

APLICAÇÃO PARA CONTROLE DE FLUXO E MENSAGEM ENTRE DISPOSITIVOS IOT – BCC

Silvio Greuel, Dalton Solano dos Reis – Orientador

Resumo: O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATÓRIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

Palavras-chave: Ciéncia da computaçáo. Sistemas de informaçáo. Monografia. Resumo. Formato.

1 INTRODUÇÃO

Um dos principais motivos para o estabelecimento dentre tantas tecnologias disponíveis na geração atual, se dá pelo modelo de código aberto, licença flexível e não proprietária em que certas tecnologias têm por base. Uma destas tecnologias se trata da própria Internet, qual possui licenciamento por dito livre desde a liberação da propriedade intelectual em 1993 pelo CERN (1993, p. 2, tradução nossa) “O CERN renuncia a todos os direitos de propriedade intelectual deste código, tanto ao código fonte quanto ao binário, e é dada permissão a qualquer pessoa para usá-lo, duplicá-lo, modificá-lo e distribuí-lo”. Tal liberação permitiu a criação de muitas outras tecnologias, como: navegadores para acesso a conteúdo disponíveis via internet, inúmeros protocolos de transmissão de dados, criptografia e até os smartphones. Seguindo o mesmo princípio de evolução dos smartphones, espera-se também, que de maneira semelhante, a utilização de dispositivos Internet of Things (IoT), cresça e se estabeleça. A procura por soluções domésticas abriu caminho para a automação residencial por meio de sistemas que se propõe a melhorar a qualidade de vida e economizar recursos (SOUZA, 2016). A fragmentação de softwares e protocolos dificulta a comunicação de dispositivos IoT e é neste ponto que o open source se torna importante. Por mais que cada fabricante programe seus produtos para funcionar de uma forma, eles estando em uma plataforma de código aberto, é possível fazer com que todos esses dispositivos diferentes trabalhem juntos (OLHAR DIGITAL, 2016).

Atualmente, existem diferentes tipos de serviços e aplicativos qual tem como principal objetivo realizar o controle e gestão de dispositivos IoT. Dentre tais aplicativos, como destaque, temos a aplicação HomeKit (Apple, 2018), que disponibiliza um framework para controle de alguns dispositivos, porém, somente dispositivos proprietários ou com o certificado exigido pela marca,

(APPLE 2017, tradução nossa) indica que “Com o app Casa, você pode acessar remotamente todos os acessórios inteligentes pela Apple TV ou iPad. É possível fechar o portão da garagem, ver a câmera de vídeo da porta da frente, pedir à Siri (assistente virtual) para diminuir a temperatura ou qualquer outra coisa que você já está acostumado a fazer quando usa o app Casa em casa”.

Outra aplicação que merece destaque, não só por se conectar com dispositivos IoT, mas, pelo modelo de controle de fluxos e mensageria entre variados serviços, chama-se If This Then That (IFTTT).

(Vorapojpisut 2015, p. 2, tradução nossa) “o aplicativo IFTTT permite aos usuários criarem, customizarem e habilitarem correntes condicionais, quais são chamadas de receitas, em que são ativadas com base em alterações em outros serviços, como Facebook, Twitter e Youtube”.

Com a simples apresentação das soluções HomeKit e IFTTT, nota-se a comodidade que os aplicativos de automação residencial, qual fazem gestão e controle, de dispositivos IoT, serviços sociais (facebook, twitter) e serviços privados (posicionamento geográfico), trazem para o cotidiano de uma pessoa, como afirmado em (FLORES, LUNDMARK e MÁHR, 2005, p. 3, tradução nossa) “Conveniência é um dos principais pontos de venda para automação residencial (...).” Apesar dos benefícios do uso de tais serviços estarem em evidência, ao exemplo da aplicação HomeKit, percebe-se, que manter exclusividade em seus serviços, limita sua utilização. Expondo então, a falta de aplicativos multiplataforma, serviços e frameworks com uma licença não proprietária, de código livre e alteração permissiva.

O presente trabalho, foca no desenvolvimento de uma aplicação multiplataforma, de código aberto e permissivo, para controle de dispositivos IoT, onde, terá como referência, o modelo de controle de fluxo e mensageria similar a aplicação IFTTT. Tendo em vista o desenvolvimento multiplataforma será utilizada a tecnologia Progressive

Web App (PWA), se trata de uma aplicação web qual são concedidos algumas permissões de acesso no contexto do sistema operacional, antes somente concedidas para aplicações nativas, como explica Fransson e Driaguine (2017, p. 5, tradução nossa) “Uma aplicação PWA é uma aplicação web, qual é incrementada com algumas tecnologias que permitem um comportamento similar a uma aplicação nativa em dispositivos móveis, enquanto mantém também seu funcionamento em um navegador“.

1.1 OBJETIVO GERAL

Criar aplicação de controle dos fluxos de mensageria entre os dispositivos IoT, qualquer mudança do fluxo das mensagens e decisões de tomada de decisão entre os dispositivos, serão controlados pela aplicação.

1.2 OBJETIVOS ESPECÍFICOS

- a) Explorar aplicação multiplataforma, validando sua viabilidade nas plataformas desktop e móvel
- b) Explorar integração de instalação nativa no sistema operacional, validando o acesso off-line a aplicação
- c) Explorar o uso da aplicação utilizando uma bancada de homologação, descrevendo de modo que possa ser futuramente reproduzido
- d) Levantar casos de uso no ambiente agrário e apresentar uma solução aplicável utilizando este trabalho
- e) Levantar casos de uso no ambiente industrial e apresentar uma solução aplicável utilizando este trabalho

2 FUNDAMENTAÇÃO TEÓRICA

Está **secção** descreve os **assuntos que**

2.1 CONCEITOS

Neste **capítulo** são expostos os resultados do trabalho, ou seja, o software desenvolvido no Capítulo 3 Metodologia. Primeiramente, é demonstrado o conjunto de Dispositivos de Automação Residencial utilizados neste capítulo. Em seguida, demonstra-se a Interface de Usuário da aplicação por meio de capturas de tela.

2.1.1 Fluxo - *Flow*

Agrega o gatilho e uma árvore de decisão de nodos atuadores. É a principal entidade d

2.1.2 Gatilho - *Trigger*

Pertence a um **fluxo**, também é um nodo atuador, porém se difere, pois é o primeiro nodo da árvore de decisão a ser processado.

2.1.3 Nodo Atuador - *Action Node*

Pertence a um **fluxo**, realizando a composição de uma **árvore** de decisão, é responsável por executar uma ação e realizar uma tomada de decisão. Possui link para outros dois nodos, um nodo de sucesso e um nodo de falha. Dada a execução da ação com falha caso houver um link para um **nodo** de falha, será realizada a chamada de execução de ação do nodo de falha. Dada a execução da ação com sucesso, caso houver um link para um **nodo** de sucesso, será realizada a chamada de execução para o nodo de sucesso. Para toda chamada da próxima execução, também é enviado um contexto, qual poderá ser usado como dicionário de registros pelos nodos subsequentes.

2.1.4 Contexto

Estrutura de dados estilo dicionário que é compartilhada entre execuções entre nodos de ação. Tal estrutura é criada primeiramente pelo nodo gatilho, atribuindo as informações dispostas para o gatilho. No caso do gatilho MQTT, a mensagem enviada será processada de forma que, todos os campos sejam atribuídos ao contexto. No caso do gatilho **BUTTON**, o contexto estará em branco.

2.1.5 Dispositivo

Objeto físico de identificação única, realiza tanto a coleta de dados via sensores como de temperatura, quanto a execução de tarefas via atuadores **como de**

2.2 TÉCNICAS

2.2.1 App Manifest

O **App Manifest**, não exatamente é uma **API**, se trata de um arquivo de definição da aplicação, é interpretado pelos navegadores em função de habilitar o acesso a APIs privilegiadas e permitir sua instalação como aplicação nativa (**MOZILLA 2018**)

2.2.2 Service Workers

Service Workers atuam essencialmente como servidores de meio, que ficam entre os aplicativos da Web, o navegador e a rede (quando disponíveis). Destinam-se, entre outras coisas, a permitir a criação de experiências off-line efetivas, interceptando solicitações de rede e tomando as medidas adequadas com base na disponibilidade da rede MOZILLA (2018).

2.2.3 Web Cache

O **Web Cache** fornece um mecanismo de armazenamento para requisições e respostas, tal armazenamento é transiente entre outras requisições e respostas (MOZILLA 2018).

A utilização do **Web Cache** é necessária para habilitar a instalação off-line da aplicação, realizando o cache das requisições do código fonte da aplicação PWA. Então quando off-line, ao se notar da falta de conexão com a internet, o **Service Worker** realiza um consulta no **Web Cache**, retornando o **ultimo** código fonte disponível para a aplicação PWA.

2.2.4 Add to Home Screen

O recurso permite a adição da aplicação PWA na tela inicial do dispositivo, é um recurso disponível nos navegadores modernos que permite aos desenvolvedores adicionar um atalho para sua tela inicial de forma fácil e conveniente (MOZILLA, 2018)

Para que o recurso funcione adequadamente, se é necessário garantir alguns atributos na aplicação: servir a aplicação via https, identificar a via **App Manifest** que a aplicação se trata de uma PWA, registrar corretamente ao mínimo um **Service Worker**

2.2.5 Árvore de Decisão

A principal vantagem das árvores de **decisões, são** que elas são um método de “caixa branca”. Isso significa que podemos modelar facilmente um fluxo de decisões (BECOMINGHUMAN 2017).

De modo geral uma árvore de decisão consiste nos seguintes componentes: nodos são perguntas ou condições, arestas são as opções de continuação em falha ou sucesso e as folhas são as ações finais.

2.3 FERRAMENTAS

2.3.1 RabbitMQ

O RabbitMQ é uma implementação popular do protocolo **AMQP**, o RabbitMQ implementa uma arquitetura de intermediário, o que significa que as mensagens são armazenadas em uma fila no servidor RabbitMQ antes de serem enviadas aos clientes. Segundo (Shailesh, Joshi e Purandare, 2018) o RabbitMQ é fácil de usar e implantar, pois recursos como roteamento, balanceamento de carga ou persistência em enfileiramento de mensagens podem ser implementados em menos linhas de código. No entanto, ele também é menos escalável e “mais lento”, já que o servidor central adiciona latência.

2.3.2 Postgresql

O PostgreSQL é um poderoso sistema de banco de dados objeto-relacional de código-fonte aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e dimensionam com segurança as cargas de trabalho de dados mais complicadas (POSTGRESQL, 2018).

O PostgreSQL ganhou uma forte reputação por sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto robusto de recursos, extensibilidade e dedicação da comunidade de código aberto por trás do software para fornecer consistentemente soluções eficazes e inovadoras (POSTGRESQL, 2018).

2.4 TRABALHOS CORRELATOS

Alguns trabalhos, nos quais as características se assemelham ao **proposto, serão** agora listados. O primeiro, seção 2.1 trata da definição de um framework similar ao modelo da aplicação IFTTT, elaborado por Vorapojpisut (2015). O segundo trabalho, seção 2.2, realiza um comparativo entre variados protocolos de comunicação com foco em dispositivos (IoT) (ADLINK, 2017). O terceiro trabalho, seção 2.3, coloca em **prova, a viabilidade** da utilização de uma aplicação utilizando a tecnologia PWA defronte a uma aplicação nativa.

Quadro 1 – Trabalho Correlato 1

| | |
|--------------------------------|--|
| Referência | Vorapojpisut (2015) |
| Objetivos | O conceito principal é formular uma automação mínima que integre múltiplas ações em fluxos de decisão que o usuário escolhe. O trabalho tenta propor uma estrutura de software que promova a flexibilidade e a capacidade de manutenção e outras métricas de qualidade. |
| Principais funcionalidades | Definição de um sistema de execução de ações arbitrárias em fluxos registrados pelo usuário |
| Ferramentas de desenvolvimento | Além de textos qual definem o sistema, o autor usou também de modelagem UML para apresentar sua solução |
| Resultados e conclusões | O autor obteve uma modelagem simples para o problema, onde, somente duas aplicações são necessárias, uma aplicação de controle e outra de processamento. Aplicação de controle realiza tem como principal funcionalidade realizar a gestão de usuários, ações e fluxos. Aplicação de processamento recebe mensagens, identificando qual fluxo deve ser executado e então o executando. |

Fonte: elaborado pelo autor.

Vorapojpisut (2015), descreve um modelo de aplicação definindo as funcionalidades e conceitos básicos que compõem o serviço IFTTT, segundo Vorapojpisut (2015, p. 2) a abordagem proposta pelo serviço IFTTT oferece vantagens entre funcionalidade e usabilidade. Em comparação com a aplicação desenvolvida neste trabalho, também será aplicado o modelo resultante do autor, qual trata de uma aplicação de controle, para realizar a gestão e uma aplicação de processamento, qual tratará da execução dos fluxos.

Quadro 2 – Trabalho Correlato 2

| | |
|--------------------------------|--|
| Referência | ADLINK (2017) |
| Objetivos | O trabalho tem como principal objetivo, comparar os protocolos de comunicação DDS, AMQP, MQTT, JMS, REST, CoAP, e XMPP, tendo como principal foco sua utilização em soluções IoT |
| Principais funcionalidades | Apresentação e descrição dos tópicos , sendo eles: segurança, performance, qualidade de serviço, intratabilidade, estratégia e estratégia de dados |
| Ferramentas de desenvolvimento | Método científico e analítico, realizando pesquisa sobre as especificações dos protocolos e os comparando em tópicos comuns |
| Resultados e conclusões | Como resultado, se obteve uma tabela comparativa de conformidades entre os protocolos. As comparações são |

Fonte: elaborado pelo autor.

O trabalho foi desenvolvido pelo grupo ADLINK (2017), enfoca nas tecnologias de mensageria que estão emergindo como as mais importantes para os dispositivos IoT, como afirmado pelo grupo (ADLINK 2017, p. 5, tradução nossa) “Existem muitas tecnologias de mensagens diferentes. No entanto, um subconjunto muito menor está emergindo como o mais importante”. São analisados os protocolos, *Data Distribution Service for Real-Time Systems* (DDS), *Advanced Message Queuing Protocol* (AMQP), *MQ Telemetry Transport* (MQTT), *Java Message Service* (JMS), *Representational State Transfer* (REST), *Constrained Application Protocol* (CoAP), *eXtensible Messaging e Presence Protocol* (XMPP). A conclusão obtida pelo trabalho, indica como utilização ideal para dispositivos IoT, os protocolos MQTT, AMQP e REST. Realizando a análise das conclusões obtidas pelo trabalho correlato 2, houve a escolha do uso de dois protocolos. As propriedades de qualidade de serviço definidas pelo protocolo AMQP, exercem vantagem sobre as demais ao se tratar da garantia de entrega, sendo então a escolha de protocolo para o processamento dos fluxos. Para a comunicação dos dispositivos IoT com a aplicação, o protocolo MQTT se torna viável por ter como um dos principais atributos, a baixa latência no envio das mensagens.

Quadro 3 – Trabalho Correlato 3

| | |
|--------------------------------|---|
| Referência | (FRANSON, DRIAGUINE, 2017) |
| Objetivos | Validar o uso de PWAs como substituto para aplicações nativas |
| Principais funcionalidades | Apresentação de duas aplicações, uma aplicação PWA e uma aplicação nativa, as funcionalidades das duas aplicações consiste em, acesso a câmera e uma tela para realizar o teste de performance de renderização da aplicação |
| Ferramentas de desenvolvimento | Metodologia científica, para identificar a performance das aplicações PWA em comparação com aplicações nativas, analisando o tempo de acesso a câmera e o tempo de renderização da |

| | |
|-------------------------|---|
| | aplicação. Metodologia de viabilidade, listando as possibilidades de acesso de uma PWA ao sistema nativo |
| Resultados e conclusões | Criação de uma tabela contendo a comparação direta de performance entre uma aplicação PWA e uma aplicação nativa, de modo geral, ao se tratar do tempo de acesso a câmera, as duas aplicações não se diferem em muito, já, o tempo de renderização da aplicação PWA em comparação com a aplicação nativa, deixa a desejar Criação de uma tabela identificando as possibilidades de acesso de uma aplicação PWA, indicando que possibilidade de uso do bluetooth, geolocalização, sistema de arquivos, modo offline entre outros. |

Fonte: elaborado pelo autor.

O trabalho de (FRANSON, DRIAGUINE, 2017) apresenta inúmeras análises comparando aplicações utilizando a tecnologia PWA e aplicações nativas. Dentre as análises, são de destaque, o consumo de bateria, o tempo de renderização completa e acessibilidade das funções disponibilizadas pelo sistema móvel, por exemplo *Global Positioning System* (GPS). A aplicação desenvolvida neste trabalho irá utilizar também o conceito de PWA fazendo o uso de acesso nativo ao Bluetooth e *Near Field Communication* (NFC).

3 DESCRIÇÃO DA APLICACÃO

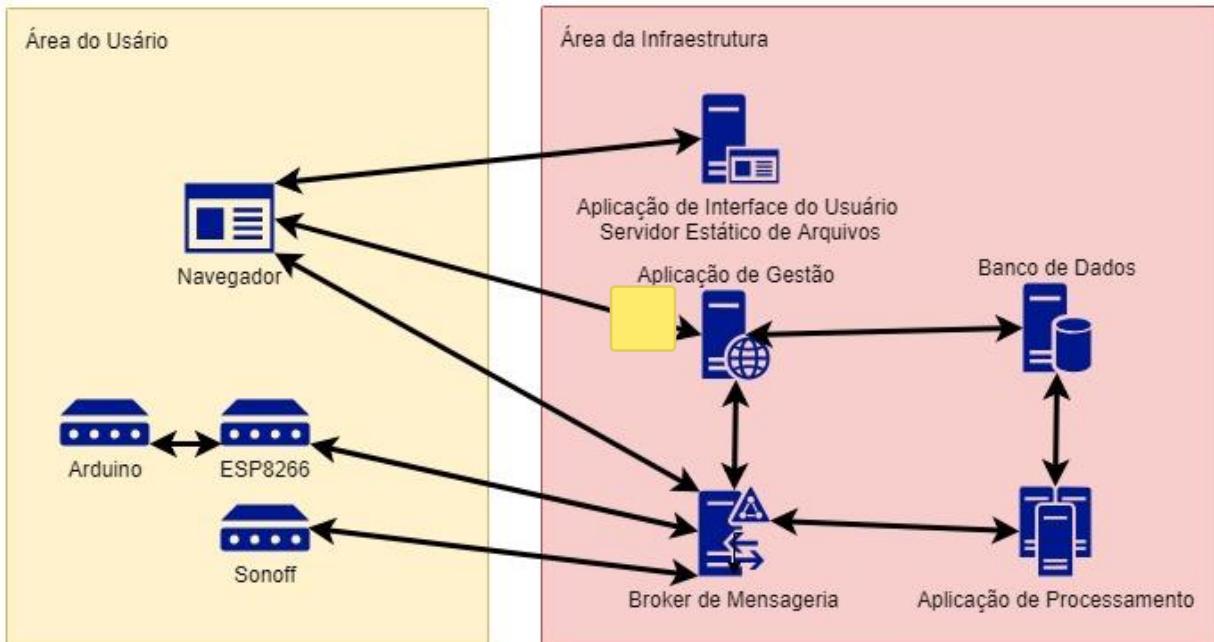
3.1 VISÃO GERAL

O aplicativo tem por objetivo a automação e intercomunicação entre dispositivos IoT, para que seu utilizador tenha o total controle dos fluxos e tomadas de decisões, mesmo estando remoto. Os principais requisitos para a criação do sistema são: facilidade de configuração, utilização de hardware de baixo custo, utilização de tecnologias abertas, configuração remota dos dispositivos sem a necessidade de atuação física por parte do utilizador, arquitetura flexível e extensível. O aplicativo é capaz de se comunicar com dispositivos via protocolo MQTT, de modo que é possível coletar informações como umidade e temperatura e executar ações. Por exemplo: é possível controlar o acionamento de um motor quando o sensor de temperatura indicar uma temperatura abaixo do que informado pelo usuário.

3.2 ARQUITETURA

A arquitetura geral é dividida em duas áreas, a área do usuário, composta por tudo que o usuário final do aplicativo consegue ver/interagir e, a área da infraestrutura, composta pelo aplicativo desenvolvido neste trabalho. A área do usuário é representada pelos dispositivos IoT e pela interface de gestão do aplicativo, já a área da infraestrutura, é representada pelo broker de mensageria, pela aplicação de gestão, pela aplicação de processamento e pelo banco de dados, ambas as áreas são retratadas na Figura 1 que apresenta também, o fluxo de comunicação entre os componentes. Durante este capítulo, será exposta a funcionalidade dos componentes pertencentes da área da infraestrutura.

Figura 1 – Arquitetura da Aplicação



Fonte: elaborado pelo autor.

3.2.1 Aplicação de Gestão

A aplicação de gestão disponibiliza via *Application Program Interface* (API), uma interface de controle e gestão. Os principais recursos que API disponibilizada são, controle de conta, gestão de dispositivos, gestão de fluxos e por fim, listagem e acionamento de botões. O controle de conta, permite o cadastro de um novo usuário ou autenticação de um usuário já existente. Ao realizar a autenticação de um usuário já existente, a API retorna um *Json Web Token* (JWT) que deverá ser utilizado nas demais requisições efetuadas para a API. A gestão dos dispositivos, permite efetuar a listagem dos dispositivos, a remoção de um dispositivo e o cadastro de um novo dispositivo. A gestão de fluxos permite a listagem dos fluxos existentes, a remoção de um fluxo, a adição de um gatilho no fluxo e a adição dos demais nodos de ação. A listagem de botões lista todos os fluxos registrados onde, o gatilho do fluxo, é um gatilho do tipo botão

3.2.2 Aplicação de Interface do Usuário

Se fez o uso das técnicas apresentadas na sessão 2.2 para criar uma aplicação PWA. A aplicação realiza o consumo da API da aplicação de gestão, apresentando os recursos em formulários de preenchimento. Há a possibilidade de realizar uma conexão via WebSocket diretamente com o broker de mensageria, dado que a conexão com o broker esteja ativa, o dispositivo passará a ser reconhecido pelo broker, podendo então receber mensagens diretas.

A aplicação, além do acesso a API de gestão e ao broker de mensageria, traz a possibilidade de acesso a APIs nativas do dispositivo. As APIs são: *Web Bluetooth*, que realiza pareamento com dispositivos BLE, NFC, que realiza escrita e leitura de RFIDs, *Notification*, que permite o envio de mensagens de notificação para o sistema operacional e por fim o *Vibration*, faz que o dispositivo vibre, somente disponível para os dispositivos móveis.

3.2.3 Broker de Mensageria

Se utilizou o broker de mensageria RabbitMQ apresentado na sessão 2.3.1, dentro da aplicação o broker mantém três principais funções. A primeira função realizada pelo broker é ser o centralizador de todas as mensagens enviadas via protocolos MQTT, AMQP e WebSocket. Segunda função realizada pelo broker é fazer com que todas as mensagens sejam enviadas para o processador de fluxos. Terceira e última função realizada pelo broker é permitir interoperabilidade entre os protocolos MQTT, AMQP e WebSocket.

3.2.4 Processador de Fluxos

Escuta todas as mensagens enviadas para o broker de mensageria, caso uma das mensagens obedeça aos critérios de ativação do gatilho do fluxo, o processador criar um contexto de execução, preenche o contexto de execução com os dados obtidos da mensagem, inicia a execução do fluxo passando o contexto então criado.

3.2.5 Banco de Dados

Se utilizou o banco de dado PostgreSQL apresentado na sessão 2.3.2, utilizado para guardar de forma relacional as entidades dispostas no diagrama a seguir

(FALTA ADICIONAR O DIAGRAMA)

(FALTA DESCREVER AS TABELAS)

(REALMENE NECESSÁRIO?)

3.3 PROTOCOLO DE MENSAGEM

Para realizar a comunicação entre os dispositivos, se criou um protocolo de mensagem. O protocolo define a estrutura de campos, conteúdo tipo, nome e valor. O protocolo define que, mensagens podem possuir um ou mais campos, cada campo, deverá ter o formato tipo:descrição:valor, caso haja mais campos na mensagem, deverá ser adicionado o caractere de final de linha /n para serem diferenciados. Os tipos possíveis para os campos são: guid, bool, int, float, double e string. Protocolo garante a tolerância de duas falhas, a falha de tipo não reconhecido e a falha de campo não informado em um formato correto. Caso seja informado um tipo não reconhecido, o protocolo considera como tipo padrão, o tipo string. Caso o campo não esteja em um formato correto, o protocolo ignora o campo. O Quadro 4 demonstra a possibilidade de interoperabilidade dos tipos definidos pelo protocolo e as linguagens de programação JAVA, CSHARP, C++ e GO.

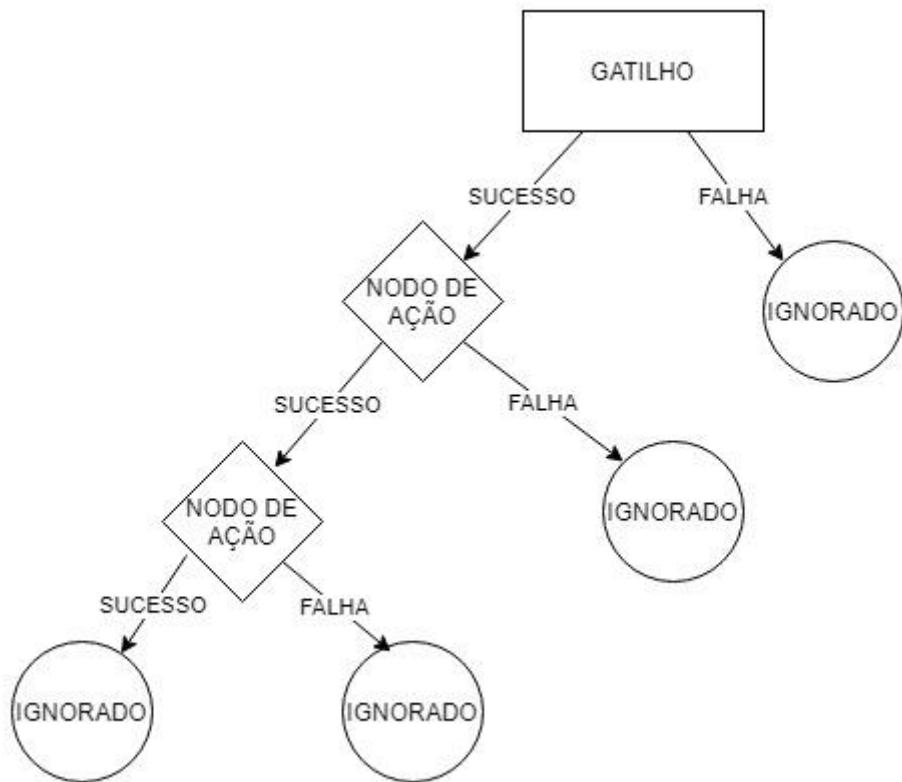
Quadro 4 – Mapa de interoperabilidade dos tipos

| PROTÓCOLO | JAVA | CSHARP | C++ | GO |
|-----------|---------|--------|-------------|---------|
| guid | UUID | guid | N/A | N/A |
| bool | boolean | bool | bool | bool |
| int | int | int | int32_t | int32 |
| float | float | float | float | float32 |
| double | double | double | double | float64 |
| string | string | string | std::string | string |

3.4 O FLUXO

O fluxo é a entidade principal deste trabalho. A elaboração do fluxo foi concebida em uma estrutura de dados chamada de árvore de decisão, que é uma estrutura muito similar a uma árvore binária, porém se difere na capacidade de um nodo realizar decisão de caminho. Em definição a árvore binária é composta por nodos possuindo o mesmo algoritmo de decisão, já a árvore de decisão permite que os nodos possuem diferentes algoritmos. Outro aspecto da árvore de decisão utilizada pelo fluxo, é a criação de um contexto pelo primeiro nodo da árvore, e o repasse deste contexto, para a execução do próximo nodo, tal contexto será explicado na sessão 3.4.3 deste capítulo. A Figura 2, expressa uma árvore de decisão utilizada pelo fluxo, se nota que o primeiro nodo possui o nome de GATILHO, já os demais NODO DE AÇÃO ambos também serão abordados nas sessões 3.4.1 e 3.4.2, respectivamente.

Figura 2 – Árvore de Decisão de um Fluxo



Fonte: elaborado pelo autor

3.4.1 Gatilhos – *Trigger*

O gatilho é o primeiro nodo da árvore de decisão, são duas as responsabilidades do gatilho, a primeira é validar se a execução da árvore de decisão é necessária e a segunda é realizar a criação de um contexto de execução, qual será repassado para os próximos nodos.

A aplicação define dois principais gatilhos, o gatilho MQTT, qual somente é disparado caso uma mensagem MQTT seja processada pelo processador de fluxos e o gatilho BUTTON, este gatilho somente é disparado por ação explícita do usuário.

3.4.2 Nodos de Ação – *Action Nodes*

Os nodos de ação são encarregados de realizarem execuções arbitrárias, podendo ou não, acessar e alterar o contexto de execução, qual foi repassado pelo nodo pai. Após a execução da ação, o nodo é responsável por realizar a chamada de um dos dois nodos filhos, sendo eles o nodo de sucesso, quando a execução ser concluída conforme o esperado, ou falha, quando houver falha na execução

A aplicação define sete ações distintas, sendo eles HTTP POST, HTTP GET, ATTRIBUTE, MQTT PUBLISH, DECISION, GPIO e por fim GUARD.

A ação HTTP POST, realiza o envio de uma requisição HTTP utilizando o verbo POST, contendo no corpo da mensagem, um campo disponível no contexto da aplicação qual foi selecionado pelo usuário.

A ação HTTP GET, realiza o envio de uma requisição HTTP utilizando o verbo GET, a resposta da mensagem é salva no contexto da aplicação em um campo selecionado pelo usuário.

A ação ATTRIBUTE, adiciona um campo no contexto da execução, dado um tipo, dado um valor e dado um nome selecionados pelo usuário.

A ação MQTT PUBLISH realiza a publicação de uma mensagem utilizando o protocolo MQTT, para o tópico do dispositivo selecionado pelo usuário utilizando como corpo da mensagem, um campo selecionado pelo usuário.

A ação DECISION executa uma validação usando uma das expressões Equal, NotEqual, GreaterThan, GreaterThanOrEqual, LessThan e LessThanOrEqual, entre dois valores contidos no contexto da execução, selecionados pelo usuário.

A ação GPIO, realiza a publicação de uma mensagem utilizando o protocolo MQTT, para um dispositivo selecionado pelo usuário, tal mensagem possui o identificador da gpio e o seu status desejado, dentre os status o usuário tem a possibilidade de escolha entre High, Low e Toggle (qual alterna entre High e Low).

A ação GUARD, valida a existência de um campo no contexto da execução, caso o campo não esteja presente no contexto, é executada a ação de falha

3.4.3 Contexto – *Context*

O contexto é uma estrutura de dados chamada de dicionário. Tal estrutura é utilizada no modelo de chave e valor, sendo a chave um texto qualquer e o valor um valor dinâmico qualquer, ou seja, pode-se definir inúmeros campos de diferentes tipos. O contexto é fundamental para realizar o repasse de informações entre os nodos da árvore de decisão, fazendo com que dados, por exemplo, do nodo HTTP GET, possam ser utilizado em um nodo subsequente.

3.5 DISPOSITIVOS

Homologação da aplicação se dá pela comunicação entre dispositivos, para isso se adquiriu os dispositivos Esp8288, Arduino Uno, Sonoff e Bluetooth HM-10 com a intenção de validar a aplicação como um todo. Nos próximos tópicos, são apresentados de forma

3.5.1 ESP8266

O ESP8266 é um microcontrolador produzido pela empresa Espressif Systems. Esse micro controlador possui um sistema de comunicação WiFi próprio, que é o seu grande diferencial, por esse motivo ele é largamente utilizado como módulo WiFi para outros micro controladores, como o Arduino, por exemplo, apesar de possuir um processador próprio e de ser possível utilizar somente o ESP8266 para criar sistemas embarcados (OLIVEIRA, 2017).

3.5.2 Arduino Uno

O Arduino é uma plataforma eletrônica de código aberto, que possui seus próprios micro controladores e a sua própria IDE. O Arduino tem uma grande comunidade de usuários, diversos trabalhos e projetos que se utilizam dessa plataforma, pela sua facilidade de uso, eficiência e sua IDE gratuita, além do grande número de ferramentas já implementadas para o mesmo (OLIVEIRA, 2017).

3.5.3 Sonoff

O Sonoff é baseado no ESP8266 e o mesmo possui um firmware que é capaz de fazer com que o módulo se conecte ao servidor na nuvem através de uma rede WiFi, desta forma permite que todo o controle sobre a placa possa ser feito remotamente (MASTERWALKERSHOP, 2018).

3.5.4 Bluetooth HM-10

O HM-10 é um dispositivo Bluetooth 4.0 disponível com base no sistema *Bluetooth Low Energy* (BLE) (MIT, 2018). O BLE funciona de uma maneira muito diferente do Bluetooth convencional. O BLE é projetado para aplicações de baixa energia e alcança isso, usando pequenos pacotes de dados enviados com pouca frequência (MIT, 2018).

3.6 PROTOCOLOS DE TRANSPORTE

Os protocolos de transporte permitem que mensagens sejam entregues entre dispositivos. Um fato importante na escolha destes protocolos, é a possibilidade de tradução dos protocolos. Este capítulo identifica os protocolos utilizados e suas funções dentro da aplicação.

3.6.1 AMQP

O protocolo AMQP, atende o fato de não ser um protocolo proprietário, nem limitado quando se trata do aspecto da possibilidade de uso entre diferentes aplicações como explica (ADLINK, 2017, p.6, tradução nossa) “AMQP é um protocolo qual tem o foco principal em transmitir mensagens [...], em uma solução de uso comum e não proprietário permitindo interoperabilidade.”. Este protocolo possui algumas simples características, tais qual, o fato do transporte da mensagem ser em formato binário, acarretando em rapidez e, o fato do protocolo ser acessível sem abstração, diretamente pela aplicação, adicionando flexibilidade e interoperabilidade em seu uso, como descrito (ADLINK, 2017, p.6, tradução nossa) “AMQP foi desenhado para suportar padrões de variados protocolos de comunicação existentes”. Outros atrativos do protocolo, seriam o fluxo da transmissão das mensagens em um modelo

de produtor/consumidor e os tipos de garantia de entrega disponíveis, (ADLINK, 2017, p.6, tradução nossa) “[...] no máximo uma vez (mensagem pode ou não ser entregue, porém, somente uma vez), ao mínimo uma vez (mensagem é entregue ao menos uma vez, porém, pode ser entregue mais de uma vez) e exatamente uma vez (mensagem é entregue uma única vez).”

O protocolo AMQP é utilizado como protocolo base de comunicação entre as aplicações da infraestrutura. A aplicação de processamento, consome as mensagens, realiza o processamento do fluxo, caso seja definido no fluxo, também realiza a publicação de uma mensagem.

3.6.2 MQTT

É um protocolo de publicação de mensagens de assinatura. Foi aprovado como um padrão ISO em 2015. É um protocolo muito leve e flexível. Requer apenas 2 bytes de informação de cabeçalho. O MQTT é uma arquitetura baseada em broker. O broker consiste em uma lista de todos os tópicos atualmente disponíveis. Todas as mensagens precisam passar pelo broker. Um dos brokers mais comuns é o Mosquitto, mas para a aplicação deste trabalho, foi utilizado o broker RabbitMQ. Um sistema MQTT típico consiste em um único ou vários brokers. O protocolo MQTT foi desenhado para ser utilizado em locais onde há alta latência, ou seja, o envio de mensagens ponta a ponta pela Internet é demorada (ADLINK, 2017, p.6, tradução nossa) “MQTT foi desenhado para transmitir mensagens em redes limitadas por velocidade.”

Protocolo MQTT foi utilizado neste trabalho, como protocolo de comunicação entre dispositivos IoT e o broker de mensageria. Ao chegar no broker de mensageria, as mensagens utilizando o protocolo MQTT são traduzidos para o protocolo base AMQP, permitindo o consumo das mensagens pela aplicação de processamento. A tradução inversa do protocolo AMQP para MQTT também ocorre, quando a aplicação de processamento necessita realizar o envio de uma mensagem para os dispositivos IoT.

3.6.3 HTTP

Hypertext Transfer Protocol (HTTP) é um protocolo de camada de aplicação para transmitir documentos hipermédia, como HTML. Ele foi projetado para comunicação entre navegadores da Web e servidores da Web, mas também pode ser usado para outras finalidades (MOZILLA, 2018). O HTTP segue um modelo clássico de cliente-servidor, com um cliente abrindo uma conexão para fazer uma solicitação e aguardando até receber uma resposta (MOZILLA, 2018).

Por ser um protocolo unidirecional, não há a possibilidade de realizar a tradução entre o HTTP e os protocolos AMQP, MQTT e MQTT sobre WebSocket, estre protocolo então, não é utilizado para realizar comunicação entre os dispositivos e o broker de mensageria, mas sim, se faz o uso do protocolo para realizar a comunicação da aplicação de interface do usuário e a aplicação de gestão.

3.6.4 MQTT sobre WebSocket

O WebSocket é uma tecnologia avançada que possibilita a abertura de uma sessão de comunicação interativa bidirecional entre o navegador do usuário e um servidor, dando a possibilidade de envio de mensagens para o servidor (MOZILLA, 2018)

Como o WebSocket dá a habilidade de comunicação bidirecional com o servidor, qual é similar ao protocolo MQTT, se tem a possibilidade de realizar interoperabilidade entre os dois protocolos, fazendo com que mensagens do protocolo MQTT sejam enviadas via Websocket e vice-versa. Este protocolo é utilizado pela aplicação

4 RESULTADOS

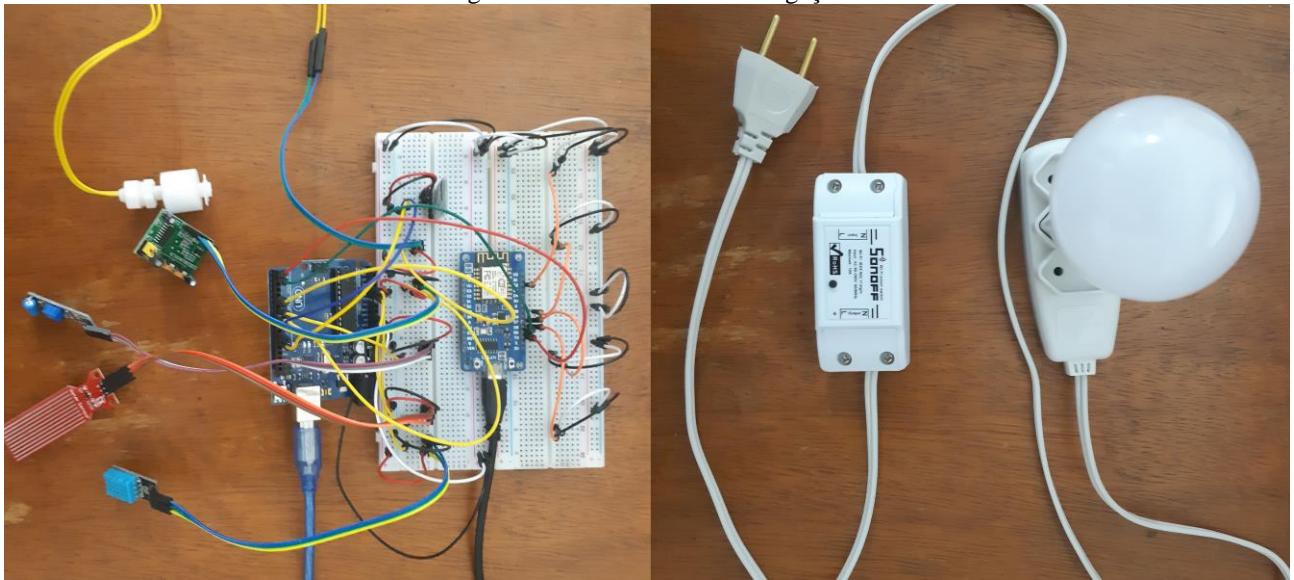
Neste capítulo são expostos os resultados do trabalho, ou seja, o software desenvolvido no Capítulo 3 DESCRIÇÃO DA APLICAÇÃO. Como produto principal deste trabalho, apresenta-se o código fonte. O código pode ser visualizado por meio da ferramenta Github, onde encontra-se hospedado os quatro repositórios relacionados ao projeto (adicionar links).

Primeiramente, é demonstrado o conjunto de Dispositivos de automação utilizados neste capítulo. Em seguida, demonstra-se a Interface de Usuário da aplicação por meio de descrições e capturas de tela. Por fim, se apresenta alguns casos de uso hipotéticos, onde uma aplicação de automação de fluxos e mensagens pode vir a ser utilizado

4.1 DISPOSITIVOS DE AUTOMAÇÃO UTILIZADOS

Como já indicado pela sessão 3.4, se adquiriu alguns dispositivos para realizar a homologação da aplicação. A Figura N, apresenta o resultado final da utilização destes dispositivos em uma bancada de homologação. Ao longo deste capítulo, será explicado a montagem da bancada separados em seus principais componentes.

Figura N – Bancada de Homologação



Fonte: elaborado pelo autor.

4.1.1 Arduino UNO

Dispositivo atua como interface de coleta de dados sensoriais, normalizando os dados recebidos dos sensores/componentes eletrônicos no protocolo de mensagem definido na sessão 3.3. Após a normalização, o dispositivo envia a mensagem utilizando dois barramentos distintos. O primeiro barramento mantém uma interface de comunicação com o dispositivo Esp8266 descrito na sessão 4.1.2, neste barramento são enviados dados sensoriais unitários, intercalando entre temperatura, humidade relativa do ar, profundidade do nível de água, vibração, movimentação e por fim nível de água. O segundo barramento mantém uma interface de comunicação com o dispositivo Bluetooth HM-10 descrito na sessão 4.1.4, neste barramento são enviados todos os dados sensoriais coletados.

Para realizar a coleta, se faz o uso de distintos componentes eletrônico. O componente DHT-11 realiza a coleta dos dados de humidade relativa do ar e de temperatura. O componente *Passive Infrared Sensor* (PIR) realiza a coleta de movimentação no ambiente. O componente SW-420 identifica se está ocorrendo vibração no local de sua fixação.

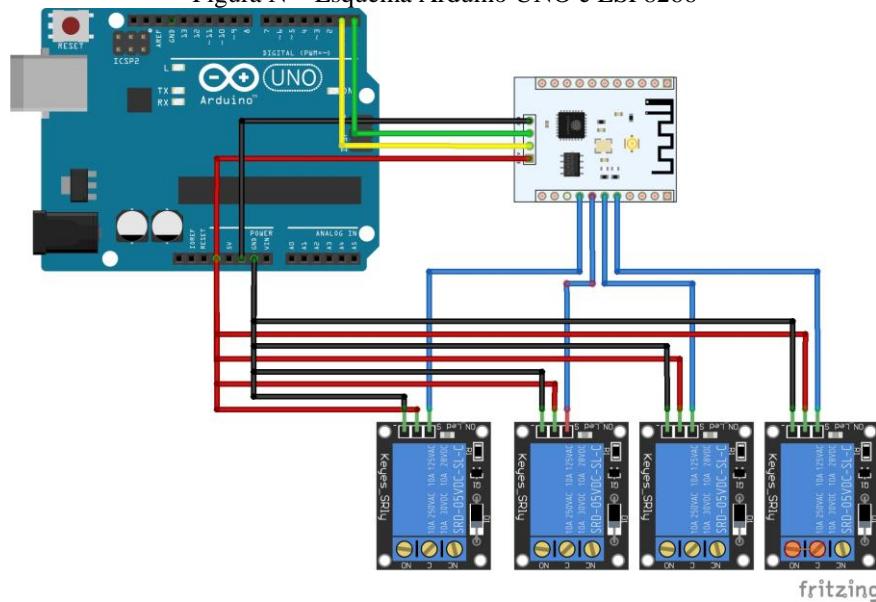
Se faz necessário obedecer ao esquema de ligação demonstrado na Figura N, também se deve realizar o envio do firmware para o Arduino UNO, disponível em código fonte no APENDICE B.

(FALTA IMAGEM DO DIAGRAMA DO ARDUINO E MODULOS DE SENSORES)

4.1.2 ESP8266

Mantém uma interface de comunicação com o dispositivo Arduino Uno. Tem como função, realizar a ponte de comunicação entre mensagens recebidas via interface e o *broker MQTT*, também age como atuador em suas portas GPIO (entra e saída de propósito geral), ao receber mensagens de liga/desliga qual, identifica a porta GPIO de atuação e seu status. A Figura N indica as conexões necessárias entre o ESP8266 e o Arduino UNO, e entre o ESP8266 e os interruptores elétricos. Após a montagem, também existe a necessidade de realizar o envio do firmware para o ESP8266 disponível em código fonte disposto no APENDICE C.

Figura N – Esquema Arduino UNO e ESP8266

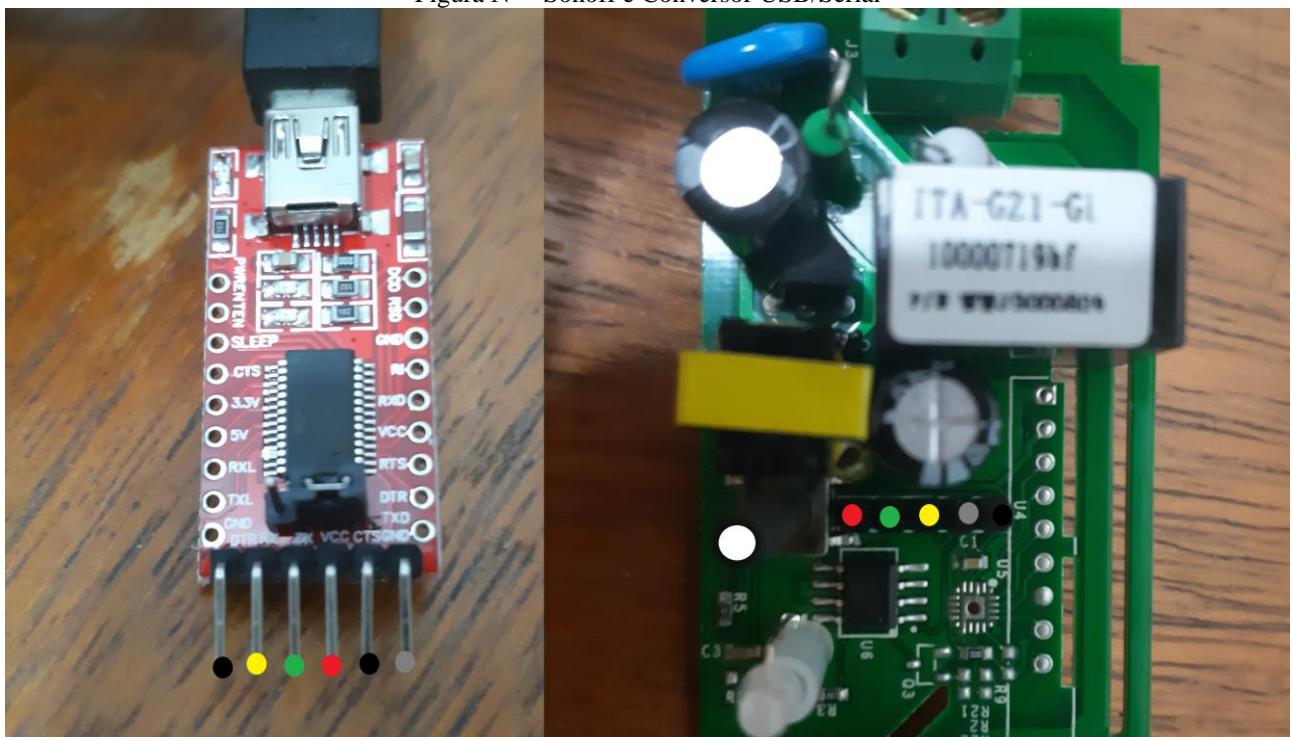


Fonte: elaborado pelo autor

4.1.3 Sonoff

Realiza a comunicação com o *broker MQTT*, age somente como atuador em sua GPIO principal, ao receber mensagens de liga/desliga permitindo ou não a passagem de corrente elétrica. Como indicado na sessão 3.4.3, o dispositivo Sonoff vem previamente instalado com um firmware proprietário, como se trata de um Esp8266, houve a troca do firmware para atender a aplicação neste trabalho. A Figura N demonstra as conexões de um conversor USB/Serial e o Sonoff, os círculos coloridos identificam a ligação entre os pinos dos dois dispositivos, os pinos identificados em cor preta, devem ser ignorados. Essas conexões permitem o envio do firmware disponível em código fonte que está disponível no APENDICE D.

Figura N – Sonoff e Conversor USB/Serial

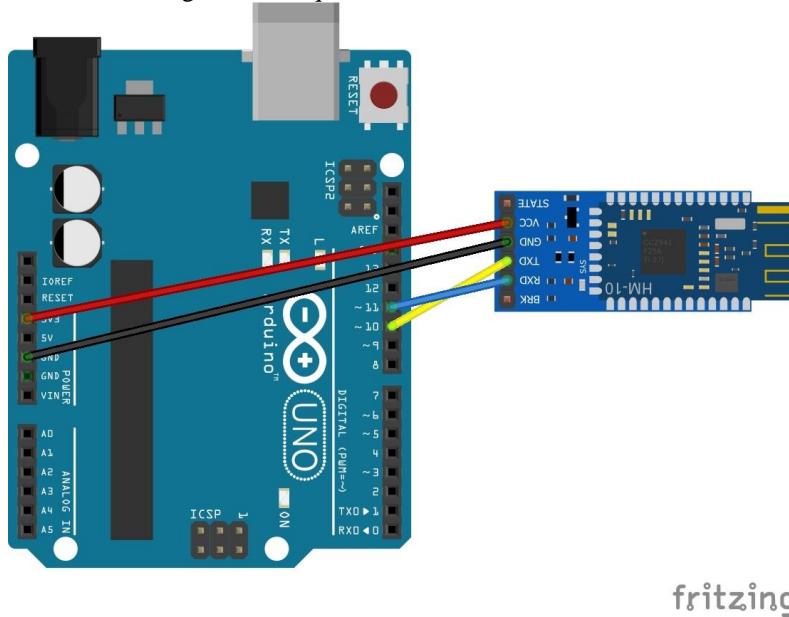


Fonte: elaborado pelo autor

4.1.4 Bluetooth HM-10

Mantém uma interface de comunicação com o dispositivo Arduino Uno, realizando a ponte da mensagem recebida via interface para o dispositivo pareado. A Figura N demonstra o esquema de conexão entre o Arduino UNO e o HM-10, ao contrário dos outros dispositivos, não há a necessidade de realizar alterações no firmware do HM-10.

Figura N – Esquema Arduino UNO e HM-10



fritzing

Fonte: elaborado pelo autor.

4.2 DEMONSTRAÇÃO DA APLICACAO DE INTERFACE DO USUÁRIO

APENDICE E

4.2.1 Pagina Inicial – Home

Corresponde a página de cadastro e autenticação do usuário. Autenticação se faz necessária para realizar a gestão de Conta, descrita na sessão 4.3.2, dos Dispositivos, descrito na sessão 4.3.3, dos Fluxos descrito na sessão 4.3.4, dos Botões descrito na sessão 4.3.5 e do MQTT descrito na sessão 4.3.6

4.2.2 Conta – Account

Demonstra os dados da conta do usuário, apresentando o identificador único do usuário no campo identificado como Id, o nome escolhido pelo usuário no campo identificado como Name e o identificador único para integração com a aplicação no campo identificado como Token.

4.2.3 Dispositivos –Devices

Permite o cadastro de um novo dispositivo, utilizando o formulário identificado como New Device, informando o nome desejado no campo identificado como Name. Ao se realizar o cadastro de um novo dispositivo, a aplicação será responsável por atribuir um identificador único, chamado de token. O token deve ser utilizado como nome de registro no broker de mensageria. Lista os dispositivos cadastrados pelo usuário, cada dispositivo dispõe de um nome e um identificador único.

4.2.4 Fluxos – Flows

Permite o cadastro de um novo fluxo, utilizando o formulário identificado como New Flow, informando o nome desejado no campo identificado como Name. Lista os fluxos existentes para o usuário corrente, permitindo sua edição no botão identificado como EDIT ou sua remoção no botão identificado como DELETE

4.2.5 Fluxo – Fluxo

Edita o fluxo selecionado, permitindo a escolha de um gatilho, subsequentemente permite a adição das próximas ações a serem executadas, definindo se ação deverá ser executada em falha identificado pelo botão FAIL ou em sucesso identificado pelo botão PASS

4.2.6 Botões – Buttons

Lista os gatilhos descritos como *Button*, permitindo o acionamento de um gatilho no botão identificado como **TRIGGER**

4.2.7 MQTT

Permite o registro do dispositivo no *broker* de mensageria, podendo assumir a identidade de qualquer outro dispositivo cadastrado. Para iniciar a comunicação com o *broker* se faz necessário acionar o botão identificado como **START**. Lista as mensagens enviadas para o dispositivo

4.2.8 Bluetooth

Permite a conexão a um dispositivo via protocolo *bluetooth*. Lista as mensagens enviadas do dispositivo pareado via protocolo *bluetooth*.

4.2.9 NFC

Permite a leitura e escrita de identificadores de rádio frequência (RFID). Lista as mensagens enviadas via *broker*, depende do registro descrito na sessão 4.3.7. Se for desejado realizar a leitura de um RFID, se deve acionar o botão identificado como **READ**. Se for desejado realizar a escrita de um RFID, se deve acionar o botão identificado como **WRITE**

4.2.10 Notificações - Local Notifications

Permite que requisições de notificação sejam enviadas para o sistema operacional nativo. Depende do registro descrito na sessão 4.3.7

4.2.11 Vibração – Vibration

Permite que requisições de vibração sejam enviadas para o sistema operacional nativo. Depende do registro descrito na sessão 4.3.7

4.3 CASOS DE USO HIPOTÉTICOS

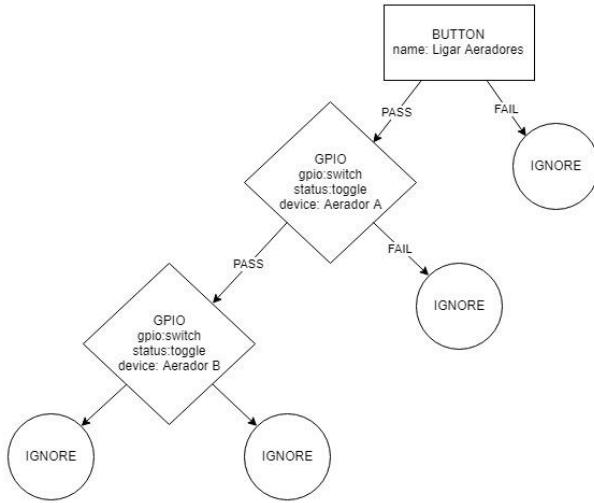
4.3.1 Automação agrícola

Segundo (SEMANTIX, 2018) normalmente, o setor agrícola é mais propenso a riscos. Muitos fatores, que variam desde imprevistos de irrigação e previsões de precipitação à má qualidade do solo e métodos de colheita/plantio, podem ter efeitos adversos sobre a produtividade geral. Porem tal fator pode ser mitigado com a utilização da IoT, em afirmação (SEMANTIX, 2018) alega que agricultores podem evitar essas situações com a ajuda da IoT. Eles podem obter dados altamente precisos e em tempo real de seus campos usando avançados agro-sensores, dependendo do tipo de decisões cruciais (“quando colher?”, “quando irrigar” etc.)

4.3.1.1 Piscicultura

MATHIAS (2018) piscicultor local de Jaraguá do Sul, levantou um processo realizado diariamente em função do manejo de cinco viveiros da cultura de tilápias, qual estão dispostos em uma propriedade distante de sua moradia. Duas vezes ao dia, no início da manhã e no final da tarde, principalmente em períodos de extremo calor, são ligados os aeradores, responsáveis por realizar a oxigenação da água, evitando a perca da cultura por morte. MATHIAS (2018) questionou a possibilidade de realizar o processo de liga e desliga dos aeradores remotamente. Sobre o caso do uso descrito por MATHIAS (2018), se apresenta como solução hipotética, a configuração do fluxo ilustrada na Figura N.

Figura N – Solução Piscicultura com Button

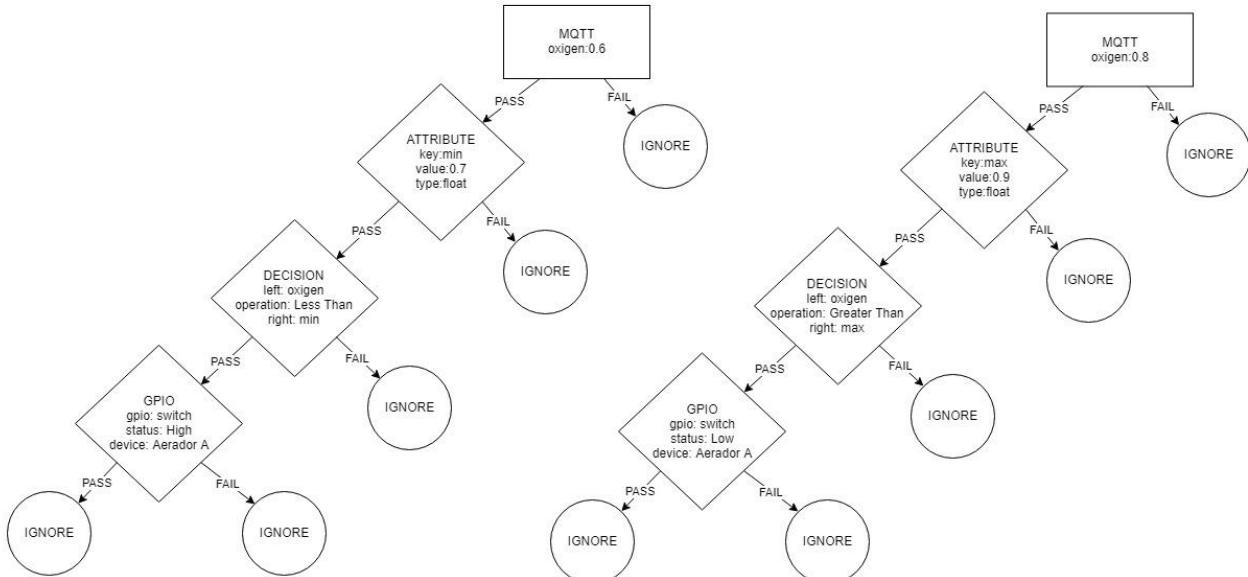


Fonte: elaborado pelo autor.

Solução apresentada na Figura N, declara a configuração de um fluxo onde um gatilho BUTTON ativa o fluxo de envio de mensagens para alterar o status de uma GPIO para dois dispositivos. O diagrama e o código fonte desta solução estão dispostos no APEDICE F

Solução apresentada acima pode ser ainda melhorada, (Zaccharias e Rocha, 2016) indicam um sensor com a capacidade de realizar a coleta do oxigênio dissolvido em água, com base nos valores apresentados por (Zaccharias e Rocha, 2016) a média ideal de oxigênio dissolvido para a cultura de tilápias é de 0,8mg/l, como possível configuração para esta solução, há a necessidade de dois fluxos, um fluxo para execução da liga dos aeradores, outro fluxo para a execução da desliga dos aeradores, conforme apresentado na Figura N

Figura N – Solução Piscicultura via Métrica



Fonte: elaborado pelo autor.

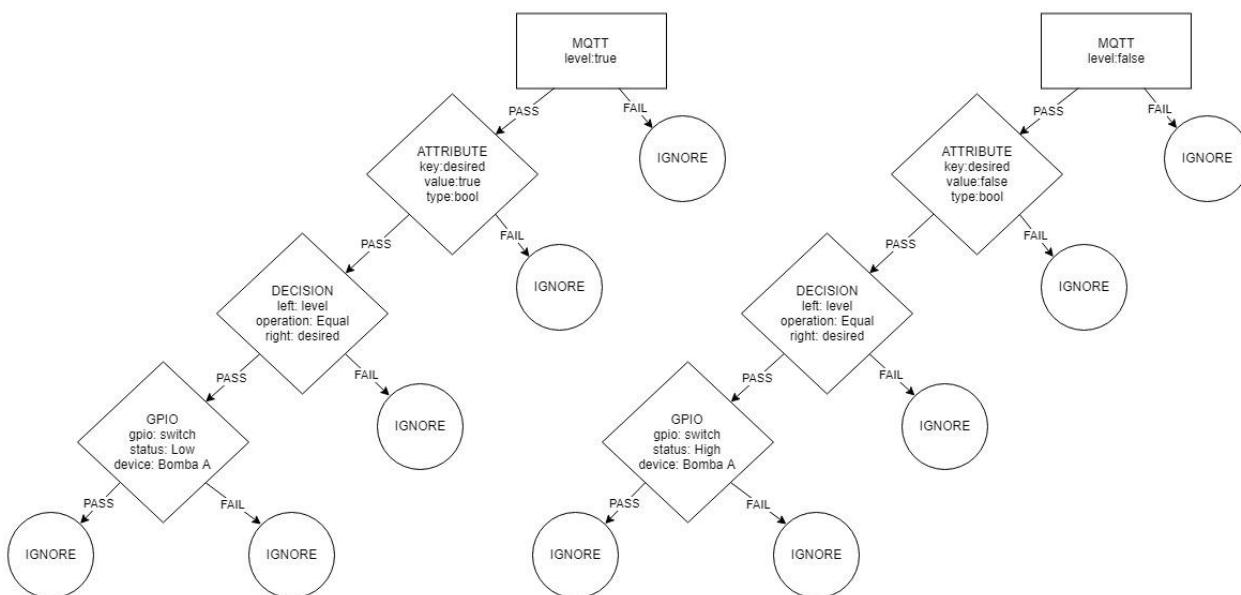
Solução apresentada na Figura N, declara a configuração de dois fluxos. O fluxo da esquerda, responsável por realizar a liga do aerador, declara um gatilho MQTT contendo o campo oxigen, em seguida é adicionado o campo min contendo o valor mínimo desejado para o nível de oxigênio, ao realizar a comparação dos dois valores no próximo passo, caso o nível de oxigênio esteja abaixo, é realizada a liga do aerador. O fluxo da direita, responsável por realizar a desliga, é muito similar, declara um gatilho MQTT contendo o campo oxigen, adiciona um campo max contendo o valor máximo desejado para o nível de oxigênio, realiza a comparação dos dois valores, caso a comparação seja verdadeira, realiza a desliga do aerador. O diagrama dos dispositivos e o código fonte desta solução estão dispostos no APENDICE G.

4.3.1.2 Cultivo de arroz irrigado

GREUEL (2018) produtor local de Jaraguá do Sul, levantou um caso de uso no cultivo de arroz irrigado, segundo GREUEL (2018) se deve manter uma lâmina de água idealmente na altura de 8cm do caule do arroz, para isto, a solução elaborada por ele, envolve um bueiro de controle que tem como função manter a lâmina na altura desejada e a irrigação continua de água, mitigando a evaporação. GREUEL (2018) criticou a solução atual, o bueiro eventualmente entope, gerando manutenção física e a irrigação gera custos financeiros em eletricidade.

Fortalecendo as afirmações de GREUEL (2018) o trabalho de (Stone, 2015) avalia como uma lâmina ideal, uma altura entre 5cm até 10cm, reforça que a evapotranspiração, deve ser contada também, no cálculo de vasão necessária de água para se manter a lâmina. No trabalho de (Stone, 2015) são realizados cálculos complexos para determinar a vasão da água necessária para manter a lâmina em altura ideal, como melhor proposta, a Figura N apresenta de forma simplificada o controle de nível de água, fazendo o uso de um dispositivo contendo um sensor de boia, qual indica se a nível de água foi ou não atingido

Figura N – Solução Arroz Irrigado



Fonte: elaborado pelo autor.

Solução muito similar ao apresentado na sessão 4.3.1.1, porém, ao invés de realizar a comparação de uma métrica, a indicação do status da boia, estando submersa level:true ou não level:false, já é o suficiente para realizar a liga ou desliga do sistema de irrigação. O diagrama dos dispositivos e o código fonte desta solução estão dispostos no APENDICE H.

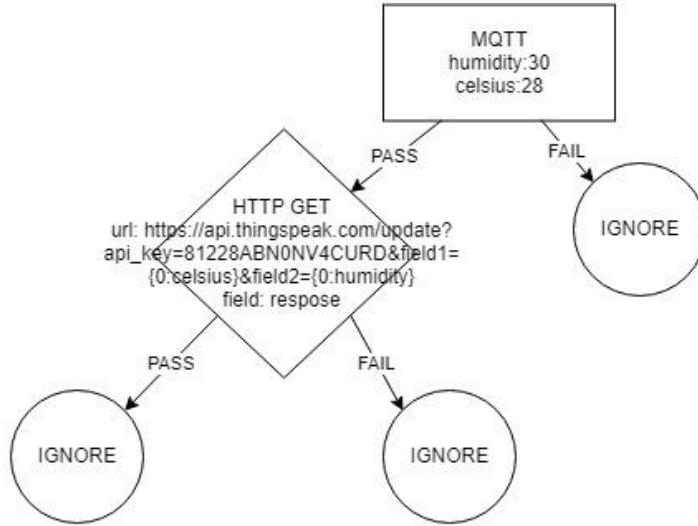
4.3.2 Automação industrial

Segundo (POLLUX, 2018) Entre as principais vantagens oferecidas pela aplicação da Internet das coisas na indústria, se destacam o aprimoramento de diagnósticos, onde a coleta de dados torna possível detectar e analisar fatores internos e externos da produção, elaborando uma análise completa do processo identificando perdas e ganhos.

4.3.2.1 Fiação de algodão

Controle de humidade e temperatura em fiações de algodão, segundo FAGUNDES (2018), é o principal fator de garantia da qualidade na produção de um fio de algodão, tornando primordial o uso controlado de humidificadores e ventiladores. Em seu parque, FAGUNDES (2018), já dispõem de uma central de controle de humidade e calor, porém indaga que para o mesmo lote de algodão com as mesmas condições na central de controle, o fio é produzido com qualidades variadas. Para melhor identificar a causa da variação na qualidade dos fios, foi proposto realizar a coleta de humidade e temperatura, dispondo os valores em um gráfico, de forma serial. A solução para o caso, faz o uso de uma plataforma de coleta de métrico externo, a plataforma se chama ThingSpeak, o fluxo é apresentado na Figura N.

Figura N – Solução Fiação de Algodão



Fonte: elaborado pelo autor.

O fluxo definido na Figura N demonstra o consumo de uma mensagem MQTT e o uso de dos campos dessa mensagem em uma integração externa, mostrando a possibilidade de integração com plataformas/ferramentas variadas. O diagrama dos dispositivos e o código fonte desta solução estão dispostos no APENDICE I.

5 CONCLUSÕES

Este é o capítulo de fechamento do trabalho, onde são discutidos os resultados obtidos, as decisões de implementação, as dificuldades superadas durante o desenvolvimento e projeto, e ainda são sugeridos alguns trabalhos futuros. Como principal forma de avaliação, se fez o levantamento de casos de uso, apresentando suas implementações de forma que possam ser reproduzidas, mesmo que em ambiente controlado. A funcionalidade da aplicação de fluxos e decisões, se tornou capaz de contemplar variados procedimentos em que, dado uma coleta, há a necessidade de realizar uma decisão seguida de tomada de ação, ao exemplo dos casos de uso em ambiente agrário, apresentados na sessão 4.3.1. A aplicação também se mostrou apta, ao realizar integração com uma plataforma externa, como contata a sessão 4.3.2. Outra meta de avaliação, foi atingida com a validação da portabilidade da aplicação de interface do usuário, a possibilidade de realizar a instalação do aplicativo de forma nativa, fazendo com que funcionalidades como BlueTooth e NFC fossem também acessíveis, além de possibilitar seu uso, sem a necessidade de uma conexão ativa com a internet, reforça a viabilidade do uso de uma PWA como substituto de uma aplicação nativa. Com a apresentação da bancada de homologação, se pode provar que a solução abrange uma variedade considerável de casos de uso além mostrar que as definições de fluxo aqui apresentadas podem ser aplicadas.

No entanto, como o broker de mensageria age como um centralizador e, estando somente disponível para os dispositivos conectados à rede mundial de computadores, o broker de mensageria se torna o principal ponto de falha. Outro caso é a falta de alguns nodos de ações que poderiam facilitar a configuração de alguns fluxos. Como extensões possíveis para este trabalho, indico as seguintes opções:

- f) Alterar a arquitetura para que a execução do fluxo possa ocorrer em uma rede local ou internamente nos dispositivos;
- g) Desenvolver novos gatilhos e novos nodos de ação, de forma a atender novos casos de uso;
- h) Definir de maneira mais detalhada o protocolo de mensagem, adicionando novos tipos de interesse ao IoT, por exemplo um tipo gpio, definindo os estados possíveis deste tipo
- i) Desenvolver um firmware universal compatível com o protocolo de mensagem
- j) Adicionar a possibilidade de realizar o envio do firmware para o dispositivo utilizando a aplicação de interface de usuário e a API WebUSB

REFERÊNCIAS

ADLINK, A Comparison Between DDS, AMQP, MQTT, JMS, REST, CoAP, and XMPP, 2017. Disponível em: <<http://www.prismtech.com/sites/default/files/documents/Messaging-Whitepaper-051217.pdf>>. Acesso em 01 de abr. 2018.

APPLE Introduction to HomeKit - Apple Developer, California, 2018. Disponível em:
<https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/HomeKitDeveloperGuide/Introduction/Introduction.html#/apple_ref/doc/uid/TP40015050>. Acesso em 01 de abr. 2018.

ref <https://becominghuman.ai/understanding-decision-trees-43032111380f> (página da internet com autor)

CERN, Statement concerning CERN W3 software release into public domain, CERN, 1993. Disponível em:
<<https://cds.cern.ch/record/1164399>>. Acesso em 01 de abr. 2018.

FLORES Carmen Montano, LUNDMARK Mattias e MÄHR Wolfgang, Control vs Convenience: Critical Factors of Smart Homes. 2005. 4f. Artigo - IT University of Göteborg, Chalmers Forskningsgången, Göteborg, 2005.

FRANSSON Rebecca e DRIAGUINE Alexandre et al. Comparing Progressive Web Applications with Native Android Applications. 2017. Trabalho de Conclusão de Curso - Linnaeus University, Faculty of Technology, Department of Computer Science, 2017.

MOZILLA, Web technology for developers, 2018, <https://developer.mozilla.org/en-US/docs/Web>. Acesso em 01 de abr. 2018.

ref <https://developer.mozilla.org/en-US/docs/Web/Manifest> (página da internet com autor)

ref https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API (página da internet com autor)

ref <https://developer.mozilla.org/en-US/docs/Web/API/Cache> (página da internet com autor)

ref https://developer.mozilla.org/en-US/docs/Web/Apps/Progressive/Add_to_home_screen (página da internet com autor)

ref https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

ref <https://developer.mozilla.org/en-US/docs/Web/HTTP>

ref <https://www.postgresql.org/about/> (página da internet com autor)

(OLIVEIRA, 2017). (ref 10)

OLHAR DIGITAL, A Importância do Open Source para o avanço tecnológico, 2016. Disponível em:
<https://olhardigital.com.br/alem_da_infra/noticia/a-importancia-do-open-source-para-o-avanco-tecnologico/61092>. Acesso em 01 de abr. 2018.

SOUZA, Marcelo Varela, Domótica de baixo custo usando princípios de IoT. 2016. 49f. Dissertação de Mestrado - Universidade Federal do Rio Grande do Norte, Natal, 2016.

VORAPOPISUT Supachai, A Lightweight Framework of Home Automation System Based on the IFTTT Model. 2015. 8f. Artigo - Thammasat University, Pathumthani, 2015.

(MIT, 2018) (ref <http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth.html>)

ref 12 Performance analysis of RabbitMQ as a Message Bus Sneha Shailesh , Kiran Joshi , Kaustubh Purandare (publicação em periódico)

(MASTERWALKERSHOP, 2018).

(ref <http://blogmasterwalkershop.com.br/automacao/conhecendo-o-sonoff-rele-wifi-para-automacao-residencial/>). (página da internet com autor)

(SEMANTIX, 2018) (ref <http://www.semantix.com.br/blog/4-maneiras-pelas-quais-iot-esta-revolucionando-agricultura/>)

(ref 14)

AUTOMAÇÃO DOS PROCESSOS DE PRODUÇÃO E
CONTROLE PARA AUMENTO DE PRODUTIVIDADE
E REDUÇÃO DE DESPERDÍCIOS NA PISCICULTURA

Renan Lacerda Zaccharias

; Rodrigo Vilela da Rocha

(ref 13)

Eficiência do Uso da Água na Cultura do Arroz Irrigado

Luís Fernando Stone

Santo Antônio de Goiás, GO 2005

(POLLUX, 2018)

(ref <https://www.pollux.com.br/blog/iot-na-industria-por-que-utilizar-e-o-que-pode-ser-feito/>)

GREUEL, Sandro. Entrevista de pesquisa de caso de uso. Entrevistador: Silvio Greuel. Jaraguá do Sul. 2018. Entrevista feita através de conversação – não publicada.

MATHIAS, Paulo. Entrevista de pesquisa de caso de uso. Entrevistador: Silvio Greuel. Jaraguá do Sul. 2018. Entrevista feita através de conversação – não publicada.

FAGUNDES, Ademir. Entrevista de pesquisa de caso de uso. Entrevistador: Silvio Greuel. Jaraguá do Sul. 2018. Entrevista feita através de conversação – não publicada.

(ref <https://www.pollux.com.br/blog/iot-na-industria-por-que-utilizar-e-o-que-pode-ser-feito/>)

[parte de um documento:]

AMADO, Gilles. Coesão organizacional e ilusão coletiva. In: MOTTA, Fernando C. P.; FREITAS, Maria E. (Org.). **Vida psíquica e organização**. Rio de Janeiro: FGV, 2000. p. 103-115.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

AMBONI, Narcisa F. **Estratégias organizacionais**: um estudo de multicasos em sistemas universitários federais das capitais da região sul do país. 1995. 143 f. Dissertação (Mestrado em Administração) - Curso de Pós-Graduação em Administração, Universidade Federal de Santa Catarina, Florianópolis.

[norma técnica:]

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências - elaboração. Rio de Janeiro, 2002a. 24 p.

_____. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002b. 7 p.

_____. **NBR 14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2011. 11 p.

[livro:]

BASTOS, Lília R.; PAIXÃO, Lyra; FERNANDES, Lúcia M. **Manual para a elaboração de projetos e relatórios de pesquisa, teses e dissertações**. Rio de Janeiro: Zahar, 1979.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

BRUXEL, Jorge L. **Definição de um interpretador para a linguagem Portugol, utilizando gramática de atributos**. 1996. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[verbete de enciclopédia em meio eletrônico:]

EDITORES gráficos. In: WIKIPEDIA, a encyclopédie libre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <http://pt.wikipedia.org/wiki/Editores_gráficos>. Acesso em: 13 maio 2006.

[artigo em evento:]

FRALEIGH, Arnold. The Algerian of independence. In: ANNUAL MEETING OF THE AMERICAN SOCIETY OF INTERNATIONAL LAW, 61, 1967, Washington. **Proceedings...** Washington: Society of International Law, 1967. p. 6-12.

[norma técnica:]

IBGE. **Normas para apresentação tabular**. 3. ed. Rio de Janeiro, 1993. 61 p. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/normastabular.pdf>>. Acesso em: 27 ago. 2013.

[artigo em periódico:]

KNUTH, Donald E. Semantic of context-free languages. **Mathematical Systems Theory**, New York, v. 2, n. 2, p. 33-50, jan./mar. 1968.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

SCHUBERT, Lucas A. **Aplicativo para controle de ferrovia utilizando processamento em tempo real e redes de Petri**. 2003. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[página da internet com autor]

SCHULER, João P. S. **Tutorial de Delphi**. Porto Alegre, [2002]. Disponível em: <<http://www.schulers.com/jpss/pascal/dtut/>>. Acesso em: 27 ago. 2013.

[página da internet sem autor]

SCHRATCH. **Program, imagine, share**. [S.I.], [2013?]. Disponível em: <<https://scratch.mit.edu/>>. Acesso em: 27 maio 2013.

[relatório de pesquisa:]

VARGAS, Douglas N. **Editor dirigido por sintaxe**. 1992. Relatório de pesquisa n. 240 arquivado na Pró-Reitoria de Pesquisa, Universidade Regional de Blumenau, Blumenau.

APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

Este apêndice é obrigatório. Nele devem constar todos os diagramas de especificação desenvolvidos no trabalho. Os diagramas devem conter legendas numeradas na sequência do artigo.

APÊNDICE B – CÓDIGO FONTE ARDUINO UNO

APÊNDICE C – CÓDIGO FONTE ESP8266

APÊNDICE D – CÓDIGO FONTE SONOFF

APÊNDICE E – IMAGENS DAS TELAS DA APLICAÇÃO DE INTERFACE DO USUÁRIO

APÊNDICE F – SOLUÇÃO PARA CASO DE USO EM PISCICULTURA UTILIZANDO BUTTONS

APÊNDICE G – SOLUÇÃO PARA CASO DE USO EM PISCICULTURA UTILIZANDO METRICAS

APÊNDICE H – SOLUÇÃO PARA CASO DE USO ARROZ IRRIGADO

APÊNDICE I – SOLUÇÃO PARA CASO DE USO FIAÇÃO DE ALGODÃO

6 DESCRIÇÃO DA FORMATAÇÃO

A seguir são apresentadas observações gerais sobre o texto do artigo do Trabalho de Conclusão de Curso (TCC). **Observa-se que esta descrição deve ser retirada do texto final.**

Na confecção do texto deve-se:

- a) usar frases curtas. Segundo Teodorowitsch (2003, p. 3), “Frases com mais de duas linhas aumentam o risco de o leitor não compreender a ideia ou de entendê-la de forma equivocada.”;
- b) usar linguagem impersonal (usar a terceira pessoa do singular) e verbo na voz ativa (a ação é praticada pelo sujeito), com conexão entre os parágrafos;
- c) não usar palavras coloquiais;
- d) não usar palavras repetidas em demasia;
- e) usar verbos no presente quando for referir-se a partes do trabalho que já se encontram disponíveis no texto;
- f) destacar palavras em língua estrangeira em itálico, conforme descrito abaixo:
 - nome de software, ferramenta, aplicativo, linguagem de programação, plataforma, empresa: não deve ser escrito em itálico (exemplos: Delphi 7, Pascal, Object Pascal, Java, JavaScript, Java 2 Micro Edition, Basic, Microsoft Visual C++, C, Windows, Linux, MySQL, Oracle, Eclipse 3.0, Enterprise Architect, Rational Rose, Microsoft, Sun Microsystems),
 - citações: o sobrenome do autor ou o nome da instituição responsável pela autoria do documento citado não deve ser escrito em itálico (exemplo: Segundo Sun Microsystems (2004), ...),
 - palavras em língua estrangeira encontradas nos dicionários nacionais: não devem ser grafadas em itálico (exemplos: software, hardware, web, Internet),

- demais palavras em língua estrangeira: devem ser escritas em itálico (exemplos: *palmtop*, *classpath*, *play*, etc.). No entanto, Teodorowitsch (2003, p. 7), sugere que alguns termos em língua inglesa devem ser substituídos por termos em português (exemplos: núcleo em vez de *kernel*, aprendizagem de máquina em vez de *machine learning*, etc.);
- g) observar as seguintes regras quanto ao uso de siglas:
- colocar as siglas entre parênteses precedidas pela forma completa do nome, quando aparecem pela primeira vez no texto (exemplos: Associação Brasileira de Normas Técnicas (ABNT), Trabalho de Conclusão de Curso (TCC)). Caso exista uma lista de siglas na parte pré-textual do volume final, pode-se usar somente a sigla, quando aparecer pela primeira vez no texto,
 - usar apenas a sigla nas demais ocorrências no texto,
 - escrever as siglas em letras maiúsculas e não usar itálico,
 - escrever o plural das siglas sem apóstrofo (exemplos: PCs, APIs, PDAs) e determinar o gênero da sigla conforme o gênero do primeiro substantivo do seu nome (exemplo: o TCC – o Trabalho de Conclusão de Curso).

6.1 FORMATAÇÃO

A formatação geral para apresentação do documento, descrita na NBR 14724 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011), é a seguinte:

- a) o texto divide-se em capítulos, seções e subseções (até cinco divisões);
- b) a apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b);
- c) a descrição das referências bibliográficas deve estar de acordo com a NBR 6023 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002a).

Observa-se ainda que todo capítulo, seção ou subseção deve ter no mínimo um texto relacionado.

A monografia deve ser digitada usando as fontes e formatação de parágrafos deste modelo, indicadas no Quadro 2.

Quadro 2– Estilos do modelo

| USO | FORMATO |
|---|--|
| título de capítulo ou seção primária (1) | TF-TÍTULO 1 (Times New Roman, 10pt, negrito, maiúsculas) |
| título de seção secundária (1.1) | TF-TÍTULO 2 (Times New Roman, 10pt, maiúsculas) |
| título de seção terciária (1.1.1) | TF-Título 3 (Times New Roman, 10pt, minúsculas, exceto a 1ª letra da 1ª palavra do título e de nomes próprios) |
| título de seção quaternária (1.1.1.1) | TF-Título 4 (mesma formatação seção ternária) |
| título de seção quinária (1.1.1.1.1) | TF-Título 5 (mesma formatação seção ternária) |
| texto | TF-TEXTO (Times New Roman, 10pt) |
| citação direta com mais de três linhas | TF-CITAÇÃO (Times New Roman, 9pt, com recuo de 4 cm) |
| itens (alíneas) | ver descrição abaixo (Times New Roman, 10pt) |
| referência bibliográfica | TF-REFERÊNCIA ITEM (Times New Roman, 10pt, alinhada à margem esquerda) |
| fonte, legenda, texto de quadro/tabela e figura | TF-FONTE (Times New Roman, 9pt, alinhada à margem esquerda do quadro/tabela) TF-LEGENDA, (Times New Roman, 10pt, centralizada) TF-TEXTO- QUADRO (Times New Roman, 10pt) TF-FIGURA (Times New Roman, 10pt, centralizada) |

Fonte: elaborado pelo autor.

O espaçamento, também definido no modelo, deve ser conforme indicado no Quadro 3.

Quadro 3 - Espaçamento

| USO | ESPAÇAMENTO |
|--|--|
| título de capítulo ou seção primária (1) | espaço simples, com 12pt antes do parágrafo |
| título de seção secundária (1.1) | |
| título de seção terciária (1.1.1) | |
| título de seção quaternária (1.1.1.1) | |
| título da seção quinária (1.1.1.1.1) | |
| texto | espaço simples, com 6 pt antes do parágrafo |
| citação direta com mais de três linhas | espaço simples com 6pt antes e depois do parágrafo |

| | |
|--------------------------------------|---|
| itens (alíneas) | espaço simples, com 6 pt antes do parágrafo |
| referência bibliográfica | espaço simples, com 6 pt antes do parágrafo |
| legenda e texto de ilustração/tabela | espaço simples, com 6 pt antes do parágrafo |
| fonte | espaço simples, com 0pt antes do parágrafo |

Fonte: elaborado pelo autor.

Na disposição gráfica de itens (alíneas) devem ser observados os seguintes quesitos:

- a) o texto que antecede os itens termina com dois pontos;
- b) cada item deve iniciar com uma letra minúscula seguida de fecha parênteses e terminar com um ponto e vírgula, sendo que o último item termina com ponto (FORMATO: TF-ALÍNEA);
- c) o texto de cada item inicia com letra minúscula, exceto nomes próprios;
- d) quando contiver subitens, os mesmos devem iniciar com hífen colocado sob a primeira letra do texto do item correspondente (FORMATO: TF-SUBALÍNEA nível 1 ou TF-SUBALÍNEA nível 2, conforme o caso). Nesse caso, cada subitem deve terminar com uma vírgula, exceto o último que termina com ponto ou com ponto e vírgula.

Segue um exemplo:

- a) cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA);
- b) cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA):
 - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1):
 - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
 - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
 - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1);
- c) cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA).

6.1.1.1 Exemplo de título de seção quaternária [FORMATO: TF-TÍTULO 4]

Formato: TF-TEXTO.

6.1.1.1.1 Exemplo de título de seção quinária [FORMATO: TF-TÍTULO 5]

Formato: TF-TEXTO.

6.1.2 Formatação de quadros, figuras e tabelas

Um quadro contém apenas informações textuais, que podem ser agrupadas em colunas. Uma figura contém, além das informações textuais, pelo menos um elemento gráfico. Uma tabela é uma apresentação tabular de informações **numéricas** relacionadas.

Os quadros, figuras e tabelas são identificados na parte superior por uma legenda (a qual deve estar centralizada) composta pela palavra designativa (Figura, Quadro ou Tabela, conforme o caso), seguida de seu número em algarismo arábico (usar numeração progressiva, uma sequência para os quadros, outra para as figuras e outra para as tabelas), de hífen e do título. As ilustrações devem:

- a) aparecer centralizadas no texto;
- b) estar delimitadas por uma moldura simples;
- c) aparecer numa única página (quando o tamanho não exceder o da página), inclusive a legenda;
- d) serem inseridas o mais próximo possível do trecho a que se referem pela primeira vez.

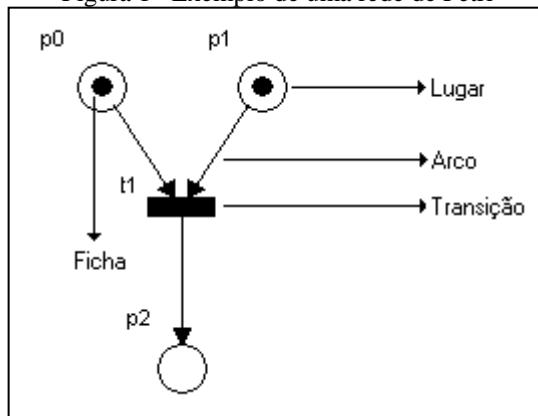
Toda ilustração deve ter fonte, alinhada à margem esquerda da ilustração. Quando foi o próprio autor que fez a ilustração, deve inserir o texto: “Fonte: elaborado pelo autor”.

Observa-se que quando um código fonte for descrito dentro de um quadro, deve-se utilizar letra do tipo **courier new 9pt.** (TF-CÓDIGO-FONTE)

Exemplos de como se deve referenciar uma figura e um quadro e como descrevê-los são mostrados a seguir.]

Um exemplo de uma rede de Petri pode ser visto na Figura 1.

Figura 1– Exemplo de uma rede de Petri



Fonte: Schubert (2003, p. 18).

Um exemplo de código fonte gerado a partir de uma especificação pode ser visto no Quadro 4.

Quadro 4 – Funções que verificam se as transições estão sensibilizadas

```
Function TEstruturaMalha.T1Sensibilizada: boolean;
begin
  result := (Fp2 and Fp4);
end;

function TEstruturaMalha.T2Sensibilizada: boolean;
begin
  result := (Fp1 and Fp3);
end;

function TEstruturaMalha.T3Sensibilizada: boolean;
begin
  result := (Fp2 and Fp4);
end;
```

Fonte: Schubert (2003, p. 63).

Não devem ser usadas bordas (linhas) verticais para fechar as tabelas. Exemplo de como se deve referenciar uma tabela e como descrevê-la é mostrado a seguir.

A quantidade de trabalhos finais realizados no Curso de Ciência da Computação (de 2010 até 2014) é apresentada na Tabela 1.

Tabela 1 – Trabalhos finais realizados no Curso de Ciência da Computação

| Ano | Estágios | TCC's | Totais |
|--------|----------|----------|------------|
| 2010/1 | 0 | 16 | 16 |
| 2010/2 | 0 | 21 | 21 |
| 2011/1 | 0 | 25 | 25 |
| 2011/2 | 0 | 23 | 23 |
| 2012/1 | 0 | 23 | 23 |
| 2012/2 | 0 | 22 | 22 |
| 2013/1 | 0 | 25 | 25 |
| 2013/2 | 0 | 16 | 16 |
| 2014/1 | 0 | 18 | 18 |
| 2014/2 | 0 | 13 | 13 |
| | | 0 | 202 |
| | | | 202 |

Fonte: elaborado pelo autor.

6.1.3 Exemplos de citações retiradas de documentos ou de nomes constituintes de uma entidade

A apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b). O sistema a ser usado é o alfabético. Exemplos de citações são: “Numa publicação recente (SEBESTA, 2000) é exposto ...” e “Segundo Silva et al. (1987), execução controlada de programas é ...”.

Quando a citação referir-se a uma parte específica do documento consultado, especificar no texto da monografia a(s) página(s). Esta(s) deverá(ão) seguir a data, separada(s) por vírgula(s) e precedida(s) pelo designativo

que a(s) caracteriza(m). Como exemplo, mostra-se: “(SCHIMT, 1999, p. 50)” ou “... visto que Schimt (1999, p. 50) implementou ...”.

As citações diretas (transcrição textual de parte da obra do autor consultado), no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem as aspas (FORMATO: TF-CITAÇÃO), conforme o exemplo a seguir.

A Associação Brasileira de Normas Técnicas (ABNT) é o Fórum Nacional de Normalização. As Normas Brasileiras, cujo conteúdo é de responsabilidade dos Comitês Brasileiros (ABNT/CB) e dos Organismos de Normalização Setorial (ABNT/ONS), são elaboradas por Comissões de Estudo (CE), formadas por representantes dos setores envolvidos, delas fazendo parte: produtores, consumidores e neutros (universidades, laboratórios e outros). (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b, p. 1).

Quando da citação de um nome (identificador) constituinte de uma entidade em um texto, deve-se utilizar o tipo de letra *courier new*, com tamanho dez (10). Para facilitar a formatação, existe o estilo de palavra denominado TF-COURIER10. Como exemplo cita-se nome de classe, atributo ou método. A seguir é apresentado um exemplo.

As classes `TTabelaTransicao` e `TExpressaoRegular` são classes de interface, porém estão sendo consideradas como classes de domínio da aplicação.