

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
( X ) PRÉ-PROJETO	( ) PROJETO	ANO/SEMESTRE: 2021/2

## COMUNICAÇÃO DE APLICATIVOS MÓVEIS: UM MODELO DE ARQUITETURA ESCALÁVEL E RESILIENTE

Ariel Adonai Souza

Prof. Francisco Adell Péricas – Orientador

### 1 INTRODUÇÃO

Estamos entrando em um mundo cada vez mais rodeado por aplicativos móveis. Estes aplicativos estão se tornando a principal ferramenta de comunicação das pessoas com a Internet, conforme citado por Oehlman e Blanc (2012). Diante deste cenário, é natural que surja uma demanda crescente de ferramentas e tecnologias que embarquem os softwares de empresas nos dispositivos móveis. Oehlman e Blanc (2012) também nos contam que em empresas com aplicações Web, os desenvolvedores reconstróem grandes porções das aplicações dentro dos aplicativos móveis para cada um dos diferentes dispositivos. Isso não parece ser uma abordagem muito viável pois demanda de muitas pessoas para manter essas diferentes aplicações.

Uma solução mais viável para dispositivos móveis é desenvolver uma aplicação Web para dispositivos móveis. Segundo os autores, aplicações Web são uma forma de escrever aplicações que, quando feito da forma correta, permite adaptar as aplicações para dispositivos móveis sem a necessidade de reescrever muito código. “Um aplicativo Web para dispositivos móveis é um aplicativo construído com as tecnologias Web centradas no cliente em HTML, CSS e JavaScript, e é especificamente projetado para os dispositivos móveis” (OEHLMAN; BLANC, 2012, p. 9).

Segundo Tanenbaum, Steen (2008), em meados de 1980 ocorreram dois avanços tecnológicos que revolucionariam a computação até os dias atuais. O primeiro foram os microcontroladores de maior capacidade, que com o passar do tempo chagavam à capacidade de processamento de um *mainframe* por um preço muito menor. O segundo avanço foi o desenvolvimento das redes de computadores e o surgimento das redes locais, as LANs. As LANs permitiram que centenas de computadores próximos pudessem trocar informações na velocidade de alguns microssegundos. O resultado do surgimento destas duas tecnologias foi o surgimento de um modelo de computação de altíssima capacidade de processamento conectado em uma rede de alta velocidade, os sistemas distribuídos.

Para Tanenbaum, Steen (2008), a definição de um sistema distribuído é “um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente.” (TANENBAUM; STEEN, 2008, p. 1). Eles ressaltam aspectos importantes: um sistema distribuído consiste em computadores autônomos e usuários (programas, aplicações ou pessoas) que devem achar que tratam com um sistema único.

Existem algumas dificuldades na comunicação entre dispositivos móveis e sistemas distribuídos. Dentre as dificuldades, pode-se citar a escalabilidade e a garantia de entrega de mensagens assíncronas. Aplicações móveis geralmente demandam uma grande capacidade dos servidores em suportar a alta volumetria das requisições. Um cenário provável é uma grande quantidade de dispositivos acessando um determinado serviço, chegando ao ponto de o serviço não conseguir comportar a quantidade de dados e cair. Um outro cenário muito comum é um dispositivo móvel parar de se comunicar (ficou sem rede ou fechou a aplicação) e um serviço precisar entregar uma mensagem para o dispositivo. Neste caso, o serviço precisaria armazenar a mensagem e aguardar até que o dispositivo se conecte novamente para entregar a mensagem.

Segundo Tanenbaum, Steen (2008), em um ambiente de computação distribuída, por conveniência, é utilizado um modelo de comunicação assíncrona e existem diversos sistemas por trás de uma única aplicação. Diante deste cenário, é possível induzir que quando um serviço deseja se comunicar com o dispositivo móvel, ele faça isso através de um serviço que faça o intermédio da conversa entre o serviço e o dispositivo móvel.

Diante do cenário apresentado, como seria possível implementar sistemas que consigam lidar com alta volumetria, que garantam a entrega das mensagens e se comuniquem de forma simples e organizada com os dispositivos?

## 1.1 OBJETIVOS

O objetivo deste trabalho é apresentar um modelo de arquitetura distribuída que seja escalável e resiliente, ou seja, deve ser capaz de comportar muitos dispositivos conectados, muitas mensagens sendo trafegadas e que garanta a entrega de mensagens.

Os objetivos específicos deste trabalho são desenvolver uma arquitetura distribuída que:

- deve possibilitar ser implementada em praticamente qualquer aplicação existente, como uma solução para comunicação entre dispositivos móveis e servidores;
- deve suportar um fluxo constante ou variável de mensagens entre as aplicações;
- deve garantir a entrega das mensagens mesmo se o cliente estiver incomunicável no momento do envio;
- deve atender a um contexto de computação distribuída, ou seja, suportar comunicação entre diversas aplicações, não somente uma ou duas.

## 2 TRABALHOS CORRELATOS

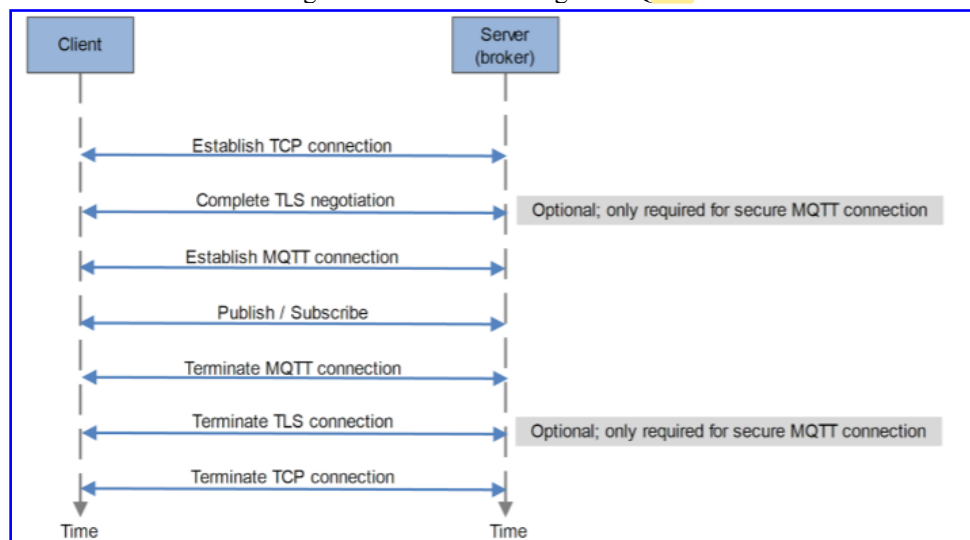
Não foram encontrados trabalhos correlatos que fossem diretamente relacionados ao objetivo deste estudo. Desta forma, foram utilizados trabalhos que possuam um ou mais aspectos do objetivo deste estudo. O primeiro trabalho foi desenvolvido por Koneski (2018) e descreve a implementação da comunicação entre dispositivos IoT utilizando os protocolos *Transport Layer Security* (TLS) e *Message Queuing Telemetry Transport* (MQTT). O segundo trabalho foi desenvolvido por Kodali, Gorantla (2017) e descreve a utilização de dispositivos IoT e do protocolo MQTT para monitoramento do clima. O terceiro trabalho é a tese de mestrado desenvolvida por Silva (2017) que apresenta a plataforma EcoCIT, uma plataforma de *middleware* escalável que provê suporte para a integração de dispositivos de IoT à internet.

### 2.1 AMBIENTE DE COMUNICAÇÃO SEGURA DE INTERNET DAS COISAS COM A UTILIZAÇÃO DO MQTT E TLS

Koneski (2018) desenvolveu um trabalho voltado para o uso de dispositivos *Internet of Things* (IoT) na área de plantação agrícola e disserta a respeito da importância da utilização e a melhoria na qualidade da lavoura que o uso de tais tecnologias para monitoramento de sensores pode proporcionar. Dentre vários problemas que o ambiente agrícola possui, Koneski (2018) nos conta sobre o principal problema segundo especialistas: a importância da monitoração das áreas que são pulverizadas e que mesmo assim surgem problemas com doenças e inóculos de ervas daninhas.

O trabalho justifica que falta segurança no protocolo *Message Queuing Telemetry Transport* (MQTT), o que deixa a comunicação vulnerável a outros usuários sem autorização obterem o acesso e alterarem valores de medição coletados pelos sensores por meio de interceptação dos dados. Sendo assim, o objetivo do trabalho foi aumentar a segurança na comunicação dos dispositivos IoT. Koneski (2018) propôs implementar uma camada de segurança na comunicação entre dispositivos IoT e servidores utilizando *Transport Layer Security* (TLS). Conforme a Figura 1 nos mostra, a comunicação MQTT é estabelecida somente depois do TLS.

Figura 1: Troca de mensagens MQTT.



Fonte: Koneski (2018).

Koneski (2018) explica que os resultados foram satisfatórios, atenderam aos testes garantindo segurança, integridade e confidencialidade aos dados trafegados. Os testes consistiam em mostrar que o sistema como um todo era capaz de realizar a o envio, recebimento e armazenamento dos dados de forma correta. Foram utilizadas algumas ferramentas para comprovar o funcionamento das aplicações, uma delas foi o *Wireshark* para monitorar o tráfego na rede em tempo real.

## 2.2 WEATHER TRACKING SYSTEM USING MQTT AND SQLITE

Kodali, Gorantla (2017) escreveram o artigo “Weather Tracking System using MQTT and SQLite” que também é voltado para dispositivos IoT, porém, utilizaram a tecnologia para montar um *framework* que possibilita a implementação de sistemas de monitoramento de sensores. No artigo utilizaram o *framework* para implementar um sistema de monitoramento de clima utilizando sensores de temperatura e umidade.

O *framework* é dividido em duas partes: uma parte é o servidor que fica hospedado em uma placa Raspberry Pi; a outra parte é o cliente, um dispositivo IoT que utiliza o microcontrolador ESP8266 e faz a leitura dos sensores. Todos os dados são formados em formato JSON para a comunicação entre os dispositivos e o servidor, bem como o armazenamento também é em JSON.

A aplicação servidora foi escrita em Python, armazena os dados em um banco de dados SQLite e se comunica com os dispositivos IoT através do protocolo MQTT, tudo hospedado em uma placa Raspberry Pi. O servidor possui uma validação na entrada dos dados enviados pelo cliente e decide se irá ou não armazenar o dado recebido. O servidor também possui um servidor HTTP e é possível visualizar os dados através de um navegador.

Os dispositivos IoT escolhidos são placas com o microcontrolador ESP8266, que tem a responsabilidade de ler os dados dos sensores e enviar para o servidor através do protocolo MQTT.

Kodali, Gorantla (2017) nos mostram que o resultado é modelo de baixo custo, com baixíssimo consumo de energia e alta eficiência para um sistema de monitoramento de clima.

## 2.3 ECOCIT: UMA PLATAFORMA ESCALÁVEL PARA DESENVOLVIMENTO DE APLICAÇÕES DE IOT

Silva (2017) nos conta que desenvolver aplicações IoT, à primeira vista, parece ser uma tarefa fácil. Estima-se que no ano de 2020 haveria mais de 200 bilhões de dispositivos IoT conectados na Internet. Comportar tantos dispositivos conectados ao mesmo tempo, bem como armazenar e processar todos esses dados é uma tarefa difícil. Outro fator que dificulta é que diferentes dispositivos IoT utilizam protocolos diferentes de comunicação. Diante destes cenários, muitas aplicações têm surgido para abstrair para os desenvolvedores a utilização dos diferentes protocolos e dispositivos. Estas aplicações têm como objetivo entregar uma interface padronizada para acesso aos dispositivos. O trabalho de Silva (2017) tem como objetivo apresentar “a plataforma EcoCIT, uma plataforma de *middleware* escalável que provê suporte para a integração de dispositivos IoT com requisitos de escalabilidade através do uso de serviços computacionais providos sob demanda por plataformas de computação em nuvem” (SILVA, 2017, p. 8).

Silva (2017) conta que o EcoCIT é uma evolução de uma outra plataforma chamada EcoDiF. “[...] é uma plataforma de *middleware* que integra dispositivos de IoT e os conecta à internet, fornecendo funcionalidades de controle, visualização e armazenamento de dados em tempo real.” (SILVA, 2017, p. 56). Apesar de diversas funcionalidades e facilidades que a EcoDiF proporciona, ela possui algumas limitações como a escalabilidade e capacidade de gerenciar grandes volumes de dados.

A EcoCIT surge com o objetivo de solucionar as limitações que a EcoDiF possui. A arquitetura é composta de diversos componentes distribuídos para permitir que a EcoCIT possa fazer uso da elasticidade que os ambientes de computação em nuvem podem prover. A arquitetura permite que a aplicação seja executada em máquinas virtuais que trabalham maneira agrupada e possui um balanceador de carga para distribuir a demanda entre as máquinas virtuais. A aplicação também faz uso de outros recursos escaláveis e distribuídos como os bancos de dados e outras estruturas de armazenamento.

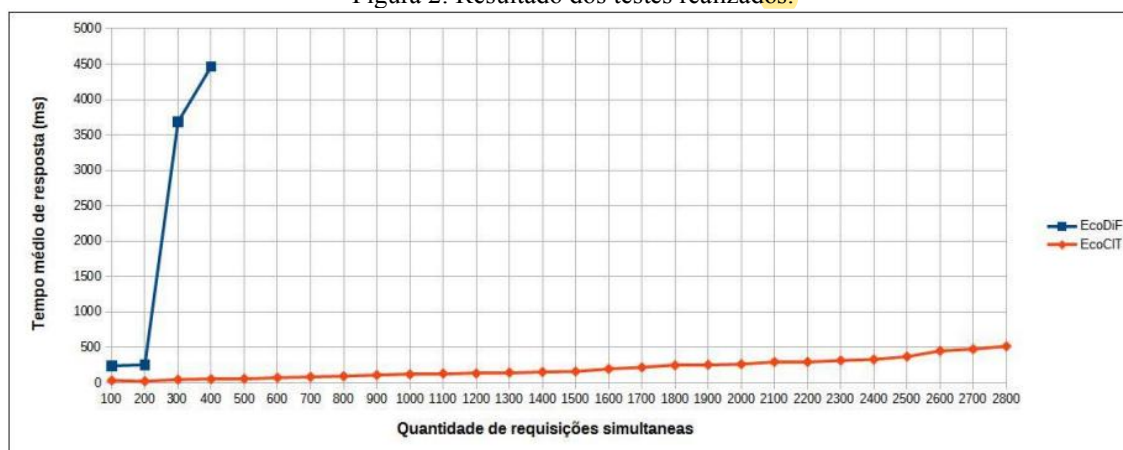
Por se tratar de uma evolução de outra ferramenta, o trabalho descreve diversas outras alterações realizadas em muitos módulos da plataforma para que fosse possível tornar a EcoDiF em uma plataforma escalável.

Silva (2017) realizou 3 experimentos em seu trabalho. O primeiro teve como objetivo avaliar a substituição do sistema gerenciador de banco de dados utilizado no gerenciamento de *feeds*. Os resultados mostraram que a substituição do banco de dados contribuiu para a melhor performance do processo. O autor

aponta que o experimento apresentou uma performance ligeiramente menor para a operação de consulta de *feeds*, porém, o novo banco de dados era muito mais escalável, sendo capaz de suportar cargas de trabalho superiores.

O segundo experimento teve como objetivo comprar o tempo de resposta entre a EcoDiF e a EcoCIT em cenários similares, com o mesmo hardware. O experimento foi realizado acrescentando a quantidade de requisições simultâneas em 100 por teste. A EcoDiF foi capaz de aguentar no máximo 400 requisições simultâneas, atingindo um tempo médio de resposta de 4466ms. Já a EcoCIT, foi capaz de aguentar 2800 requisições simultâneas e atingiu um tempo médio de resposta de 517ms. A Figura 2 é um gráfico que apresenta o resultado dos testes realizados.

Figura 2: Resultado dos testes realizados.



Fonte: Silva (2017)

O terceiro experimento teve como objetivo avaliar a elasticidade da plataforma. Ou seja, estressar a plataforma ao ponto de disparar os mecanismos de elasticidade, fazendo com que mais instâncias fossem criadas quando a demanda aumentasse e destruindo as instâncias quando a demanda diminuísse. “Esse experimento mostrou que [...] é capaz de monitorar a carga de trabalho nas VMs e tornar as medidas necessárias para evitar que as VMs se tornem sobrecarregadas ou ociosas, ajustando a demanda por recursos computacionais ao uso” (SILVA, 2017, p. 118).

### 3 PROPOSTA DA ARQUITETURA

Este capítulo tem como objetivo justificar este trabalho, apresentar os seus motivadores e a metodologia de desenvolvimento que será utilizada.

#### 3.1 JUSTIFICATIVA

Com o aumento do uso dos dispositivos móveis e a quantidade cada vez maior de aplicativos, é natural que alguns problemas comecem a aparecer. Um destes problemas é a comunicação entre os dispositivos e os servidores, principalmente quando se fala de um contexto de computação distribuída.

O Quadro 1 apresenta algumas características que mostram que este trabalho pode contribuir para sanar esta lacuna na comunicação entre os dispositivos. Durante a pesquisa e nos trabalhos correlatos selecionados, nenhum deles apresenta uma proposta de arquitetura de comunicação de aplicativos móveis com servidores que tenha o objetivo de ser escalável e resiliente. Um cenário que necessite de uma solução assim pode ser encontrado facilmente num contexto de dispositivos móveis e servidores utilizando a arquitetura de computação distribuída. Este trabalho pode contribuir com um modelo que simplifique a comunicação, melhore a estabilidade e tirar a responsabilidade dos serviços de se preocuparem com a entrega de mensagens para aplicações móveis.

Quadro 1 - Comparativo dos trabalhos correlatos

Trabalhos Correlatos	Koneski (2018)	Kodali, Gorantla (2017)	Silva (2017)
Características			
Escalável			X
Resiliente	X	X	
Utilização de alguma ferramenta de mensageria	X	X	

Fonte: elaborado pelo autor.

Acima foram apresentadas algumas características que este trabalho se propõe a atender em comparação aos trabalhos correlatos. Todos os trabalhos deixam margem para que seja buscada uma solução que possa atender a todas as características citadas. O trabalho desenvolvido por Koneski (2018) apresenta uma forma de trocar a comunicação MQTT mais segura através da utilização de uma camada TLS na comunicação. Entretanto, o trabalho não é apresentado como uma solução escalável e para dispositivos móveis. Já o trabalho desenvolvido por Kodali, Gorantla (2017) apresenta um *framework* desenvolvido no artigo com o objetivo de armazenar e apresentar dados extraídos de sensores utilizando algumas das tecnologias IoT. Este trabalho também não nos apresenta uma solução escalável para comunicação, nem para dispositivos móveis. O trabalho desenvolvido por Silva (2017) apresenta uma evolução de uma plataforma para abstrair o gerenciamento de dispositivos IoT que atendam o requisito de ser escalável. Porém, não apresenta uma solução resiliente, de comunicação assíncrona e para dispositivos móveis.

### 3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A arquitetura descrita neste trabalho deverá:

- ser possível comportar uma volumetria cada vez maior (escalável) (RF);
- armazenar as mensagens até conseguir entregá-las ao destino (RF);
- garantir a entrega das mensagens pendentes (ainda não enviadas para o destinatário) assim que os dispositivos conectarem novamente (RF);
- ser possível que os dispositivos móveis enviem mensagens quando o serviço estiver fora do ar (RF);
- ser possível que os serviços enviem mensagens quando o dispositivo móvel estiver fora do ar (RF);
- fazer com que todas as mensagens sejam assíncronas, ou seja, a aplicação que enviar não deve esperar um retorno de confirmação da outra aplicação (RF);
- utilizar um banco de dados relacional para armazenar as mensagens temporariamente (Requisito Não Funcional - RNF);
- utilizar uma aplicação de mensageria para troca de mensagens internas utilizando o protocolo AMQT (RNF);
- possibilitar a comunicação dos dispositivos móveis através do protocolo WebSocket (RNF);
- possibilitar a comunicação dos serviços através do protocolo AMQT (RNF);

### 3.3 METODOLOGIA

Tendo base no problema, objetivos e requisitos apresentados, será adotada uma estratégia para montar uma arquitetura que possa resolver este problema.

O trabalho será desenvolvido observando as seguintes etapas:

- Estudo das tecnologias e ferramentas disponíveis no mercado;
- Desenvolver um diagrama com um modelo de arquitetura viável;
- Desenvolver uma aplicação para dispositivos móveis e dois serviços utilizando a arquitetura proposta;
- Testar as aplicações em cenários que contemplem cada um dos requisitos;
- Avaliar os resultados;
- Escrever os resultados do trabalho.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2022									
	fev.		mar.		abr.		mai.		jun.	
	1	2	1	2	1	2	1	2	1	2
Estudo das tecnologias e ferramentas disponíveis no mercado	X	X								
Desenvolver um diagrama com um modelo de arquitetura viável		X	X							
Desenvolver uma aplicação para dispositivos móveis e dois serviços utilizando a arquitetura proposta				X	X	X				
Testar as aplicações em cenários que contemplem cada um dos requisitos							X	X		
Avaliar os resultados									X	
Escrever os resultados do trabalho										X

Fonte: elaborado pelo autor.

## 4 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve brevemente os assuntos que fundamentarão o estudo a ser realizado para montar uma arquitetura que possa resolver o problema exposto anteriormente: apresentar uma solução de arquitetura para comunicação assíncrona de dispositivos móveis escalável e resiliente em um ambiente de computação distribuída.

Um sistema distribuído é definido por: “... um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente” (TANENBAUM; STEEN, 2008, p. 1). Os autores também contam sobre a estrutura e funcionamento do modelo de comunicação por mensagens.

O protocolo de comunicação Web que permite uma comunicação constante e bidirecional é definida pela WebSocket API, conforme é descrito por Lombardi (2015). Este protocolo será utilizado para comunicação dos dispositivos móveis e o servidor.

Aplicações móveis são aplicações escritas para dispositivos móveis e têm se tornado a principal ferramenta de comunicação das pessoas com a Internet. Desenvolver aplicações Web para dispositivos móveis pode ser uma forma mais viável para atender diferentes dispositivos sem a necessidade de reescrever muitas partes de código, conforme a explicação dada por Oehlman e Blanc (2012).

## REFERÊNCIAS

KODALI, Ravi Kishore; GORANTLA, Venkata Sundeep Kumar. **Weather tracking system using MQTT and SQLite**. In: 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). IEEE, 2017. p. 205-208.

KONESKI, Eduardo de Meireles. **Ambiente de comunicação segura de Internet das Coisas com a utilização do MQTT e TLS**. 2018. Trabalho de conclusão de curso (Bacharel em Sistemas de Informação) - Universidade Federal de Santa Catarina, [S. l.], 2018. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/192153/TCC%20-%20Eduardo.pdf?sequence=1>. Acesso em: 18 set. 2021.

LOMBARDI, Andrew. **WebSocket: LIGHTWEIGHT CLIENT-SERVER COMMUNICATIONS**. 1. ed. Sebastopol, CA: O'Reilly Media, Inc., 2015.

OEHLMAN, Damon; BLANC, Sébastien. **Aplicativos Web Pro Android: Desenvolvimento Pro Android Usando HTML5, CSS3 & JavaScript**. 1. ed. Rio de Janeiro: Ciência Moderna, 2012.

SILVA, Jorge Pereira da. **EcoCIT: uma plataforma escalável para desenvolvimento de aplicações de IoT**. 2017. 138f. Dissertação (Mestrado em Sistemas e Computação) - Centro de Ciências Exatas e da Terra, Universidade Federal do Rio Grande do Norte, Natal, 2017.

TANENBAUM, Andrew S.; STEEN, Maarten Van. **SISTEMAS DISTRIBUIDOS: princípios e paradigmas**. 2. ed. São Paulo: Pearson, 2008.