



UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

RAPHAEL DE SOUZA NUNES

**PROJETO DE UM SISTEMA DE ABASTECIMENTO PLUVIAL
BASEADO EM *IOT***

MANAUS/AM
2021

RAPHAEL DE SOUZA NUNES

**PROJETO DE UM SISTEMA DE ABASTECIMENTO PLUVIAL
BASEADO EM IOT**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas - UEA, unidade Escola Superior de Tecnologia - EST, como pré-requisito para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: **Prof. Dr. Israel Mazaira Morales**
Universidade do Estado do Amazonas

MANAUS/AM
2021

RESUMO

Com base na busca da produção de tecnologias para diminuir a degradação do meio ambiente e, unindo-se a isto, a procura pela implementação de *Smart Things*, dentro do âmbito do *IoT - Internet Of Things*, o projeto a ser desenvolvido visa a elaboração de um protótipo de uma **Cisterna Inteligente - CI** a qual contará com um sistema automático para coleta e distribuição de água da chuva, integrando as três áreas de um projeto *IoT*: *Hardware*, *Firmware* e *Software*. No segmento de *Firmware*, por meio do uso de microcontroladores e microprocessadores em integração com sensores, atuadores e a *Intranet*, será possível coletar dados e realizar ações pré-estabelecidas ou em tempo real oriundas de comandos do usuário, assim como receber atualizações para melhoria contínua do produto (*Firmware Update*). Na parte de *Software*, a utilização de novas tecnologias para modelagem de aplicações *mobile* e *desktop*, como *React Native* e *Electron.js*, garantirão a criação de interfaces amigáveis para rotinas de cadastro, atualização, monitoramento e controle dos processos da cisterna. Por fim, na parte de *Hardware*, serão desenvolvidos: placas de circuito impresso, com dispositivos *SMD - Surface Mounted Device*, possuindo todas as estruturas de alimentação, conversão de sinais, comunicação, entre outros; peças estruturais, com base em manufatura aditiva, para a implementação *in loco*. Edição de teste

Palavras-chave: *IoT - Internet of Things*. *Smart Things*. *Intranet*. Cisterna Inteligente. *Firmware*. *Hardware*. *Software*. *React Native*. *Electron.js*.

LISTA DE FIGURAS

Figura 5.1 – Corte Esquemático do Sistema de Aproveitamento Proposto.	11
Figura 5.2 – Simulação elaborada no <i>Software Proteus</i>	12
Figura 5.3 – Microcontrolador <i>Philips 80C552</i> acoplado ao kit <i>CW 552</i>	12
Figura 5.4 – Visão geral do projeto.	13
Figura 5.5 – Visualização da interface desenvolvida no <i>Node-RED</i> para receber entradas do usuário e mostrar dados dos sensores.	14
Figura 5.6 – Configuração dos dispositivos	15
Figura 5.7 – Comparação entre o <i>layout 3D</i> e o dispositivo físico montado	16
Figura 5.8 – Esquema para aplicação do protocolo <i>ZiWi</i>	17
Figura 6.1 – Esquema de demonstração de uma cisterna no subsolo	18
Figura 6.2 – Esquema básico do projeto.	20
Figura 6.3 – <i>ESP8266EX</i>	21
Figura 6.4 – Alguns exemplos de módulos com base no <i>ESP8266EX</i>	21
Figura 6.5 – Vista superior do <i>ESP-12S</i>	21
Figura 6.6 – Vista superior do <i>ESP32</i>	22
Figura 6.7 – Vista superior da <i>Raspberry Pi Zero W</i>	23
Figura 6.8 – Vista superior da fonte	23
Figura 6.9 – Bomba d'água <i>DC</i>	24
Figura 6.10–Representação gráfica do sensor <i>ACS712</i>	24
Figura 6.11–Aplicação típica do <i>ACS712</i>	25
Figura 6.12–Diagrama de ativação com partida lenta	25
Figura 6.13–Transistor <i>IRF1404</i>	26
Figura 6.14–Sensor ultrassônico <i>JSTN-SR04T</i> e módulo de conversão	26
Figura 6.15–Válvula solenoide	27
Figura 6.16–Chave de nível <i>RF-0H21D</i>	28
Figura 6.17–Gerenciador de placas com suporte oficial às placas <i>ESP32</i> e <i>ESP8266</i>	29
Figura 6.18–Fluxo de trabalho geral <i>Yocto Project</i>	30
Figura 6.19–Exemplos de <i>Templates</i> produzidos com <i>React Native</i>	31
Figura 6.20–Aplicações que utilizam <i>Electron</i>	31
Figura 7.1 – Visão geral do quadro <i>Kanban</i> criado na ferramenta <i>Trello</i>	33
Figura 7.2 – Visão geral repositório criado no <i>Github</i>	33

LISTA DE SIGLAS

IoT	Internet Of Things
PCB	Printed Circuit Board
SMCC	Sistema de Medição e Controle da Cisterna
SMCCD	Sistema de Medição e Controle da Caixa D'água
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
GPIO	General Purpose Input Output
HAS	Home Automation System
DC	Direct Current
AC	Alternate Current
PWM	Pulse-Width Modulation
NF	Normalmente Fechada
IDE	Integrated Development Environment
CLI	Command Line Interface
GUI	Graphic User Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheet

SUMÁRIO

1	INTRODUÇÃO	7
2	PROBLEMÁTICA	8
3	OBJETIVOS	9
3.1	Objetivo geral	9
3.2	Objetivos específicos	9
4	HIPÓTESE E JUSTIFICATIVA	10
4.1	Hipótese	10
4.2	Justificativa	10
5	TRABALHOS RELACIONADOS	11
5.1	Automação e Controle em Sistema de Aproveitamento de Água da Chuva para Fins Não Potáveis	11
5.2	Automação residencial Usando protocolo <i>MQTT</i> , <i>NODE-RED</i> e <i>Mosquitto Broker</i> com <i>ESP32</i> e <i>ESP8266</i>	13
5.3	Desenvolvimento de plataformas embarcadas aplicadas a implementação de <i>Smart Buildings</i> com base no <i>framework SmartLVGrid</i> .	15
5.4	Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on <i>MQTT</i> and <i>ZigBee-WiFi</i> Sensor Nodes	16
6	FUNDAMENTAÇÃO TEÓRICA	18
6.1	Análise de funcionamento e aplicação de uma cisterna	18
6.2	Elementos de <i>Hardware</i>	20
6.2.1	Os microcontroladores <i>ESP8266</i> e <i>ESP32</i>	20
6.2.2	A Raspberry Pi Zero W	22
6.2.3	A fonte de alimentação DC	23
6.2.4	A motobomba ou bomba d'água	23
6.2.5	O sensor de corrente	24
6.2.6	Os elementos de ativação e desativação	25
6.2.7	O sensor de distância ultrassônico	26
6.2.8	A válvula solenoide	27
6.2.9	A chave de nível tipo boia	27
6.3	Elementos de <i>Firmware</i>	28
6.3.1	O Ambiente de Desenvolvimento Integrado <i>Arduino</i>	28

6.3.2	Modelagem de um <i>SO</i> embarcado com <i>Yocto Project</i>	29
6.4	Elementos de <i>Software</i>	30
6.4.1	O <i>framework React Native</i>	30
6.4.2	O <i>framework Electron</i>	31
7	METODOLOGIA	32
7.1	Levantamento do referencial bibliográfico e capacitação	33
7.2	Desenvolvimento dos elementos de <i>Hardware</i>	33
7.3	Desenvolvimento dos elementos de <i>Firmware</i>	34
7.4	Desenvolvimento dos elementos de <i>Software</i>	34
8	CRONOGRAMA	35
	REFERÊNCIAS	37

1 INTRODUÇÃO

A água é um recurso básico para a sustentação humana em um ambiente e ao longo do tempo várias civilizações evoluíram e padeceram em função de sua relação de uso com este recurso. Nos últimos anos, crises relacionadas ao abastecimento e à qualidade da água potável têm sido observadas em todo o Globo. Tomando o Brasil como exemplo, percebe-se uma distribuição desigual: No norte do país há grandes reservas de água, porém nas regiões Nordeste e Sudeste há problemas de escassez e poluição dos rios.

A região norte, possuindo a maior reserva de água potável do Brasil é, segundo a Globo, a região que possui índices mais altos de desperdício. O desperdício pode ser encontrado no ambiente doméstico, falta de orientação e nas várias etapas do processo: coleta, armazenamento, processamento e finalidade final do recurso.

A aplicação de uma cisterna automatizada garante uma supervisão do nível de água constantemente assim como o controle/acionamento de bombas para alimentação de tanques ou caixas d'água para tarefas específicas trazendo comodidade e fomentando os motivos de aplicação. O aproveitamento pode contribuir muito também para a redução da incidência de inundações nas grandes cidades decorrentes do volume excessivo de chuvas em determinadas épocas do ano. Isso porque o excesso de áreas impermeabilizadas nos ambientes urbanos, é um dos fatores preponderantes para ocorrência de enchentes, pois a água da chuva não tem como infiltrar no solo e chegar ao lençol freático, acumulando-se nas ruas e transbordando nos rios. Assim, a implantação de sistemas de reaproveitamento dessas águas iria reduzir sensivelmente esse problema.

2 PROBLEMÁTICA

Segundo a Gartner, Inc BizMeet desde 2017 existem mais objetos conectados à Internet do que as 7 bilhões de pessoas no mundo. Isso demonstra uma crescente busca pela obtenção e controle das mais diversas tarefas. Os objetos e dispositivos estão adquirindo funcionalidades as quais se identificam com os ramos da *IoT*, tornando-se possível a concepção de diversas melhorias no funcionamento, através da obtenção e distribuição de dados. A utilização da **Internet das Coisas** está tão conectada ao nosso cotidiano que ocasiona o nascimento de tecnologias necessariamente conectadas, como *Smart Home* e *Smart Buildings*.

Diante disso, em ambientes como na região Amazônica, onde há altos índices de precipitação, a implantação de cisternas para utilização da água da chuva se torna bastante viável. Sabe-se que a água da chuva não é própria para o consumo e para o preparo de alimentos, porém a mesma pode ser utilizada em afazeres como limpeza de locais e também em descargas de banheiros, onde há o maior nível de desperdício.

Em virtude do que foi mencionado, a aplicação de uma cisterna se torna algo viável para a economia de água ajudando na conservação do meio ambiente, porém, a automação do processo de coleta e distribuição trará reais vantagens quando comparada à um processo manual? Quais serão os problemas recorrentes ao se implementar tal sistema? Será possível uma aplicação em diversos contextos e localidades? O auxílio de tecnologias como o *WIFI* unido à *IoT* fará com que a aplicação seja mais vantajosa? A utilização da água coletada por meio de uma cisterna automatizada, onde haverá gastos, como consumo energia elétrica, aplicação de sensoriamento, motobombas, sistemas hidráulicos, entre outros, trará benefícios financeiros quando comparado à utilização de água através, unicamente, da rede tradicional?

3 OBJETIVOS

3.1 Objetivo geral

O objetivo desse trabalho de conclusão de curso é a elaboração e aplicação de um sistema automatizado de coleta, armazenamento e distribuição de água da chuva com base em uma cisterna pluvial. Realizando medições de volume, através de sensores, acionamento de válvulas solenoides e motobombas por meio de comandos via interfaces desktop e mobile, o projeto também contará com a elaboração de placas de circuito impresso - *PCB's*, o uso de microcontroladores e microprocessadores em conjunto com diretrizes de Internet das Coisas - *IoT*, assim como o uso de *Frameworks* atuais baseados em *Javascript* para criação de interfaces Homem-Máquina.

3.2 Objetivos específicos

- (a) Elaborar um módulo denominado **SMCC - Sistema de Medição e Controle da Cisterna** para aplicação no reservatório principal, realizando medições de nível, acionamento de *motobomba*, direcionamento do fluxo de água e possuindo conexão com outros dispositivos via *Wi-Fi*.
- (b) Elaborar um módulo denominado **SMCCD - Sistema de Medição e Controle da Caixa D'água** para aplicação no reservatório auxiliar, realizando medições de nível, direcionamento do fluxo de água e possuindo conexão com outros dispositivos via *Wi-Fi*.
- (c) Elaborar um aplicação *Android Mobile* (denominada **SCCP APP**) e *Desktop* (denominada **SCCP DESKTOP**) para executar as ações: ativação e desativação de uma bomba d'água, direcionamento do fluxo de água, visualização de dados provenientes de sensores e a possibilidade de definir as condições que executem rotinas de acionamento automático;
- (d) Elaborar um sistema operacional embarcado conciso (baseado em *kernel Linux*) aplicando-o a um microprocessador. O dispositivo deve possuir conexão com a *Intranet* e servir como uma central de controle e armazenamento de dados, bem como ser hospedeiro do serviço de *Broker MQTT*;
- (e) Incluir no sistema, rotinas de calibração para configurar os níveis e estados de operação em cisternas e caixas d'água de tamanhos e formatos diferentes.
- (f) Organizar um repositório *online* para realização de futuros *updates* ou *upgrades* do *firmware* embarcado;
- (g) Planejar a implementação *in loco*, analisando todos os fatores estruturais. Adquirir ou confeccionar instrumentos para proteção e suporte dos itens citados nos quesitos (a) e (b).

4 HIPÓTESE E JUSTIFICATIVA

4.1 Hipótese

A inclusão da *IoT* nos mais diversos processos nos traz múltiplos benefícios, como: obter maiores informações e atuar sobre tais processos. Automatizar tarefas que são, muitas vezes repetitivas ou mesmo insalubres faz com que as tecnologias denominadas *Smart* se tornem cada vez mais um foco de pesquisa e desenvolvimento.

No âmbito do meio ambiente, a implantação de uma cisterna garante um menor desperdício de água, utilizando-a para tarefas específicas assim como uma economia no consumo. Porém a implantação de uma cisterna como não atrai indivíduos pois existe a necessidade de trabalho braçal assim como a constante análise da quantidade de água disponível.

Esse TCC busca elaborar uma cisterna autônoma trazendo benefícios, primeiramente financeiros, e também ecológicos assim como benefícios da comodidade para a retirada e verificação dos níveis de água, tornando a ideia da aplicação mais atraente para possíveis clientes, em caso do projeto se tornar um produto. Com a redução significativa de consumo de água, será possível que o investimento de construção e aplicação retorne como economia.

4.2 Justificativa

Buscando estar de acordo com a convergência de produtos e processos conectados em rede, percebeu-se a necessidade e oportunidade da utilização de conceitos e tecnologias emergentes nos contextos atuais, como *Smart Things*. A aplicação de serviços que unam a busca pela preservação ambiental e conceitos citados anteriormente se destacam por possibilitar uma gama de vantagens em qualquer processo.

O presente trabalho se justifica pelos altos índices de precipitação na região do Estado do Amazonas, onde será aplicado. A coleta de água da chuva acarretará, além da diminuição do consumo de água (trazendo economia para o usuário e ajudando na preservação ambiental), em uma distribuição mais inteligente da água não potável: que pode ser utilizada para limpezas, descargas e até irrigações; e em uma motivação para investimentos futuros reduzindo, ainda mais, os gastos de implementação.

O projeto trará motivações e investidores para aplicação de novas pesquisas focadas em contextos diferentes dentro do estado, onde há: **I.** lugares que não possuem estruturas de rede ou que demandam comunicação à grandes distâncias: utilização de tecnologias como *LoRaWan*, *SigFox* e *Zigbee*; **II.** lugares que não possuem energia elétrica: união dos conceitos desse trabalho com fontes renováveis, como alimentação fotovoltaica.

5 TRABALHOS RELACIONADOS

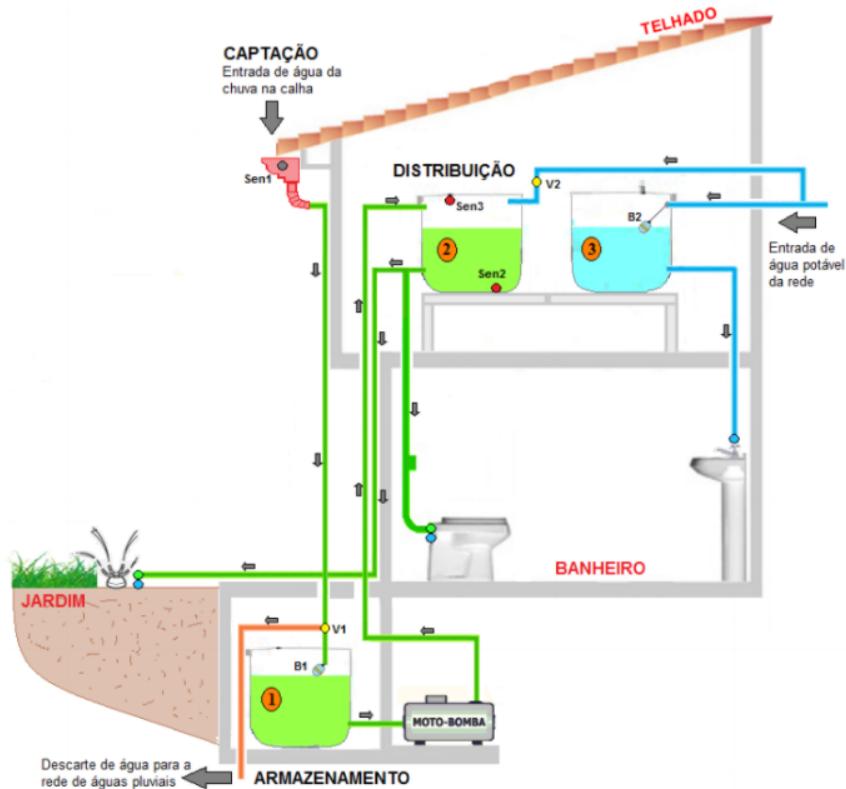
5.1 Automação e Controle em Sistema de Aproveitamento de Água da Chuva para Fins Não Potáveis

Autora: Lilia Rodrigues Lucas Magalhães

Este trabalho de conclusão de curso realizado por Lilia Rodrigues Lucas Magalhães pelo Centro Universitário de Brasília - UniCEUB apresenta uma protosta muito semelhante ao trabalho a ser desenvolvido.

A ideia deste projeto foi demonstrar o funcionamento automatizado de um sistema (representado na Figura 5.1) de aproveitamento de água de chuva por meio da elaboração de um protótipo composto pela maquete de um banheiro, confeccionada em madeira, juntamente com três reservatórios em acrílico. Esse cenário representa o sistema de distribuição de água de chuva dentro de uma residência e destina-se à utilização em descarga sanitária.

Figura 5.1 – Corte Esquemático do Sistema de Aproveitamento Proposto.



O projeto, implementado fisicamente e virtualmente no *software Proteus* (Figura 5.2), foi baseado para uma aplicação sem a conexão com redes remotas e utilizando componentes eletrônicos mais simples, como portas analógicas. O microcontrolador uti-

lizado foi o *Philips 80C552* acoplado ao Kit *CW 552*, considerado defasado atualmente. Esse trabalho fomentou a ideia da utilização de válvulas solenoides para direcionamento do fluxo de água, assim como métodos para medição de nível e uma abordagem sobre as rotinas de funcionamento do microcontrolador.

Figura 5.2 – Simulação elaborada no *Software Proteus*.

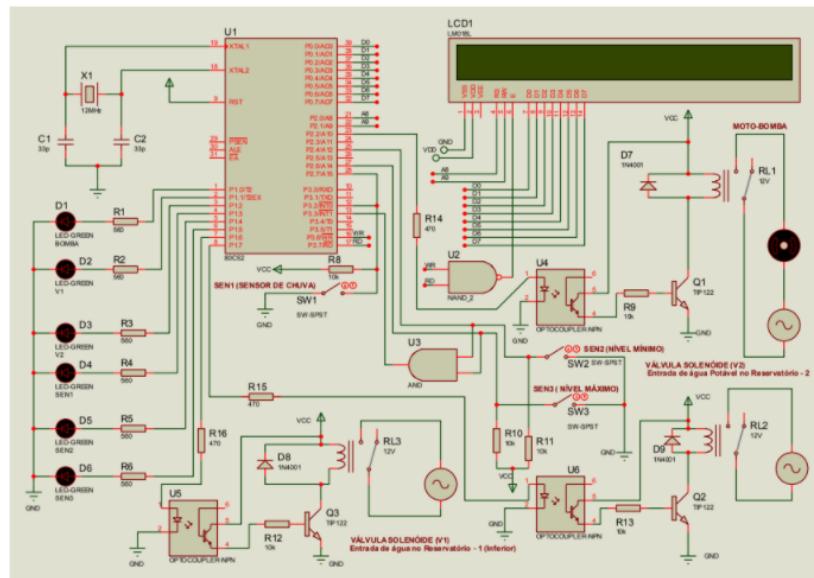


Figura 5.3 – Microcontrolador *Philips 80C552* acoplado ao kit *CW 552*.



5.2 Automação residencial Usando protocolo *MQTT*, *NODE-RED* e *Mosquitto Broker* com *ESP32* e *ESP8266*

Autor: Victor Ferreira Martins

Este trabalho de conclusão de curso desenvolvido por Victor Ferreira Martins teve o objetivo de propor uma solução de automação residencial que fosse capaz de integrar três diferentes sistemas em uma casa: monitoramento de temperatura, alarmes de segurança e acionamento de iluminação e tomadas. O objetivo dessa integração foi ajudar a alcançar três dos grandes objetivos da automação residencial: conforto, segurança e economia de energia. Como características importantes destacam-se: a utilização do protocolo *MQTT* para comunicação através do *Mosquitto Broker*, assim como o uso da ferramenta de programação visual *Node-RED* e dos microcontroladores *ESP32* e *ESP8266*.

Em relação ao trabalho pode-se abstrair os seguintes dados:

1. A utilização do protocolo *MQTT*, por meio do *Broker Mosquitto* em conjunto com os microcontroladores da *ESPRESSIF*;
2. A utilização de diversos sensores, validando que a conexão e transferência de dados através do protocolo *MQTT*;
3. Uma abordagem para a organização dos tópicos *MQTT* (Tabela 5.1), definindo dispositivos de publicação (*publishers*) e de inscrição (*subscribes*);
4. A utilização do *Node-RED* como agente de interfaceamento do projeto (Figura 5.1).

Figura 5.4 – Visão geral do projeto.

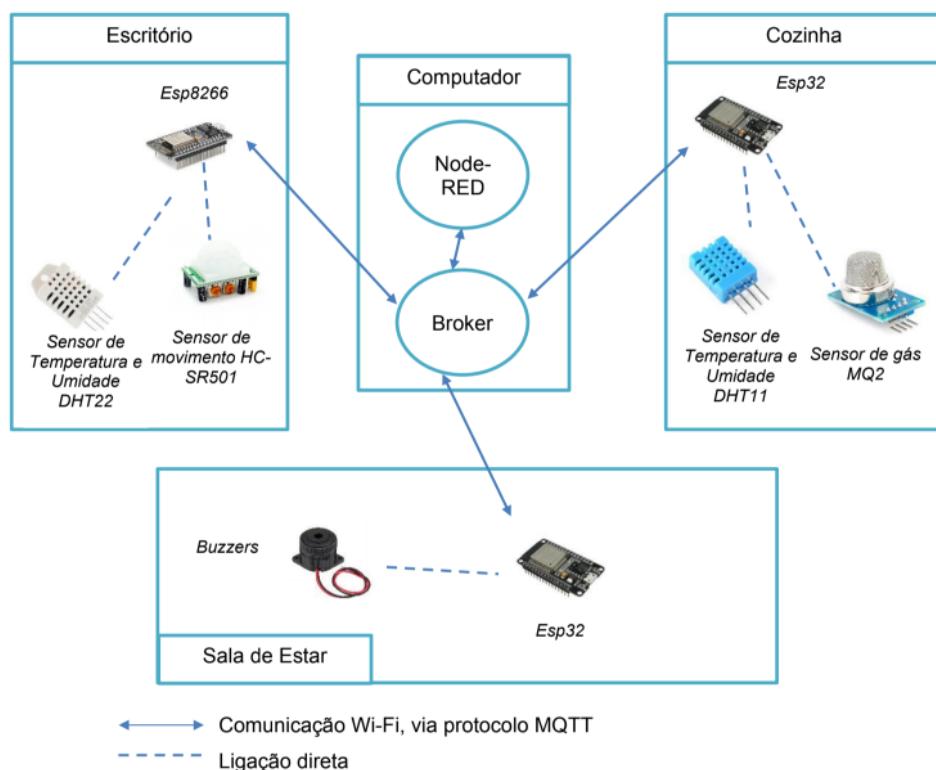


Figura 5.5 – Visualização da interface desenvolvida no *Node-RED* para receber entradas do usuário e mostrar dados dos sensores.

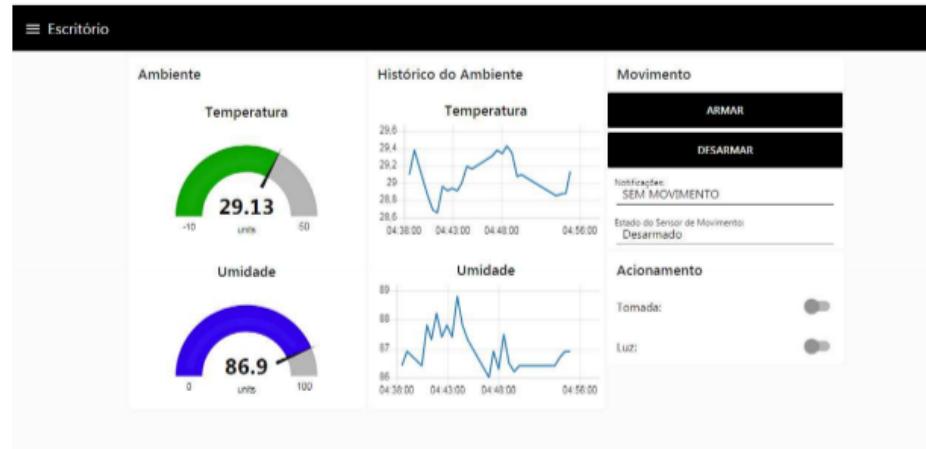


Tabela 5.1 – Organização dos tópicos *MQTT*.

Dispositivo	Inscrição	Publicação
(Escritório)	casa/escritório/esp8266/movimento	casa/escritório/esp8266/temperatura casa/escritório/esp8266/umidade
	casa/escritório/esp8266/tomada	casa/escritório/esp8266/movimento/ estado
	casa/escritório/esp8266/luz	casa/escritório/esp8266/movimento/ notificação
	casa/esp32/cozinha/fumaça	casa/cozinha/esp32/temperatura
ESP32 (Cozinha)	casa/cozinha/esp32/ tomada	casa/cozinha/esp32/umidade
	casa/cozinha/esp32/luz	casa/cozinha/esp32/fumaça/estado casa/cozinha/esp32/fumaça/notificação
	casa/escritório/esp8266/notificação	-
ESP32 (Sala de estar)	casa/cozinha/esp32/notificação	-

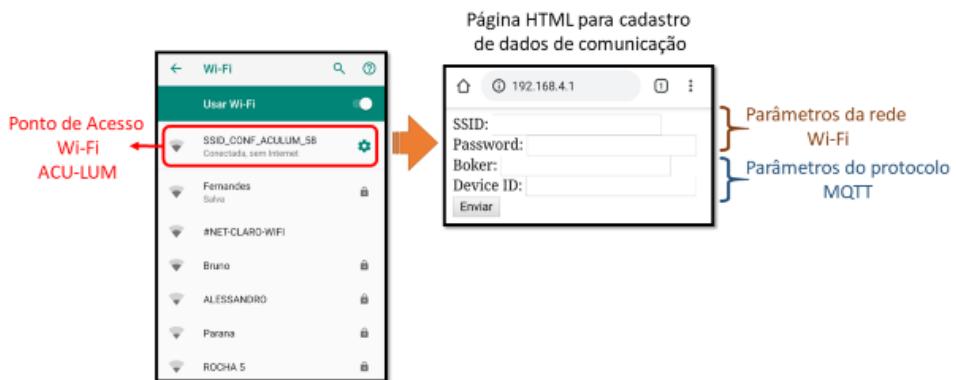
5.3 Desenvolvimento de plataformas embarcadas aplicadas a implementação de *Smart Buildings* com base no framework *SmartLVGrid*

Autor: Rubens de Andrade Fernandes

Este trabalho de conclusão de curso desenvolvido por Rubens de Andrade Fernandes pela Universidade do Estado do Amazonas concentrou-se no desenvolvimento de algorítimos para *software* embarcado e dispositivos de *hardware*, associados a plataformas microcontroladas e microprocessadas, com o objetivo de realizar a convergência *smart building* em sistemas de iluminação e medição de energia elétrica sem recursos de automação, comunicação e controle.

Os pontos relevantes deste trabalho em relação à para implementação deste TCC estão na criação de rotinas de cadastro geradas exclusivamente pelos microcontroladores *ESP32* utilizados (Figura 5.6).

Figura 5.6 – Configuração dos dispositivos



Essa rotina de cadastro faz com que o microcontrolador forneça um ponto de acesso local possibilitando qualquer o acesso de qualquer dispositivo. Ao realizar uma conexão através de um computador ou *smartphone*, por exemplo, o usuário pode informar dados de cadastro como para o dispositivo operar em rede.

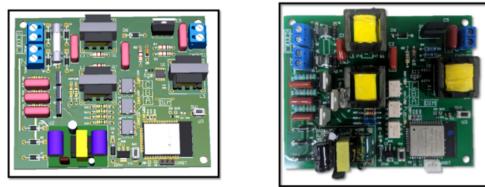
O segundo ponto que se pode destacar desse trabalho foi a criação de bibliotecas que auxiliam na operação do microcontrolador como demonstrado na tabela abaixo.

Tabela 5.2 – Bibliotecas implementadas para elaboração do *firmware*

Bibliotecas	Descrição
DriverAcuLum	Utilizada para implementar os métodos referentes as DRFs a serem executadas
DriverMQTT	Utilizada para garantir confiabilidade e segurança no uso do protocolo MQTT
DriverWIFI	Utilizada para garantir uma conexão mais estável, robusta e segura na rede Wi-Fi
SaveData	Utilizada para armazenar os parâmetros do ACU-LUM na EEPROM do ESP32

Outro ponto importante foi a confecção dos módulos propostos, utilizando técnicas de modelagem 3D para visualização *PCB's* e a realização de soldagem e inserção de componentes.

Figura 5.7 – Comparaçāo entre o *layout 3D* e o dispositivo físico montado



5.4 Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes

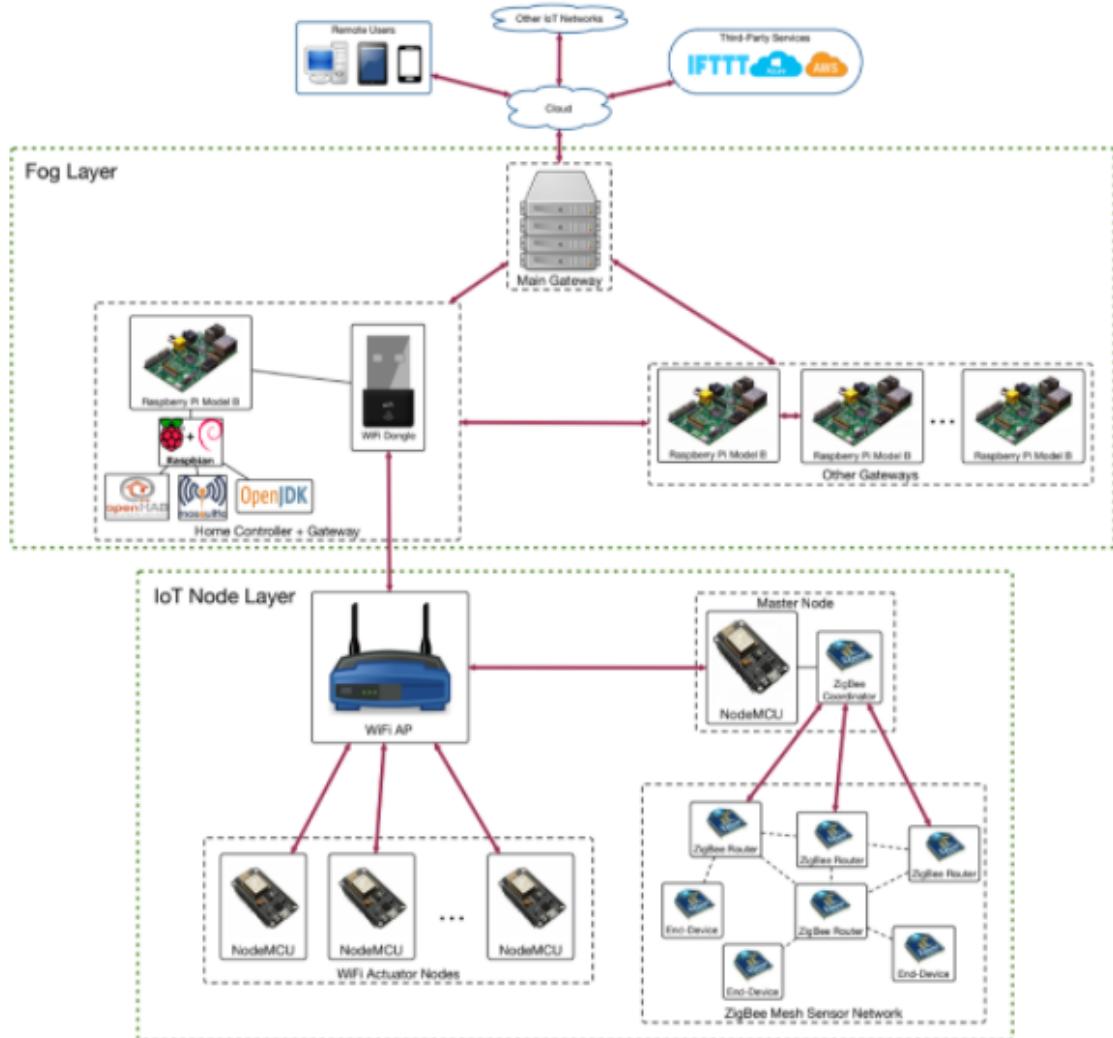
Autores: Iván Froiz-Míguez, Tiago M. Fernández- Camaramés, Paula Fraga-Lamas e Luis Castedo

Este artigo desenvolvido por membros do departamento de engenharia de computação, da Universidade da Coruña, Espanha, apresentou um estudo de caso e aplicação onde buscou-se unir duas tecnologias dentro das características da *IoT*: *WI-FI* e o protocolo *ZigBee*. O protocolo *ZigBee* é muito utilizado quando se tem a necessidade de uma comunicação à longas distâncias com um baixo consumo energético.

O trabalho apresentou a formulação de uma nova tecnologia: o *ZiWi*, um *HAS - Home Automation System* de computação em nuvem que preenche a lacuna entre dispositivos *ZigBee* e *Wi-Fi*, conectando sensores e atuadores perfeitamente para fazer uso de tais tecnologias em uma residência. O *ZiWi* utiliza o *Wi-Fi* para comunicação com atuadores já que, em geral, há a necessidade de estar continuamente operando e ouvindo comandos assíncronos. Enquanto o *ZigBee* é aplicado para sensores porque é ideal para enviar dados em intervalos periódicos, a fim de economizar energia (já que muitos sistemas dependem de baterias). Além disso, o sistema se concentra no crescente mercado de *IoT* e na utilização de sensores emergentes. Além disso, *ZiWi* faz uso da computação em nuvem, paradigma para fornecer conectividade entre o usuário e os diferentes eletrodomésticos, não apenas permitindo ao usuário controlá-los, mas também oferecendo a possibilidade dos dispositivos aprenderem com bancos de dados *online*. Isso é alcançado graças à natureza distribuída do *ZiWi*: o hardware do controlador doméstico é mantido com o mínimo de execuções de tarefas em tempo real, delegando o processamento dos dados e as decisões em tempo não real para servidores em nuvem remotos.

Com a utilização o *Wi-Fi*, o trabalho procurou integrar o protocolo *MQTT* (como a figura de um dispositivo *Broker*, a publicação e inscrição em tópicos) para dispositivos que trabalhavam com o protocolo *ZigBee*. A técnica desenvolvida se tornou muito interessante para aplicações em trabalhos posteriores a esse TCC, como a utilização em ambientes mais isolados.

Figura 5.8 – Esquema para aplicação do protocolo *ZiWi*



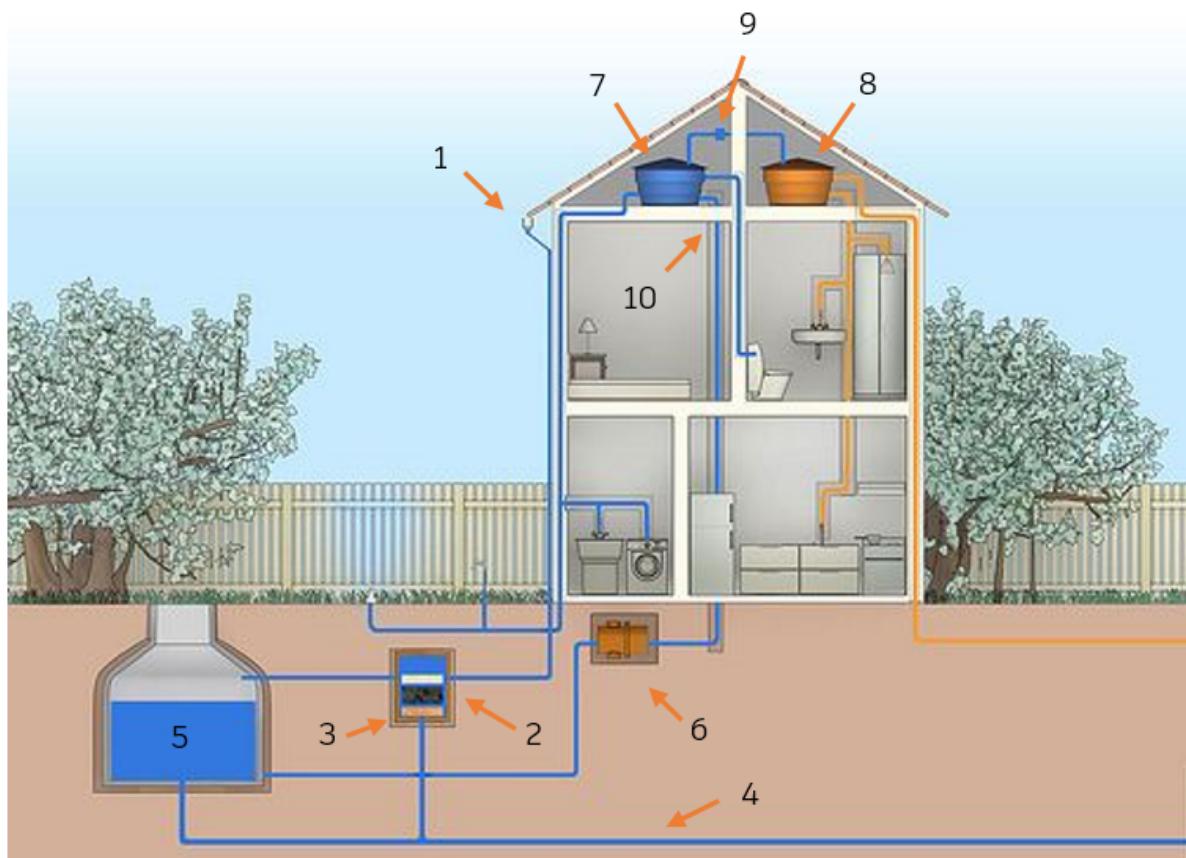
6 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho, estabelecida nas seções posteriores, evidenciará cada parte do projeto com as determinadas ferramentas e recursos a serem utilizados.

6.1 Análise de funcionamento e aplicação de uma cisterna

Para desenvolvimento do trabalho, parte-se do princípio de uma estrutura previamente construída, demonstrada na Figura 6.1. A partir dos elementos dessa figura, evidenciados na Tabela 6.1, é possível realizar uma análise sobre o que pode ser automatizado, levando em consideração as variáveis e características do sistema.

Figura 6.1 – Esquema de demonstração de uma cisterna no subsolo



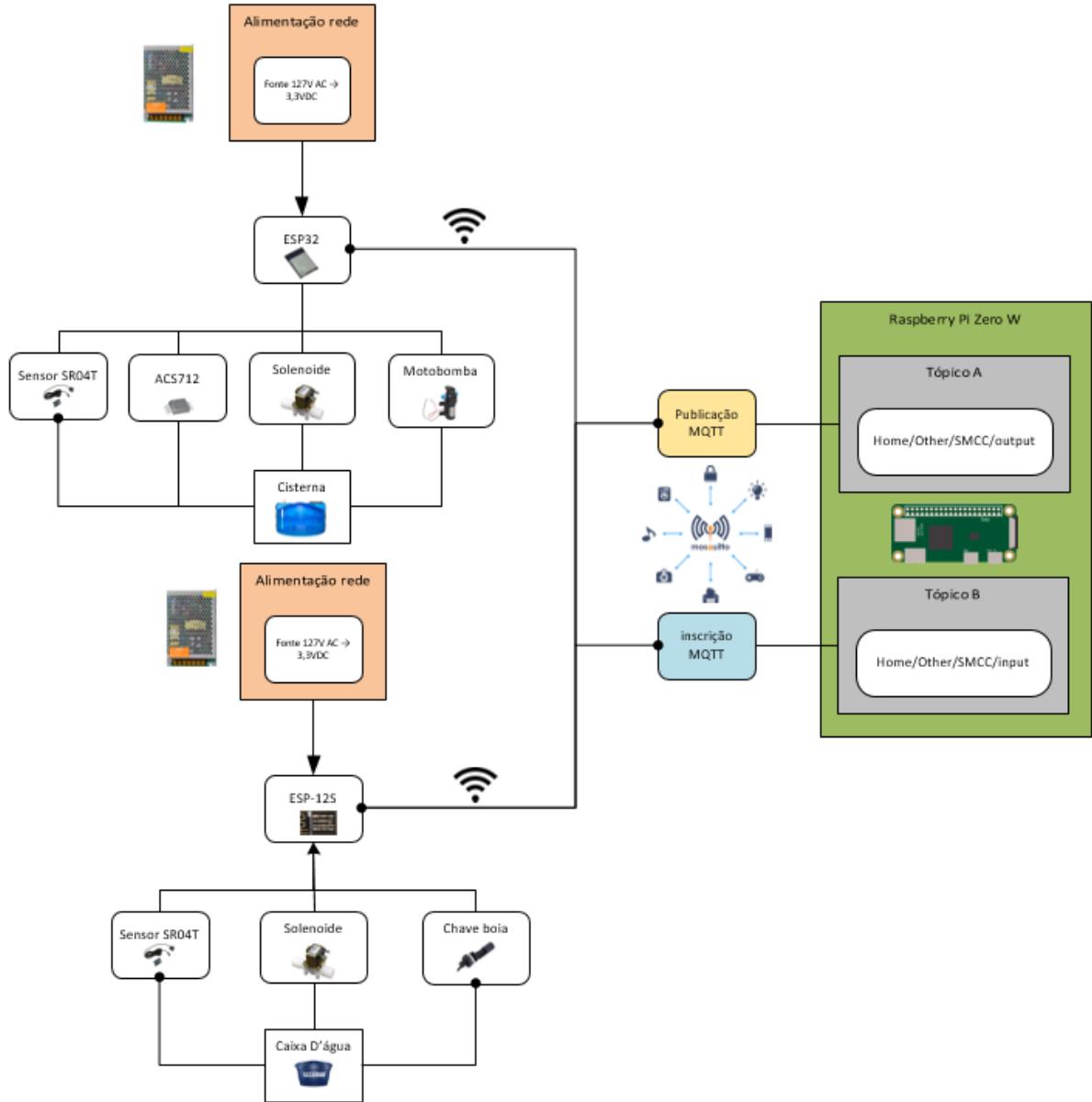
Fonte: Adaptado (ECOMONTES, 2016)

Identificador	Elemento	Descrição
1	Calha coletora	Elemento convencional para coleta e descarte de água da chuva
2	Filtro A (cascalho fino)	Elemento para realização de filtragem de pequenas impurezas
3	Filtro B (cascalho grosso)	Elemento para realização de filtragem de impurezas
4	Tubulação de descarte	Tubulação utilizada como rota de escoamento quando o reservatório não está em uso ou está cheio
5	Reservatório de coleta	Cisterna propriamente dita
6	Motobomba ou bomba d'água	Elemento utilizado para realização do ganho de elevação da água
7	Caixa d'água auxiliar	Caixa d'água convencional com alimentação oriunda da bomba d'água
8	Caixa d'água convencional	Caixa d'água padrão com alimentação da estação de água da cidade
9	Elo de ligação	Ligaçāo utilizada para abastecer a caixa d'água auxiliar quando a cisterna está seca ou em manutenção
10	Distribuidor	Elementos de distribuição de água para pontos estratégicos

Tabela 6.1 – Identificação dos elementos da Figura 6.1.

Após a evidenciar todas as características da estrutura citada anteriormente o um diagrama foi criado tendo em vista a esquematização, a organização e o esclarecimento dos requisitos.

Figura 6.2 – Esquema básico do projeto.



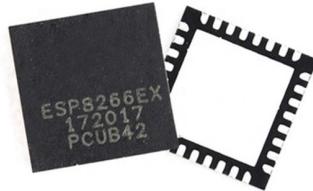
Fonte: Própria

6.2 Elementos de *Hardware*

6.2.1 Os microcontroladores ESP8266 e ESP32

O *ESP8266* (Figura 6.3) é um chip microcontrolador da fabricante chinesa *Espressif Systems*. Construído em torno de um processador Tensilica Xtensa LX3, inclui *Wi-Fi on-board*. Originalmente concebido como um adaptador *UART* para *Wi-Fi* (utilizado em *tablets*), permitindo que outros microcontroladores se conectem a uma rede *Wi-Fi* e façam conexões *TCP/IP* simples usando comandos do estilo *Hayes*, o *ESP8266* rapidamente se tornou popular como um microcontrolador autônomo devido ao seu ponto de preço baixo.

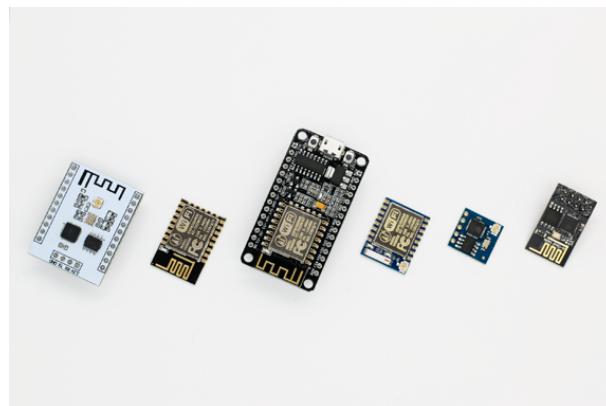
Figura 6.3 – *ESP8266EX*.



Fonte: DIGIKEY, 2020

Apesar da falta de documentação inicial, uma grande comunidade foi formada em torno do ESP8266, e a comunidade integrou e deu suporte ao *firmware* para o chip, fazendo-o compatível com a plataforma *Arduino*. Embora o chip *ESP8266* seja feito pela *Espressif*, existem diversos módulos criados para aplicações distintas (Figura 6.4).

Figura 6.4 – Alguns exemplos de módulos com base no *ESP8266EX*.



Fonte: FILIPEFLOP, 2020

Dentre os modelos, o *ESP-12S* (Figura 6.5) é um módulo, baseado no microcontrolador *ESP8266EX*, que será aplicado neste projeto por ser recente, externar apenas pinos utilizáveis e possuir 14 *GPIOs* - *General Purpose Input Output* sendo ideal para a leitura de uma gama de sensores.

Figura 6.5 – Vista superior do *ESP-12S*.



Fonte: FILIPEFLOP, 2020

O *ESP32*, apresentado na Figura 6.6, é um *chip* também desenvolvido pela *Espressif Systems*. Ele fornece além da conectividade *Wi-Fi*, também presente no *ESP8266*, a possibilidade de conexão *Bluetooth*. Possuindo também uma maior *perfomance* em âmbitos de processamento (dois núcleos), o *ESP32* fornece uma maior quantidade de *GPIOs*, que totalizam 34. Embora o dispositivo seja tecnicamente apenas o *chip*, existem também diversos módulos fabricados por empresas para aplicações específicas que carregam o nome ”*ESP32*”, como o da figura abaixo.

Figura 6.6 – Vista superior do *ESP32*.



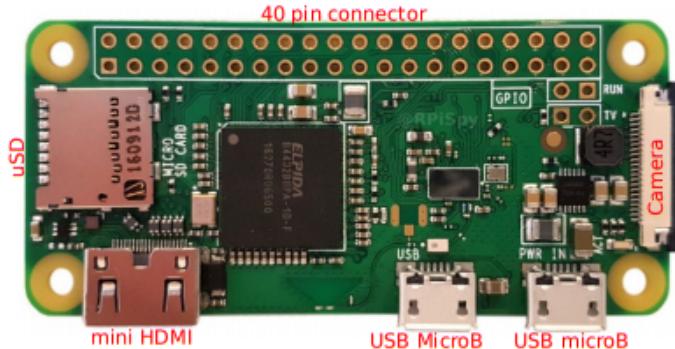
Fonte: DIGIKEY, 2020

6.2.2 A Raspberry Pi Zero W

A *Raspberry Pi Zero W* é basicamente um computador de placa única. Ela possui recursos como *slot* para cartão *microSD*, conectores *HDMI* e de câmera, conectividade *Wi-Fi* e *Bluetooth 4.0*, um conector macho de entrada-saída (*GPIO*) de 40 pinos e mini conector de alimentação + 5VDC. O microcontrolador baseado no processador *BCM2835 ARMv7 system-on-chip (SoC)* alimenta o Pi Zero W.

Essa versão de *Hardware* é muito compacta, ficando no meio termo entre as fases de prototipação e aplicação. Com o microprocessador *ARM BCM2835* de 1GHz, memória *RAM* de 512MB e um sistema operacional enxuto instalado, a *Raspberry Pi Zero W* é ideal para rodar aplicações como um *Broker MQTT*.

Figura 6.7 – Vista superior da *Raspberry Pi Zero W*

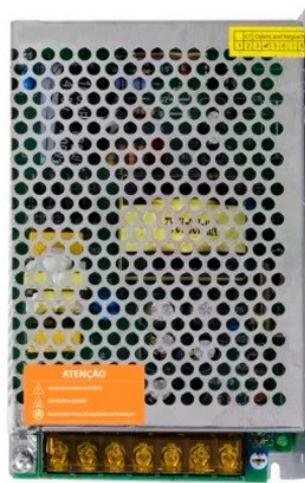


Fonte: SPARKFUN, 2020

6.2.3 A fonte de alimentação DC

Para alimentação elétrica de todos os dispositivos eletrônicos envolvidos no processo de automação da cisterna, a fonte chaveada $12V/10A$ representada na Figura 6.8, é ideal por possuir características robustas, como proteção mecânica e eletrônica. Esse tipo de fonte também disponibiliza vários conectores de saída, os quais podem ser interligados para pontos específicos de alimentação.

Figura 6.8 – Vista superior da fonte



Fonte: AMAZON, 2020

6.2.4 A motobomba ou bomba d’água

A bomba d’água selecionada, Figura 6.9 , trabalha com fontes de alimentação de correntes contínuas (*DC*) o que permite um maior controle do funcionamento através de sistemas *microprocessados*. Essa bomba opera a $12V$, com potência máxima de $80W$. Ela possui a capacidade de exercer uma vazão de $5,5L/min$ com ganho de elevação de no máximo $40m$.

Figura 6.9 – Bomba d'água DC



Fonte: ENERGIA TOTAL, 2020

6.2.5 O sensor de corrente

O sensor de corrente ACS712 selecionado é barato e oferece precisão em soluções para detecção de corrente AC ou DC na indústria, comercial e sistemas de comunicação. O dispositivo permite fácil implementação: aplicações típicas incluem controle de motor, detecção de carga e gerenciamento, fontes de alimentação comutadas e sobrecorrente proteção contra falhas.

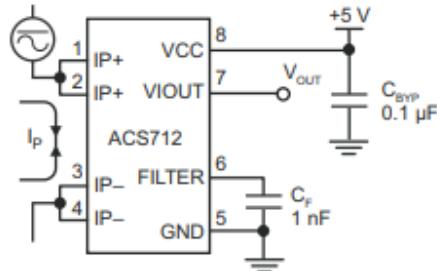
Figura 6.10 – Representação gráfica do sensor ACS712



Fonte: ALEGRO, 2020

O dispositivo consiste em um operador *Hall* linear preciso, a corrente aplicada flui através do material de cobre gerando um campo magnético que é detectado pelo *Hall* integrado e convertido em um proporcional de tensão. A precisão do dispositivo é otimizada por meio do fechamento proximidade do sinal magnético ao transdutor *Hall*. A resistência interna do material condutor atinge uma média de $1,2m\Omega$, proporcionando baixa potência. De acordo *datasheet* do componente, pode-se observar a simples aplicação típica representada na figura abaixo.

Figura 6.11 – Aplicação típica do ACS712



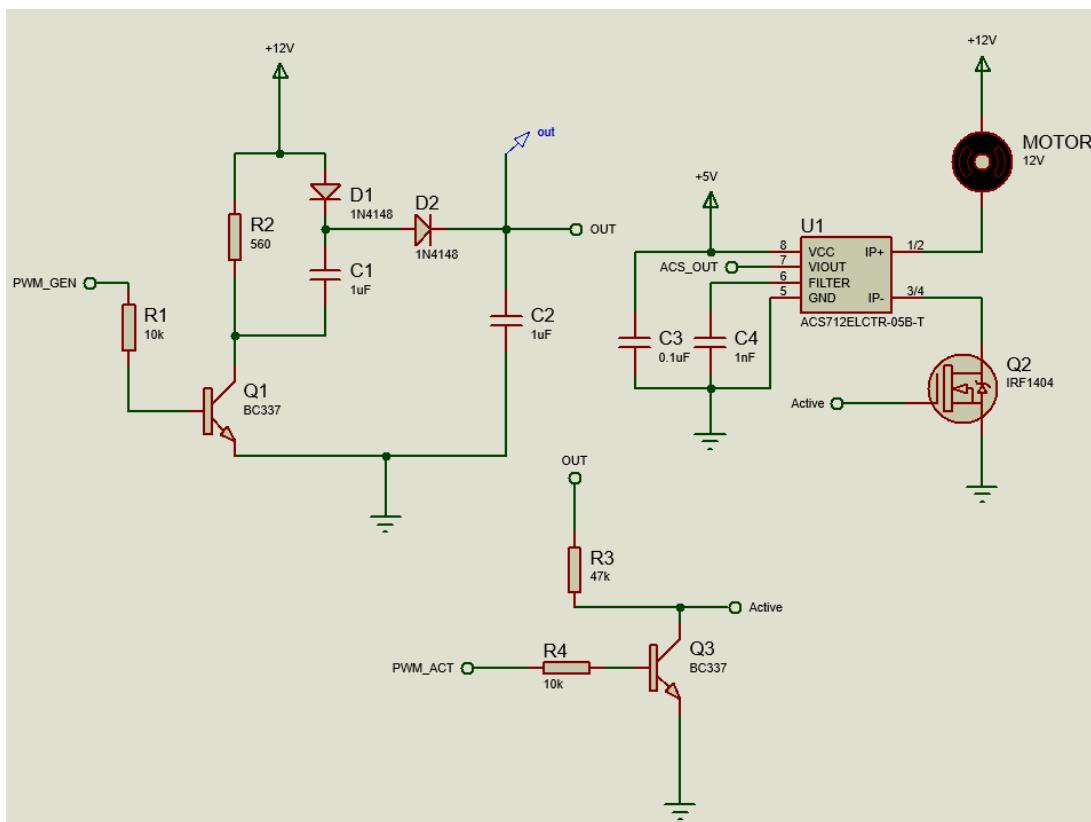
Fonte: ALEGRO, 2020

6.2.6 Os elementos de ativação e desativação

Tendo em vista os elementos descritos nos itens anteriores, tornou-se necessário a implementação de um circuito para ativação e desativação da bomba d'água assim como a execução de técnicas de proteção e isolamento.

O diagrama desenvolvido no software *Proteus* (Figura 6.12) mostra a integração de como seria o sistema de ativação e desativação da motobomba.

Figura 6.12 – Diagrama de ativação com partida lenta



Fonte: Própria

A partir deste diagrama utilizou-se técnicas de partida lenta através do chaveamento transistorizado (*BC337*). Por meio do *PWM - Pulse-Width Modulation* oriundo do microcontrolador, o transistor realiza a alteração sobre o valor eficaz de tensão aplicada na bomba d'água.

Para saturação do transistor *IRF1404* (Figura 6.13) foi desenvolvido o circuito dobrador de tensão também evidenciado na Figura 6.12. A ideia desse circuito é garantir uma queda de tensão entre *GATE* e *SOURCE* duas vezes maior (em torno de $24V$) que a tensão de alimentação do circuito.

Figura 6.13 – Transistor IRF1404



Fonte: ALLDATASHEET, 2020

6.2.7 O sensor de distância ultrassônico

O Sensor Ultrassônico de Distância selecionado foi o *JSEN-SR04T*, desenvolvido exclusivamente para aperfeiçoar projetos de robótica e microeletrônica, mostrando-se capaz de medir distâncias, em relação a objetos, na faixa entre $25cm$ à $150cm$.

Figura 6.14 – Sensor ultrassônico *JSEN-SR04T* e módulo de conversão



Fonte: USINAINFO, 2020

O ponto interessante desse sensor é sua ampla e eficiente resistência à umidade, sendo principalmente utilizado em ambientes úmidos, permitindo manter ampla distância

do microcontrolador. Ele possui com um fio de *2,5 metros* de comprimento e um módulo especialmente desenvolvido para atuar em conjunto com microcontroladores necessitando de apenas quatro portas de conexão: *5V (VCC)*, *Trig (RX)*, *Echo (TX)* e *GND*.

O funcionamento é simples, o sensor emite sinais ultrassônicos que são refletidos ao atingirem determinado objeto. Esses sinais refletidos retornam ao sensor e são processados, tomando como base o tempo de emissão e recepção através do módulo. Em pesquisas realizadas esse sensor apresentou-se ser muito eficiente para a realização da medição de nível.

6.2.8 A válvula solenoide

O modelo de válvula solenoide selecionado possui características como: *1/2" x 1/2"*, tensão de operação em torno de *12V* e consumo médio, quando ativada, de *120mA*. Possui o objetivo de controlar o fluxo de água por meio de comandos elétricos e pode ser utilizada em sistemas de irrigação, tratamento de água ou em qualquer projeto que necessite de um controle sobre o fluxo de líquidos. Essa válvula em seu estado desenergizado impede o fluxo de líquido (*NF - Normalmente Fechada*), isso garante que durante a eventual falta de energia o líquido se manterá contido.

Figura 6.15 – Válvula solenoide



Fonte: SMART KITS, 2020

6.2.9 A chave de nível tipo boia

Para questões de segurança a chave de nível *RF-0H21D* será utilizada para fomentar a redundância do sistema, caso aconteçam falhas na medição do sensor citado na seção 6.2.7. Essa chave indicará para o microcontrolador, por meio de contato elétrico, o momento exato para desativação da bomba d'água, evitando possíveis vazamentos.

Figura 6.16 – Chave de nível *RF-0H21D*



Fonte: SMART KITS, 2020

6.3 Elementos de *Firmware*

6.3.1 O Ambiente de Desenvolvimento Integrado *Arduino*

A plataforma Arduino, criada em 2005, não abrange somente as placas microcontroladas mas também agrega-se de uma Interface de Desenvolvimento, ou seja uma IDE, que possui um compilador *C++*. A utilização dessa IDE se tornou tão difundida atualmente que grandes empresas de tecnologia têm buscado tornar ou fabricar microcontroladores completamente compatíveis com a interface. Com esta interface, será possível a programação dos microcontroladores *ESP8266* e *ESP32* de maneira facilitada e sem a necessidade do aprendizado de uma nova linguagem. Todas as funcionalidades, como rotinas de conexão *HTTP* e/ou *MQTT* através do *Wi-Fi*, Interrupção, Timers, Modos de Energia são completamente acessíveis.

Figura 6.17 – Gerenciador de placas com suporte oficial às placas *ESP32* e *ESP8266*



Fonte: Própria

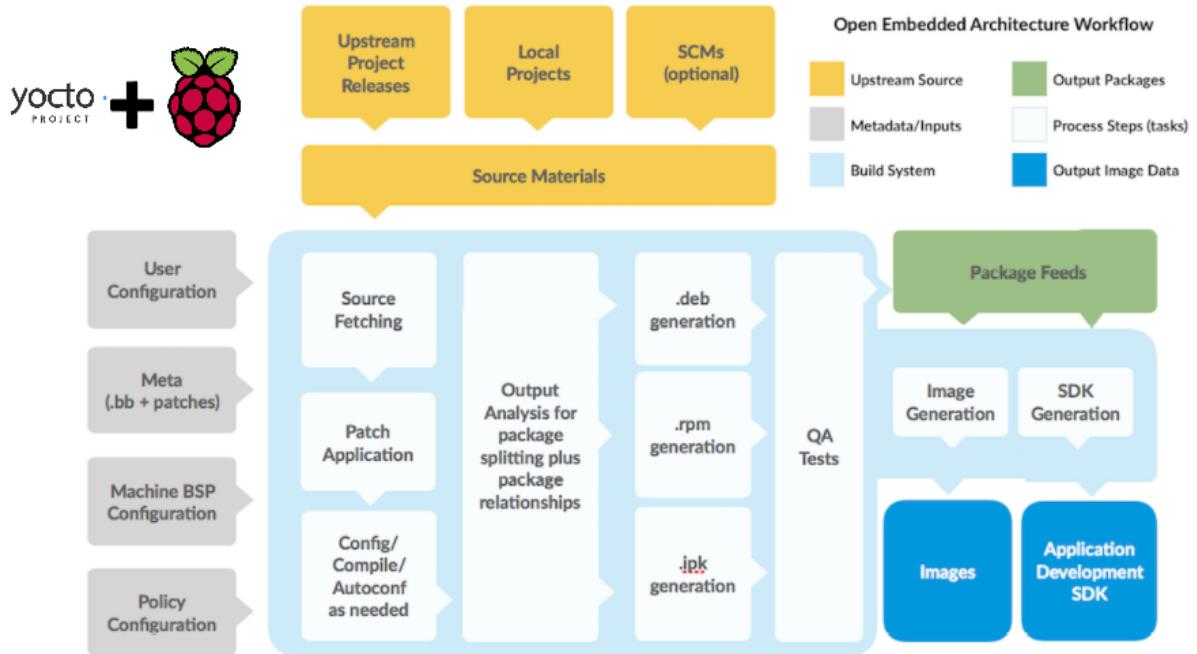
6.3.2 Modelagem de um *SO* embarcado com *Yocto Project*

O *Yocto Project* é um projeto colaborativo de código aberto que fornece modelos, ferramentas e métodos facilitando a criação de sistemas personalizados baseados em Linux para implantações de sistemas embarcados em dispositivos conectados, servidores ou ambientes virtuais, independentemente da arquitetura de hardware.

Por ser um projeto *open source*, opera com uma estrutura de governança hierárquica baseada na meritocracia e gerenciada por seu arquiteto-chefe. Isso permite que o projeto permaneça independente de qualquer um de seus organizadores membros, que participam de várias maneiras e fornecem recursos para o projeto.

O projeto é apoiado e administrado por líderes da indústria de alta tecnologia que se comprometeram financeiramente, com suporte de plataforma e esforços de marketing para tornar o *Yocto Project* um padrão seguro, estável e adaptável da indústria.

Figura 6.18 – Fluxo de trabalho geral *Yocto Project*



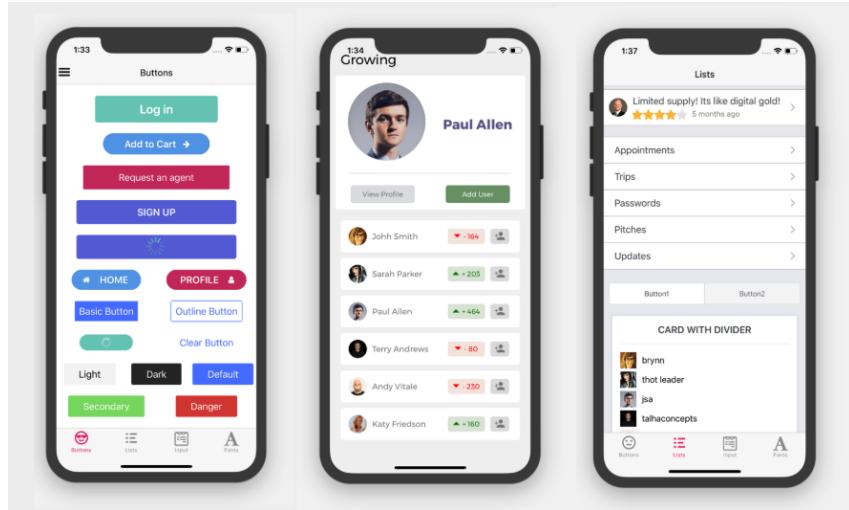
Fonte: Adaptado (yoctoproject.org, 2020)

6.4 Elementos de Software

6.4.1 O framework React Native

Baseado no *React*, para desenvolvimento *WEB*, o *React Native* é um *framework* de código aberto desenvolvido pela equipe do *Facebook* que suporta a criação de aplicativos *mobile* multiplataforma (Android e iOS), sem que haja a preocupação de lidar com as linguagens padrões como *Java* ou *Swift*, usando apenas *Javascript*. Como ponto positivo, ao contrário de outros *frameworks* com o mesmo propósito, todo código desenvolvido com *React Native* será convertido para a linguagem nativa do sistema operacional, o que torna o aplicativo mais performático.

Figura 6.19 – Exemplos de *Templates* produzidos com *React Native*

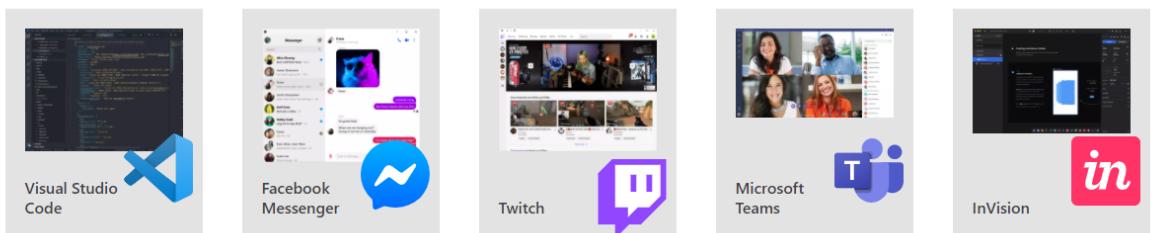


Fonte: Adaptado (instamobile.io, 2020)

6.4.2 O framework *Electron*

O *Electron* é um *framework* multiplataforma (Windows, Linux e MacOS) para criação de interfaces possibilitando o usuário acessar serviços do sistema operacional tanto via linha de comando - *CLI* e interface gráfica - *GUI*.

Figura 6.20 – Aplicações que utilizam *Electron*



Fonte: Adaptado (electronjs.org, 2020)

Por meio dele podemos desenvolver aplicações *desktop* utilizando *HTML*, *CSS* e *Javascript*. O *Electron* vem com navegador *Chromium*, um projeto *open source* de onde surgiu o *Google Chrome*. Toda a parte visual, janelas, etc. são renderizadas nessa camada e o *BackEnd* é executado em *Node.js*. Ambos tem acesso um ao outro via *RPC* (Remote Procedure Call).

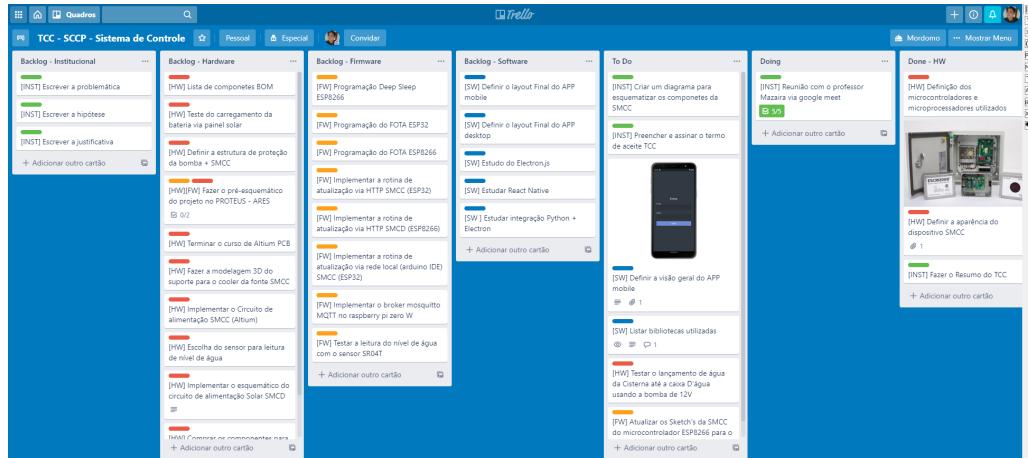
7 METODOLOGIA

A metodologia para a realização deste trabalho de conclusão de curso, iniciou-se com a criação do quadro *Kanban* (Figura 7.1), para auxílio da organização das tarefas, e com a criação do repositório no *Github* (Figura 7.2) para armazenamento e versionamento dos códigos das vertentes de *Firmware* e *Software* do projeto. Deu-se prosseguimento com os estudos e levantamentos bibliográficos relacionando três áreas do projeto. Buscou-se encontrar os métodos, ferramentas, tecnologias, bibliotecas e *frameworks* mais adequados para a implementação do projeto. Diante de cada seleção feita, foram executadas análises para que fosse definido o funcionamento do sistema total, munido da combinação de cada uma das três áreas citadas anteriormente e visando a geração de um produto que pudesse ser empregado no mercado: atrativo economicamente e seguindo diretrizes sustentáveis.

Primeiramente, na seção de *hardware*, foram selecionados as ferramentas para modelagem de placas de circuito impresso - *PCB's* e para modelagem de peças *3D*, através de manufatura aditiva, a escolha de todos os dispositivos eletrônicos a serem utilizados e a análise do local de aplicação. Posteriormente, na parte de *firmware*, já estando selecionados os microcontroladores e o microprocessador, foram determinadas todas as rotinas de operação e escolhidas, respectivamente, as linguagens para programá-los e o *framework* para criação de um sistema operacional embarcado baseado em *kernel Linux*. Na parte de *software*, foram selecionadas as ferramentas para a criação de interfaces dentro da camada *front-end*: aplicações *mobile* e *desktop*.

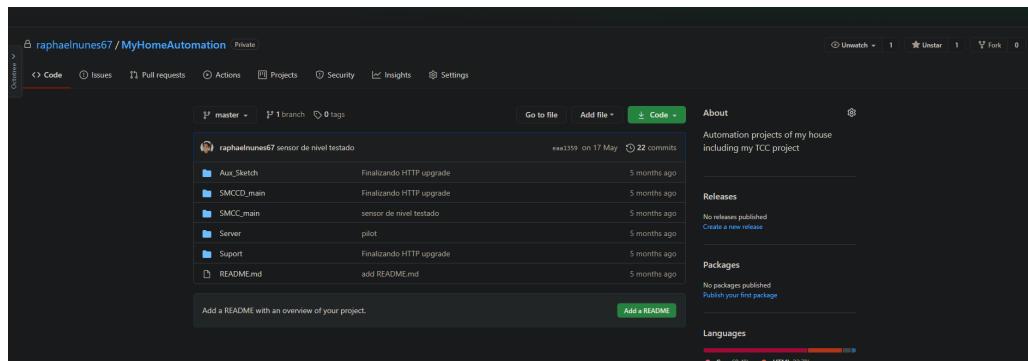
Por fim, elaborou-se a lista de materiais necessários para construção o projeto, tendo objetivo de validação em ambiente real, efetuando testes de longos períodos, averiguando a integridade do sistema, a robustez dos componentes e a identificação de casos não previstos anteriormente, tornando possível a aplicação de melhorias posteriores a entrega deste trabalho.

Figura 7.1 – Visão geral do quadro *Kanban* criado na ferramenta *Trello*



Fonte: Própria.

Figura 7.2 – Visão geral repositório criado no *Github*



Fonte: Própria.

7.1 Levantamento do referencial bibliográfico e capacitação

Essa parte do trabalho confeve-se no levantamento de referências bibliográficas relacionadas com os temas de *IoT*, programação de microcontroladores, criação de aplicações com *frameworks* baseados em *Javascript*. Também buscou-se a capacitação em cursos oferecidos pelas plataformas *Alura*, *Udemy* e *Skylab (Rocketseat)* o aprendizado de conteúdos complementares ao curso de formação em engenharia de controle e automação, como a criação de placas de circuito impresso, treinamentos sobre *Linux* embarcado, implementação do protocolo *MQTT*, criação de *APPs Android* com *React Native* e aplicações *Desktop* com *Electron.js*.

7.2 Desenvolvimento dos elementos de *Hardware*

Nesse momento foram definidos todos os elementos de *hardware* necessários para a execução do projeto: para os dispositivos eletrônicos, documentos como *Datasheets*,

catálogos e informativos de dispositivos elétricos foram coletados para consulta durante o decorrer do trabalho; para a parte estrutural executou-se a enumeração de peças que deveriam o desenhadas com auxílio do *Software Autodesk Inventor* e posteriormente construídas através de manufatura aditiva. Realizou-se uma busca no mercado pelos componentes necessários efetuando as possíveis compras e elaborando o orçamento geral do projeto.

7.3 Desenvolvimento dos elementos de *Firmware*

A definição e desenvolvimento dos elementos de *Firmware* se deu após todos os levantamentos de requisitos do trabalho. Esse momento concentrou-se na escolha de tecnologias mais fáceis de implementação, que tivessem uma gama de documentações e que fossem empregadas em produtos oficiais. Para lidar com o uso dos microprocessadores e microcontroladores, foram feitas pesquisas visando consolidar conceitos aprendidos durante o curso de graduação.

7.4 Desenvolvimento dos elementos de *Software*

Para o desenvolvimento dos elementos de *Software* buscou-se a participação de cursos básicos e avançados sobre aplicações *front-end*, as quais servem de conteúdo complementar aos temas abordados na graduação em engenharia de controle e automação.

Os cursos adquiridos trouxeram uma série de conhecimentos para uma maior interação entre desenvolvedor e usuário, possibilitando a criação de interfaces intuitivas e agregando novas informações e visões a outros temas, como na programação de sistemas embarcados.

8 CRONOGRAMA

2020	Atividade	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
1	Elaboração do Tema do TCC	20-fev										
2	Escolha do orientador		10-mar									
3	Ajustes de pontuais do tema de acordo com reuniões com o orientador		23-mar									
4	Recolhimento e entrega do formulário de aceite do orientador		25-mar									
5	Realização das primeiras pesquisas sobre trabalhos relacionados			3-abr								
6	Produção da problemática (pergunta da pesquisa)			14-abr								
7	Formulação da Hipótese e Justificativa			20-abr								
8	Levantamento e aquisição de materiais necessários para realização de subpartes do sistema			*	*							
9	Realização de cursos complementares para a produção do TCC				*	*	*					
9	Formulação dos objetivos gerais e específicos							15-ago				
10	Entrega dos trabalhos relacionados								2-set			
11	Entrega do referencial teórico								29-set			
12	Entrega da metodologia e cronograma									7-out		
13	Entrega da versão final do TCC I									10-out		
14	Formulação da apresentação do TCC I									12-out		
15	Defesa do projeto de pesquisa (TCC I)									20-out		
16	Início das atividades do TCC II										2-nov	
17	Desenvolvimento do FW embarcado ESP32 e ESP8266										20-nov	
18	Desenvolvimento do FW embarcado Raspberry Pi Zero W											2-dez

Representação das atividades desenvolvidas no ano de 2020.

2021	Atividade	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
19	Aquisição final de dispositivos e componentes	20-jan										
20	Recolhimento e entrega do formulário de aceite do orientador	25-jan										
20	Desenvolvimento das interfaces desktop e mobile	30-jan										
21	Entrega do Referencial Teórico		2-fev									
22	Modelagem as PCB's e das peças 3D		15-fev									
23	Entrega da Metodologia		17-fev									
24	Montagem dos módulos (soldagem dos componentes e posicionamento dos sensores)			10-mar								
25	Realização testes de integração e sistema			30-mar								
26	Entrega dos Resultados e Discussão				7-abr							
27	Entrega das Considerações Finais e Trabalhos Futuros				10-abr							
28	Entrega do Artigo				15-abr							
29	Apresentação TCC2				15-abr							

Representação das atividades desenvolvidas no ano de 2021.

REFERÊNCIAS