



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Serviços do SWIM Registry Brazil: uma nova abordagem com Blockchain

Igor da Silva Bonomo

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador

Prof. Dr. Li Weigang

Coorientador

Prof. Dr. Vinícius R. P. Borges

Brasília
2020

Resumo

O *System Wide Information Management (SWIM)* abarca definições, infraestrutura e governança, habilitando o controle da informação de tráfego aéreo (*ATM*). A partir desse conceito, surge também a definição de *SWIM Registry*, que visa aprimorar a visibilidade das informações relacionadas ao *ATM* e serviços disponíveis do *SWIM*. O *Departamento de Controle do Espaço Aéreo (DECEA)* e o *Laboratório de Modelo Computacional para Transporte Aéreo (TransLab)* desenvolveram o *SWIM Registry Brazil*, que consiste em um repositório contendo os serviços disponíveis no *SWIM* para o âmbito do Brasil. Dentre os serviços disponíveis, um é o de Plano de Voo, entretanto esse serviço apresenta algumas limitações referentes à segurança das informações dos planos de voo e autenticidade dos usuários do sistema. Dessa forma, este trabalho tem por objetivo propor uma nova abordagem para a criação dos serviços de plano de voo. Sendo assim, a metodologia proposta consiste na criação de uma aplicação *blockchain* para executar as operações relacionadas ao serviço de plano de voo. Essa aplicação é baseada em uma adaptação da estrutura básica do *blockchain* para receber as informações referentes a um plano de voo, para em seguida simular em diferentes cenários de teste a implementação do serviço. A partir dos diversos testes foi possível observar que o serviço de plano de voo pode ser executado como uma aplicação *blockchain*, obtendo uma maior segurança as informações dos dados de plano de voo e a autenticidade dos usuários do sistema.

Palavras-chave: *Blockchain, SWIM, SWIM Registry Brazil*

Abstract

System Wide Information Management (SWIM) consists of definitions, infrastructure and governance, enabling the Air Traffic Management (ATM) information. From this concept, comes the definition of SWIM Registry, which aims to improve the visibility of information related to ATM and services available from SWIM. The *Departamento de Controle do Espaço Aéreo (DECEA)* and the *Laboratório de Modelo Computacional para Transporte Aéreo (TransLab)* developed SWIM Registry Brazil, which consists of a repository containing the services available in SWIM for the Brazilian context. Among the services available, one is the flight plan, however this service has some limitations regarding the security of the flight plan information and the authenticity of the users. Thus, this work aims to propose a new approach to the creation of flight plan services. Therefore, the proposed methodology consists of creating a blockchain application to perform operations related to flight plan service. This application is based on an adaptation of the basic blockchain structure to receive information regarding a flight plan, and then simulate the implementation of the service in different test scenarios. From the various tests it was possible to observe that the flight plan service can be executed as a blockchain application obtaining greater security of the information of flight plan data and the authenticity of the system users.

Keywords: Blockchain, SWIM, SWIM Registry Brazil

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Objetivos	3
1.2.1	Objetivos Específicos	4
1.3	Metodologia	4
1.4	Organização do Documento	5
2	Fundamentos	6
2.1	Sistemas de Aviação	6
2.1.1	SWIM	6
2.1.2	SWIM Registry	8
2.1.3	SWIM Registry Brazil	9
2.1.4	Demonstração do conceito SWIM	12
2.1.5	Plano de Voo	12
2.2	Blockchain	14
2.2.1	Introdução	14
2.2.2	Estrutura Básica	14
2.2.3	Função Hash	15
2.2.4	Transações	16
2.2.5	Tolerância a Falha Bizantina	17
2.2.6	Protocolos de Consenso	18
2.2.7	Smart Contracts	21
2.2.8	Considerações	22
3	Trabalhos Relacionados	23
3.1	Aplicações Blockchain	23
3.2	Blockchain no Sistema Aéreo	25
3.3	Considerações	26

4 Metodologia Proposta	27
4.1 Arquitetura Proposta	27
4.1.1 Justificativa	29
4.2 Fonte de dados	30
4.2.1 Plano de Voo	30
4.2.2 Modelagem dos Dados	31
4.3 Implementação	32
4.3.1 <i>Hyperledger Composer</i>	32
4.3.2 Arquitetura Hyperledger Composer	34
4.3.3 <i>Hyperledger Composer no SWIM</i>	36
4.3.4 Hyperledger Caliper	40
5 Simulação e Resultados	43
5.1 Indicadores de Performance	43
5.2 Cenários de Teste	44
5.2.1 Benchmark - Parâmetros Gerais	45
5.2.2 Network – Seleção de Parâmetros	46
5.2.3 Cenário de Teste 1	47
5.2.4 Cenário de Teste 2	53
5.2.5 Cenário de Teste 3	59
5.2.6 Cenário de Teste 4	65
5.3 Discussão e Análise dos Resultados	71
5.3.1 Criação de um plano de voo	71
5.3.2 Consulta a um plano de voo	74
6 Conclusão e Trabalhos Futuros	77
6.1 Contribuições	77
6.2 Trabalhos Futuros	78
6.3 Considerações Finais	79
Referências	80

Lista de Figuras

2.1	Escopo do SWIM adaptado do ICAO. International Civil Aviation Organization, “Manual on System Wide Information Management (SWIM) Concept, (Doc 10039)”, Montreal, 2015.	7
2.2	Site do <i>SWIM Registry Brazil</i>	10
2.3	Demonstração do conceito SWIM Registry.	12
2.4	Exemplo de uma cadeia de blocos.	15
2.5	Exemplo da função <i>hash</i>	16
2.6	Exemplo adaptado do Problema dos Generais Bizantinos.	17
2.7	<i>Smart Contracts</i>	22
4.1	Primeira etapa da arquitetura proposta.	28
4.2	Arquitetura proposta para implementação do serviço de plano de voo.	28
4.3	Questionário adaptado de Wust [1].	29
4.4	Consulta de planos de voo do CGNA.	31
4.5	Alterações propostas no campo <i>data</i>	32
4.6	Arquitetura do Hyperledger Composer.	34
4.7	Ativo de Plano de Voo.	36
4.8	Fluxograma dos Estados de Plano de Voo.	37
4.9	Participantes - Stakeholder.	37
4.10	Transação - Validar Plano.	38
4.11	Arquitetura do Hyperledger Caliper adaptada.	42
5.1	Criação de Plano de Voo com 5 Clientes.	50
5.2	Criação de Plano de Voo com 20 Clientes.	50
5.3	Criação de Plano de Voo com 50 Clientes.	51
5.4	Consultar um plano de voo com 5 Clientes	52
5.5	Consultar um plano de voo com 20 Clientes	52
5.6	Consultar um plano de voo com 50 Clientes	53
5.7	Criação de Plano de Voo com 5 Clientes	56
5.8	Criação de Plano de Voo com 20 Clientes	57

5.9	Criação de Plano de Voo com 50 Clientes	57
5.10	Consultar um plano de voo com 5 Clientes	58
5.11	Consultar um plano de voo com 20 Clientes	58
5.12	Consultar um plano de voo com 50 Clientes	59
5.13	Criação de Plano de Voo com 5 Clientes	62
5.14	Criação de Plano de Voo com 20 Clientes	62
5.15	Criação de Plano de Voo com 50 Clientes	63
5.16	Consultar um plano de voo com 5 Clientes	64
5.17	Consultar um plano de voo com 20 Clientes	64
5.18	Consultar um plano de voo com 50 Clientes	65
5.19	Criação de Plano de Voo com 5 Clientes	68
5.20	Criação de Plano de Voo com 20 Clientes	68
5.21	Criação de Plano de Voo com 50 Clientes	69
5.22	Consultar um plano de voo com 5 Clientes	70
5.23	Consultar um plano de voo com 20 Clientes	70
5.24	Consultar um plano de voo com 50 Clientes	71
5.25	Criação de Plano de Voo com 2-1	72
5.26	Criação de Plano de Voo com 3-1	72
5.27	Criação de Plano de Voo com 3-2	73
5.28	Criação de Plano de Voo com 3-3	73
5.29	Consulta de Plano de Voo com 2-1	74
5.30	Consulta de Plano de Voo com 3-1	75
5.31	Consulta de Plano de Voo com 3-2	75
5.32	Consulta de Plano de Voo com 3-3	76

Lista de Tabelas

5.1	Cenário de Teste 1 - 5 Clientes	47
5.2	Cenário de Teste 1 - 20 Clientes	47
5.3	Cenário de Teste 1 - 50 Clientes	48
5.4	Cenário de Teste 1 - Query 5 Clientes	48
5.5	Cenário de Teste 1 - Query 20 Clientes	49
5.6	Cenário de Teste 1 - Query 50 Clientes	49
5.7	Cenário de Teste 2 - 5 Clientes	53
5.8	Cenário de Teste 2 - 20 Clientes	54
5.9	Cenário de Teste 2 - 50 Clientes	54
5.10	Cenário de Teste 2 - Query 5 Clientes	55
5.11	Cenário de Teste 2 - Query 20 Clientes	55
5.12	Cenário de Teste 2 - Query 50 Clientes	55
5.13	Cenário de Teste 3 - 5 Clientes	59
5.14	Cenário de Teste 3 - 20 Clientes	60
5.15	Cenário de Teste 3 - 50 Clientes	60
5.16	Cenário de Teste 3 - Query 5 Clientes	61
5.17	Cenário de Teste 3 - Query 20 Clientes	61
5.18	Cenário de Teste 3 - Query 50 Clientes	61
5.19	Cenário de Teste 4 - 5 Clientes	65
5.20	Cenário de Teste 4 - 20 Clientes	66
5.21	Cenário de Teste 4 - 50 Clientes	66
5.22	Cenário de Teste 4 - Query 5 Clientes	67
5.23	Cenário de Teste 4 - Query 20 Clientes	67
5.24	Cenário de Teste 4 - Query 50 Clientes	67
5.25	Comparativo de melhores casos de performance	71
5.26	Comparativo de melhores casos de performance	74

Capítulo 1

Introdução

1.1 Contextualização

No ano de 2017, o transporte aéreo de passageiros representou mais de 67% do transporte regular [2]. Além disso, foram registrados mais de 800 mil voos, transportando em média 90 milhões de pessoas e gerando uma variação de aproximadamente 3% na demanda de transporte aéreo [2]. Para viabilizar essa grande demanda de passageiros, os aeródromos devem fornecer a infraestrutura de solo pertinente, como: pistas de pouso e decolagem, luzes, indicadores, pistas de táxi, entre outros. De modo que os objetivos principais, como melhoria na segurança e maximização da capacidade aeroportuária, sejam alcançados.

O sistema aéreo brasileiro opera sobre aeródromos por meio de duas operações: *Regras de Voo Visual*, que seguem a forma de orientação visual e os pilotos das aeronaves são orientados por visuais externos, como sinalização luminosa e indicador de direção do vento. Outro tipo de operação consiste nas *Regras de Voo por Instrumentos*, que se baseiam nas cartas de navegação, sendo utilizadas com auxílio de instrumentos aeronáuticos [3]. De maneira que o sistema baseado em instrumentos funcione como forma de navegação aérea da maioria dos voos comerciais [3]. Sendo assim, para assegurar as informações necessárias de infraestrutura de navegação aérea e apoio às operações de voo, é necessário gerenciar o tráfego aéreo apropriadamente.

Atualmente, o gerenciamento de tráfego aéreo ou *air traffic management (ATM)* é feito através do *Conceito Operacional ATM Global*, que é um sistema baseado em sete componentes principais: Organização e Gerenciamento do Espaço Aéreo, Operações de Aeródromo, Balanceamento de Demanda e Capacidade, Sincronização de Tráfego, Operações dos Usuários do Espaço Aéreo, Gerenciamento de Conflitos e Gerenciamento de Entregas de Serviços *ATM* [4]. Além desses componentes, o *ATM* brasileiro apresenta duas características essenciais. A primeira se trata do gerenciamento da informação, que visa proporcionar informações confiáveis, oportunas e com garantia de qualidade. A se-

gunda característica consiste na decisão colaborativa que objetiva a ampla participação de todos os interessados [4].

Nas últimas décadas, o tráfego aéreo tem aumentado em todo o mundo e tende a se manter nesse ritmo nos próximos anos [5]. Para atender essa demanda e resolver as limitações do *ATM* atual, como a comunicação entre os agentes envolvidos e a qualidade dos dados nas trocas de informação. Dessa maneira, novas soluções de controle de tráfego aéreo estão sendo desenvolvidas conforme a descrição no plano de navegação aérea global (*GNAP*) da *International Civil Aviation Organization (ICAO)* [6]. Com essa finalidade, diferentes programas começaram a aprimorar soluções de navegação, como o *Single European Sky ATM Research (SESAR)* na Europa e o *Next Generation Air Transportation System (NextGen)* nos EUA. Esses programas convergem para promover a evolução do controle de tráfego aéreo (*ATM*) em um ambiente automatizável, integrado e interoperável [7] [8].

Nesse contexto, surge o conceito de *System Wide Information Management (SWIM)*, que consiste em definições, infraestrutura e governança habilitando o controle da informação *ATM* e trocas entre partes qualificadas através de serviços interoperáveis [9]. Além disso, existe o *SWIM Registry*, que é o elemento chave do *SWIM*, visando aprimorar a visibilidade das informações relacionadas ao *ATM* e serviços disponíveis no *SWIM*.

O sistema *ATM* considera o intercâmbio e o gerenciamento da informação utilizada nos serviços e processos, sendo uma ferramenta para garantir a coesão e vinculação entre os componentes. A principal finalidade é proporcionar dados confiáveis, oportunos e de qualidade garantida para que sejam utilizados em apoio ao *ATM* [4].

Devido às características específicas do controle de tráfego aéreo brasileiro, foi necessário elaborar um modelo que gerencie todas informações no país e realize a integração de todos os agentes envolvidos. Para cumprir esse propósito, o *Departamento de Controle do Espaço Aéreo (DECEA)* e o *Instituto de Controle do Espaço Aéreo (ICEA)* propuseram o *SWIM Registry Brazil*, que consiste em um repositório contendo os serviços disponíveis no *SWIM*.

O *SWIM Registry Brazil* foi desenvolvido e é mantido pelo *Laboratório de Modelo Computacional para Transporte Aéreo (Translab)* na *Universidade de Brasília (UnB)*. Atualmente é um protótipo que conta com um *website*, um sistema núcleo do *Registry* e sistema de administração do *Registry*. O *SWIM Registry Brazil* requer segurança nas operações realizadas entre as entidades envolvidas. No momento, o *SWIM Registry Brazil* possui dois serviços implementados, tendo seus dados armazenados e inseridos através de um sistema com a segurança de apenas *login*, senha dos *stakeholders* e pelo email de confirmação dos responsáveis pelo *SWIM*, que não satisfaz os requisitos de segurança.

A partir desses aspectos, sistemas *ATM* podem ser aprimorados utilizando a tecnologia *blockchain* [10]. O *blockchain* foi empregado com sucesso como uma ferramenta de gerenciar operações no sistema de criptomoedas *Bitcoin*. O *blockchain* é essencialmente um banco de dados distribuído, também definido como um livro-razão de todas as transações ou eventos digitais que estão sendo executados e compartilhados com os participantes da rede. O *blockchain* é estruturado como uma cadeia de blocos, em que cada bloco armazena um conjunto de transações [11]. Os blocos são ligados por meio de campos específicos como o *hash*, uma assinatura digital criptografada, do bloco, o *hash* do bloco anterior e um índice em sequência.

A partir dessa estrutura do bloco e da formação de uma cadeia é necessário um consenso dos participantes da rede para que ocorra uma transação e a informação seja validada. Após a etapa de validação pelos participantes, ocorre o processo de inserção no bloco. Caso o bloco esteja cheio, é necessário que aconteça a mineração de um novo bloco (*proof-of-work*). Isto é, encontrar um valor de hash que satisfaça as condições: dificuldade escolhida, que é a quantidade de zeros iniciais e número de tentativas, que é a quantidade de tentativas para encontrar o valor do hash. Com isso minere o bloco para ser inserido ao fim da cadeia.

A principal hipótese em utilizar o *blockchain* em sistemas *ATM* se deve à criação de um sistema de consenso entre as entidades envolvidas, uma vez que todos os eventos realizados estão registrados de maneira irrefutável no livro-razão [11], garantindo uma maior segurança aos dados contidos no sistema. Além disso, pode-se promover uma maior transparência das informações aos *stakeholders*, graças às características do *blockchain* e permissões de segurança contidas na aplicação.

O *blockchain* foi adaptado com sucesso em outras aplicações de diversos domínios do conhecimento como medicina [12], eleição [13], segurança da informação [14]. Nesse sentido, a implantação do *blockchain* no *SWIM Registry Brazil* pode demandar ajustes específicos na estrutura do bloco, nas etapas de validação e consenso digital dos agentes interessados na rede *blockchain*. Além disso, deve-se levar em consideração que o *SWIM* deve ser uma rede colaborativa, automatizando o gerenciamento das informações, visando a busca pela qualidade dos dados.

1.2 Objetivos

O objetivo geral desse projeto é realizar uma integração da tecnologia *blockchain* com os serviços do *SWIM Registry Brazil*. Desta maneira, criando uma aplicação *blockchain* para promover uma maior segurança as informações do serviço de plano de voo.

1.2.1 Objetivos Específicos

Para cumprir o objetivo geral, os seguintes objetivos específicos podem ser descritos:

- **Implementar a tecnologia *blockchain* nos serviços do *SWIM Registry Brazil*:** analisar o documento que descreve o *SWIM Registry Brazil*, identificando os serviços descritos e implementando-os em uma rede *blockchain*. Nesse sentido, cada serviço descrito do contexto do *SWIM Registry Brazil* será associado a uma cadeia individual do *blockchain*;
- **Desenvolver uma arquitetura de acordo com os serviços do *SWIM Registry Brazil*:** através dos documentos e serviços presentes no *SWIM Registry*, desenvolver uma arquitetura global que permita a criação de cada serviço específico;
- **Desenvolver uma aplicação baseada em *blockchain* para o serviço de plano de voo:** de acordo com a arquitetura proposta e a análise do serviço de Plano de Voo, desenvolver uma aplicação *blockchain* para esse serviço;
- **Identificação dos melhores parâmetros:** realizar uma análise comparativa dos cenários de testes, identificando os melhores resultados de performance sobre a aplicação.

1.3 Metodologia

A metodologia proposta é uma solução alternativa em relação àquela utilizada atualmente nos serviços do *SWIM Registry Brazil*. Desta maneira, pretende-se criar uma aplicação *blockchain* para o serviço de plano de voo. A pesquisa será realizada de acordo com as seguintes etapas:

- **Análise dos serviços do SWIM:** estudo aprofundado dos serviços presentes do *SWIM Registry Brazil*;
- **Levantamento do estado da arte das aplicações *blockchain*:** estudo de aplicações *blockchain* em diversas áreas do conhecimento;
- **Definição do serviço:** proposta que identifica o serviço de plano de voo para ser implementado na aplicação *blockchain*;
- **Levantamento de dados de planos de voo:** pesquisa de informações dos planos de voo realizados e de seus campos específicos, além de uma análise de fluxo de estados;

- **Definição da arquitetura:** proposta que realiza a integração do serviço de plano de voo com *blockchain* e que identifique a melhor composição da rede blockchain dentro dos indicadores de performance;
- **Implementação da arquitetura:** desenvolvimento de uma aplicação que implementa a arquitetura proposta e seus componentes;
- **Implementação dos cenários de teste:** implementação dos cenários de teste para a realização das simulações da aplicação com a utilização de diferentes parâmetros para a rede;
- **Simulação:** realização das simulações da aplicação dentro dos cenários de teste;
- **Análise de resultados:** avaliação e discussão dos resultados obtidos de acordo com os indicadores de performance.

Em última análise, a solução proposta deve contribuir para melhorar a segurança das informações dentro do *SWIM Registry Brazil*, além de possibilitar o desenvolvimento de novos serviços do *SWIM Registry Brazil* baseados em *blockchain*.

1.4 Organização do Documento

O documento está organizado da seguinte maneira: O Capítulo 2 apresenta a fundamentação teórica referente a sistemas de transporte aéreo e uma revisão de literatura que expõe os conceitos de *ATM*, *SWIM*, *SWIM Registry* e *SWIM Registry Brazil*, as definições e conceitos de *blockchain*.

O Capítulo 3 apresenta os trabalhos recentes na literatura abordando: 1) aplicações que utilizam o blockchain e 2) utilização do blockchain no sistema aéreo. O Capítulo 4 apresenta a modelagem da arquitetura, demonstrando seus aspectos, fonte de dados, estrutura e justificativa, além disso é detalhado a implementação da arquitetura, através da explicação da ferramenta e das funcionalidades.

O Capítulo 5 apresenta uma descrição detalhada dos indicadores de performance, além dos detalhes e descrições dos cenários de testes. Assim como uma exposição dos dados obtidos em cada cenário e uma análise sobre os mesmos.

O Capítulo 6 apresenta as conclusões sobre a pesquisa, assim como sugestões de trabalhos futuros.

Capítulo 2

Fundamentos

2.1 Sistemas de Aviação

2.1.1 SWIM

No contexto do *System Wide Information Management (SWIM)*, a arquitetura orientada a serviços é utilizada para representar e definir a troca de informações entre usuários do SWIM, isto é, um provedor de informações publica e expõe os serviços a serem utilizados pelos consumidores. Isto é feito através de registros interconectados que listam os serviços e especificam os detalhes para seus consumidores. Um dos benefícios dessa abordagem é promover um acoplamento suave, significando que o provedor da informação tem impacto reduzido no consumidor, ou seja, o consumidor precisa entender apenas o que é estritamente necessário para utilizar o serviço [9]. O conceito SWIM é baseado em serviços interoperacionais, em que cada camada tem suas próprias funções, combinações de padrões e mecanismos interoperacionais. A interoperabilidade do SWIM abrange as seguintes camadas:

- Aplicações de informações e provedores de informação utilizando o SWIM;
- Serviços de troca de informações definidos para cada domínio de informação de *ATM* e para fins de domínios cruzados;
- Modelos de troca de informações utilizando definições de um tema específico para serviços de compartilhamento de informações;
- Infraestrutura que alimenta os serviços básicos para o compartilhamento de informações;
- Rede de conectividade que promove serviços de telecomunicações consolidados.

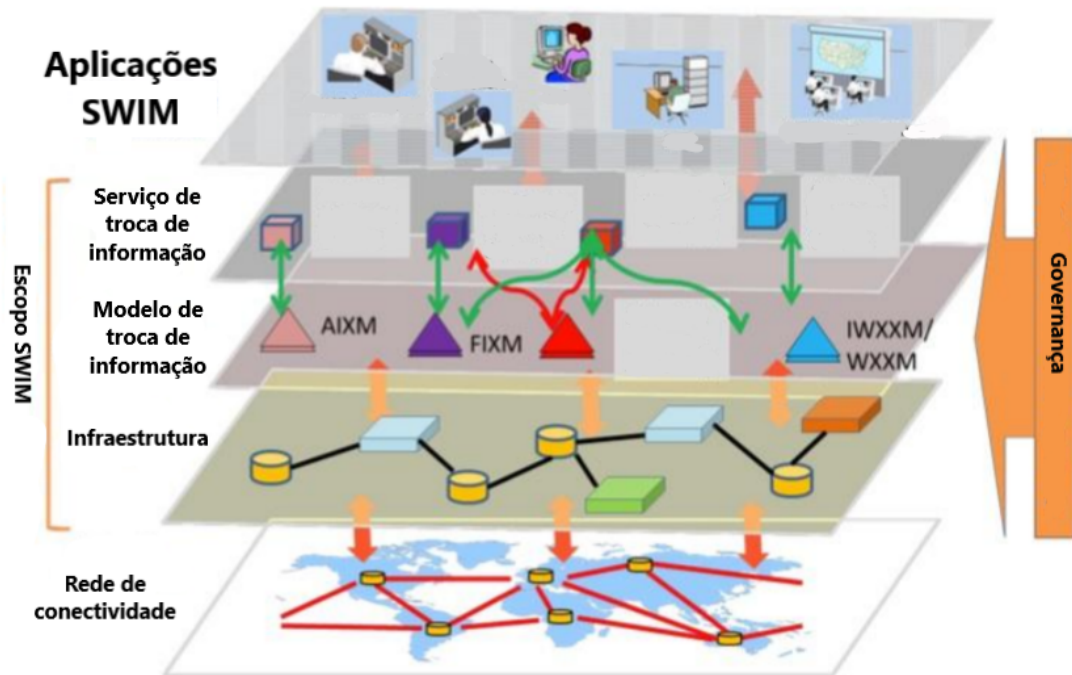


Figura 2.1: Escopo do SWIM adaptado do ICAO. International Civil Aviation Organization, “Manual on System Wide Information Management (SWIM) Concept, (Doc 10039)”, Montreal, 2015.

O escopo do SWIM é limitado a três camadas intermediárias (Serviços de Troca de Informação, Modelos de Troca de Informação e Infraestrutura SWIM) e ao gerenciamento dessas camadas [9] que pode ser visualizado pela Figura 2.1. O propósito de cada camada na ferramenta de interoperabilidade *SWIM* é descrita por:

- **Aplicações *SWIM***: compreende os sistemas aptos a requisitar os serviços de informações. A interoperabilidade semântica baseada em um entendimento comum das informações utilizadas é requerida.
- **Serviço de troca de informações**: as características do serviço de informações requisitado são descritas de uma forma tecnologicamente neutra.
- **Modelo de troca de informações**: as características dos dados utilizados pelo serviço de troca de informações incluem a descrição do conteúdo, estrutura e formato das informações.
- **Infraestrutura**: alimenta o núcleo de serviços SWIM assim como a interface de gerenciamento, troca de mensagens, segurança e gerenciamento de serviços de entendimento. Nessa camada, a interface de troca de mensagens requisitada para entregar o serviço de informação requerido é implementada de acordo com o protocolo definido para a interoperabilidade com o entregador dos serviços através de

uma interface de gerenciamento de funções que também gerenciam os requisitos de performance.

- **Rede de conectividade:** a mensagem é transportada através de uma rede global, na qual é entregue a um destinatário específico responsável pelo provimento da interface de mensagem à aplicação SWIM.

2.1.2 SWIM Registry

O *SWIM Registry* é o elemento chave no *SWIM* que garante os aspectos associados à interoperabilidade e permite a aplicação de regras quando apropriado. Isto permite aos fornecedores das informações publicar serviços e aos consumidores encontrar esses serviços a partir das informações disponíveis no *Registry* [15]. O *Registry* visa aprimorar a visibilidade e acessibilidade das informações relacionadas ao *ATM* e serviços disponíveis *SWIM*, permitindo que os usuários interessados compartilhem uma visão comum do *SWIM*. O *Registry* é a fonte de referência para serviços disponíveis em aplicações *SWIM* e detalha o conjunto de serviços fornecidos, com informações qualitativas, consolidadas e estruturadas, assim como as condições da operação [15]. Em resumo, o *SWIM* agrega diferentes participantes de distintas áreas de negócios em um único ambiente de troca de informações. Neste contexto, o *SWIM* garante um controle de acesso eficiente, segurança da informação, disponibilidade de serviços, entretanto, o grande desafio é promover a confiabilidade dos dados.

Arquitetura do SWIM Registry

A arquitetura do *SWIM Registry* se baseia em três entidades principais: provedores de serviços, consumidores de serviços e da autoridade reguladora [9]. Provedores de serviços são organizações por trás da implementação dos serviços relacionados ao *SWIM*, enquanto que consumidores de serviços são as organizações responsáveis por implementar as aplicações *SWIM*. Os serviços são acessados por aplicações *SWIM* assim como por outros serviços. As aplicações são acessadas e utilizadas por usuários *SWIM*, com a intenção de solicitar alguma informação para realização de uma tarefa. A autoridade reguladora do *SWIM* auxilia o registro de serviços e aplicação no *Registry*. A autoridade reguladora também é responsável por informações relacionadas ao serviço de publicação, como definições, modelos de troca, políticas etc. Para gerenciar e catalogar serviços, o *SWIM Registry* tem que abranger certas funcionalidades, sendo que as mais importantes estão listadas abaixo [9]:

- Publicar serviços de forma básica;

- Buscar por serviços relacionados ao *SWIM* e acesso à suas informações;
- Sistema de notificação para usuários e partes interessadas.

Serviços de publicação são a funcionalidade central do *Registry*. A publicação tem que ser feita de uma forma básica seguindo regras previamente definidas pela autoridade reguladora. A autoridade SWIM define as políticas as quais os serviços deverão obedecer para publicar no Registry. Este conjunto de políticas contribuem para o gerenciamento completo e a interoperabilidade no SWIM. Outra funcionalidade central do *Registry* é permitir que consumidores solicitem serviços. Como um repositório de serviços, o *Registry* precisa oferecer uma ferramenta de busca com filtros que permite aos consumidores facilmente descobrir os serviços disponíveis. A arquitetura do *Registry* tem que permitir que os usuários busquem serviços por uma certa quantidade de atributos, como categoria de dados de *ATM*, categoria de atividades de *ATM*, descrição, palavras-chave, entre outros.

A descoberta de serviços é relacionada ao registro de serviços de uma forma que registrar serviços de metadados úteis contribuem para que o usuário localize serviços mais facilmente em sua busca. Gerenciar mudanças em serviços de forma clara e transparente é um aspecto desejável da arquitetura do *Registry*. Essa funcionalidade serve como uma forma de checar o progresso de serviços na plataforma SWIM. O serviço de notificações contribui para um ambiente SWIM aberto a atualizações [9]. Quando a aplicação SWIM toma conhecimento de mudanças em um serviço, este pode ser atualizado para a nova versão sem prejuízo nos dados. Usuários podem ser também notificados quando mudanças nas informações ao *ATM* acontecerem, como modelos de referência, serviços de troca de informação, documentação etc. O sistema deve permitir que usuários tenham a opção de receber notificações sobre atualizações em serviços e outras informações de seu interesse. O *SWIM Registry* encoraja usuários a buscarem serviços e informações relacionadas ao *ATM* ao integrá-los em um único ambiente. Além disso, o *Registry* torna mais básica a forma com a qual serviços e aplicações são publicados, criando um ambiente mais consistente e interoperável para seus usuários [9]. Por último, o *Registry* agrega em um único lugar parceiros confiáveis e usuários ativos.

2.1.3 SWIM Registry Brazil

O *SWIM Registry Brazil* é um esforço para criar um *SWIM Registry* para o ambiente brasileiro de *ATM*. O protótipo do *SWIM Registry Brazil* é atualmente mantido no TransLab na Universidade de Brasília, como pode ser visualizado na Figura 2.2. O *SWIM Registry* demanda que os usuários terão seu acesso garantido ao *SWIM Registry Brazil* de acordo com o papel que eles desempenharam no *Registry*. O DECEA deverá desempenhar o pa-

pel da autoridade reguladora do ambiente SWIM, sendo responsável pela administração do *Registry* e da definição das políticas e definições utilizados no ambiente. Aeroportos, aeródromos, companhias aéreas, instituições de ensino e pesquisa, agências públicas, entre outros serão os provedores de serviços e consumidores de serviços do *Registry*.

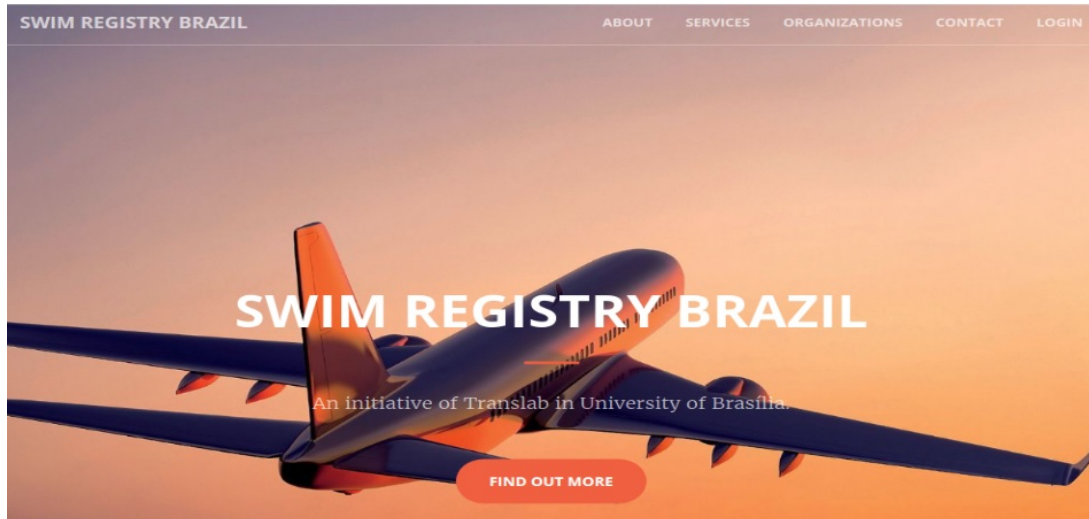


Figura 2.2: Site do *SWIM Registry Brazil*.

O *SWIM Registry Brazil* pode ser dividido em três principais subsistemas: *website*, sistema núcleo do *Registry* e sistema de administração do *Registry*. O principal propósito do *website* será compartilhar notícias sobre os atuais estágios de desenvolvimento do *Registry*, atrair usuários para a nova plataforma e mostrar os tipos de serviço disponíveis na plataforma. O sistema núcleo do *Registry* inclui as principais funcionalidades do *SWIM Registry* das perspectivas do consumidor e do provedor de informações. Para ter acesso a essas funcionalidades, o usuário deverá se registrar no sistema e ter seu cadastro analisado e aprovado pela autoridade reguladora. Já o sistema de administração do *Registry* é responsável por dar suporte às atividades da autoridade reguladora. Entre outras atribuições, a autoridade reguladora é responsável por reforçar definições e políticas junto às partes interessadas do *SWIM* ao publicar e manter atualizadas todas as informações relativas ao *ATM*. A respeito das informações relacionadas ao *ATM* disponíveis no *SWIM Registry Brazil*, podemos categorizá-las da seguinte forma:

- Lista dos serviços SWIM;
- Serviço de descrição da informação;
- Modelos de referência;
- Definições de troca de informação.

A informação relacionada a serviços inclui todos os metadados associados àquele serviço específico. Vale notar que as aplicações do *SWIM* podem também ser publicadas e catalogadas no *Registry* e têm que satisfazer certos requisitos em ordem para ter sua informação publicada no *Registry* do mesmo modo que os serviços. Modelos de referência e definições de troca de informação são importantes para definir a forma na qual a informação é representada pelos serviços. Dessa forma, desenvolvedores não vão desperdiçar tempo tentando entender novos modelos e conceitos já que eles estarão familiarizados com as definições de modelos de referência. Os participantes do ambiente *SWIM Registry Brazil* podem ser categorizados como usuários e organizações. Usuários somente terão acesso ao *SWIM Registry Brazil*, enquanto as organizações serão responsáveis por publicar serviços ou aplicações. O *SWIM Registry* também contém uma lista de aplicações e serviços relacionados ao *SWIM* para cada organização com a finalidade de auxiliar a comunidade a acompanhar as contribuições e atividades da companhia no *Registry*. Finalmente, outra importante ferramenta disponível no *SWIM Registry Brazil* é o sistema de notificações. Este sistema permite que usuários cadastrem serviços específicos para receberem atualizações sobre os mesmos. Usuários também podem optar por receber notificações de atualizações para se manter a par do progresso do *Registry*, como novas aplicações, serviços e organizações na plataforma.

2.1.4 Demonstração do conceito SWIM

Para demonstrar a aplicabilidade do ambiente *SWIM* e do *SWIM Registry Brazil* no contexto brasileiro de *ATM*, uma prova do conceito foi coordenada pelo Instituto de Controle do Espaço Aéreo (ICEA) e pela Universidade de Brasília (UnB). A arquitetura proposta consiste no *SWIM Registry Brazil*, dois serviços de *ATM* em uma aplicação *ATM*. A arquitetura está representada na Figura 2.3:

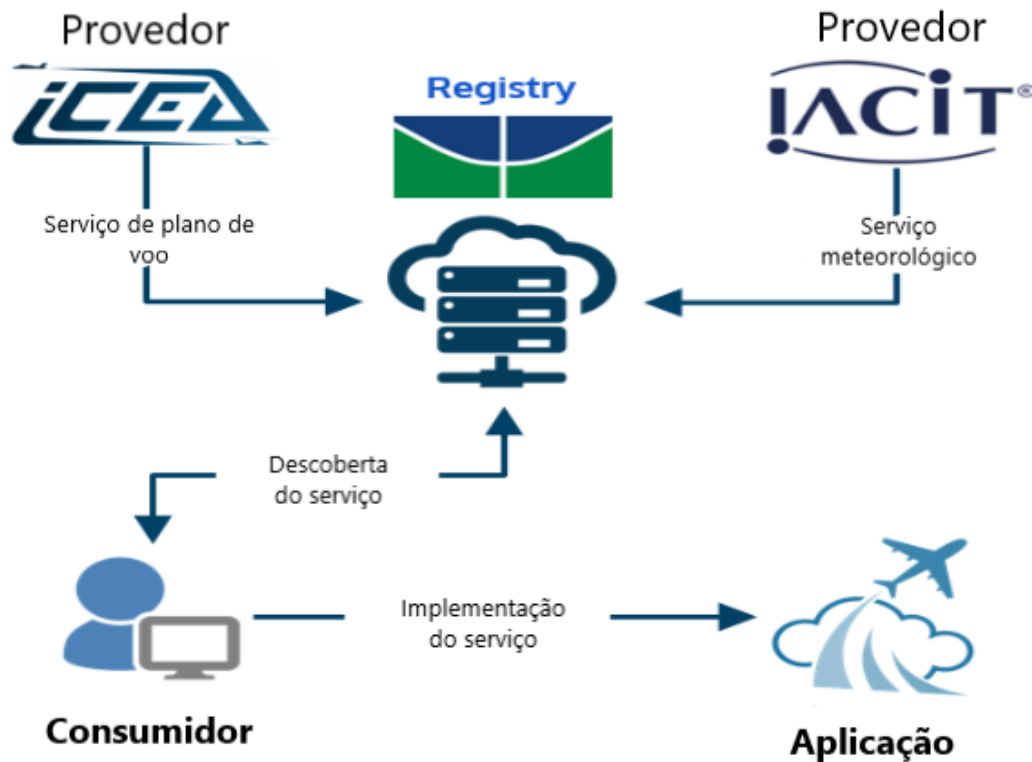


Figura 2.3: Demonstração do conceito SWIM Registry.

A Figura 2.3 apresenta os provedores que são as entidades que provém os serviços, no caso o *ICEA* e o *IACIT*, os consumidores que são as entidades que consomem os serviços do *SWIM Registry Brazil* e os serviços que são as atividades que as entidades provém ao *Registry*, no caso serviço de plano de voo, pelo *ICEA*, e serviços meteorológicos, pelo *IACIT*.

2.1.5 Plano de Voo

Para a operação de voos no Brasil, é necessário um plano de voo, sendo esse escrito pelo piloto ou comandante da aeronave. O plano de voo é caracterizado como um documento que abarca informações relacionadas com um voo planejado de uma aeronave. Esses dados

são proporcionados para os órgãos responsáveis que prestam os serviços trafego aéreo, sendo necessário a posse dos mesmos antes da realização do voo [16]. Neste contexto, existem três tipos de plano de voo:

- Plano de Voo Completo;
- Plano de Voo Simplificado;
- Plano de Voo Repetitivo.

Para um plano de voo ser valido pelo órgãos competentes, certos dados devem ser informados no preenchimento do plano de voo, como:

- **Identificação da aeronave:** Conjunto de caracteres alfanuméricos únicos para identificação da aeronave;
- **Regras de voo:** Identificação do tipo de navegação no espaço aéreo que o piloto utilizará;
- **Tipo de voo:** Identifica qual o tipo do voo;
- **Número de aeronaves:** Identifica a quantidade de aeronaves, quando se trata de voo em formação;
- **Tipo de aeronave:** Identifica o designador apropriado para o tipo da aeronave;
- **Categoria da esteira de turbulência:** Identifica a esteira de turbulência;
- **Equipamento e capacidades:** Identifica os equipamentos e a capacidade dos mesmos;
- **Aeródromo de partida:** Identifica o indicador de localidade do aeródromo de partida;
- **Hora:** Identifica a hora real da decolagem;
- **Velocidade de Cruzeiro:** Identifica a velocidade verdadeira de cruzeiro;
- **Nível de Cruzeiro:** Identifica a altitude do voo;
- **Rota:** Identifica a rota do voo, incluindo as mudanças de velocidade, nível e regras de voo;
- **Aeródromo de destino:** Identifica o indicador de localidade do aeródromo de destino;
- **Duração total de prevista de voo:** Identifica a duração total prevista do voo;

- **Aeródromo alternativo de destino:** Identifica os indicadores de localidade de aeródromos alternativos ao de destino;
- **Outros dados:** Identifica qualquer outra informação necessária ao voo;
- **Autonomia:** Identifica a autonomia do avião em horas e minutos;
- **Pessoas a bordo:** Identifica o número total de passageiros e tripulantes a bordo;
- **Responsável, código ANAC e assinatura:** Identifica o nome do responsável pelo preenchimento do plano de voo e assinatura do mesmo;

2.2 Blockchain

2.2.1 Introdução

A tecnologia do *blockchain* foi introduzida por *Satoshi Nakamoto*, sendo popularmente reconhecida pelo seu emprego no *Bitcoin* [10]. O *blockchain* pode ser definido com uma estrutura de dados, funcionando como uma sequência de blocos conectados, cada um armazenando um conjunto finito de transações que podem ser operações ou registro de dados. Cada bloco apresenta uma assinatura digital, que é referenciada criptograficamente por um valor de *hash*, promovendo autenticidade e identidade para a criação e verificação das transações. A estrutura que define a organização dos blocos em cadeia é alcançada em virtude das seguintes características: o valor de *hash* do bloco anterior, ou seja, do bloco pai, e o valor de *hash* do próprio bloco, exceto pelo primeiro bloco, conhecido por bloco genesis. O *blockchain* pode ser estruturado da seguinte maneira:

- Estrutura básica
- Função Hash
- Protocolos de Consenso
- Mineração dos blocos
- Smart Contracts

2.2.2 Estrutura Básica

O *blockchain* possui uma estrutura definida, caracterizada por um conjunto de blocos que são sequencialmente conectados, formando uma espécie de cadeia. A estrutura básica do *blockchain* consiste em alguns campos gerais que estão presentes na maioria das estruturas, que são os campos *index*, *timestamp*, *hash*, o valor *hash* do bloco anterior e o campo *data*, que depende do tipo de dados.

- *Index*: Contém o índice do bloco;
- *Timestamp*: Contém a data e hora de quando o bloco foi minerado e colocado na cadeia;
- *Hash*: Contém o valor do *hash* do bloco;
- *Hash* do bloco anterior: Contém o valor do *hash* do bloco pai, a partir desse valor que é feito a conexão dos blocos;
- *Data*: Contém o campo de dados do bloco na área de conhecimento na qual o *blockchain* foi modelado.

Os blocos são conectados uns aos outros através do valor de *hash* armazenado no bloco anterior (bloco pai) fazendo que a estrutura do *blockchain* forme uma cadeia, na qual nós recém criados sejam adicionados ao final da cadeia conectado com o bloco pai. A Figura 2.4 mostra o exemplo dos blocos em cadeia.

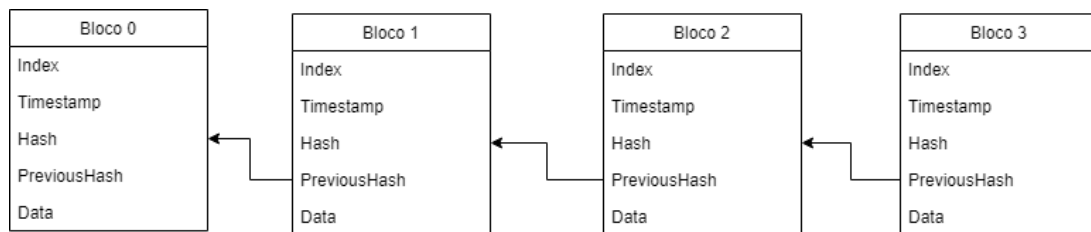


Figura 2.4: Exemplo de uma cadeia de blocos.

2.2.3 Função Hash

A tecnologia *blockchain* apresenta duas funções importantes em sua abordagem, que são a função *hash* e o registro de data e hora. A função *hash* consiste em executar um algoritmo de computador sobre um arquivo de conteúdo e obter como resultado uma cadeia de caracteres compactada que não pode ser computada para sua forma original. Essa cadeia, conhecida como *hash*, representa exatamente o conteúdo do arquivo original. Sendo assim, caso o conteúdo do arquivo precise ser reconfirmado, executa-se a mesma função *hash*, se o arquivo não tiver sido alterado, a assinatura *hash* será a mesma [17].

No *blockchain*, o *hash* é pequeno o suficiente para ser incluído como um campo texto nas transações, fornecendo o registro da hora de quando ocorre uma transação. Devido a função *hash*, o conteúdo do arquivo original é codificado dentro do *blockchain* [17]. A ideia principal é utilizar a função *hash* como forma de verificar e validar as informações do bloco, sendo de extrema importância. Visto que essa função, dentro do *blockchain*, serve como base para provar a existência e o conteúdo de qualquer documento ou outro ativo em um determinado momento [17].

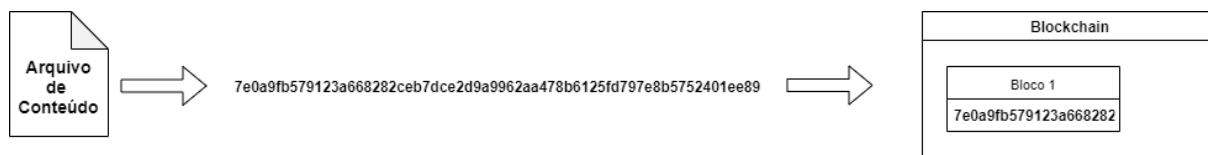


Figura 2.5: Exemplo da função *hash*.

A Figura 2.5 demonstra como ocorre a função *hash* no *blockchain*. Primeiro tem-se o arquivo com conteúdo, em seguida é executado uma função *hash* sobre o mesmo, gerando como resultado uma cadeia de caracteres. Logo após, essa sequência de caracteres é armazenada dentro do bloco da *blockchain*.

No contexto do *Bitcoin*, o valor do *hash* é gerado através de uma função criptográfica conhecida como *sha256*. Esta função é aplicada nos campos do bloco, que são o valor do *hash* do bloco anterior, *timestamp* e os dados do campo data do bloco. Em seguida um valor de *hash* é gerado e armazenado em seu campo dentro do bloco. Esse valor é iniciado por uma quantidade de zeros definida pela dificuldade que foi imposta no *blockchain*.

2.2.4 Transações

Transações são estruturas de dados que codificam as transferências de ativos entre participantes de um sistema que utiliza o *blockchain*. No caso do sistema *Bitcoin*, representa a transferência de moedas entre os participantes da rede. As transações foram projetadas para que possam ser criadas, propagadas pela rede, validadas e ao fim serem adicionadas ao *blockchain*. Além disso, cada transação é considerada uma entrada pública para ser registrada no *blockchain* do sistema [18].

As transações são caracterizadas como uma transferência de ativos de uma fonte, chamada transação de entrada, até o destino, chamada transação de saída. Entretanto, esses tipos não são necessariamente relacionados com contas ou identidades de usuários [18].

O ciclo de uma transação começa na sua criação, em seguida ocorrer as assinaturas digitais, indicando uma autorização sobre os ativos da transação. Depois dessa assinatura, ocorre a transmissão na rede *blockchain*, em que cada participante valida e propaga a transação até alcançar todos os participantes da rede. Por fim, a transação é verificada por um nó minerador e incluída em um bloco de transações que é registrada no *blockchain*. Uma vez registrada no *blockchain* e confirmada por um número suficiente de blocos subsequentes, a transação faz parte permanente da rede e é considerada válida por todos os participantes. Posteriormente, o ativo utilizado poderá ser empregado em uma nova transação, começando assim um novo ciclo.

2.2.5 Tolerância a Falha Bizantina

Um componente defeituoso apresenta um tipo de comportamento oposto ao padrão, enviando informações conflitantes para diferentes partes do sistema. Sendo assim, um sistema computacional tem que estar apto a operar com a falha de um ou mais componentes [19]. Esse tipo de falha é expresso no *Problema dos Generais Bizantinos*.

Problema dos Generais Bizantinos

O *Problema dos Generais Bizantinos* (BGP) [19] descreve que diversas divisões do exército *Bizantino* cercam uma cidade inimiga, onde cada divisão é comandada por seu próprio general. Esses somente podem se comunicar por mensagem. Sendo assim, após observar o inimigo, os generais decidem executar o plano de ação. Porém, alguns generais podem ser traidores.

Considerando este aspecto, é necessário que os generais leais tenham um algoritmo que garanta que os mesmos alcancem um comum acordo. Os comandantes leais vão seguir o que o algoritmo propõem, enquanto os traidores farão aquilo que desejam [19]. Sendo assim, a ideia é garantir que as partes confiáveis atinjam um consenso sobre o melhor plano a ser executado.

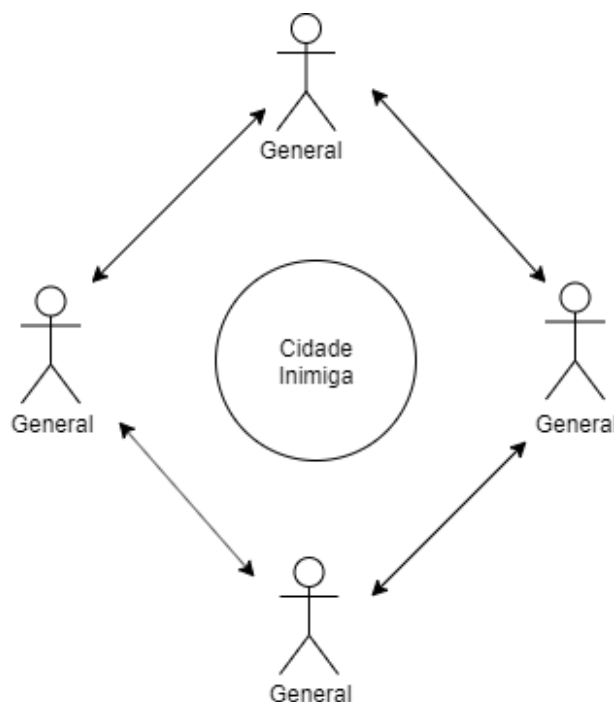


Figura 2.6: Exemplo adaptado do Problema dos Generais Bizantinos.

A Figura 2.6 exemplifica o *BGP*, sendo possível perceber os generais cercando uma cidade inimiga, esperando o sinal de comunicação entre os mesmos, para assim tomarem a decisão de atacar ou não a cidade.

Esse problema ilustra o aspecto da falha bizantina para sistemas de computadores, no qual um ou mais componentes podem falhar e passar uma informação equivocada para o sistema, gerando um mal funcionamento. Sendo assim, o sistema precisa se resguardar dessas possíveis falhas e adotar métodos para se proteger.

É neste contexto que surge a definição de Tolerância a Falha Bizantina (*BFT*), que pode ser descrito como algoritmos que garantem que um sistema seja tolerante a classe de falhas do *BGP*. Essa característica é muito importante para sistemas sem uma autoridade central de controle, como o *blockchain*. Visto que se algum nó transmitir alguma transação falsa, o sistema será capaz de identificar e reparar o dano [20].

Nesse aspecto, surge o *Proof-of-work*, uma solução probabilística para o *BGP*, desenvolvida por *Satoshi Nakamoto*, durante a invenção do *Bitcoin* [10], sendo um dos tipos de protocolo de consenso da tecnologia *blockchain*.

2.2.6 Protocolos de Consenso

A rede *blockchain* não possui uma entidade central que regulamenta e valida as transações entre os nós da rede, conhecidos como *stakeholders* que são os agentes envolvidos na rede *blockchain*, ou seja, aquelas pessoas que vão ter acesso ao sistema para poderem ler e/ou escrever dentro de um bloco. Então para ajudar a rede a alcançar uma visão global de mundo, surge a ideia de consenso da rede ou protocolo de consenso, onde cada rede *blockchain* precisa estabelecer um certo conjunto de regras na qual cada transação do banco de dados deverá seguir. Essas regras são programadas dentro de cada *blockchain* e são utilizadas para verificar se uma transação é válida e conseqüentemente se é redistribuída por toda rede ou não. O protocolo de consenso varia de acordo com as regras específicas, o tipo de rede *blockchain*, podendo ter diversos protocolos como: *proof-of-work (PoW)*, *proof-of-stack (PoS)* etc.

Proof-of-work

A mineração no *blockchain* é o processo que consiste em achar um bloco válido, seguindo os padrões da estrutura básica, para ser inserido na cadeia. Esta etapa ocorre seguindo alguns passos:

- Dificuldade escolhida: Consiste em definir uma quantidade inicial de zeros, para inicializar o valor de *hash* que será gerado.

- Número de tentativas: Consiste na quantidade de tentativas para gerar o valor de *hash*, que alcançará a dificuldade dada.

A mineração no *blockchain* consiste em uma solução para um problema criptográfico, que é encontrar um valor de *hash* que comece com a quantidade de zeros no seu início. Esse processo se baseia em uma estratégia de tentativa e erro, demandando um grande processamento computacional. Ao final, armazena-se dentro do bloco a variável que contém a quantidade de tentativas.

Verificação da Cadeia

A verificação da cadeia é uma parte muito importante dentro do contexto do *blockchain*. Este processo visa garantir a validade da cadeia, isto é, se o *blockchain* segue os padrões. Para que uma cadeia seja válida:

- Campo *index* de cada bloco deve estar em sequência, ou seja, o campo que contém o valor do *hash* anterior deve ter o mesmo valor do campo de *hash* do bloco pai;
- Campo *hash* deve seguir o padrão determinado com a quantidade de zeros iniciais.

Este processo é um algoritmo que verifica todas essas etapas e tem como objetivo garantir a segurança do *blockchain*. A partir dessa verificação, pode-se detectar qualquer alteração ou tentativa de alteração dos dados e com isso garantir o funcionamento e a segurança do *blockchain*.

Tipos de protocolo

Como o *PoW* apresenta algumas desvantagens evidentes, como a latência de decisão do bloco, vazão relativamente baixa e a sustentabilidade do processo de mineração, surgem outros protocolos de consenso com base em outros critérios, como capacidade de armazenamento, comprometimento com a rede, quantidade de *tokens* etc.

Proof-of-Stake

O protocolo de consenso *PoW* apresenta algumas limitações e dependências, uma dessas é o elevado consumo de energia elétrica, o principal fator que motivou a criação do *Proof-of-Stake (PoS)* [21]. A ideia principal do protocolo baseia-se no dono da moeda e na quantidade de moedas que o mesmo possui. A criação de um novo bloco, no *PoS*, é determinada por um sistema randômico que considera a quantidade de moedas que o usuário está possuindo no momento.

Por exemplo: dado um conjunto de usuário com permissão para validar a rede. Com João possuindo 5 moedas, Paulo 3 moedas e Pedro 2 moedas. A chance de criação de um bloco será de 50% para o João, 30% para o Paulo e 20% para o Pedro. Neste contexto, ao invés de despendar grande poder computacional, como no PoW, a criação do bloco é proporcional a quantidade de moedas que os usuários possuem. Sendo assim, esse tipo de protocolo apresenta as características do *PoW*, como: mineração, verificação das cadeias.

Delegated Proof-of-Stake (DpoS)

O protocolo de consenso *DpoS* se assemelha ao *PoS*, porém com algumas diferenças fundamentais na estrutura, no grau de influência e a escolha de um representante da rede, características do *DpoS*. Esse protocolo utiliza de uma estrutura que combina votação em tempo real com um sistema de reputação social para alcançar o consenso. Desta maneira, os usuários da rede que possuem um *token* da mesma, tem certo grau de influência na rede *blockchain*. Sendo assim, os usuários têm direito a votar para escolher um representante da rede que será responsável por fazer com que a rede funcione de maneira adequada e segura, buscando assim os melhores interesses da rede. Ou seja, qualquer usuário da rede que possuir um *token*, pode votar para um validador da rede. O validador com mais votos passa a ser o delegado que valida as transações e recebe recompensas por executar essa tarefa. Esse sistema com delegados torna a produção e validação dos blocos mais eficiente em relação ao *PoW*.

Proof of Capacity (PoC)

O protocolo de consenso *PoC* surge também como uma alternativa ao *PoW*, visando uma abordagem que diminua os custos computacionais sendo mais sustentável economicamente e ecologicamente. Esse algoritmo consiste em um sistema de dois passos: *plotting* e *mining*. Na etapa de *plotting* é criada uma solução randômica criptografada que é armazenada no *HD* do minerador. Na etapa de *mining*, os mineradores buscam a solução e o primeiro que encontrá-la, ganha o direito de minerar o próximo bloco. Ou seja, quanto mais espaço o minerador tiver para armazenar as soluções, maiores são as chances de minerar o bloco. Esse fator faz com que o protocolo economize energia em relação ao *PoW*, tornando-o a opção mais sustentável.

Proof of Burn (PoB)

O protocolo de consenso *PoB* é uma alternativa aos protocolos mais comuns. Nesse caso, os mineradores do *PoB* enviam *tokens*/moedas para um endereço desconhecido, fazendo com que os mesmos sejam tirados de circulação. Essas transações que inviabilizam a

utilização dessas moedas, são registradas na rede *blockchain*, garantindo que as mesmas não sejam gastas novamente e o usuário que executou essa tarefa seja recompensado. A ideia por trás desse protocolo é que o usuário mostre um comprometimento maior com a moeda, assumindo perdas a curto prazo em razão dos ganhos a longo prazo. Sendo assim, o usuário da rede *blockchain* que utiliza esse protocolo recebe incentivos para continuar nesse processo de inviabilização das moedas e assim recebendo cada vez mais recompensas.

2.2.7 Smart Contracts

Smart Contracts podem ser definidos como um protocolo de computador que executa os termos de um contrato, ou seja, traduzir as cláusulas para uma linguagem de programação e inseri-las na rede *blockchain* de forma que seja autoexecutáveis [22]. Os *smarts contracts* surgem para minimizar a necessidade de intermediários confiáveis entre as partes que executam as transações.

No contexto do *blockchain*, os *smart contracts* são *scripts* que são armazenados dentro da cadeia de blocos. Desta maneira, os contratos possuem endereços únicos e podem ser executados através de uma transação atrelada, sendo processados de modo independente e automático, seguindo as cláusulas e regras as quais foram programados [23].

O intuito dos *smart contracts* é utilizar das cláusulas de um contrato ou de regras de negócio para criar códigos autoexecutáveis. O exemplo a seguir demonstra o funcionamento dos *smart contracts*.

Considere uma rede *blockchain* com os participantes João e Pedro, e o ativo digital de moedas X. Nesse contexto, João implementa um *smart contract* na rede *blockchain*, com as seguintes cláusulas:

- Função *Deposito*, que permite que sejam feitos depósitos de unidades de X na conta;
- Função *Transferencia*, que permite que João receba uma unidade de X a cada transferência realizada entre contas;
- Função *Saque*, que permite João realize o saque de todas as unidades de X as quais possui, quando ocorrer uma transferência.

A partir dessas cláusulas é possível perceber que os *smart contracts* agem sobre os ativos da rede *blockchain* e permitem que sejam expressos as regras negociais em forma de linguagem de programação. Além disso, o exemplo mostra a necessidade de mensagens/transações para que as cláusulas sejam executadas. Desta maneira, os *smart contracts* podem ser aplicados em diversas áreas do conhecimento, por sua capacidade de traduzir as cláusulas de um contrato ou uma regra negocial em código.

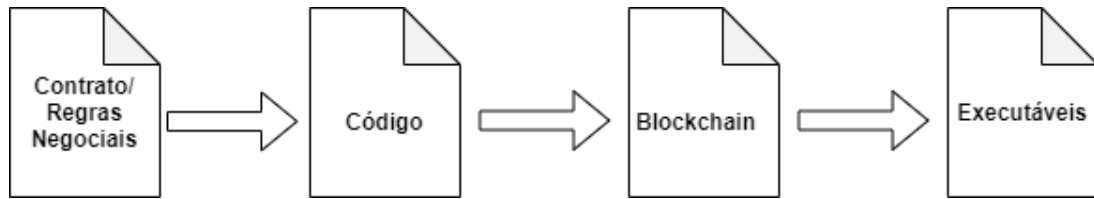


Figura 2.7: *Smart Contracts*.

A Figura 2.7 exemplifica a ideia dos *smart contracts*, a primeira etapa é entender e compreender as cláusulas de um contrato ou as regras negociais do projeto. A partir disso, é gerado o código, na linguagem de programação, que traduz os componentes da primeira etapa, que é o *smart contract* em si. Após a programação, os *smart contracts* são implantados na rede *blockchain*, com um endereço e assinatura específicos. Após essa etapa, os programas esperam uma mensagem ou transação para serem executados.

Os *smart contracts* estão sendo utilizados em aplicações de sucesso como na Internet da coisas [23], cadeia de suplementos [24], direitos autorias [25], entre outros. Isto permite que a utilização dos contratos englobe diversos domínios do conhecimento.

2.2.8 Considerações

O *blockchain* está amplamente ligado às criptomoedas, sendo seu uso mais conhecido e melhor empregado até hoje é no *Bitcoin*. Essa tecnologia apresenta vantagens, sendo assim é possível utilizá-las em outros domínios. Deixando o âmbito de criptomoedas, há inúmeras aplicações que podem usufruir das vantagens do *blockchain*, que são:

- Maior segurança aos dados;
- Maior autenticidade dos usuários;
- Maior transparência das informações;
- Melhor acesso as informações;
- Melhor eficiência no compartilhamento das informações;

Apesar de todas essas vantagens, um dos maiores desafios do *blockchain* é adaptar essa tecnologia para outras aplicações, mostrando uma rede confiável para os usuários envolvidos e interessados nos projetos que utilizarão a tecnologia *blockchain* [26, 27, 28].

Capítulo 3

Trabalhos Relacionados

Neste capítulo, serão abordados os trabalhos que possuem uma relação com os temas: *blockchain* e sistemas de aviação. Desta maneira, serão tratados os aspectos de projetos que envolvam a aplicação do *blockchain* distante do âmbito das criptomoedas e da utilização dessa tecnologia no sistema aéreo.

3.1 Aplicações Blockchain

Wust et al. [1] propôs uma análise crítica sobre a utilização do *blockchain* como solução em aplicações com requisitos específicos. A metodologia aplicada consiste no estudo das propriedades de diferentes tipos de *blockchain*, seguido por identificar qual melhor se encaixa dentro dos requisitos do problema. Os autores propõem um questionário para ser respondido pelos requisitos do problema, através do mesmo é possível identificar se o *blockchain* é uma solução viável ou não e qual tipo melhor se adapta aquele problema. O artigo aborda três estudos de casos diferentes: gerenciamento de cadeia de suplementos, pagamentos bancários e organizações autônomas descentralizadas. Neste contexto, é possível perceber as diferenças de cada um dos casos, observando os requisitos envolvidos das aplicações. O texto apresenta uma estrutura metodológica que considera os requisitos da aplicação, os participantes envolvidos e características técnicas, para assim decidir sobre a utilização do *blockchain*.

Lone et al. [29] propôs a utilização *blockchain* para tratar das informações de evidências digitais de crimes para garantir integridade, autenticidade e auditabilidade. A metodologia aplicada consiste na utilização de uma camada de aplicação do *blockchain* através do *Hyperledger Composer*¹. Os autores propõem adaptar as evidências digitais para o contexto de dados do *blockchain*, sendo assim, a ideia é fazer todo o processo que ocorre nos dias atuais através de uma rede *blockchain*, desde a criação da evidencia,

¹<https://www.hyperledger.org/projects/composer>

inserção no bloco, possíveis exclusões e visualizações das mesmas. O artigo traz uma abordagem de testes realizados, mostrando os melhores aspectos que o *Hyperledger Composer* pode gerar na aplicação, como a latência e a taxa de vazão. O texto consegue apresentar os conceitos de evidência digital e forense, trazendo-os para o contexto do *blockchain*, abordando suas principais características em prol da segurança, autenticidade e integridade dos dados. Porém ainda é uma Prova de Conceito, sendo necessário uma elaboração mais detalhada do desenvolvimento da aplicação e nos dados que serão utilizados.

Jiang et al. [12] propôs a criação de um sistema baseado em *blockchain* para troca de informações de saúde. A metodologia empregada consiste na criação de dois sistemas integrados de *blockchain* com dados de saúde de diferentes fontes, utilizando de uma combinação de *off-chain* para armazenamento e *on-chain* para verificação dos dados e autenticidade. Os dados para a realização do projeto são divididos em dois: dados de instituições médicas e dados individuais de pessoas, obtidos por meio de dispositivos de saúde e de *IoT*. O artigo sugere a criação de dois algoritmos para a rede *blockchain*, um para verificar a vazão dos dados e outro para que todos os membros da rede tenham a chance de criar um bloco, através desses dois algoritmos conseguiu se observar vantagens e melhorias do que de escolhas aleatórias para a montagem e preenchimento dos blocos. O artigo propôs a utilização da rede *blockchain* para a troca de informações voltadas para área de saúde, demonstrando como é feito a implementação e as melhorias que sua utilização pode proporcionar, porém poderia ficar evidente como que ocorre a implementação e a explicação do sistema [12].

Outchakoucht et al. [14] propôs a criação de um *framework* para controle e acesso de dados no contexto da *Internet das Coisas*. A metodologia adotada foi a criação de um *framework* baseado em *blockchain* para melhorar a política de segurança, utilizando em conjunto com algoritmos de aprendizagem de máquina. Os autores propõem a utilização do *blockchain* para fazer essa política de acesso às informações, empregando os *smart contracts* para verificar se o usuário pode acessar ou não a informação. Após essa fase, é apresentado o algoritmo de aprendizagem em reforço, treinado com o *feedback* dos *smart contracts*. Em vista da enorme quantidade de dispositivos de *IoT*, tem-se a ideia de ter políticas para acesso as informações cada vez mais seguras, entretanto, a implementação não está clara e não a incidências de resultados [14].

Dinh et al. [30] propôs uma junção entre duas tecnologias disruptivas: o *blockchain* e a inteligência artificial (IA). A metodologia consiste em utilizar o *blockchain* para suprir as desvantagens da inteligência artificial e vice-versa, potencializando ao máximo as vantagens e pontos fortes das duas tecnologias. São apresentadas possíveis utilizações do *blockchain* para a IA, como melhorar o compartilhamento de informações, tornando-o mais seguro. Outro ponto é da IA para o *blockchain*, como *blockchains* mais seguros, mais

escaláveis e mais automatizados. O artigo apresenta como as duas abordagens podem funcionar em conjunto, alavancando a utilização do *blockchain* e da IA, principalmente da aprendizagem de máquina. No entanto, o artigo é bastante teórico, fornecendo poucos exemplos práticos para a viabilização da estratégia [30].

3.2 Blockchain no Sistema Aéreo

Dehez et al. [31] propôs a utilização da tecnologia *blockchain* para o gerenciamento de tráfego aéreo. A metodologia aplicada consiste em utilizar dos conceitos presentes na tecnologia *blockchain* para o compartilhamento de informações ATM, mais especificamente plano de voo. Os autores propõem uma descrição em alto nível para explicar sua metodologia, na qual utilizam dos planos de voo para demonstrar uma possível utilização dessa tecnologia no ambiente de tráfego aéreo. Sendo assim, o artigo apresenta boas perspectivas para sua utilização e implementação no futuro.

Reisman et al. [32] propôs a utilização do *blockchain* para promover uma melhora na segurança dos dados e informações do sistema *Automatic dependent surveillance-broadcast (ADS-B)*. A metodologia aplicada consiste no desenvolvimento de uma camada de aplicação do *blockchain* através do *framework* do *Hyperledger Fabric*. O autor propõe uma nova abordagem com a utilização do *blockchain* para proporcionar uma maior privacidade e anonimato enquanto promove uma maior segurança e eficiência nas comunicações do tráfego aéreo. O artigo aborda os conceitos de aviação e gerenciamento de tráfego, assim como os conceitos do *blockchain*, propondo a criação de uma infraestrutura *blockchain* para auxiliar as comunicações entre as aeronaves e as bases de operações terrestres. O texto aborda melhorar a segurança, a eficiência e a privacidade das comunicações do gerenciamento de tráfego aéreo. Entretanto, não ficou claro como que será feita essa implementação, além de outros aspectos e características importantes relacionadas a tecnologia *blockchain*.

Bonomo et al [33] propôs a utilização do *blockchain* para a criação dos serviços do SWIM Registry Brazil. A metodologia aplicada consiste em adaptar a estrutura básica do *blockchain* para que esse possa receber os serviços disponíveis do SWIM Registry. O autor propõe uma nova abordagem com a utilização do *blockchain* para proporcionar uma maior segurança as informações e maior autenticidade aos usuários. O artigo aborda os conceitos de *blockchain*, aviação, SWIM e SWIM Registry, assim como a modelagem da estrutura básica do *blockchain* para receber as informações do serviço de plano de voo.

3.3 Considerações

Após a leitura desses trabalhos, percebe-se que muitas pesquisas estão propondo aplicações utilizando *blockchain* em diversas áreas do conhecimento, porém ainda em um âmbito mais conceitual. Além disso, estão surgindo diversos estudos de aplicações *blockchain* no transporte aéreo.

Neste contexto, é possível observar inúmeras possibilidades para se trabalhar em conjunto com a tecnologia *blockchain* no sistema aéreo, mais precisamente no *SWIM*. A integração dessas metodologias pode produzir oportunidades de inovação e resolver algumas limitações do *SWIM*, gerando benefícios que podem ir além da área da aviação e abrangendo outras áreas do conhecimento.

Capítulo 4

Metodologia Proposta

Neste capítulo serão apresentados conceitos gerais da aplicação *blockchain* para o serviço de plano de voo do *SWIM Registry Brazil* como: definição da arquitetura, fonte de dados, organização dos componentes, relação entre os mesmos, etc. Os conceitos da solução serão detalhados de acordo com suas modelagens e implementações.

A solução proposta está dividida na seguintes seções:

- **Arquitetura Proposta:** seção que detalha a arquitetura proposta para a implementação do *blockchain* no *SWIM Registry Brazil*;
- **Fonte de Dados:** seção que demonstra o levantamento dos dados de plano de voo, como suas características e informações pertinentes, além da modelagem dos dados para a estrutura básica do *blockchain*;
- **Implementação:** seção que demonstra a implementação da arquitetura proposta, assim como toda a estrutura utilizada de frameworks e as adaptações do plano de voo para o *blockchain*.

4.1 Arquitetura Proposta

A arquitetura proposta pode ser dividida em duas etapas: a primeira consiste na identificação e escolha de um serviço do *SWIM Registry Brazil*, e a segunda consiste na implementação desse serviço em uma aplicação *blockchain*. A solução proposta segue os seguintes passos:

- **Fonte de dados:** estudo dos dados do serviço de plano de voo;
- **Modelagem dos dados:** adaptação dos dados para o *blockchain*;
- **Implementação do blockchain:** implementação de uma camada de aplicação *blockchain* para o serviço de plano de voo;

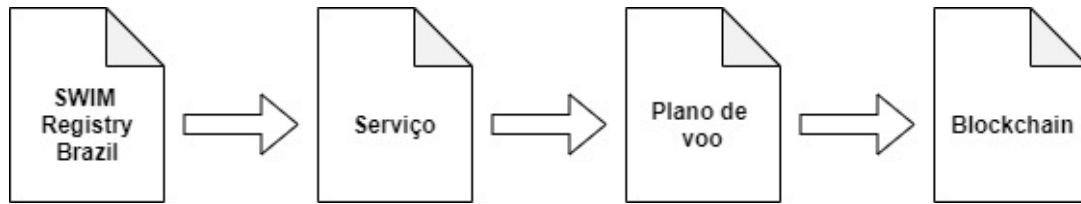


Figura 4.1: Primeira etapa da arquitetura proposta.

A Figura 4.1 representa a primeira parte da arquitetura proposta, na qual através de um estudo do *SWIM Registry Brazil*, foi escolhido o serviço de plano de voo para sua implementação no *blockchain*.

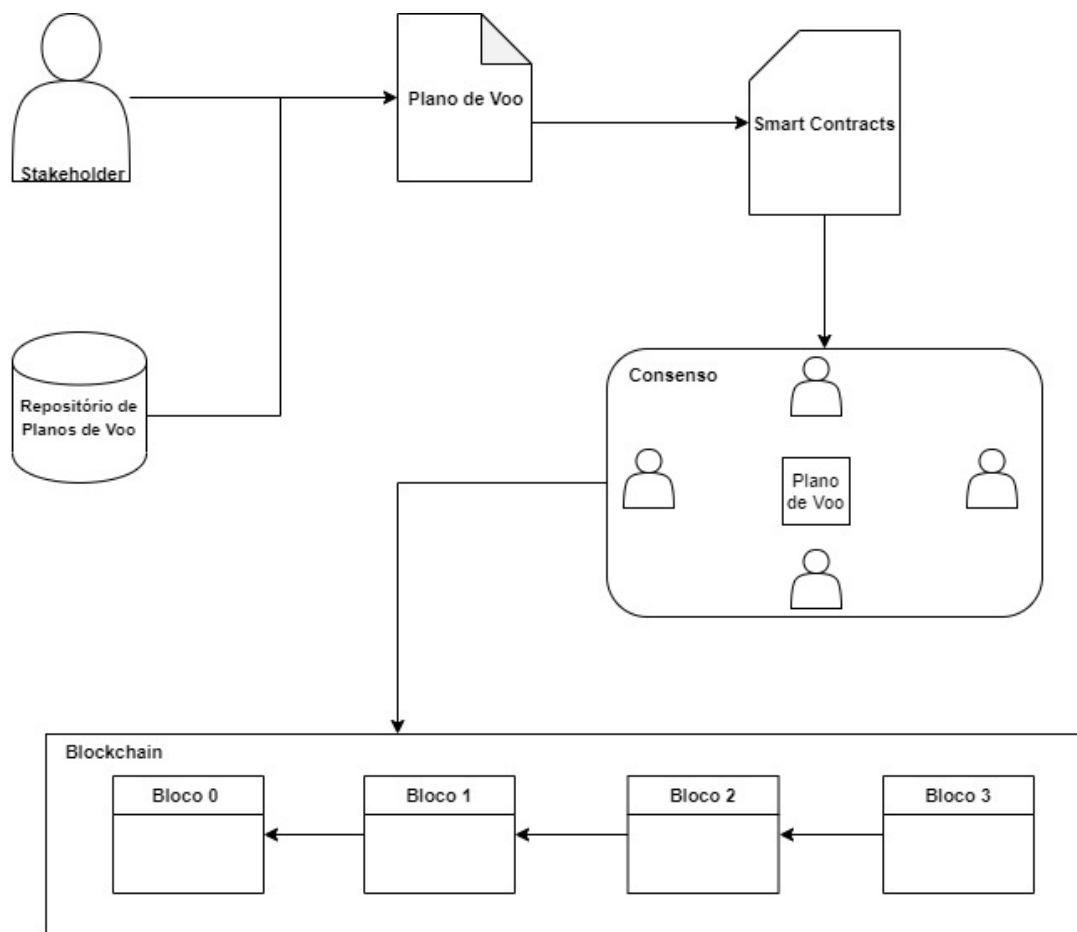


Figura 4.2: Arquitetura proposta para implementação do serviço de plano de voo.

A Figura 4.2 representa a segunda etapa da arquitetura, uma visão geral do sistema, em que os dados provém de um repositório de planos de voo ou de algum *stakeholder*, no caso pilotos de companhias aéreas. Desta maneira, os planos de voo são modelados para a abordagem *blockchain*, respeitando as características da estrutura. Após esse processo de modelagem, os dados são verificados pelas regras negociais do sistema contidas nos

smart contracts para em seguida serem validadas pelo protocolo de consenso. Logo após, os planos de voo são inseridos na rede *blockchain* do *SWIM Registry Brazil*.

4.1.1 Justificativa

Com intuito de determinar se o *blockchain* é uma solução viável para o serviço de plano de voo do *SWIM Registry Brazil*, foi utilizado um questionário proposto por *Wust e Gervais* [1] referente aos requisitos do sistema.

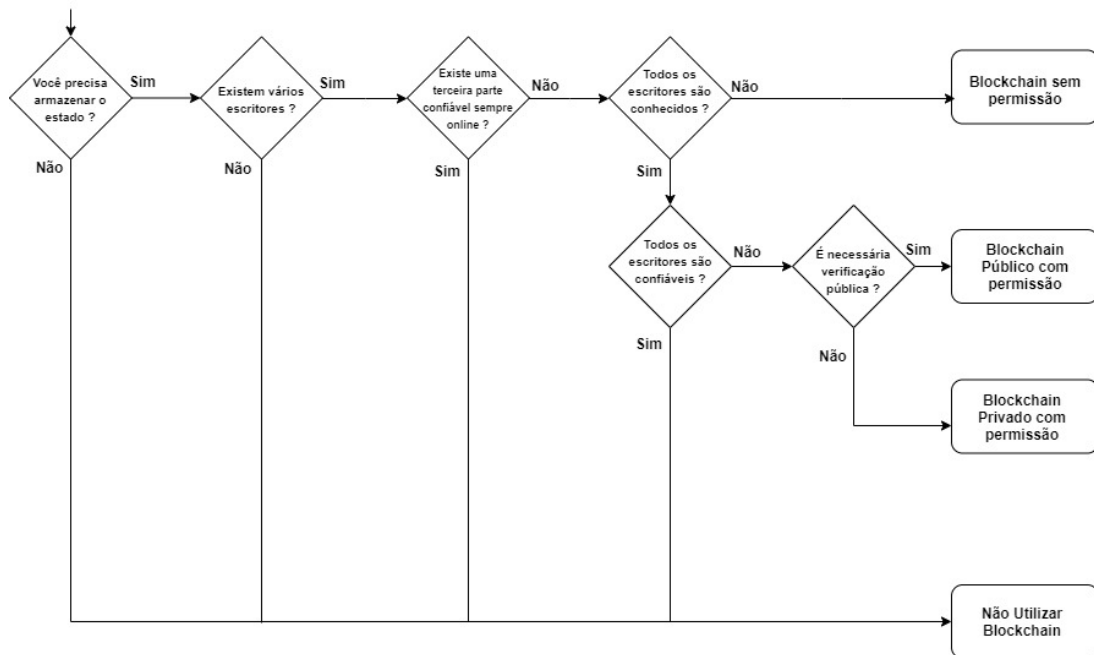


Figura 4.3: Questionário adaptado de Wust [1].

De acordo com a Figura 4.3, as questões do questionário são:

- Q1: Você precisa armazenar o estado?
- Q2: Existem vários escritores?
- Q3: Existe uma terceira parte confiável sempre online?
- Q4: Todos os escritores são conhecidos?
- Q5: Todos os escritores são confiáveis?
- Q6: É necessária verificação pública?

Nesse contexto, a resposta da Q1 é sim, porque os planos de voo são analisados por diferentes usuários do sistema, com o intuito de verificá-los e assim validá-los, sendo assim é necessário armazenar os estados dos planos de voo. A resposta para o questionamento

Q2 é sim, porque existem vários *stakeholders* presentes no *SWIM Registry*. A resposta para o questionamento Q3 é não devido ao fato de querer eliminar o ponto central de falha e confiança. A resposta para o questionamento Q4 é sim, porque somente usuários cadastrados e autenticados têm acesso ao sistema para ler e escrever. A resposta para o questionamento Q5 é sim, porque não se pode afirmar que todos os usuários envolvidos no sistema são confiáveis. Finalmente, a resposta da Q6 é não, visto que não se tem a necessidade de uma verificação pública do sistema. Sendo assim, com base no questionário, pode-se concluir que a utilização de um sistema de *blockchain* privado com permissão é viável para o contexto do *SWIM Registry Brazil*.

4.2 Fonte de dados

Após um estudo aprofundado do *SWIM Registry Brazil* e de seus serviços, foi escolhido o serviço de plano de voo para ser implementado em uma aplicação *blockchain*, visto que era um dos serviços que estavam em processo mais avançado de desenvolvimento no *SWIM Registry Brazil*.

4.2.1 Plano de Voo

O piloto é responsável pelo preenchimento detalhado do plano de voo. Logo após, esse documento é validado pelas autoridades e depois armazenado no Centro de Gerenciamento de Navegação Aérea (CGNA)¹. As informações de plano de voo são atualizadas diariamente e os dados podem ser filtrados de acordo com alguns critérios: data, companhia aérea, localidades etc.

Os planos de voo realizados e armazenados no site, estão disponíveis publicamente para consulta no repositório da instituição. Sendo assim, é possível visualizar, transferir e utilizar esses dados. As informações presentes nos planos de voo são:

- **CIA:** Companhia Aérea responsável;
- **Início de Validade:** Data que contém o início da validade;
- **Valido Até:** Data que contém o fim da validade;
- **Dias OP:** Dias da semana que o plano de voo está em operação;
- **Ident Anv:** Identificador da aeronave;
- **Tipo Turb:** Tipo de turbina da aeronave;

¹<http://portal.cgna.gov.br/>

- **Adep Eobt:** Identificador do aeródromo de partida e da hora da decolagem;
- **Vel:** Velocidade da aeronave;
- **FL:** Identificador do nível de voo da aeronave;
- **Rota:** Identificador da rota que aeronave percorrerá;
- **Dest Eet:** Identificador do aeródromo de destino e da hora do pouso;
- **Observacoes:** Outras informações importantes para o voo.

Planos de Voo Repetitivos
Classificação: Ident Anv

INÍCIO DE VALIDADE: 28Out2019

PAG.: 1

CIA: TAM	VALIDO DESDE	VALIDO ATE	DIAS STQSSD	OP ANV	IDENT ANV	TIPO TURB	ADEP E0BT	VEL	FL	ROTA	DEST EET	OBSERVACOES
	161019	280320	1234567	TAM3000	A320/M	SBCT1820	N0450	270	DCT	EDVEB UM548 ERODU DCT	SBSP0055	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV RMK/TCAS PBN NO B6
	181019	280320	1234500	TAM3001	A320/M	SBSP0940	N0450	260	DCT	SOVSI DCT	SBCT0050	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV RMK/TCAS PBN NO B6
	281019	280320	1234500	TAM3002	A320/M	SBVT0900	N0450	340	MABSI	UZ44 PAGOG	SBSP0130	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV EET/SBCH0015 RMK/TCAS PBN NO B6
	271019	280320	0000007	TAM3002	A319/M	SBVT0900	N0450	340	MABSI	UZ44 PAGOG	SBSP0130	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV EET/SBCH0015 RMK/TCAS PBN NO B6
	271019	280320	0000060	TAM3002	A319/M	SBVT0915	N0450	340	MABSI	UZ44 PAGOG	SBSP0130	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV EET/SBCH0015 RMK/TCAS PBN NO B6
	171019	280320	1030000	TAM3003	A321/M	SBCT0005	N0450	290	DCT	KOXAG UM671 ANISE DCT	SBGR0100	EQPT/SDE2FGHI1RWXYZ PBN/A1B1C1D1L102S2 DAT/SV RMK/TCAS PBN NO B6
	171019	280320	0204560	TAM3003	A320/M	SBCT0005	N0450	290	DCT	KOXAG UM671 ANISE DCT	SBGR0100	EQPT/SDE2FGHI1RWXYZ PBN/A1B1C1D1L102S2 DAT/SV RMK/TCAS PBN NO B6
	161019	280320	1234560	TAM3005	A319/M	SBSP0930	N0450	320	GERTU	UZ21	SBSR0105	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV EET/SBBS0021 RMK/TCAS PBN NO B6
	161019	280320	1234560	TAM3006	A319/M	SBSR1115	N0450	250	DCT	ENPEG/N0450F260 UZ5 ANSOK/N0450F250 UZ42 UGOVU	SBSP0050	EQPT/SDE2FGHI1RWXYZ STS/ATFMX PBN/A1B1C1D1L102S2 DAT/SV EET/SBCH0014 SBBS0032

Figura 4.4: Consulta de planos de voo do CGNA.

A Figura 4.4 mostra uma consulta realizada ao site do CGNA que traz as informações de planos de voo como a companhia aérea que está operando, o identificador da aeronave que utilizou o plano de voo, assim como descrição de rota que a aeronave percorreu.

4.2.2 Modelagem dos Dados

Com intuito de desenvolver uma camada de aplicação *blockchain* para o *SWIM Registry Brazil*, é necessário uma modelagem dos dados de plano de voo. Neste contexto, em conjunto com as características básicas do *blockchain*, foi proposto uma modificação no campo de dados (*data*) do *blockchain* para armazenar as informações dos planos de voo. Dessa maneira, permitindo que o bloco mantivesse a mesma estrutura com essas modificações.

Block 1
Index: 1
Timestamp: 18/04/2018 00:31
Hash: 000fd8ac0791ec07bb...
PreviousHash: 00be30f5201...
InicioValidade: 16/04/2018
Cia: TAM
ValidoDesde: 18/02/2018
ValidoAte: 20/10/2018
DiasOP: 1234567
IdentAnv: TAM 3000
TipoTurb: A320/M
AdepEobt: SBTC1825
Vel: N0450
FI: 270
Rota: EDVEB UM 548 ERODU
DestEet: SBSP0055
Observacoes: EQPT/SDE2F

Figura 4.5: Alterações propostas no campo *data*.

A Figura 4.5 representa essa alteração no campo *data*, sendo possível perceber as inserções de informações do plano de voo dentro do bloco e a preservação dos campos básicos, como: *index*, *timestamp*, *hash* do bloco anterior e *hash*.

4.3 Implementação

Nesta seção será descrito os *frameworks* utilizados na implementação, assim como as mudanças feitas para adaptação do plano de voo para o *blockchain*.

4.3.1 *Hyperledger Composer*

O *Hyperledger Composer*² é um *framework* de código aberto com intuito de facilitar a programação de aplicações *blockchain*. A ferramenta tem como objetivo acelerar o tempo de desenvolvimento e simplificar a integração do *blockchain* com as regras negociais do sistema. O *Hyperledger Composer* suporta toda a infraestrutura dos protocolos de consenso para garantir que as transações sejam validadas de acordo com a política decidida pelos participantes do sistema.

Como parte da criação da aplicação utilizando esse *framework*, é preciso definir alguns aspectos de acordo com as regras negociais do sistema, que são:

²<https://www.hyperledger.org/projects/composer>

- **Ativos:** Podem ser caracterizados como bens tangíveis ou intangíveis, serviços ou propriedade. Devem conter um identificador único e as propriedades escolhidas de acordo com o sistema. Além disso, é possível relacioná-los com outros ativos ou participantes.
- **Participantes:** São membros do sistema que podem possuir os ativos e executar transações. Os participantes são modelados de acordo com as propriedades e requisitos do sistema.
- **Transações:** É o mecanismo pelo qual os participantes interagem com os ativos. Além disso, as transações são desenvolvidas de acordo com as regras negociais do sistema.

Uma parte importante do *Hyperledger Composer* é o controle de acesso e autorização, além de uma arquitetura de segurança compartilhada pelos membros envolvidos no *blockchain*. Desta maneira, o administrador do sistema determina quais recursos ou dados os participantes têm autorização para visualizar ou processar. Sendo assim, as operações e transações do sistema que são realizadas pelos agentes interessados seguirão os requerimentos do controle de acesso que é armazenado no *blockchain*. As regras do controle de acesso são divididas em duas áreas:

- Autorização para acesso ao sistema, rede ou recursos administrativos;
- Autorização para acesso aos ativos e operações do sistema.

Pré-requisitos

A utilização do *Hyperledger Composer* requer a instalação de ferramentas de desenvolvimento como pré-requisito que são:

- **Docker Engine:** É uma tecnologia de contêiner de código aberto combinada com um fluxo de trabalho para a criação e armazenamento de aplicações. Sendo necessário a versão 17.03 ou superior.
- **Docker-Composer:** É uma ferramenta para definir e executar aplicações Docker de múltiplos contêineres. Sendo necessário a versão 1.8 ou superior.
- **Node:** É um interpretador de JavaScript assíncrono de código aberto. Sendo necessário a versão 8.9 ou superior.
- **Npm:** É um gerenciador de pacotes para a linguagem de programação JavaScript. Sendo necessário a versão v5.x.

4.3.2 Arquitetura Hyperledger Composer

O *Hyperledger Composer* possibilita o desenvolvimento de soluções *blockchain*, criando a lógica de negócio que é executada no *blockchain*, uma *API REST* que expõe a lógica *blockchain* para aplicações *web* e *mobile*. Desta maneira, a arquitetura do *Hyperledger* possui componentes que permitem todo esse processo de criação da aplicação que são:

- **Yeoman:** é uma ferramenta *open source* que possibilita a criação de uma estrutura do projeto em forma de linguagem de programação, como aplicações *angular* e *Node.js*³;
- **Angular:** é um *framework open source* para desenvolvimento de aplicações baseadas em *JavaScript*;
- **Rest Server:** cria automaticamente uma aplicação *REST* do modelo de negócios, possibilitando a implementação das funções *Criar*, *Ler*, *Atualizar* e *Excluir* dos ativos e participantes, além de permitir que as transações sejam enviadas para processamento;

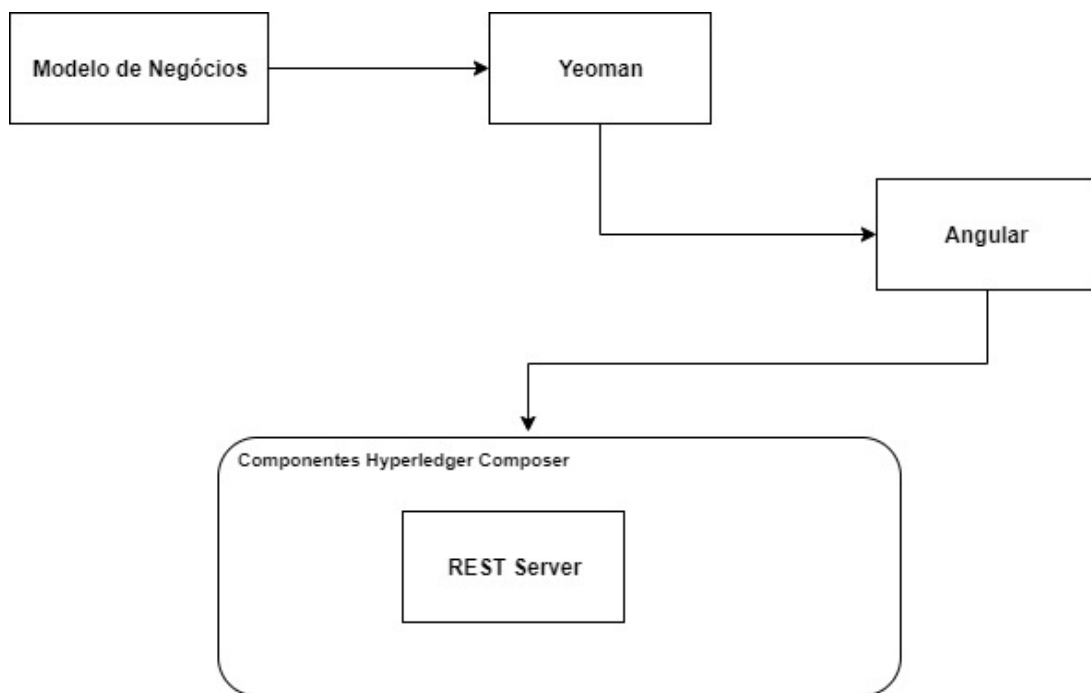


Figura 4.6: Arquitetura do Hyperledger Composer.

A Figura 4.6 apresenta a arquitetura do *Hyperledger Composer*. A partir do modelo de negócios, o *Yeoman* gera uma estrutura de projeto em *Angular*, na qual essa aplicação utiliza dos recursos da *API REST* criada pelo *Rest Server*.

³<https://nodejs.org/en/>

Protocolo de Consenso no Hyperledger Composer

O *Hyperledger Composer* pode utilizar diferentes tipos de protocolo de consenso, dependendo apenas de sua configuração. Desta maneira, é aplicado o protocolo configurado como padrão. Este protocolo de consenso é o *KAFKA*, que consiste em ser um abordagem com permissão baseada em um sistema de votação, no qual o líder executa uma etapa do processo a de *ordering*. Neste contexto, apenas réplicas sincronizadas podem ser escolhidas como líder. Essa abordagem promove a tolerância a falha de colisão, entretanto não é tolerante a falha bizantina.

O *KAFKA* pode ser dividido em três etapas:

- **Endorsement:** etapa que é orientada por políticas definidas pelo sistema, na qual os participantes podem assegurar uma transação;
- **Ordering:** etapa que aceita as transações endossadas e concorda com a ordem de publicá-las na *blockchain*;
- **Validation:** etapa que verifica o bloco de transações ordenadas e valida a correteza dos resultados, incluindo a política de endossamento e gastos duplos.

Organizações

No *Hyperledger* uma organização corresponde a um grupo de membros gerenciados, que pode ser uma companhia, um grupo, um *stakeholder*. Desta maneira, cada organização dispõe de uma chave pública que compreende um Certificado de Autoridade (CA) que serve para identificar digitalmente as identidades das organizações e autenticar as mesmas durante as trocas de mensagem na rede. Outro aspecto importante das organizações é o *Membership Service Provider (MSP)* que serve para identificar as mesmas. Neste contexto, o *MSP* identifica os certificados de cada organização e verifica se são confiáveis, garantindo assim que aquele participante faz parte da rede. Além disso, o *MSP* identifica papéis específicos que uma organização pode ter e define os privilégios de acesso na rede.

Peer

A rede *blockchain* é compreendida por um conjunto de nós (Peer), que é o elemento fundamental da rede por conter os *ledgers* e os *smart contracts*. Desta maneira, cada organização pode possuir um ou mais *Peers*, de acordo com a configuração desejada para o *blockchain*. No Hyperledger os *Peers* podem ser criados, inicializados, parados, reconfigurados e excluídos.

4.3.3 *Hyperledger Composer no SWIM*

Dentre os aspectos presentes do *Hyperledger Composer*, as informações do serviço de plano de voo precisam ser modeladas para atender os requisitos. Dessa forma foram criados os ativos, os participantes e as transações.

Ativo de Plano de Voo

Plano Voo	Estados
String - Id	String - Intenção
String - Ident	String - Rejeitada
String - Acft	String - Ajustes
String - Partida	String - Confirmada
String - Vel	String - Realizada
String - Fi	
String - Rota	
String - Destino	
String - horaPartida	
String - horaChegada	
Estado - EstadoPlano	
Stakeholder - membro	

Figura 4.7: Ativo de Plano de Voo.

A Figura 4.7 apresenta os campos presentes no ativo *Plano Voo*, de acordo com a fonte de dados dos Planos de Voo. Desta maneira, o ativo foi modelado com o campo de texto contendo um identificador único *Id*, os campos de texto com as informações importantes do plano de voo: *Ident*, *Acft*, *Partida*, *Vel*, *Fi*, *Rota*, *Destino*, *HoraPartida*, *HoraChegada*. Além disso, o campo Estado que contém as informações sobre o Plano de Voo, *Intencao*, *Rejeitada*, *Ajustes*, *Confirmada*, *Realizada*. Por último, apresenta o campo *Stakeholder* que contém as informações do participante que está de posse do ativo Plano de Voo.

Fluxograma dos Estados do Plano de Voo

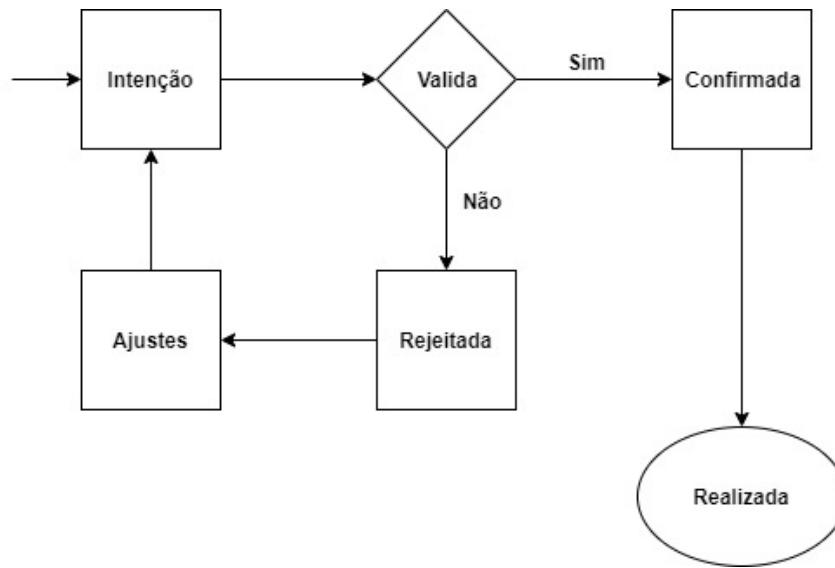


Figura 4.8: Fluxograma dos Estados de Plano de Voo.

A Figura 4.8 representa o fluxograma de Estados do ativo de Plano de Voo. Desta maneira, quando um plano de voo é preenchido e enviado pelo piloto, o mesmo recebe o estado de *Intenção*. Logo após, ocorre uma validação para verificar as informações, se as mesmas estiverem válidas, o estado muda para *Confirmada* e assim após a realização do voo é alterado para *Realizada*. Se as informações não estiverem válidas, o plano de voo recebe o estado *Rejeitada*, sendo necessários ajustes na intenção de voo. Desta maneira, o estado é modificado para *Ajustes* quando o piloto é notificado e assim que o mesmo comece a resolver os ajustes necessários o estado é alterado para *Intenção* de novo, recomeçando o fluxograma.

Participante Stakeholder

Stakeholder	Tipo
String - StakeId	String - Companhia Aerea
String - Nome	String - Agencia
Tipo - Tipo	String - Governo
String - Descricao	

Figura 4.9: Participantes - Stakeholder.

A Figura 4.9 apresenta os participantes do *blockchain*, sendo descritos como *Stakeholder* do sistema. Desta maneira, os atributos presentes são um campo de texto que contém um identificador único do participante o *StakeId*, os campos de texto contendo o *nome* e a *descrição* do participante. Além disso, um campo *Tipo* com as informações da categoria do participante: *Companhia Aerea*, *Agencia*, *Governo*.

Transação validarPlano

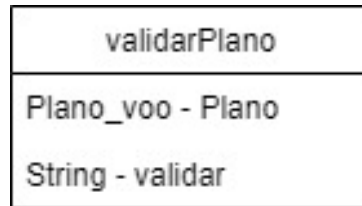


Figura 4.10: Transação - Validar Plano.

A Figura 4.10 apresenta a transação do *blockchain*, com o campo de *Plano Voo* que contém o identificador do plano de voo e um campo de texto *Validar* para realizar a validação do plano de voo em questão. Essa transação é representada pelo algoritmo a seguir.

Algorithm 1: Validar Plano de Voo

Data: Plano Voo[Id]

Result: Validação do plano de voo

```

1 if Plano Voo[Id] != null then
2   | if Atributos(Plano Voo[Id]) = validos then
3   |   | return Sim
4   | else
5   |   | return Nao
6 else
7   | return False

```

O algoritmo descreve o procedimento de validar um plano de voo, no qual o *stakeholder*, com as permissões adequadas de avaliação, recebe o plano de voo do *Id* determinado e verifica os atributos do mesmo. Após esse procedimento, o membro responsável decide se o plano de voo é válido.

Algoritmos

Com a definição dos ativos, das transações e dos participantes, o modelo compreende funções básicas que são criação, exclusão, listagem dos planos de voo dentro do *blockchain*.

- **Criação do Plano de Voo:** Essa função recebe como entrada o *Id* do plano de voo e seus atributos pertinentes. Desta maneira, o *Estado* do plano de Voo recebe o valor de *Intencao* e o *Stakholder* o valor do *hash* do criador do plano de voo. Com essas informações o plano de voo é criado e inserido na *blockchain*. O algoritmo a seguir representa essa função.

Algorithm 2: Criação do Plano de Voo

Data: Plano Voo[Id], atributos

Result: Insere um plano de voo no bloco

```

1 if Plano Voo[Id] != null then
2   | return False
3 else
4   | Estado = Intencao;
5   | Stakholder = hash do criador;
6   | Insere no blockchain (Plano Voo[Id], Atributos, Estado, Stakeholder)

```

- **Listagem do Plano de Voo:** Essa função retorna uma lista com todos os planos de voo e as informações relacionadas aos mesmos. O algoritmo a seguir representa essa função.

Algorithm 3: Lista Planos de Voo

Data:

Result: Lista de Plano de Voo

```

1 if Plano Voo != null then
2   | return Lista de Plano de Voo
3 else
4   | return False

```

- **Exclusão do Plano de Voo:** Essa função recebe como entrada o *Id* do plano de voo e realiza a exclusão do mesmo. A exclusão não ocorre fisicamente, e sim da consulta de outros *stakeholders* do sistema. Sendo assim, aqueles usuários com as permissões adequadas podem consultar esses planos de voo que foram excluídos. O algoritmo a seguir representa essa função.

Algorithm 4: Exclusão do Plano de Voo

Data: Plano Voo[Id]

Result: Remove Plano de Voo da Blockchain

```

1 if Plano Voo[Id] != null then
2   | Remove o Plano Voo[ID]
3 else
4   | return False

```

- **Consultar Plano de Voo:** Essa função recebe como entrada o *Id* do plano de voo e realiza uma consulta dos atributos do mesmo. Os usuários com as permissões adequadas podem consultar esses planos de voo. O algoritmo a seguir representa essa função.

Algorithm 5: Consultar Plano de Voo

Data: Plano Voo[Id]
Result: Consulta os dados do Plano de Voo da Blockchain

```

1 if Plano Voo[Id] != null then
2   | Atributos do Plano Voo[ID]
3 else
4   | return False

```

Essas funções são condicionadas aos usuários, isto é, os membros com acesso ao sistema e com as devidas permissões, podem executar as funcionalidades da aplicação.

4.3.4 Hyperledger Caliper

O *Hyperledger Caliper* é uma ferramenta de *benchmark* do *blockchain* que permite usuários mensurar a performance de uma implementação específica de um *blockchain* baseado em casos de teste predefinidos. Esse *framework* produz relatórios com os indicadores de performance escolhidos, como: transações por segundo, latência, utilização de recurso entre outros. O objetivo do *Caliper* é produzir resultados para serem usados como suporte a escolha de uma implementação *blockchain*.

O *Hyperledger Caliper* possui características chave que são:

- Um *benchmark* unificado da rede *blockchain* que permite realizar uma comparação com outras aplicações e que facilite a realização de diferentes testes na aplicação;
- Uma definição de indicadores de performance comum, que permite a comparação com outras aplicações *blockchain*;
- Conjunto de casos de teste, que permite a comparação de cada solução *blockchain* em diferentes cenários possíveis.

Arquitetura do Hyperledger Caliper

A arquitetura do *Hyperledger Caliper* contém três camadas: Camada de *Benchmark*, Camada de Interface e Camada de Adaptação. Cada uma apresenta características importantes para facilitar a execução dos testes para medir a performance do *blockchain*. A Camada de *Benchmark* contém um conjunto de teste e casos de uso implementados para

cenários do *blockchain*. Desta maneira, cada teste possui sua própria configuração na qual são definidos os argumentos de testes e o *blockchain*. A ferramenta disponibiliza um *benchmark* padrão para ajudar os desenvolvedores a entender o *Caliper* e a programarem os próprios casos de teste. A Camada de Interface implementa as funções básicas e provém interfaces para as aplicações, que são:

- **Interface de operação do *blockchain*:** Contém as operações do *blockchain*, como implantação dos *smart contracts*, utilização dos mesmos e consultas de estados do *blockchain*;
- **Monitor de Recursos:** Contém as operações de monitoramento de recursos do sistema *blockchain*, incluindo a utilização da *CPU*, memória e operações de entrada e saída;
- **Analizador de Performance:** Contém operações para analisar estatísticas de performance predefinidas e imprimir os resultados obtidos, de acordo com as métricas escolhidas;
- **Gerador de Relatórios:** Contém operações para criar um relatório de teste em formato *HTML*.

A Camada de Adaptação é utilizada para realizar a integração dos sistemas *blockchain* existentes com a ferramenta do *Hyperledger Caliper*. Desta maneira, cada implementação *blockchain* possui um adaptador específico para realizar a integração.

A Figura 4.11 apresenta a arquitetura do *Hyperledger Caliper*, desta maneira é possível perceber a Camada de *Benchmark* com os casos de usos que serão testados. Logo abaixo, a camada de Interface com as funções que são utilizadas para a realização dos testes e em seguida a Camada de Adaptação contendo os adaptadores para as possíveis aplicações *blockchain* utilizarem o *Caliper*.

Benchmark

Dentro da Camada de *Benchmark*, existe um mecanismo para o desenvolvimento e utilização dos *benchmarks* para o sistema *blockchain*. Esse mecanismo pode ser dividido em dois processos: *Master* e *Cliente*. O processo *Master* implementa um fluxo de teste padrão que contém três estágios:

- **Estágio de Preparação:** Neste estágio, a *Master* cria e inicializa um objeto interno do *blockchain* de acordo com o arquivo de configuração, implanta os *smart contracts* como especificado e começa a monitorar o consumo de recursos do sistema;

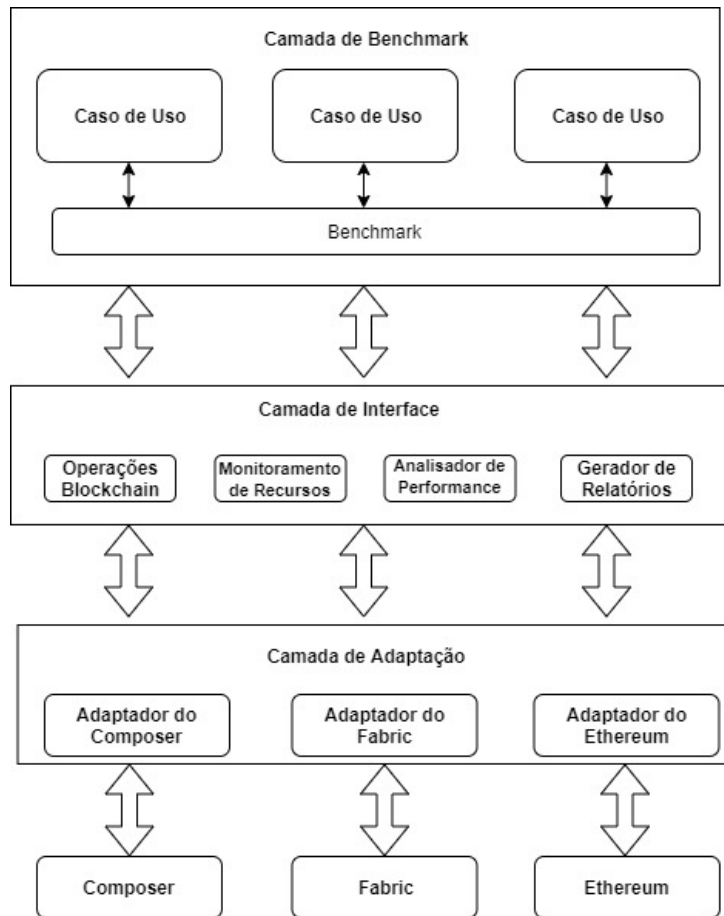


Figura 4.11: Arquitetura do Hyperledger Caliper adaptada.

- **Estágio de Teste:** Neste estágio, a *Master* inicia um laço para executar os testes de acordo com o arquivo de configuração do *benchmark*. Desta maneira, tarefas são criadas e assinadas para o cliente de acordo com as definições de workload, com isso os dados estatísticos de performance são armazenados para uma análise posterior;
- **Estágio de Relatório:** Neste estágio, as estatísticas de cada rodada de teste são analisadas e um relatório em *HTML* é gerado automaticamente.

O processo *Cliente* pode ser dividido em duas partes, a primeira de *Clientes Locais* no qual a *Master* utiliza um modulo de *cluster* para simular múltiplos clientes locais para executar as tarefas de teste. A segunda parte *Clientes Z* no qual múltiplos clientes são independentes, desta maneira um cliente se registra e aguarda pelas tarefas de teste, logo após o cliente recebe um relatório contendo os resultados estatísticos de performance.

Capítulo 5

Simulação e Resultados

Para o estudo de caso, foi implementado a solução de acordo com a arquitetura proposta na seção anterior, com a utilização do *framework* do *Hyperledger Composer*. Neste contexto, para que ocorra a simulação, verificação dos dados e dos indicadores de performance foi utilizado o *Hyperledger Caliper*.

Neste capítulo são apresentados os experimentos e testes utilizados para análise e validação da arquitetura proposta. Desta maneira, foram definidos diferentes cenários de teste, de acordo com o número de organizações e a quantidades de nós, e os argumentos da implementação a serem testados, como os ativos e as transações. Em seguida, para cada cenário de teste são apresentados os resultados obtidos e uma análise sobre os mesmos.

5.1 Indicadores de Performance

No contexto do *Hyperledger Composer*, é necessário diferenciar dois aspectos importantes: *Read* que são operações que não ocorrem mudança de estado e *Transação* que é uma operação que realiza uma transição de estado que modifica os dados no *blockchain*. Para a realização dos testes e simulações da arquitetura proposta é preciso definir indicadores de performance para uma análise mais aprofundada sobre a aplicação *blockchain*, que são:

- **Taxa de Sucesso:** Mensura todas as transações de sucesso e falha que são realizadas durante um ciclo de teste;
- **Latência Read (LR):** É o intervalo de tempo no qual uma requisição *Read* é enviada e sua resposta é recebida, sendo descrita pela equação a seguir.

$$LR = \text{Tempo de Resposta} - \text{Tempo de Envio}$$

- **Taxa de Transferência *Read* (TR):** É o valor de quantas operações *Read* são completadas em um período de tempo específico, expresso em *Reads* por segundo (RPS), sendo descrito pela equação a seguir.

$$TR = \frac{Total\ de\ Operações}{Tempo\ total}$$

- **Latência de Transação (LT):** É o intervalo de tempo em que uma transação é enviada até o seu resultado estar disponível na rede, incluindo o tempo de propagação e o tempo do protocolo de consenso, sendo descrito pela equação a seguir.

$$LT = Tempo\ de\ Confirmação\ da\ Transação - Tempo\ de\ Envio$$

- **Taxa de Transferência de Transação(TT):** É a taxa na qual as transações válidas são confirmadas na rede *blockchain* em um período de tempo específico, sendo descrito pela equação a seguir.

$$TT = \frac{Total\ de\ Operações\ de\ Transação}{Tempo\ total}$$

- **Consumo de Recursos:** Mensura a utilização de memória e o consumo da *CPU*, durante um ciclo de teste.

5.2 Cenários de Teste

Configuração dos Casos de Teste

Para realizar o *benchmark* e analisar os indicadores de performance, é necessário um arquivo de configuração com as definições dos argumentos para serem testados. Os argumentos contidos nesse arquivo são:

- **Nome e Descrição:** Nome e descrição do *benchmark*;
- **Clientes:** Define o tipo de cliente, se o tipo escolhido for *Local* o número de clientes precisa ser definido como um atributo;
- **Label:** Define o nome da transação que será utilizada para o teste de performance;
- **txNumber:** Define um *array* com diferentes números de transações para serem executadas em cada rodada de testes;
- **txDuration:** Define um *array* com diferentes tempos para a execução dos testes;

- **rateControl:** Define um *array* com taxas de controle personalizadas que são utilizadas durante os testes de *benchmark* nas sub-rotinas;
- **trim:** Realiza operações nos resultados do cliente para eliminar dados que não são relevantes no contexto dos testes;
- **arguments:** Define os argumentos que serão passados diretamente para o módulo de teste do usuário;
- **callback:** Especifica o módulo definido pelo usuário para ser utilizado nas sub-rotinas;
- **Monitor:** Define o tipo de recursos e objetos a serem monitorados, assim como o intervalo de tempo de monitoração;
 - **Docker:** É utilizado para monitorar contêineres específicos de *docker* em ambiente local ou remoto;
 - **Process:** É utilizado para monitorar um processo local específico;

Para executar a aplicação no *Hyperledger Composer* e os testes no *Hyperledger Caliper*, foi utilizada uma Máquina Virtual com as seguintes configurações: sistema operacional com *Ubuntu 16.04*, memória de 4GB e espaço de armazenamento de 80GB.

Um cenário de teste é composto por uma quantidade de Organizações e por uma quantidade de *Peers* presentes nas mesmas, além de uma configuração do protocolo de consenso embutida no nó. Para cada cenário de teste definido, a solução proposta é aplicada através dos *benchmarks* desenvolvidos e de *scripts* que executam e simulam o funcionamento da aplicação, como: o ativo de plano de voo, a criação dos mesmos, criação dos blocos para armazenar os dados, a transação de validar o plano de voo.

5.2.1 Benchmark - Parâmetros Gerais

O *benchmark* do *SWIM* contém uma série de transações para simular as operações do Plano de voo, como a criação do plano de voo, validação do mesmo entre outras. A partir desse *benchmark* realiza-se os testes de performance sobre os serviço de plano de voo.

Para executar os testes é necessário realizar a configuração do arquivo do *benchmark*. Desta maneira é preciso definir a quantidade de planos de voo que serão criados, como um atributo dentro da configuração. Logo após são geradas um conjunto de transações randômicas que serão executadas sobre esses planos de voo. O arquivo de configuração do *benchmark* é dividido em duas partes: operações e configuração. A parte de operações contém os seguintes atributos:

- **Criação do Plano de Voo:** Contém a função de teste que criará os planos de voo de acordo com os parâmetros, durante a rotina de testes;
- **Consultar Plano de Voo:** Contém a função de teste que consultará os planos de voo de acordo com os parâmetros, durante a rotina de testes;
- **Workload:** Contém a função de criação do *workload* do *benchmark*, com os valores dos planos de voo criados, assim como as operações que serão executadas sobre os mesmos, no caso validar plano de voo.

A parte de configuração contém os seguintes atributos:

- **Cientes:** Contém o tipo de cliente que será executado, no caso *Local*;
- **txNumber:** Com o valor de 500;
- **arguments:** Com o valor de 1000;
- **callback:** Contém a chamada a rotina das operações do plano de voo.

O valor do atributo do *txNumber* foi definido de acordo com as configurações padrões dos arquivos de modelo do próprio *Hyperledger Caliper*. Além disso, as operações escolhidas foram aquelas que modificam os estados do plano de voo.

5.2.2 Network – Seleção de Parâmetros

Para que ocorra uma validação da rede e uma comparação em relação aos indicadores de performance é importante analisar alguns aspectos da rede *blockchain* que serão realizados na aplicação. Desta maneira, as principais características que podem influenciar o sistema são a quantidade de organizações presentes e o número de nós em cada organização. Nesse contexto, os cenários de testes foram escolhidos de acordos com os seguintes critérios:

- **2 Organizações:** Definido pela configuração padrão do *Hyperledger Caliper*;
- **3 Organizações:** Definido pela quantidade de tipos de *stakeholders* definidos na arquitetura proposta;
- **1 Peer:** Definido pela configuração padrão do *Hyperledger Caliper*;
- **2 Peers e 3 Peers:** Definido para promover uma capacidade maior de tolerância a possíveis falhas.

5.2.3 Cenário de Teste 1

O Cenário de Teste 1 compreende a configuração padrão do *Hyperledger Caliper* com duas Organizações participantes e cada possuindo um *Peer*. Desta forma foram executados os testes em duas operações: Criação de Plano de Voo e Consulta de Plano de Voo.

Tabela 5.1: Cenário de Teste 1 - 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	600	0	10.1 tps	0.48 s	0.22s	0.29 s	10.0 tps
2	Criar um Plano de Voo	900	0	15.1 tps	0.52 s	0.11 s	0.37 s	15.0 tps
3	Criar um plano de Voo	1200	0	20.1 tps	0.42 s	0.05 s	0.24 s	20.0 tps
4	Criar um plano de Voo	1501	0	25.0 tps	0.41 s	0.05 s	0.20 s	24.9 tps
5	Criar um plano de Voo	1800	0	30.1 tps	0.33 s	0.06 s	0.18 s	30.0 tps
6	Criar um plano de Voo	2100	0	35.1 tps	0.35 s	0.06 s	0.17 s	35.0 tps
7	Criar um plano de Voo	2401	0	40.0 tps	0.35 s	0.05 s	0.15 s	39.9 tps
8	Criar um plano de Voo	2700	0	45.1 tps	0.31 s	0.04 s	0.14 s	45.0 tps
9	Criar um plano de Voo	3000	0	50.1 tps	0.37 s	0.04 s	0.14 s	50.0 tps
10	Criar um plano de Voo	3600	0	60.1 tps	0.65 s	0.05 s	0.19 s	59.7 tps
11	Criar um plano de Voo	4200	0	70.1 tps	0.82 s	0.06 s	0.16 s	70.0 tps
12	Criar um plano de Voo	4681	0	77.9 tps	2.57 s	0.11 s	0.62 s	77.3 tps
13	Criar um plano de Voo	4689	0	78.1 tps	1.43 s	0.20 s	0.62 s	77.6 tps
14	Criar um plano de Voo	4727	0	78.7 tps	1.31 s	0.21 s	0.64 s	78.1 tps
15	Criar um plano de Voo	4670	0	77.8 tps	2.38 s	0.19 s	0.71 s	77.3 tps
16	Criar um plano de Voo	4730	0	78.8 tps	1.36 s	0.23 s	0.66 s	78.2 tps
17	Criar um plano de Voo	4685	0	78.1 tps	1.40 s	0.29 s	0.64 s	77.4 tps

A Tabela 5.1 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 5 clientes. Sendo assim, a cada rodada de testes a taxa de envio dos dados aumenta assim como a taxa de vazão. Entretanto, a latência média foi diminuindo até chegar no menor valor na faixa de 70 transações por segundo (tps) e nas rodadas de testes seguintes obteve valores maiores. Com base nesse cenário de testes, o menor valor de latência foi de 0,16s, com a taxa de envio de 70,1tps e a maior taxa de vazão dos dados foi de 78,2tps com a taxa de envio de 78,8tps.

Tabela 5.2: Cenário de Teste 1 - 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	602	0	9.8 tps	1.20 s	0.07 s	0.31 s	9.8 tps
2	Criar um Plano de Voo	903	0	15.0 tps	0.85 s	0.07 s	0.30 s	14.9 tps
3	Criar um plano de Voo	1200	0	20.3 tps	0.62 s	0.14 s	0.30 s	20.2 tps
4	Criar um plano de Voo	1502	0	24.8 tps	1.06 s	0.07 s	0.43 s	24.7 tps
5	Criar um plano de Voo	1801	0	30.0 tps	1.01 s	0.16 s	0.33 s	29.9 tps
6	Criar um plano de Voo	2100	0	35.0 tps	0.88 s	0.06 s	0.34 s	34.9 tps
7	Criar um plano de Voo	2400	0	40.3 tps	0.48 s	0.16 s	0.30 s	40.1 tps
8	Criar um plano de Voo	2701	0	45.0 tps	1.85 s	0.17 s	0.34 s	44.8 tps
9	Criar um plano de Voo	2974	0	49.3 tps	5.22 s	0.08 s	0.65 s	48.5 tps
10	Criar um plano de Voo	3337	0	54.2 tps	6.02 s	0.24 s	2.27 s	54.0 tps
11	Criar um plano de Voo	3481	0	57.8 tps	3.33 s	0.39 s	2.37 s	56.5 tps
12	Criar um plano de Voo	3432	0	56.0 tps	5.43 s	0.39 s	2.42 s	55.6 tps
13	Criar um plano de Voo	3573	0	59.3 tps	5.67 s	0.48 s	2.43 s	57.9 tps
14	Criar um plano de Voo	3430	0	56.9 tps	5.99 s	0.48 s	2.47 s	55.8 tps
15	Criar um plano de Voo	3677	0	61.0 tps	5.62 s	0.55 s	2.33 s	59.8 tps
16	Criar um plano de Voo	3516	0	58.4 tps	5.56 s	0.60 s	2.41 s	56.9 tps
17	Criar um plano de Voo	3715	0	61.5 tps	5.59 s	0.81 s	2.57 s	59.0 tps

A Tabela 5.2 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 20 clientes. Sendo assim a cada rodada de teste a taxa de envio aumenta até atingir o valor máximo em 61,5tps. Nesse caso os melhores valores de latência ocorreram

com a taxas de envio mais baixas. Desta maneira, o melhor valor de latência média foi de 0,30s que ocorreu com a taxa de envio de 40,3tps e o melhor valor de taxa de vazão foi de 59,8tps que ocorreu na faixa de 61,0 tps de envio de dados.

Tabela 5.3: Cenário de Teste 1 - 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	601	0	9.7 tps	2.60 s	0.13 s	0.59 s	9.7 tps
2	Criar um Plano de Voo	903	0	14.7 tps	1.52 s	0.15 s	0.68 s	14.6 tps
3	Criar um plano de Voo	1203	0	19.6 tps	1.73 s	0.13 s	0.43 s	19.5 tps
4	Criar um plano de Voo	1500	0	25.1 tps	1.63 s	0.11 s	0.65 s	25.0 tps
5	Criar um plano de Voo	1802	0	29.5 tps	2.35 s	0.08 s	0.58 s	29.4 tps
6	Criar um plano de Voo	2100	0	35.4 tps	2.30 s	0.13 s	0.86 s	35.0 tps
7	Criar um plano de Voo	2400	0	39.2 tps	4.60 s	0.20 s	0.92 s	39.0 tps
8	Criar um plano de Voo	2697	1	44.3 tps	10.63 s	0.18 s	1.20 s	37.8 tps
9	Criar um plano de Voo	2993	0	49.1 tps	14.60 s	0.87 s	4.20 s	47.2 tps
10	Criar um plano de Voo	3110	2	50.9 tps	15.62 s	0.92 s	6.05 s	48.0 tps
11	Criar um plano de Voo	3009	9	47.7 tps	15.27 s	0.68 s	6.48 s	39.0 tps
12	Criar um plano de Voo	3186	3	52.0 tps	16.05 s	1.01 s	6.31 s	45.5 tps
13	Criar um plano de Voo	3149	2	51.7 tps	15.52 s	1.40 s	6.45 s	48.8 tps
14	Criar um plano de Voo	3079	0	50.6 tps	12.85 s	1.33 s	6.70 s	47.7 tps
15	Criar um plano de Voo	3160	2	51.8 tps	17.28 s	0.88 s	6.49 s	49.1 tps
16	Criar um plano de Voo	3170	0	51.7 tps	14.56 s	1.16 s	6.53 s	49.1 tps
17	Criar um plano de Voo	3012	0	49.4 tps	12.84 s	1.41 s	6.92 s	46.8 tps

A Tabela 5.3 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 50 clientes. Sendo assim a cada rodada de testes a taxa de envio aumenta até atingir um valor máximo de 52,0tps. Nesse conjunto de testes o melhor resultado para a latência média foi de 0,43s que ocorreu no terceiro round dos testes e o pico da taxa de vazão foi de 49,1tps na faixa de taxa de envio de 51,7tps.

Tabela 5.4: Cenário de Teste 1 - Query 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	85	0	37.2 tps	0.31 s	0.01s	0.05s	36.9 tps
2	Consultar um Plano de voo	2036	0	67.7 tps	0.95 s	0.02 s	0.25 s	67.6 tps
3	Consultar um Plano de voo	1914	0	63.7 tps	1.14 s	0.01 s	0.26 s	63.3 tps
4	Consultar um Plano de voo	4176	0	69.6 tps	1.04 s	0.01 s	0.25 s	69.4 tps
5	Consultar um Plano de voo	4188	0	69.7 tps	0.86 s	0.01 s	0.26 s	69.6 tps
6	Consultar um Plano de voo	4189	0	69.8 tps	1.04 s	0.01 s	0.26 s	69.7 tps

A Tabela 5.4 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 5 clientes. Sendo assim, é possível perceber que a taxa de envio alcançou valores próximos, com o pico de 69,7tps. Nesse conjunto de teste, o melhor valor de taxa de latência média foi de 0,25s, descartando o valor do primeiro round que não foi muito conclusivo para a análise desse caso e a melhor taxa de vazão foi de 69,7tps.

A Tabela 5.5 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 20 clientes. Sendo assim, é possível perceber que a taxa de envio ficou

Tabela 5.5: Cenário de Teste 1 - Query 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	266	0	5.8 tps	0.62 s	0.01 s	0.15 s	5.8 tps
2	Consultar um Plano de voo	2026	0	66.4 tps	1.70 s	0.17 s	0.96 s	66.0 tps
3	Consultar um Plano de voo	1901	0	62.1 tps	3.11 s	0.20 s	1.04 s	61.5 tps
4	Consultar um Plano de voo	4002	0	66.2 tps	2.30 s	0.15 s	1.00 s	65.9 tps
5	Consultar um Plano de voo	4165	0	68.9 tps	2.44 s	0.13 s	0.97 s	68.6 tps
6	Consultar um Plano de voo	3922	0	64.9 tps	2.60 s	0.16 s	1.00 s	64.6 tps

próxima na maioria dos rounds, alcançando o pico em 68,9tps. Nesse conjunto de testes a latência média teve seu menor valor, desconsiderando a primeira rodada de testes, em 0,96s e o maior valor de taxa de vazão foi de 68,6tps.

Tabela 5.6: Cenário de Teste 1 - Query 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	587	0	8.8 tps	0.80 s	0.01 s	0.17 s	8.8 tps
2	Consultar um Plano de voo	4120	0	132.6 tps	2.10 s	0.01 s	1.08 s	132.5 tps
3	Consultar um Plano de voo	4020	0	131.7 tps	2.21 s	0.22 s	1.11 s	130.6 tps
4	Consultar um Plano de voo	8360	0	137.7 tps	2.11 s	0.17 s	1.09 s	137.3 tps
5	Consultar um Plano de voo	8445	0	139.7 tps	2.40 s	0.28 s	1.07 s	139.0 tps
6	Consultar um Plano de voo	7822	0	127.8 tps	2.75 s	0.03 s	1.15 s	127.8 tps

A Tabela 5.6 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 50 clientes. Sendo assim, é possível perceber que a taxa de envio alcançou o pico de 139,7tps. Nesse conjunto de testes, o melhor valor para a latência média foi de 1,07s, desconsiderando o primeiro round, e o maior valor de taxa de vazão foi de 139,0tps.

Criar plano de Voo

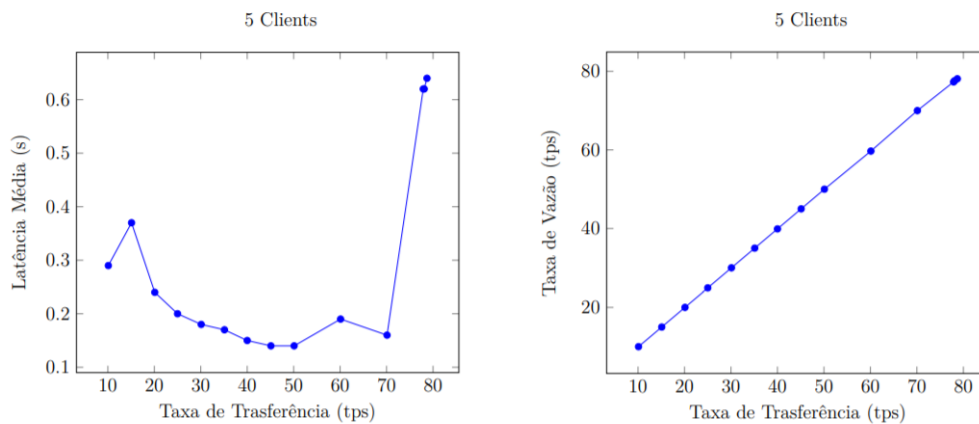


Figura 5.1: Criação de Plano de Voo com 5 Clientes.

A Figura 5.1 apresenta dois gráficos dos testes da operação de criação de plano de voo com 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, sendo possível observar que a latência média começa em um valor de 0,29s, aumenta e depois alcança os valores mais baixos. Porém, a partir da taxa de envio de 70,1tps a latência começa a subir atingindo os valores mais altos dos testes. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que apresenta uma linha crescente entre os valores até chegar no pico de valor da taxa de vazão que ocorre em 78,2tps.

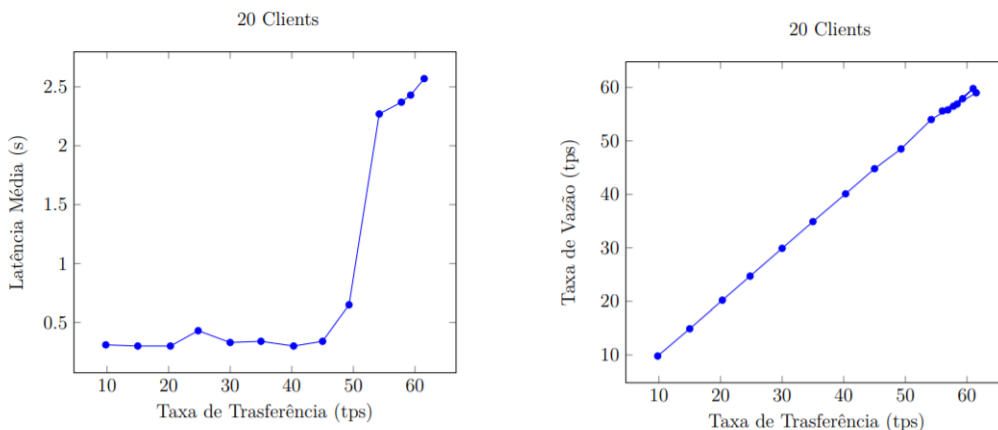


Figura 5.2: Criação de Plano de Voo com 20 Clientes.

A Figura 5.2 apresenta dois gráficos dos testes da operação de criação de plano de voo com 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média com os valores começando em média por 0,30s e se mantendo assim até

começar a subir na faixa de 50 e 60 tps, desta maneira é possível observar uma crescente acentuada nessa faixa de taxa de transferência. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que apresenta uma linha crescente entre os valores até alcançar o pico da taxa de vazão que ocorre em 59,0tps.

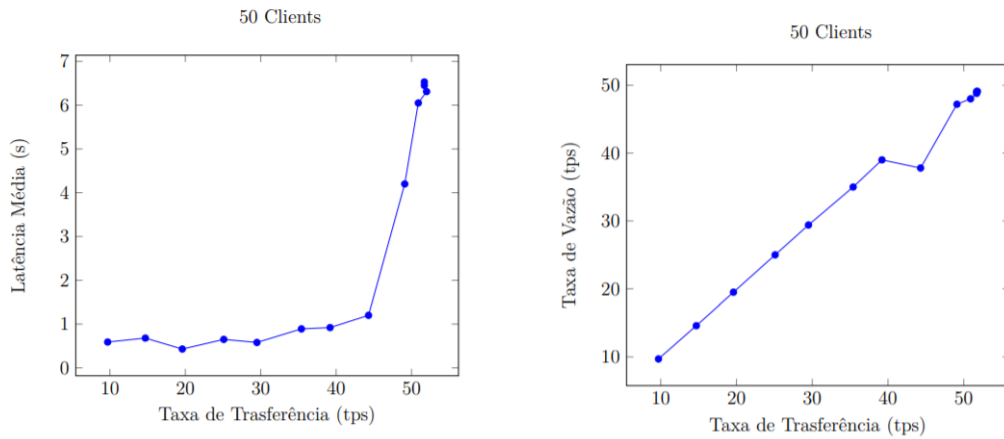


Figura 5.3: Criação de Plano de Voo com 50 Clientes.

A Figura 5.3 apresenta dois gráficos dos testes da operação de criação de plano de voo com 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média com os valores começando com 0,59s e se mantendo abaixo de 1 segundo até a taxa de envio em média de 40tps, em seguida ocorre uma crescente acentuada até atingir o valor máximo de latência que foi de 6,92s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que apresenta uma linha crescente entre os valores até atingir o pico da taxa de vazão que ocorre em 49,1tps.

Consultar Plano de voo

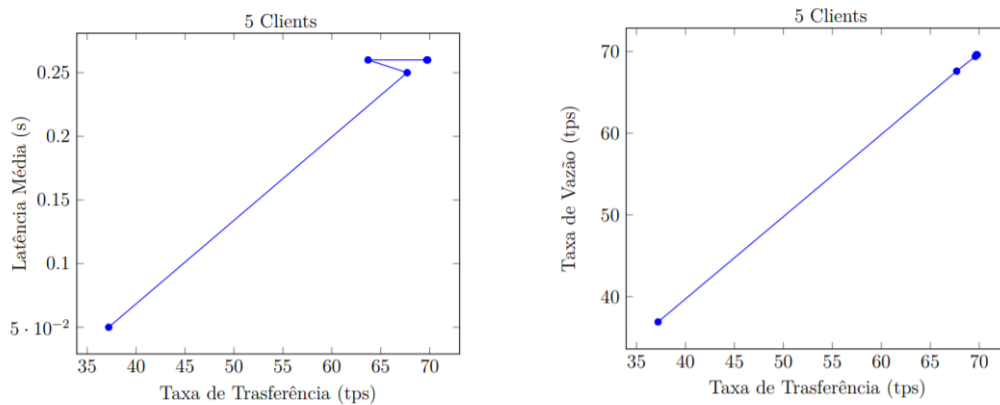


Figura 5.4: Consultar um plano de voo com 5 Clientes

A Figura 5.4 apresenta dois gráficos com os resultados da operação de consultar plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível perceber que a latência começa em um valor baixo e sobe até atingir o pico de 0,26s e se mantém nessa média. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão, sendo assim uma linha crescente até se estabilizar entre 69tps e atingir o pico em 69,8tps

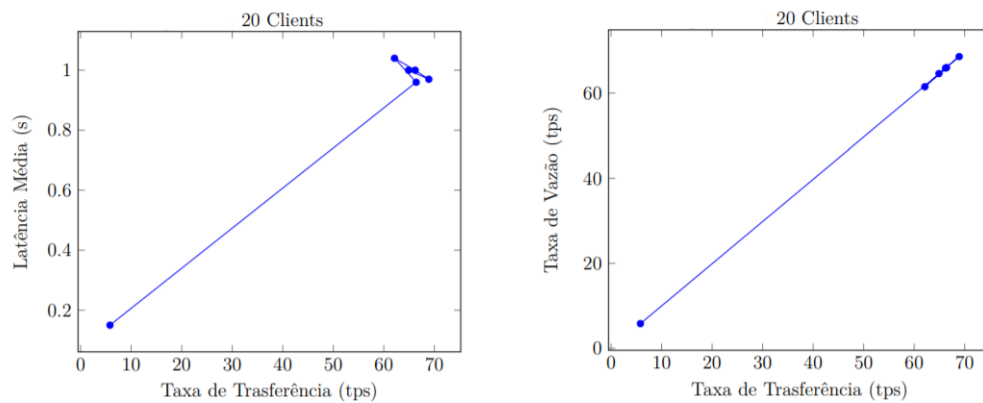


Figura 5.5: Consultar um plano de voo com 20 Clientes

A Figura 5.5 apresenta dois gráficos com os resultados da operação de consultar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível perceber que a latência média começa em um valor baixo e sobe até atingir o maior valor que foi de 1,04s e se mantém nessa média de 1,00s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão,

sendo assim, é possível perceber uma linha crescente até se estabilizar por volta dos 65tps e atingir o pico de 68,6tps.

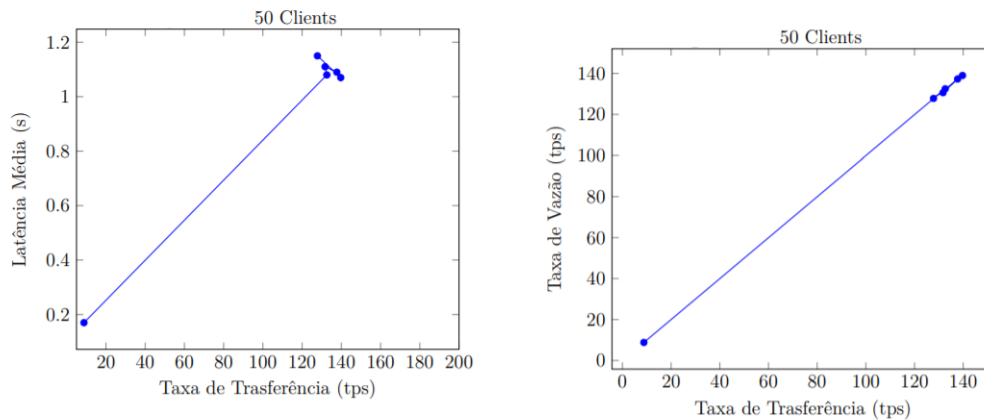


Figura 5.6: Consultar um plano de voo com 50 Clientes

A Figura 5.6 apresenta dois gráficos com os resultados dos testes da operação de consultar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, sendo assim, é possível perceber que a latência média começa em um valor baixo até atingir o pico de 1,15s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão, dessa maneira, é possível constatar uma linha crescente até se estabilizar na faixa dos 130tps e atingir o pico de 139tps.

5.2.4 Cenário de Teste 2

O Cenário de Teste 2 compreende a configuração com três Organizações participantes e cada uma possuindo um *Peer*.

Tabela 5.7: Cenário de Teste 2 - 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	600	0	10.1 tps	0.60 s	0.17 s	0.29 s	10.0 tps
2	Criar um Plano de Voo	901	0	15.0 tps	0.57 s	0.10 s	0.48 s	15.2 tps
3	Criar um plano de Voo	1200	0	20.1 tps	0.47 s	0.08 s	0.38 s	19.9 tps
4	Criar um plano de Voo	1500	0	25.1 tps	0.55 s	0.06 s	0.63 s	25.0 tps
5	Criar um plano de Voo	1800	0	30.1 tps	0.80 s	0.09 s	0.46 s	29.48tps
6	Criar um plano de Voo	2101	0	35.0 tps	0.40 s	0.06 s	0.20 s	34.8 tps
7	Criar um plano de Voo	2401	0	40.0 tps	0.45 s	0.08 s	0.20 s	39.8 tps
8	Criar um plano de Voo	2701	0	45.0 tps	0.59 s	0.08 s	0.21 s	44.8 tps
9	Criar um plano de Voo	3001	0	50.0 tps	0.84 s	0.08 s	0.26 s	49.8 tps
10	Criar um plano de Voo	3600	0	60.0 tps	2.53 s	0.15 s	1.14 s	59.5 tps
11	Criar um plano de Voo	3806	0	63.3 tps	5.99 s	0.16 s	3.46 s	61.4 tps
12	Criar um plano de Voo	3902	0	64.8 tps	8.19 s	0.24 s	3.63 s	62.1 tps
13	Criar um plano de Voo	3613	0	59.8 tps	4.23 s	0.33 s	2.11 s	59.2 tps
14	Criar um plano de Voo	3592	0	59.8 tps	6.06 s	0.70 s	3.09 s	58.6 tps
15	Criar um plano de Voo	3789	0	63.0 tps	7.00 s	0.98 s	2.60 s	61.5 tps
16	Criar um plano de Voo	3695	0	61.5 tps	8.60 s	0.90 s	4.37 s	59.3 tps
17	Criar um plano de Voo	3602	0	60.0 tps	5.20 s	0.91 s	3.00 s	58.8 tps

A Tabela 5.7 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 5 clientes. Sendo assim, a cada rodada de testes a taxa de envio dos

dados aumenta até atingir um pico de 64,8tps, dessa maneira é possível perceber que a taxa de vazão aumenta junto, atingindo um pico de 62,1tps. Entretanto, a latência média não atende um padrão definido, que pode variar de acordo com a taxa de envio. Sendo assim, com bases nesse cenário de teste, o menor valor de latência média alcançado foi de 0,20s na faixa de 40,0tps e o melhor valor de taxa de vazão foi de 62,1tps e ocorre na faixa de envio de 64,8tps.

Tabela 5.8: Cenário de Teste 2 - 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	600	0	10.0 tps	1.18 s	0.07 s	0.31 s	10.0 tps
2	Criar um Plano de Voo	901	0	15.3 tps	0.87 s	0.23 s	0.48 s	15.2 tps
3	Criar um plano de Voo	1201	0	20.0 tps	0.72 s	0.11 s	0.38 s	19.9 tps
4	Criar um plano de Voo	1502	0	25.0 tps	3.00 s	0.08 s	0.63 s	25.0 tps
5	Criar um plano de Voo	1801	0	30.0 tps	1.57 s	0.11 s	0.46 s	29.48tps
6	Criar um plano de Voo	2102	0	34.4 tps	1.26 s	0.09 s	0.38 s	34.3 tps
7	Criar um plano de Voo	2401	0	40.0 tps	3.34 s	0.24 s	0.50 s	39.5 tps
8	Criar um plano de Voo	2616	0	43.1 tps	6.49 s	0.43 s	2.07 s	41.8 tps
9	Criar um plano de Voo	2839	0	46.9 tps	6.05 s	0.47 s	3.70 s	45.3 tps
10	Criar um plano de Voo	2780	0	45.9 tps	7.42 s	0.48 s	3.52 s	44.6 tps
11	Criar um plano de Voo	2862	0	47.1 tps	8.12 s	0.88 s	4.60 s	46.1 tps
12	Criar um plano de Voo	2892	0	47.6 tps	8.04 s	0.74 s	4.00 s	46.3 tps
13	Criar um plano de Voo	3001	0	49.4 tps	8.95 s	0.68 s	5.41 s	47.3 tps
14	Criar um plano de Voo	2916	0	47.3 tps	7.14 s	1.01 s	3.90 s	46.1 tps
15	Criar um plano de Voo	2929	0	48.1 tps	7.88 s	0.88 s	3.90 s	46.8 tps
16	Criar um plano de Voo	2868	0	46.8 tps	6.38 s	0.96 s	3.94 s	45.8 tps
17	Criar um plano de Voo	2828	0	46.6 tps	8.17 s	1.38 s	4.03 s	45.3 tps

A Tabela 5.8 apresenta os resultados obtidos na execução do teste de criação de plano de voo para 20 clientes. Sendo assim, a cada rodada de testes a taxa de envio aumenta até atingir uma média de 46tps e um pico de 49,4tps, desse modo a taxa de vazão também foi aumentando à medida que os rounds aconteciam, até atingir o pico de 47,3tps. Entretanto a latência média não segue um padrão definido, que pode variar de acordo com a taxa de envio. Com base nesse cenário de teste, o menor valor de latência alcançado foi de 0,31s na taxa de envio de 10tps e o melhor valor de taxa de vazão foi de 47,3tps na taxa de envio de 49,4tps.

Tabela 5.9: Cenário de Teste 2 - 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	602	0	9.6 tps	4.20 s	0.16 s	0.92 s	9.5 tps
2	Criar um Plano de Voo	902	0	14.5 tps	2.26 s	0.12 s	0.59 s	14.4 tps
3	Criar um plano de Voo	1203	0	19.5 tps	2.35 s	0.12 s	0.64 s	19.4 tps
4	Criar um plano de Voo	1503	0	23.9 tps	8.98 s	0.13 s	1.38 s	23.8 tps
5	Criar um plano de Voo	1757	0	27.7 tps	21.78 s	0.07 s	2.66 s	23.5 tps
6	Criar um plano de Voo	2020	0	32.4 tps	18.94 s	1.22 s	4.92 s	24.1 tps
7	Criar um plano de Voo	2244	0	32.6 tps	25.41 s	1.16 s	7.72 s	24.4 tps
8	Criar um plano de Voo	2173	0	36.4 tps	11.27 s	1.08 s	8.40 s	33.7 tps
9	Criar um plano de Voo	2185	0	35.0 tps	20.44 s	1.30 s	8.65 s	25.4 tps
10	Criar um plano de Voo	2371	0	34.9 tps	19.94 s	1.20 s	10.13 s	25.7 tps
11	Criar um plano de Voo	2179	0	37.2 tps	16.19 s	1.35 s	8.80 s	28.4 tps
12	Criar um plano de Voo	2357	0	35.4 tps	19.04 s	2.03 s	9.22 s	26.1 tps
13	Criar um plano de Voo	2436	0	37.2 tps	19.41 s	1.63 s	10.23 s	27.4 tps
14	Criar um plano de Voo	2436	0	38.3 tps	12.38 s	1.59 s	8.68 s	36.3 tps
15	Criar um plano de Voo	2351	0	37.8 tps	19.20 s	1.53 s	8.92 s	27.4 tps
16	Criar um plano de Voo	2277	0	36.1 tps	19.26 s	2.22 s	9.11 s	27.7 tps
17	Criar um plano de Voo	2113	0	34.2 tps	20.13 s	2.10 s	9.44 s	24.9 tps

A Tabela 5.9 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 50 clientes. Sendo assim, a cada rodada de testes a taxa de envio aumenta até uma média de 35,0tps e atingir o pico de 38,3tps, desse modo a taxa de vazão também foi aumentando à medida que as rodadas aconteciam até atingir o pico de 36,3tps. Entretanto, a latência média começou nos primeiros rounds com valores menores que 1s e depois foi crescendo. Com bases nesse cenário de teste, o menor valor de latência média alcançado foi de 0,59s na faixa de 14,5tps de taxa de envio e o melhor valor de taxa de vazão que foi de 36,3tps alcançado na faixa de taxa de envio de 38,3tps.

Tabela 5.10: Cenário de Teste 2 - Query 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	65	0	23.5 tps	0.15 s	0.02 s	0.06 s	23.3 tps
2	Consultar um Plano de voo	1732	0	56.1 tps	1.19 s	0.02 s	0.34 s	56.0 tps
3	Consultar um Plano de voo	1175	0	39.0 tps	1.84 s	0.01 s	0.48 s	38.1 tps
4	Consultar um Plano de voo	3292	0	54.9 tps	1.18 s	0.02 s	0.34 s	54.7 tps
5	Consultar um Plano de voo	3487	0	58.0 tps	1.11 s	0.04 s	0.36 s	57.9 tps
6	Consultar um Plano de voo	3652	0	60.4 tps	1.59 s	0.02 s	0.41 s	60.3 tps

A Tabela 5.10 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 5 clientes. Sendo assim é possível perceber que a taxa de envio apresenta dois valores abaixo de 40tps e depois valores próximos entre 54 e 61 tps, atingindo assim o pico de 60,4tps. Nesse conjunto de testes, o melhor valor de latência média foi de 0,34s, desconsiderando o primeiro round de testes que não foi conclusivo para análise do cenário e o melhor valor de taxa de vazão foi de 60,3tps.

Tabela 5.11: Cenário de Teste 2 - Query 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	325	0	9.2 tps	0.23 s	0.01 s	0.05 s	9.2 tps
2	Consultar um Plano de voo	3245	0	97.2 tps	1.85 s	0.03 s	0.63 s	97.1 tps
3	Consultar um Plano de voo	3543	0	110.8 tps	1.93 s	0.03 s	0.59 s	110.7 tps
4	Consultar um Plano de voo	7062	0	116.5 tps	2.01 s	0.03 s	0.61 s	116.4 tps
5	Consultar um Plano de voo	7382	0	118.6 tps	2.04 s	0.02 s	0.60 s	118.5 tps
6	Consultar um Plano de voo	7722	0	128.4 tps	1.57 s	0.09 s	0.59 s	127.8 tps

A Tabela 5.11 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 20 clientes. Sendo assim, a taxa de envio começa com um valor muito baixo, podendo ser desconsiderado da análise, e em seguida os valores sobem até atingir o pico de 128,4tps. Nesse conjunto de testes, o melhor valor de latência média foi de 0,59s e o melhor valor de taxa de vazão foi de 127,8tps.

Tabela 5.12: Cenário de Teste 2 - Query 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	553	0	7.1 tps	1.57 s	0.01 s	0.33 s	7.1 tps
2	Consultar um Plano de voo	3189	0	102.6 tps	3.94 s	0.27 s	1.65 s	101.2 tps
3	Consultar um Plano de voo	3290	0	106.9 tps	2.42 s	0.31 s	1.61 s	105.7 tps
4	Consultar um Plano de voo	6459	0	106.7 tps	3.31 s	0.37 s	1.67 s	105.6 tps
5	Consultar um Plano de voo	6645	0	109.2 tps	2.95 s	0.26 s	1.65 s	108.3 tps
6	Consultar um Plano de voo	6541	0	106.9 tps	3.13 s	0.17 s	1.65 s	106.6 tps

A Tabela 5.12 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 50 clientes. Sendo assim, a taxa de envio começa com um valor muito baixo, podendo ser desconsiderado da análise, em seguida sobe com valores muito próximos, atingido a faixa de 100 a 110tps, com o pico de 109.2tps. Nesse conjunto de testes, o melhor valor de latência média foi de 1,61s na segunda rodada e o melhor valor de taxa de vazão foi de 108,3tps alcançado na quinta rodada.

Criar plano de Voo

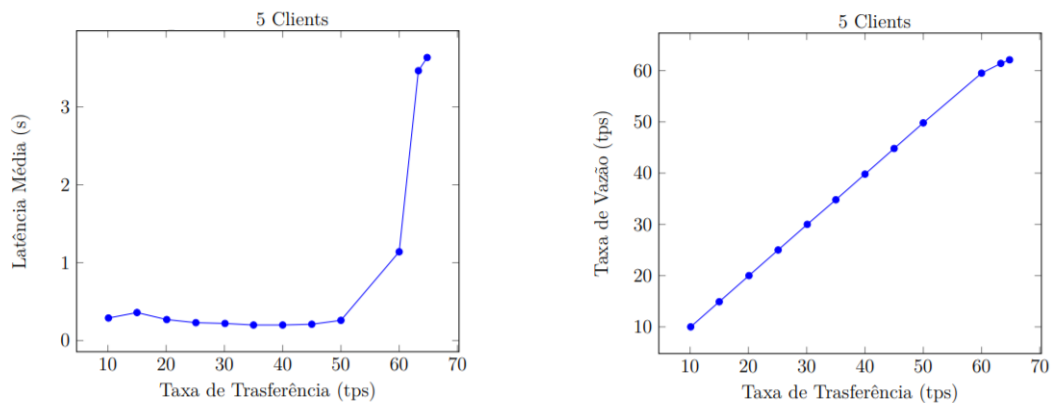


Figura 5.7: Criação de Plano de Voo com 5 Clientes

A Figura 5.7 apresenta dois gráficos dos testes da operação de criação de plano de voo com 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência media começa 0,29s e se mantém entre 0 e 1 segundo a medida que a taxa de transferência aumenta, até atingir um ponto que faz com que a latência alcance o pico de 4,37s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que apresenta uma linha crescente entre os valores até chegar no pico do valor que ocorre em 62,1tps.

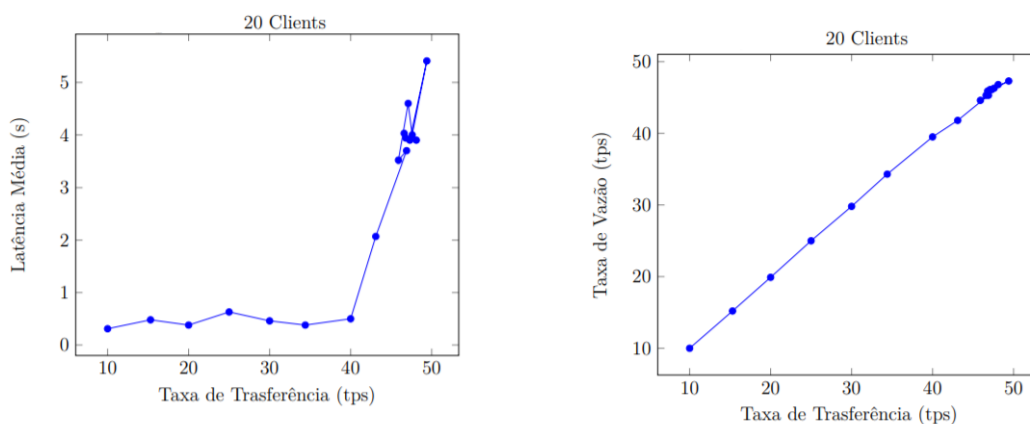


Figura 5.8: Criação de Plano de Voo com 20 Clientes

A Figura 5.8 apresenta dois gráficos dos testes da operação de criação de plano de voo com 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa em 0,31s e se mantém entre 0 e 1s a medida que a taxa de transferência aumenta, a partir da faixa de 43,1tps, a latência média se eleva até atingir o pico de 5,41s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que apresenta uma linha crescente entre os valores até chegar no pico de 47,3tps.

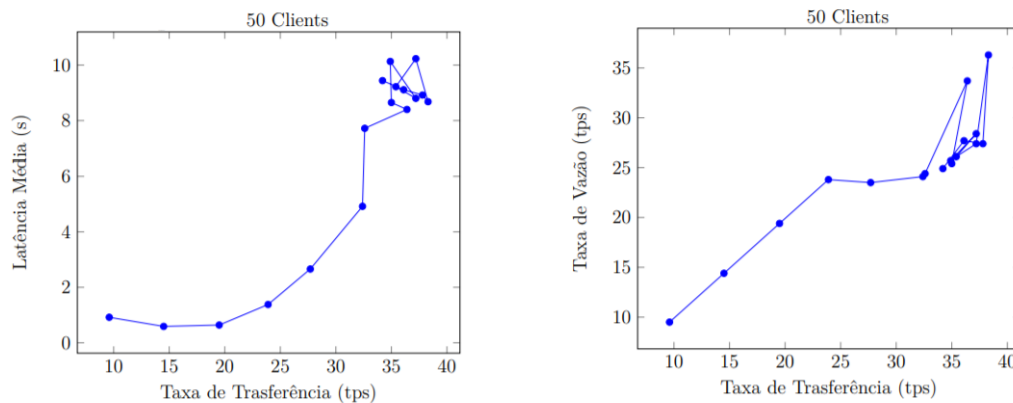


Figura 5.9: Criação de Plano de Voo com 50 Clientes

A Figura 5.9 apresenta dois gráficos dos testes da operação de criação de plano de voo com 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que os primeiros valores são abaixo de 1 segundo e em seguida a taxa começa a subir e se mantém em valores altos, atingindo assim o pico de 10,23s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que cresce nos primeiros valores e se estabiliza entre 25 e 35tps, dessa forma atingindo o pico de 36,3tps.

Consultar Plano de voo

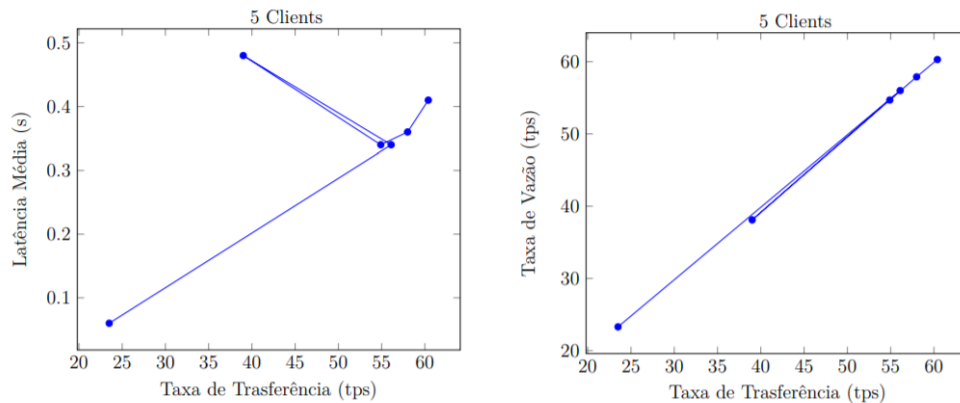


Figura 5.10: Consultar um plano de voo com 5 Clientes

A Figura 5.10 apresenta dois gráficos dos testes da operação de consulta de plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que começa em 0,06s e cresce ficando na faixa de 0,30 a 0,50s, sendo o pico no valor de 0,48s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 23,3tps, cresce até atingir o pico de 60,3tps.

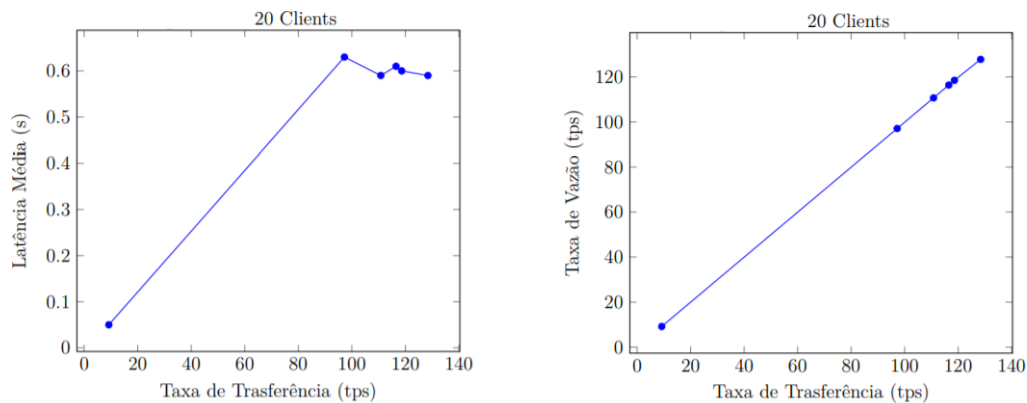


Figura 5.11: Consultar um plano de voo com 20 Clientes

A Figura 5.11 apresenta dois gráficos dos testes da operação de consultar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,05s, cresce atingindo o pico de 0,63s e se estabiliza na média de 0,60s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 9,2tps e cresce até atingir o pico de 127,8tps.

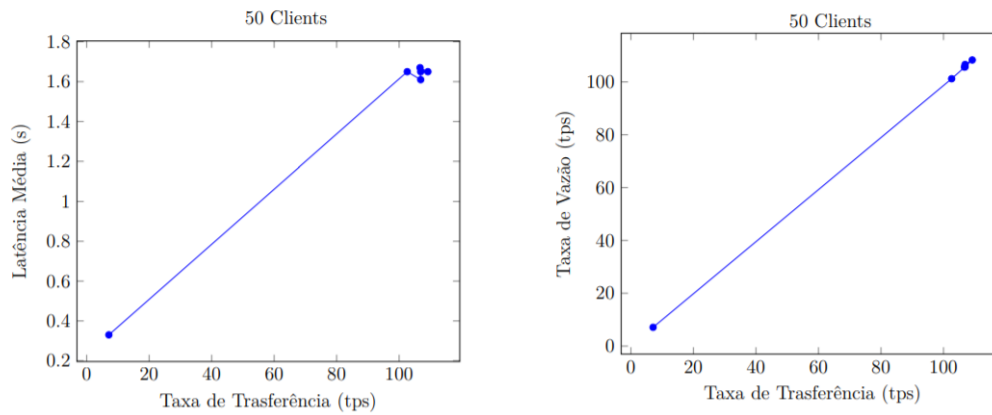


Figura 5.12: Consultar um plano de voo com 50 Clientes

A Figura 5.12 apresenta dois gráficos dos testes da operação de consultar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,33s, cresce atingindo o pico de 1,67s e se estabiliza no valor de 1,65s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 7,1tps e cresce até atingir o pico de 108,3tps.

5.2.5 Cenário de Teste 3

O Cenário de Teste 3 compreende a configuração com três Organizações participantes e cada possuindo dois *Peers*.

Tabela 5.13: Cenário de Teste 3 - 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	600	0	10.1 tps	0.63 s	0.19 s	0.32 s	10.0 tps
2	Criar um Plano de Voo	900	0	15.1 tps	0.60 s	0.12 s	0.43 s	15.0 tps
3	Criar um plano de Voo	1200	0	20.1 tps	0.73 s	0.09 s	0.31 s	20.0 tps
4	Criar um plano de Voo	1500	0	25.1 tps	0.56 s	0.12 s	0.28 s	24.9 tps
5	Criar um plano de Voo	1801	0	30.0 tps	0.61 s	0.13 s	0.27 s	29.9 tps
6	Criar um plano de Voo	2100	0	35.1 tps	0.62 s	0.13 s	0.31 s	35.0 tps
7	Criar um plano de Voo	2400	0	40.1 tps	10.91 s	0.12 s	5.49 s	36.8 tps
8	Criar um plano de Voo	2261	0	37.6 tps	6.65 s	0.42 s	3.35 s	36.1 tps
9	Criar um plano de Voo	3002	0	35.0 tps	20.44 s	1.30 s	8.65 s	25.4 tps
10	Criar um plano de Voo	3282	0	54.6 tps	28.33 s	0.55 s	18.75 s	40.7 tps
11	Criar um plano de Voo	2671	0	44.4 tps	17.45 s	0.82 s	10.30 s	37.8 tps
12	Criar um plano de Voo	2823	0	46.9 tps	18.80 s	0.89 s	9.86 s	39.9 tps
13	Criar um plano de Voo	643	0	37.2 tps	19.41 s	1.63 s	10.23 s	27.4 tps
14	Criar um plano de Voo	410	0	38.3 tps	12.38 s	1.59 s	8.68 s	36.3 tps
15	Criar um plano de Voo	494	0	37.8 tps	19.20 s	1.53 s	8.92 s	27.4 tps
16	Criar um plano de Voo	531	0	36.1 tps	19.26 s	2.22 s	9.11 s	27.7 tps
17	Criar um plano de Voo	1229	0	34.2 tps	20.13 s	2.10 s	9.44 s	24.9 tps

A Tabela 5.13 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 5 clientes. Sendo assim, nos primeiros rounds a taxa de envio sobe constantemente, entretanto a partir da oitava rodada, os valores de taxa de envio começam a variar bastante, atingindo o seu pico de 54,6tps. A latência média tem seus primeiros

valores abaixo de 1s, porém após essas rodadas de testes, seu valor começa a aumentar, atingindo o pico de 18,75s. A taxa de vazão aumenta de acordo a medida que as rodadas de testes ocorrem, sofrendo algumas variações de valores até atingir o pico de 40.7tps.

Tabela 5.14: Cenário de Teste 3 - 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	601	0	9.9 tps	4.31 s	0.15 s	0.70 s	9.8 tps
2	Criar um Plano de Voo	900	0	15.3 tps	1.20 s	0.34 s	0.67 s	15.1 tps
3	Criar um plano de Voo	1202	0	19.9 tps	1.54 s	0.17 s	0.57 s	19.8 tps
4	Criar um plano de Voo	1502	0	25.0 tps	8.18 s	0.28 s	1.56 s	24.8 tps
5	Criar um plano de Voo	1797	0	29.8 tps	10.84 s	0.40 s	4.62 s	27.9 tps
6	Criar um plano de Voo	1749	0	28.2 tps	19.89 s	0.75 s	10.54 s	27.0 tps
7	Criar um plano de Voo	2394	0	39.7 tps	28.60 s	0.96 s	19.48 s	28.9 tps
8	Criar um plano de Voo	2275	0	37.6 tps	24.16 s	0.84 s	15.37 s	29.0 tps
9	Criar um plano de Voo	2261	0	37.0 tps	28.80 s	1.04 s	19.50 s	28.5 tps
10	Criar um plano de Voo	1190	1356	41.7 tps	30.24 s	1.34 s	17.49 s	13.4 tps
11	Criar um plano de Voo	2305	0	37.9 tps	27.41 s	1.56 s	18.95 s	29.2 tps
12	Criar um plano de Voo	1423	1071	41.2 tps	30.54 s	1.78 s	19.69 s	16.4 tps
13	Criar um plano de Voo	2298	0	37.3 tps	29.27 s	2.97 s	18.29 s	28.7 tps
14	Criar um plano de Voo	1290	1104	39.0 tps	31.08 s	1.80 s	19.76 s	15.1 tps
15	Criar um plano de Voo	788	1722	41.3 tps	30.41 s	1.74 s	19.54 s	8.9 tps
16	Criar um plano de Voo	2109	340	40.5 tps	30.60 s	2.05 s	22.18 s	24.8 tps
17	Criar um plano de Voo	873	1698	42.2 tps	30.22 s	3.59 s	18.67 s	9.6 tps

A Tabela 5.14 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 20 clientes. Sendo assim, a cada rodada de testes a taxa de envio de dados aumenta, apresentando algumas variações em alguns rounds, atingindo o pico de 54,6tps. Entretanto a latência média tem seus primeiros valores abaixo de 1s e logo em seguida seu valor começa a variar e aumentar, atingindo o seu pico de 19,50s. A taxa de vazão aumenta de acordo com as rodadas de testes, sofrendo algumas variações até atingir o pico de seu valor em 40,7tps.

Tabela 5.15: Cenário de Teste 3 - 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	604	0	9.3 tps	8.27 s	0.26 s	2.47 s	9.3 tps
2	Criar um Plano de Voo	901	0	14.6 tps	2.81 s	0.24 s	1.30 s	14.5 tps
3	Criar um plano de Voo	1091	0	18.2 tps	22.02 s	0.86 s	7.47 s	13.3 tps
4	Criar um plano de Voo	1101	0	19.9 tps	23.09 s	0.77 s	7.66 s	12.8 tps
5	Criar um plano de Voo	1439	0	23.6 tps	27.67 s	1.34 s	15.46 s	15.8 tps
6	Criar um plano de Voo	1339	0	23.3 tps	31.75 s	1.54 s	17.36 s	14.9 tps
7	Criar um plano de Voo	1555	0	19.6 tps	29.82 s	0.33 s	17.84 s	18.2 tps
8	Criar um plano de Voo	1681	0	26.7 tps	27.38 s	1.54 s	18.15 s	20.6 tps
9	Criar um plano de Voo	1449	0	24.0 tps	30.96 s	1.42 s	16.63 s	16.0 tps
10	Criar um plano de Voo	1389	0	23.4 tps	32.62 s	2.57 s	17.93 s	16.5 tps
11	Criar um plano de Voo	1730	0	28.2 tps	32.02 s	3.42 s	20.01 s	18.6 tps
12	Criar um plano de Voo	1742	0	30.7 tps	31.92 s	3.30 s	23.23 s	19.2 tps
13	Criar um plano de Voo	1478	0	24.4 tps	23.52 s	2.52 s	14.54 s	16.9 tps
14	Criar um plano de Voo	1607	0	27.3 tps	31.30 s	3.06 s	20.55 s	17.1 tps
15	Criar um plano de Voo	1508	0	26.5 tps	33.05 s	4.10 s	20.42 s	16.9 tps

A Tabela 5.15 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 50 clientes. Sendo assim, a taxa de envio aumenta até atingir a faixa de 20 a 30tps, apresentando variações em seu valor nas rodadas de testes, desta maneira o pico da taxa de envio foi de 30,7tps. À medida que a taxa de envio aumenta os valores de latência média passam a variar e aumentar, até atingir uma faixa de 15 a 25s, alcançando

o pico de 23,23s. A taxa de vazão aumenta até atingir a faixa de 12 a 21tps, apresentando variações de seu valor alcançando o pico de 20,6tps.

Tabela 5.16: Cenário de Teste 3 - Query 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	69	0	29.9 tps	0.09 s	0.01 s	0.04 s	29.5 tps
2	Consultar um Plano de voo	1909	0	63.5 tps	1.32 s	0.02 s	0.36 s	63.2 tps
3	Consultar um Plano de voo	1968	0	65.4 tps	1.21 s	0.01 s	0.33 s	65.2 tps
4	Consultar um Plano de voo	3435	0	57.2 tps	1.06 s	0.01 s	0.31 s	57.1 tps
5	Consultar um Plano de voo	4008	0	66.7 tps	1.19 s	0.01 s	0.39 s	66.6 tps
6	Consultar um Plano de voo	4259	0	70.9 tps	1.13 s	0.01 s	0.38 s	70.8 tps

A Tabela 5.16 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 5 clientes. Sendo assim, a taxa de envio começa com 29,9tps, valor que pode ser desconsiderado da análise, em seguida começa a subir alcançando valores próximos, até atingir o pico de 70,9tps. Nesse conjunto de testes, o melhor valor de latência média foi de 0,31s obtido na quarta rodada e o melhor valor de taxa de vazão foi de 70,8tps alcançado na sexta rodada.

Tabela 5.17: Cenário de Teste 3 - Query 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	317	0	7.0 tps	0.43 s	0.01 s	0.08 s	7.0 tps
2	Consultar um Plano de voo	3356	0	110.7 tps	1.02 s	0.14 s	0.66 s	109.9 tps
3	Consultar um Plano de voo	3427	0	113.7 tps	1.59 s	0.12 s	0.65 s	112.6 tps
4	Consultar um Plano de voo	6798	0	110.4 tps	2.23 s	0.02 s	0.65 s	110.4 tps
5	Consultar um Plano de voo	6856	0	114.0 tps	2.95 s	0.12 s	0.75 s	113.3 tps
6	Consultar um Plano de voo	7146	0	118.9 tps	1.59 s	0.15 s	0.80 s	118.1 tps

A Tabela 5.17 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 20 clientes. Sendo assim, a taxa de envio começa com um valor muito baixo que pode ser desconsiderado da análise, em seguida sobe com valores próximos na faixa de 110 a 120tps, sendo o maior valor obtido de 118,9tps. Nesse conjunto de testes, o melhor valor de latência média foi de 0,65s e o melhor valor de taxa de vazão foi de 118,1tps.

Tabela 5.18: Cenário de Teste 3 - Query 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	595	0	4.5 tps	1.88 s	0.01 s	0.42 s	4.5 tps
2	Consultar um Plano de voo	3338	0	107.9 tps	2.45 s	0.36 s	1.61 s	105.9 tps
3	Consultar um Plano de voo	3230	0	105.7 tps	2.40 s	0.41 s	1.63 s	103.6 tps
4	Consultar um Plano de voo	6244	0	102.9 tps	5.08 s	0.43 s	1.76 s	101.8 tps
5	Consultar um Plano de voo	6422	0	105.8 tps	3.24 s	0.49 s	1.82 s	104.4 tps
6	Consultar um Plano de voo	5243	0	82.4 tps	3.70 s	0.03 s	1.89 s	82.4 tps

A Tabela 5.18 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 50 clientes. Sendo assim, a taxa de envio começa com um valor que pode ser desconsiderado da análise dos testes, em seguida alcança seu pico no valor de 107,9tps. Nesse conjunto de testes, o melhor valor de latência média foi de 1,61s e o melhor valor de taxa de vazão foi de 105,9tps.

Criar plano de Voo

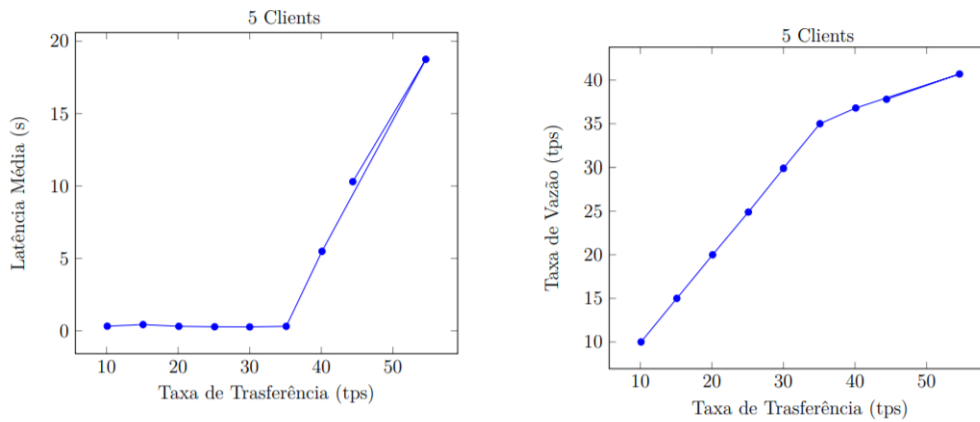


Figura 5.13: Criação de Plano de Voo com 5 Clientes

A Figura 5.13 apresenta dois gráficos dos testes da operação de criar plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com os valores abaixo de 1s e começa a subir até atingir o pico de 18,75s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 10tps e cresce até atingir o pico de 40,7tps.

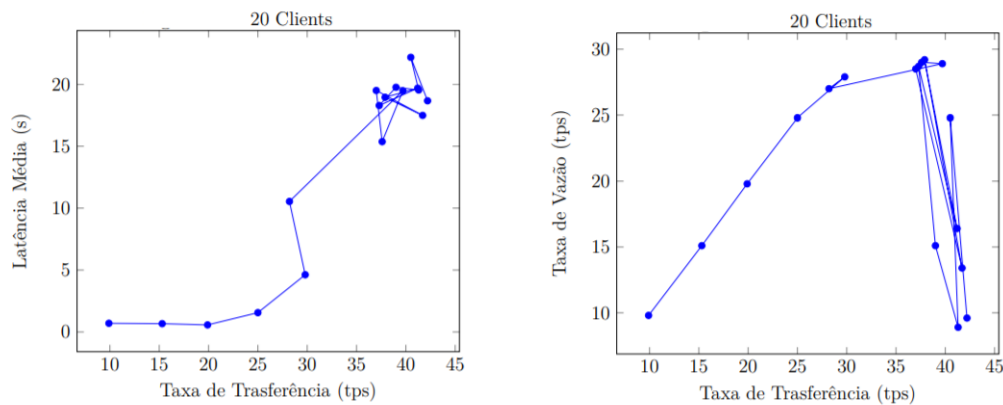


Figura 5.14: Criação de Plano de Voo com 20 Clientes

A Figura 5.14 apresenta dois gráficos dos testes da operação de criar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência média começa com os valores abaixo de 1s e começa a subir até uma faixa de 17 a 23 segundos, com o pico de 22,18s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa

com 9,3tps, cresce até atingir o pico de 29,2tps e fica alternando entre pontos de alta e baixa taxa de vazão.

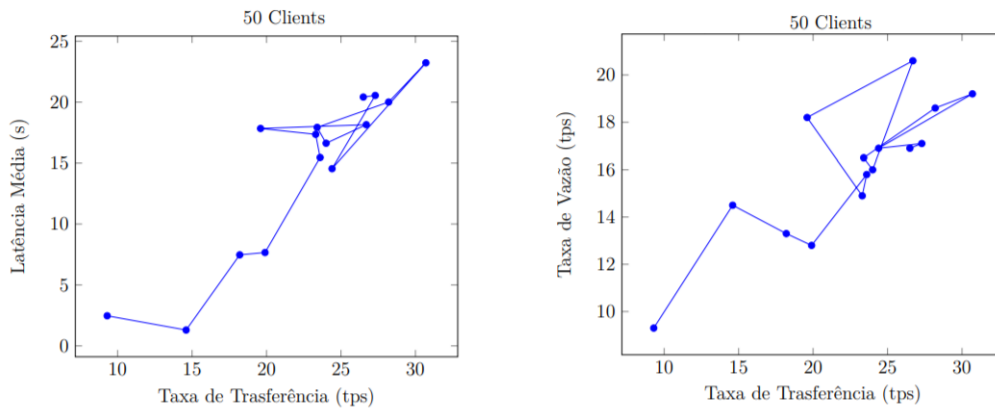


Figura 5.15: Criação de Plano de Voo com 50 Clientes

A Figura 5.15 apresenta dois gráficos dos testes da operação de criar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência média começa com 2,47s, cresce e fica variando na faixa de 15 a 24 segundos, até atingir seu pico de 23,23s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 9,3tps, aumenta e sofre variações dentro da faixa de 12 a 21tps, até atingir o pico de 20,6tps.

Consultar Plano de voo

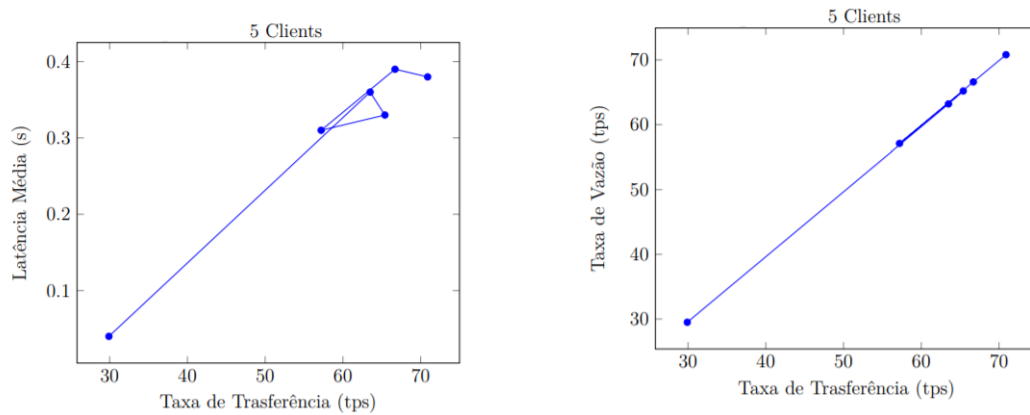


Figura 5.16: Consultar um plano de voo com 5 Clientes

A Figura 5.16 apresenta dois gráficos dos testes da operação de consultar plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,04s e cresce até a faixa de 0,31 a 0,40s alcançando o pico de 0,39s. O segundo gráfico consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 29,5tps e cresce até atingir o pico de 70,8tps.

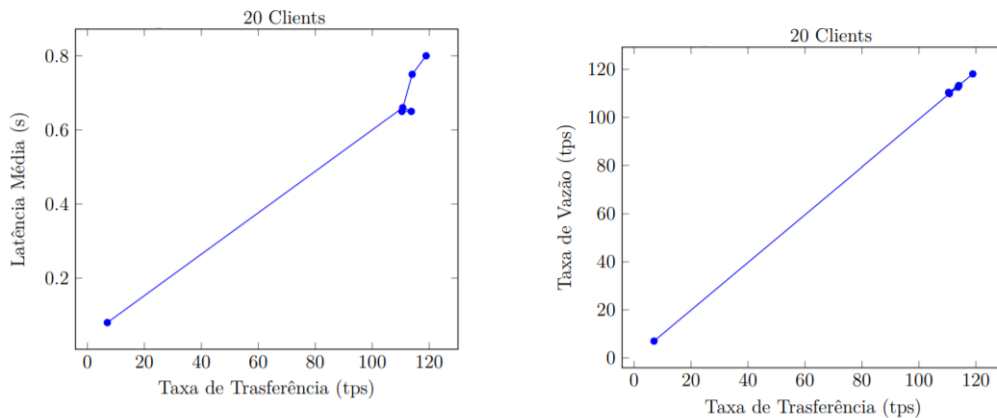


Figura 5.17: Consultar um plano de voo com 20 Clientes

A Figura 5.17 apresenta dois gráficos dos testes da operação de consultar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,08s e cresce até atingir o pico de 0,80s. O segundo gráfico consiste em um gráfico de taxa de transferência pela taxa de vazão que começa com 7,0tps e aumenta até atingir o pico de 118,1tps.

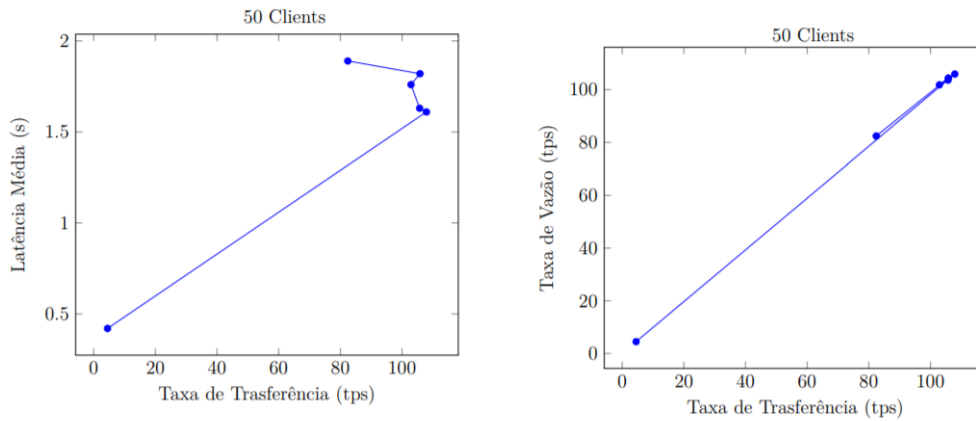


Figura 5.18: Consultar um plano de voo com 50 Clientes

A Figura 5.18 apresenta dois gráficos dos testes da operação de consultar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,42s e sobe até alcançar o pico do valor que é de 1,89s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa com 4,5tps, sobe até atingir o pico de 105,9tps e depois desce até o valor de 82,4tps na última rodada de testes.

5.2.6 Cenário de Teste 4

O Cenário de Teste 4 compreende a configuração com três Organizações participantes e cada possuindo três *Peers*.

Tabela 5.19: Cenário de Teste 4 - 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	600	0	10.1 tps	0.76 s	0.23 s	0.38 s	10.0 tps
2	Criar um Plano de Voo	901	0	15.0 tps	0.79 s	0.16 s	0.47 s	15.0 tps
3	Criar um plano de Voo	1201	0	20.0 tps	0.69 s	0.13 s	0.36 s	19.9 tps
4	Criar um plano de Voo	1501	0	25.0 tps	0.68 s	0.17 s	0.39 s	24.9 tps
5	Criar um plano de Voo	1800	0	30.1 tps	1.68 s	0.14 s	0.52 s	30.0 tps
6	Criar um plano de Voo	2100	0	35.1 tps	13.97 s	0.36 s	8.20 s	31.4 tps
7	Criar um plano de Voo	2400	0	40.1 tps	24.24 s	0.31 s	14.95 s	32.5 tps
8	Criar um plano de Voo	762	1772	42.1 tps	30.02 s	0.32 s	13.97 s	8.5 tps
9	Criar um plano de Voo	628	2156	46.4 tps	30.17 s	0.82 s	15.35 s	7.0 tps
10	Criar um plano de Voo	526	2442	49.2 tps	29.96 s	0.83 s	15.50 s	5.8 tps
11	Criar um plano de Voo	565	2396	49.3 tps	30.01 s	1.18 s	15.69 s	6.3 tps
12	Criar um plano de Voo	536	2440	49.5 tps	29.89 s	1.82 s	16.45 s	5.9 tps
13	Criar um plano de Voo	519	2494	49.9 tps	29.96 s	1.96 s	16.43 s	5.7 tps
14	Criar um plano de Voo	501	2535	50.5 tps	30.03 s	4.55 s	17.72 s	5.6 tps
15	Criar um plano de Voo	501	2463	49.3 tps	30.00 s	2.50 s	16.39 s	5.6 tps
16	Criar um plano de Voo	522	2505	50.3 tps	30.01 s	1.99 s	16.01 s	5.8 tps
17	Criar um plano de Voo	481	2544	50.4 tps	30.15 s	1.68 s	15.79 s	5.3 tps

A Tabela 5.19 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 5 clientes. Sendo assim, a cada rodada de testes a taxa de envio aumenta, até atingir o pico de 50,5tps. À medida que a taxa de envio sobe os valores de

latência média passam a aumentar até atingir o pico de 17,72s, enquanto isso a taxa de vazão cresce até atingir o pico do seu valor que é de 32,5tps.

Tabela 5.20: Cenário de Teste 4 - 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	601	0	9.7 tps	3.41 s	0.24 s	0.68 s	9.7 tps
2	Criar um Plano de Voo	900	0	15.3 tps	5.09 s	0.49 s	1.75 s	15.1 tps
3	Criar um plano de Voo	1200	0	20.2 tps	7.71 s	1.27 s	4.23 s	19.1 tps
4	Criar um plano de Voo	1491	0	23.3 tps	22.21 s	0.71 s	12.99 s	19.3 tps
5	Criar um plano de Voo	1650	104	28.6 tps	31.38 s	1.16 s	21.26 s	18.7 tps
6	Criar um plano de Voo	787	1284	34.0 tps	30.25 s	1.28 s	15.83 s	8.7 tps
7	Criar um plano de Voo	752	1262	33.3 tps	30.33 s	1.61 s	17.14 s	8.3 tps
8	Criar um plano de Voo	513	1813	38.2 tps	30.16 s	1.53 s	16.49 s	5.6 tps
9	Criar um plano de Voo	679	1258	31.3 tps	30.19 s	9.25 s	18.91 s	7.4 tps
10	Criar um plano de Voo	593	1557	35.1 tps	30.32 s	1.96 s	17.62 s	6.5 tps
11	Criar um plano de Voo	550	1802	38.8 tps	29.96 s	2.02 s	17.23 s	6.1 tps
12	Criar um plano de Voo	282	1852	33.9 tps	30.64 s	18.12 s	24.99 s	3.0 tps
13	Criar um plano de Voo	460	1954	39.6 tps	30.27 s	2.13 s	17.22 s	5.1 tps
14	Criar um plano de Voo	219	2037	36.2 tps	30.13 s	18.24 s	24.83 s	2.4 tps
15	Criar um plano de Voo	509	1774	37.6 tps	30.20 s	2.08 s	18.39 s	5.6 tps
16	Criar um plano de Voo	450	2071	41.5 tps	30.16 s	2.15 s	17.40 s	5.0 tps
17	Criar um plano de Voo	106	2197	37.1 tps	30.83 s	25.63 s	28.40 s	1.2 tps

A Tabela 5.20 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 20 clientes. Sendo assim, a cada rodada de testes a taxa de envio aumenta, até atingir a faixa de 30 a 42tps, desta maneira o pico do valor foi de 41,5tps. À medida que a taxa de envio aumenta os valores de latência média passam a variar e subir, até atingir a faixa de 15 a 25s, alcançando o pico de 24,99s, enquanto isso a taxa de vazão aumenta até atingir o pico de 19,3tps e depois decai.

Tabela 5.21: Cenário de Teste 4 - 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Criar um Plano de voo	477	121	9.1 tps	25.79 s	1.50 s	9.53 s	5.7 tps
2	Criar um Plano de Voo	666	123	12.0 tps	32.11 s	1.85 s	10.03 s	7.7 tps
3	Criar um plano de Voo	766	309	16.1 tps	30.44 s	1.42 s	14.39 s	8.5 tps
4	Criar um plano de Voo	687	351	15.2 tps	23.24 s	1.40 s	12.82 s	7.5 tps
5	Criar um plano de Voo	777	488	19.0 tps	34.31 s	2.23 s	21.00 s	8.2 tps
6	Criar um plano de Voo	473	911	20.4 tps	32.56 s	2.10 s	18.50 s	4.8 tps
7	Criar um plano de Voo	444	1091	23.7 tps	31.07 s	2.96 s	18.58 s	4.7 tps
8	Criar um plano de Voo	395	1089	23.4 tps	31.67 s	2.85 s	18.07 s	4.2 tps
9	Criar um plano de Voo	813	486	19.6 tps	33.22 s	4.00 s	20.33 s	9.4 tps
10	Criar um plano de Voo	519	835	20.6 tps	33.52 s	3.54 s	21.09 s	5.4 tps

A Tabela 5.21 apresenta os resultados obtidos na execução dos testes de criação de plano de voo para 50 clientes. Sendo assim, a cada rodada de testes a taxa de envio aumenta até atingir a faixa de 15 a 25tps, desta maneira o pico do valor foi de 23,7tps. À medida que a taxa de envio sobe os valores de latência média passam a aumentar e variar até atingir o pico de 21,09s, enquanto isso, a taxa de vazão sofre algumas variações dentro da faixa de 4 a 10tps, atingindo o pico de 9,4tps.

Tabela 5.22: Cenário de Teste 4 - Query 5 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	80	0	1.6 tps	0.15 s	0.01 s	0.04 s	1.6 tps
2	Consultar um Plano de voo	1205	0	40.0 tps	4.34 s	0.02 s	0.64 s	39.7 tps
3	Consultar um Plano de voo	1369	0	45.7 tps	1.68 s	0.03 s	0.51 s	45.3 tps
4	Consultar um Plano de voo	2788	0	46.4 tps	1.47 s	0.02 s	0.38 s	46.3 tps
5	Consultar um Plano de voo	2894	0	48.2 tps	4.58 s	0.04 s	0.60 s	47.9 tps
6	Consultar um Plano de voo	2928	0	48.8 tps	7.24 s	0.01 s	0.72 s	48.6 tps

A Tabela 5.22 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 5 clientes. Sendo assim, a taxa de envio começa com um valor que pode ser desconsiderado da análise dos testes, em seguida cresce até atingir o pico no valor de 48,8tps. Nesse conjunto de testes, o melhor valor de latência foi de 0,38s e o melhor valor de taxa de vazão foi de 48,6tps.

Tabela 5.23: Cenário de Teste 4 - Query 20 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	301	0	7.8 tps	0.50 s	0.01 s	0.13 s	7.8 tps
2	Consultar um Plano de voo	2977	0	89.5 tps	1.45 s	0.04 s	0.77 s	89.4 tps
3	Consultar um Plano de voo	2921	0	91.0 tps	1.89 s	0.04 s	0.79 s	90.9 tps
4	Consultar um Plano de voo	5795	0	93.3 tps	2.68 s	0.06 s	0.77 s	93.0 tps
5	Consultar um Plano de voo	5855	0	96.9 tps	3.67 s	0.16 s	0.89 s	96.6 tps
6	Consultar um Plano de voo	6119	0	101.2 tps	2.71 s	0.15 s	0.88 s	100.8 tps

A Tabela 5.23 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 20 clientes. Sendo assim, a taxa de envio começa com um valor baixo que pode ser desconsiderado da análise, em seguida cresce até atingir o pico no valor de 101,2tps. Nesse conjunto de testes, o melhor valor de latência média foi de 0,77s e o melhor valor de taxa de vazão foi de 100,8tps.

Tabela 5.24: Cenário de Teste 4 - Query 50 Clientes

Round	Nome	Sucesso	Falha	Taxa de Envio	Latência Máxima	Latência Mínima	Latência Média	Taxa de Vazão
1	Consultar um Plano de voo	545	0	4.6 tps	4.08 s	0.01 s	0.64 s	4.6 tps
2	Consultar um Plano de voo	983	0	24.6 tps	12.61 s	0.11 s	4.45 s	24.5 tps
3	Consultar um Plano de voo	1590	0	42.3 tps	7.06 s	0.09 s	3.36 s	42.2 tps
4	Consultar um Plano de voo	2003	0	32.2 tps	13.87 s	1.42 s	5.41 s	29.2 tps
5	Consultar um Plano de voo	2036	0	31.0 tps	10.03 s	0.06 s	4.84 s	31.0 tps
6	Consultar um Plano de voo	2687	0	42.1 tps	7.76 s	0.56 s	4.12 s	41.8 tps

A Tabela 5.24 apresenta os resultados obtidos na execução dos testes de consulta de plano de voo para 50 clientes. Sendo assim, a taxa de envio começa com um valor baixo que pode ser desconsiderado da análise, em seguida cresce até atingir o pico no valor de 42,3tps. Nesse conjunto de testes, o melhor valor de latência média foi de 3,36s e o melhor valor de taxa de vazão foi de 42,2tps.

Criar plano de Voo

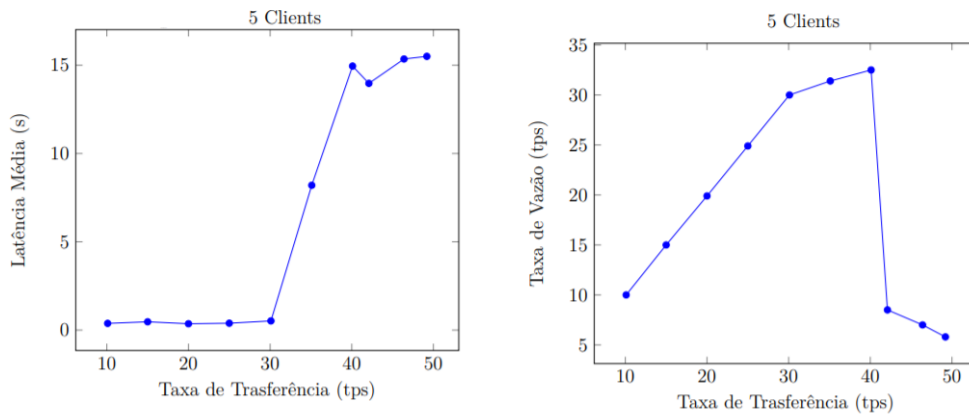


Figura 5.19: Criação de Plano de Voo com 5 Clientes

A Figura 5.19 apresenta dois gráficos dos testes da operação criar plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível perceber que a latência média começa com valores abaixo de 1s e começa a subir até atingir o pico de 17,72s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 10tps e começa a aumentar até atingir o pico de 32,5tps, entretanto após atingir seu maior valor, a taxa de vazão decaí e se mantém baixa em uma faixa entre 5 e 10tps.

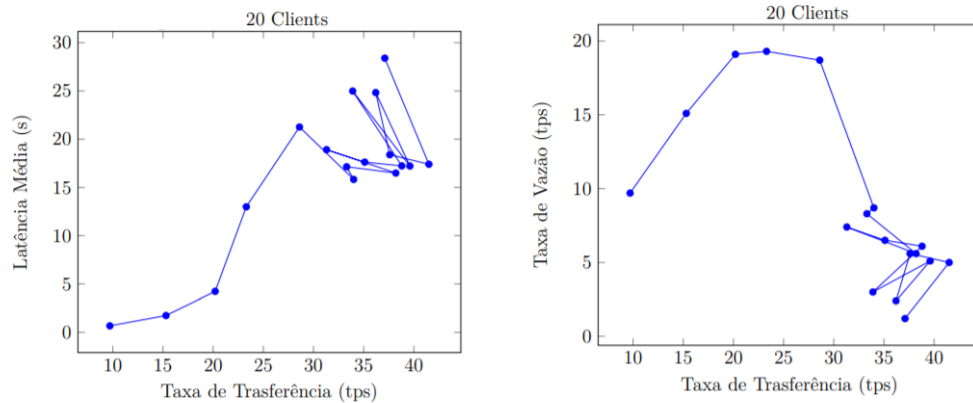


Figura 5.20: Criação de Plano de Voo com 20 Clientes

A Figura 5.20 apresenta dois gráficos dos testes da operação de criar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível perceber que a latência média começa com 0,68s e sobe, sofrendo variações dentro da faixa de 15 a 30tps, até atingir o pico de 28,40s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 9,7tps e

aumenta até atingir o pico de 19,3tps, entretanto após alcançar esse valor, a taxa começa a decair e se mantém baixa na faixa de 1 a 10tps.

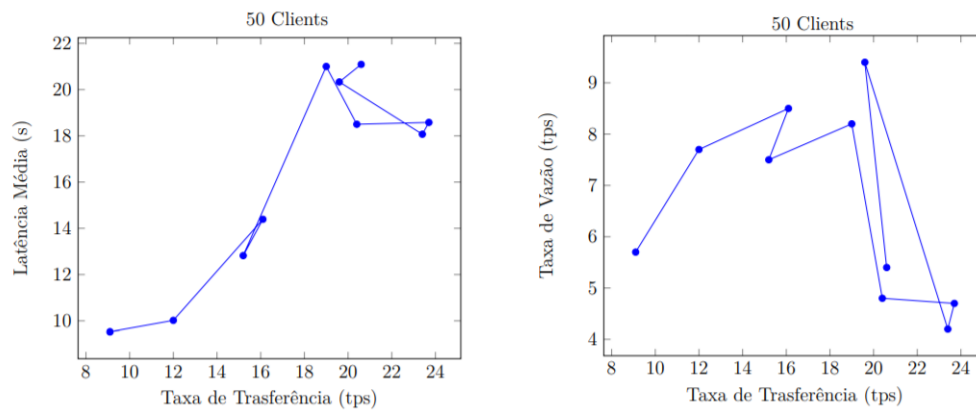


Figura 5.21: Criação de Plano de Voo com 50 Clientes

A Figura 5.21 apresenta dois gráficos dos testes da operação de criar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível perceber que a latência média começa com o valor de 9,53s e cresce até atingir o pico de 21.09s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa em 5,7tps e aumenta até atingir o pico de 9,4tps.

Consultar Plano de voo

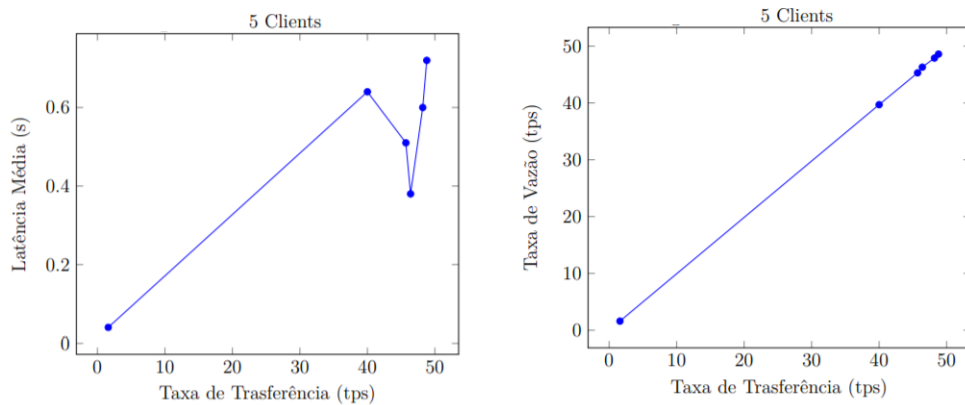


Figura 5.22: Consultar um plano de voo com 5 Clientes

A Figura 5.22 apresenta dois gráficos dos testes da operação de consultar plano de voo para 5 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,04s e sobe até alcançar o pico do valor que é de 0,72s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa com 1,6tps e sobe até atingir o pico de 48,6tps.

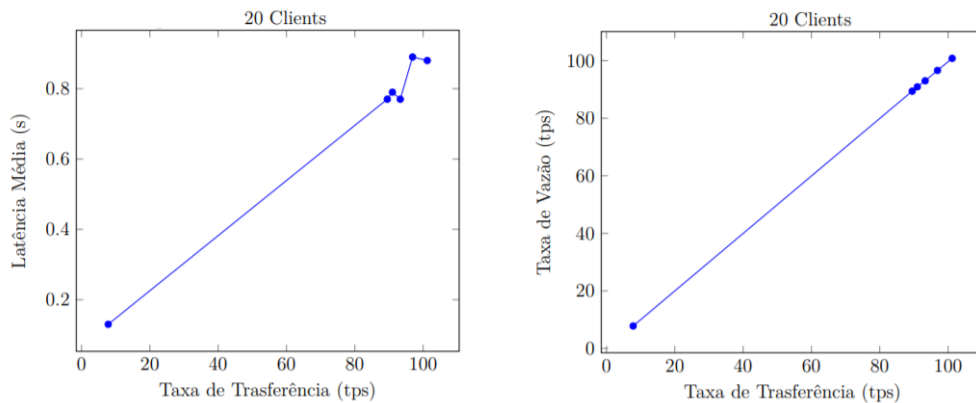


Figura 5.23: Consultar um plano de voo com 20 Clientes

A Figura 5.23 apresenta dois gráficos dos testes da operação de consultar plano de voo para 20 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,13s e sobe até alcançar o pico do valor que é de 0,89s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa com 7,8tps e sobe até atingir o pico de 100,8tps.

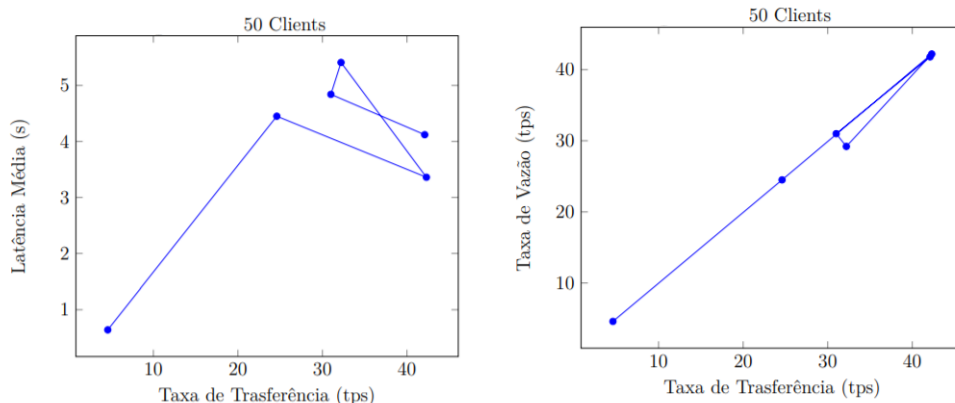


Figura 5.24: Consultar um plano de voo com 50 Clientes

A Figura 5.24 apresenta dois gráficos dos testes da operação de consultar plano de voo para 50 clientes. O primeiro consiste em um gráfico de taxa de transferência pela latência média, dessa maneira é possível observar que a latência começa com 0,64s e sobe até alcançar o pico do valor que é de 5,41s. O segundo consiste em um gráfico de taxa de transferência pela taxa de vazão que começa com 4,6tps e sobe até atingir o pico de 42,2tps.

5.3 Discussão e Análise dos Resultados

A seção a seguir contém uma análise detalhada dos resultados obtidos em relação as operações de criação e consulta de plano de voo, nos diversos cenários de testes.

5.3.1 Criação de um plano de voo

Tabela 5.25: Comparativo de melhores casos de performance

Criação do plano de voo	2-1	3-1	3-2	3-3
5 Clients	78.2 tps e 0.66 s	62.1 tps e 3.63 s	39.9 tps e 9.86 s	31.4 tps e 8.20 s
20 Clients	59.8 tps e 2.33s	47.3 tps e 5.41 s	39.9 tps e 9.86 s	19.1 tps e 4.23 s
50 Clients	49.1 tps e 6.53s	36.3 tps e 8.68 s	29.2 tps e 18.95 s	19.3 tps e 12.99 s

A Tabela 5.25 apresenta um comparativo dos melhores resultados de performance da operação de criação de plano de voo, em relação aos cenários de testes e a variação de clientes. Desta maneira, é possível perceber que o primeiro caso de teste (2-1) apresenta os melhores resultados em comparação com os outros. Entretanto, é um cenário base para realizar as comparações com os outros casos e para analisar o sistema, sendo assim os melhores resultados acontecem no caso de teste 3-1.

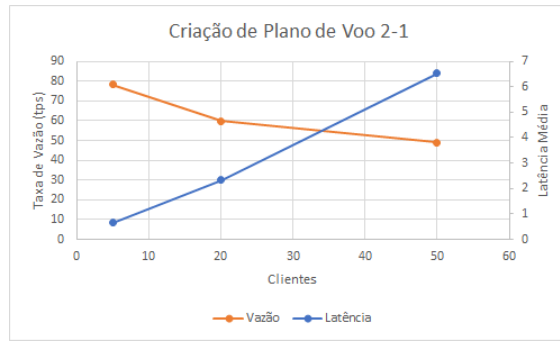


Figura 5.25: Criação de Plano de Voo com 2-1

A Figura 5.25 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 2 Organizações e 1 peer de cada. Desta maneira, é possível perceber que o tempo de latência, em segundos, foi aumentando, enquanto a vazão dos dados decaiu. Esse fato ocorre devido à quantidade de clientes que estão enviando transações dentro da rede, esperando a resposta do protocolo de consenso, por conta da fila de espera para registrar as transações nos peers. Sendo assim, esse cenário de teste alcança a melhor performance com 5 clientes, graças a taxa de vazão em valores maiores e a latência média em valores menores, comparados com 20 e 50 clientes.

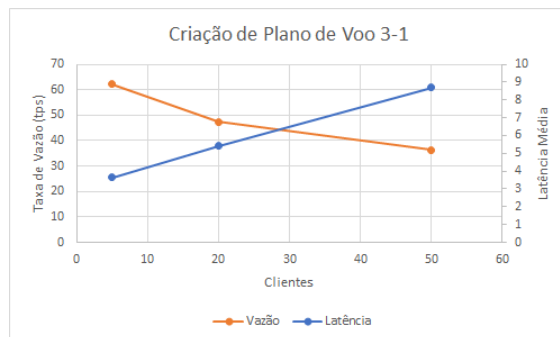


Figura 5.26: Criação de Plano de Voo com 3-1

A Figura 5.26 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 1 peer de cada. Desta maneira, é possível perceber que o tempo de latência teve um acréscimo, enquanto a vazão dos dados decaiu. Esse fato ocorre devido à quantidade de clientes que estão enviando transações para a rede e esperando pela resposta do protocolo de consenso para validar as mesmas, por causa da fila de espera para registrar as transações nos peers, em comparação com o cenário de testes 2-1, esse aumento da latência é maior devido ao aumento da quantidade de peers totais da rede. Sendo assim, esse cenário de teste alcança a melhor performance com 5 clientes, com uma taxa de vazão de 62,1tps e

uma latência média de 3,63s, valores menores comparados com 20 e 50 clientes no mesmo cenário de teste.

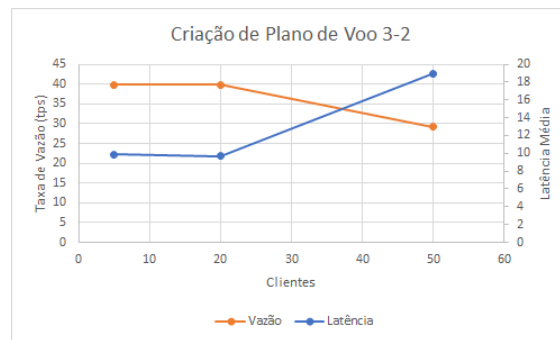


Figura 5.27: Criação de Plano de Voo com 3-2

A Figura 5.27 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 2 peers de cada. Desta maneira, é possível perceber que a latência média e a taxa de vazão se mantêm com o mesmo valor, nos casos de 5 e 20 clientes. Logo após ocorre uma queda na taxa de vazão e um aumento na latência média. Essa fato ocorre devido à quantidade de clientes que estão enviando transações para a rede e esperando pela resposta do protocolo de consenso para validar as mesmas, porém com 50 clientes esse aumento de latência fica acentuado, graças a quantidade de peers da rede. Sendo assim, esse cenário de teste alcança a melhor performance com 5 e 20 clientes, com uma taxa de vazão de 39,9tps e uma latência média de 9,86s, valores menores que os obtidos com 50 clientes no mesmo cenário de teste.

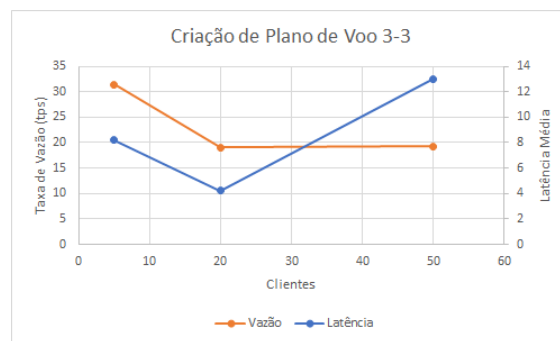


Figura 5.28: Criação de Plano de Voo com 3-3

A Figura 5.28 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 3 peers de cada. Desta maneira é possível perceber que a latência média começa alta, diminui com 20 clientes e depois aumenta de novo, até atingir seu maior valor. Entretanto a taxa de vazão começa no melhor valor, atingi o menor valor com 20 clientes e sobe um

pouco com 50 clientes. Esse fato ocorre devido à quantidade de clientes que estão enviando as transações e esperando pela resposta do protocolo de consenso para validar as mesmas. Sendo assim, esse cenário de teste alcança a melhor performance com 5 clientes, com uma taxa de vazão de 31,4tps e uma latência média de 8,20s, valores de performance melhores que os obtidos com 20 e 50 clientes no mesmo cenário de teste.

5.3.2 Consulta a um plano de voo

Tabela 5.26: Comparativo de melhores casos de performance

Consulta do plano de voo	2-1	3-1	3-2	3-3
5 Clients	69.7 tps e 0.26 s	60.3 tps e 0.41 s	70.8 tps e 0.38 s	48.6 tps e 0.72 s
20 Clients	68.6 tps e 0.97 s	127.8 tps e 0.59 s	113.3 tps e 0.75 s	100.8 tps e 0.88 s
50 Clients	139.0 tps e 1.07s	108.3 tps e 1.65 s	105.9 tps e 1.61 s	42.2 tps e 3.36 s

A Tabela 5.26 apresenta um comparativo dos melhores resultados de performance da operação de consultar plano de voo, em relação aos cenários de testes e a variação de clientes. Desta maneira, é possível perceber que os melhores indicadores variam de acordo com a quantidade de clientes da rede, com o melhor caso para 5 clientes no cenário 3-2, já para 20 clientes o melhor caso é no cenário 3-1 e para 50 clientes o melhor caso é no cenário 3-1, desconsiderando o caso base.

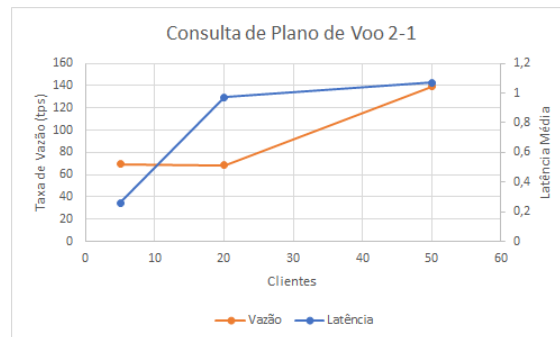


Figura 5.29: Consulta de Plano de Voo com 2-1

A Figura 5.29 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 2 Organizações e 1 peer de cada. Desta maneira, é possível perceber que a latência média sofre um aumento entre os testes com 5 e 20 clientes, e um leve aumento de 20 a 50, atingindo o pico de 1,07s de latência média. Enquanto isso a taxa de vazão decaí de 5 a 20 clientes e tem um aumento atingido o pico de 139,0tps. Esse fato ocorre devido a essa operação de consultar não precisar ficar em uma fila esperando para registrar alguma transação e sim apenas uma consulta aos dados presentes nos peers.

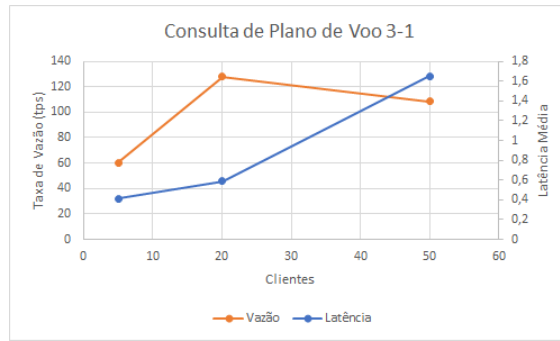


Figura 5.30: Consulta de Plano de Voo com 3-1

A Figura 5.30 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 1 peer de cada. Desta maneira é possível perceber que a latência média tem um pequeno aumento entre 5 e 20 clientes e sobe entre 20 a 50 atingindo seu maior valor. Enquanto isso a taxa de vazão tem um aumento entre 5 e 20 clientes, atingindo o pico de 127,8tps e sofre uma queda entre 20 a 50 clientes.

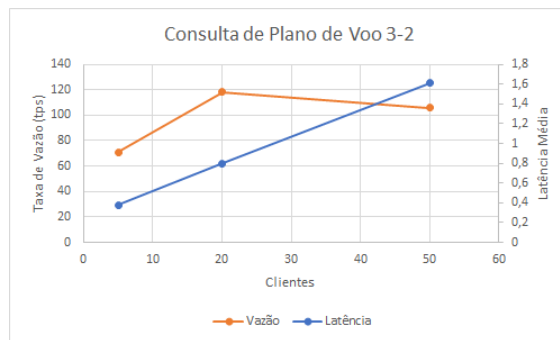


Figura 5.31: Consulta de Plano de Voo com 3-2

A Figura 5.31 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 2 peers de cada. Desta maneira é possível perceber que a latência média aumentou de 5 a 20 clientes e aumentou consideravelmente de 20 a 50 clientes. Enquanto isso a taxa de vazão tem um aumento de 5 a 20 clientes, atingindo assim o seu pico de 113,3tps e entre 20 e 50 clientes ocorre uma queda.

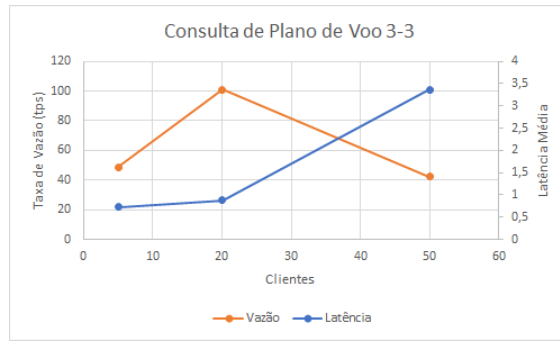


Figura 5.32: Consulta de Plano de Voo com 3-3

A Figura 5.32 apresenta um gráfico de latência média e taxa de vazão em variação da quantidade de clientes presentes na rede Blockchain com a configuração de 3 Organizações e 3 peers de cada. Desta maneira é possível perceber que a latência média aumentou entre 5 e 20 clientes e aumentou consideravelmente entre 20 e 50 clientes. Enquanto isso a taxa de vazão teve um aumento de 5 a 20 clientes, atingindo o pico de 100,8 tps e teve uma queda entre 20 e 50 clientes.

Capítulo 6

Conclusão e Trabalhos Futuros

Este capítulo apresenta as contribuições, conclusões à respeito da aplicação blockchain desenvolvida para os serviços do *SWIM Registry Brazil*, em específico o serviço de Plano de Voo. Além de possíveis trabalhos futuros e de considerações finais.

6.1 Contribuições

O *SWIM Registry Brazil* ainda está se desenvolvendo, com o intuito de trazer grandes benefícios aos *stakeholders* do sistema, visando aprimorar a visibilidade das informações relacionadas ao *ATM* e serviços disponíveis no *SWIM*. Desta forma, neste trabalho é proposto uma solução para garantir uma maior segurança das informações e uma maior autenticidade dos usuários do sistema.

Neste contexto, foi apresentado uma solução em *blockchain* para os serviços do *SWIM Registry Brazil*, no caso o serviço de plano de voo. Desta maneira, foi modelado as informações de plano de voo para a estrutura básica do *blockchain*, respeitando suas características. A partir dessa modelagem foi possível desenvolver uma aplicação *blockchain* que execute o serviço de plano de voo, desde a etapa de inserção do plano de voo, consulta do mesmo, além de determinar as permissões específicas que cada *stakeholder* possui. Sendo assim, garantindo uma maior segurança as informações e maior autenticidade aos usuários do sistema.

Os resultados obtidos nos cenários de testes são promissores, na medida que foi possível: 1) implementar o serviço de plano de voo, de forma que atenda as necessidades do *SWIM Registry Brazil*; 2) obter bons resultados, baseados nos indicadores de performance, sobre a rede *blockchain* criada para atender o serviço de plano de voo. De acordo com a execução dos testes nos diversos cenários, o desempenho obtido foi influenciado pela máquina na qual o projeto foi realizado, além da quantidade de *peers* presentes em cada cenário. Deste maneira, o esforço computacional vai aumentando à medida que aumenta a

quantidade de *peers* e clientes presentes na rede, o que seria um problema se a quantidade de clientes alcance um valor na casa de dezena de milhares de usuários do sistema.

Após a conclusão do presente trabalho, é possível listar de forma objetiva as contribuições do estudo realizado que são:

- Implementação da tecnologia *blockchain* no serviço do *SWIM Registry Brazil*;
- Desenvolvimento da arquitetura dos serviços SWIM para a criação dos mesmos com a tecnologia blockchain;
- Desenvolvimento de uma aplicação *Blockchain* para o serviço de Plano de Voo;
- Identificação dos parâmetros ótimos para os componentes da aplicação *blockchain*.

Para a confirmação das contribuições acima citadas, foram desenvolvidas as seguintes atividades:

- Estudo dos serviços do SWIM Registry Brazil;
- Estudo sobre blockchain;
- Definição da arquitetura para implementação dos serviços do SWIM Registry Brazil em uma aplicação blockchain;
- Modelagem da estrutura básica do blockchain para receber as informações de plano de voo;
- Implementação de um sistema utilizando a arquitetura proposta;
- Simulação das operações do serviço de plano de voo em diferentes cenários de testes;
- Definição dos indicadores de performance para avaliar e comparar os resultados obtidos através dos experimentos;
- Seleção dos melhores parâmetros de acordo com a análise dos resultados obtidos.

6.2 Trabalhos Futuros

A partir das contribuições obtidas, é possível desenvolver pesquisas futuras:

- Criação de outros serviços do SWIM Registry Brazil utilizando a tecnologia *blockchain*;
- Desenvolver novos cenários de testes que possam obter melhores resultados de performance em relação aos obtidos;

- A utilização de aprendizado de máquina para promover uma melhora na performance obtida nos testes realizados.
- Utilização de novos frameworks do blockchain para realização de novos testes de performance;
- Utilização de novos protocolos de consenso para realização de novos testes de performance;

6.3 Considerações Finais

Por fim, pretende-se utilizar este trabalho para a criação dos serviços do *SWIM Registry Brazil*. A implementação do sistema desenvolvido, assim como a utilização do mesmo pelos agentes interessados representa o maior desafio para o prosseguimento do presente trabalho, dependendo de um interesse maior dos *stakeholders* do *SWIM Registry Brazil*.

Além disso, este trabalho apresentou resultados significativos que produziram dois artigos, que são:

- BONOMO, IGOR S. ; BARBOSA, IURI R. ; MONTEIRO, LUCAS ; BASSETTO, CAMILA ; DE BARROS BARRETO, ALEXANDRE ; BORGES, VINICIUS R.P.; Weigang, Li . **Development of SWIM Registry for Air Traffic Management with the Blockchain Support.** In: 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC), 2018, Maui. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). USA: IEEE ITSC, 2018. p. 3544-3549.
- BONOMO, I. S. ; BARBOSA, I. S. R. ; MONTEIRO, L. ; WEIGANG, L. ; BORGES, V. R. P.; BASSETTO, C.; BARRETO, A. B. . **Utilização do Blockchain para suporte do controle de tráfego aéreo.** In: Simpósio de Pesquisa em Transporte Aéreo, 2018, Sao Paulo. SITRAER. Sao Paulo: SBTA, 2018.

Desta maneira, espera-se produzir mais artigos com os resultados apresentados nesta dissertação.

Referências

- [1] Wüst, Karl e Arthur Gervais: *Do you need a blockchain?* Em *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, páginas 45–54. IEEE, 2018. vii, 23, 29
- [2] Aviação Civil, title = Painel de Indicadores do Transporte Aéreo 2017, url = <http://www.anac.gov.br/assuntos/dados-e-estatisticas/mercado-de-transporte-aereo/painel-de-indicadores-do-transporte-aereo> Agência Nacional de. 1
- [3] Espaço Aéreo, Departamento de Controle do: *Departamento de controle do espaço aéreo: Requisitos para operação vfr ou ifr em aeródromos*. <https://publicacoes.decea.gov.br/?i=publicacao&id=4915>. 1
- [4] Espaço Aéreo, Departamento de Controle do: *Departamento de controle do espaço aéreo: Conceção operacional atm nacional*. <https://publicacoes.decea.gov.br/?i=publicacao&id=3678>. 1, 2
- [5] Transportes, Portos e Aviação Civil, Secretaria Nacional de Aviação Civil Departamento de Planejamento e Gestão Aeroportuária Ministério dos: *Projeção de demanda da aviação civil 2017-2037: Passageiros, aeronaves e carga*, 2018. <http://www.transportes.gov.br/images/AEROPORTOS/ProjDemandaPress.pdf>. 2
- [6] Organization, International Civil Aviation: *Global Air Navigation Plan Fourth Edition*. 2013. 2
- [7] Leite, Alessandro Ferreira, Li Weigang, Jose Alexandre Fregnani e Italo Romani de Oliveira: *Big data management and processing in the context of the system wide information management*. Em *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, páginas 1–8, 2017. 2
- [8] Weigang, Li, Alessandro F Leite, Vitor F Ribeiro, Jose A Fregnani e Italo R de Oliveira: *Towards intelligent system wide information management for air traffic management*. Em *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, páginas 584–593. Springer, 2017. 2
- [9] Organization, International Civil Aviation: *Manual on system wide information management (swim) concept, (doc 10039)*. International Civil Aviation Organization (ICAO), 2015. 2, 6, 7, 8, 9
- [10] Nakamoto, Satoshi: *Bitcoin: A peer-to-peer electronic cash system*. 2008. 3, 14, 18

- [11] Crosby, Michael, Pradan Pattanayak, Sanjeev Verma e Vignesh Kalyanaraman: *Blockchain technology: Beyond bitcoin*. Applied Innovation, 2:6–10, 2016. 3
- [12] Jiang, S., J. Cao, H. Wu, Y. Yang, M. Ma e J. He: *Blochic: A blockchain-based platform for healthcare information exchange*. Em *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, páginas 49–56, June 2018. 3, 24
- [13] Kshetri, Nir e Jeffrey Voas: *Blockchain-enabled e-voting*. IEEE Software, 35(4):95–99, 2018. 3
- [14] Outchakoucht, Aissam, ES Samaali Hamza e Jean Philippe Leory: *Dynamic access control policy based on blockchain and machine learning for the internet of things*. International Journal of Advanced Computer Science and Applications (IJACSA), 8(7):417–424, 2017. 3, 24
- [15] Barbosa, I. S. R., L. B. Monteiro, Li Weigang, C. Bassetto, F. S. Queiroz, J. S. Santos e A. B. Barreto: *Desenvolvimento do swim registry brasileiro*. Em *Simpósio de Transporte Aéreo*, páginas 1–6, 2017. 8
- [16] Aeronáutica, Brasil Comando da: *Plano de voo*. 2016. 13
- [17] Swan, Melanie: *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015. 15
- [18] Antonopoulos, Andreas M: *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", 2014. 16
- [19] Lamport, Leslie, Robert Shostak e Marshall Pease: *The byzantine generals problem*. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3):382–401, 1982. 17
- [20] Castro, Miguel, Barbara Liskov *et al.*: *Practical byzantine fault tolerance*. Em *OSDI*, volume 99, páginas 173–186, 1999. 18
- [21] King, Sunny e Scott Nadal: *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake*. self-published paper, August, 19, 2012. 19
- [22] Szabo, Nick: *The idea of smart contracts*. Nick Szabo's Papers and Concise Tutorials, 6, 1997. 21
- [23] Christidis, K. e M. Devetsikiotis: *Blockchains and smart contracts for the internet of things*. IEEE Access, 4:2292–2303, 2016, ISSN 2169-3536. 21, 22
- [24] Korpela, Kari, Jukka Hallikas e Tomi Dahlberg: *Digital supply chain transformation toward blockchain integration*. Em *proceedings of the 50th Hawaii international conference on system sciences*, 2017. 22
- [25] Savelyev, Alexander: *Copyright in the blockchain era: Promises and challenges*. Computer law & security review, 34(3):550–561, 2018. 22

- [26] Azaria, Asaph, Ariel Ekblaw, Thiago Vieira e Andrew Lippman: *Medrec: Using blockchain for medical data access and permission management*. Em *IEEE International Conference on Open and Big Data (OBD)*, páginas 25–30, 2016. 22
- [27] Peterson, Kevin, Rammohan Deeduvanu, Pradip Kanjamala e Kelly Boles: *A blockchain-based approach to health information exchange networks*. Em *NIST Workshop Blockchain Healthcare*, volume 1, páginas 1–10, 2016. 22
- [28] Xu, Xiwei, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran e Shiping Chen: *The blockchain as a software connector*. Em *IEEE/IFIP Conference on Software Architecture*, páginas 182–191, 2016. 22
- [29] Lone, Auqib Hamid e Roohie Naaz Mir: *Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer*. *Digital Investigation*, 28:44–55, 2019. 23
- [30] Dinh, Thang N e My T Thai: *Ai and blockchain: A disruptive integration*. *Computer*, 51(9):48–53, 2018. 24, 25
- [31] Clementi, Marina Dehez, Mohamed Kaafar, Nicolas Larrieu, Hassan Asghar e Emmanuel Lochin: *When air traffic management meets blockchain technology: a blockchain-based concept for securing the sharing of flight data*. 2019. 25
- [32] Reisman, Ronald J: *Air traffic management blockchain infrastructure for security, authentication, and privacy*. 2019. 25
- [33] Bonomo, Igor S, Iuri R Barbosa, Lucas Monteiro, Camila Bassetto, Alexandre de Barros Barreto, Vinicius RP Borges e Li Weigang: *Development of swim registry for air traffic management with the blockchain support*. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 3544–3549. IEEE, 2018. 25