao chip que desejamos realizar uma escrita, devemos enviar o endereço do Registrador que deve ser escrito com o *bit* [MSB](#) ligado. Isso é feito utilizando a função (endereço do registrador | 0b1000000). Respeitando os tempos de atraso descritos no *datasheet* podemos enviar os dados que correspondem aos valores a serem gravados no registrador e enfim finalizar a comunicação. Deve-se respeitar o tamanho do registrador de destino, neste caso foi enviado um valor de 16 *bits*.

```
Core  config.h$  funcoes_ADE7753.h$  funcoes_MQTT.h$  funcoes_OLED.h  funcoes_outras.h
void write16(unsigned long reg, unsigned long value){ // value 16bits
    enableChip();
    SPI.transfer( (reg | 0b10000000) ); //colocar 1 no MSB indica gravacao no registrador
    delayMicroseconds(20);
    SPI.transfer(value>>8); //envia os 8 bits mais significativos
    delayMicroseconds(20);
    SPI.transfer(value&0xFF); //envia os 8 bits menos significativos
    delayMicroseconds(100);
    disableChip();
}
```

**Figura 34 - Método de escrita nos registradores**

A leitura dos registradores é executada seguindo o mesmo princípio. No código abaixo pode-se observar seu funcionamento. Após habilitar a comunicação enviamos o endereço do registrador com o [MSB](#) nulo. Com isso o *chip* entende que será feita uma leitura do valor do registrador. Para executar a leitura dos registradores utilizamos a função *SPI.transfer* para enviar um valor qualquer e coletamos o resultado da função. A cada valor enviado recebe-se uma palavra de oito *bits* de informação. Ao coletar todos os *bytes* do registrador finaliza-se a comunicação e retorna-se o valor contendo a informação completa.

```

Core   config.h   funcoes_ADE7753.h §   funcoes_MQTT.h   funcoes_OLED.h   funcoes_ou
unsigned long read24(char reg){ //Le um registrador de 24bits
    enableChip();
    unsigned char b2,b1,b0; //tres variaveis de 8 bits
    SPI.transfer(reg); //inicia a comunicacao
    delayMicroseconds(5);
    b2=SPI.transfer(0x81); //envia um dummy e recebe 8 bits por vez
    delayMicroseconds(5);
    b1=SPI.transfer(0x81);
    delayMicroseconds(5);
    b0=SPI.transfer(0x81);
    delayMicroseconds(150);
    disableChip();
    return ((unsigned long)b2<<16 | (unsigned long)b1<<8 | (unsigned long)b0);
}

```

**Figura 35 - Método de leitura dos registradores**

A função *valorMedio()* permite a execução das funções de leitura dos registradores várias vezes, soma seus valores e define um valor médio, para evitar imprecisões pontuais nos valores mensurados. Ela recebe três parâmetros: a quantidade de amostras que devem coletadas, a função de leitura que retorna os valores e o registrador a ser consultado.

```

unsigned long valorMedio( int qtde_amostras, unsigned long (*function)(char), char reg ){
    unsigned long media = 0;
    for (int x=0; x<qtde_amostras; x++)
    {
        media = media + (*function)(reg);
    }
    media = media / qtde_amostras;
    return media;
}

```

**Figura 36 - Método de cálculo de média**

As funções de leitura dos parâmetros da rede são muito similares, os conceitos aplicados a uma podem ser facilmente adaptados para o entendimento das demais. Citando como exemplo a função de leitura da frequência da rede temos o código da Figura 37. A função *valorMedio()* realiza 10 leituras do registrador PERIOD utilizando a função *read16()*. Este registrador possui o valor do período da onda que estamos monitorando no canal 2 do ADE7753. Sabendo que desejamos obter a frequência que é dada pelo inverso do período, e que o período depende do *clock* aplicado ao sistema, definimos uma equação para o cálculo da frequência.

Core	config.h	funcoes_ADE7753.h §	funcoes_MQTT.h
------	----------	---------------------	----------------

```

float retHz() { //le periodo e retorna frequencia
    const String reg = "PERIOD";
    unsigned long measure;
    float calculado;
    measure = (valorMedio(10, read16, PERIOD));
    calculado = CLOCKIN/(8.00*measure);
    if (imprimir_na_serial){
        printoutData(reg, measure, calculado);
    }
    return calculado;
}

```

Figura 37 - Exemplo de método de retorno de valor

### 3.6.3 INTERFACE WEB

Os dados que são coletados pelo microcontrolador são enviados para o *broker* e ficam armazenados em um serviço chamado *Dynamo DB*, que é um serviço de armazenamento de dados da [AWS](#). Para manipular estes dados temos várias opções dadas por um [SDK](#). Estes kits estão disponíveis para diversas linguagens conforme descrito na documentação do desenvolvedor para *Dynamo DB*. Neste trabalho faremos uso da [SDK](#) para [PHP](#).

Quando se deseja programar uma página *web*, tem-se como boa prática realizar a escrita do código em uma estação local, para somente após a conclusão e os testes finais exportar e disponibilizar o código completo para um servidor na internet. Para isso, utilizou-se um computador com os mesmos serviços que serão posteriormente instalados na estação [EC2](#), que incluem o servidor *Apache* com interpretador [PHP](#).

Após a instalação do servidor *Apache*, as páginas criadas estarão acessíveis através do endereço de *localhost* do navegador. A programação deve ser feita dentro do diretório 'htdocs' do pacote *XAMPP* de hospedagem da página web que por padrão fica instalado na raiz do sistema.

Criou-se uma pasta com o nome da nossa aplicação web dentro de 'htdocs' e então teremos nosso espaço de trabalho preparado para iniciar a programação. Dentro desta pasta faremos também o download dos arquivos [SDK PHP](#) para [AWS](#) [11] que nos permitirão a consulta dos dados.

Conforme instruções da documentação, deve-se utilizar o seguinte trecho de código para incluir a [SDK](#) em nosso código. É necessário corrigir o endereço para o caminho correto da biblioteca no nosso sistema.

```
<?php
    require '/path/to/aws-autoloader.php';
?>
```

**Figura 38 - Inclusão da AWS SDK no projeto**

Utilizamos o trecho de código a seguir para selecionar qual servidor estamos utilizando e realizar a autenticação.

```
$sdk = new Aws\Sdk([
    'region' => 'us-east-2', //localização física dos servidores utilizados
    'version' => 'latest',
    'credentials' => [
        'key' => 'AKIAIOSFODNN7EXAMPLE',
        'secret' => 'wJalrXU3FJOQJH76tKtZn9iK3VJpwab7k',
    ],
]);
```

**Figura 39 - Configuração de região e credenciais no código PHP**

O código abaixo permite que se consulte dados na tabela chamada 'IoTData'. São passados três parâmetros: 'device\_id', 'begin\_time' e 'end\_time', que selecionam o id do dispositivo e intervalo de tempo no qual será feita a pesquisa. O resultado é devolvido em formato de uma matriz pela variável *\$result*. De posse dos resultados podemos manipulá-los conforme desejado utilizando as diversas ferramentas [PHP](#) e [HTML](#). O código final da aplicação apresenta algumas alterações para permitir a seleção de um índice de registros, permitindo a consulta de uma amostragem parcial dos valores conforme o período da busca.

```

$dynamodb = $sdk->createDynamoDb(); //inicia uma sessao de consulta de dados

$tableName = 'IoTData'; //nome da tabela

$time1 = (string) strtotime($_GET['begin_time']); //parametros de consulta
$time2 = (string) strtotime($_GET['end_time']);

$eav = array //especifica uma query das chaves que serao buscados
(
    ':device_id' => array('S' => '0'),
    ':time1' => array('S' => $time1),
    ':time2' => array('S' => $time2)
);

$params = [ //parametros de busca
    'TableName' => $tableName,
    'KeyConditionExpression' =>
    '#dvcid = :device_id and #tmstamp between :time1 and :time2',
    'ExpressionAttributeNames' => [ '#dvcid' => 'id', '#tmstamp' => 'timestamp'],
    'ExpressionAttributeValues' => $eav
];

try {
    $result = $dynamodb->query($params); //coloca o array recebido na variavel $result
    //echo "Query succeeded.\n";
} catch (DynamoDbException $e) {
    echo "Unable to query:\n";
    echo $e->getMessage() . "\n";
}

```

Figura 40 – Solicitação de busca de dados em PHP

A aplicação web desenvolvida tem sua estrutura de arquivos definida conforme apresentado na Figura 41.

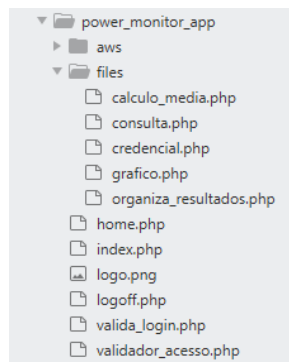


Figura 41 - Estrutura de arquivos da aplicação web

## 4 RESULTADOS ALCANÇADOS

Infelizmente não foi possível obter resultados da comparação entre os valores medidos pelo dispositivo e os valores reais aferidos por equipamentos calibrados conforme normas internacionais, dado as dificuldades da pandemia covid-19. No entanto pode-se destacar o conjunto de resultados alcançados o funcionamento de todos diversos componentes que interagem neste sistema.

### 4.1 INTEGRAÇÃO

A configuração do módulo *Node MCU* foi concluída com sucesso e realizou a conexão ao serviço de [MQTT](#) conforme se esperava. A integração do dispositivo com os serviços da plataforma [IOT Core](#) foi executada com êxito. O envio dos dados foi realizado corretamente e seu armazenamento também se comportou conforme o esperado.

### 4.2 INTERFACE WEB

A interface web apresentou inicialmente alguns bugs que foram tratados, como, por exemplo, erros durante a solicitação de dados que não existiam no banco de dados. Também foi tratado um erro relacionado à busca de um número muito grande de dados em uma única pesquisa, tornando a interface instável. Para resolução deste erro foi desenvolvido um algoritmo que limita a quantidade de dados retornados, fazendo com que haja uma busca seletiva do banco de dados dependendo do intervalo de pesquisa. Para intervalos muito amplos, não é feita uma solicitação de todos os dados de maneira contínua, mas sim amostrada por intervalos definidos pelo algoritmo de busca.

### 4.3 COMPUTAÇÃO NA NUVEM

O serviço de computação em nuvem [EC2](#) utilizado para hospedagem da aplicação *web* se mostrou muito confiável, não apresentou indisponibilidade e permitiu múltiplos acessos sem apresentar lentidão ou travamentos. Após a realização da configuração inicial, o acesso torna-se simples e não gera dificuldades para possíveis atualizações na programação dos serviços hospedados.

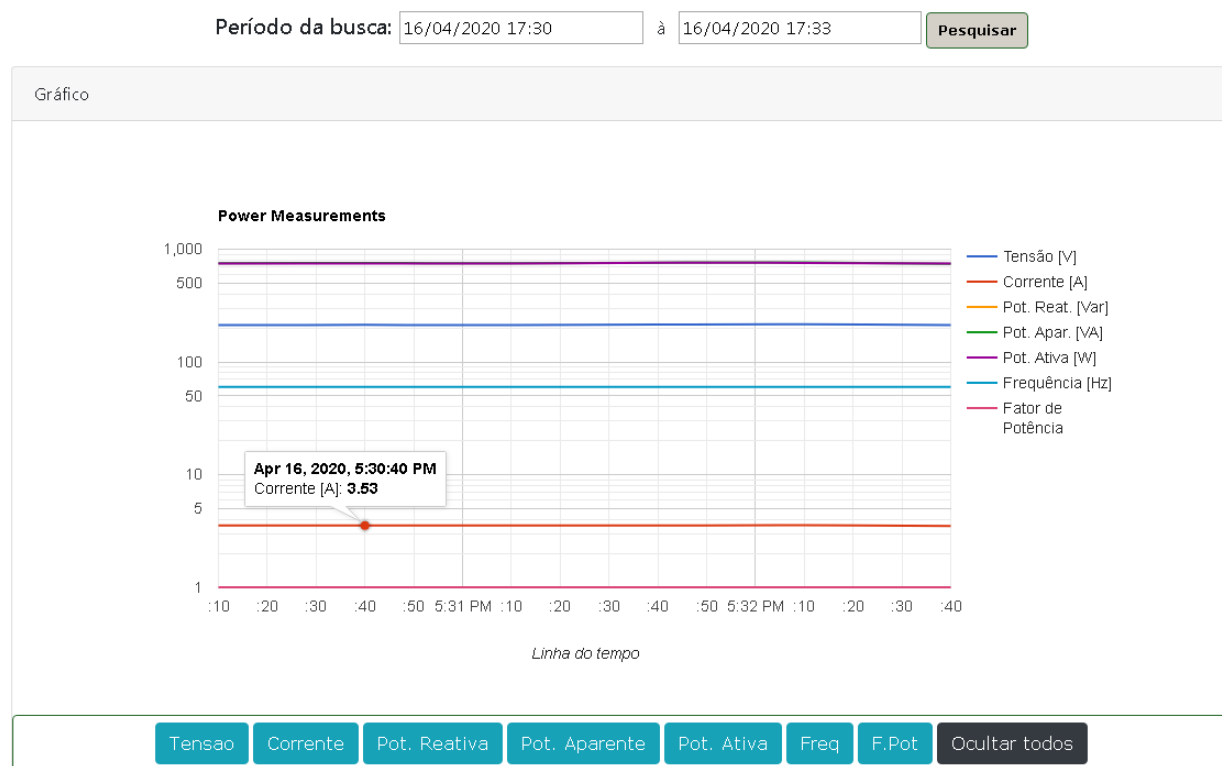
### 4.4 MEDIÇÕES REALIZADAS

Os resultados relacionados à medição da rede elétrica realizada pelo circuito da *Analog Devices* também apresentaram resultados coerentes. Conforme mencionado anteriormente,

por motivo da pandemia covid-19 não houve acesso aos instrumentos de medição do IFSC, logo, para a realização das simulações foram utilizadas cargas conhecidas que apesar de não possuírem uma tolerância restrita de suas especificações de consumo, permitirão que sejam realizados testes de funcionamento das medições. Parâmetros da rede que são muito estáveis também são considerados como sendo parâmetros padrões para efeito de comparação.

Utilizou-se um multímetro MINIPA TRUE RMS Modelo ET-1639 para realizar as calibrações das medições de corrente e tensão e um multímetro Modelo FLUKE 107 para comparações das medições de frequência da rede. Durante os testes utilizou-se de uma carga estática puramente resistiva de 750 W.

A Figura 42 apresenta uma coleta de dados apresentada na interface *web*.



**Figura 42 - Dados analisados**

Os valores das medições realizadas não serão apresentados neste relatório devido à imprecisão do sistema montado apenas em *protoboard*, no entanto, podemos validar o funcionamento de todos os componentes analisando os resultados no escopo geral da Figura



42. Percebemos que os resultados, apesar de sofrerem influência de imprecisões das calibrações sem equipamentos adequados e da instalação precária, são coerentes com a realidade dos testes realizados.

No caso de uma montagem adequada com a confecção de placa de circuito impresso as medições podem ter um resultado bem preciso, chegando a um máximo de 0,5% para uma medição em fundo de escala, conforme a especificação da *Analog Devices*. Este erro depende também da precisão dos outros componentes utilizados.

Desenvolveu-se ainda uma seção de médias e estimativas de consumo que está apresentada na Figura 43. Estes resultados são calculados apenas com os dados presentes no gráfico e não como o conteúdo inteiro do banco de dados, logo, quanto maior o período de amostragem mais preciso se mostra a estimativa. Essa estimativa é calculada e se baseia na média diária, multiplicada pelo número de dias que tem um mês, que não leva em conta outros fatores sazonais.

Valores médios no período selecionado:

Tensão: 213.37 V
Corrente: 3.53 A
Pot. Reativa: 0.00 VAR
Pot. Aparente: 754.15 VA
Pot. Ativa: 750.17 W
Frequência: 59.92 Hz
Fator de potência: 1.00

Estimativa para 30 dias:

540 kWh
---------

**Figura 43 - Médias e estimativas de consumo**

## 5 CONCLUSÕES

Como comentários conclusivos pertinentes a este trabalho, pode-se afirmar que os objetivos propostos foram plenamente atingidos.

O hardware de aquisição, embora não fabricado, teve seu projeto em tamanho compacto, sendo que o invólucro com todos os componentes apresentou as dimensões de 120 x 70 x 50 [mm].

O projeto do produto seguiu todas as premissas básicas de segurança para equipamentos elétricos, como por exemplo, o isolamento galvânico do circuito de tensão da rede do restante dos componentes. Além disso, manteve todos os componentes com risco de choque elétrico protegidos pela barreira do invólucro de polímero, garantindo assim os objetivos de oferecer segurança aos usuários. A instalação foi simplificada pela facilidade de conexão em uma tomada elétrica e uso do sensor de corrente não invasivo no circuito.

O custo do produto ficou dentro das expectativas previstas, mas os custos dos serviços da AWS devem ser considerados no caso de uma aplicação do produto em âmbito comercial. Neste projeto foram utilizados os serviços oferecidos pela empresa gratuitamente em uma conta para período de experimentação.

A plataforma *web* desenvolvida apresentou-se capaz de realizar as tarefas especificadas de consulta e apresentação de dados para o usuário de maneira fácil de utilizar e gerenciando os dados com clareza e agilidade.

O sistema concebido de medição dos diversos parâmetros de energia elétrica realizou com êxito o envio dos dados e sua posterior hospedagem na nuvem por uma rede sem fio. O acesso aos dados ficou protegido por um sistema seguro de *login* para consulta.

Um resultado que ficou aquém do esperado está relacionado à calibração das medições e a ausência de testes elaborados envolvendo cargas indutivas e capacitivas. Poderia ter sido desenvolvido um trabalho mais completo com ferramentas de calibração e cargas adequadas que estavam disponíveis apenas nas instalações do instituto e ficaram indisponíveis durante a escrita deste trabalho.

Apesar ausência da avaliação dos resultados de exatidão da medição, podemos mencionar que o equipamento realiza de modo satisfatório o trabalho para o qual foi desenvolvido conforme apresentado na seção 4.4.

As interfaces de usuário desenvolvidas trabalharam conforme o esperado, os serviços de [IOT](#) também se mostraram confiáveis e o microcontrolador ESP8266 executou suas funções com boa confiabilidade. Isso mostra que o sistema desenvolvido durante este trabalho poderia servir como base para o desenvolvimento de um produto comercial sem a necessidade de grandes mudanças.

Como melhorias futuras neste projeto, poderíamos citar a calibração correta do sistema utilizando as ferramentas adequadas, melhorias no sistema de identificação do dispositivo, para que os certificados e o *id* do dispositivo possam ser configurados externamente ao código programado no microcontrolador. Uma sugestão poderia ser a de realizar a configuração direta por meio de uma aplicação web hospedada no dispositivo que pode ter um modo de configuração como ponto e acesso ([AP](#)) e outro modo como cliente, assim como é realizado nos roteadores e outros dispositivos de internet. A confecção da placa de circuito impressa e do invólucro de polímero para sua enclausura também são melhoramentos que poderiam ser executados. Além disso, o chip AD7753 da *Analog Devices* permite a leitura de algumas falhas na rede como, por exemplo, afundamentos de tensão, picos de tensão entre outros que poderiam ser monitorados e gerar alertas nos próprios dispositivos e registros específicos no banco de dados.

Também poderia ter sido feito um estudo sobre os custos financeiros de todos estes serviços utilizados neste trabalho e comparar com serviços alternativos que podem ter um melhor custo-benefício para o cliente. Os serviços da [AWS](#) tem um custo atrelado à utilização, por isso foi utilizada uma taxa de uploads menor, causando uma perda de parte das informações de consumo do ambiente monitorado. Talvez se possam obter custos menores testando outros tipos de serviço que possam armazenar uma quantidade maior de dados de maneira menos onerosa, além de executar os cálculos na própria plataforma para restringir a quantidade de tráfego a ser transferida pela rede durante uma consulta.

## 5.1 DIFICULDADES ENCONTRADAS

Durante o projeto e o desenvolvimento deste trabalho, surgiram diversas dificuldades. A maior dificuldade pode ter sido inexperiência com as tecnologias utilizadas. O autor não possuía nenhum conhecimento relacionado à programação de páginas *web* ou com os dispositivos utilizados. O circuito integrado ADE7753 utilizado também exige bastante estudo, devido a sua complexidade. Os serviços da [AWS](#) possuem bastante documentação, mas isso não significa que seu aprendizado seja trivial. Todos estes aspectos exigiram muita dedicação do autor.

Uma dificuldade adicional que se apresentou durante a produção deste trabalho foi a repercussão da pandemia *COVID19*, na qual o estado de quarentena foi mantido durante todo o período de pesquisa e produção tanto da parte teórica quanto prática deste trabalho. A suspensão das atividades escolares e o isolamento causaram uma redução dos recursos disponíveis relacionados à estrutura física, disponibilidade de local de trabalho e equipamentos da instituição. Todas as tarefas relacionadas a este trabalho foram executadas no domicílio do autor, utilizando os equipamentos disponíveis. Isso pode ter prejudicado parcialmente os resultados, porém manteve significativo o aprendizado com o desenvolvimento deste trabalho.

## 6 REFERÊNCIAS

[1] - NICEGEAR. Datasheet: Current transformer. Modelo SCT013-020. Publicação Eletrônica, 2020. Disponível em: <https://nicegear.nz/obj/pdf/SCT-013-datasheet.pdf> Acesso em: 07 de abril de 2020.

[2] - GITHUB. Arduino core for the ESP32. 2018. Publicação eletrônica, 2020. Disponível em <https://github.com/espressif/arduino-esp32> Acesso em 09 de abril de 2020.

[3] - NODEMCU. Features. Publicação Eletrônica, 2020. Disponível em: [https://www.nodemcu.com/index\\_en.html](https://www.nodemcu.com/index_en.html) Acesso em: 08 de abril de 2020.

[4] - FILIPEFLOP. Módulo WiFi ESP8266 NodeMcu ESP-12. Publicação Eletrônica, 2020. Disponível em: <https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/> Acesso em 09 de abril de 2020.

[5] - ARDUINO. ESP8266 Arduino Core's documentation. Publicação Eletrônica, 2020. Disponível em: <https://arduino-esp8266.readthedocs.io/en/latest/> Acesso em: 08 de abril de 2020.

[6] - ANALOG DEVICES. Datasheet: ADE7753. Publicação Eletrônica, 2010. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7753.pdf> Acesso em: 08 de abril de 2020.

[7] - ESPRESSIF. Hardware ESP8266 Resources. Publicação Eletrônica, 2020. Disponível em: <https://www.espressif.com/en/products/hardware/esp8266ex/resources> Acesso em: 08 de abril de 2020.

[8] - ETT. A guide of Module OLED Model. OLED 1.3 I2C. Publicação Eletrônica, 2020. Disponível em: <http://www.ett.co.th/prod2017/OLED%201.3%20I2C%20ENG.pdf> Acesso em: 08 de abril de 2020.

[9] - AWS. What Is Amazon DynamoDB? Publicação Eletrônica, 2020. Disponível em: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html> Acesso em: 08 de abril de 2020.

[10] - AWS. PHP e DynamoDB. Publicação Eletrônica, 2020. Disponível em: [https://docs.aws.amazon.com/pt\\_br/amazondynamodb/latest/developerguide/GettingStarted.PHP.html](https://docs.aws.amazon.com/pt_br/amazondynamodb/latest/developerguide/GettingStarted.PHP.html) Acesso em: 08 de abril de 2020.

[11] - AWS. Installing the AWS SDK for PHP Version 3. Publicação Eletrônica, 2020. Disponível em: [https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/getting-started\\_installation.html](https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/getting-started_installation.html) Acesso em: 08 de abril de 2020.

[12] - AWS. Documentação do Amazon Elastic Compute Cloud. Publicação Eletrônica, 2020. Disponível em: <https://docs.aws.amazon.com/ec2/index.html> Acesso em: 08 de abril de 2020.

[13] - AWS. What Is AWS IoT? Publicação Eletrônica, 2020. Disponível em: <https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html> Acesso em: 08 de abril de 2020.

[14] - AWS. Connecting to Your Linux Instance from Windows Using PuTTY Publicação eletrônica, 2020. Disponível em: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html> Acesso em: 08 de abril de 2020.

[15] - AWS. Como autorizar tráfego de entrada em suas instâncias. Publicação Eletrônica, 2020. Disponível em: [https://docs.aws.amazon.com/pt\\_br/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html](https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html) Acesso em: 08 de abril de 2020.

[16] - 김동일. Connecting ESP8266 to AWS IoT Core using Arduino Mqtt. Publicação Eletrônica, 2019. Disponível em: <https://www.youtube.com/watch?v=4eF6bfiURN0> Acesso em: 08 de abril de 2020.

[17] - GITHUB. Biblioteca PubSubClient.h . Publicação Eletrônica, 2019. Disponível em: <https://github.com/knolleary/pubsubclient/blob/master/src/PubSubClient.h> Acesso em: 20 de abril de 2020.

[18] - GOOGLE. Google Charts – Glossary. Publicação Eletrônica, 2020. Disponível em: <https://developers.google.com/chart/glossary> Acesso em: 08 de abril de 2020.

[19] - APACHEFRIENDS. XAMPP Documentation. Publicação Eletrônica, 2020. Disponível em: <https://www.apachefriends.org/docs/> Acesso em: 08 de abril de 2020.

[20] - AMDS4. Comparação Shunts, Transformadores e Transdutores. Publicação Eletrônica, 2019. Disponível em: <https://blog.amds4.com.br/comparacao-shunts-transformadores-e-transdutores/> Acesso em: 08 de abril de 2020.

[21] - YANASHI. TRANSDUTOR com Isolamento Galvânico. Publicação Eletrônica, 2020. Disponível em: <http://yanashi.com.br/data/documents/Catalogo-Transdutor-com-Isolamento-Galvanico.pdf> Acesso em: 08 de abril de 2020.

[22] - IEFEP. Medições de Ondas Sinusoidais. Publicação Eletrônica, 2020. Disponível em: [https://elearning.iefp.pt/pluginfile.php/49293/mod\\_resource/content/0/Electricidade\\_-\\_Electronica/Electricidade/Onda%20Sinusoidal/Medicoes%20de%20uma%20Onda%20Sinusoidal.pdf](https://elearning.iefp.pt/pluginfile.php/49293/mod_resource/content/0/Electricidade_-_Electronica/Electricidade/Onda%20Sinusoidal/Medicoes%20de%20uma%20Onda%20Sinusoidal.pdf) Acesso em: 08 de abril de 2020.

[23] - AYUSH GEMINI. MQTT in Android. Publicação Eletrônica, 2019. Disponível em: <https://medium.com/@ayushgemini/mqtt-in-android-ed87b5278193> Acesso em: 09 de abril de 2020.

## APÊNDICE A – CÓDIGO FONTE

Todos os códigos de programação encontram-se disponíveis no seguinte endereço:

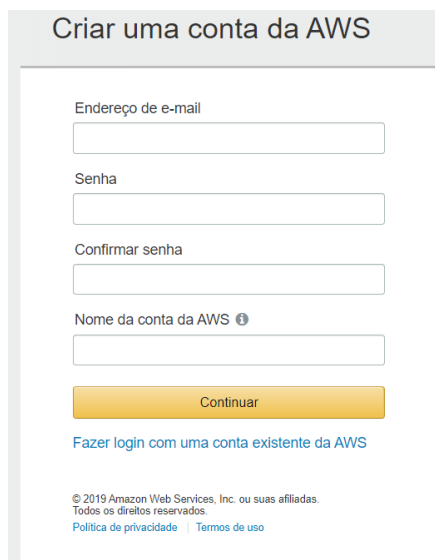
<https://github.com/cristianritter/EnergyMeter>

## APÊNDICE B – CONFIGURAÇÃO DOS SERVIÇOS AWS

Inicialmente é necessária a criação de uma conta na AWS por meio do link [aws.amazon.com/pt](https://aws.amazon.com/pt).



Preenchemos os dados solicitados

A imagem mostra o formulário de criação de uma conta da AWS. O formulário contém os seguintes campos: 'Endereço de e-mail', 'Senha', 'Confirmar senha' e 'Nome da conta da AWS'. Abaixo dos campos, há um botão amarelo rotulado 'Continuar'. Na base do formulário, há um link azul rotulado 'Fazer login com uma conta existente da AWS' e informações de copyright: '© 2019 Amazon Web Services, Inc. ou suas afiliadas. Todos os direitos reservados. Política de privacidade | Termos de uso'.

As contas da AWS incluem 12 meses de acesso ao nível gratuito que começa a contar a partir da inscrição contemplando mensalmente o uso de até 5Gb para armazenamento no banco de dados Dynamo DB e 250 mil mensagens publicadas ou recebidas pelo serviço IoT Core.

Após a conclusão de todos os passos de criação e verificação da conta podemos acessar o console de gerenciamento AWS.





A partir do console de gerenciamento acessamos a guia Serviços.

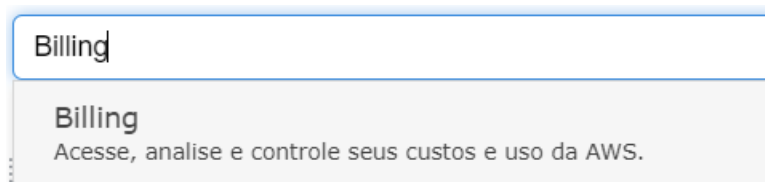


Na caixa de busca temos acesso aos serviços que iremos utilizar.



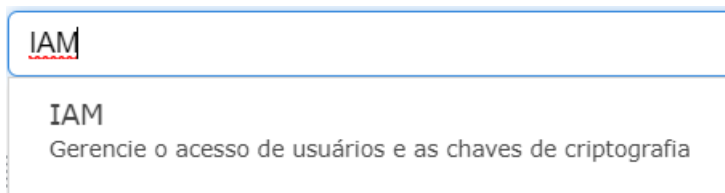
## Conta - Billing

Primeiramente iremos habilitar a conta adicionando uma forma de pagamento, para isso podemos fazer uma busca por Billing, acessar o serviço e a guia Métodos de Pagamento. Após a configuração de um método de pagamento e ativação do faturamento a conta está pronta para utilização. Mesmo a utilização gratuita exige o cadastro de uma forma de pagamento para verificação. É possível que hajam cobranças caso o uso do serviço exceda as cotas definidas para uso gratuito.

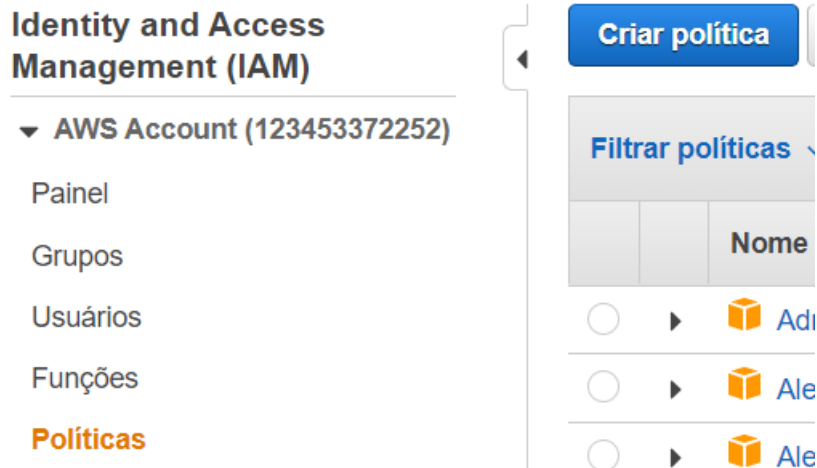


## Permissões - Policy

A próxima etapa necessária envolve a criação de uma política de acesso utilizando o serviço de identificação IAM que também pode ser encontrado utilizando a caixa de busca.



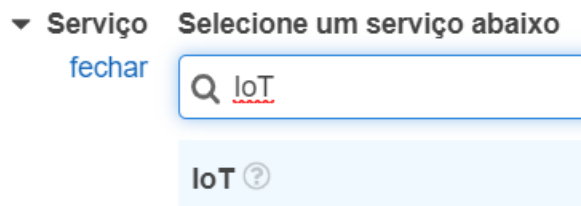
Iremos realizar primeiramente a criação de uma nova Política, para isso acessamos a guia Políticas e então clicamos no botão Criar Política.



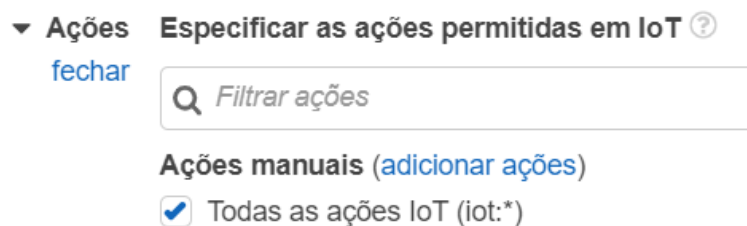
Faremos a seleção do tipo de serviço utilizando a opção Escolha um serviço.

▶ **Serviço** [Escolha um serviço](#)

Buscamos pelo serviço IoT.



Após a seleção do serviço IoT, configuramos as ações permitidas marcando a caixa chamada Todas as ações IoT (iot:\*)



Na guia recursos selecionamos a opção Todos os recursos.

- ▼ Recursos  Especifico  
fechar  Todos os recursos

Avançamos para a revisão selecionando Revisar Política

Cancelar

Revisar política

Adicionamos um nome e uma descrição à política e então selecionamos Criar política

Revisar política

Nome\*

Use caracteres alfanuméricos e "+, @, \_" Máximo de 128 caracteres.

Descrição

Máximo de 1000 caracteres. Use caracteres alfanuméricos e "+, @, \_"

Resumo

Q Filtro:

Serviço ▼	Nível de acesso	Recurso	Condição para solicitação
Permitir (1 de 201 serviços) <a href="#">Exibir restantes 200</a>			
IoT	Acesso total	Todos os recursos	Nenhum

\* Obrigatório

Cancelar

Anterior

Criar política

## Dispositivos - Things

Após a criação da política de acesso, acessamos novamente a caixa de busca e buscamos por IoT Core para a criação de uma “thing” que será fará as publicações.

  
  
Conecte dispositivos à nuvem

Dentro da página do serviço IoT Core, acessamos a guia Gerenciar.



## Monitoramento

Incluir

Gerenciar

Criamos um dispositivo utilizando o botão Criar.



Realizamos a criação de um dispositivo único utilizando o botão Criar uma única coisa.

Registrar uma única coisa do AWS IoT  
Criar uma coisa em seu registro



Preenchemos o campo com um nome definido para o dispositivo

CRIAR UMA COISA

### Adicionar seu dispositivo ao registro de coisas

Essa etapa cria uma entrada no registro de coisas e um thing shadow para o seu dispositivo.

**Nome**

Podemos deixar os outros campos do modo que estão e avançamos para a próxima etapa.



Nesta etapa vamos adicionar um certificado por meio da opção “Criar certificado”.

CRIAR UMA COISA

### Adicionar um certificado à sua coisa

ETAPA 2/3

Um certificado é usado para autenticar a conexão de seu dispositivo com o AWS IoT.

Criar certificado com um clique (recomendável)

Isso gerará um certificado, uma chave pública e uma privada usando a autoridade de certificado do AWS IoT.

[Criar certificado](#)

Após a criação do certificado, precisamos realizar o download do certificado, da chave pública, da chave privada e da CA raiz para o AWS IoT. Salvamos todos estes arquivos em uma pasta conhecida para uso posterior.

## Certificado criado!

Faça download desses arquivos e salve-os em um local seguro. É possível recuperar os certificados a qualquer momento, mas as chaves privadas e públicas não poderão ser recuperadas depois que você fechar esta página.

Para conectar um dispositivo, você precisa fazer download do seguinte:

Um certificado para essa coisa	9a2b6e1305.cert.pem	<a href="#">Fazer download</a>
Uma chave pública	9a2b6e1305.public.key	<a href="#">Fazer download</a>
Uma chave privada	9a2b6e1305.private.key	<a href="#">Fazer download</a>

Você também precisa fazer download de uma CA raiz para o AWS IoT:  
Uma CA raiz para o AWS IoT [Fazer download](#)

O download da CA raiz para o AWS IoT é feita de um modo um pouco diferente das demais. Ao clicar em Download somos direcionados pra outra página na qual devemos abrir o link chamado Amazon Root CA 1.

- + What Is AWS IoT?
- + Getting Started with AWS IoT
- + AWS IoT Rules Tutorials
- + Using the AWS IoT SDKs on a Raspberry Pi
- + AWS IoT Other Tutorials
- + Managing Devices with AWS IoT
- + Tagging Your AWS IoT Resources
- Security
  - Security in AWS IoT
  - Authentication
    - Server Authentication
      - + Client Authentication
      - + Custom Authentication

endpoints are fully interoperable with devices operating on symmetric endpoints any re-registration.

### CA Certificates for Server Authentication

Depending on which type of data endpoint you are using and which cipher suite authentication certificates are signed by one of the following root CA certificate

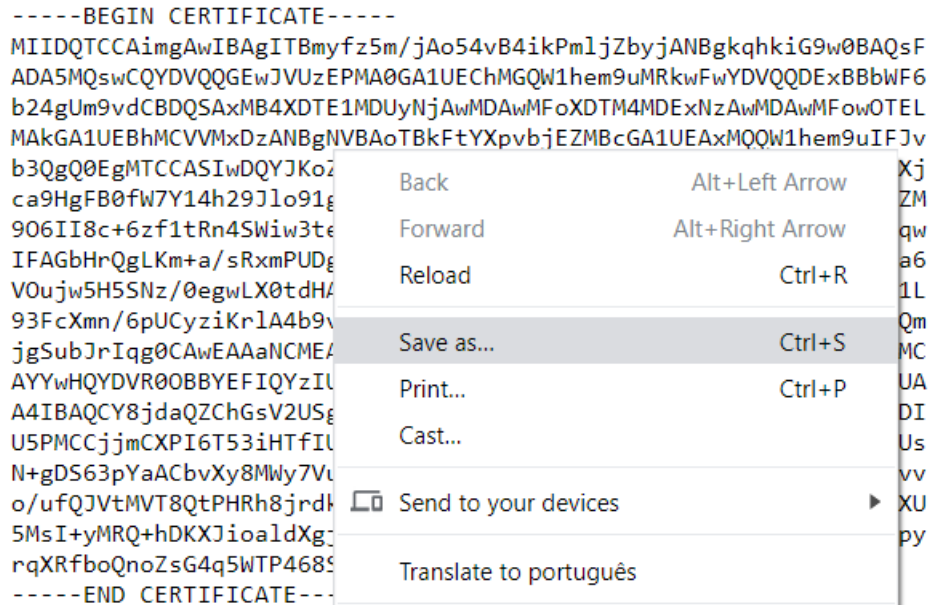
#### VeriSign Endpoints (legacy)

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

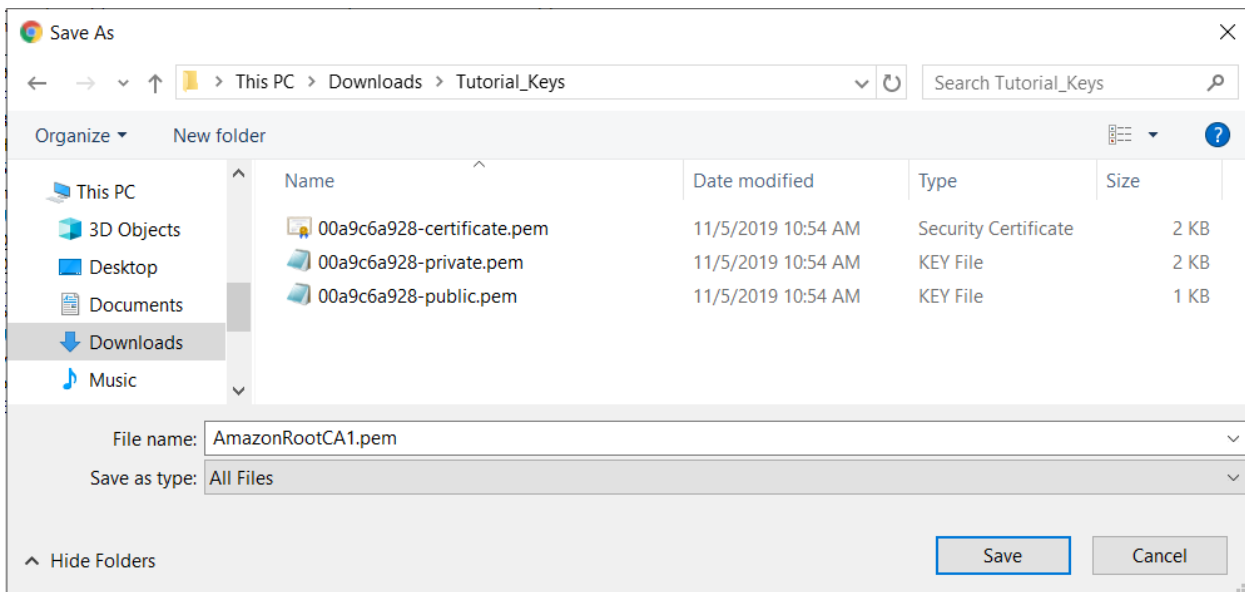
#### Amazon Trust Services Endpoints (preferred)

- RSA 2048 bit key: [Amazon Root CA 1](#).
- RSA 4096 bit key: Amazon Root CA 2. Reserved for future use.
- ECC 256 bit key: [Amazon Root CA 3](#).
- ECC 384 bit key: Amazon Root CA 4. Reserved for future use.

Este link abrirá um arquivo de certificado. Podemos salva-lo clicando com o botão direito do mouse e selecionando Save as...



Antes de salvar alteramos o tipo de arquivo para *All files* e salvamos com o nome original juntamente à pasta com as outras três chaves e certificados.



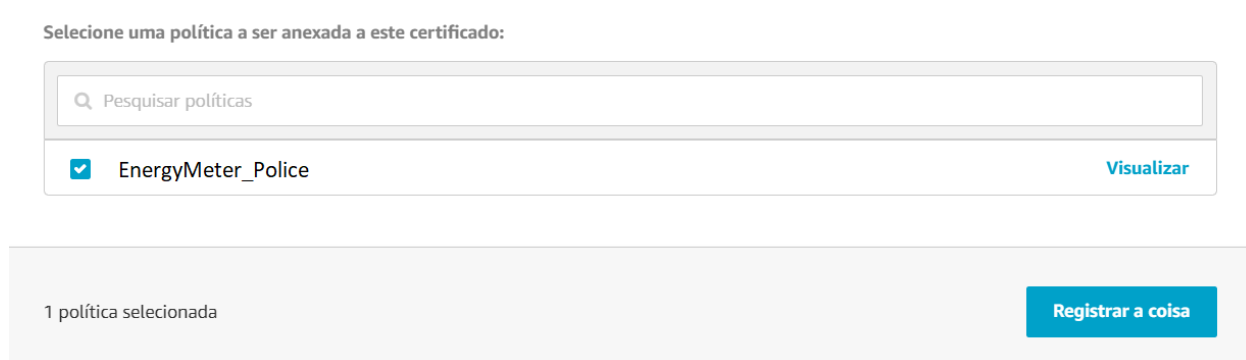
Após o download dos arquivos voltamos a página anterior e clicamos em Ativar.



Em seguida clicamos em Anexar uma política



Selecionamos a política criada anteriormente e clicamos em Registrar a coisa



Neste ponto nosso broker IOT está preparado para receber dados do nosso dispositivo.

## DynamoDB - Configurando Ações

Podemos configurar uma ação a ser executada pelo serviço IoT Core ao receber uma mensagem em determinado tópico. Para isso acessamos o serviço

▼ Serviços acessados r



E acessamos a guia Agir

Agir

Criamos uma nova regra clicando em



Completamos os dados solicitados

Crée uma regra para avaliar mensagens enviadas por suas coisas e determine o que fazer quando uma mensagem é recebida (por exemplo, armazenar dados em uma tabela do DynamoDB ou invocar uma função Lambda).

Nome

Regra\_de\_TESTE

Configuramos as instruções de consulta das regras

### Instrução da consulta da regra

SELECIONAR <Atributo> DE <Filtro do tópico> ONDE <Condição>. Por exemplo: SELECIONE temperatura ONDE temperatura > 50. Para saber mais, consulte [Referência de SQL do AWS IoT](#).

```
1 SELECT * FROM 'iot/topic'
```

Neste caso estamos selecionando todos os dados do tópico `iot/topic`.

Adicionamos uma ação e selecionamos o envio dos dados para uma tabela do DynamoDB como colunas separadas.





Ao final da página selecionamos

**Configurar ação**

E então criamos um novo recurso (tabela no Dynamodb para acomodar nossos dados)

A ação do DynamoDBv2 permite gravar todas ou parte de uma mensagem do MQTT em uma tabela do DynamoDB. Cada atributo gravado em uma coluna separada no banco de dados DynamoDB. Mensagens processadas por essa ação devem estar no formato

\*Nome da tabela

Escolher um recurso



**Criar um novo recurso**

Seremos direcionados para a plataforma do serviço Dynamo DB. Clicamos em Criar tabela

**Criar tabela**

Especificamos os detalhes da tabela que incluem nome, chave primária e chave de classificação.

O DynamoDB é um banco de dados sem esquema que requer somente o nome de uma tabela e a chave primária da tabela é constituída de um ou dois atributos que identificam itens, particionam os dados, os dados dentro da partição de maneira exclusiva.

Nome da tabela\*

tabela\_teste



Chave primária\*

Chave de partição

id\_dispositivo

String



Adicionar chave de classificação

timestamp

String



A chave de partição em conjunto com a chave de classificação, não podem se repetir. Logo temos que garantir que não existam dois registros idênticos sob o risco de serem substituídos.

Clicamos em Criar

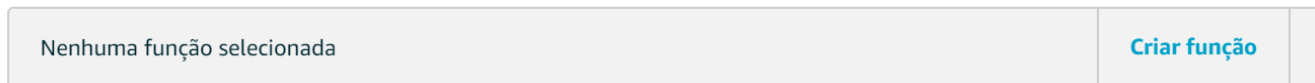


Estaremos com a tabela criada e podemos voltar ao console e selecionar AWS IoT Core.

Repetimos os passos anteriores e selecionamos a tabela criada.

Selecionamos a regra que nos dá a autorização para executar as ações, essa regra é criada em IAM. Se não houver uma regra podemos criar.

Escolha ou crie uma função para conceder ao AWS IoT para executar essa ação.



Clicamos em Adicionar ação



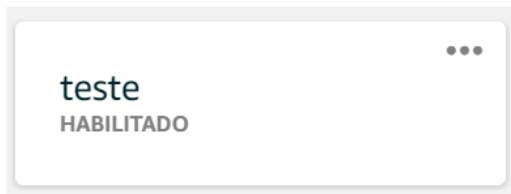
E então clicamos em



Se quiséssemos, poderíamos também criar uma regra para publicação em caso de erros, como, por exemplo uma que publicasse em um tópico aws/rules/erros.

Assim poderíamos assinar o tópico e verificar se algum erro está sendo gerado.

Temos tudo criado conforme exibido abaixo.



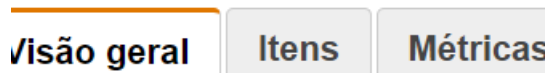
Agora vamos testar a regra.

Para isso vamos abrir nossa tabela no DynamoDB

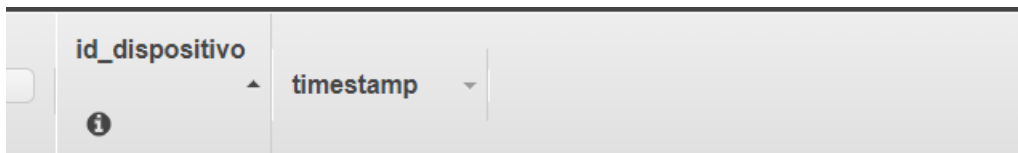
Acessamos o console AWS e então DynamoDB, após isso procuramos a guia Tabelas

## Tabelas

E Selecionamos a nossa tabela e a guia Itens



Podemos ver que não existem itens ainda.



Um item consiste em um ou mais atributos. Cada atributo consiste em um nome, um valor e um tipo. São aqueles que compõem a chave primária. [Mais informações](#)

Agora vamos publicar no tópico `iot/topic` utilizando a ferramenta de testes do módulo AWS IoT

Para isso acessamos o console AWS e procuramos por AWS IoT Core

E então acessamos a guia Teste

## Teste

Antes de publicar podemos assinar o tópico para verificar a mensagem que será enviada.

#### Assinar

Os dispositivos publicam mensagens MQTT em tópicos. Você pode usar esse cliente para assinar um tópico e receber essas mensagens.

#### Tópico de assinatura

Captura máxima de mensagens [?](#)

Agora vamos publicar, para isso clicamos em publicar em um tópico

### Publicar em um tópico

Definimos o tópico no qual publicar e a mensagem a ser publicada. Na mensagem enviaremos um Array JSON contendo os dados definidos na tabela como chave primária e de classificação assim como outros dados quaisquer. Precisamos manter o padrão de chave e valor, sendo que a chave é a primeira coluna e o valor será adicionado às linhas de dados. O valor de chaves primária e secundária precisam estar presentes e corretos para que não ocorram erros.

Assinar um tópico

Publicar em um tópico

iot/topic

×

#### Publicar

Determine um tópico e uma mensagem para publicar com QoS de 0.

```
1 {
2   "id_dispositivo": "algum id",
3   "timestamp": "algum timestamp",
4   "coluna": "valor"
5 }
```

Enviamos clicando em Publicar.

Podemos perceber que foi recebida uma mensagem no tópico

iot/topic

27 de mar de 2020 18:16:36

Ex

```
{
  "id_dispositivo": "algum id",
  "timestamp": "algum timestamp",
  "coluna": "valor"
}
```

Agora vamos verificar se os valores foram corretamente adicionados à database do Dynamodb.

Na página onde estão os itens acionamos o botão de atualizar itens



Podemos ver que os itens foram adicionados corretamente.

<input type="checkbox"/>	id_dispositivo ⓘ	timestamp	coluna
<input type="checkbox"/>	algum id	algum timestamp	valor

Se ocorrer algum erro, verifique as permissões em IAM, buscando pelas políticas das funções de lot. Tais políticas foram criadas juntamente à criação da 'thing'.

## Identity and Access Management (IAM)

### Painel

#### ▼ Gerenciamento de acesso

Grupos

Usuários

**Funções**

Políticas

teste

Serviço da AWS: iot

Nenhum

## Resumo

Excluir política

Nome de recurso da Amazon (ARN) da política: arn:aws:iam::123453372252:policy/service-role/aws-iot-role-dynamoPut\_107725447

Descrição

Permissões Utilização da política Versões da política Consultor de acesso

Resumo da política {} JSON Editar política

Q Filtro:

Serviço	Nível de acesso	Recurso	Condição para solicitação
Permitir (1 de 224 serviços) Exibir restantes 223			
DynamoDB	Limitado: Gravação	Todos os recursos	Nenhum

## Instância E2C - Criação

Partindo do console de gerenciamento da AWS, localize o serviço EC2.

### serviços da AWS



Acesse a guia Instances.

### INSTANCES

Instances

E acesse o botão Launch Instance.



Selecione o sistema operacional e clique em Select.

	<b>Ubuntu Server 16.04 LTS (HVM), SSD Volume Type</b> - ami-08cec7c429219e339 (64-bit x86) / ami-0bb2a062146998209 (64-bit Arm)	<b>Select</b>
Free tier eligible	Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes		

Selecione o tipo de instancia e clique em Review and Launch.

<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Será solicitada que seja criada uma chave de acesso. Crie uma chave ou utilize a mesma utilizada anteriormente. Ela será necessária para acessar a sessão da máquina posteriormente.

**Select an existing key pair or create a new key pair** ×

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair ▼

**Select a key pair**

EnergyMeter\_key ▼

I acknowledge that I have access to the selected private key file (EnergyMeter\_key.pem), and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**

Aguarde até que a instancia seja preparada e apresente o status conforme a imagem.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public IP
<input checked="" type="checkbox"/>	i-0d125a8d26338bb2d	t2.micro	us-east-2c	running	2/2 checks ...	None	ec2-

## Instância E2C - Acesso remoto

O acesso a instancia é realizado remotamente utilizando o protocolo SSH seguindo as instruções a seguir.

Clique em Conect



Uma janela com as informações de conexão será exibida.

**Connect to your instance** X

**Connection method**

- A standalone SSH client i
- Session Manager i
- EC2 Instance Connect (browser-based SSH connection) i

---

**To access your instance:**

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (EnergyMeter\_key.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:
 

```
chmod 400 EnergyMeter_key.pem
```
4. Connect to your instance using its Public DNS:
 

```
ssh -i EnergyMeter_key.pem ubuntu@ec2-53-223-33-108.us-east-2.compute.amazonaws.com
```

**Example:**

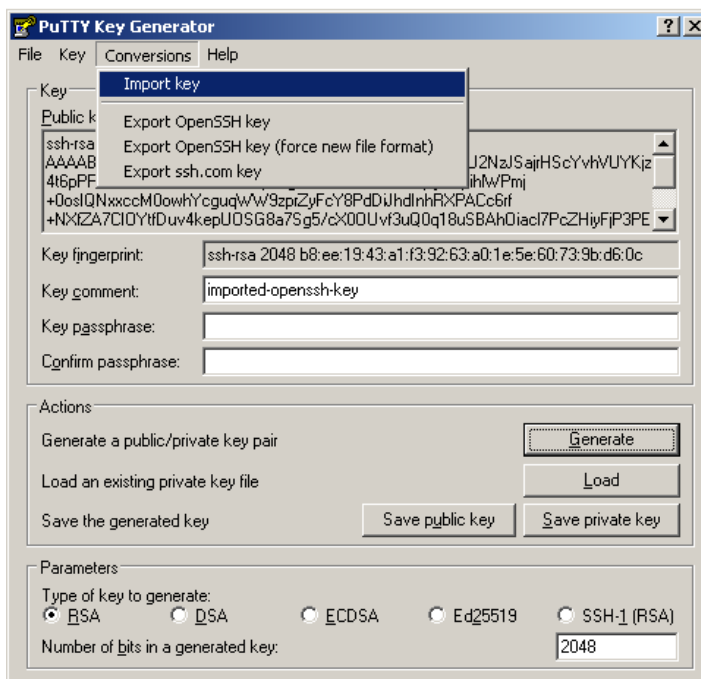
```
ssh -i "EnergyMeter_key.pem" ubuntu@ec2-53-223-33-108.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

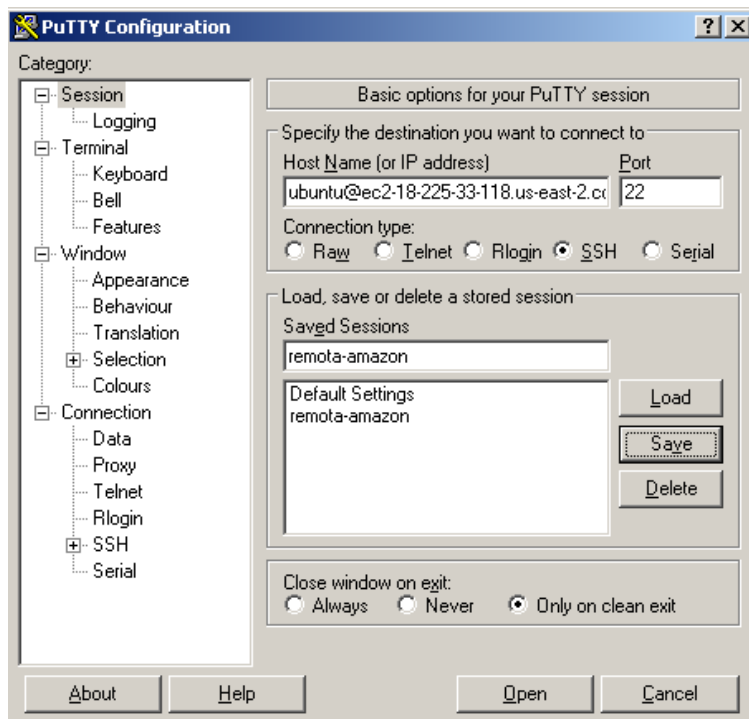
Close

Faça o download e utilize o aplicativo PuTTYgen para converter a chave anteriormente gerada com extensão .pem em .ppk. Importe a chave .pem e clique em Save private key.

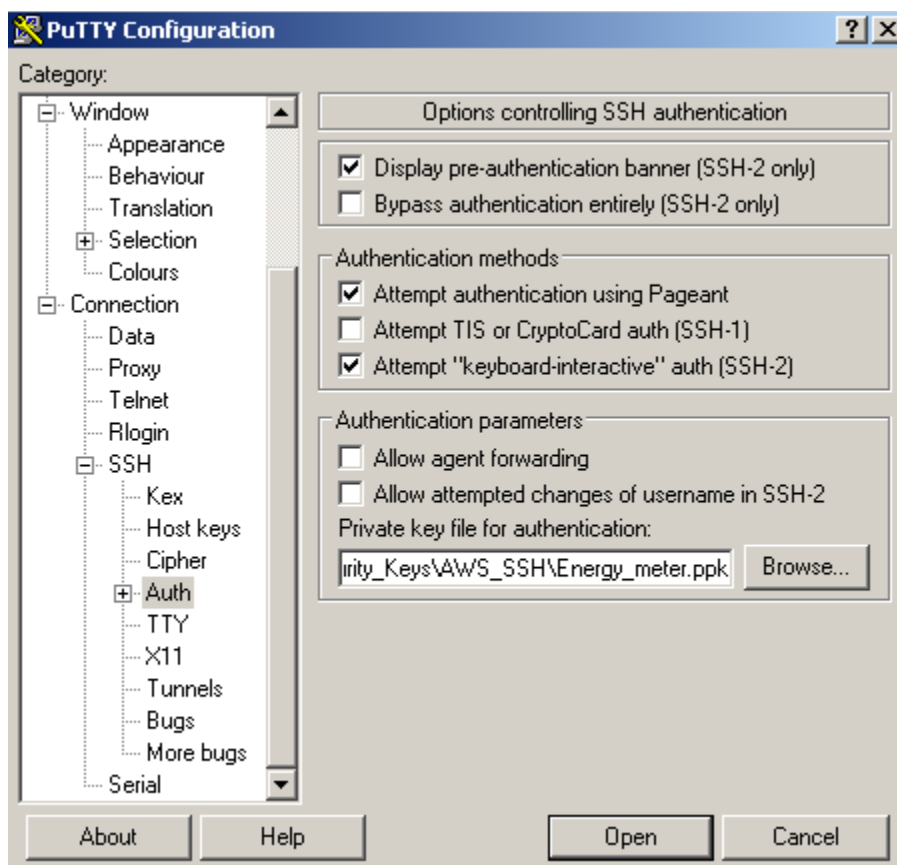


Na guia session adicione o seu endereço de dns personalizado conforme apresentado nas instruções de sua sessão.



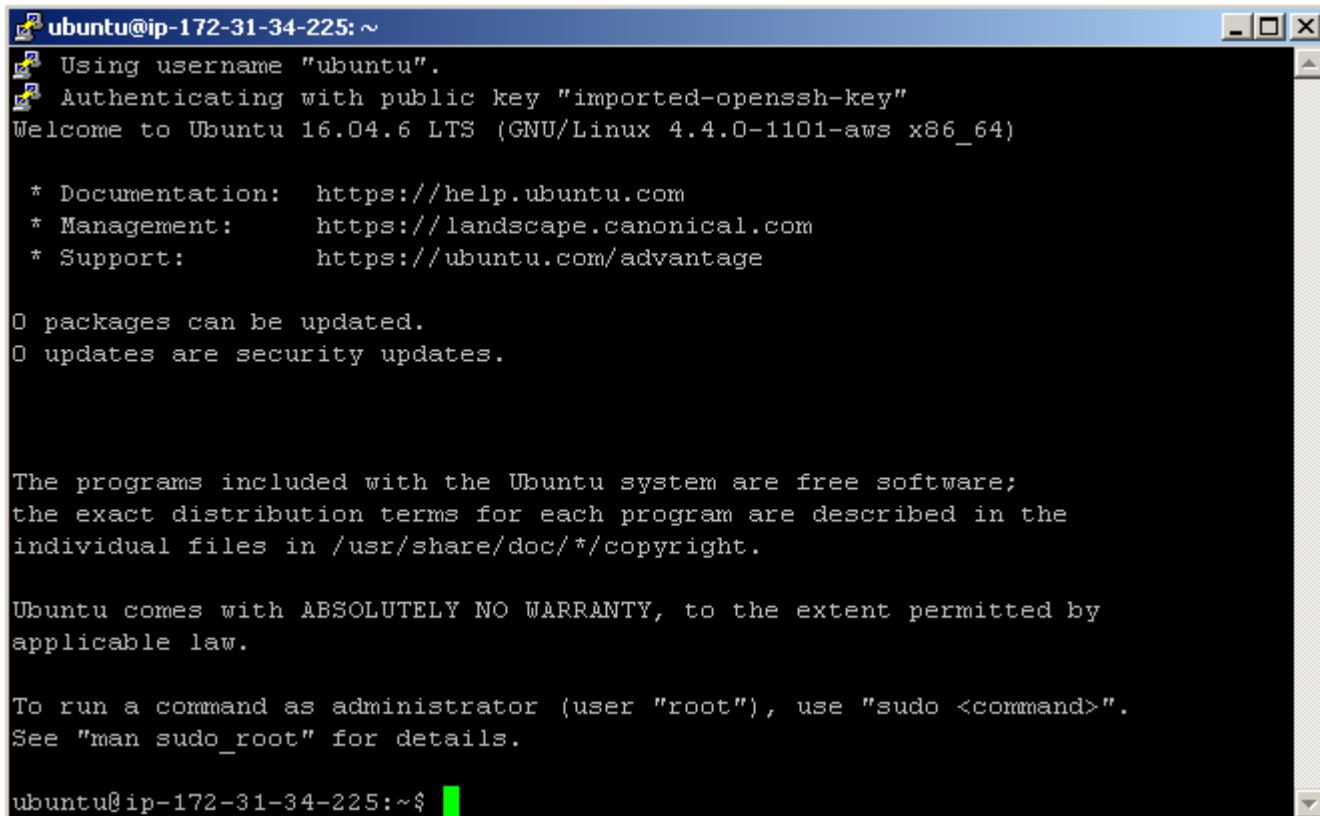


Após acesse a guia de Configurações SSH e selecione o certificado criado anteriormente.



Clique em Open.

Agora estamos conectados a sessão remota de nossa estação Linux de Cloud Computing AWS. Podemos dar continuidade instalando os serviços que utilizaremos.



```
ubuntu@ip-172-31-34-225: ~  
Using username "ubuntu".  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1101-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-34-225:~$
```

## Instância E2C - Instalação de serviços

Instalação do Xampp na estação remota.

Verifique o link com a versão atualizada do endereço <https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/>

<https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/7.4.3/xampp-linux-x64-7.4.3-0-installer.run/download>

Faça o download da última versão do XAMPP para o seu sistema e salve-o com o nome xampp-installer.run com o seguinte comando:

```
wget "https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/7.4.3/xampp-linux-x64-7.4.3-0-installer.run/download" -O xampp-installer.run
```

```
ubuntu@ip-172-31-34-225:~$ wget "https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/7.4.3/xampp-linux-x64-7.4.3-0-installer.run/download" -O xampp-installer.run
```

Torne o arquivo executável com o comando:

```
chmod +x xampp-installer.run
```

Inicie a instalação do XAMPP, com o comando;

```
sudo ./xampp-installer.run
```

Quando aparecer, siga os passos do assistente de instalação e configuração:

Responda Y para todos os questionamentos.

Para iniciar o Apache, use:

```
sudo /opt/lampp/xampp startapache
```

Precisamos ainda liberar o acesso para serviços HTTP na maquina criada. Para isso voltamos em instancias e selecionamos Security Groups.

Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time	Security Groups
18-225-33-118.us-e...	18.225.33.118	-	EnergyMeter_...	disabled	March 26, 2020 at 8:19:29 A...	<a href="#">launch-wizard-2</a>

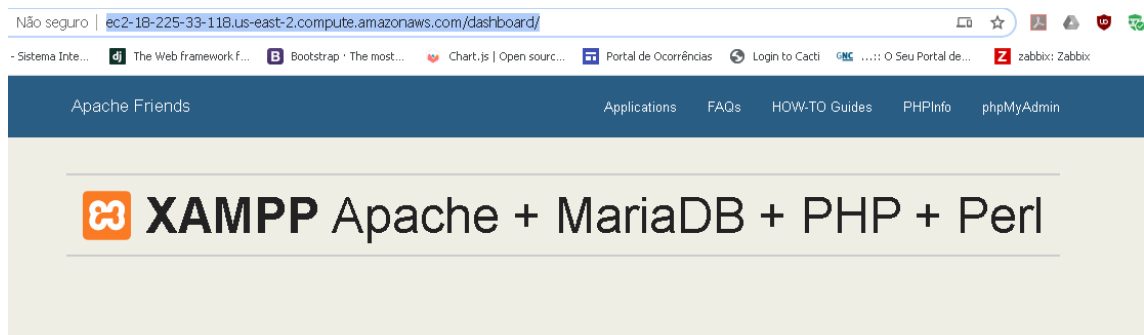
Editamos as regras de entrada

The screenshot shows the AWS Management Console interface for Security Groups. At the top, there's a search bar with the text "Filter security groups" and a search filter "search: sg-0c889336c9dc25bc2". Below the search bar is a table with columns: Security group ID, Security group name, VPC ID, and Owner. The table contains one entry: sg-0c889336c9dc25bc2, launch-wizard-2, vpc-cb0efca0, and launch-wizard-2 create... The context menu is open over the "launch-wizard-2" row, showing options: View details, Edit inbound rules (highlighted), Edit outbound rules, Manage tags, and Manage stale rules. There are also buttons for "Actions" and "Create security group" at the top right.

Adicionamos uma regra para acesso HTTP para todos os endereços IP.

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
HTTP	TCP	80	Custom <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	Delete
SSH	TCP	22	Custom <input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	Delete

Agora podemos acessar nosso servidor apache usando o DNS publico da estação. Estamos com a página padrão do XAMPP.



## Welcome to XAMPP for Linux 7.4.3

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP.

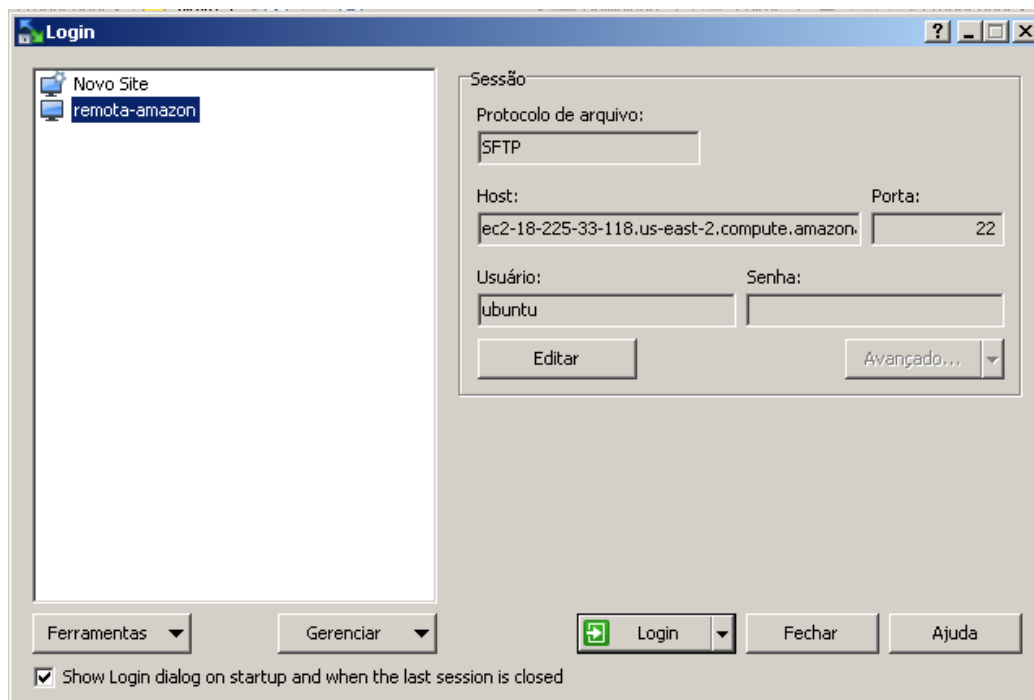
## Instância E2C - Servidor web

Vamos utilizar o WINSCP para transferir nossa própria pagina da web para o servidor.

Antes de continuar precisamos colocar uma senha de root em nosso servidor.

Para isso use os comandos su para acessar a super user da instância remota use o comando passwd para definir uma nova palavra passe.

Execute a instalação do WINSCP no computador local e importe as configurações do Putty

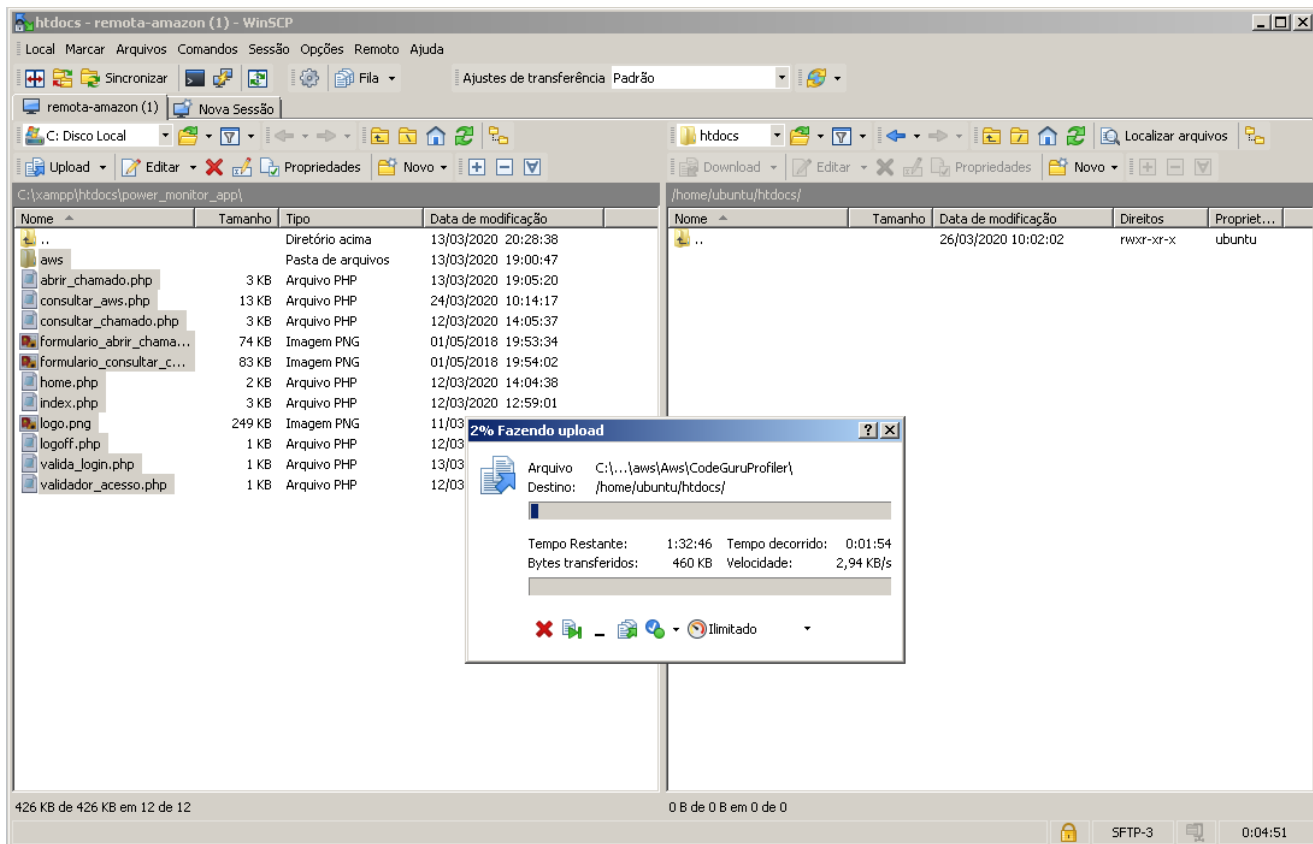


Agora temos acesso ao sistema de arquivos do nosso servidor.

Os arquivos do servidor apache ficam localizados na pasta `/opt/lampp/htdocs`. Vamos substituí-los pelos arquivos de nossa aplicação web.

Mas não podemos editar diretamente pelo WINSOCP, pois nosso acesso remoto não permite.

Vamos copiar nossos arquivos de projeto para uma pasta temporária no `home/ubuntu/htdocs/`



E após a conclusão iremos utilizar o Putty para mover essa pasta para /opt/lampp/htdocs utilizando o superusuário.

Primeiramente iremos renomear a pasta atual do XAMPPs para BKP utilizando os comandos:

```
cd /opt/lampp/htdocs
```

```
mv htdocs/ bkp/
```

Movemos o novo conteúdo para a pasta com o comando:

```
mv /home/ubuntu/htdocs/ htdocs/
```

Agora nosso servidor está funcionando corretamente e disponível através da internet.

Login