

Revisão do Projeto

Disciplina: Trabalho de Conclusão de Curso I – BCC

Caro orientando,

segue abaixo a tabela de cálculo da média das notas obtidas no Pré-Projeto e Projeto, as DUAS revisões do seu projeto contendo a avaliação do professor “avaliador” e professor “TCC1”. Lembro que os ajustes indicados nestas revisões não precisam ser feitos no projeto, mas sim quando levarem o conteúdo do projeto para o artigo (se for o caso). Este material contendo todo o histórico das revisões é encaminhado para o professor de TCC2.

Atenciosamente,

Nome	PreProjeto										Projeto													
	TCC1								Avaliador		Banca		TCC1								Avaliador			
	A	P	N	Nota	A	P	N	Nota	Ori.	Esp.	Nota	A	P	N	Nota	A	P	N	Nota	Média				
ArielAdonalSouza	13	3	0	16	9,38	8	7	0	15	8,44	9,00	7,00	8,00	14	2	0	16	9,58	14	1	0	15	9,78	9,25

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

MIDDLEWARE ESCALÁVEL E RESILIENTE PARA COMUNICAÇÃO ENTRE APLICAÇÕES WEB MÓVEIS EM AMBIENTES DE COMPUTAÇÃO DISTRIBUÍDA

Ariel Adonai Souza

Prof. Francisco Adell Péricas – Orientador

1 INTRODUÇÃO

O mundo está 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 desse cenário, é natural que surja uma demanda crescente de ferramentas e tecnologias que embarquem os seus softwares corporativos nos dispositivos móveis. Oehlman e Blanc (2012) também contam que em empresas com aplicações Web, os desenvolvedores reconstroem grandes porções dessas aplicações Web dentro dos aplicativos móveis para cada um dos diferentes dispositivos. “Para algumas companhias que constroem aplicativos móveis, esta é uma metodologia aceitável. É, contudo, uma das menos sustentáveis a longo prazo.” (OEHLMAN; BLANC, 2012, p. 9). Os autores apontam para um futuro em que a demanda por desenvolvedores seria muito grande para conseguir manter todas essas aplicações por conta da grande variedade de dispositivos móveis disponíveis hoje no mercado.

Uma solução mais viável é desenvolver uma aplicação Web para dispositivos móveis. Segundo Oehlman e Blanc (2012), 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 e Steen (2008), em meados de 1980 ocorreram dois avanços tecnológicos que revolucionariam a computação até os dias atuais. O primeiro foi a evolução de microcontroladores de maior capacidade, que com o passar do tempo chegavam 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 *Local Area Networks* (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.

A definição de 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.

Segundo Tanenbaum e 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 desse cenário, é possível imaginar que seja complexa uma aplicação Web para dispositivos móveis, que está fora desta rede, se comunicar com algum serviço interno.

Por mais que à primeira vista pareça complexo estabelecer a comunicação entre o dispositivo móvel e os serviços, existe um tipo de software que soluciona esta situação: os *middlewares*. Um *middleware* funciona “[...] de forma essencial como uma camada oculta de tradução, o middleware permite a comunicação e o gerenciamento de dados para aplicativos distribuídos.” (AZURE, 2021). Com a utilização de um *middleware*, a aplicação Web se comunica de forma transparente com os serviços.

Diante do cenário apresentado, levando em consideração o aumento do uso de dispositivos móveis, a facilidade e conveniência em implementar aplicações Web para dispositivos móveis, como seria possível implementar um *middleware* que consiga lidar com alta volumetria, que seja escalável e que garanta a entrega das mensagens?

1.1 OBJETIVOS

O objetivo deste trabalho é apresentar um *middleware* que seja escalável e resiliente para comunicação cliente/servidor entre aplicações Web para dispositivos móveis e serviços em um ambiente de computação distribuída.

Os objetivos específicos deste trabalho são desenvolver um *middleware* para uma arquitetura distribuída que:

Excluído: complexo

Comentado [GJ1]: Fora da rede local?

Comentado [GJ2]: O que é um “serviço interno”?

Comentado [GJ3]: Ainda permaneci com dúvida sobre a diferença da sua proposta com o que outras ferramentas de mercado, com o Kafka, RabbitMQ, etc, oferecem.

- a) seja escalável;
- b) seja resiliente na entrega de mensagens, ou seja, a mensagem deverá ser enviada até mesmo se o cliente estiver incomunicável no momento do envio (a entrega da mensagem ocorrerá quando o cliente se conectar novamente no *middleware*);
- c) atenda aos protocolos disponíveis em aplicações Web para comunicação;
- d) atenda a um contexto de computação distribuída, ou seja, deve abstrair a comunicação das aplicações Web com os diversos serviços que possam existir.

2 TRABALHOS CORRELATOS

Foi encontrado um trabalho que se assemelha bastante com o objetivo deste estudo, o trabalho desenvolvido por Fernandes (2013). Os demais trabalhos não possuem uma relação direta com o objetivo deste estudo, porém, possuem um ou mais aspectos do objetivo deste estudo. O primeiro trabalho foi desenvolvido por Fernandes (2013) e descreve um *framework* para aplicações Web *Real Time*. O segundo trabalho foi desenvolvido por Souza (2016) e é um comparativo de APIs de sockets. 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.

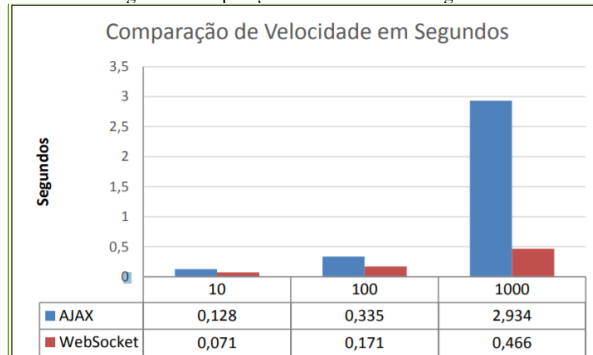
2.1 DESENVOLVIMENTO DE UMA FRAMEWORK REAL TIME WEB PARA HTML5

Fernandes (2013) desenvolveu um *framework* para aplicações Web *Real Time*. O objetivo do trabalho é desenvolver o *framework* respondendo algumas perguntas voltadas a entender se é um paradigma a ser considerado na implementação de soluções de internet, como utilizar as tecnologias do momento para melhorar a comunicação de aplicações utilizando paradigma de *publish/subscribe* e avaliar qual a tecnologia mais adequada na utilização de *real time web*.

A metodologia do trabalho constitui em fazer um levantamento da literatura sobre *real time web*, pesquisar trabalhos sobre *publish/subscribe*, estudar os pontos fortes e fracos, levantar os requisitos, definir um *framework* que implemente o paradigma *publish/subscribe*, desenvolver as aplicações para testes e avaliar os resultados.

O trabalho comparou algumas tecnologias para descobrir quais performavam melhor dentro do contexto dos testes realizados. Foram realizados comparativos de velocidade entre AJAX e WebSocket, tamanho dos dados em XML e JSON e tamanho dos *headers* utilizando AJAX e WebSocket.

Figura 1: Comparação de velocidade em Segundos

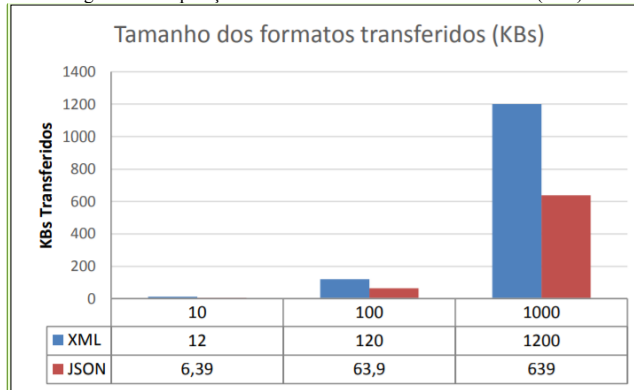


Fonte: Fernandes (2013).

Comentado [GJ4]: A figura não está sendo citada no texto

Comentado [GJ5]: O que é o eixo horizontal? O que significa "10", "100" e "1000"?

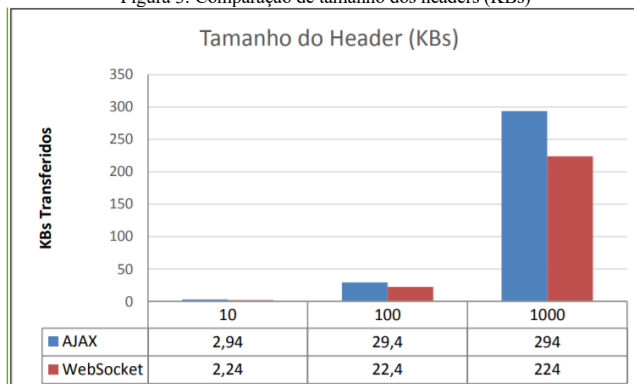
Figura 2: Comparação de tamanho dos formatos transferidos (KBs)



Fonte: Fernandes (2013).

Comentado [GJ6]: A figura não está sendo citada no texto

Figura 3: Comparação de tamanho dos headers (KBs)



Fonte: Fernandes (2013).

Comentado [GJ7]: A figura não está sendo citada no texto

O resultado do trabalho concluiu que o protocolo WebSocket obteve um melhor resultado em todos os aspectos testados. Aponta também que “[...] WebSockets e o paradigma *publisher/subscriber* são as tecnologias mais adequadas na implementação de soluções que necessitem de integrar dados em tempo real. Soluções que podem ser bastante diversificadas como já foram referidas: finanças, apostas desportivas, dados, ou mesmo o Big Data.” (FERNANDES, 2013).

2.2 ANÁLISE DE DESEMPENHO DE APIS DE SOCKETS

Souza (2016) desenvolveu um estudo comparativo de APIs *socket* visando medir o desempenho. O trabalho comparou as seguintes APIs: ZeroMQ, NanoMSG e Berkeley Sockets.

O ZeroMQ foi projetado para facilitar o desenvolvimento de aplicações distribuídas, uma API concebida para ter o mínimo de complexidade. Oferece quatro tecnologias para transporte: TCP, *MULTICAST*, *Instructions per cycle* (IPC) e *Inter-Process Communication* (INPROC).

O NanoMSG é uma concorrente do ZeroMQ. Foi concebida com a intenção de aumentar a rapidez, prover estabilidade da camada de rede e facilitar o desenvolvimento. O NanoMSG provê todos os padrões do ZeroMQ e provê também o uso do protocolo WebSocket.

Berkeley Sockets foram desenvolvidos na década de 1980 na *University of California at Berkeley*. Suporta somente TCP e UDP, e é uma API de mais baixo nível se comparada as demais citadas anteriormente.

Formatado: Fonte: Itálico

Formatado: Fonte: Itálico

Souza (2016) escolheu como métricas de avaliação a vazão e latência. Vazão é a quantidade de mensagens lidas em um determinado tempo. Latência é o tempo gasto para leitura de cada mensagem. Souza (2016) traz um comparativo entre as APIs e expõe alguns pontos relevantes: quantidade de linhas de código que são necessárias para realizar as operações de criar um socket, conectar, enviar dados, receber dados e fechar a conexão; qual a facilidade da programação; qual o desempenho com a utilização de cada uma das aplicações.

Figura 4: Comparativo das APIs

API	Quantidade de Código Gerado	Facilidade de Programação	Desempenho
Berkeley	Muito	Pouco	Excelente
ZeroMQ	Pouco	Muito	Razoável
NanoMSG	Muito	Média	Excelente

Fonte: Souza (2016).

A conclusão do trabalho, após a análise das métricas obtidas, foi que a NanoMSG obteve o melhor desempenho para os cenários dos experimentos. Souza (2016) enfatiza que esse trabalho pode ser usado como referência pela comunidade para escolha de uma das opções citadas no trabalho.

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

Silva (2017) conta que desenvolver aplicações IoT, à primeira vista, parece ser uma tarefa fácil. Ele estimou 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 com o propósito de 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) teve 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. O 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.

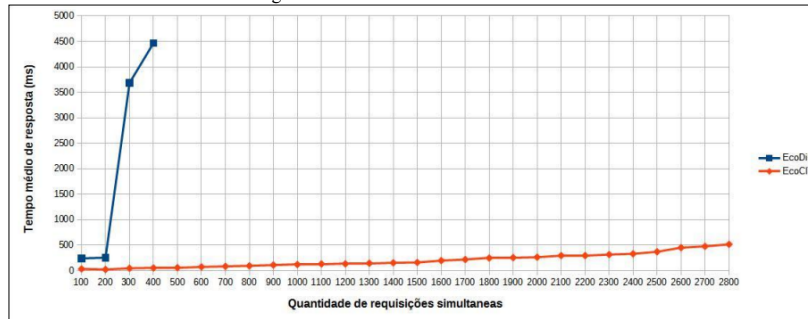
A EcoCIT surge com o objetivo de solucionar as limitações que a EcoDiF possui. A arquitetura é composta de diversos componentes distribuídos para permitir que a EcoCIT possa fazer uso da elasticidade que os ambientes de computação em nuvem podem prover. A arquitetura permite que a aplicação seja executada em máquinas virtuais que trabalham de 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 uma plataforma escalável.

Silva (2017) realizou três 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 *feeds* são basicamente os dados que os dispositivos enviam para o servidor. 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 comparar 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 suportar 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.

Excluído: aguentar

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 a solução [...] 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 DO MIDDLEWARE

Esta seção 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 entregando um *middleware* para comunicação de aplicações Web para dispositivos móveis e serviços em um ambiente distribuído. Durante a pesquisa e nos trabalhos correlatos selecionados, nenhum deles apresenta uma proposta de arquitetura de comunicação de aplicações Web para dispositivos móveis com servidores que tenham o objetivo de ser escaláveis e resilientes. 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 tire a responsabilidade dos serviços de se preocuparem com a entrega de mensagens para aplicações móveis.

Quadro 1 - Comparativo dos trabalhos correlatos

Trabalhos Correlatos	Fernandes (2013)	Souza (2016)	Silva (2017)
Características			
Escalável			X
Resiliente	X	X	
Utilização de alguma ferramenta de mensageria			X
Voltado para comunicação Web	X		

Fonte: elaborado pelo autor.

No Quadro 1 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 Fernandes (2013) apresenta um *framework* para aplicações *real time web*, que se assemelha a este trabalho por ser voltado para a Web. Entretanto, o trabalho não é apresentado como uma solução escalável e para um contexto distribuído. Já o trabalho de Souza (2016) comparou diversas APIs de sockets, porém, ele trabalhou em uma camada mais baixo nível e não com uma comunicação Web. O trabalho dele pode ser utilizado para a execução deste trabalho na implementação dos sockets de comunicação. 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

O *middleware* da arquitetura proposta 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);
- possibilitar que os dispositivos móveis enviem mensagens quando o serviço estiver fora do ar (RF);
- possibilitar 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 não 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).

Comentado [GJ8]: Como se pretende testar o framework que vai ser construído? Serão criadas aplicações de demonstração que se comunicam num ambiente distribuído?

Comentado [GJ9]: O que são “mensagens internas”?

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:

- estudar as tecnologias e ferramentas disponíveis no mercado;
- especificar um diagrama com os componentes internos do *middleware*;
- desenvolver o *middleware* proposto;
- desenvolver uma aplicação Web para dispositivos móveis e dois serviços para testar o *middleware*;
- testar a aplicação e avaliar os resultados;
- escrever os resultados do trabalho.

As etapas serão realizadas nos períodos relacionados no Quadro 6.

Quadro 2 - Cronograma

etapas / quinzenas	2022									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
estudar as tecnologias e ferramentas disponíveis no mercado	X	X								
especificar um diagrama com os componentes internos do <i>middleware</i>		X	X							
desenvolver o <i>middleware</i> proposto				X	X	X				
desenvolver uma aplicação Web para dispositivos móveis e dois serviços para testar o <i>middleware</i>							X	X		
testar a aplicação e avaliar os resultados									X	
escrever os resultados do trabalho		X	X					X	X	X

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve brevemente os assuntos que fundamentarão o estudo a ser realizado para construir o *middleware* que possa resolver o problema exposto anteriormente: apresentar um *middleware* que seja escalável e resiliente para comunicação cliente/servidor entre aplicações Web para dispositivos móveis e serviços 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 esclarecem 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

FERNANDES, Pedro Manuel da Conceição. **Desenvolvimento de uma Framework Real Time Web para HTML5**. 2013. TCC (Graduação) - Curso de Licenciatura em Sistemas e Tecnologia da Informação, Universidade Atlântica, Barbarena, 2013.

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

MICROSOFT AZURE. **O que é middleware?** Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-middleware>. Acesso em: 20 de novembro de 2021.

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.

SOUZA, Samuel Pereira de. **ANÁLISE DE DESEMPENHO DE APIS DE SOCKETS**. 2016. TTC (Graduação) - Curso de Graduação em Redes de Computadores, Universidade Federal do Ceará, Quixadá, 2016.

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

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR AVALIADOR

Avaliador(a): Gilvan Justino

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?		X	
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	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?	X		
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

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: (X) APROVADO () REPROVADO

Excluído: (

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

MIDDLEWARE ESCALÁVEL E RESILIENTE PARA COMUNICAÇÃO ENTRE APLICAÇÕES WEB MÓVEIS EM AMBIENTES DE COMPUTAÇÃO DISTRIBUÍDA

Ariel Adonai Souza

Prof. Francisco Adell Péricas – Orientador

1 INTRODUÇÃO

O mundo está 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 desse cenário, é natural que surja uma demanda crescente de ferramentas e tecnologias que embarquem os seus softwares corporativos nos dispositivos móveis. Oehlman e Blanc (2012) também contam que em empresas com aplicações Web, os desenvolvedores reconstroem grandes porções dessas aplicações Web dentro dos aplicativos móveis para cada um dos diferentes dispositivos. “Para algumas companhias que constroem aplicativos móveis, esta é uma metodologia aceitável. É, contudo, uma das menos sustentáveis a longo prazo.” (OEHLMAN; BLANC, 2012, p. 9). Os autores apontam para um futuro em que a demanda por desenvolvedores seria muito grande para conseguir manter todas essas aplicações por conta da grande variedade de dispositivos móveis disponíveis hoje no mercado.

Uma solução mais viável é desenvolver uma aplicação Web para dispositivos móveis. Segundo Oehlman e Blanc (2012), 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 e Steen (2008), em meados de 1980 ocorreram dois avanços tecnológicos que revolucionariam a computação até os dias atuais. O primeiro foi a evolução de microcontroladores de maior capacidade, que com o passar do tempo chegavam 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 *Local Area Networks* (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.

A definição de 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.

Segundo Tanenbaum e 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 desse cenário, é possível imaginar que seja complexo uma aplicação Web para dispositivos móveis, que está fora desta rede, se comunicar com algum serviço interno.

Por mais que à primeira vista pareça complexo estabelecer a comunicação entre o dispositivo móvel e os serviços, existe um tipo de software que soluciona esta situação: os *middlewares*. Um *middleware* funciona “[...] de forma essencial como uma camada oculta de tradução, o *middleware* permite a comunicação e o gerenciamento de dados para aplicativos distribuídos.” (AZURE, 2021). Com a utilização de um *middleware*, a aplicação Web se comunica de forma transparente com os serviços.

Diante do cenário apresentado, levando em consideração o aumento do uso de dispositivos móveis, a facilidade e conveniência em implementar aplicações Web para dispositivos móveis, como seria possível implementar um *middleware* que consiga lidar com alta volumetria, que seja escalável e que garanta a entrega das mensagens?

1.1 OBJETIVOS

O objetivo deste trabalho é apresentar um *middleware* que seja escalável e resiliente para comunicação cliente/servidor entre aplicações Web para dispositivos móveis e serviços em um ambiente de computação distribuída.

Os objetivos específicos deste trabalho são desenvolver um *middleware* para uma arquitetura distribuída que:

- a) seja escalável;
- b) seja resiliente na entrega de mensagens, ou seja, a mensagem deverá ser enviada até mesmo se o cliente estiver incomunicável no momento do envio (a entrega da mensagem ocorrerá quando o cliente se conectar novamente no *middleware*);
- c) atenda aos protocolos disponíveis em aplicações Web para comunicação;
- d) atenda a um contexto de computação distribuída, ou seja, deve abstrair a comunicação das aplicações Web com os diversos serviços que possam existir.

2 TRABALHOS CORRELATOS

Foi encontrado um trabalho que se assemelha bastante com o objetivo deste estudo, o trabalho desenvolvido por Fernandes (2013). Os demais trabalhos não possuem uma relação direta com o objetivo deste estudo, porém, possuem um ou mais aspectos do objetivo deste estudo. O primeiro trabalho foi desenvolvido por Fernandes (2013) e descreve um *framework* para aplicações Web *Real Time*. O segundo trabalho foi desenvolvido por Souza (2016) e é um comparativo de APIs de sockets. 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.

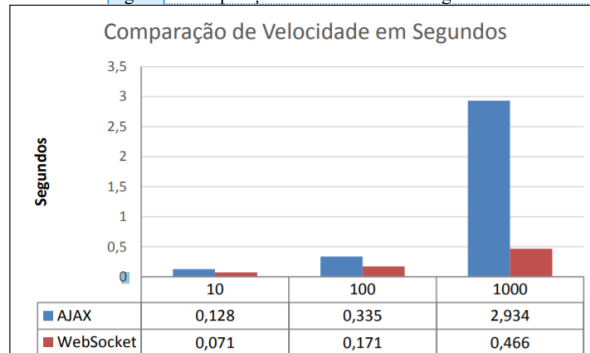
2.1 DESENVOLVIMENTO DE UMA FRAMEWORK REAL TIME WEB PARA HTML5

Fernandes (2013) desenvolveu um *framework* para aplicações Web *Real Time*. O objetivo do trabalho é desenvolver o *framework* respondendo algumas perguntas voltadas a entender se é um paradigma a ser considerado na implementação de soluções de internet, como utilizar as tecnologias do momento para melhorar a comunicação de aplicações utilizando paradigma de *publish/subscribe* e avaliar qual a tecnologia mais adequada na utilização de *real time web*.

A metodologia do trabalho constitui em fazer um levantamento da literatura sobre *real time web*, pesquisar trabalhos sobre *publish/subscribe*, estudar os pontos fortes e fracos, levantar os requisitos, definir um *framework* que implemente o paradigma *publish/subscribe*, desenvolver as aplicações para testes e avaliar os resultados.

O trabalho comparou algumas tecnologias para descobrir quais performavam melhor dentro do contexto dos testes realizados. Foram realizados comparativos de velocidade entre AJAX e WebSocket, tamanho dos dados em XML e JSON e tamanho dos *headers* utilizando AJAX e WebSocket.

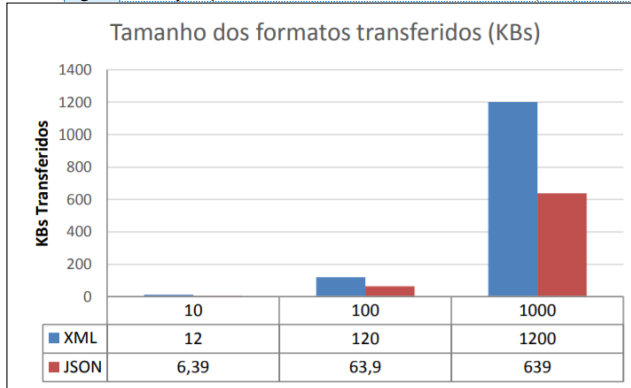
Figura 2: Comparação de velocidade em Segundos



Fonte: Fernandes (2013).

Comentado [DSdR10]: A figura deve ser citada e descrita no texto.

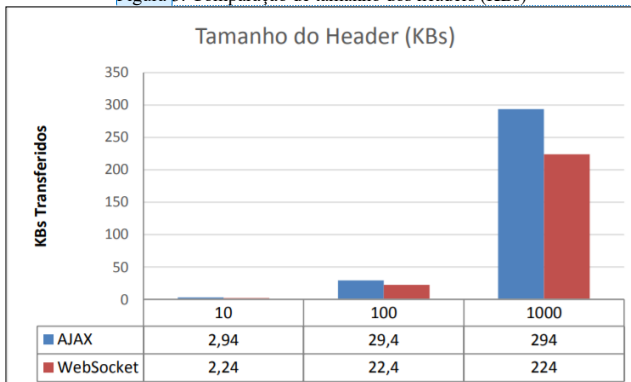
Figura 2: Comparação de tamanho dos formatos transferidos (KBs)



Fonte: Fernandes (2013).

Comentado [DSdR11]: A figura deve ser citada e descrita no texto.

Figura 3: Comparação de tamanho dos headers (KBs)



Fonte: Fernandes (2013).

Comentado [DSdR12]: A figura deve ser citada e descrita no texto.

O resultado do trabalho concluiu que o protocolo WebSocket obteve um melhor resultado em todos os aspectos testados. Aponta também que “[...] WebSockets e o paradigma publisher/subscriber são as tecnologias mais adequadas na implementação de soluções que necessitem de integrar dados em tempo real. Soluções que podem ser bastante diversificadas como já foram referidas: finanças, apostas desportivas, dados, ou mesmo o Big Data.” (FERNANDES, 2013).

2.2 ANÁLISE DE DESEMPENHO DE APIS DE SOCKETS

Souza (2016) desenvolveu um estudo comparativo de APIs *socket* visando medir o desempenho. O trabalho comparou as seguintes APIs: ZeroMQ, NanoMSG e Berkeley Sockets.

O ZeroMQ foi projetado para facilitar o desenvolvimento de aplicações distribuídas, uma API concebida para ter o mínimo de complexidade. Oferece quatro tecnologias para transporte: TCP, *MULTICAST*, *Instructions per cycle* (IPC) e *Inter-Process Communication* (INPROC).

O NanoMSG é uma concorrente do ZeroMQ. Foi concebida com a intenção de aumentar a rapidez, prover estabilidade da camada de rede e facilitar o desenvolvimento. O NanoMSG provê todos os padrões do ZeroMQ e provê também o uso do protocolo WebSocket.

Berkeley Sockets foram desenvolvidos na década de 1980 na *University of California at Berkeley*. Suporta somente TCP e UDP, e é uma API de mais baixo nível se comparada as demais citadas anteriormente.

Souza (2016) escolheu como métricas de avaliação a vazão e latência. Vazão é a quantidade de mensagens lidas em um determinado tempo. Latência é o tempo gasto para leitura de cada mensagem. Souza (2016) traz um comparativo entre as APIs e expõe alguns pontos relevantes: quantidade de linhas de código que são necessárias para realizar as operações de criar um socket, conectar, enviar dados, receber dados e fechar a conexão; qual a facilidade da programação; qual o desempenho com a utilização de cada uma das aplicações.

Figura 4: Comparativo das APIs

API	Quantidade de Código Gerado	Facilidade de Programação	Desempenho
Berkeley	Muito	Pouco	Excelente
ZeroMQ	Pouco	Muito	Razoável
NanoMSG	Muito	Média	Excelente

Fonte: Souza (2016).

A conclusão do trabalho, após a análise das métricas obtidas, foi que a NanoMSG obteve o melhor desempenho para os cenários dos experimentos. Souza (2016) enfatiza que esse trabalho pode ser usado como referência pela comunidade para escolha de uma das opções citadas no trabalho.

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

Silva (2017) conta que desenvolver aplicações IoT, à primeira vista, parece ser uma tarefa fácil. Ele estimou 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 com o propósito de 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) teve 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. O 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.

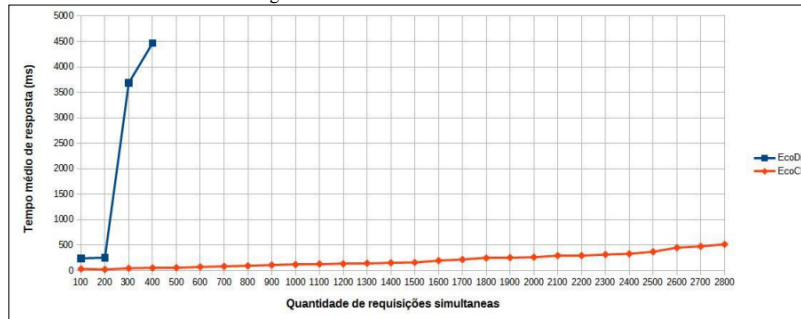
A EcoCIT surge com o objetivo de solucionar as limitações que a EcoDiF possui. A arquitetura é composta de diversos componentes distribuídos para permitir que a EcoCIT possa fazer uso da elasticidade que os ambientes de computação em nuvem podem prover. A arquitetura permite que a aplicação seja executada em máquinas virtuais que trabalham de 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 uma plataforma escalável.

Silva (2017) realizou três 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 *feeds* são basicamente os dados que os dispositivos enviam para o servidor. 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 comparar 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.

Comentado [DSdR13]: A figura deve ser citada e descrita no texto.

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 DO MIDDLEWARE

Esta seção 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 entregando um *middleware* para comunicação de aplicações Web para dispositivos móveis e serviços em um ambiente distribuído. Durante a pesquisa e nos trabalhos correlatos selecionados, nenhum deles apresenta uma proposta de arquitetura de comunicação de aplicações Web para dispositivos móveis com servidores que tenham o objetivo de ser escaláveis e resilientes. 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 tire a responsabilidade dos serviços de se preocuparem com a entrega de mensagens para aplicações móveis.

Quadro 3 - Comparativo dos trabalhos correlatos

Características \ Trabalhos Correlatos	Fernandes (2013)	Souza (2016)	Silva (2017)
Escalável			X
Resiliente	X	X	
Utilização de alguma ferramenta de mensageria			X
Voltado para comunicação Web	X		

Fonte: elaborado pelo autor.

No Quadro 1 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 Fernandes (2013) apresenta um *framework* para aplicações *real time web*, que se assemelha a este trabalho por ser voltado para a Web. Entretanto, o trabalho não é apresentado como uma solução escalável e para um contexto distribuído. Já o trabalho de Souza (2016) comparou diversas APIs de sockets, porém, ele trabalhou em uma camada mais baixo nível e não com uma comunicação Web. O trabalho dele pode ser utilizado para a execução deste trabalho na implementação dos sockets de comunicação. 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

O *middleware* da arquitetura proposta 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);
- possibilitar que os dispositivos móveis enviem mensagens quando o serviço estiver fora do ar (RF);
- possibilitar 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 não 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:

- estudar as tecnologias e ferramentas disponíveis no mercado;
- especificar um diagrama com os componentes internos do *middleware*;
- desenvolver o *middleware* proposto;
- desenvolver uma aplicação Web para dispositivos móveis e dois serviços para testar o *middleware*;
- testar a aplicação e avaliar os resultados;
- escrever os resultados do trabalho.

As etapas serão realizadas nos períodos relacionados no Quadro 6.

Quadro 4 - Cronograma

etapas / quinzenas	2022									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
estudar as tecnologias e ferramentas disponíveis no mercado	X	X								
especificar um diagrama com os componentes internos do <i>middleware</i>		X	X							
desenvolver o <i>middleware</i> proposto				X	X	X				
desenvolver uma aplicação Web para dispositivos móveis e dois serviços para testar o <i>middleware</i>							X	X		
testar a aplicação e avaliar os resultados									X	
escrever os resultados do trabalho		X	X					X	X	X

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve brevemente os assuntos que fundamentarão o estudo a ser realizado para construir o *middleware* que possa resolver o problema exposto anteriormente: apresentar um *middleware* que seja escalável e resiliente para comunicação cliente/servidor entre aplicações Web para dispositivos móveis e serviços 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 esclarecem 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).

Comentado [DSdR14]: A seção de Revisão Bibliográfica do pré-projeto para o projeto deve ser complementada com mais conteúdo!!!

REFERÊNCIAS

FERNANDES, Pedro Manuel da Conceição. **Desenvolvimento de uma Framework Real Time Web para HTML5**. 2013. TCC (Graduação) - Curso de Licenciatura em Sistemas e Tecnologia da Informação, Universidade Atlântica, Barbarena, 2013.

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

MICROSOFT AZURE. **O que é middleware?** Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-middleware>. Acesso em: 20 de novembro de 2021.

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.

SOUZA, Samuel Pereira de. **ANÁLISE DE DESEMPENHO DE APIS DE SOCKETS**. 2016. TTC (Graduação) - Curso de Graduação em Redes de Computadores, Universidade Federal do Ceará, Quixadá, 2016.

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

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR TCC I

Avaliador(a): Dalton Solano dos Reis

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. JUSTIFICATIVA São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
ASPECTOS METODOLÓGICOS	Os métodos, recursos e o cronograma estão devidamente apresentados?	X		
	5. 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?		X	
	6. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		
	7. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?	X		
	8. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?		X	
	9. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?	X		
As citações obedecem às normas da ABNT?		X		
Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?		X		

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: (X) APROVADO () REPROVADO

Revisão do Pré-projeto

Disciplina: Trabalho de Conclusão de Curso I – BCC

Caro orientando,

segue abaixo o Termo de Compromisso, as DUAS revisões do seu pré-projeto contendo a avaliação do professor “avaliador” e professor “TCC1”, junto com as avaliações da defesa na banca de qualificação. É muito importante que revise com cuidado e discuta possíveis dúvidas decorrente das revisões com o seu professor orientador, e com o professor de TCC1. Sempre procure fazer todos os ajustes solicitados, até mesmo o menores detalhes, pois todos são importantes e irão refletir na sua nota nesta disciplina.

Mas, caso o professor orientador julgue que algumas anotações das revisões não devam ser feitas, ou mesmo que sejam feitas de forma diferente a solicitada pelo revisor, anexe ao final do seu projeto a ficha “Projeto: Observações – Professor Orientador” disponível no material da disciplina, e justifique o motivo.

Lembrem que agora o limite de páginas do projeto é no máximo 12 (doze) páginas. E que a seção de “Revisão Bibliográfica” deve ser complementada.

Atenciosamente,

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

TERMO DE COMPROMISSO

I – IDENTIFICAÇÃO DO ALUNO	
Nome:	Ariel Adonai Souza
CV Lattes:	http://lattes.cnpq.br/7521486727613551
E-mail:	arieladonai@furb.br
Telefone:	(47) 99218-1824
II – IDENTIFICAÇÃO DO TRABALHO	
Título provisório:	COMUNICAÇÃO DE APLICATIVOS MÓVEIS: UM MODELO DE ARQUITETURA ESCALÁVEL E RESILIENTE
Orientador:	Francisco Adell Péricas
Coorientador (se houver):	
Linha de Pesquisa:	<input type="checkbox"/> Tecnologias aplicadas à informática na educação <input checked="" type="checkbox"/> Tecnologias aplicadas ao desenvolvimento de sistemas
III – COMPROMISSO DE REALIZAÇÃO DO TCC	
Eu (aluno),	Ariel Adonai Souza
comprometo-me a realizar o trabalho proposto no semestre 2022/1, de acordo com as normas e os prazos determinados pela FURB, conforme previsto na resolução nº.20/2016.	
Assinatura:	NÃO É NECESSÁRIO – Encaminhar por e-mail ao orientador
IV – COMPROMISSO DE ORIENTAÇÃO	
Eu (orientador),	Francisco Adell Péricas
comprometo-me a orientar o trabalho proposto no semestre 2022/1, de acordo com as normas e os prazos determinados pela FURB, conforme previsto na resolução nº.20/2016.	
Assinatura:	NÃO É NECESSÁRIO – Encaminhar por e-mail ao professor de TCC I

Blumenau, 07 de Agosto de 2021

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 contam que em empresas com aplicações Web, os desenvolvedores reconstroem 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.

A definição de 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;

Comentado [GJ15]: Não utilizar 1ª pessoa

Comentado [GJ16]: seriam "softwares desktops" ou "softwares web"?
Talvez utilizar "softwares corporativos" ou "aplicações corporativas".

Excluído: nos

Comentado [GJ17]: Acho que vale a pena explicar o porquê das empresas optarem por esta abordagem.

Comentado [GJ18]: Isto é uma opinião sua ou de Oehlman e Blanc (2012). Se for desses autores, referenciar.

Comentado [GJ19]: "viável" foi usado no final do parágrafo abaixo. Evitar repetição próximas.

Excluído: Para Tanenbaum, Steen (2008), a

Excluído: um sistema distribuído é

Comentado [GJ20]: Os sistemas distribuídos podem ser constituídos de dispositivos móveis. Talvez o mais apropriado seria dizer que "há algumas dificuldades no desenvolvimento de sistemas distribuídos em que os componentes são dispositivos móveis".

Comentado [GJ21]: quem disse isso? Citar uma obra

Comentado [GJ22]: quem disse isso? Citar uma obra

Comentado [GJ23]: serviço ou sistema? O "serviço" seria um sistema centralizado?

Comentado [GJ24]: O "dispositivo móvel" seria apenas o hardware. A comunicação deste serviço seria com o software que está sendo executado neste dispositivo móvel, correto?

Comentado [GJ25]: Esta arquitetura seria "cliente/servidor" ou "p2p"?

Comentado [GJ26]: Parece ser um objetivo bem amplo. Sugiro delimitar.

- d) deve atender a um contexto de computação distribuída, ou seja, suportar comunicação entre diversas aplicações, não somente uma ou duas.

2 TRABALHOS CORRELATOS

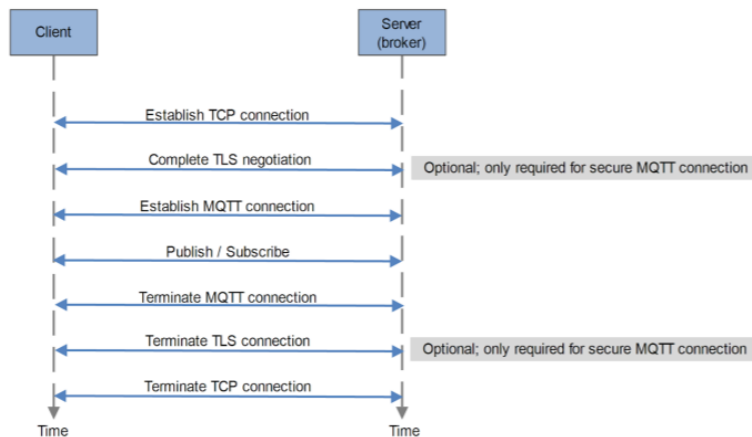
Não foram encontrados trabalhos correlatos que fossem diretamente relacionados ao objetivo deste estudo. Desta forma, foram utilizados trabalhos que possuam um ou mais aspectos do objetivo deste estudo. O primeiro trabalho foi desenvolvido por Koneski (2018) e descreve a implementação da comunicação 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 insetos de erva 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 a 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 3: 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.

Comentado [GJ27]: Sistemas de mensageria, como "ActiveMq", "Apache Kafka" e "RabbitMq" parecem atender ao objetivo principal de possibilitar a construção de um sistema distribuído e escalável. Com a leitura da Introdução deste pré-projeto me veio imediatamente à mente estas tecnologias. Não entendi o que se pretende resolver que estas ferramentas não resolvam. O que este trabalho vai resolver de diferente?

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 **trafegados** em formato JSON para a comunicação entre os dispositivos e o servidor, bem como o armazenamento também é em JSON.

Excluído: formados

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, **hospedados** 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.

Excluído: tudo

Excluído: possui

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).

Comentado [GJ28]: uso de 1ª pessoa

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 **de** 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.

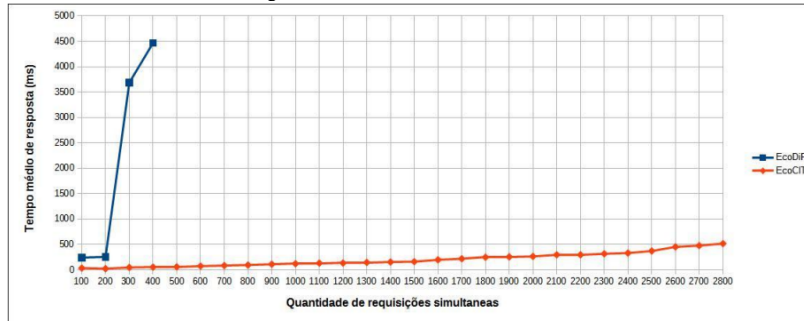
Silva (2017) realizou 3 experimentos em seu trabalho. O primeiro teve como objetivo avaliar a substituição do sistema gerenciador de banco de dados utilizado no **gerenciamento de feeds**. Os resultados mostraram que a substituição do banco de dados contribuiu para a melhor performance do processo. O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

Comentado [GJ29]: O que são os feeds?

Excluído: altor

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 tire a responsabilidade dos serviços de se preocuparem com a entrega de mensagens para aplicações móveis.

Quadro 5 - Comparativo dos trabalhos correlatos

Características \ Trabalhos Correlatos	Koneski (2018)	Kodali, Gorantla (2017)	Silva (2017)
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:

Comentado [GJ30]: Acredito que estes três experimentos podem ser resumidos aqui. Não precisa detalha-los tanto.

Comentado [GJ31]: Ainda não está claro qual é o problema a ser resolvido.

- a) ser possível comportar uma volumetria cada vez maior (escalável) (RF);
- b) armazenar as mensagens até conseguir entregá-las ao destino (RF);
- c) garantir a entrega das mensagens pendentes (ainda não enviadas para o destinatário) assim que os dispositivos conectarem novamente (RF);
- d) ser possível que os dispositivos móveis enviem mensagens quando o serviço estiver fora do ar (RF);
- e) ser possível que os serviços enviem mensagens quando o dispositivo móvel estiver fora do ar (RF);
- f) 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);
- g) utilizar um banco de dados relacional para armazenar as mensagens temporariamente (Requisito Não Funcional - RNF);
- h) utilizar uma aplicação de mensageria para troca de mensagens internas utilizando o protocolo AMQT (RNF);
- i) possibilitar a comunicação dos dispositivos móveis através do protocolo WebSocket (RNF);
- j) possibilitar a comunicação dos serviços através do protocolo AMQT (RNF);

Comentado [GJ32]: Usando um banco de dados relacional é possível garantir isso?

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:

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

As etapas serão realizadas nos períodos relacionados no Quadro 6.

Quadro 6 - 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.

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Avaliador(a): **Gilvan Justino**

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS ¹		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?		X	
	O problema está claramente formulado?		X	
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?		X	
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?		X	
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?		X	
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	x		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?		X	
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	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?	X		
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?		X	



UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DISCIPLINA: TRABALHO DE CONCLUSÃO DE CURSO I
CURSO: CIÊNCIA DA COMPUTAÇÃO - BCC

ATA DA DEFESA: BANCA DO PRÉ-PROJETO

Venho, por meio deste, manifestar minha avaliação sobre a **apresentação** do Pré-Projeto de TCC

realizado pelo(a) acadêmico(a), Ariel Adonai Souza no **SEGUNDO SEMESTRE DE 2021**, com o título **COMUNICAÇÃO DE APLICATIVOS MÓVEIS: UM MODELO DE ARQUITETURA ESCALÁVEL E RESILIENTE**, sob orientação do prof(a). Francisco Adell Péricas.

A referida apresentação obteve a seguinte nota:

Componente da Banca	Nota (de 0 a 10)
Professor(a) Avaliador(a): Gilvan Justino	7

ATENÇÃO. A nota acima se refere somente a apresentação do pré-projeto e vai ser repassada para o aluno (orientando). Favor preencher os campos acima e enviar por e-mail ao professor de TCC1 (dalton@furb.br). Não passar o arquivo com as anotações da revisão já enviado ao professor de TCC1 para o orientando e nem para o professor orientador. Após o professor de TCC1 receber esta ata preenchida, o professor de TCC1 vai disponibilizar para o orientando/orientador os arquivos com as revisões. Caso julgue necessário fazer mais alguma consideração relacionada ao pré-projeto ou a defesa, favor usar o espaço abaixo.

Observações da apresentação:

Através da apresentação do aluno não ficou claro o que se pretendia construir.

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:

Comentado [DSdR33]: Oehlman e Blanc (2012)

Comentado [DSdR34]: Local Area Network (LAN)

Comentado [DSdR35]: de “um

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

2 TRABALHOS CORRELATOS

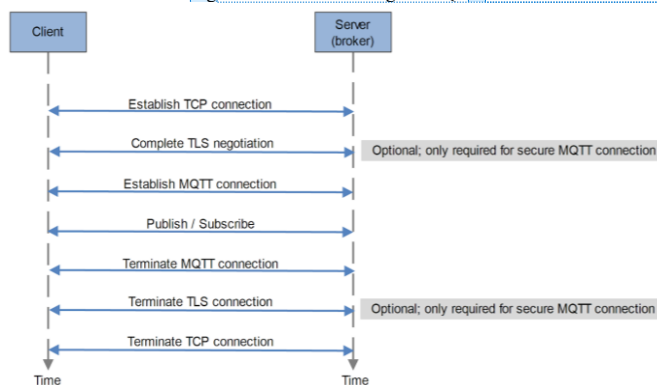
Não foram encontrados trabalhos correlatos que fossem diretamente relacionados ao objetivo deste estudo. Desta forma, foram utilizados trabalhos que possuam um ou mais aspectos do objetivo deste estudo. O primeiro trabalho foi desenvolvido por Koneski (2018) e descreve a implementação da comunicação 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 inços de erva 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 a 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 4: 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.

Comentado [DSdR36]: Borda envolta da figura.

Comentado [DSdR37]: Remover ponto final.

Comentado [DSdR38]: realizar o envio

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 comparar 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.

Comentado [DSdR39]: JavaScript Object Notation (JSON)→

Comentado [DSdR40]: HyperText Transfer Protocol (HTTP)

Comentado [DSdR41]: Juntar com o parágrafo anterior. Evitar parágrafos com uma só frase.

Comentado [DSdR42]: surgiu abstrair

Comentado [DSdR43]: EcoDiF. O EcoDiF “[...] é

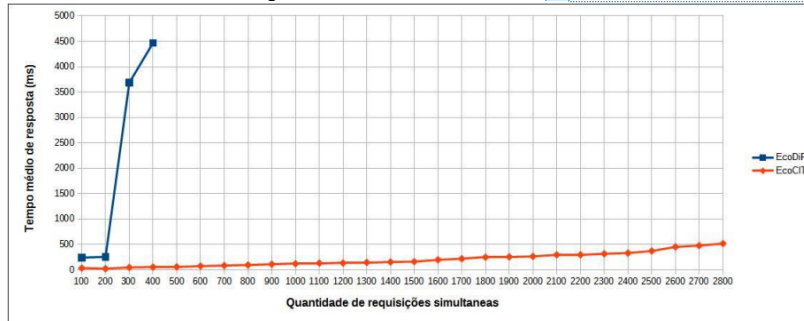
Comentado [DSdR44]: trabalham de maneira

Comentado [DSdR45]: Juntar com parágrafo anterior.

Comentado [DSdR46]: três

Comentado [DSdR47]: comparar

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 tire a responsabilidade dos serviços de se preocuparem com a entrega de mensagens para aplicações móveis.

Quadro 7 - 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);

Comentado [DSdR52]: Ponto final.

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.

Comentado [DSdR53]: Juntar com o parágrafo anterior

Comentado [DSdR54]: Iniciar os itens com letra minúscula.

Comentado [DSdR55]: estudar as

Comentado [DSdR56]: especificar

As etapas serão realizadas nos períodos relacionados no Quadro 6.

Quadro 8 - 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.

Comentado [DSdR57]: maio

Comentado [DSdR58]: Iniciar os itens com letra minúscula.

Comentado [DSdR59]: estudar

Comentado [DSdR60]: especificar

Comentado [DSdR61]: desenvolver

Comentado [DSdR62]: testar

Comentado [DSdR63]: avaliar

Comentado [DSdR64]: escrever

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.

Comentado [DSdR65]: Esta seção

Comentado [DSdR66]: "[...] um

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.

Comentado [DSdR67]: Padrão ABNT para congresso.

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.

Comentado [DSdR68]: Não negrito.
Só a inicial em maiúsculo.

Comentado [DSdR69]: Não negrito.

Comentado [DSdR70]: Não negrito.

Comentado [DSdR71]: Só as iniciais em maiúsculo.

Comentado [DSdR72]: Não negrito.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR TCC I

Avaliador(a): Dalton Solano dos Reis

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. JUSTIFICATIVA São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
ASPECTOS METODOLÓGICOS	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados?		X	
	5. 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?	X		
	6. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		
	7. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?	X		
	8. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?		X	
	9. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?		X	
	As citações obedecem às normas da ABNT?	X		
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?	X		

Comentado [DSdR73]: Indicados no texto.

Comentado [DSdR74]: Indicados no texto.

Comentado [DSdR75]: Indicados no texto.



UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DISCIPLINA: TRABALHO DE CONCLUSÃO DE CURSO I
CURSO: CIÊNCIA DA COMPUTAÇÃO - BCC

ATA DA DEFESA: BANCA DO PRÉ-PROJETO

Venho, por meio deste, manifestar minha avaliação sobre a **apresentação** do Pré-Projeto de TCC realizado pelo(a) acadêmico(a), Ariel Adonai Souza no **SEGUNDO SEMESTRE DE 2021**, com o título COMUNICAÇÃO DE APLICATIVOS MÓVEIS: UM MODELO DE ARQUITETURA ESCALÁVEL E RESILIENTE.

A referida apresentação obteve a seguinte nota:

Componente da Banca	Nota (de 0 a 10)
Professor(a) Orientador(a): Francisco Adell Péricas	9,0

A apresentação aconteceu em 21/10/2021 na sala de reunião virtual do MS-Teams, tendo início às 8h05 hs e foi encerrada às 8h35 hs.

ATENÇÃO. A nota acima se refere somente a apresentação do pré-projeto e vai ser repassada para o aluno (orientando). Favor preencher os campos acima e enviar por e-mail ao professor de TCC1 (dalton@furb.br). Lembro que os arquivos com as anotações das revisões do professor de TCC1 e Avaliador serão enviados para o orientando e professor orientador após o professor de TCC1 receber esta ata preenchida. Caso julgue necessário fazer mais alguma consideração relacionada ao pré-projeto ou a defesa, favor usar o espaço abaixo.

Observações da apresentação:

A apresentação foi inicialmente marcada para dia 21/10/2021, às 8h00, porém o aluno não compareceu. Remarcada para dia 25/10/2021, a pré-banca seguiu normalmente.