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

PROTOCOLO PARA CORREIO ELETRÔNICO BASEADO EM BLOCKCHAIN

Ruan Schuartz Russi

Mauro Marcelo Mattos

1 INTRODUÇÃO

A popularidade do protocolo SMTP (Simple Mail Transfer Protocol) é inegável. Estima-se que, a cada minuto, 188 milhões de novos e-mails são enviados (PEZZOTTI, 2019). No ano de 2019, o número global de usuários de correio eletrônico totalizava 3,9 bilhões de usuários, sendo que o número deve crescer para 4,3 bilhões até 2023, número esse equiparado a metade da população mundial (NITRONEWS, 2020). Toda essa popularidade do correio eletrônico é atribuída a diversos fatores, sendo um deles a sua idade. O e-mail é considerado o serviço online mais antigo do mundo, tendo a primeira mensagem da história sido enviada no ano de 1971 (OLIVEIRA, 2020).

Apesar de ser uma tecnologia difundida e robusta, a idade da solução acaba trazendo para a mesma alguns pontos negativos. Na época em que os serviços de correio eletrônico foram desenhados, a internet como se apresenta hoje não existia e as redes de computadores existentes eram restritas a ambientes acadêmicos, não sendo liberado acesso para a população (TRAININI e CARISSIMI, 2005). Neste ambiente controlado, a preocupação com a segurança da informação era mínima e dispositivos de segurança geralmente não eram implementados. O cenário mudou com o advento da criação e popularização da internet. A rede se tornou um lugar hostil e perigoso, sendo utilizada por usuários mal-intencionados para a realização de práticas ilícitas. Por não ter sido criado com este ambiente hostil em mente, os serviços de correio eletrônico possuem uma diversidade de vulnerabilidades passíveis de serem exploradas como envio de spam, propagação de vírus, vazamento de mensagens e falsificação de identidade (TRAININI e CARISSIMI, 2005).

Uma característica importante dos servidores de e-mail é que eles formam uma rede descentralizada não existindo assim uma entidade com poder absoluto sobre as outras. Apesar disso, a implementação, devido a época em que foi feita, não aproveita características importantes de redes descentralizadas modernas. A blockchain por exemplo, tecnologia criada para dar vida ao Bitcoin, permite ter um histórico imutável de tudo que foi feito. A implementação do Bitcoin consegue garantir que um determinado usuário realizou uma transação financeira para outro de forma imutável. Já com o SMTP, é impossível determinar de modo fidedigno que um usuário enviou uma mensagem para outro. Além disso, atualmente é comum a utilização de serviços de e-mail de terceiros. Em 2016, o serviço de correio eletrônico da Google, o Gmail, chegou a marca de um bilhão de usuários ativos mensalmente (ESTADÃO, 2016). Essa dependência com entidades terceiras acaba exigindo que os usuários tenham plena confiança nas mesmas pois nada impede que elas modifiquem, excluam ou exponham mensagem privadas. Essas entidades terceiras podem até mesmo mandarem mensagens se passando pelo usuário.

Para resolver os problemas acima citados, o ideal seria a implementação de um protocolo totalmente novo que já nasça sem a presença destas falhas. O problema é que, devido a alta utilização dos serviços de e-mail, uma mudança abrupta de tecnologia se torna inviável, pois envolve uma mudança significativa na rotina de milhões de usuários. Diante deste cenário, o presente trabalho propõe o desenvolvimento de um protocolo baseado em blockchain que consiga enviar e receber mensagens de correio eletrônico de modo descentralizado e seguro. O protocolo será compatível com o protocolo SMTP permitindo assim uma migração gradual de tecnologia.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar um protocolo de envio e recebimento de e-mails baseado em blockchain e que seja compatível com o protocolo SMTP.

Os objetivos específicos são:

a) especificar um modelo de envio e recebimento de mensagens baseado no conceito de blockchain;
b) desenvolver um protótipo de cliente para demonstrar a funcionalidade do protocolo;
c) criar um conjunto de casos de testes para validar o protocolo desenvolvido;

2 TRABALHOS CORRELATOS

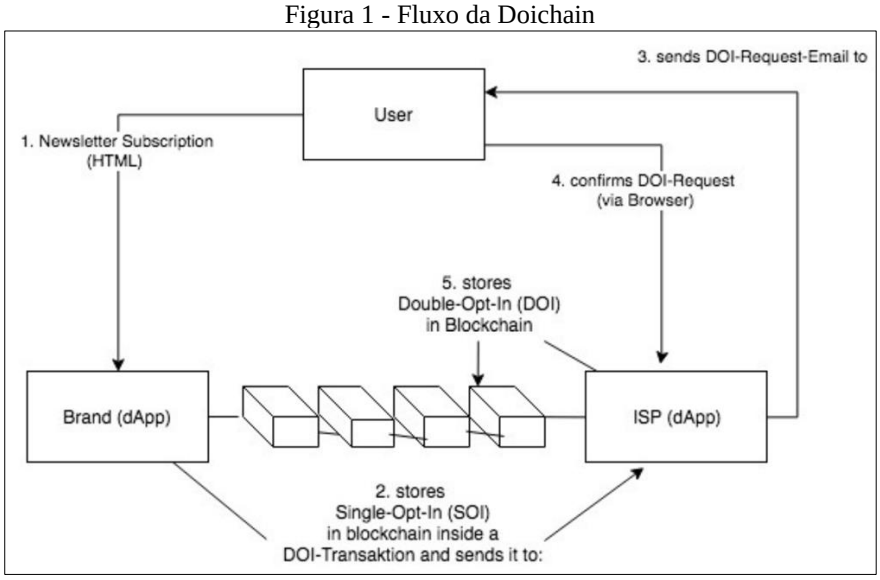
Nesta seção são apresentados trabalhos que apresentam semelhança com os principais objetivos do trabalho proposto. O primeiro trabalho é a implementação de uma blockchain capaz de armazenar e validar inscrições de usuários em *newsletters* (KRAUSE, 2018). O segundo é o estudo e implementação de um protocolo p2p para a troca segura de mensagens (WARREN, 2012). Por fim, na seção 2.3 será apresentado o trabalho de Grottenthaler (2017) que busca fazer a ligação entre envio de e-mails e transações de criptomoeda.

2.1 CONTROLE DE INSCRIÇÃO PARA RECEBIMENTE DE E-MAIL PROMOCIONAL

O trabalho desenvolvido por Krause (2018) consiste em um sistema baseado em blockchain que é capaz de provar que um determinado usuário se inscreveu em uma campanha de mail marketing. A motivação para o trabalho é resolver alguns casos comuns de spam e possibilitar a adesão de empresas de mail marketing a leis relacionadas a proteção de dados. Para tal, o autor propôs uma blockchain que armazena as inscrições dos usuários e pode ser consultada pelo dono da campanha de marketing antes de enviar algum e-mail para o usuário. Para se comunicar com a blockchain, o autor desenvolveu um aplicativo chamado dApp que serve unicamente para realizar esta comunicação. Tanto o dono da campanha de marketing quanto o dono do servidor de e-mail que o usuário está utilizando precisam configurar uma instância local do dApp.

O fluxo da plataforma proposta por Krause (2018) pode ser conferido na Figura 1. Primeiramente, o usuário (User) se inscreve em alguma campanha de e-mail. Um método comum de inscrição é a submissão do e-mail em um formulário presente em algum site. A empresa responsável pela campanha (Brand), ao receber a inscrição envia para a blockchain uma transação ainda não confirmada. O servidor de e-mail do usuário (Internet Service Provider - ISP), ao receber a transação não confirmada, envia ao usuário (User) um e-mail pedindo a confirmação da inscrição. Caso o usuário confirme inscrição, a transação é salva como confirmada na blockchain. Sendo assim, sempre que for enviar algum e-mail, o dono da campanha pode checar se o usuário realmente se inscreveu para aquela campanha.

Figura 1 - Fluxo da Doichain



Fonte: Krause (2018).

O trabalho de Krause (2018) consegue de fato ser uma solução para o problema que ele se propõe a resolver, porém, implantar tal solução em um cenário produtivo acaba exigindo bastante esforço. Por mais que a migração seja transparente para o usuário, a solução exige que pelo menos os servidores de e-mail mais populares do mercado se adequem a plataforma para que tenha algum efeito. Além disso, os donos das campanhas precisariam reconfigurar toda a sua infraestrutura para passar a se comunicar com a blockchain e validar as inscrições dos usuários. O ponto é, mesmo exigindo tamanho esforço, a plataforma se limita a resolver um cenário muito específico deixando ainda uma série de problemas sem solução.

2.2 PROTOCOLO P2P SEGURO PARA TROCA DE MENSAGENS

O protocolo desenvolvido por Warren (2012) visa resolver os problemas atuais que o protocolo SMTP apresenta como segurança das mensagens e envio de spam. A ideia é baseada no funcionamento do Bitcoin e traz alguns conceitos da arquitetura da criptomoeda para o mundo do correio eletrônico. Um desses conceitos, é um sistema de *Proof of Work* para envio de mensagens. Sempre que um determinado usuário decidir enviar uma mensagem, ele precisa realizar algum processamento computacional que é configurado para durar em média quatro minutos. Desse modo, um usuário ao tentar fazer envio de spam estaria limitado a enviar apenas uma mensagem a cada quatro minutos. Outro ponto inspirado no Bitcoin é o endereçamento baseado em chaves públicas. No

protocolo proposto, não existe o conceito de domínio de e-mail. O endereço utilizado para identificar um usuário é um valor gerado a partir da sua chave pública. É impossível que qualquer um que não seja o remente ou o destinatário da mensagem identifique os endereços envolvidos nela.

O fluxo da proposta de Warren (2012) se inicia com um usuário querendo enviar uma mensagem para algum outro usuário da rede. O remetente precisa, antes de tudo, da chave pública do destinatário. Em posse da chave pública, ele faz a criptografia da mensagem e envia para a rede. Todos os nós conectados na rede recebem a mensagem, porém, só o destinatário consegue ler a mensagem pois é o único que possui a chave capaz de descriptografar ela. Ao receber a mensagem, o destinatário precisa enviar uma confirmação de recebimento. Caso o remetente não receba a confirmação em até dois dias ele faz a retransmissão após mais dois dias. O processo de retransmissão será feito até que o destinatário confirme o recebimento.

Pelo fato de a solução de Warren (2012) se propor a ser uma rede descentralizada, todos os nós da rede armazenam todas as mensagens. Para evitar a necessidade de uma alta capacidade de armazenamento, cada nó está configurado para apagar as mensagens não referentes a ele a cada dois dias. Mesmo assim é possível que a capacidade de armazenamento para guardar todas as mensagens dos últimos dois dias seja muito grande. Devido a isso, o autor implementou uma lógica que, após um determinado limite de armazenamento ser atingido, os nós começam a se agrupar em clusters dividindo o armazenamento entre si.

A solução de Warren (2012) também apresenta uma proposta para campanhas de marketing. Primeiramente, o dono da campanha precisa publicar uma chave que consiga descriptografar as mensagens enviadas por ele. Os usuários que quiserem receber as mensagens precisam pegar a chave e configurar o seu nó para aceitar mensagens para ela. Deste modo, sempre que um nó receber uma mensagem, ele vai tentar primeiramente abrir ela com a chave privada do usuário e, não tendo sucesso, vai tentar com todas as chaves presentes na lista de inscrições do nó.

O trabalho de Warren (2012) consegue resolver a grande maioria dos problemas atuais envolvendo correio eletrônico. A solução é fácil de usar e configurar e já possuí criptografia por padrão eliminando a necessidade da utilização de tecnologias terceiras. Apesar dos seus vários pontos positivos, a solução tem o ponto fraco de não ser compatível com o SMTP. Desse modo, seria necessária uma mudança abrupta de tecnologia, o que acaba se tornando inviável devido a grande utilização dos serviços de e-mail. Outro ponto fraco é a impossibilidade de identificar as partes relacionadas no envio de uma mensagem. Por mais que tal funcionalidade seja boa em alguns casos, ela acaba dificultando a adoção da tecnologia no meio corporativo.

2.3 LIGAÇÃO DE E-MAILS COM TRANSAÇÕES DE CRIPTOMOEDAS

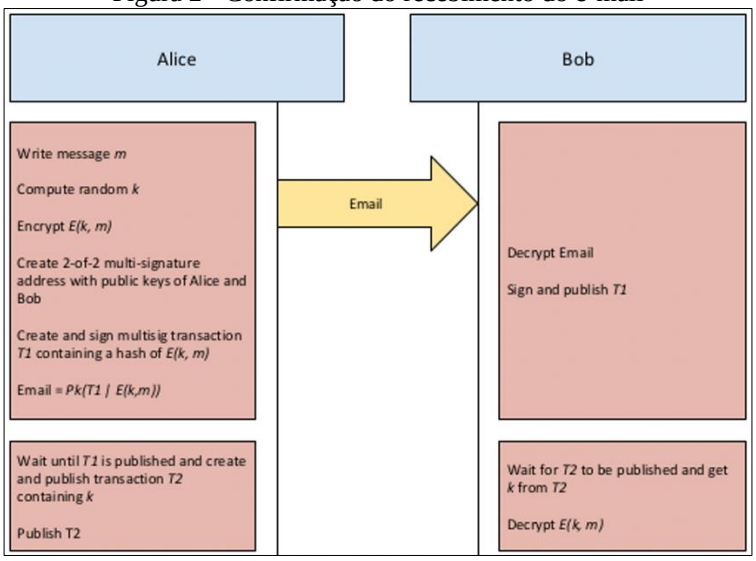
O trabalho desenvolvido por Grottenthaler (2017) busca resolver alguns problemas encontrados na implementação de correio eletrônico criando um relacionamento de cada e-mail enviado com uma transação de criptomoeda. As transações são salvas e validadas por uma blockchain. O estudo realizado foca na solução de dois problemas: envio de spam e confirmação do recebimento de mensagens.

O fluxo da plataforma proposta por Grottenthaler (2017) inicia com um usuário querendo enviar um e-mail para outro usuário. O remetente, para realizar o envio do e-mail, precisa adicionar dois cabeçalhos na requisição: Mailcoin-Txid e Mailcoin-Header. O Mailcoin-Txid é o id da transação que será salva na blockchain e serve para o destinatário identificar a transação e validar se é autêntica. O Mailcoin-Header contém informações referentes ao envio, sendo elas a versão do sistema, a data atual, o e-mail do destinatário e um valor gerado de modo aleatório que serve unicamente para evitar conflitos quando um *hash* precisar ser gerado utilizando tais dados. Após a geração dos valores, o remetente cria uma transação na blockchain contendo como id o Mailcoin-Txid e como valor um hash gerado a partir do Mailcoin-Header e então faz o envio do e-mail. O destinatário, ao receber o e-mail, recupera o Mailcoin-Txid do cabeçalho e busca pela transação na blockchain. Se ele confirmar que a transação existe, ele recupera o valor do hash armazenado na transação, gera o hash utilizando o valor que vier no cabeçalho Mailcoin-Header e então compara os valores. Se forem iguais, a mensagem é autêntica.

Para o tratamento dos casos de spam, a solução criada por Grottenthaler (2017) exige que qualquer envio de e-mail gere algum custo financeiro para o remetente. Esse custo é descontado das criptomoedas que o usuário possui na blockchain. O valor cobrado é muito baixo, porém, levando isso para um cenário de milhares de envios como acontece nos casos de spam, o valor final ficaria muito alto e dificultaria a prática de tal atividade. Um problema aparente é que qualquer pessoa teria custo para enviar e-mails, mesmo que ela não esteja enviando spam. Para resolver este problema, a plataforma encoraja os usuários a devolverem o custo da transação para o remetente. Caso o remetente não receba o valor de volta, ele pode decidir não enviar mais mensagens para o usuário que não fez a devolução.

A arquitetura do fluxo desenvolvido para identificar se um usuário de fato recebeu um e-mail é apresentada na Figura 2. Grottenthaler (2017) propôs um sistema baseado em dupla assinatura, na qual, para um usuário conseguir ler um e-mail, ele obrigatoriamente precisa notificar o remetente que recebeu a mensagem. Por exemplo, a usuária Alice, ao enviar um e-mail para o usuário Bob, criptografa a mensagem com uma chave aleatória `k` e cria uma transação no qual assina com a sua chave privada. No momento que Bob recebe a transação, ele identifica que é uma transação de dupla assinatura e que então, para ler o conteúdo do e-mail, precisa da chave `k`. Sendo assim, ele assina a transação recebida e a envia para a blockchain. Quando Alice recebe a confirmação de que a transação foi assinada, ela cria uma nova transação contendo a chave `k`. Bob então com o valor da chave `k` consegue acessar o conteúdo do e-mail. Com tal sistema é possível identificar se o usuário recebeu o e-mail pois basta verificar se ele realizou o processo de assinatura da transação.

Figura 2 - Confirmação do recebimento do e-mail



Fonte: Grottenthaler (2017).

O trabalho de Grottenthaler (2017) consegue criar uma extensão para o protocolo STMP trazendo para o mesmo várias vantagens oriundas da utilização de blockchain, porém, por se tratar de uma extensão, a base de tudo ainda continua sendo o SMTP. Desse modo, a solução não consegue resolver problemas que são inerentes do protocolo como, por exemplo, a garantia de imutabilidade das mensagens e a garantia que um determinado usuário enviou de fato uma mensagem com um determinado conteúdo.

3 PROPOSTA DO PROTOCOLO

Nesta seção serão apresentadas as justificativas para o desenvolvimento do estudo proposto, bem como um quadro comparativo com os trabalhos correlatos, os requisitos funcionais e não funcionais e a metodologia utilizada no desenvolvimento da solução.

3.1 JUSTIFICATIVA

Devido a sua grande massa de usuários, os serviços de e-mail passaram a ser um alvo muito atraente para criminosos. Eles buscam explorar as mais diversas vulnerabilidades encontradas neste tipo de serviço. Além disso, existe uma tendência mundial para a concentração de usuários em servidores SMTP de terceiros como Gmail e Outlook. Isso acaba exigindo que os usuários confiem suas informações pessoais e mensagens a empresas privadas que podem realizar qualquer tipo de processo sobre elas. Diante desse cenário, é natural que existam trabalhos que visam resolver tais problemas. O Quadro 1 faz uma análise das características dos trabalhos correlatos encontrados.

Quadro 1 – Comparativo entre os trabalhos correlatos

Características	Doichain	MailCoin	Bitmessage
Compatível com SMTP	Sim	Sim	Não
Sistema de Proof of Work	Não	Sim	Sim
Controle de inscrição em newsletter	Sim	Não	Não
Mensagens imutáveis	Não	Não	Sim
Assinatura de mensagens	Não	Sim	Sim
Controle para spam	Não	Sim	Sim
Identificação dos usuários	Sim	Sim	Não

Fonte: elaborado pelo autor.

Conforme análise do Quadro 1, é possível identificar que os trabalhos buscam resolver diferentes problemas. O Doichain, por estar unicamente centrado no processo de identificar se um usuário se inscreveu em um *newsletter*, acaba não abordando nenhum outro cenário. O MailCoin consegue abordar um universo maior de possibilidades, porém acaba deixando de fora um ponto importante que é a garantia de imutabilidade e histórico das mensagens. Isso acontece pois ele mantém como base o protocolo SMTP. Já o Bitmessage, por se tratar de um protocolo novo consegue ter solução para a grande maioria dos problemas citados, porém, ele tem o ponto fraco de exigir uma completa mudança de tecnologia, o que dificulta a migração para o protocolo. Além disso, o Bitmessage não permite que um terceiro usuário identifique os usuários envolvidos em uma mensagem. Isso se torna um impeditivo em ambientes corporativos.

Neste contexto, o presente trabalho se torna relevante pois tem potencial para impactar diretamente os bilhões de usuários de sistemas de correio eletrônico. Espera-se que esses usuários tenham disponível uma solução mais segura e robusta, que não apresente os problemas que os atuais sistemas de e-mail possuem. Além disso, o trabalho também deverá contribuir em pesquisas relacionadas a modernização dos serviços de e-mail.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos funcionais são:

a) permitir enviar um e-mail de um servidor que esteja utilizando o novo protocolo para um servidor SMTP que não esteja utilizando o novo protocolo;
b) permitir enviar um e-mail de um servidor SMTP para um servidor que esteja utilizando o novo protocolo;
c) permitir o envio de mensagens entre dois nós da rede utilizando o novo protocolo;
d) permitir identificação dos usuários através de um endereço;
e) permitir padrões diferentes de endereços para identificação de domínios pertencentes a organizações;
f) disponibilizar uma API que possibilite a comunicação de softwares clientes com a blockchain;

Os requisitos não funcionais são:

a) implementar os nós da rede utilizando a linguagem Golang;
b) utilizar a arquitetura REST para a API para comunicação com a blockchain;
c) receber mensagens SMTP na porta 587;
d) armazenar as mensagens em uma blockchain compartilhada entre todos os nós da rede;
e) obrigar a realização de um processo de Proof of Work para envio de mensagens;
f) criptografar as mensagens armazenadas na blockchain;

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

a) levantamento bibliográfico: realizar levantamento bibliográfico com relação a implementação de soluções utilizando blockchain. Pesquisar também a bibliografia referente a implementação de servidores SMTP;

b) elicitação de requisitos: nesta etapa será feito o refinamento dos requisitos tomando como base a pesquisa realizada;

c) especificação da arquitetura da blockchain: nesta etapa será definido como a blockchain deverá funcionar. Será especificada a estrutura de armazenamento, criptografia e envio dos e-mails utilizando diagramas da UML;

d) desenvolvimento da blockchain: será realizada a implementação da blockchain utilizando a linguagem Golang. Será desenvolvida com base na arquitetura especificada no passo c;

e) desenvolvimento da API: nesta etapa será desenvolvida a API responsável por fazer a interface com a blockchain;

f) desenvolvimento do servidor SMTP: implementar a compatibilidade dos nós com o protocolo SMTP;

g) testes: realizar o teste da rede validando o envio de mensagens e a compatibilidade com o protocolo SMTP;

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

	2021									
	fev.		mar.		abr.		mai.		jun.	
etapas / quinzenas	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação da arquitetura da blockchain										
desenvolvimento da blockchain										
desenvolvimento da API										
desenvolvimento do servidor SMTP										
testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Este capítulo tem como objetivo fazer um estudo inicial sobre os principais temas envolvidos no desenvolvimento do projeto. A seção 4.1 faz uma análise referente a implementação de sistemas descentralizados. A seção 4.2 aborda os fundamentos básicos referentes a blockchain. Por último, a seção 4.3 faz uma análise do funcionamento do protocolo SMTP.

4.1 SISTEMAS DESCENTRALIZADOS

Sistemas descentralizados, diferente dos sistemas cliente/servidor, não possuem nenhuma autoridade central. Cada integrante da rede pode atuar tanto como cliente quanto como servidor. Esses sistemas podem ser implementados utilizando várias abordagens diferentes, porém a mais famosa é a utilização da arquitetura P2P (pares em pares). De acordo com Silva (2010), os sistemas P2P são utilizados para diversos fins como computação distribuída, troca de mensagens, trabalho colaborativo e compartilhamento de dados.

Segundo Silva (2010), uma rede, para ser considerada P2P, deve possuir as seguintes características:

a) cada nó se conecta diretamente com os outros nós;

b) cada nó é responsável pelos seus dados;

c) um nó pode entrar e sair da rede a qualquer momento;

d) um nó pode atuar tanto como cliente quanto como servidor;

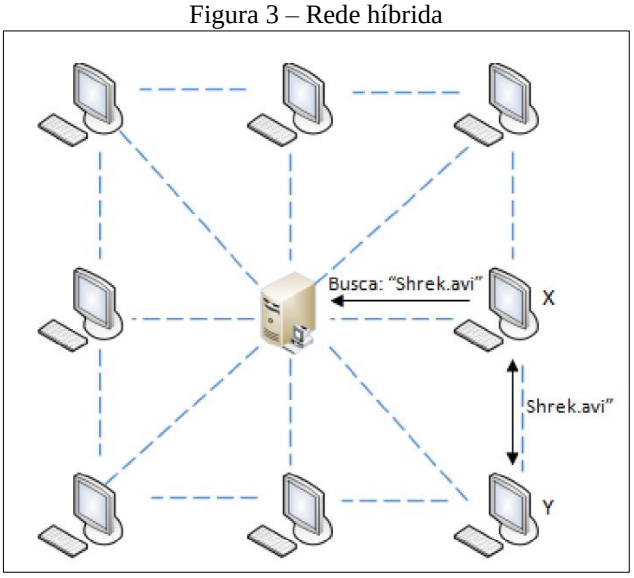
e) não existe autoridade central.

Conforme definido por Silva (2010), diferente dos sistemas cliente/servidor em que o número de servidores é fixo, sistemas P2P tendem a ser mais flexíveis e extensíveis. Isto acontece porque, sempre que um novo nó entra na rede, a capacidade total do sistema aumenta. Este aumento de capacidade acontece porque a entrada de um

novo nó significa que a rede ganhou uma nova unidade de processamento. Essa característica dos sistemas P2P traz uma série de benefícios como robustez, balanceamento de carga, auto-organização, entre outros.

Segundo Flores (2005), uma rede P2P pode ser dividida em dois tipos: pura e híbrida. A rede é considerada pura quando sua arquitetura é totalmente distribuída, não necessitando de nenhum elemento central para fazer seu gerenciamento. Em contrapartida, a rede é considerada híbrida quando existe um nó responsável por fazer o gerenciamento dos outros nós. A Figura 3 demonstra um exemplo de uma rede híbrida. No exemplo, o nó X faz uma requisição ao nó de gerenciamento para buscar o recurso "Shrek.avi". O nó de gerenciamento então identifica que o nó Y possuí este recurso e estabelece uma conexão entre X e Y. A partir deste ponto, a rede funciona do mesmo modo que uma rede pura.

Figura 3 – Rede híbrida



Fonte: Silva (2010).

4.2 BLOCKCHAIN

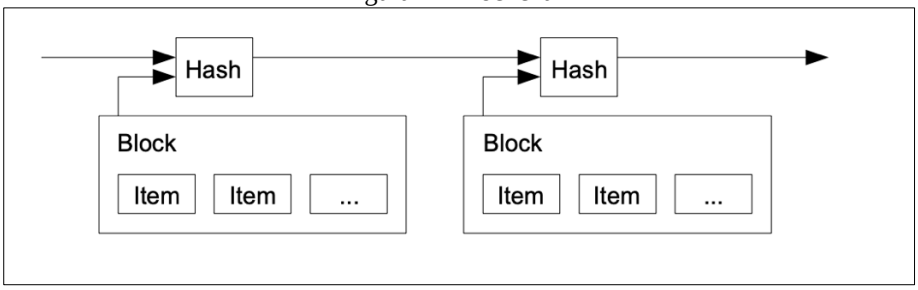
De acordo com Chervinski e Kreutz (2016), blockchain é uma base de dados distribuída e descentralizada. Com ela, é possível armazenar de forma segura dados sensíveis entre vários participantes sem a necessidade de uma autoridade centralizadora. Um dado, para entrar na blockchain, precisa passar pela aprovação da maioria dos participantes da rede. Uma vez persistido, o dado não pode mais ser modificado.

A primeira aparição da tecnologia blockchain ocorreu no trabalho de Nakamoto (2008). O autor propôs a implementação do Bitcoin, um sistema de pagamentos eletrônico totalmente descentralizado e baseado na arquitetura P2P. De acordo com Crosby *et al.* (2016), por mais que a blockchain seja uma tecnologia à parte, com diversas outras aplicações além do setor financeiro, seu funcionamento está fortemente ligado ao funcionamento do Bitcoin. Isso acontece pois, além da blockchain, o Bitcoin definiu uma série de tecnologias e técnicas que, quando utilizadas em conjunto, resultam em sistemas descentralizados confiáveis e robustos.

Segundo Nakamoto (2008), um dos maiores problemas de criar um sistema financeiro eletrônico descentralizado é garantir que o usuário não gastou a mesma moeda em duas compras diferentes. Isso pode acontecer pois, sem a existência de uma autoridade centralizadora, os participantes recebem as mensagens em ordens diferentes o que impossibilita eles saberem quanto um determinado usuário pode gastar. Foi para resolver este problema que Nakamoto (2008) desenvolveu a base do que conhecemos como blockchain. A ideia, conforme pode ser vista na Figura 4, foi implementar um modo de coletar todas as mensagens criadas em um determinado período de tempo e salvar elas em um bloco de dados. Esse bloco, para ser considerado autêntico, precisa ser aceito pela maioria dos participantes da rede.

Segundo Crosby *et al.* (2016), a abordagem acima ainda deixa um problema em aberto, pois qualquer usuário, mal-intencionado ou não, pode adicionar novos blocos na blockchain. Para resolver isto, Nakamoto (2008) sugeriu a utilização de uma tecnologia chamada *proof of work*. Nesta abordagem, um bloco só é aceito caso o criador dele apresente a resolução de um problema matemático específico. Este problema não é trivial e serve para provar que a criação do bloco exigiu uma certa quantia de poder computacional. Vários participantes da rede disputam entre si para resolver o problema. O primeiro a encontrar a resposta é o único que consegue criar o bloco.

Figura 4 - Blockchain



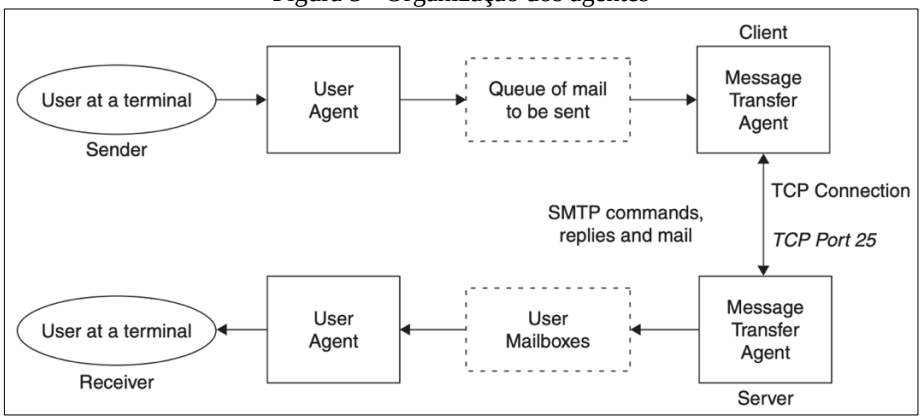
Fonte: Nakamoto (2008).

Conforme definido por Chervinski e Kreutz (2016), cada bloco de uma blockchain possui um identificador único que é gerado através da aplicação de uma função *hash* sobre o seu conteúdo. Alguns dos campos presentes no conteúdo de um bloco que impactam na geração do seu identificador são versão, resposta do *proof of work* e identificador do bloco anterior. O fato de se utilizar o identificador do bloco anterior para gerar o identificador do próximo bloco torna a blockchain uma base de dados imutável. Isso acontece, pois, para um usuário mal-intencionado alterar o conteúdo de um bloco, ele precisa gerar toda a sequência de blocos subsequentes ao bloco alterado. O único modo dele ter sucesso nessa operação é caso ele possua sozinho mais poder computacional que todo o restante da rede.

4.3 SMTP

São vários os componentes que são necessários em um sistema de correio eletrônico. Segundo Riabov (2005), os agentes básicos são o Mail Transfer Agent (MTA) e o User Agent (UA). O MTA é o agente que de fato transmite os e-mails. Um sistema de correio eletrônico precisa ter um agente MTA tanto no cliente para fazer o envio do e-mail quanto no servidor para fazer o recebimento. Servidores de e-mail diferentes também se comunicam utilizando os agentes MTA. Por outro lado, o UA é o programa de usuário responsável por se comunicar com o MTA e fazer as devidas ações como recuperar ou enviar e-mails. A organização dos agentes em um sistema de correio eletrônico pode ser vista na Figura 5.

Figura 5 - Organização dos agentes



Fonte: Riabov (2005).

O protocolo SMTP é o responsável por intermediar a conversa entre diferentes agentes MTA. De acordo com Riabov (2005), o SMTP define quais comandos podem ser enviados e define as possíveis respostas para esses comandos. O funcionamento desta conexão inicia com um MTA enviando uma requisição na porta 25 de outro MTA. A partir desse ponto, os dois agentes estão conectados e começam a transferir as mensagens definidas pelo SMTP. Uma lista com algumas possíveis mensagens pode ser vista no Quadro 3.

Quadro 3 - Comandos SMTP

Comando	Descrição	Exemplo
DATA	Comando utilizado para definir a mensagem	DATA Bom dia.
HELLO	Utilizado pelo cliente para se identificar	HELLO: furb.br
MAIL FROM	Utilizado pelo cliente para identificar o remetente	MAIL FROM: rrussi@furb.br
VRFY	Valida se o destinatário é valido	VRFY: schuartzrussi@gmail.com
RCPT	Utilizado pelo cliente para identificar o destinatário. Se existe mais de um destinatário, o comando é repetido	RCPT: schuartzrussi@gmail.com
RSET	Reinicia a conexão. Todos os dados enviados são perdidos	RSET
QUIT	Finaliza o envio	QUIT

Fonte: Riabov (2005).

Para finalizar o fluxo de envio, o e-mail deve chegar no aplicativo UA do destinatário. Conforme descrito por Riabov (2005), após o comando QUIT ser utilizado, a conexão entre os agentes MTA é desfeita e a mensagem é armazenada no servidor. Cabe ao aplicativo UA acessar o servidor de e-mail e recuperar as mensagens. A comunicação entre o cliente de e-mail e o servidor de e-mail é regulada por uma série de padrões conhecidos como protocolo de acesso a e-mail.

Segundo Riabov (2005), dois protocolos de acesso a e-mail muito famosos são o POP3 e o IMAP. O POP3 pode ser configurado para utilizar o protocolo SMTP para transferência das mensagens. Ele acessa o servidor de e-mail e baixa todas as mensagens para a máquina do usuário. Os e-mails por padrão, após serem enviados para o cliente, são apagados do servidor. Já o IMAP é um protocolo que trabalha com sincronização. Ele também utiliza o SMTP como protocolo de transmissão, porém, diferente do POP3, ele apenas sincroniza a caixa de e-mail local do usuário com a caixa de e-mail do servidor. As mensagens, após serem enviadas para a máquina do usuário ainda continuam no servidor e podem ser sincronizadas novamente.

REFERÊNCIAS

CHERVINSKI, João Otávio Massari; KREUTZ, Diego. **Introdução às tecnologias dos blockchains e das criptomoedas**. Alegrete: Universidade Federal do Pampa. 2016. Disponível em: http://seer.upf.br/index.php/rbca/article/download/9394/114114824/. Acesso em: 21 nov. 2020.

CROSBY, Michael et al. **BlockChain Technology: Beyond Bitcoin**. 2016. Disponível em: https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf. Acesso em: 21 nov. 2020.

ESTADÃO. **Gmail supera 1 bilhão de usuários ativos no mundo**. [S.l.], [2016]. Disponível em: https://link.estadao.com.br/noticias/cultura-digital,gmail-supera-1-bilhao-de-usuarios-ativos-no-mundo,10000028528. Acesso em: 01 out. 2020.

FLORES, Roberto Costa. **REDES PEER-TO-PEER: Um estudo sobre aspectos de segurança e mobilidade**. Rio de Janeiro: UFRJ. 2005. Disponível em: https://pantheon.ufrj.br/bitstream/11422/3112/1/RFlores.pdf. Acesso em: 21 nov. 2020.

GROTTENTHALER, Martin. **MailCoin — Blockchain Technology for Emails**. 2017. Tese de mestrado - University of Applied Sciences Upper Austria, Hagenberg.

KRAUSE, Nico. **Doichain: The Atomic "Double-Opt-In" and email spam protection system on the blockchain**. [2018]. Disponível em: https://raw.githubusercontent.com/Doichain/dapp/master/doc/Doichain-WhitePaper.pdf. Acesso em: 01 out. 2020.

NAKAMOTO, Satoshi. **Bitcoin: A Peer-to-Peer Electronic Cash System**. [2008]. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 02 out. 2020.

NITRONEWS. **10 estatísticas de email marketing que você precisa conhecer**. [S.l.], [2020]. Disponível em: https://www.nitronews.com.br/blog/10-estatisticas-de-email-marketing-que-voce-precisa-conhecer/. Acesso em: 30 set. 2020.

OLIVEIRA, Bruno Ribeiro de. **Conheça a história do e-mail e como evoluiu até hoje**. [2020]. Disponível em: https://www.linknacional.com.br/blog/historia-do-email/. Acesso em: 30 set. 2020.

PEZZOTTI, Renato. **Com 3,9 bilhões de usuários no mundo, o que acontece na web em um minuto?**. São Paulo, [2019]. Disponível em: https://economia.uol.com.br/noticias/redacao/2019/04/01/com-39-bilhoes-de-usuarios-no-mundo-o-que-acontece-na-web-em-um-minuto.htm. Acesso em: 30 set. 2020.

SILVA, Edemberg Rocha da. **Sistemas P2P e PDMSs**. Recife: Centro de Informática - UFPE. 2010. Disponível em: https://www.cin.ufpe.br/~speed/papers/TI-1-EdembergRocha.pdf. Acesso em: 21 nov. 2020.

RIABOV, Vladimir V. **SMTP (Simple Mail Transfer Protocol)**. Rivier College. 2005. Disponível em: https://www2.rivier.edu/faculty/vriabov/Information-Security-SMTP_c60_p01-23.pdf. Acesso em: 21 nov. 2020.

TRAININI, Paulo Ricardo Silveira; CARISSIMI, Alexandre da Silva. **Análise das Vulnerabilidades do Sistema de Correio Eletrônico**. Porto Alegre: Instituto de Informática, Universidade Federal do Rio Grande do Sul. 2005. Disponível em: https://www.researchgate.net/profile/Alexandre_Carissimi/publication/237499799_Analise_das_Vulnerabilidades_do_Sistema_de_Correio_Eletronico/links/564cd5cb08aeafc2aaaf8985/Analise-das-Vulnerabilidades-do-Sistema-de-Correio-Eletronico.pdf. Acesso em: 30 set. 2020.

WARREN, Jonathan. **Bitmessage: APeer-to-Peer Message Authentication and Delivery System**. [2012]. Disponível em: https://bitmessage.org/bitmessage.pdf. Acesso em: 01 out. 2020.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

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

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

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): Ruan Schuartz Russi _____

Avaliador(a): Francisco Adell Péricas_____

	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		
	1. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	2. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	3. 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		
	4. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	5. 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		
	6. 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		
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
ASPECTOS METODOLÓGICOS	7. 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		

PARECER – PROFESSOR AVALIADOR:
(PREENCHER APENAS NO PROJETO)

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

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

PARECER: (X) APROVADO ( ) REPROVADO

Assinatura: Francisco Adell Péricas_____ Data: 02/12/2020 _____

[1] 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.

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

PROTOCOLO PARA CORREIO ELETRÔNICO BASEADO EM BLOCKCHAIN

Ruan Schuartz Russi

Mauro Marcelo Mattos

1 INTRODUÇÃO

A popularidade do protocolo SMTP (Simple Mail Transfer Protocol) é inegável. Estima-se que, a cada minuto, 188 milhões de novos e-mails são enviados (PEZZOTTI, 2019). No ano de 2019, o número global de usuários de correio eletrônico totalizava 3,9 bilhões de usuários, sendo que o número deve crescer para 4,3 bilhões até 2023, número esse equiparado a metade da população mundial (NITRONEWS, 2020). Toda essa popularidade do correio eletrônico é atribuída a diversos fatores, sendo um deles a sua idade. O e-mail é considerado o serviço online mais antigo do mundo, tendo a primeira mensagem da história sido enviada no ano de 1971 (OLIVEIRA, 2020).

Apesar de ser uma tecnologia difundida e robusta, a idade da solução acaba trazendo para a mesma alguns pontos negativos. Na época em que os serviços de correio eletrônico foram desenhados, a internet como se apresenta hoje não existia e as redes de computadores existentes eram restritas a ambientes acadêmicos, não sendo liberado acesso para a população (TRAININI e CARISSIMI, 2005). Neste ambiente controlado, a preocupação com a segurança da informação era mínima e dispositivos de segurança geralmente não eram implementados. O cenário mudou com o advento da criação e popularização da internet. A rede se tornou um lugar hostil e perigoso, sendo utilizada por usuários mal-intencionados para a realização de práticas ilícitas. Por não ter sido criado com este ambiente hostil em mente, os serviços de correio eletrônico possuem uma diversidade de vulnerabilidades passíveis de serem exploradas como envio de spam, propagação de vírus, vazamento de mensagens e falsificação de identidade (TRAININI e CARISSIMI, 2005).

Uma característica importante dos servidores de e-mail é que eles formam uma rede descentralizada não existindo assim uma entidade com poder absoluto sobre as outras. Apesar disso, a implementação, devido a época em que foi feita, não aproveita características importantes de redes descentralizadas modernas. A blockchain por exemplo, tecnologia criada para dar vida ao Bitcoin, permite ter um histórico imutável de tudo que foi feito. A implementação do Bitcoin consegue garantir que um determinado usuário realizou uma transação financeira para outro de forma imutável. Já com o SMTP, é impossível determinar de modo fidedigno que um usuário enviou uma mensagem para outro. Além disso, atualmente é comum a utilização de serviços de e-mail de terceiros. Em 2016, o serviço de correio eletrônico da Google, o Gmail, chegou ~~aà~~ marca de um bilhão de usuários ativos mensalmente (ESTADÃO, 2016). Essa dependência com entidades terceiras acaba exigindo que os usuários tenham plena confiança nas mesmas pois nada impede que elas modifiquem, excluam ou exponham mensagem privadas. Essas entidades terceiras podem até mesmo mandar~~em~~ mensagens se passando pelo usuário.

~~Para resolver~~A fim de solucionar os problemas acima citados, o ideal seria a implementação de um protocolo totalmente novo, desenvolvido ~~que já nasça~~ sem a presença destas falhas. O problema é que, devido ~~aà~~ alta utilização dos serviços de e-mail, uma mudança abrupta de tecnologia se torna inviável, pois envolve uma mudança significativa na rotina de milhões de usuários. Diante deste cenário, o presente trabalho propõe o desenvolvimento de um protocolo baseado em blockchain que consiga enviar e receber mensagens de correio eletrônico de modo descentralizado e seguro. O protocolo será compatível com o protocolo SMTP permitindo assim uma migração gradual de tecnologia.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar um protocolo de envio e recebimento de e-mails baseado em blockchain e que seja compatível com o protocolo SMTP.

Os objetivos específicos são:

a) especificar um modelo de envio e recebimento de mensagens baseado no conceito de blockchain;
b) desenvolver um protótipo de cliente para demonstrar a funcionalidade do protocolo;
c) criar um conjunto de casos de testes para validar o protocolo desenvolvido;

2 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos que apresentam semelhança com os principais objetivos do trabalho proposto. O primeiro trabalho ~~é~~apresenta a implementação de uma blockchain capaz de armazenar e validar inscrições de usuários em *newsletters* (KRAUSE, 2018). O segundo é o estudo e implementação de um

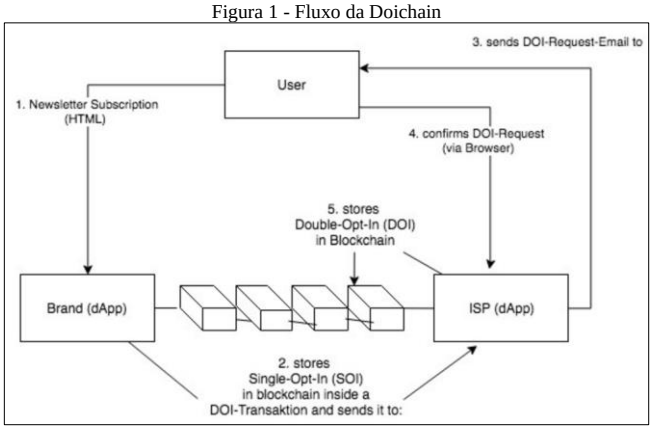
protocolo p2p para a troca segura de mensagens (WARREN, 2012). Por fim, na seção 2.3 será apresentado o trabalho de Grottenthaler (2017) que busca fazer a ligação entre envio de e-mails e transações de criptomoeda.

2.1 CONTROLE DE INSCRIÇÃO PARA RECEBIMENTE DE E-MAIL PROMOCIONAL

O trabalho desenvolvido por Krause (2018) consiste em um sistema baseado em blockchain que é capaz de provar que um determinado usuário se inscreveu em uma campanha de mail marketing. A motivação para o trabalho é resolver alguns casos comuns de spam e possibilitar a adesão de empresas de mail marketing relacionadas a leis ~~relacionadas a~~ de proteção de dados. Para tal, o autor propôs uma blockchain que armazena as inscrições dos usuários e pode ser consultada pelo dono da campanha de marketing antes de enviar algum e-mail para o usuário. Para se comunicar com a blockchain, o autor desenvolveu um aplicativo chamado dApp que serve unicamente para realizar esta comunicação. Tanto o dono da campanha de marketing quanto o dono do servidor de e-mail que o usuário está utilizando precisam configurar uma instância local do dApp.

O fluxo da plataforma proposta por Krause (2018) pode ser conferido na Figura 1. Primeiramente, o usuário (User) se inscreve em alguma campanha de e-mail. Um método comum de inscrição é a submissão do e-mail em um formulário presente em algum site. A empresa responsável pela campanha (Brand), ao receber a inscrição envia para a blockchain uma transação ainda não confirmada. O servidor de e-mail do usuário (Internet Service Provider - ISP), ao receber a transação não confirmada, envia ao usuário (User) um e-mail pedindo a confirmação da inscrição. Caso o usuário confirme inscrição, a transação é salva como confirmada na blockchain. Sendo assim, sempre que for enviar algum e-mail, o dono da campanha pode checar se o usuário realmente se inscreveu para aquela campanha.

Figura 1 - Fluxo da Doichain



Fonte: Krause (2018).

O trabalho de Krause (2018) consegue de fato ser uma solução para o problema que ele se propõe a resolver, porém, implantar tal solução em um cenário produtivo acaba exigindo bastante esforço. Por mais que a migração seja transparente para o usuário, a solução exige que pelo menos os servidores de e-mail mais populares do mercado se adequem a plataforma para que tenha algum efeito. Além disso, os donos das campanhas precisariam reconfigurar toda a sua infraestrutura para passar a se comunicar com a blockchain e validar as inscrições dos usuários. O ponto é, mesmo exigindo tamanho esforço, a plataforma se limita a resolver um cenário muito específico deixando ainda uma série de problemas sem solução.

Comentado [AS1]: Utilize linguagem formal.

2.2 PROTOCOLO P2P SEGURO PARA TROCA DE MENSAGENS

O protocolo desenvolvido por Warren (2012) visa resolver os problemas atuais que o protocolo SMTP apresenta como segurança das mensagens e envio de spam. A ideia é baseada no funcionamento do Bitcoin e traz alguns conceitos da arquitetura da criptomoeda para o mundo do correio eletrônico. Um desses conceitos, é um sistema de *Proof of Work* para envio de mensagens. Sempre que um determinado usuário decidir enviar uma

mensagem, ele precisa realizar algum processamento computacional que é configurado para durar em média quatro minutos. Desse modo, um usuário ao tentar fazer envio de spam estaria limitado a enviar apenas uma mensagem a cada quatro minutos. Outro ponto inspirado no Bitcoin é o endereçamento baseado em chaves públicas. No protocolo proposto, não existe o conceito de domínio de e-mail. O endereço utilizado para identificar um usuário é um valor gerado a partir da sua chave pública. É impossível que qualquer um que não seja o remente ou o destinatário da mensagem identifique os endereços envolvidos nela.

O fluxo da proposta de Warren (2012) se inicia com um usuário querendo enviar uma mensagem para algum outro usuário da rede. O remetente precisa, antes de tudo, da chave pública do destinatário. Em posse da chave pública, ele faz a criptografia da mensagem e envia para a rede. Todos os nós conectados na rede recebem a mensagem, porém, só o destinatário consegue ler a mensagem, pois é o único que possui a chave capaz de fazer a descriptografiafar ela. Ao receber a mensagem, o destinatário precisa enviar uma confirmação de recebimento. Caso o remetente não receba a confirmação em até dois dias ele faz a retransmissão após mais dois dias. O processo de retransmissão será feito até que o destinatário confirme o recebimento.

Pelo fato de a solução de Warren (2012) se propor a ser uma rede descentralizada, todos os nós da rede armazenam todas as mensagens. Para evitar a necessidade de uma alta capacidade de armazenamento, cada nó está configurado para apagar as mensagens não referentes a ele a cada dois dias. Mesmo assim, é possível que a capacidade de armazenamento para guardar todas as mensagens dos últimos dois dias seja muito grande. Devido a isso, o autor implementou uma lógica que, após um determinado limite de armazenamento ser atingido, os nós começam a se agrupar em clusters dividindo o armazenamento entre si.

A solução de Warren (2012) também apresenta uma proposta para campanhas de marketing. Primeiramente, o dono da campanha precisa publicar uma chave que consiga descriptografar as mensagens enviadas por ele. Os usuários que quiserem receber as mensagens precisam pegar a chave e configurar o seu nó para aceitar mensagens para ela. Deste modo, sempre que um nó receber uma mensagem, ele vai tentar primeiramente abrir ela com a chave privada do usuário e, não tendo sucesso, vai tentar com todas as chaves presentes na lista de inscrições do nó.

O trabalho de Warren (2012) consegue resolver a grande maioria dos problemas atuais envolvendo correio eletrônico. A solução é fácil de usar e configurar e já possuí criptografia por padrão eliminando a necessidade da utilização de tecnologias terceiras. Apesar dos seus vários pontos positivos, a solução tem o ponto fraco de não ser compatível com o SMTP. Desse modo, seria necessária uma mudança abrupta de tecnologia, o que acaba se tornando inviável devido aà grande utilização dos serviços de e-mail. Outro ponto fraco é a impossibilidade de identificar as partes relacionadas no envio de uma mensagem. Por mais que tal funcionalidade seja boa em alguns casos, ela acaba dificultando a adoção da tecnologia no meio corporativo.

2.3 LIGAÇÃO DE E-MAILS COM TRANSAÇÕES DE CRIPTOMOEDAS

O trabalho desenvolvido por Grottenthaler (2017) busca resolver alguns problemas encontrados na implementação de correio eletrônico criando um relacionamento de cada e-mail enviado com uma transação de criptomoeda. As transações são salvas e validadas por uma blockchain. O estudo realizado foca na solução de dois problemas: envio de spam e confirmação do recebimento de mensagens.

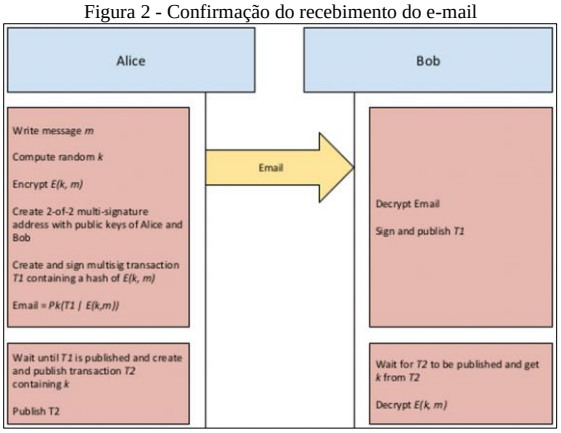
O fluxo da plataforma proposta por Grottenthaler (2017) inicia com um usuário querendo enviar um e-mail para outro usuário. O remetente, para realizar o envio do e-mail, precisa adicionar dois cabeçalhos na requisição: Mailcoin-Txid e Mailcoin-Header. O Mailcoin-Txid é o id da transação que será salva na blockchain e serve para o destinatário identificar a transação e validar se é autêntica. O Mailcoin-Header contém informações referentes ao envio, sendo elas a versão do sistema, a data atual, o e-mail do destinatário e um valor gerado de modo aleatório que serve unicamente para evitar conflitos quando um *hash* precisar ser gerado utilizando tais dados. Após a geração dos valores, o remetente cria uma transação na blockchain contendo como id o Mailcoin-Txid e como valor um hash gerado a partir do Mailcoin-Header e então faz o envio do e-mail. O destinatário, ao receber o e-mail, recupera o Mailcoin-Txid do cabeçalho e busca pela transação na blockchain. Se ele confirmar que a transação existe, ele recupera o valor do hash armazenado na transação, gera o hash utilizando o valor que vier no cabeçalho Mailcoin-Header e então compara os valores. Se forem iguais, a mensagem é autêntica.

Para o tratamento dos casos de spam, a solução criada por Grottenthaler (2017) exige que qualquer envio de e-mail gere algum custo financeiro para o remetente. Esse custo é descontado das criptomoedas que o usuário possui na blockchain. O valor cobrado é muito baixo, porém, levando isso para um cenário de milhares de envios como acontece nos casos de spam, o valor final ficaria muito alto e dificultaria a prática de tal atividade. Um problema aparente é que qualquer pessoa teria custo para enviar e-mails, mesmo que ela não esteja enviando spam. Para resolver este problema, a plataforma encoraja os usuários a devolverem o custo da transação para o remetente.

Caso o remetente não receba o valor de volta, ele pode decidir não enviar mais mensagens para o usuário que não fez a devolução.

A arquitetura do fluxo desenvolvido para identificar se um usuário de fato recebeu um e-mail é apresentada na Figura 2. Grottenthaler (2017) propôs um sistema baseado em dupla assinatura, na qual, para um usuário conseguir ler um e-mail, ele obrigatoriamente precisa notificar o remetente que recebeu a mensagem. Por exemplo, a usuária Alice, ao enviar um e-mail para o usuário Bob, criptografa a mensagem com uma chave aleatória `k` e cria uma transação no qual assina com a sua chave privada. No momento que Bob recebe a transação, ele identifica que é uma transação de dupla assinatura e que então, para ler o conteúdo do e-mail, precisa da chave `k`. Sendo assim, ele assina a transação recebida e a envia para a blockchain. Quando Alice recebe a confirmação de que a transação foi assinada, ela cria uma nova transação contendo a chave `k`. Bob então com o valor da chave `k` consegue acessar o conteúdo do e-mail. Com tal sistema é possível identificar se o usuário recebeu o e-mail pois basta verificar se ele realizou o processo de assinatura da transação.

Figura 2 - Confirmação do recebimento do e-mail



Fonte: Grottenthaler (2017).

O trabalho de Grottenthaler (2017) consegue criar uma extensão para o protocolo STMP trazendo para o mesmo várias vantagens oriundas da utilização de blockchain, porém, por se tratar de uma extensão, a base de tudo ainda continua sendo o SMTP. Desse modo, a solução não consegue resolver problemas que são inerentes do protocolo como, por exemplo, a garantia de imutabilidade das mensagens e a garantia que um determinado usuário enviou de fato uma mensagem com um determinado conteúdo.

3 PROPOSTA DO PROTOCOLO

Nesta seção serão apresentadas as justificativas para o desenvolvimento do estudo proposto, bem como um quadro comparativo com os trabalhos correlatos, os requisitos funcionais e não funcionais e a metodologia utilizada no desenvolvimento da solução.

3.1 JUSTIFICATIVA

Devido a sua grande massa de usuários, os serviços de e-mail passaram a ser um alvo muito atraente para criminosos. Eles buscam explorar as mais diversas vulnerabilidades encontradas neste tipo de serviço. Além disso, existe uma tendência mundial para a concentração de usuários em servidores SMTP de terceiros como Gmail e Outlook. Isso acaba exigindo que os usuários confiem suas informações pessoais e mensagens a empresas privadas que podem realizar qualquer tipo de processo sobre elas. Diante desse cenário, é natural que existam trabalhos que visam resolver tais problemas. O Quadro 1 faz uma análise das características dos trabalhos correlatos encontrados.

Quadro 1 – Comparativo entre os trabalhos correlatos

Características	Doichain	MailCoin	Bitmessage
Compatível com SMTP	Sim	Sim	Não
Sistema de Proof of Work	Não	Sim	Sim
Controle de inscrição em newsletter	Sim	Não	Não
Mensagens imutáveis	Não	Não	Sim
Assinatura de mensagens	Não	Sim	Sim
Controle para spam	Não	Sim	Sim
Identificação dos usuários	Sim	Sim	Não

Fonte: elaborado pelo autor.

Conforme análise do Quadro 1, é possível identificar que os trabalhos buscam resolver diferentes problemas. O Doichain, por estar unicamente centrado no processo de identificar se um usuário se inscreveu em um *newsletter*, acaba não abordando nenhum outro cenário. O MailCoin consegue abordar um universo maior de possibilidades, porém acaba deixando de fora um ponto importante que é a garantia de imutabilidade e histórico das mensagens. Isso acontece pois ele mantém como base o protocolo SMTP. Já o Bitmessage, por se tratar de um protocolo novo consegue ter solução para a grande maioria dos problemas citados, porém, ele tem o ponto fraco de exigir uma completa mudança de tecnologia, o que dificulta a migração para o protocolo. Além disso, o Bitmessage não permite que um terceiro usuário identifique os usuários envolvidos em uma mensagem. Isso se torna um impeditivo em ambientes corporativos.

Neste contexto, o presente trabalho se torna relevante pois tem potencial ~~para~~ por impactar diretamente nos bilhões de usuários de sistemas de correio eletrônico. Espera-se que esses usuários tenham disponível uma solução mais segura e robusta, que não apresente os problemas que os atuais sistemas de e-mail possuem. Além disso, o trabalho também deverá contribuir em pesquisas relacionadas a modernização dos serviços de e-mail.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos funcionais são:

a) permitir enviar um e-mail de um servidor que esteja utilizando o novo protocolo para um servidor SMTP que não esteja utilizando o novo protocolo;
b) permitir enviar um e-mail de um servidor SMTP para um servidor que esteja utilizando o novo protocolo;
c) permitir o envio de mensagens entre dois nós da rede utilizando o novo protocolo;
d) permitir identificação dos usuários através de um endereço;
e) permitir padrões diferentes de endereços para identificação de domínios pertencentes a organizações;
f) disponibilizar uma API que possibilite a comunicação de softwares clientes com a blockchain.~~;~~

Os requisitos não funcionais são:

a) implementar os nós da rede utilizando a linguagem Golang;
b) utilizar a arquitetura REST para a API para comunicação com a blockchain;
c) receber mensagens SMTP na porta 587;
d) armazenar as mensagens em uma blockchain compartilhada entre todos os nós da rede;
e) obrigar a realização de um processo de Proof of Work para envio de mensagens;
f) criptografar as mensagens armazenadas na blockchain.~~;~~

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

a) levantamento bibliográfico: realizar levantamento bibliográfico com relação a implementação de soluções utilizando blockchain. Pesquisar também a bibliografia referente a implementação de servidores SMTP;

b) elicitação de requisitos: nesta etapa será feito o refinamento dos requisitos tomando como base a pesquisa realizada;

c) especificação da arquitetura da blockchain: nesta etapa será definido como a blockchain deverá funcionar. Será especificada a estrutura de armazenamento, criptografia e envio dos e-mails utilizando diagramas da UML;

d) desenvolvimento da blockchain: será realizada a implementação da blockchain utilizando a linguagem Golang. Será desenvolvida com base na arquitetura especificada no passo c;

e) desenvolvimento da API: nesta etapa será desenvolvida a API responsável por fazer a interface com a blockchain;

f) desenvolvimento do servidor SMTP: implementar a compatibilidade dos nós com o protocolo SMTP;

g) testes: realizar o teste da rede validando o envio de mensagens e a compatibilidade com o protocolo SMTP.;

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

	2021									
	fev.		mar.		abr.		mai.		jun.	
etapas / quinzenas	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação da arquitetura da blockchain										
desenvolvimento da blockchain										
desenvolvimento da API										
desenvolvimento do servidor SMTP										
testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Este capítulo tem como objetivo fazer um estudo inicial sobre os principais temas envolvidos no desenvolvimento do projeto. A seção 4.1 faz uma análise referente a implementação de sistemas descentralizados. A seção 4.2 aborda os fundamentos básicos referentes a blockchain. Por último, a seção 4.3 faz uma análise do funcionamento do protocolo SMTP.

Comentado [AS2]: Coloca no nome da sigla

4.1 SISTEMAS DESCENTRALIZADOS

Sistemas descentralizados, diferente dos sistemas cliente/servidor, não possuem nenhuma autoridade central. Cada integrante da rede pode atuar tanto como cliente quanto como servidor. Esses sistemas podem ser implementados utilizando várias abordagens diferentes, porém, a mais famosa é a utilização da arquitetura P2P (pares em pares). De acordo com Silva (2010), os sistemas P2P são utilizados para diversos fins como computação distribuída, troca de mensagens, trabalho colaborativo e compartilhamento de dados.

Segundo Silva (2010), uma rede, para ser considerada P2P, deve possuir as seguintes características:

a) cada nó se conecta diretamente com os outros nós;

b) cada nó é responsável pelos seus dados;

c) um nó pode entrar e sair da rede a qualquer momento;

d) um nó pode atuar tanto como cliente quanto como servidor;

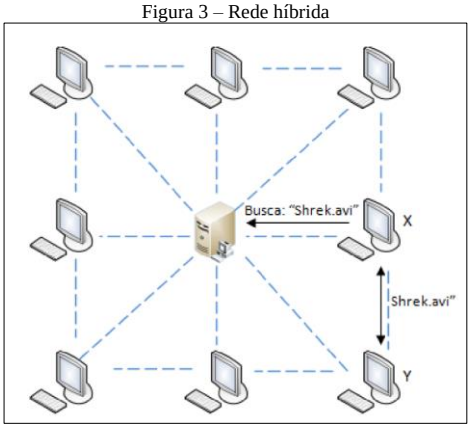
e) não existe autoridade central.;

Conforme definido por Silva (2010), diferente dos sistemas cliente/servidor em que o número de servidores é fixo, sistemas P2P tendem a ser mais flexíveis e extensíveis. Isto acontece porque, sempre que um novo nó entra na rede, a capacidade total do sistema aumenta. Este aumento de capacidade acontece porque a entrada de um

novo nó significa que a rede ganhou uma nova unidade de processamento. Essa característica dos sistemas P2P traz uma série de benefícios como robustez, balanceamento de carga, auto-organização, entre outros.

Segundo Flores (2005), uma rede P2P pode ser dividida em dois tipos: pura e híbrida. A rede é considerada pura quando sua arquitetura é totalmente distribuída, não necessitando de nenhum elemento central para fazer o gerenciamento da mesma. Em contrapartida, a rede é considerada híbrida quando existe um nó responsável por fazer o gerenciamento dos outros nós. A Figura 3 demonstra um exemplo de uma rede híbrida. No exemplo, o nó X faz uma requisição ao nó de gerenciamento para buscar o recurso "Shrek.avi". O nó de gerenciamento então identifica que o nó Y possuí este recurso, e estabelece uma conexão entre X e Y. A partir deste ponto, a rede funciona do mesmo modo que uma rede pura.

Figura 3 – Rede híbrida



Fonte: Silva (2010).

4.2 BLOCKCHAIN

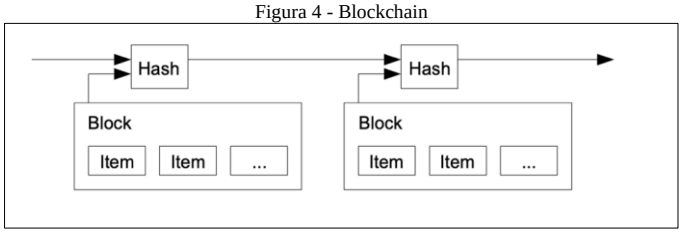
De acordo com Chervinski e Kreutz (2016), blockchain é uma base de dados distribuída e descentralizada. Com ela, é possível armazenar de forma segura dados sensíveis entre vários participantes sem a necessidade de uma autoridade centralizadora. Um dado, para entrar na blockchain, precisa passar pela aprovação da maioria dos participantes da rede. Uma vez persistido, o dado não pode mais ser modificado.

A primeira aparição da tecnologia blockchain ocorreu no trabalho de Nakamoto (2008). O autor propôs a implementação do Bitcoin, um sistema de pagamentos eletrônico totalmente descentralizado e baseado na arquitetura P2P. De acordo com Crosby *et al.* (2016), por mais que a blockchain seja uma tecnologia à parte, com diversas outras aplicações além do setor financeiro, seu funcionamento está fortemente ligado ao funcionamento do Bitcoin. Isso acontece pois, além da blockchain, o Bitcoin definiu uma série de tecnologias e técnicas que, quando utilizadas em conjunto, resultam em sistemas descentralizados confiáveis e robustos.

Segundo Nakamoto (2008), um dos maiores problemas de criar um sistema financeiro eletrônico descentralizado é garantir que o usuário não gastou a mesma moeda em duas compras diferentes. Isso pode acontecer pois, sem a existência de uma autoridade centralizadora, os participantes recebem as mensagens em ordens diferentes o que impossibilita eles saberem quanto um determinado usuário pode gastar. Foi para resolver este problema que Nakamoto (2008) desenvolveu a base do que conhecemos como blockchain. A ideia, conforme pode ser vista na Figura 4, foi implementar um modo de coletar todas as mensagens criadas em um determinado período de tempo e salvar elas em um bloco de dados. Esse bloco, para ser considerado autêntico, precisa ser aceito pela maioria dos participantes da rede.

Segundo Crosby *et al.* (2016), a abordagem acima ainda deixa um problema em aberto, pois qualquer usuário, mal-intencionado ou não, pode adicionar novos blocos na blockchain. Para resolver isto, Nakamoto (2008) sugeriu a utilização de uma tecnologia chamada *proof of work*. Nesta abordagem, um bloco só é aceito caso o criador dele apresente a resolução de um problema matemático específico. Este problema não é trivial e serve para provar que a criação do bloco exigiu uma certa quantia de poder computacional. Vários participantes da rede disputam entre si para resolver o problema. O primeiro a encontrar a resposta é o único que consegue criar o bloco.

Figura 4 - Blockchain



Fonte: Nakamoto (2008).

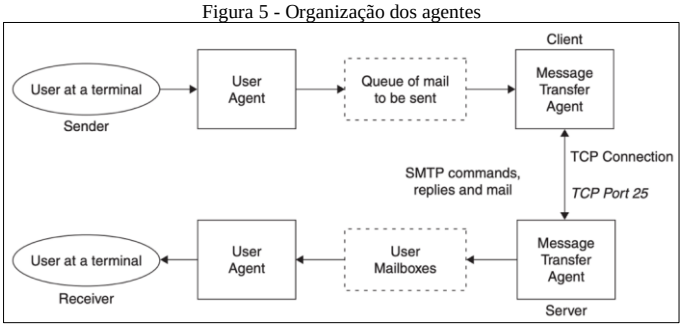
Conforme definido por Chervinski e Kreutz (2016), cada bloco de uma blockchain possuí um identificador único que é gerado através da aplicação de uma função *hash* sobre o seu conteúdo. Alguns dos campos presentes no conteúdo de um bloco que impactam na geração do seu identificador são versão, resposta do *proof of work* e identificador do bloco anterior. O fato de se utilizar o identificador do bloco anterior para gerar o identificador do próximo bloco torna a blockchain uma base de dados imutável. Isso acontece, pois, para um usuário mal-intencionado alterar o conteúdo de um bloco, ele precisa gerar toda a sequência de blocos subsequentes ao bloco alterado. O único modo dele ter sucesso nessa operação é caso ele possua sozinho mais poder computacional que todo o restante da rede.

4.3 SMTP

Comentado [AS3]: Coloca no nome da sigla.

São vários os componentes que são necessários em um sistema de correio eletrônico. Segundo Riabov (2005), os agentes básicos são o Mail Transfer Agent (MTA) e o User Agent (UA). O MTA é o agente que de fato transmite os e-mails. Um sistema de correio eletrônico precisa ter um agente MTA tanto no cliente para fazer o envio do e-mail quanto no servidor para fazer o recebimento. Servidores de e-mail diferentes também se comunicam utilizando os agentes MTA. Por outro lado, o UA é o programa de usuário responsável por se comunicar com o MTA e fazer as devidas ações como recuperar ou enviar e-mails. A organização dos agentes em um sistema de correio eletrônico pode ser vista na Figura 5.

Figura 5 - Organização dos agentes



Fonte: Riabov (2005).

O protocolo SMTP é o responsável por intermediar a conversa entre diferentes agentes MTA. De acordo com Riabov (2005), o SMTP define quais comandos podem ser enviados ~~e também define~~bem como as possíveis respostas para esses comandos. O funcionamento desta conexão inicia com um MTA enviando uma requisição na porta 25 de outro MTA. A partir desse ponto, os dois agentes estão conectados e começam a transferir as mensagens definidas pelo SMTP. Uma lista com algumas possíveis mensagens pode ser vista no Quadro 3.

Quadro 3 - Comandos SMTP

Comando	Descrição	Exemplo
DATA	Comando utilizado para definir a mensagem	DATA Bom dia.
HELLO	Utilizado pelo cliente para se identificar	HELLO: furb.br
MAIL FROM	Utilizado pelo cliente para identificar o remetente	MAIL FROM: rrussi@furb.br
VRFY	Valida se o destinatário é valido	VRFY: schuartzrussi@gmail.com
RCPT	Utilizado pelo cliente para identificar o destinatário. Se existe mais de um destinatário, o comando é repetido	RCPT: schuartzrussi@gmail.com
RSET	Reinicia a conexão. Todos os dados enviados são perdidos	RSET
QUIT	Finaliza o envio	QUIT

Fonte: Riabov (2005).

Para finalizar o fluxo de envio, o e-mail deve chegar no aplicativo UA do destinatário. Conforme descrito por Riabov (2005), após o comando QUIT ser utilizado, a conexão entre os agentes MTA é desfeita e a mensagem é armazenada no servidor. Cabe ao aplicativo UA acessar o servidor de e-mail e recuperar as mensagens. A comunicação entre o cliente de e-mail e o servidor de e-mail é regulada por uma série de padrões conhecidos como protocolo de acesso a e-mail.

Segundo Riabov (2005), dois protocolos de acesso a e-mail muito famosos são o POP3 e o IMAP. O POP3 pode ser configurado para utilizar o protocolo SMTP para transferência das mensagens. Ele acessa o servidor de e-mail e baixa todas as mensagens para a máquina do usuário. Os e-mails por padrão, após serem enviados para o cliente, são apagados do servidor. Já o IMAP é um protocolo que trabalha com sincronização. Ele também utiliza o SMTP como protocolo de transmissão, porém, diferente do POP3, ele apenas sincroniza a caixa de e-mail local do usuário com a caixa de e-mail do servidor. As mensagens, após serem enviadas para a máquina do usuário ainda continuam no servidor e podem ser sincronizadas novamente.

REFERÊNCIAS

CHERVINSKI, João Otávio O. MassariM.; KREUTZ, Diego. **Introdução às tecnologias dos blockchains e das criptomoedas**. Alegrete: Universidade Federal do Pampa. 2016. Disponível em: http://seer.upf.br/index.php/rbca/article/download/9394/114114824/. Acesso em: 21 nov. 2020.

CROSBY, Michael et al. **BlockChain Technology: Beyond Bitcoin**. 2016. Disponível em: https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf. Acesso em: 21 nov. 2020.

ESTADÃO. **Gmail supera 1 bilhão de usuários ativos no mundo**. [S.l.], [2016]. Disponível em: https://link.estadao.com.br/noticias/cultura-digital,gmail-supera-1-bilhao-de-usuarios-ativos-no-mundo,10000028528. Acesso em: 01 out. 2020.

FLORES, Roberto Costa. **REDES PEER-TO-PEER: Um estudo sobre aspectos de segurança e mobilidade**. Rio de Janeiro: UFRJ. 2005. Disponível em: https://pantheon.ufrj.br/bitstream/11422/3112/1/RFlores.pdf. Acesso em: 21 nov. 2020.

GROTTENTHALER, Martin. **MailCoin — Blockchain Technology for Emails**. 2017. Tese de mestrado - University of Applied Sciences Upper Austria, Hagenberg.

KRAUSE, Nico. **Doichain: The Atomic "Double-Opt-In" and email spam protection system on the blockchain**. [2018]. Disponível em: https://raw.githubusercontent.com/Doichain/dapp/master/doc/Doichain-WhitePaper.pdf. Acesso em: 01 out. 2020.

NAKAMOTO, Satoshi. **Bitcoin: A Peer-to-Peer Electronic Cash System**. [2008]. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 02 out. 2020.

NITRONEWS. **10 estatísticas de email marketing que você precisa conhecer**. [S.l.], [2020]. Disponível em: https://www.nitronews.com.br/blog/10-estatisticas-de-email-marketing-que-voce-precisa-conhecer/. Acesso em: 30 set. 2020.

OLIVEIRA, Bruno Ribeiro de. **Conheça a história do e-mail e como evoluiu até hoje**. [2020]. Disponível em: https://www.linknacional.com.br/blog/historia-do-email/. Acesso em: 30 set. 2020.

PEZZOTTI, Renato. **Com 3,9 bilhões de usuários no mundo, o que acontece na web em um minuto?**. São Paulo, [2019]. Disponível em: https://economia.uol.com.br/noticias/redacao/2019/04/01/com-39-bilhoes-de-usuarios-no-mundo-o-que-acontece-na-web-em-um-minuto.htm. Acesso em: 30 set. 2020.

SILVA, Edemberg Rocha da. **Sistemas P2P e PDMSs**. Recife: Centro de Informática - UFPE. 2010. Disponível em: https://www.cin.ufpe.br/~speed/papers/TI-1-EdembergRocha.pdf. Acesso em: 21 nov. 2020.

RIABOV, Vladimir V. **SMTP (Simple Mail Transfer Protocol)**. Rivier College. 2005. Disponível em: https://www2.rivier.edu/faculty/vriabov/Information-Security-SMTP_c60_p01-23.pdf. Acesso em: 21 nov. 2020.

TRAININI, Paulo ~~Ricardo~~ R. ~~Silveira~~S.; CARISSIMI, Alexandre da S~~ilva~~. **Análise das Vulnerabilidades do Sistema de Correio Eletrônico**. Porto Alegre: Instituto de Informática, Universidade Federal do Rio Grande do Sul. 2005. Disponível em: https://www.researchgate.net/profile/Alexandre_Carissimi/publication/237499799_Analise_das_Vulnerabilidades_do_Sistema_de_Correio_Eletronico/links/564cd5cb08aeafc2aaaf8985/Analise-das-Vulnerabilidades-do-Sistema-de-Correio-Eletronico.pdf. Acesso em: 30 set. 2020.

WARREN, Jonathan. **Bitmessage: APeer-to-Peer Message Authentication and Delivery System**. [2012]. Disponível em: https://bitmessage.org/bitmessage.pdf. Acesso em: 01 out. 2020.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

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

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

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR TCC I

Acadêmico(a): Ruan Schuartz Russi _____

Avaliador(a): Andreza Sartori_____

	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		
	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		
ASPECTOS METODOLÓGICOS	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		

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

O projeto de TCC será reprovado se:
- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: (x) APROVADO ( ) REPROVADO

Assinatura: _____ Data: 07/12/2020 _____

[1] 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.