

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(x) PROJETO	ANO/SEMESTRE: 2018-1

APLICAÇÃO PARA CONTROLE DE FLUXO E MENSAGERIA ENTRE SERVIÇOS E DISPOSITIVOS IOT

Silvio Greuel

Prof. Dalton Solano dos Reis – Orientador

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 tem 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).

(Vorapojsut 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, notas-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 MÄHR, 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.

A proposta para o desenvolvimento deste trabalho, será criar 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, a utilização de tecnologias Web para o desenvolvimento se torna atraente, na proposta, será também utilizada a tecnologia *Progressive Web App* (PWA) que é basicamente 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 OBJETIVOS

O objetivo é desenvolver uma aplicação de controle de fluxo, mensageria e gatilhos entre dispositivos IoT.

Os objetivos específicos são:

- a) desenvolver um aplicativo multiplataforma;
- b) criar uma *Application Programming Interface* (API) qual tratará da gestão de fluxos e mensageria entre dispositivos IoT;
- c) criar ao menos dois gatilhos para a execução dos fluxos;
- d) criar processadores de computação para os fluxos definidos;
- e) controlar ao menos dois dispositivos IoT.

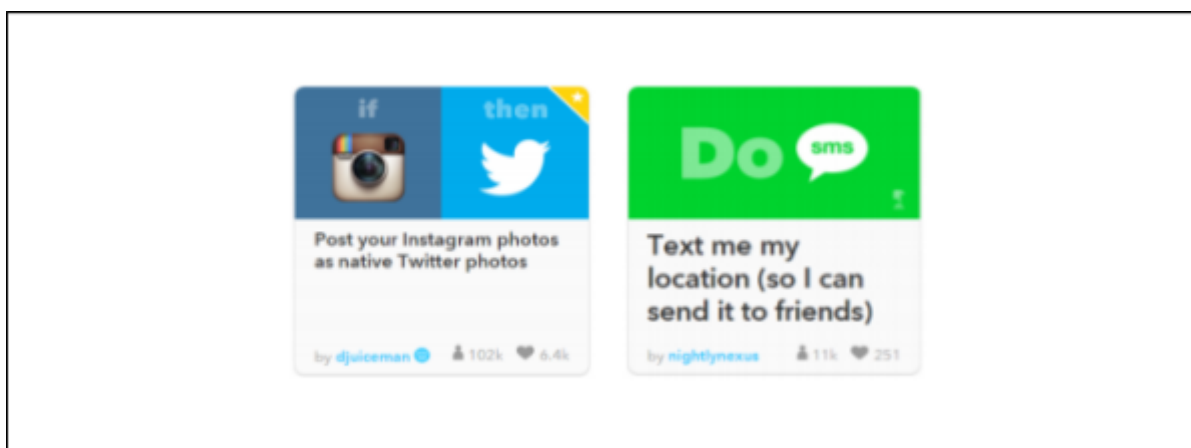
2 TRABALHOS CORRELATOS

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.

2.1 A LIGHTWEIGHT FRAMEWORK OF HOME AUTOMATION SYSTEMS BASED ON THE IFTTT MODEL

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 como demonstrado na Figura 1, que apresenta duas simples receitas de ações. O da esquerda, em azul, que, ao ser ativado, toda vez que, alguma postagem ser realizada no serviço *Instagram*, irá também realizar a mesma postagem no serviço *Twitter*. O da direita, em verde, que, toda vez que ser ativado, via *Short Message Service* (SMS), mandará a posição geográfica do dispositivo, para uma lista de contatos.

Figura 1 - Exemplo das receitas baseados nos canais disponíveis pela aplicação IFTTT



Fonte: Vorapojpisut (2015).

Conforme (Vorapojpisut 2015, p. 2, tradução nossa) “são três as características chaves da aplicação, dentre elas, o modelo de papéis, o modelo de aplicação e por fim o modelo de segurança”. O modelo de papéis definido por Vorapojpisut (2015) identifica três principais grupos de papéis. O primeiro papel é chamado de IFTTT, tendo como principal responsabilidade definir integrações chamados de canais, de maneira que possam ser compostas e apresentadas de maneira fácil para os usuários. O segundo papel é desempenhado pelo então chamado de canal, este canal disponibiliza uma coleção de serviços e operações em conformidade com as especificações fornecidas pela aplicação IFTTT. O terceiro papel quem exerce é o usuário, que por sua vez compõe fluxos utilizando os canais disponibilizados pelo IFTTT.

Vorapojpisut (2015, p. 3, tradução nossa) indica que “o modelo de aplicação mantém duas diferentes perspectivas; a chamada perspectiva do desenvolvedor (canal) e a perspectiva do usuário (receita)”. O escopo do modelo de aplicação é abstrair serviços como de meteorologia ou de tráfego, em uma coleção de gatilhos ou ações, a ideia principal, além de simplificar a utilização de tais serviços, é criar um fluxo onde os serviços possam ser interdependentes.

De acordo com (Vorapojpisut 2015, p. 4, tradução nossa) “o modelo de segurança do ambiente IFTTT tem duas preocupações principais, a confiabilidade para com um canal e a privacidade dos usuários”. O modelo de segurança da aplicação IFTTT, obriga que o usuário

seja autenticado e, para cada canal disponível, o usuário também deverá, caso necessário, se autenticar no serviço exposto pelo canal.

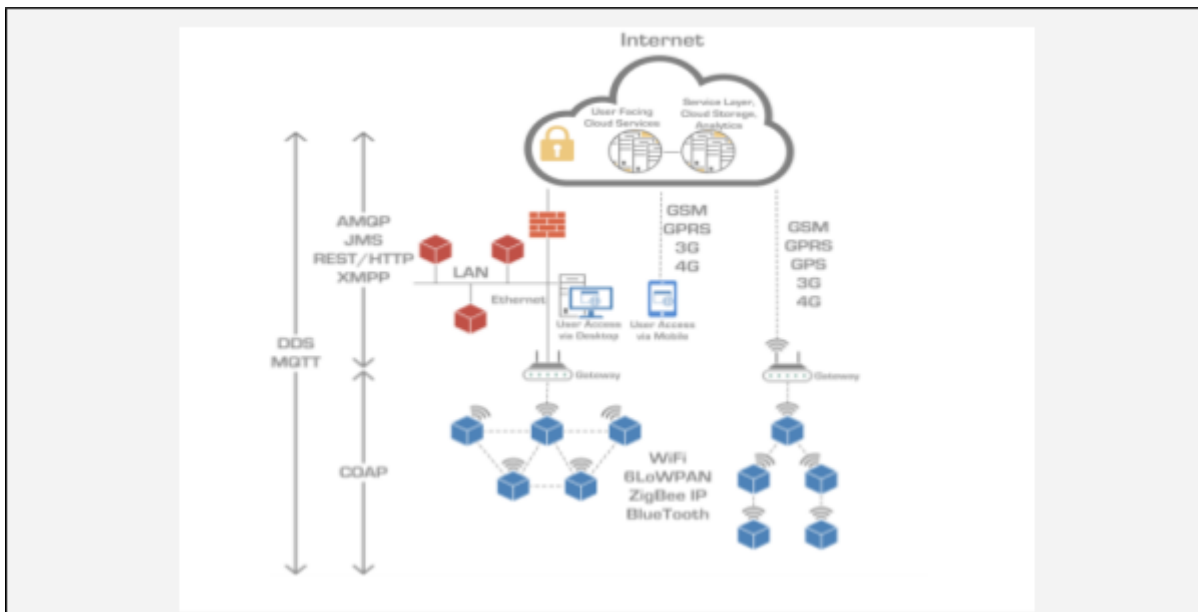
2.2 A COMPARISON BETWEEN DDS, AMQP, MQTT, JMS, REST, COAP AND XMPP

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.

É abordado algumas das problemáticas enfrentadas na conectividade e mensageria dos dispositivos, como indica a Figura 2. Na direita, as camadas de redes, seus dispositivos e protocolos de transporte, onde, todo dispositivo acessa um *gateway* que realiza a tradução dos variados protocolos de transporte, como *BlueTooth* e *ZigBee* para *TCP/IP* por exemplo. Na esquerda, a área de atuação dos protocolos de mensageria, os que utilizam *TCP/IP*: AMQP, JMS, REST/HTTP e XMPP; os que utilizam *BlueTooth* e *ZigBee*: COAP; os que utilizam ambos: DDS, MQTT.

Figura 2 - Problemática da conectividade



Fonte: ADLINK (2017).

2.3 COMPARING PROGRESSIVE WEB APPLICATIONS WITH NATIVE ANDROID

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).

Contudo, a atração pela tecnologia PWA não se justifica somente pela performance, mas também pela portabilidade, como explicam (Fransson e Driaguine 2017, p. 4, tradução nossa) “Aplicações Web, geralmente são descritas como multiplataforma. São acessíveis perante uma lista variante de navegadores, rodando em diferentes tipos de sistemas operacionais”.

Desenvolvedores têm usado tecnologias Web para desenvolver aplicações multiplataforma para dispositivos móveis utilizando ferramentas como *Cordova* e *PhoneGap* como explica (PIRAN, 2014) "Entre os frameworks mais utilizados para o desenvolvimento de aplicações híbridas encontra-se o PhoneGap/Cordova ele serve como uma ponte entre as aplicações e os recursos de hardware das plataformas". Estas aplicações são instaladas na perspectiva do sistema operacional, sendo executadas em um contexto nativo, onde, todas funções que o sistema operacional dispõe podem ser acessados. (Fransson, Driaguine (2017,

p. 4, tradução nossa) comentam “Aplicativos Web, que não são executados pelo contexto nativo, são limitados pelo contexto dos navegadores. As PWAs podem solucionar este problema”.

Foi apresentado como conclusão deste trabalho, que não houve diferença significativa de performance ao se tratar da renderização e consumo de bateria, entre PWA e aplicações nativas. O único ponto limitante de uma PWA, é não ter acesso a algumas funções somente disponíveis para aplicações que são executadas no contexto do sistema operacional.

3 PROPOSTA DA APLICAÇÃO

Nesta seção, serão expostos algumas justificativas que validam o desenvolvimento da aplicação proposta, além de também apresentar os requisitos funcionais, os requisitos não funcionais e a metodologia utilizada para implementação.

3.1 JUSTIFICATIVA

No Quadro 1, são destacadas algumas características tecnológicas, tais características são desejadas para a produção da aplicação e, disponíveis atualmente em licenças permissivas e abertas para o uso. Há também uma relação das característica desejadas para com os temas abordados nos trabalhos correlatos.

Quadro 1 - Características desejadas abordadas nos trabalhos correlatos

Características	Correlatos	Vorapojsut (2015)	ADLINK (2017)	Fransson, Driaguine (2017)
Aborda aplicação WEB		SIM	NÃO	SIM
Aborda interface móvel		SIM	NÃO	SIM
Aborda multiplataforma		NÃO	NÃO	SIM
Aborda definição do controle de fluxos		SIM	NÃO	NÃO
Aborda definição do controle de dispositivos		SIM	NÃO	NÃO
Aborda utilização do Protocolo REST		NÃO	SIM	NÃO
Aborda utilização do Protocolo AMQP		NÃO	SIM	NÃO
Aborda utilização do Protocolo MQTT		NÃO	SIM	NÃO

Fonte: elaborado pelo autor.

Nota-se que todas as características desejadas são abordadas ou, ao menos citadas nos trabalhos correlatos. Porém, nenhum trabalho correlato aborda todas as características, nem relata a produção de um software que integre as tecnologias IoT e PWA, em função da criação de um modelo de gatilhos e fluxos definidos pela aplicação IFTTT, onde, ações do usuário, via algum aplicativo, ou alterações no ambiente como aumento da temperatura, disparam eventos (gatilhos), qual por sua vez, iniciam um encadeamento de ações

automatizadas, como ligar algum dispositivo IoT (fluxos)

Seguindo a premissa do desenvolvimento de software livre, levanto a relevância deste trabalho, como porta de entrada para futuros desenvolvimentos de aplicações e dispositivos *Internet of Things* (IoT), de modo que não haja exclusividade de uso legal.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O trabalho deverá apresentar os seguintes requisitos:

1) Requisitos funcionais (RF):

- a) permitir o cadastro dos usuários utilizando o método de autenticação OAUTH;
- b) permitir a gestão de dispositivos IoT;
- c) permitir a gestão de gatilhos;
- d) permitir a gestão de fluxos;
- e) permitir a utilização dos protocolos MQTT, HTTP e AMQP;

2) Requisitos não funcionais (RNF):

- a) ser desenvolvido como uma PWA;
- b) manter performance similar a uma aplicação nativa;
- c) utilizar criptografia para comunicação dos dispositivos para com os serviços;
- d) manter um padrão adequado na interface de usuário para uso em dispositivos móveis;
- e) ser desenvolvido utilizando o *framework* VUE;
- f) ser desenvolvido utilizando o *broker* de mensageria RabbitMQ.

3.3 METODOLOGIA

O trabalho se fundamentará aplicando o conceito dos doze fatores de uma aplicação de software, será desenvolvido orientado a testes, utilizando-se dos fundamentos de programação reativa e funcional para a construção da interface gráfica e, programação orientada a objetos para a construção da API.

Será idealizado observando as seguintes etapas:

- a) levantamento bibliográfico: levantar documentações sobre a tecnologia PWA, a *framework* VUE e o *broker* de mensageria RabbitMQ e trabalhos correlatos;
- b) levantamento tecnológico: validar o uso das tecnologias levantadas na etapa (a) com o desenvolvimento de provas de conceito;

- c) levantamento dos requisitos: caso necessário, incrementar os requisitos funcionais e não funcionais propostos na seção 3.2 de modo a atender os objetivos do software proposto em sua completude;
- d) diagramação: diagramar o software seguindo o padrão *Unified Modeling Language* UML utilizando a ferramenta Visio;
- e) implementação: implementar a solução qual inclui três aplicações, uma de controle *frontend* que fará do uso da tecnologia PWA, uma de controle *backend* utilizando-se do protocolo REST, uma aplicação de processamento de fluxos e gatilhos qual será a porta de entrada comum para a mensageria entre os dispositivos IoT;
- f) documentação: documentar de maneira simples e que seja de fácil reprodução;
- g) testes: realizar os testes atendendo os objetivos listados na seção 1.1.

As etapas serão realizadas nos períodos relacionados no Quadro 1.

Quadro 1 - Cronograma

etapas / quinzenas	2018									
	ago.		set.		out.		nov.		dez.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
levantamento tecnológico										
levantamento dos requisitos										
diagramação										
implementação										
documentação										
testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve brevemente os fundamentos do estudo a ser realizado. Na seção 4.1 são apresentados os benefícios do software livre. Na seção 4.2 descreve dispositivos *Internet of Things* (IoT) e sua utilidade. Na seção 4.3 são colocados em evidência alguns protocolos de mensageria para dispositivos *Internet of Things* (IoT). Por fim na seção 4.4 será abordado o uso de uma PWA para o desenvolvimento de uma aplicação multiplataforma.

4.1 O SOFTWARE LIVRE

O termo software livre, refere-se principalmente ao modelo de licença utilizada pelo software, onde, para ser considerado livre, deve dispor de permissividades flexíveis a utilização do mesmo. Entende-se por software livre (free software) todo software cujo esquema de licenciamento permite ao seu usuário as condições de uso, reprodução, alteração e distribuição do seu código fonte, seja na sua forma original ou modificada, seja com ou sem custos (CARMONA, 2008, p. 11). Software livre, com permissividades de alteração e utilização, detém uma grande responsabilidade sobre as inovações atualmente. O escopo do software livre usualmente é associado ao intenso desenvolvimento tecnológico que as últimas décadas trouxeram, especialmente na área denominada tecnologia da informação (CARMONA, 2008, p. 7)

4.2 A IOT E SUA UTILIDADE

A IoT, tem como principal característica, ser um objeto qual se conecta e se comunica pela internet, como define Souza (2016, p. 13) “A principal característica deste conceito é a presença de um conjunto de objetos, sensores e dispositivos eletrônicos [...] que são capazes de interagir entre si, a partir de uma rede [...]”.

4.3 OS PROTOCOLOS AMQP E MQTT

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 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.”

4.4 UTILIZANDO PWA COMO APLICAÇÃO MULTIPLATAFORMA

Para melhor entender como as PWAs funcionam, se é necessário conhecer algumas das APIs disponíveis atualmente em aplicações WEB, Service Workers, Fetch, Cache e App Manifest. Toda aplicação WEB é executada em um único processo, não existe processamento paralelo nem processamento concorrente, porém, existe uma API disponibilizada pelo browser chamada de Service Worker, qual deve ser tratado como outro processo, é executado fora da aplicação principal, somente é acessível via uma API de comunicação restrita, possui acesso limitado a APIs de armazenamento e comunicação externa (MOZILLA 2018). A API Fetch, é uma api de comunicação e transferência de dados externa, similar a API XMLHttpRequest, mas, seu conteúdo pode ser alterado por um Service Worker (MOZILLA 2018). O Cache, é uma das APIs de armazenamento que são acessíveis por um Service Worker, em junção a API Fetch, podemos realizar o armazenamento de alguns dados, para, quando ocorrer uma consulta idêntica, não mais haver a necessidade de realizar uma chamada de comunicação externa (MOZILLA 2018). 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)

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.

CARMONA, André Inácio Straginski. *O software livre no limite da propriedade intelectual: uma breve apresentação*. 2008. 43f. Monografia - Universidade Federal de Santa Catarina, Florianópolis, 2008.

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 Forskningsgängen, 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.

GOOGLE, **Introduction to Progressive Web App Architectures**, 2018, <https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures>. Acesso em 01 de abr. 2018.

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.

PIRAN Férlon M, *Estudo da tecnologia phonegap/cordova e a aplicação em um estudo de caso*. 2014. 25f. Artigo - Instituto Federal de Educação,

Ciência e Tecnologia Sul - Rio - Grandense, Passo Fundo, 2014.

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.

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

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR TCC I

Acadêmico(a):

Avaliador(a):

ASPECTOS AVALIADOS ¹		aten de	aten de parc ialm ente	não aten de
A S P E C T O S T É C N I C O S	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
A S P E C T O S M E T O D O L Ó G I C O S	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			
	9. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?			
	10. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?			
	11. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?			
	As citações obedecem às normas da ABNT?			
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?			

PARECER – PROFESSOR DE TCC I OU COORDENADOR DE TCC (PREENCHER APENAS NO PROJETO):

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 4 (quatro) itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos 4 (quatro) itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER:

() APROVADO

() REPROVADO

Assinatura:

Data:

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a):

Avaliador(a):

ASPECTOS AVALIADOS ¹		aten de	atend e parci alme nte	não aten de
ASPE CTOS TÉCN ICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	12. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	13. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	14. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	15. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	16. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	17. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
ASPE CTOS MET ODO LÓGI COS	18. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			

PARECER – PROFESSOR AVALIADOR: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 5 (cinco) tiverem resposta ATENDE PARCIALMENTE.

PARECER:

() APROVADO

() REPROVADO

Assinatura:

Data: