Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Uma Proposta de Barramento de Dados para Integração de Serviços Públicos Digitais

Kelson Victor Praxedes de Almeida

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Computação Distribuída | Sinais, Sistemas Digitais e Gráficos

Prof. Dr. Rostand Edson Oliveira Costa
(Orientador)

João Pessoa, Paraíba, Brasil
Kelson Victor Praxedes de Almeida, 23 de Julho de 2021

Catalogação na publicação
Divisão de Processos Técnicos

A447p Almeida, Kelson Victor Praxedes de.
Uma proposta de barramento de dados para integração de serviços públicos digitais / Kelson Victor Praxedes de Almeida. - João Pessoa, 2021.
92 f. : il.

Orientação: Rostand Edson Oliveira Costa.
Dissertação (Mestrado) – UFPB/CI.

1. Informática. 2. Interoperabilidade. 3. Governo eletrônico. 4. Tecnologias – Troca de dados. 5. Criptografia. I. Costa, Rostand Edson Oliveira. II. Título.

UFPB/BC CDU 004(043)

Elaborado por Larissa Silva Oliveira de Mesquita - CRB-15/746

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Kelson Victor Praxedes de Almeida, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 30 de julho de 2021.

Aos trinta dias do mês de julho do ano de dois mil e vinte e um, às dezesseis horas, por meio de videoconferência, reuniram-se os membros da Banca Examinadora constituída para julgar o trabalho do sr. Kelson Victor Praxedes de Almeida, vinculado a esta Universidade sob a matrícula nº 20191000586, candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores: Rostand Edson Oliveira Costa (PPGI), Orientador e Presidente da Banca; Tiago Maritan Ugulino de Araujo (PPGI), Examinador Interno; Daniel Faustino Lacerda de Souza (UFPB), Examinador Externo ao Programa. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou a finalidade da reunião e passou a palavra ao candidato para que ele fizesse a exposição oral do trabalho de dissertação intitulado: "Uma Proposta de Barramento de Dados Para Integração de Serviços Públicos Digitais". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "***aprovado***". Do ocorrido, eu, Ruy Alberto Pisani Altafim, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da banca examinadora. João Pessoa, 30 de julho de 2021

Prof. Dr. Ruy Alberto Pisani Altafim

Prof. Rostand Edson Oliveira Costa
Orientador (PPGI-UFPB)

Prof. Tiago Maritan Ugulino de Araujo
Examinador Interno (PPGI-UFPB)

Prof. Daniel Faustino Lacerda de Souza
Examinador Externo ao Programa (UFPB)

Resumo

O uso tecnologias que visam desburocratizar serviços públicos são de suma necessidade para as sociedades ao redor de todo o mundo. Com a utilização de artefatos e práticas tecnológicas, é possível economizar financeiramente e em tempo de trabalho. Em um país de dimensões continentais como o Brasil, esse desafio se torna ainda mais complexo, devido a grande quantidade de ministérios, órgãos, empresas públicas, secretarias, dentre outros presentes nos mais de 5570 municípios e Distrito Federal. Analisando estes fatos, a interoperabilidade de dados e informações entre os diferentes Sistemas de Informação dessas esferas é fundamental no processamento, validação e fornecimento de serviços públicos para a população. Este trabalho propõe a utilização de um barramento de troca de dados que vise auxiliar no impulso tecnológico do Governo Digital do Brasil. Essa camada interoperável tem o objetivo de fornecer um melhor atendimento tecnológico na interoperabilidade de Governo Eletrônico. Tal proposta de barramento, chamado *IO Data Bus*, mostrou-se eficiente através de simulações práticas do contexto do fornecimento do Auxílio Emergencial pelo governo federal. Os experimentos foram realizados utilizando como prova de conceito a camada de troca de dados *X-Road*, que proporcionou a base tecnológica da Estônia Digital.

Palavras-chave: Interoperabilidade, Governo Digital, Barramento, IO Data Bus, X-Road.

Abstract

The use of technologies that aim to reduce the bureaucracy of public services is extremely necessary for societies around the world. Using artifacts and technological practices, it is possible to save money and work time. In a country of continental dimensions like Brazil, this challenge becomes even more complex, due to the large number of ministries, agencies, public companies, secretariats and others spread over more than 5570 municipalities and the Federal District. Analyzing these facts, the interoperability of data and information between the different Information Systems of these spheres is fundamental in the processing, validation and provision of services. This work proposes the use of a data exchange bus that aims to assist in the technological impulse of the Digital Government of Brazil, this interoperable layer can provide security, standardization, availability and reliability in e-gov communications. The proposal, called IO Data Bus, shows to be efficient through practical simulations using the context of the provision of Emergency Aid by the federal government. The experiments were carried out using the X-Road data exchange layer as a proof of concept, which provided the technological base of Estonia Digital.

Keywords: Interoperability, Digital Government, Bus, IO Data Bus, X-Road.

Agradecimentos

Primeiramente à Deus, ao Divino Espírito Santo e a minha mãe, advogada e intercessora fiel, Maria Santíssima.

À minha esposa Erica Renally por todo amor, carinho, dedicação e companheirismo durante esta e em muitas outras caminhadas da vida.

Aos meus pais Kelson Virgílio e Maria de Fátima e à minha irmã Priscilla Praxedes por toda educação, amor e ensinamentos da jornada. Como também ao meu cunhado Pedro Jansen e ao meu sobrinho Levi.

Aos meus sogros Rita e Edmilson, cunhados Emilly, Alex e Francisco por todo apoio e motivação.

A todos tios e tias, todos os familiares e amigos que são verdadeiras dádivas de Deus.

Ao meu orientador Dr. Rostand Costa por todos os ensinamentos, paciência e solicitude durante todo o período acadêmico. Gratidão também por todas as valiosas dicas e avaliações da banca examinadora, composta pelo Prof. Dr. Tiago Maritan e pelo Prof. Dr. Daniel Faustino.

"Sê humilde para evitar o orgulho, mas voa alto para alcançar a sabedoria."

(Santo Agostinho)

Conteúdo

Lista de Símbolos

NIIS : *Nordic Institute for Interoperability Solutions*

e-PING : *o Padrões de Interoperabilidade de Governo Eletrônico*

TIC : *Tecnologia da Informação e Comunicação*

e-gov : *Governo Eletrônico*

DLT : *Distributed Ledger Technology*

REST : *Representational State Transfer*

DNI : *(Documento Nacional de Identificação*

JSON : *JavaScript Object Notation*

CNIS : *Cadastro Nacional de Informações Sociais*

Lista de Figuras

Lista de Tabelas

Lista de Códigos Fonte

Capítulo 1

Introdução

No Brasil, a prestação de serviços públicos considerada por alguns doutrinadores como a principal justificativa e finalidade do Estado é um dever da Administração Pública [Purificação 2019]. Esta é composta por um conjunto de órgãos instituídos pelo Governo para a gestão de bens e interesses qualificados da comunidade no âmbito de todas as esferas, tendo como principal objetivo, a prestação do serviço público em benefício da coletividade [MEIRELLES et al. 2010].

Sistemas de Computação e artefatos tecnológicos já se tornaram itens indispensáveis na busca pela diminuição de tarefas burocráticas e economia em esforço e tempo de trabalho entre pessoas, empresas e governos ao redor do mundo. Em grandes instituições ou principalmente em governos, esses diferentes Sistemas de Informação e base de dados, em muitos dos casos, precisam trocar informações entre si para a realização das tarefas necessárias.

O estudo de técnicas que visem a troca de dados entre sistemas de maneira segura, rápida e íntegra pode ser avaliado como um fator que alavanca o processo de digitalização de serviços públicos, reduzindo assim tempo de trabalho, tempo de resposta em atendimentos e solicitações da sociedade e público em geral e além do mais, pode fornecer mais transparência e padronização.

A conexão entre diferentes sistemas neste universo é determinante na busca pela digitalização de processos e serviços, principalmente entre grandes entidades governamentais, nas quais geralmente há uma enorme quantidade de requisições e armazenamento de informações.

Práticas ultrapassadas ainda assombram a forma de como serviços, geralmente essen-

ciais, são repassados para a população brasileira. Só para se ter uma dimensão, empresas brasileiras gastam quase duas mil horas e R$60 bilhões apenas em burocracia tributária anualmente [ALVARENGA 2017].

Em contrapartida, pequenos países europeus como a Estônia e a Finlândia são exemplos em digitalização e desburocratização dos seus serviços públicos, onde estes são prestados quase em 100% de maneira digital e interoperável, utilizando-se como suporte uma identificação única para cada cidadão ou empresa e uma camada de troca de dados chamada *X-Road* [e-Estonia 2020].

Os benefícios da interoperabilidade podem trazer a redução de custos de produção ou transação, normalmente se utilizando de padronização e processos automatizados [Choi e Whinston 2000]. Então, observando esse aspecto da importância da comunicação entre diferentes sistemas e entidades, podemos perceber que pesquisas e aprofundamentos técnicos-teóricos sobre interoperabilidade são de suma relevância para o desenvolvimento, planejamento e execução da digitalização e interligação entre os dados, processos e demandas de diferentes entidades em uma única ou várias cadeias de componentes de *software*.

Segundo [Wegner 1996], interoperabilidade é a habilidade de dois ou mais componentes de *softwares* cooperarem, apesar da possível diferença de linguagem, interface, e plataforma de execução. Atualmente, o uso de práticas de interoperabilidade entre diferentes Sistemas de Informação possibilita que a digitalização de cadeias de informações de diferentes propósitos possam se conectar em um só objetivo de tratamento, processamento e fornecimento de dados.

É possível verificar uma forte demanda por interoperabilidade entre os serviços públicos oferecidos por meio de *e-gov*. O termo Governo Eletrônico ou *e-gov* trata da utilização da Tecnologia da Informação e da Comunicação (TIC) por órgãos públicos, no intuito de auxiliar e facilitar a vida dos prestadores de serviço e da população. Em outras palavras, o *e-gov* pode ser visto como a utilização de tecnologias na entrega, manutenção e fornecimento de serviços públicos (*e-serviços*) mantidos pelo governo.

1.1 Justificativa

Há anos o governo brasileiro vem utilizando de maneira intensiva recursos de TIC para o fornecimento e entrega de serviços públicos com o intuito de propiciar melhorias na gestão e operação de processos da administração pública, visando o aumento da oferta de serviços e melhorias na sua qualidade e efetividade [Araujo, Reinhard e Cunha 2018].

Em meados 1996, quando surgiu o conceito de Governo Eletrônico no Brasil acompanhado da crescente ascensão da internet naquela época, as iniciativas de *e-gov* eram em grande parte passivas e não integradas [Mesquita 2020]. No entanto, com as mudanças socioeconômicas que geram contínuas transformações a nível mundial, não é mais aceitável que as informações fiquem isoladas [Cruz et al. 2020]. Em razão do avanço tecnológico ao decorrer dos anos, a maneira em que diferentes sistemas trocam informações (interoperabilidade) foram se aprimorando, não só no contexto de governos, como também em vias gerais.

O governo brasileiro possui, inclusive, um conjunto Padrões de Interoperabilidade de Governo Eletrônico (e-gov), mais conhecido pela sigla *e-PING*. Esse padrão, [Brasileiro 2018] define um conjunto mínimo de premissas, políticas e especificações técnicas que regulamentam a utilização da TIC na interoperabilidade de serviços de Governo Eletrônico, estabelecendo as condições de interação com os demais Poderes e esferas de governo e com a sociedade em geral.

Porém, como se tratam apenas de padrões e regras, o Governo Eletrônico brasileiro não apresenta uma solução técnica, como uma camada ou barramento que forneça a interoperabilidade entre todos ou a maioria dos serviços eletrônicos.

Em [Barros, Cepik e Canabarro 2010] foi feito um levantamento sobre três possíveis alternativas para a plataforma de interoperabilidade no governo brasileiro. A primeira alternativa é dita como o modelo ponto-a-ponto, que seria a adoção de um *framework* de interoperabilidade, como o *e-PING*, modelo este que já está em vigor e visa a interoperabilidade sem implantação de uma infraestrutura tecnológica ou desenvolvimento de interfaces que produzem intercâmbio entre dados. Já o segundo e o terceiro modelo discorrem sobre uma solução tecnológica que forneça suporte às práticas interoperáveis presentes no guia teórico.

Como citado anteriormente, tal utilização de solução tecnológica foi uma das principais

razões do *case* de sucesso estoniano [e-Estonia 2020]. Neste exemplo, com a adoção de um barramento para viabilizar a interoperabilidade foi possível ampliar o universo de digitalização de serviços do governo com o objetivo de formar uma base tecnológica que diminua esforços, tempo e dinheiro e ajude a melhorar as diferentes comunicações e serviços que atendem a população.

Entretanto, em um país de realidades culturais, econômicas, populacional e territoriais bem distintas das nações atualmente utilizadoras de camadas de troca de dados *X-Road*, a exemplo da Estônia e Finlândia, será necessário analisar e comparar como se dão os fluxos atuais de troca de dados e como esse barramento pode melhorar essa interoperabilidade. Há um contexto inicial do governo brasileiro [Governo Digital 2020], que traz um planejamento acerca de melhorias e evoluções no Governo Digital e até cita como estratégia a integração de serviços através de uma camada tecnológica, mencionando o *X-Road* como principal referência.

Neste sentido, através do estudo do atual cenário de Governo Eletrônico brasileiro e de seus conceitos relacionados, este trabalho apresenta os resultados de uma pesquisa voltada para a verificação da utilização de artefatos e tecnologias consolidadas, como o *X-Road*. Visando assim, o apoio tecnológico que pode propiciar maior integração entre serviços públicos digitais e contribuindo com o aspecto de digitalização de meios governamentais.

1.2 Objetivos

1.2.1 Objetivo Geral

Levando em conta o cenário exposto anteriormente, o objetivo geral deste trabalho é propor um barramento de troca de dados que vise uma melhor interoperabilidade entre serviços públicos eletrônicos. Em nossa prova de conceito utilizaremos uma abstração do Auxílio Emergencial, atualmente em vigor no governo.

Tal barramento, chamado de *IO Data Bus*, tem como meta diminuir as questões de ineficiência interoperável entre sistemas de diferentes entidades. Para isso, o barramento visa realizar a integração tecnológica entre sistemas de informação.

1.2.2 Objetivos Específicos

Para atingir tal objetivo geral, este trabalho possui os seguintes objetivos específicos:

- Aprofundar os estudos sobre interoperabilidade, *X-Road* e serviços públicos digitais;
- Analisar como ocorre, atualmente, o fluxo de troca de dados no serviço digital do Auxílio Emergencial e propor sua sua troca de dados através de uma simulação de prova de conceito utilizando nosso barramento;
- Implementar novos modelos de semânticas de dados e políticas de acesso ao barramento;
- Utilizar algumas premissas de nível tecnológico, presentes no *e-PING*, para dar apoio teórico na nossa prova prática e ter como base regras atualmente utilizadas na interoperabilidade do nosso *e-gov*;
- Buscar os benefícios que um barramento de troca de dados e um *e-serviço* de identidade digital única pode trazer no processo de digitalização e desburocratização.

1.3 Estrutura da Dissertação

Os capítulos subsequentes estão organizados da seguinte maneira:

- **Capítulo 1:** Traz os aspectos introdutórios da dissertação e o estudo do estado da arte;
- **Capítulo 2:** Os conceitos de Governo Eletrônico, Interoperabilidade, Tecnologias de Troca de Dados, Criptografia e DLT são apresentados em detalhes no capítulo de Fundamentação Teórica;
- **Capítulo 3:** Discorre sobre a proposta do *IO Data Bus* e suas principais características;
- **Capítulo 4:** Contempla o planejamento experimental e resultados obtidos;
- **Capítulo 5:** Traz um recorte de fontes correlatas, presentes na literatura, que se assemelham com a nossa proposta de trabalho;
- **Capítulo 6:** Temos a conclusão e trabalhos futuros;

- Por fim, são apresentadas as referências bibliográficas utilizadas no trabalho.

Capítulo 2

Fundamentação Teórica

2.1 Governo Eletrônico (*e-gov*)

O termo Governo Eletrônico ou *e-gov* trata da utilização da Tecnologia da Informação e da Comunicação (TIC) por órgãos públicos, no intuito de auxiliar e facilitar a vida dos prestadores de serviço e da população. Em outras palavras, o *e-gov* pode ser visto como a utilização de tecnologias na entrega, manutenção e fornecimento de serviços públicos mantidos por governos.

Em relatório das Nações Unidades (2002), chamado de "*Benchmarking e-government: A Global Perspective Assessing*", o governo eletrônico pode oferecer numerosas possibilidades de melhorar a forma como uma nação ou setor público responde às necessidades básicas de seus cidadãos [Souza, Curi e Nuintin 2019].

Na literatura, o termo governo eletrônico vem explanado de várias maneiras. Por exemplo, a Organização para Cooperação e Desenvolvimento Econômico (OCDE) [Field et al. 2003], define governo eletrônico como a utilização de tecnologias de informação e comunicação, em particular a Internet, como ferramenta para se obter um melhor governo. Para [Santos 2010], o ambiente ideal para transações de *e-gov* é aquele que os usuários têm acesso às informações e serviços a partir de um único ponto de acesso. O autor ainda afirma que, de forma simplificada, a implementação de *e-gov* geralmente envolve uma evolução em 3 pontos: presença na internet através de informações básicas; capacidade de transação para indivíduos e empresas; e informações e transações integradas, com a colaboração de agências.

Para [BARBOSA, FARIA e PINTO 2004], a implantação de um governo eletrônico é sustentada por quatro pilares:

- Governança de TIC: definição de conjunto de modelos, padrões, regras e instrumentos de relações entre a administração pública e partes interessadas através de instrumentos de TI;
- Sistemas de Informação, Arquiteturas de TI e Provedores de Infraestrutura Tecnológica: adoção de práticas, técnicas e regras para a concepção de soluções governamentais de TI;
- Segurança da Informação: planos consistentes que resguardem a segurança das informações sob a guarda do Estado;
- Provedores de Infraestrutura Tecnológica: definição de níveis de serviços para aquisição de bens e serviços de TIC que visem a operação de serviços públicos digitais.

Em países em desenvolvimento, a adoção de práticas de *e-gov* ainda é um grande desafio. Divisão digital entre a população menos e mais favorecida, precários serviços de governo eletrônico e disponibilidade e acesso a tecnologia pelas pessoas podem ser ilustrados como fatores determinantes da lenta digitalização de serviços públicos nesses países [Joshi e Islam 2018].

No Brasil, de acordo com [Aguair et al. 2010], a gestão dos recursos de TIC no Governo Federal se iniciou de forma sistemática a partir de 1994, com a publicação do Decreto nº 1.048/1994, que criou o Sistema de Recursos de Informação e Informática (SISP), dando início à chamada administração eletrônica da gestão interna do Governo. Desde então, uma série de projetos caminharam em direção da criação de uma política de Governo Eletrônico (*e-gov*) brasileira.

Porém, foi apenas na criação do Grupo de Trabalho Interministerial de TIC, em 2000, que foram executadas ações no estabelecimento das diretrizes e metas que permitiram a definição de um modelo conceitual de *e-gov* para promover formas eletrônicas de interação entre Governo e cidadãos [Aguair et al. 2010].

Ressalta-se que práticas de Governo Eletrônico, para que possam ser efetivas, devem facilitar o acesso à informação, ter transparência e realizar a prestação de contas pelos gestores

para a população, que deve conhecer e avaliar a gestão dos recursos públicos [Souza, Curi e Nuintin 2019].

2.1.1 Padrões de Interoperabilidade do Governo Eletrônico - e-PING

Os Padrões de Interoperabilidade do Governo Eletrônico do Brasil, o *e-PING*, foram inspirados por outros marcos nacionais de interoperabilidade, como os exemplos da Austrália, Estados Unidos, Nova Zelândia e Reino Unido. A primeira versão do *e-PING* foi publicada em 31 de maio de 2004 [Santos 2011].

De acordo com [Brasileiro e Eletrônico 2018], esses Padrões de Interoperabilidade de Governo Eletrônico têm como propósito ser o paradigma para o estabelecimento de políticas e especificações técnicas que permitam a prestação de serviços eletrônicos de qualidade à sociedade.

O documento aborda cinco itens de políticas gerais que visam dar fundamento as especificações técnicas e segmentos que caracterizam o *e-PING*, esses itens são:

- Adoção Preferencial de Padrões Abertos: Há preferência na utilização de padrões abertos nas especificações técnicas;
- Uso de *Software* Público e/ou *Software* Livre: Priorizar a utilização de *software* público e/ou livre, que atendam as normas definidas no âmbito do SISP.
- Transparência: De acordo com LAI (Lei de Acesso à Informação) de 2011, o acesso é regra e o sigilo constitui uma exceção, busca da interoperabilidade para a publicidade dos dados que tem como objetivo a diminuição de interações do cidadão com o governo;
- Segurança: Considerar o nível de segurança requerido pelo serviço;
- Existência de Suporte de mercado: o *ePING* deve considerar a utilização de soluções amplamente utilizadas no mercado com o objetivo de diminuição de custos e dos riscos na utilização, desenvolvimento e implantação de serviços nos sistemas de informações governamentais.

O e-PING é concebido como uma estrutura básica na estratégia do governo eletrônico [Brasileiro e Eletrônico 2018]. Inicialmente esses padrões foram desenvolvidos apenas para o âmbito do governo federal (Poder Executivo), mas, posteriormente, não restringiu a participação, por adesão voluntária, de outros poderes e esferas do governo.

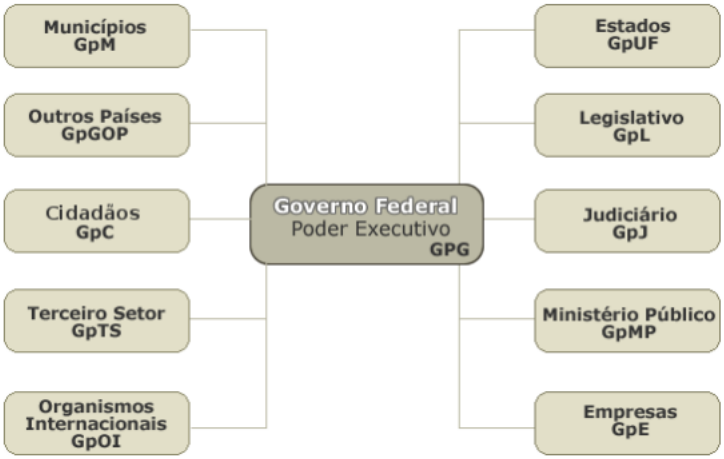

Figura 2.1: Relacionamentos na Interoperabilidade do Governo Federal

A Figura 2.1 ilustra o relacionamento entre os sistemas do governo federal e as interações com os cidadãos, governos estaduais e municipais, Poderes Legislativo e Judiciário, além do Ministério Público Federal, Organismos Internacionais, Governos de outros países, empresas do Brasil e no mundo e o Terceiro Setor.

O guia completo do *e-PING* está presente em um extenso documento [Brasileiro e Eletrônico 2018]. Em sua versão mais atualizada, a de 2018, os padrões estão organizados através de um detalhamento do escopo do documento, das políticas gerais dos padrões, segmentação, classificação das especificações técnicas, Governança e Gestão do *e-PING*, entre outros.

Sobre a segmentação, a arquitetura do *e-PING* foi segmentada em cinco partes, isso desde a sua versão inicial (chamada de versão zero) [Santos 2011] [Brasileiro e Eletrônico 2018]. Para cada segmento há um grupo de trabalho que é formado por profissionais atuantes em órgãos dos governos federal, estadual e municipal, especialistas em cada assunto [Brasileiro e Eletrônico 2018].

Tabela 2.1: Segmentação do e-PING

Segmento	**Aspectos**
Interconexão	Estabelece as condições para que as redes dos órgãos possam se interconectar, para promover a interoperabilidade.
Segurança	Aspectos de Segurança de TIC.
Meios de Acesso	Padrões dos dispositivos de acesso aos serviços de e-gov.
Organização e Intercâmbio de Informações	Tratamento e transferência de informações nos serviços do e-gov.
Áreas de Integração para Governo Eletrônico	Utilização/Construção de especificações técnicas para sustentar o intercâmbio de informações no e-gov.

O intuito dessa segmentação em 5 partes é a organização das definições do padrão. Esses segmentos ainda possuem subdivisões em componentes que visam demonstrar as especificações técnicas para serem adotadas pelo governo federal no âmbito das práticas. Tais especificações técnicas estão classificadas em 4 níveis ao decorrer do documento. [Brasileiro e Eletrônico 2018] Esses níveis são descritos a seguir:

- **Adotado (A):** Item foi adotado pelo governo como padrão na arquitetura *e-PING*;
- **Recomendado (R):** Atende as políticas do *e-PING*, é reconhecido como um item que deve ser utilizado, mas ainda não foi submetido a um processo formal de homologação;
- **Em Transição (T):** O governo não recomenda, porém está no *e-PING* devido ao uso ainda significativo em alguns órgãos, mas tende a ser desativado assim que houver um componente capaz de substituir;
- **Em Estudo (E):** Está em avaliação e poderá ser adotado quando a avaliação estiver concluída.

O documento é finalizado com uma extensa lista das especificações técnicas, dividas pelos segmentos onde são definidos os níveis das situações de cada especificação. Por exemplo, na segmentação de interconexão o componente de sincronismo de tempo tem a especificação da utilização de *Network Time Protocol* (NTP) na versão 4 e essa especificação está com a situação (A) ou adotada. Outro exemplo dessa lista é que no segmento de segurança, o componente de transferência de dados em redes inseguras possui a especificação técnica para a utilização de *Transport Layer Security* (TLS) na versão 1 podendo anular o SSL v3 nos

sistemas governamentais, como nível de situação essa especificação está como (R) ou recomendada até o momento. Várias outras especificações técnicas e situações de implantação estão contidas no decorrer da arquitetura *ePING*.

2.1.2 Caso de Sucesso: Estônia Digital

De acordo com [e-Estonia 2020], 67% da população da Estônia utiliza a carteira digital de identidade regularmente, 99% dos serviços públicos estão online, 2773 diferentes serviços podem se comunicar por uma camada de troca de dados chamada *X-Road*. A incrível digitalização dos serviços faz da Estônia um exemplo tecnológico de qualidade, rapidez e não burocrático a ser seguido.

Em termos proporcionais, a Estônia pode ser chamada como a nação mais digital do planeta. Logo após se tornar independente da União Soviética, em 1991, o país percebeu um novo modelo de digitalização de serviços públicos para simplificar a vida da sua sociedade.

Antes da independência, menos da metade da população estoniana possuía linha telefônica, o ministro da época, *Mart Laar*, ajudou no impulso para um período de modernização, estabelecendo bases necessárias para levar o país à era digital [e-Estonia 2020].

Quando houve a separação da *URSS*, a Estônia contou apenas com US$100 milhões para construir toda sua estrutura de nação independente. Mesmo com o pouco investimento conseguiu superar as adversidades e gerou um forte crescimento tecnológico nos serviços públicos.

Como um primeiro passo na corrida da digitalização, logo após o surgimento do *browser* Mosaic, o primeiro navegador web do mundo, o governo estoniano viu a oportunidade de conectar todas as escolas públicas do país e melhorar o serviço de educação [COSTA 2019].

O *e-Estônia* possui uma vasta gama de serviços online, em diferentes áreas como saúde, educação, segurança, transportes, residências e embaixadas [FERRARA 2018]. O site oficial do projeto [e-Estonia 2020], traz uma cronologia do processo de amadurecimento da transformação digital.

Em [e-Estonia 2020], essa cronologia se inicia partir de 1994, quando o parlamento lança o primeiro rascunho dos princípios da política de informação da Estônia para o desenvolvimento de TI no país e 2 anos depois, em 1996, foi iniciado o desenvolvimento de Infraestrutura de Tecnologia da Informação do país.

Já nos anos 2000, vários outros marcos foram amadurecendo o projeto *e-estonia*, entre eles os principais destaques ficam por conta do projeto *X-Road* como a camada de troca de dados entre os serviços, desenvolvido em 2001, além do sistema *e-ID* com uma identificação eletrônica e assinatura digital única disponível para todos os cidadãos, lançado em 2002 e outros como o *e-Voting* (para votações online).

Com o pioneiro *KSI Blockchain*, pouco tempo depois do surgimento do *Bitcoin*, a Estônia já utilizava a tecnologia *blockchain* em 2008 para armazenar e gerenciar vários registros governamentais.

O *e-Health* e *e-Prescription* revolucionaram o serviço de saúde pública do país. Para se ter uma ideia, todos os dados de saúde, histórico médico e as prescrições médicas ficam associadas ao *e-ID* nacional da pessoa através desses *e-serviços*.

De acordo com [e-Estonia 2020], o projeto *e-Residency* traz uma sociedade digital sem fronteiras para qualquer cidadão global. Esse é um dos projetos mais surpreendentes do *e-estonia*. Através dele é possível que qualquer pessoa, em qualquer lugar do mundo, possa ser residente digital da Estônia e adquirir alguns direitos como aberturas de empresas, acessar serviços bancários, declarar impostos, assinar documentos, etc. Isso levou muitos negócios e *startups* do mundo inteiro para a Estônia, um outro ótimo fator para a sua economia.

2.1.3 Camada de Interoperação: X-Road

O *X-Road* é um sistema de camada de troca de dados desenvolvido pela Estônia para dar suporte à digitalização dos serviços públicos estonianos.

Em um certo momento, a Estônia não estava conseguindo arcar com os altos custos de um servidor central único para suprir toda a demanda dos projetos vigentes, visto que a digitalização de praticamente todos os serviços públicos do país crescia de forma exponencial.

Com isso, em 2001, o projeto e-estonia apresentou a sua camada de troca de dados, o sistema *x-Road*. Por meio dele o projeto passou a ser mais descentralizado, não possuía apenas um banco de dados central, mas, agora, cada projeto deveria ter o seu próprio servidor provedor e banco de dados [e-Estonia 2020].

Trabalhando como uma espécie de barramento para troca de dados e informações, o *X-Road* também pode ser visto como um "adaptador" que conecta os diferentes serviços digitais do país.

Segundo [e-Estonia 2020], o *X-Road* permite que os vários Sistemas de Informação nacionais do setor público e até privado funcionem em harmonia. Por fornecer uma descentralização entre os diferentes sistemas, os softwares não necessitam ser desenvolvidos com a mesma tecnologia ou linguagem de programação, visto que eles podem se comunicar através de protocolos de mensagem *SOAP* e/ou *REST* [e-Estonia 2020][X-Road Architecture 2020].

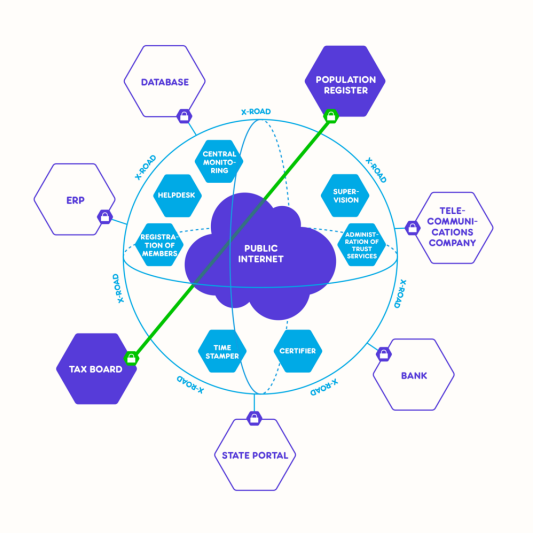

Figura 2.2: Ilustração do Funcionamento do X-Road [X-Road Architecture 2020]

No entanto, é preciso entender como funciona a arquitetura dessa camada de troca de dados. A Figura 2.2 traz uma ilustração do funcionamento básico do X-Road. No exemplo, os serviços são o de *Population Register* e o *Tax Board* e o *X-Road* faz a mediação da troca de dados e informações entre eles.

Essa camada não garante apenas a conexão entre os *e-serviços*, mas também fornece transparência, interoperabilidade e integração entre os Sistemas de Informação do governo. Assim, desde 2007 todo o projeto *e-estonia* funciona em cima dessa camada de troca de dados.

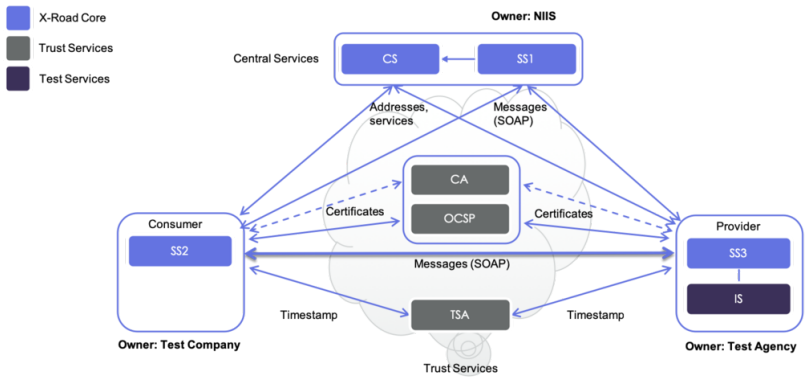

Figura 2.3: Arquitetura do X-Road Playground

O *X-Road Playground* é uma plataforma disponibilizada pelo projeto *e-estonia* para que o público em geral conheça e teste o funcionamento da camada por meio de simulações de troca de dados. A Figura 2.3, ilustra o fluxo de funcionamento da arquitetura do X-Road baseado no *Playground*.

Como é observado na figura anterior, estão presentes uma empresa de testes (*Test Company*) e uma agência de testes (*Test Agency*). Elas são responsáveis por requisitar e fornecer as informações de serviços respectivamente. A intenção é evidenciar de forma clara como o *X-Road* funciona na prática.

Em [X-Road Architecture 2020] são documentados e descritos os componentes presentes na arquitetura dessa camada de troca de dados. Os elementos principais se dão pelo servidor central do *X-Road* e por servidores de segurança de cada entidade.

O servidor central é responsável pelo gerenciamento dos bancos de dados dos membros (entidades) e seus respectivos servidores de segurança, além de possuir outras competências como listar as autoridades de certificação e *timestamps* confiáveis, como também de status de revogações de certificados digitais.

Cada entidade possui um Servidor de Segurança. Podemos citar em uma situação hipotética o Sistema de Informação de um hospital na Estônia tem o seu servidor de segurança (requisitante dos dados) e o serviço de *e-Health* possui outro servidor de segurança, e os mesmos se conectam através do servidor central do X-Road.

O *Security Server* é responsável pela mediação das chamadas e respostas das requisições entre os Sistemas de Informação. Ele encapsula diversos aspectos da segurança da infraestrutura do *X-Road*. Entre os principais estão o de gerenciamento de chaves de assinatura e

autenticação, registros de *timestamps* e oferecimento de protocolos de mensagens *SOAP* e *REST*.

Os Sistemas de Informação (*SI*) compreendem os *softwares* e tecnologias utilizadas pelos usuários finais via *X-Road*. Geralmente é no *SI* que são implementados os serviços de *Simple Object Access Protocol (SOAP)* ou de *Representational State Transfer (REST)* para a comunicação entre os *e-serviços*.

Outros componentes da arquitetura do *X-Road* são: *Proxy* de Configuração (disponibiliza toda configuração para os outros componentes); *Daemon* de Monitoramento Operacional (coletar e armazenar dados operacionais dos servidores de segurança);e, Monitoramento Ambiental (informações sobre o ambiente operacional dos servidores de segurança).

Tabela 2.2: Características Gerais do X-Road

Funcionamento	Principais Postulados	Troca de Dados
Independe de Plataforma	SI's separados	Via Internet
Multilateralismo	Falhas independentes	Canal Protegido (TLS)
Segurança	Somente autorizados	Dados Definidos (WSDL)
Abertura e Padronização	Autonomia	
	Valor evidencial (logs)	

A Tabela 2.2 aborda os principais pontos das características do funcionamento, postulados e de como ocorrem as trocas de dados via *X-Road*. No funcionamento é possível observar que para a comunicação entre partes não é necessário que as mesmas possuam idênticas plataformas ou tecnologias (Independente de Plataforma).

Atualmente a Estônia e a Finlândia fazem uso e coordenam o andamento da tecnologia (Multilateralismo), vários aspectos que reforçam a segurança entre as comunicações norteiam o projeto (Segurança). O sistema possui abertura para diferentes *e-serviços* públicos e privados, mas mantém uma padronização de como os dados são transmitidos entre as diferentes plataformas (Abertura e Padronização).

Como Principais Postulados, os *e-serviços* possuem diferentes Sistemas de Informação (SI's Separados). Dessa forma, não existe ponto central de falha e possíveis ataques à sistemas irão afetar apenas aquele sistema alvo, sem prejudicar o funcionamento de outros. Somente pessoas autorizadas têm acessos as regras e informações pré-definidas (Somente

Autorizados). Cada Sistema de Informação é autônomo para a concessão de acessos (Autonomia) e, tudo que ocorre nas comunicações gera valores evidenciais (Valor Evidêncial - *logs*).

Na troca de dados, toda comunicação entre as partes é feita pela Internet (Via Internet). Todo canal de comunicação é protegido criptograficamente por protocolo *TLS* (Canal Protegido - *TLS*) e a comunicação na troca de dados é feita por Serviços de Dados Definidos *WSDL* (Dados Definidos *WSDL*).

2.2 Interoperabilidade

Os Sistemas de Informação estão em constante modernização tecnológica que visam atender as demandas cada vez mais exigentes por parte de seus usuários. Quando se trata de uma cadeia de dados e informações, esses sistemas devem ter uma sintonia na comunicação entre eles. A troca de dados entre softwares é um item fundamental em que organizações e governos utilizam na validação, processamento e análise de dados por diferentes entidades.

Em [Bishr 1997], é dito que "a interoperabilidade é a capacidade que um sistema possui de compartilhar e trocar informações e aplicações." Em sistemas modernos esse processo de troca de informações se faz necessário, principalmente no fato de que as comunicações entre diferentes bases de dados poupam acúmulos de informações em pontos centrais, evitando assim problemas como ponto crítico de falha. A interoperabilidade permite racionalizar investimentos em TIC, através de compartilhamento, reúso e de intercâmbio de artefatos e dados tecnológicos [Brasileiro 2018].

De acordo com [Farinelli e Almeida 2014], a interoperabilidade pode ser dividida conforme os tipos a seguir:

- **Interoperabilidade técnica:** trata dos padrões de comunicação, de transporte, de armazenamento e de representação de informações;
- **Interoperabilidade semântica:** é dita como o significado da informação originada em diferentes sistemas;
- **Interoperabilidade organizacional:** diz respeito ao contexto organizacional, busca o alinhamento entre processos e informações presentes na arquitetura corporativa;

- **Interoperabilidade política e humana:** a forma de como a informação é divulgada;
- **Interoperabilidade intercomunitária:** trata do acesso das informações originadas em diferentes fontes;
- **Interoperabilidade legal:** exigências e implicações legais de tornar a informação livre e disponível;
- **Interoperabilidade internacional:** cooperação em escala internacional, havendo uma grande diversidade de padrões e normas, além de problemas por barreiras linguísticas.

Devido a essas características, os Governos Eletrônicos devem se empenhar para alcançar a interoperabilidade entre os seus serviços digitais. [Farinelli e Almeida 2014] ainda enumeram mais algumas razões: Primeira razão - ajuda a melhorar a tomada de decisão e a interoperabilidade vai permitir que dados de diferentes órgãos estejam disponíveis e acessíveis; Segunda razão - permite uma melhor coordenação das ações governamentais para a oferta se serviços à população; Terceira razão - pela interoperabilidade ser a base na oferta de serviços através de um ponto único de acesso, voltado para a oferta de serviços ao cidadão; e, Quarta razão - se trata da redução de custos, sistemas interoperáveis evitam a implantação de novos sistemas que anteriormente se fariam necessários e ainda aumenta o leque de alternativas de aquisições de soluções evitando a dependência de fornecedores únicos.

2.3 Tecnologias de Troca de Dados

2.3.1 API e REST/RESTful

A maior parte das aplicações web proveem acesso aos seus serviços graças a APIs REST [Barbaglia, Murzilli e Cudini 2017]. A sigla API vem de *Application Programming Interface* ou Interface de Programação de Aplicativos. As APIs são criadas com a intenção de que diferentes aplicações possam se comunicar e trocar informações, permitindo assim uma interoperabilidade entre serviços.

A sigla REST vem de *Representaional State Transfer* ou Transferência Representacional de Estado, se referindo a um estilo arquitetural [Surwase 2016]. O REST vem ganhando

muita popularidade na web ao decorrer dos anos. A propósito, essa representação foi apresentada por [Fielding e Taylor 2000] em sua tese de doutorado e desde então se disseminou na internet como auxílio importante nas práticas interoperáveis entre sistemas.

De acordo com [Fielding e Taylor 2000], o estilo arquitetural REST descreve seis restrições que definem a base do estilo RESTful. A nomenclatura de *RESTful* define a capacidade de determinados sistemas aplicar os princípios e características da arquitetura REST. Em [Surwase 2016] e [Plansky 2014] são listadas e caracterizadas as seguintes restrições:

1. *Client-Server*: Essa é uma restrição básica para aplicações REST. Separa as responsabilidades nos ambiente do cliente e no ambiente do servidor. O consumidor do serviço não realiza tarefas como comunicação com banco de dados, *cache*, *log*, etc. Enquanto o servidor (provedor de serviço), não executa tarefas como experiência com o usuário, interface, etc, gerando assim independência entre as funções.

2. *Stateless*: As requisições precisam ser independentes e tais requisições ainda precisam ter todas as informações necessárias para que o servidor possa processá-las corretamente.

3. *Cacheable*: Se trata da intenção de aumentar a performance para responder as requisições repetidas. Quando uma requisição é solicitada pela primeira vez, esta fica armazenada em uma memória cache temporária para que das próximas vezes em que ela for solicitada, o servidor já tenha a resposta pronta e evitando assim o reprocessamento.

4. *Uniform Interface*: Contrato da comunicação entre cliente-servidor. A intenção é que o componente fique mais simples de ser refatorado e melhorado. Há um guia para realizar essa comunicação uniforme:

    - Identificando o *resource*: Cada *resource* precisa de uma *URI* específica, como no exemplo abaixo.

    ```
    HTTP/1.1 POST http://iodatabus.gov.br/dni/32233234534
    ```

    - Representação do *resource*: Forma de como a resposta da requisição vai ser devolvida ao cliente. Alguns exemplos desses formatos são *HTML, XML, JSON, TXT,* etc. O exemplo abaixo ilustra o formato JSON.

```
{
    "nome_completo": "Maria Sampaio",
    "DNI": 23323243433,
    "data_nascimento": "18/01/1990"
}
```

- Resposta auto-explicativa: Metadados na requisição e resposta. Como ilustra o exemplo abaixo.

```
GET /#!/gov-br/tse/dni HTTP/1.1
User-Agent: Chrome/37.0.2062.94
Accept: application/json
Host: iodatabus.gov.br
```

- *Hypermedia*: Informações na resposta da requisição para que os navegadores possam interpretar e ter acesso a todos os recursos da aplicação, como se pode ver no exemplo abaixo.

  Requisição

```
HTTP/1.1 POST http://iodatabus.gov.br/mte/dni/2332324
   3433
```

  Resposta

```
{
    "post": {
        "dni": 23323243433,
        "nome_completo": "Maria Sampaio",
        "empregado": false,
        "_acoes": [
            {
                "href": "/mte/dni/23323243433",
                "method": "GET",
                "rel": "self"
```

```
            },
            {
                "href": "/mte/dni/23323243433",
                "method": "POST",
                "rel": "update"
            } ]
        }
```

5. *Layered System*: O cliente deve se comunicar com um intermediador antes de realizar a requisição para um servidor. Esse intermediador pode ser um *load balancer* ou alguma máquina que gerencie essas requisições. Com a utilização de um intermediador a estrutura fica mais flexível à mudanças.

6. *Code-on-Demand* (opcional): Ao cliente é permitido executar códigos sob demanda, através de applets ou scripts, por exemplo. Essa restrição pode ser utilizada quando é executada alguma parte do serviço do lado do cliente.

2.3.2 GraphQL

O *GraphQL* foi apresentado pelo *Facebook* como uma alternativa à arquitetura *REST*. Segundo [Brito, Mombach e Valente 2019], essa tecnologia se trata de uma nova linguagem para implementação de APIs baseadas em web. A supracitada API inclui um *endpoint* de busca que permite clientes recuperarem metadados. De acordo com [Hartig e Pérez 2018], essa linguagem está em ascensão e já é suportada por importantes *web services*, como os providos pelo *GitHub* e *Pinterest*, por exemplo.

A Figura 2.4 ilustra um exemplo de fluxo de dados utilizando a interface pública do *Github* em *GraphQL*. A parte (a) da figura mostra uma consulta através da interface, (b) ilustra a resposta dessa consulta e (c) traz um segundo nível da consulta requisitando os donos dos repositórios que são listados para cada dono de repositório no resultado da primeira consulta (a).

```
query {
  user(login: "danbri") {
    repositories(first: 2) {
      nodes {
        owner {
          login
} } } } }
```

(a) initial example query

```
data: {
  user: {
    repositories: {
      nodes: [
        { owner: { login: "danbri" } },
        { owner: { login: "danbri" } }
      ]
} } }
```

(b) result of initial example query

```
query {
  user(login: "danbri") {
    repositories(first: 2) {
      nodes { owner {
        repositories(first: 2) {
          nodes { owner {
            login
} } } } } } } }
```

(c) level-2 version of the example query

Figura 2.4: Consultas e respostas através da interface de GraphQL no Github [Hartig e Pérez 2018]

O tipo *Mutation* presente na linguagem serve para a definição do contrato que irá manipular os dados. A funcionalidade dele é comparável com os verbos *POST, PUT, PATCH* e *DELETE*, por exemplo. O Código Fonte 2.1 mostra um exemplo de *type Mutation* [Souza 2018].

Código Fonte 2.1: Mutation no GraphQL

```
1 type Mutation {
2     addStudy(study: StudyInput!): Study
3 }
```

Uma das grandes vantagens dessa linguagem é a possibilidade de construir aplicações que permitem extração e solicitações de dados de várias fontes em uma única chamada de API e mais flexibilidade para adicionar ou remover campos. O *GraphQL* é dividido itens principais, que são os *schemas*, *queries* e *resolver*. O *schema* é criado para descrever os dados disponíveis através do serviço. Esses dados são disponibilizados ou modificados através das *queries*. Por sua vez, o desenvolvedor da API anexa os campos de um determinado *schema* em funções denominadas *resolvers* [RedHat 2021]. A Figura 2.5 ilustra um *schema* da linguagem.

```
type Starship {
  id: ID
  name: String
  length: Float
}

interface Character {
  id: ID
  name: String
  friends: [Character]
}

type Droid implements Character {
  id: ID
  name: String
  friends: [Character]
  primaryFunction: String
}
```

```
type Human implements Character {
  id: ID
  name: String
  friends: [Character]
  starships: [Starship]
}

enum Episode { NEWHOPE EMPIRE JEDI }

union SearchResult = Human | Droid |
                     Starship
type Query {
  hero(episode: Episode): Character
  search(text: String): [SearchResult]
}

schema {
  query: Query
}
```

Figura 2.5: Exemplo de um schema do GraphQL [Hartig e Pérez 2018]

A [RedHat 2021] faz uma lista das principais vantagens e desvantagens da utilização do *GraphQL*. A Tabela 2.3 traz o detalhamento desse levantamento.

Tabela 2.3: Vantagens e Desvantagens da Utilização de GraphQL

Vantagens	**Desvantagens**
Schemas definem uma fonte de verdade única. Trata-se de uma maneira da organização federar a API inteira.	Questões culturais, como desenvolvedores acostumados com o *REST*.
Solicitações processadas em única transmissão, clientes recebem exatamente o que solicitaram, evitando *overfetching*.	Muito trabalho voltado para o servidor, aumentando a complexidade para desenvolvedores.
Tipos de dados bem definidos.	Exigência de novas estratégias para o gerenciamento da API.
Introspectivo, cliente pode solicitar lista de dados disponíveis.	Armazenamento em *cache* mais complexo do que na arquitetura REST.
Permite a evolução de uma API sem prejudicar as consultas já existentes.	Manutenção da API pode requerer tarefa extra para escrita de esquema *GraaphQL* que possa ser submetido à manutenção.
Muitas extensões *open-source* disponíveis.	
Não determina uma arquitetura de aplicação específica.	

O GraphQL vem ganhando cada vez mais notoriedade entre desenvolvedores. Isso devido à sua simplicidade na utilização e a vasta coleção de ferramentas de suporte [Taelman, Sande e Verborgh 2018]. As principais diferenças entre *REST* e *GraphQL* estão em como os dados são expostos. Na tecnologia proposta pelo *Facebook* é como um tipo de esquema, já em *REST* as aplicações de servidor implementam uma lista de *endpoints*. Diferentemente do *REST*, há especificação precisa de campos nas consultas do *GraphQL*. No *REST* as consultas são definidas por meio de *endpoints*, onde cada um desses retorna um conjunto predefinido de campos que representam dados sobre algum recurso. Entretanto, com o *GraphQL* a resposta é semelhante à estrutura da consulta [Brito, Mombach e Valente 2019].

2.4 Criptografia

O termo criptografia vem do grego *kriptos* ou oculto e *grapho* ou escrita. Em tempos de digitalização massiva e globalização de formas eletrônicas de comunicação, questões de privacidade e segurança dos dados são de suma importância. Desde meados do século V a. C., a ocultação de mensagens podiam ser feitas através de uma técnica secular chamada esteganografia, termo que vem do grego *steganos* ou coberto e *graphien* que significa escrever. Trata-se de uma técnica de ocultação de mensagens, no entanto essa técnica possui várias vulnerabilidades que podem ser facilmente exploradas [Fiarresga et al. 2010].

Nesse contexto, verificando a questão criptográfica, diferentemente da esteganografia, o seu objetivo já não parte de esconder a existência da mensagem, mas sim de ocultar o seu real significado. De acordo com [Fiarresga et al. 2010], a criptografia tem como conjunto de objetivos os seguintes itens:

- **Confidencialidade:** Deixa o real conteúdo da mensagem secreto, exceto para quem realmente tenha acesso à mensagem;
- **Integridade:** Assegura que não há alteração por pessoas não autorizadas;
- **Autenticação:** Identificar pessoas ou processos na comunicação estabelecida;
- **Não repudiação:** Evita que qualquer parte da comunicação negue o envio ou recebimento de uma informação.

Com isso, é fundamental que artefatos tecnológicos e computacionais forneçam suporte para que as características e objetivos de proteção da informação sejam efetuados. A criptografia apresenta-se em dois tipos: criptografia simétrica ou chave privada e criptografia assimétrica ou de chave pública [Oliveira 2012].

2.4.1 Criptografia Simétrica

Os algoritmos de criptografia utilizam-se de chaves para cifrar e decifrar as mensagens nas comunicações. De acordo com [Oliveira 2012], a criptografia simétrica é o modelo mais antigo dos algoritmos criptográficos. Nesse modelo a chave, ou seja, o elemento que dá acesso a mensagem oculta, trocada pelas partes é simétrica ou iguais. Essa chave, tipicamente, é representada por uma senha, usada tanto pelo remetente, para codificar a mensagem, quanto pelo destinatário, para realizar a descodificação. A Figura 2.7 ilustra o fluxo da troca de mensagens utilizando algoritmo de criptografia assimétrica.

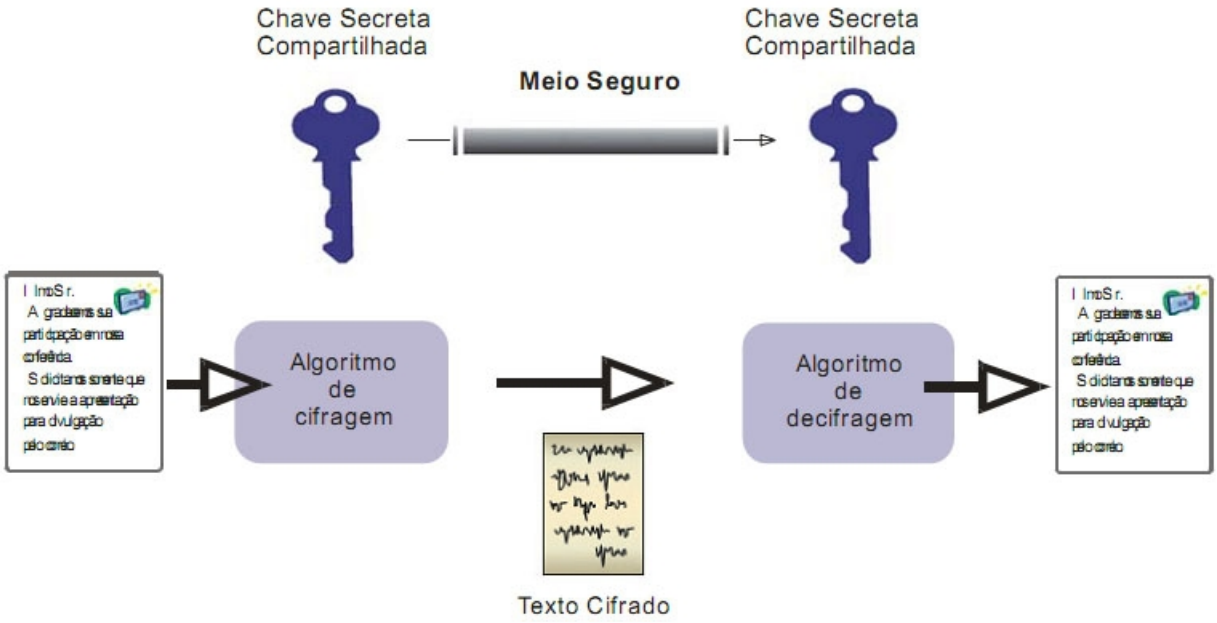

Figura 2.6: Fluxo da Criptografia Simétrica [Brocardo, Rolt e Fernandes 2006]

Segundo [Oliveira 2012], a principal vantagem é a simplicidade, pois essa técnica apresenta facilidade de uso e rapidez para executar os processos criptográficos. [Kouicem, Bouabdallah e Lakhlef 2018] também definem as chaves simétricas como um compartilhamento

das chaves criptográficas entre as entidades no sistema. Tal explicação pode ser melhor visualizada na Figura 2.7, onde a mesma chave está sendo utilizada para criptografar e descriptografar o conteúdo enviado. [Ludwig, Rebelatto e Silva 2020] elenca os principais tipos de algoritmos de chave simétrica:

- **AES:** *Advance Encryption Standard* (AES), desenvolvido por Joan Daemen e Vicent Rijmen em 1998, é uma cifra simétrica de bloco de chaves que suporta qualquer combinação de dados e comprimento de chave de 128, 192 e 256 bits;
- **AES S-box:** Matriz utilizada no algoritmo AES, funciona como uma caixa de substituição e atua como uma tabela de pesquisa;
- **DES:** Cifra de bloco de chave simétrica. O comprimento da chave é 56 bits e o tamanho do bloco é 64;
- **3DES:** Aplica 3 vezes a cifra DES para cada bloco de dados. O tamanho do bloco é de 64 bits e comprimento da chave 112 bits;
- ***Blowfish*:** Comprimento de chave que varia de 32 à 448 bits, tamanho de bloco de 64 bits;
- **RC4:** Cifra com tamanho da chave até 2048 bits. A cifra pode não ser segura devido ao comprimento da chave ser limitado.

2.4.2 Criptografia Assimétrica

É um modelo de criptografia criado na década de 1970, por Clifford Cocks, para o serviço secreto inglês (CGHQ). Neste tipo de algoritmo criptográfico, cada entidade da comunicação utiliza chaves distintas (assimétricas) e complementares, uma privada e outra pública [Oliveira 2012]. Dessa vez as chaves não são mais simples senhas, agora correspondem a arquivos digitais complexos. Nesse modelo a chave pública fica disponível para qualquer entidade que irá iniciar uma comunicação segura com outra, mas a respectiva chave privada deve ficar de propriedade única e exclusivamente de seu/sua proprietário(a). Segundo [PAVANATI et al. 2017], as abordagens assimétricas tradicionais agrupam todos os métodos com base em chaves públicas e requer autoridade para emitir certificados para diferentes usuários do sistema. A Figura 2.7 detalha o fluxo da mensagem através desse modelo.

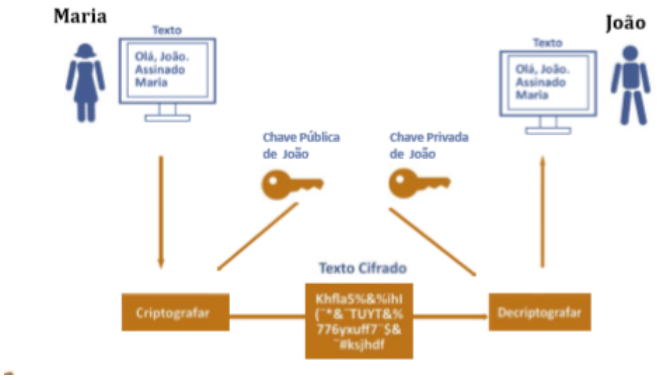

Figura 2.7: Fluxo da Criptografia Assimétrica [Brocardo, Rolt e Fernandes 2006]

De acordo com [Oliveira 2012], a grande vantagem desse sistema é permitir a qualquer um enviar uma mensagem secreta, apenas utilizando a chave pública de quem irá recebê-la. Com a chave pública disponível, não existe a necessidade do envio de chaves como é feito no modelo simétrico, enquanto a chave privada estiver segura, a confidencialidade da mensagem é garantida. Caso contrário, se terceiros possuírem a chave privada, a confidencialidade estará comprometida. O autor ainda cita alguns exemplos de algoritmos de chave assimétrica:

- **RSA:** Algoritmo assimétrico mais utilizado. Esse modelo utiliza números primos e sua premissa consiste na facilidade de multiplicar dois números primos para obter um terceiro número, no entanto, é muito difícil recuperar os dois primos a partir daquele terceiro. No Brasil, esse modelo é utilizado pela ICP-Brasil na emissão de certificados digitais.

- **El Gamal:** Utiliza manipulação matemática de grandes quantidades numéricas através de cálculo de logaritmo discreto em um corpo finito, assim obtendo a sua dificuldade.

- **Diffie-Hellman:** Também é baseado no problema do logaritmo discreto. Não permite ciframento ou assinatura digital e é um dos modelos mais antigos de criptografia assimétrica;

- **Curvas Elípticas:** Esse modelo consiste em modificações do modelo El Gamal, por exemplo, só que agora trabalhando com curvas elípticas, em vez de domínio dos corpos finitos.

2.4.3 Certificados Digitais

Por questões de segurança, com o uso do modelo de criptografia deve-se ter uma garantia de que a chave pública de uma entidade que se deseja comunicar seja realmente de posse da mesma. Sem essa garantia, entidades não autorizadas podem se passar por entes confiáveis, causar uma interceptação e fornecer chaves públicas forjadas. De acordo com [Oliveira 2012], "a garantia para evitar este tipo de ataque é representada pelos certificados de chave pública, comumente chamados de certificado digital. Tais certificados consistem em chaves públicas assinadas por uma pessoa de confiança".

Os certificados digitais podem ser utilizados como parte de processos de autenticação em sistemas e sites na internet, até mesmo para acompanhar ou retificar o imposto de renda no Brasil, assinar documentos e inúmeras outras possibilidades [CertiSign 2019]. O autor ainda cita alguns dos benefícios na sua utilização:

- **RG do mundo digital:** Por meio da criptografia pode garantir a autenticidade e integridade das transações realizadas;
- **Para assinar documentos:** Por conta do rigoroso processo de emissão e do fator criptográfico possibilita assinar transações e documentos no meio eletrônio;
- **Garante mobilidade:** Devido as características descritas anteriormente, é possível garantir mobilidade de vários tipos de transações distância;
- **Reduz custos:** Diminui despesas de itens físicos como impressão e armazenamento de papel, além de transporte e mão de obra.

O proprietário de um certificado digital, geralmente possui alguma autoridade, de sua confiança, que assine o mesmo. Esse tipo de autoridade tem o nome de *Certification Authority - CA* do português Autoridade de Certificação. Sendo assim, um certificado digital pode ser definido como um documento eletrônico que possui uma assinatura digital de uma terceira parte confiável. No Brasil, a autoridade certificadora raiz é a ICP-Brasil. No país ainda existem algumas outras autoridades como Caixa Econômica Federal (AC-Caixa), Serasa Experian (AC-Serasa), Receita Federal do Brasil (AC-RFB), Certisign (AC-Certisign), entre outras [Oliveira 2012].

2.4.4 Função Hash

Um *hash* criptográfico, ou simplesmente *hash* (função *hash*), se trata de um algoritmo matemático onde qualquer bloco de dado é transformado em uma série de caracteres de comprimento fixo. Independentemente do tamanho do bloco de entrada, a saída em *hash* sempre terá o mesmo comprimento [Karpersky 2014].

A Figura 2.8 ilustra exemplos de entrada de blocos de mensagens onde são executadas funções *hashes* que alteram o sentido da mensagem para uma saída aleatória de tamanho fixo.

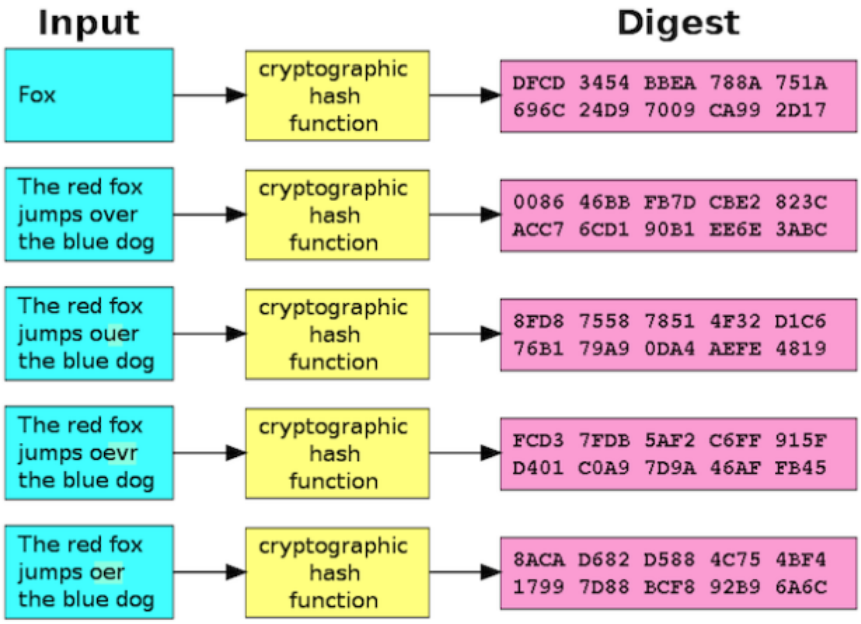

Figura 2.8: Transformando dados em hash [Karpersky 2014]

De acordo com [Oliveira 2012], "a assinatura digital obtida através do uso da criptografia assimétrica ou de chave pública infelizmente não pode ser empregada, na prática, de forma isolada, é necessário o emprego de um mecanismo fundamental para o adequado emprego da assinatura digital. Este mecanismo é a função *hashing*". O autor ainda destaca os principais tipos de funções:

- **SHA-1:** Secure Hash Algoritm 1, uma função unidirecional, criada pela NSA que gera um valor de 160 bits a partir de um tamanho arbitrário de mensagem. No ano de 2005 foram descobertas fortes vulnerabilidades nesse modelo, que logo obteve sua atualização;

- **SHA-2:** Evolução do SHA-1, também desenhado pela NSA, se trata de uma família de duas funções hash similares, mas com tamanhos diferentes (SHA-256 com 256 bits e SHA-512 com 512 bits). O ICP-Brasil já adota o padrão SHA-512 em substituição ao seu anterior (SHA-1);

- **MD5:** Função unidirecional, criada no MIT, produz um valor de hash de 128 bits. Por produzir apenas essa quantidade de bits é preterida devido a possíveis vulnerabilidades;

- **MD2 e MD4:** Antecessor do MD5, não é mais utilizado devido a fraqueza de sua segurança, também produz 128 bits de hash.

2.5 DLT: Distributed Ledger Technology

A tecnologia de Livro-Razão Distribuído, do inglês *Distributed Ledger Technology*, vem se firmando cada vez mais ao longo dos anos. De acordo com [Tapscott e Tapscott 2016], "as DLTs, também conhecida como *blockchain*, tem o poder de transformação para mudar tudo desde o meio de operação do comércio até mesmo levar transformação à economia em escala global". O NIST, que é o Departamento de Comércio dos Estados Unidos, define *blockchains* como livros digitais à prova de violação, implementados de forma distribuída e geralmente sem uma autoridade central.

Blockchain por si só pode ser considerado um banco de dados de apenas inserção que mantêm uma lista ordenada de dados chamados de blocos, que contêm diferentes tipos de transações [Casino, Dasaklis e Patsakis 2019]. Segundo [Isaja e Soldatos 2018], "um livro-razão distribuído é um sistema de dados que são replicados e armazenados em sincronia entre múltiplos nós de uma rede, onde todos os nós são pares da rede e não há um nó *master* que possui a cópia do livro-razão". Originalmente foi utilizada como histórico de transações de cripto-moedas digitais como a *bitcoin* [Zhu e Zhou 2016].

Com isso, referenciando sua pretensão tecnológica inicial, em [Nakamoto 2008], essa tecnologia foi apresentada como uma arquitetura segura que desse suporte a proposta da cripto-moeda *Bitcoin*. O termo *Blockchain* sugere uma "cadeia de blocos", onde cada bloco pode armazenar um conjunto de dados e transações. De acordo com [Hou 2017], *blockchain*

"é definida como um livro-razão distribuído que mantêm continuamente crescendo uma lista de dados publicamente acessíveis criptograficamente seguros de adulterações e revisões". A Figura 2.9 ilustra a conexão entre os blocos, através dos *links* entre os blocos e seus antecessores. Os blocos são distribuídos por ordem cronológica de entrada na rede. O primeiro bloco da rede é chamado de bloco gênesis e não possui um bloco antecessor [Sharma e Jain 2019].

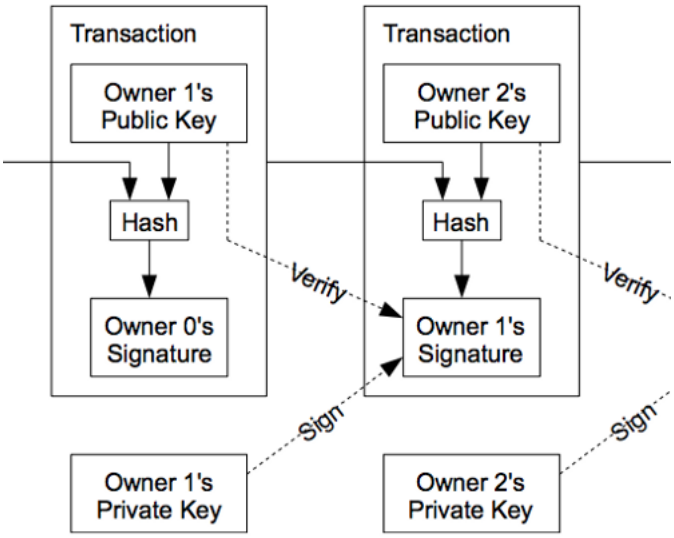

Figura 2.9: Fluxo da Cadeia de Blocos [Nakamoto 2008]

Como foi ilustrado na Figura 2.9 e mais detalhado na Figura 2.10, de acordo com [Sharma e Jain 2019], um bloco é constituído por um cabeçalho, o valor *hash* do seu antecessor e seu próprio, contador de transação, timestamp e uma árvore de merkle. O *timestamp* e a referência criptografada do bloco anterior são características que previnem que o conteúdo dos blocos não sejam modificados [Gao et al. 2021].

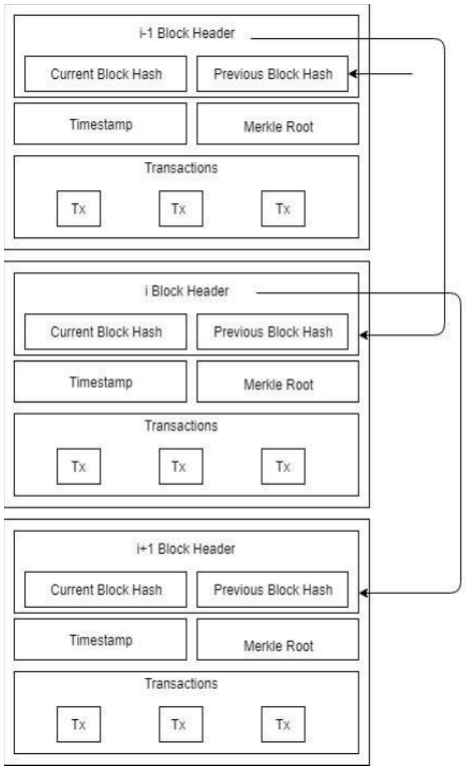

Figura 2.10: Conteúdo dos blocos e seus fluxos [Sharma e Jain 2019]

Para especificar os termos técnicos referentes à tecnologia, [Tasatanattakool e Techapanupreeda 2018] faz uma lista com o detalhamento de cada significado, a mesma é descrita abaixo.

- **Descentralizado:** O sistema que armazena informações através da rede;
- **Transparente:** Todos na rede conseguem visualizar o livro-razão, além de compartilhar informações;
- **Minerador:** Verificador de transações. (Exemplo: Em caso de *Proof-of-Work*);
- **Consenso:** Um método usado para verificar a transação;
- ***Forks*:** O problema que surge quando o nó é utilizado para uma diferente versão da *blockchain*;
- ***Hash*:** Uma função unilateral para checar a integridade de transações ou mensagens;
- ***Nó*:** O livro-razão no sistema *blockchain*;

- ***Timestamp*:** Data e hora no sistema computacional usado como carimbo de tempo eletrônico da transação.

Para garantir igualdade e justiça entre transações que podem rodar em dispositivos desconhecidos e descentralizados, é necessário que haja a concordância com algum protocolo. Esses protocolos são conhecidos como algoritmos de consenso e são o núcleo das DLTs/-blockchains decidindo como a rede irá funcionar [Sharma e Jain 2019].

Segundo [Datta 2019], a replicação das informações entre múltiplos nós da rede é necessário para reforçar a imutabilidade e integridade dos dados armazenados na DLT, onde, os nós replicados atuam como testemunhas. Desse modo, alterações em registros anteriores ou introdução de novos registros não são possíveis, em vez disso, as decisões são tomadas estabelecendo um consenso no estado compartilhado.

Originalmente, [Nakamoto 2008] propôs o algoritmo de consenso *Prof-of-Work* (PoW) ou algoritmo de Prova de Trabalho. A cripto-moeda *Bitcoin* utiliza um "quebra-cabeça" criptográfico para permitir adicionar novos blocos na *blockchain*, além de razões de autenticação e autorização [Datta 2019]. A Figura 2.11 ilustra o fluxo do *PoW*. Como pode ser visto, neste protocolo um grupo de participantes, denominados "mineradores", competem para a produção de novos blocos na cadeia. Esse processo pode levar à recompensas e as taxas de transação relativas ao novo bloco descoberto. Tal processo ocorre através da busca por uma "Nonce", através de um desafio de força-bruta computacional. O "Nonce" é um campo de 32 bits que pode assumir qualquer valor. A dificuldade para a busca desse valor aumenta a cada 10 minutos em média [Ribeiro e Mendizabal 2021].

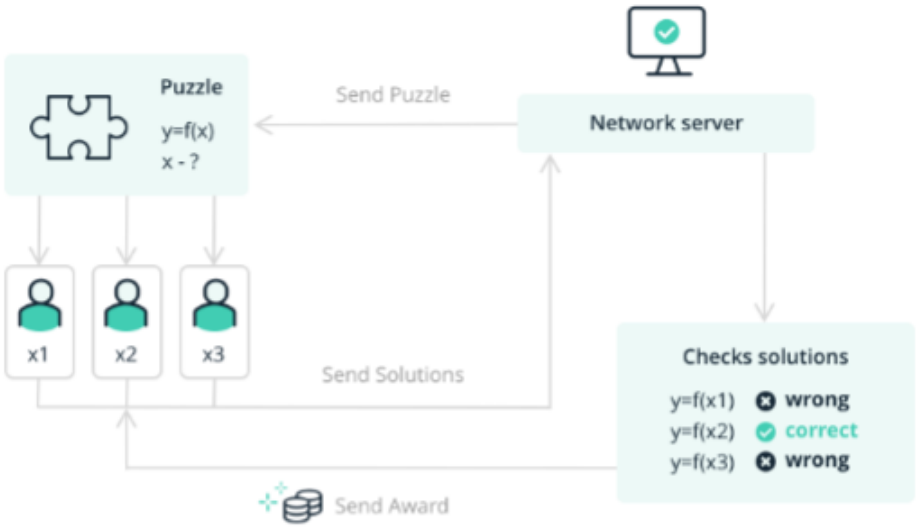

Figura 2.11: Algorítimo de Consenso de Prova de Trabalho

Analisando o fluxo de uma transação em uma rede DLT com PoW, pode-se observar que a transação é iniciada por um nó e é digitalmente assinada por sua chave privada. Um bloco é criado para representar essa transação que por sua vez esse bloco é compartilhado com todos os nós da rede. Um grupo de nós da rede é designado para validar o bloco e dar um retorno, eles recebem uma recompensa do *Proof-of-Work* e o bloco validado passa a integrar a rede [Sharma e Jain 2019].

A rede do *bitcoin* em que ocorrem esses processos é pública. As DLTs podem ser classificadas em públicas e permissionadas. As *blockchains* públicas são abertas para qualquer novo integrante da rede para replicar dados da blockchain e participar no processo de consenso [Datta 2019].

2.5.1 Utilização de DLTs Fora do Contexto Financeiro

Conforme explicitado anteriormente, a pretensão inicial da tecnologia se deu por meio do surgimento da cripto-moeda *bitcoin*, porém, [Tasatanattakool e Techapanupreeda 2018] discorre que a cripto-moeda criada por [Nakamoto 2008], apesar de ter sido a primeira, não é mais única aplicação utilizando a tecnologia de livro-razão distribuído. Isso porque já existem várias outras cripto-moedas presentes no mercado e além da tecnologia de *blockchain*

que são utilizadas para diversos outros tipos de aplicações fora do contexto financeiro.

Segundo [Bashir 2018], as *blockchains* podem ser classificadas em três fases: *blockchain* 1.0 que diz respeito a concepção inicial da tecnologia, dar suporte à cripto-moeda bitcoin; *blockchain* 2.0 com a proposta de contratos inteligentes; e, a *blockchain* 3.0 que leva as DLTs para além do contexto financeiro, empregando os seus benefícios em diversas áreas como cidades inteligentes, saúde, comércio e governos, por exemplo.

A utilização da tecnologia de DLTs em ecossistemas governamentais pode ser um facilitador devido aos benefícios que as *blockchains* podem proporcionar para diversos casos. Porém, a padronização e sistemas gerenciais, processos e responsabilidades das aplicações devem ser bem definidos para tal uso [Hou 2017].

Em [Datta 2019] são listados casos de usos de implantação da tecnologia de livros-razão distribuídos em Governos Eletrônicos. Entre esses itens estão os serviços de registro e notarias que é dito como o tipo de aplicação mais natural de serviços públicos digitais. Identidade Digital é outro serviço citado, inclusive o *The European Union Blockchain Observatory & Forum* enxerga a utilização de Identidade Digital como um bloco de construção fundamental, citado por [Lyons, Courcelas e Timsit 2018], que podem ser realizados usando *blockchain*, para atuar como identidade digital equivalente às identificações já emitidas pelos governos. [Datta 2019] ainda cita exemplos reais do uso de Identidades Digitais com DLTs em alguns países como o sistema de votação eletrônica, *e-voting* em Zug na Suíça, certificação de treinamento de trabalhadores e *check-in* e *check-out* no transporte ferroviário suíço.

2.6 Considerações do Capítulo

Este capítulo apresentou temas acerca da proposta desta dissertação. Na primeira seção do capítulo foram apresentados conceitos básicos que envolvem a temática de Governo Eletrônico. Na segunda seção foi possível apresentar um *overview* sobre Interoperabilidade. Na terceira seção foram apresentados os principais conceitos de tecnologias de troca de dados utilizadas no nosso trabalho, como o *REST* e *GraphQL*.

Os conceitos básicos acerca de Criptografia que utilizamos na nossa proposta também foram explorados na quarta seção do capítulo. E por fim, na quinta seção, houve uma síntese do estado da arte de conceitos envolvendo Livros-Razão Distribuídos.

Tais conceitos são de suma importância para o trabalho, pois será proposto no próximo capítulo um barramento de dados que visa uma melhor integração de serviços públicos digitais. Os conceitos fundamentados neste capítulo dão suporte para chegarmos a nossa proposta de defesa.

Capítulo 3

Trabalho Proposto

Após uma breve abordagem, no Capítulo 2, sobre Governo Eletrônico, Interoperabilidade, Criptografia, DLTs e outros assuntos pertinentes ao trabalho, o presente capítulo trata do detalhamento da nossa proposta. A presente proposta consiste assim, em um barramento tecnológico de troca de dados para a integração de serviços públicos digitais que auxilie na interoperabilidade entre diferentes entidades governamentais.

3.1 Interoperability on Data Bus - IO Data Bus

A utilização de uma camada de troca de dados que garanta uma melhor comunicação entre diferentes sistemas, proposta por este trabalho, se mostrou como uma poderosa aliada na busca pela digitalização de serviços públicos em países como Estônia e Finlândia.

Atualmente o governo brasileiro utiliza apenas padrões e regras de interoperabilidade, através do *e-PING*. Também já existe, no governo, um contexto de digitalização em andamento com uma proposta chamada de Governo Digital [Governo Digital 2020], cuja ideia é de unificar vários documentos de identificação dos cidadãos em um único. Esta unificação de documentos, de forma digital, é um dos fatores de sucesso na busca pela diminuição da burocracia em serviços públicos governamentais em outras nações [e-Estonia 2020].

Analisando essas situações, este trabalho propõe o *IO Data Bus* ou *Interoperability on Data Bus*, do português Interoperabilidade no Barramento de Dados. Esse barramento tem como objetivo conectar todos ou a maior parte dos serviços públicos digitais de esferas governamentais.

3.2 e-PING: Utilização de Segmentos do Framework

Como o nosso barramento propõe otimização em práticas interoperáveis em universos governamentais digitais, vamos utilizar algumas premissas presentes na arquitetura *e-PING*, do governo eletrônico brasileiro, para dar embasamento às técnicas propostas e partir de soluções que já são empregadas no *e-gov* do nosso país e que estão contidas nas especificações técnicas da segmentação do documento. A seguir listamos alguns itens que serão utilizados para dar suporte ao funcionamento do *IO Data Bus*:

Tabela 3.1: Componentes e Especificações do ePING presentes na nossa proposta

Segmento do ePING	**Componente a ser utilizado**	**Especificação**
Interconexão	Protocolos de transferência de aquivos	HTTP e FTP
Interconexão	Sincronismo de Tempo	NTP
Interconexão	Serviços de Nomeação de Domínio	DNS (gov.br) e DNSec
Interconexão	Transporte	TCP e UDP (quando necessário)
Segurança	Transferência de Dados em Redes Inseguras	TLS
Segurança	Certificado Digital	X.509 v3 - ICP-BRASIL
Segurança	Carimbo do tempo	Time-Stamp Protocol
Organização e Intercâmbio de Informações	Linguagem para intercâmbio de dados	JSON
Organização e Intercâmbio de Informações	Protocolo para acesso a Web Service	HTTP/1.1

A Tabela 3.1 lista os componentes que abstraímos da arquitetura de interoperabilidade do governo eletrônico, *e-PING* [Brasileiro 2018]. A maioria desses itens já estão com o nível de situação como (A) ou adotado pelo governo, mas alguns ainda estão em (R) ou recomendado, como o caso do Carimbo de Tempo, Certificados Digitais e Transferência de Dados em Redes Inseguras. Porém, na nossa proposta todos estes itens devem entrar como (A) adotados no funcionamento do barramento.

3.3 Arquitetura do Barramento

3.3.1 Núcleo: X-Road

Como principal suporte tecnológico na busca pela integração proposta na nossa camada, utilizaremos a camada de troca de dados da *e-estonia*, o *X-Road*. Com este já consolidado artefato, criaremos um ambiente de núcleo com sua implantação.

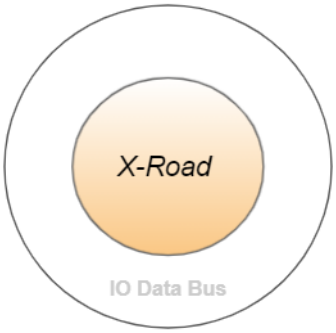

Figura 3.1: X-Road como Núcleo da Proposta

A Figura 3.1 ilustra um arcabouço do nosso barramento com a camada estoniana como núcleo das operações interoperáveis do nosso ambiente. Como explanado no capítulo de fundamentação teórica, o *case* de sucesso da Estônia Digital justifica a nossa escolha por seu barramento.

3.3.2 Visão Geral da Troca de Dados

Com a utilização do *X-Road* como núcleo da nossa arquitetura, abstraímos um arcabouço básico da visão arquitetural do nosso barramento.

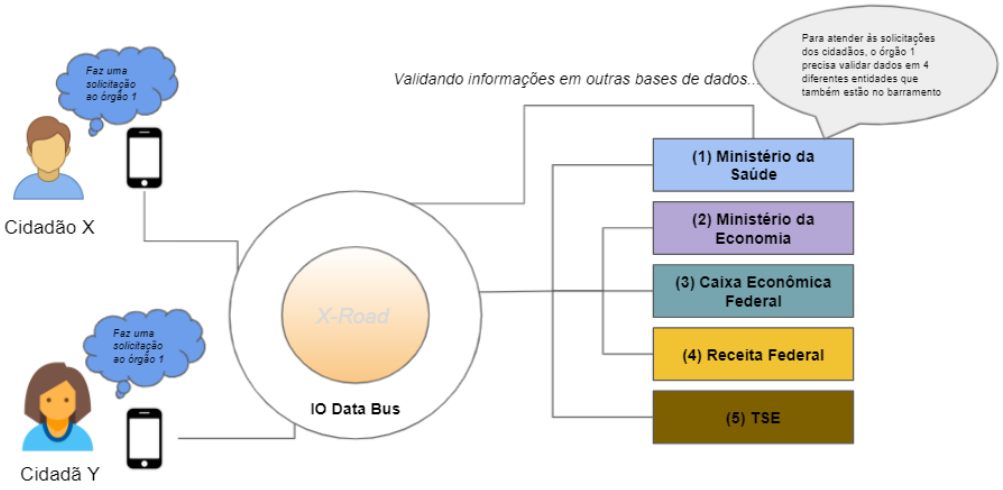

Figura 3.2: Visão Geral da Troca de Dados

A Figura 3.2 ilustra uma possível estrutura básica em um contexto genérico do *e-gov* brasileiro. Na interação ilustrada é possível observar que os cidadãos solicitam um serviço ou atendimento, no Ministério da Saúde, por exemplo.

Para atender tais solicitações, o Ministério da Saúde precisaria validar algumas informações presentes em outras entidades como o Ministério da Economia, CEF, Receita Federal e TSE. O *IO Data Bus* será responsável pelo gerenciamento da troca de dados entre estes *stakeholders*. O trabalho é desenvolvido no planejamento e execução de uma implementação real que forneça um protótipo funcional do barramento e o seu caso de uso simulatório.

3.3.3 Características da Arquitetura

Para a definição da característica da nossa visão arquitetural, utilizamos, como preceito, itens presentes na documentação da *e-PING*, além de características técnicas presentes na documentação oficial do repositório do *X-Road*. A seguir especificamos algumas características do nosso sistema:

- **Divisão em módulos:** O barramento é dividido em módulos *Cliente* e *Servidor*, que estão presente no núcleo do sistema;
- **Gravação de valores evidenciais (referências de *logs*) de transações com cópias em DLTs:** Visando garantir uma melhor segurança e imutabilidade de valor evidencial das transações ocorridas dentro do barramento;

- **Segurança criptográfica entre as entidades autorizadas do barramento:** Todos os sistemas conectados ao barramento irão se comunicar apenas por mensagens criptografadas por canal de segurança TLS, além da disponibilização de certificação digital de AC-Raiz ICP-Brasil para cada órgão integrante do barramento;

- **Descentralização de bancos de dados:** Apesar de possuir uma comunicação centralizada, por meio de cliente-servidor via barramento, os diversos Sistemas de Informação contarão com uma descentralização de dados e processamento, evitando possíveis pontos críticos de falha.

A Tabela 3.2 detalha os elementos que estão presentes na arquitetura do *IO Data Bus* e que englobam o processo de interoperabilidade no barramento para o Governo Eletrônico. Os módulos central, clientes e servidores são incorporados no núcleo pelo *X-Road*.

Tabela 3.2: Elementos do IO Data Bus

Elemento	**Característica**
Barramento	Próprio Sistema IO Data Bus.
Sistemas de Informação (SI)	Sistemas e Bancos de Dados governamentais que estarão conectados ao barramento.
Módulo Central	Gerência segurança, carimbos de tempo e logs.
Módulos Clientes	Módulos escaláveis que recebem requisições.
Módulos Servidores	Módulos escaláveis que recebem as requisições dos módulos servidores e encaminham aos SIs de destino.

Para um melhor entendimento desse cenário de arquitetura, vamos supor que um cidadão precisa ser atendido por algum serviço digital prestado pela DATAPREV, por exemplo. Para atender a solicitação desse cidadão, o SI desta empresa precisa de dados externos que se encontram nos bancos de dados dos órgãos Ministério do Trabalho e Receita Federal. Estes estão presentes no *IO Data Bus*, como a ilustração a seguir demonstra:

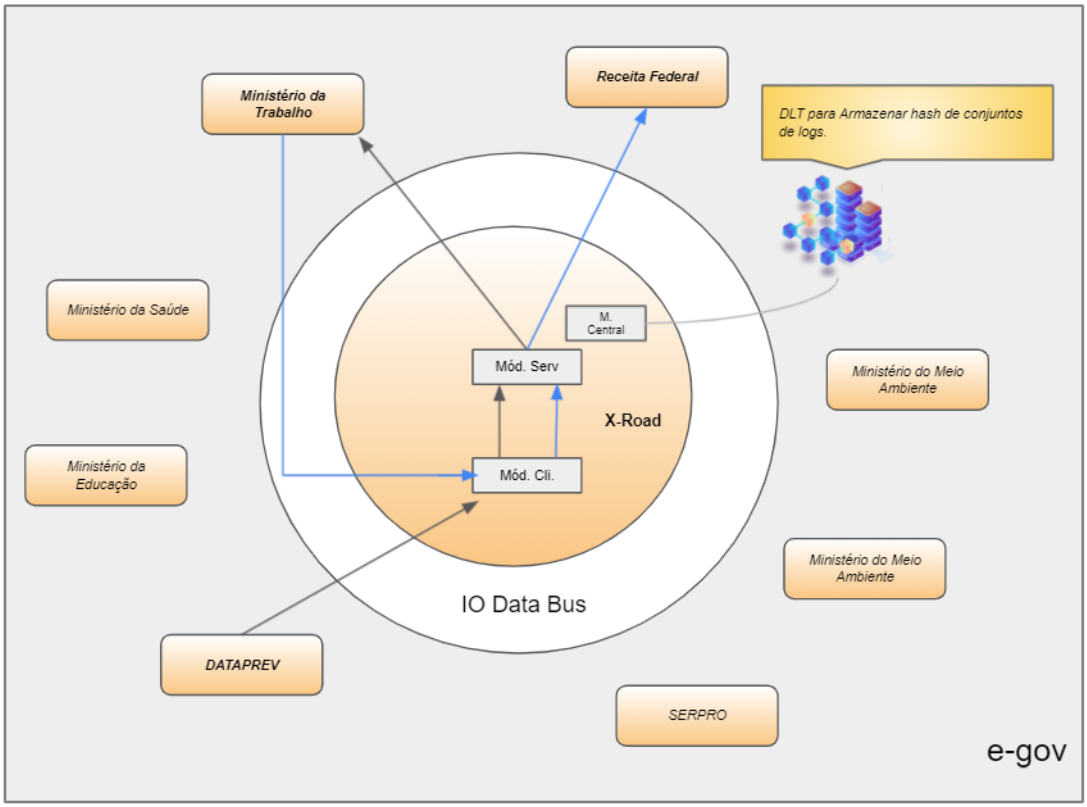

Figura 3.3: Ilustração da dinâmica entre órgãos no barramento

A Figura 3.3 mostra a dinâmica do exemplo citado acima. Antes de atender a solicitação da DATAPREV, o Ministério do Trabalho precisa buscar uma outra informação no banco de dados da Receita Federal, então, para isso, o Ministério do Trabalho faz a requisição para um Módulo Cliente que busca o Módulo Servidor que por sua vez entrega a requisição à Receita.

Durante este processo, a priori a evidência da solicitação via DATAPREV será enviada partindo do módulo central para uma DLT permissionada que será melhor detalhada na seção 3.5.

3.4 Fluxo de Integração ao Barramento

3.4.1 Autenticação de Serviços ao Barramento

Cada serviço governamental que ingressa no ecossistema interoperável do *IO Data Bus* precisa da obtenção de um certificado digital com AC-Raiz ICP-Brasil, seguindo a conformi-

dade com o guia *e-PING*.

Este certificado será utilizado pelos Sistemas de Informação participantes do barramento com o objetivo de atender as restrições de segurança de autenticação. Como ilustra a Figura 3.4, uma entidade governamental será responsável pelo gerencialmente do *IO Data Bus*. Este ente fornece o certificado digital necessário para que o serviço que precisa participar do sistema interoperável possa se autenticar no ambiente.

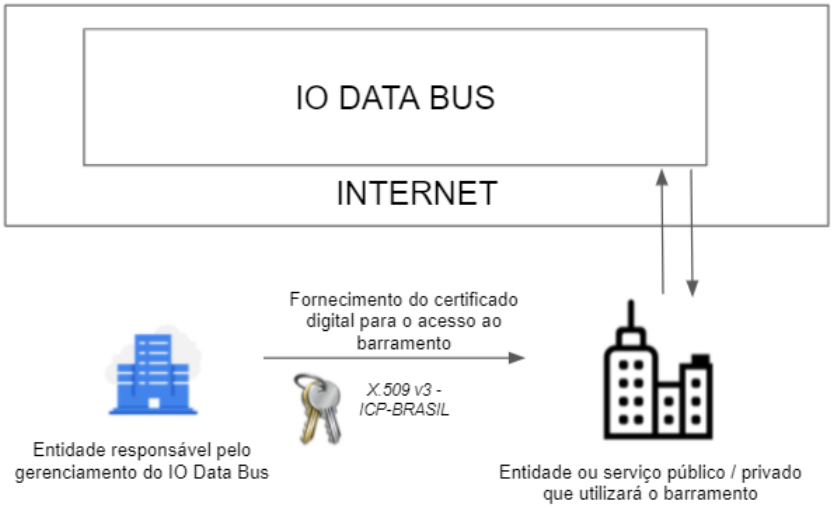

Figura 3.4: Estratégia de utilização do barramento

3.4.2 Semântica e Tratamento dos Dados

Com o arcabouço técnico da interoperabilidade definido, é preciso analisar como se dará a transformação dos dados para que aconteça uma semântica na interoperabilidade comum entre as distintas entidades participantes.

A comunicação via *REST*, presente nas especificações da *e-PING*, foi seguida ao utilizar o *X-Road* como núcleo da nossa proposta, visto que a camada estoniana utiliza esse tipo de tecnologia na sua troca de dados.

Para o lado do *IO Data Bus* foi escolhida a linguagem *GraphQL* para tratar os dados que se comunicam através do barramento. Assim, esta poderosa e recente linguagem de troca de dados, proposta pelo *Facebook* em 2012, poderá trazer a semântica para que as partes interessadas na comunicação estejam em sintonia na transformação de dados.

A Figura 3.5 ilustra uma possível comunicação entre dois serviços conectados ao barramento trocando mensagens em mesmo nível semântico, independentemente de plataforma ou linguagem de programação do SI.

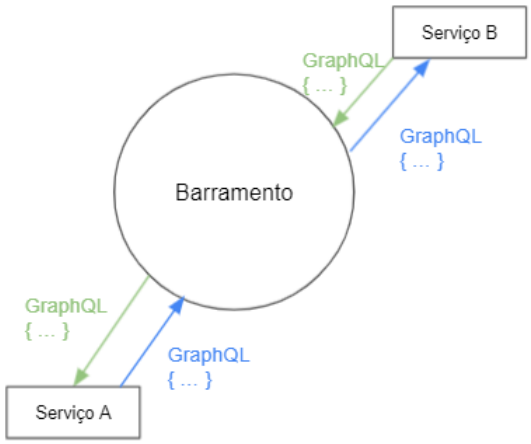

Figura 3.5: Comunicação utilizando o *GraphQL*

O Código Fonte 3.1 exemplifica uma *query* do fluxo de dados ilustrado anteriormente na Figura 3.3. Para atender a situação fictícia de requisição por parte da DATAPREV, são executadas as consultas de procurar um DNI na base do TSE. Com o respectivo DNI será possível consultar a regularidade do CPF na base da Receita Federal e também o situação empregatícia na base do MTE. O serviço de DNI será melhor detalhado no próximo capítulo.

Código Fonte 3.1: Abstração de *Query* GraphQL para Comunicação com as Diferentes Bases

```
1 type Query {
2     findCpfAndCTPSbyDni(dni: Integer): Cpf
3     findCtpsbyDni(dni: Integer): Ctps
4     findCpfRegular(cpf: Integer): CpfRegular
5     findDesempregados(ctps: Integer): SituacaoCtps
6     }
```

A estrutura de *Mutations*, para a execução das tarefas, é exemplificada no Código Fonte 3.2. No exemplo está sendo executada a função para a execução da solicitação de auxílio emergencial, que será melhor detalhado na nossa prova de conceito.

Código Fonte 3.2: Abstração de *Mutation* do Tratamento dos Dados em GraphQL

```
1
2 type Mutation {
```

```
3       solicitaAuxilioEmergencial(dni: Integer): AuxilioEmergencial
4   }
```

Com dois tipos tipos de tecnologias de troca de dados distintas, analisamos a utilização de uma camada de tradução de dados. Implementamos o *GraphQL-Rest-Adapter*, tradutor que recebe *queries* vindas em *GraphQL* e faz conversão das mesmas para REST (JSON) para assim serem reconhecidas no ecossistema do núcleo *X-Road*.

A Figura 3.6 ilustra uma possível comunicação entre dois serviços conectados ao barramento trocando mensagens em mesmo nível semântico, independentemente de plataforma ou linguagem de programação do SI.

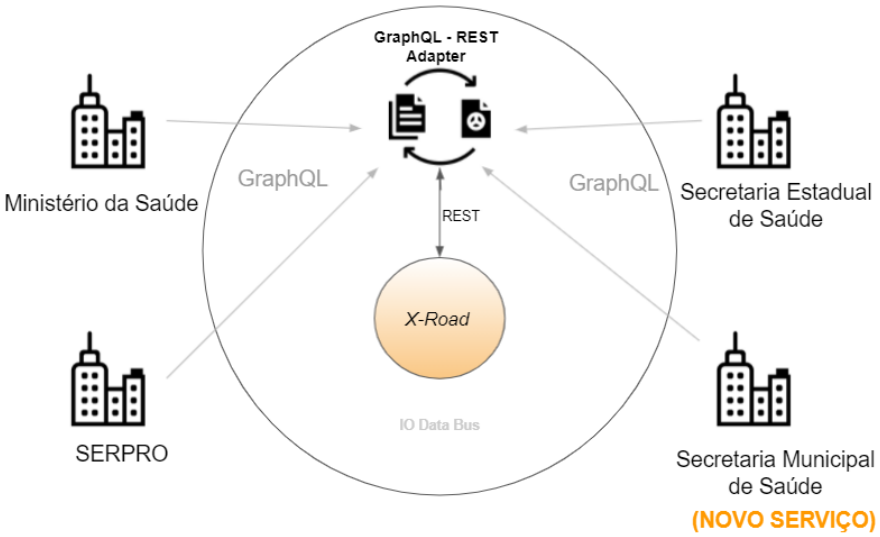

Figura 3.6: Camada de tradução utilizando *GraphQL* e *REST*

3.4.3 Integrando Novos Serviços ao Barramento

Para conceituar a integração de novos serviços, iremos utilizar como a exemplo a situação hipotética da Figura 3.6. Nesta ilustração há um fluxo de troca de dados, da área da saúde, entre o Ministério, SERPRO, Secretária Estadual da Saúde e também uma Secretaria Municipal de Saúde.

Supondo que um novo serviço da Secretaria Municipal, ilustrada, precise se integrar nessa troca de dados com os demais entes governamentais da interação, o fluxo ilustrado na Figura 3.7 traz a sequencia de passos para que isso ocorra.

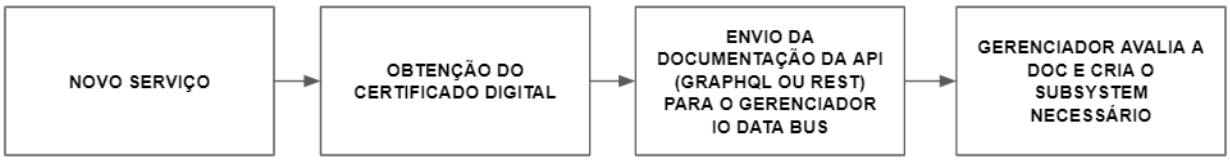

Figura 3.7: Fluxo para Integração de Novo Serviço

No primeiro passo, teríamos um serviço do controle da vacinação da *covid-19* no Brasil, desenvolvido pela SERPRO e gerido pelo Ministério da Saúde. Esse sistema atualiza diariamente a quantidade vacinas aplicadas em todos os municípios brasileiros. Os demais passos são descritos a seguir:

- **Obtenção do Certificado Digital:** O novo serviço integrado solicita sua certificação digital para a autenticação no barramento ao órgão gerente do *IO Dat Bus*;
- **Envio da Documentação da API:** O ente responsável pelo novo serviço a ser integrado prepara uma documentação padrão (a ser obtida junto ao gerenciador do barramento) descrevendo a necessidade a necessidade da integração juntamente com as *queries* de requisições e/ou especificações da API receptora de dados. Estas informações irão depender de cada aplicação e suas finalidades, para seguir a semântica dos dados será necessário apenas preparar esta troca de dados com a tecnologia *GraphQL*.

Código Fonte 3.3: Query GraphQL de Envio da Quantidade de Doses Aplicadas

```
1 type Query {
2     enviaQtdAplicacoes(quantidadeAplicacoes: Integer): Aplicacoes
3     }
```

O Código Fonte 3.3 pode ser utilizado, nesta exemplificação genérica, para demonstrar a *query GraphQL* responsável por enviar a quantidade de doses aplicadas no dia para Secretaria Estadual de Saúde, sendo esta a responsável por repassar ao sistema desenvolvido pela SERPRO e gerido pelo Ministério da Saúde.

- **Avaliação do Gerenciador e Criação dos *SubSystems* no X-Road:** Com todas as definições dos serviços validadas, o *subsystem* para o serviço em questão é criado dentro do Servidor de Segurança responsável, presente no núcleo *X-Road*.

3.5 Armazenamento de *hashes* de conjuntos de logs em DLT

Como já discorrido no capítulo de fundamentação teórica, a tecnologia de *blockchain*, através das DLTs, proporcionaram uma nova vertente para a utilização dos seus benefícios em contextos fora do mundo das cripto-moedas. Como ilustra a Figura 3.8, a priori a intenção de se utilizar uma DLT permissionada será para o armazenamento de conjuntos de *hashes* de evidencias das solicitações que serão detalhadas nos nossos experimentos.

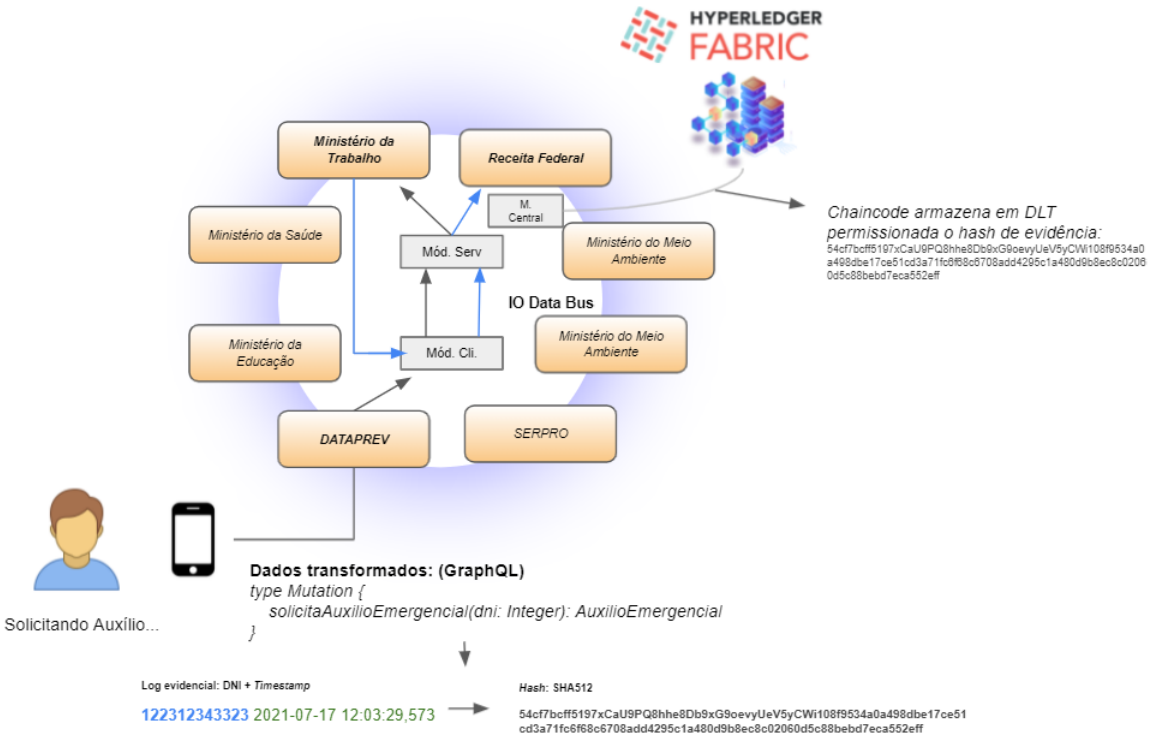

Figura 3.8: Armazenando *hash* evidencial em DLT permissionada

Na ilustração é possível observar que a cada nova requisição, o valor evidencial (id + *timestamp*) das transações são armazenadas em um ambiente *Hyperledger Fabric*. Dessa forma, é possível garantir a consistência de imutabilidade para evidencias de transações ocorridas em serviços públicos digitais.

3.6 Considerações do Capítulo

Neste capítulo foi discutido acerca da proposta da defesa, foram considerados aspectos interoperáveis para o fluxo de serviços públicos digitais. Apresentamos as principais características do *IO Data Bus*, tendo o enfoque nas considerações tecnológicas que deram o suporte para o nosso barramento. Na primeira seção houve um breve resumo introdutório da intenção da nossa pesquisa.

A seção seguinte trouxe um estudo da possibilidade da extração de alguns itens pertencentes aos Padrões de Interoperabilidade do Governo Eletrônico - ePING. Os itens extraídos da segmentação desse guia deram um maior suporte para propor vias tecnológicas que já estão em uso na interoperabilidade do nosso *e-gov*.

Posteriormente foram demonstradas as principais características da nossa proposta arquitetural do barramento. Explanamos o *X-Road* como a camada de interoperação núcleo do sistema para assim embasar nossa própria proposta de barramento.

O restante do capítulo trouxe as políticas para o acesso e utilização do barramento, como se dará a semântica e o tratamento dos dados trafegados na camada e uma proposta de armazenamento de *hashes* de evidências (*logs*) em DLT permissionada.

O próximo capítulo traz a prova de conceito da nossa proposta, contendo os experimentos práticas realizados para a obtenção dos resultados da pesquisa.

Capítulo 4

Experimentos e Resultados

Após a abordagem detalhada da nossa proposta, iremos especificar a implementação e testes para a validação da nossa prova de conceito a ser defendida neste trabalho.

4.1 Prova de Conceito

Para a proposta do *IO Data-Bus*, além da camada de troca de dados tecnológica, é interessante, menos burocrático e eficaz se ter um *e-serviço* auxiliar de Identificação Única e Digital que auxiliará a interoperabilidade entre diferentes sistemas governamentais. Algumas ideias de um documento unificado já foram apresentadas no Brasil, até mesmo pelo governo, mas ainda nenhuma dessas propostas foi colocada em prática para a população.

E sobre esses casos de identificação, este trabalho irá usar como base a proposta apresentada pelo governo federal no ano de 2018, chamada de DNI (Documento Nacional de Identificação) [DECRETO Nº 9.278 2018]. Através do Decreto nº 9.278, de 5 fevereiro de 2018, o DNI chegou a ser anunciado e uma de suas características era de que a centralidade dos dados e informações dessa documentação única seriam de responsabilidade do Tribunal Superior Eleitoral (TSE). Alguns dos documentos essenciais como Registro Geral (RG), Cadastro de Pessoa Física (CPF), Título de Eleitor e Certidão de Nascimento e Casamento estariam englobados nesse documento digital.

Como uma prova de conceito para a utilização do *IO Data-Bus*, foi escolhido um cenário atual que está sendo bastante utilizado no cenário do *e-gov* brasileiro e para uma maior adaptabilidade com o contexto de digitalização de serviços através de um barramento in-

teroperável, será utilizado o DNI como a identificação principal da população nas situações apresentadas durante esse prova. Esse cenário irá tratar da solicitação do auxílio emergencial para trabalhadores informais e desempregados disponibilizado pelo governo federal para o momento de crise causada pela pandemia do novo coronavírus.

Como referência para os *e-serviços* a seguir, utilizamos uma modelagem similar ao fluxo atual de troca de dados e informações desses serviços reais que estão vigentes no Governo Digital do Brasil, exceto o DNI. O objetivo dessa Prova de Conceito é caracterizar a simulação desses serviços digitais, demonstrando todo o fluxo de dados, através da camada *X-Road*, nas simulações das situações.

4.1.1 *e-Serviço* 1: DNI

Tabela 4.1: Alguns dos documentos englobados no DNI da prova de conceito

Documento	**Base de Dados**
RG	SSPs
CPF	RFB
Título de Eleitor	TSE
CTPS	MTE

A Tabela 4.1 lista alguns documentos que são armazenados na documentação única proposta. O Registro Geral - RG, estará sob responsabilidade das bases de dados das Secretarias de Segurança Públicas, o Cadastro de Pessoas Físicas- CPF estará na Receita Federal do Brasil,o Título de Eleitor no próprio TSE e a Carteira de Trabalho e Previdência Social é de responsabilidade do Ministério do Trabalho e Emprego. Ressaltamos que esses casos foram analisados meramente como hipóteses, com o intuito apenas de dar embasamento para a simulação prática.

Para envolver todos ou a maioria dos documentos da população brasileira, essa lista seria bem mais extensa, porém, propomos essa listagem de documentos contidos no DNI apenas para satisfazer as provas de conceito, prototipações e testes decorrentes às nossas abordagens. Com isso, o TSE armazenará essas informações neste *e-serviço* de identificação e quando necessário utilizará a camada de troca de dados interoperável para consultar as outras bases de dados.

4.1.2 *e-Serviço* 2: Auxílio Emergencial

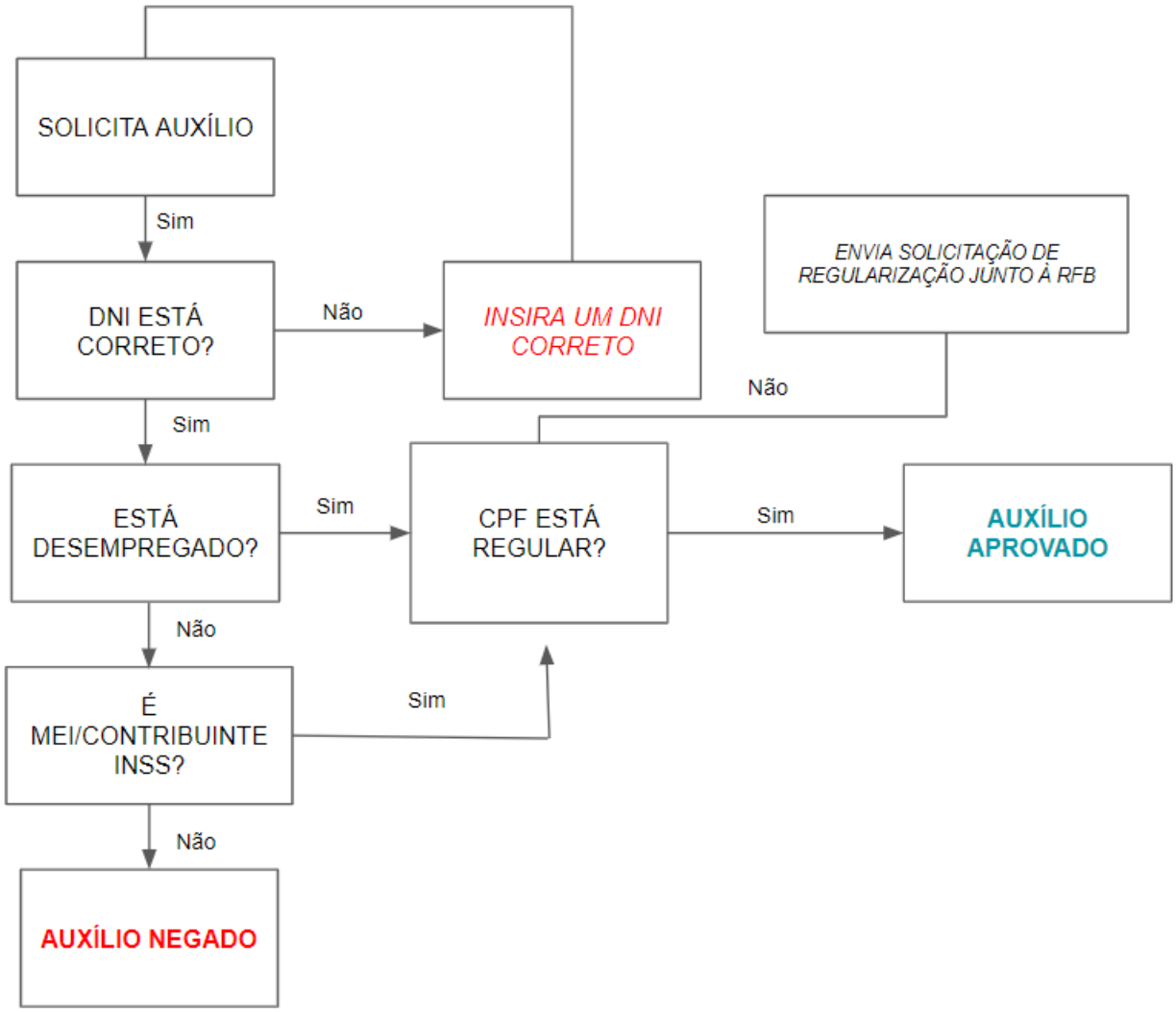

Figura 4.1: Fluxograma da Solicitação do Auxílio Emergencial na Prova de Conceito

Atualmente as solicitações de auxílios são processadas pela Empresa de Tecnologia e Informações da Previdência - DATAPREV, esta é a responsável por fazer todos os cruzamentos das informações em outras bases de dados para dessa forma, aprovar ou negar o auxílio de acordo com alguns pré-requisitos desse benefício, divulgados pela Caixa Econômica Federal [Auxílio Emergencial 2020]. Tal processamento e análise dos dados para a concessão do auxílio vem sendo bastante questionado devido a longa demora para a análise e parecer final da possível aprovação ao cidadão.

Nessa prova de conceito foram abstraídos alguns desses pré-requisitos para se realizar o experimento desse serviço através da camada de troca de dados *X-Road*, a mesma utilizada na Estônia Digital [e-Estonia 2020]. A Figura 4.1 ilustra o fluxograma dos requisitos que serão utilizados nessa prova para autorizar ou não o auxílio. Como é possível observar através do fluxo de dados, o auxílio emergencial só será aprovado para os DNIs que forem válidos, que estejam como desempregado(a) na base de dados do Ministério do Trabalho e que o CPF esteja como regular na base de dados da Receita Federal. Também estarão aptos ao auxílio os DNIs que estejam cadastrados como Micro Empreendedor Individual (MEI) no banco de dados da Receita ou que através da sua CTPS (Carteira de Trabalho e Previdência Social) estejam cadastros como Contribuinte Individual no INSS.

4.2 Prototipação

4.2.1 Prototipação do barramento de troca de dados utilizando o X-Road

Para a validação dos cenários de prova de conceitos proposta, foi utilizado o *X-Road* como núcleo do ecossistema interoperável. Nesse protótipo o barramento de troca de dados estoniano foi instalado com 4 módulos pertencentes a estrutura básica da arquitetura dessa camada. Essa camada foi instalada em um Sistema Operacional Ubuntu Server 18.04 rodando uma máquina virtual do *VirtualBox* com 10GB de Memória *RAM* e 100gb de espaço em disco.

```
r1@r1:~$ lxc list
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
|   NAME   |  STATE  |         IPV4         |                     IPV6                     |    TYPE    | SNAPSHOTS |
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
| demo-ca  | RUNNING | 10.206.36.5 (eth0)   | fd42:86cf:f305:4c67:216:3eff:fe0d:b861 (eth0) | PERSISTENT | 0         |
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
| demo-cs  | RUNNING | 10.206.36.29 (eth0)  | fd42:86cf:f305:4c67:216:3eff:fe69:b01e (eth0) | PERSISTENT | 0         |
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
| demo-ss1 | RUNNING | 10.206.36.253 (eth0) | fd42:86cf:f305:4c67:216:3eff:fe4a:3c48 (eth0) | PERSISTENT | 0         |
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
| demo-ss2 | RUNNING | 10.206.36.23 (eth0)  | fd42:86cf:f305:4c67:216:3eff:fe61:ca0e (eth0) | PERSISTENT | 0         |
+----------+---------+----------------------+----------------------------------------------+------------+-----------+
r1@r1:~$
```

Figura 4.2: *Conteiners* com os módulos do *X-Road*

A Figura 4.2 traz o *printscreen* da listagem dos *linux containers* instalados, um para cada

módulo do barramento. O módulo *demo-ca* presente no primeiro *container* é responsável por gerenciar os certificados de autoridade e gerenciar as evidências de transações dos dados através de carimbos de tempo (*timestamp*). O *demo-cs* traz o *Central Server* do *X-Road*, por conseguinte os módulos *demo-ss1* e *demo-ss2* trazem os dois Servidores de Segurança instalados nessa arquitetura da camada.

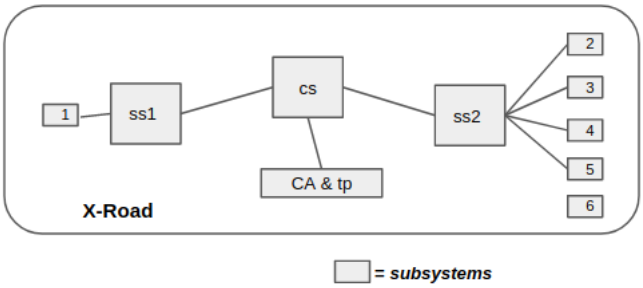

Figura 4.3: Arquitetura genérica do *X-Road* na simulação

Para atender as simulações de acesso de dados e informações em diferentes entidades do governo, foi configurado no Servidor de Segurança 2 (SS2) cinco *subsystems* que se conectarão aos seus respectivos órgãos e o Servidor de Segurança 1 terá apenas 1 *subsystem* no SS1, como ilustrado na Figura 4.3. Esse será responsável por receber (ser cliente) das requisições do SI de solicitação do Auxílio Emergencial e das requisições do sistema da Caixa.

A Figura 4.4 traz o *printscreen* da interface gráfica do *X-Road*, exibindo os clientes e *subsystems* instalados no Servidor de Segurança 2, além da demonstração do serviço para o módulo do MTE, apontando para a API desenvolvida, que será melhor detalhada a seguir.

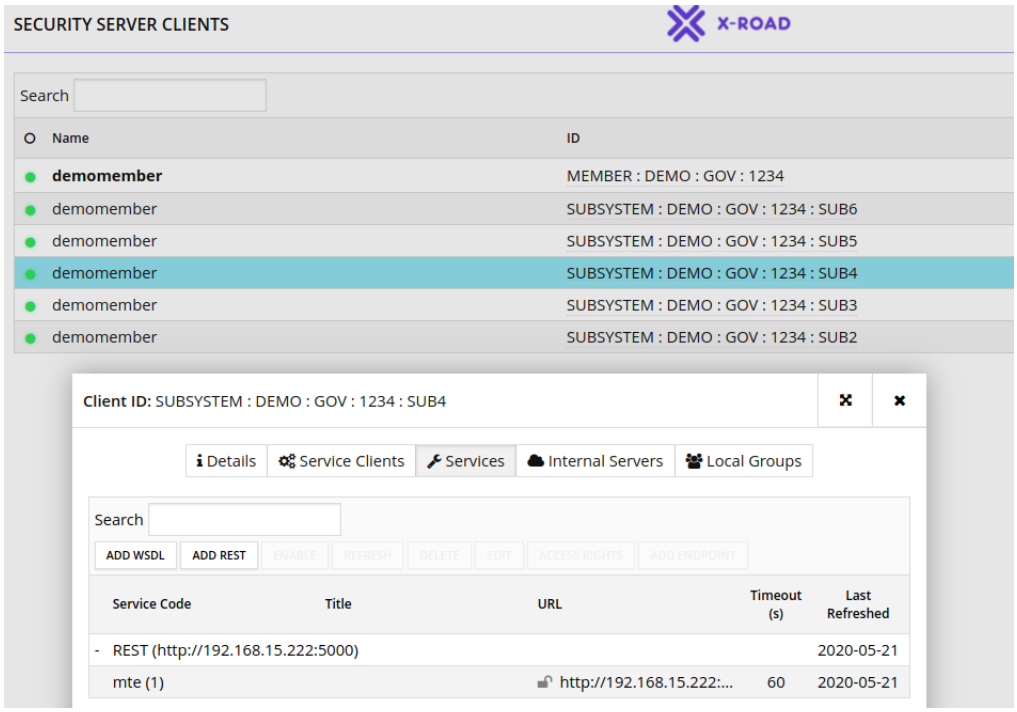

Figura 4.4: *Subsystems* do SS2 e Serviço do SUB4

4.2.2 Prototipação funcional dos e-serviços de DNI e Auxílio Emergencial

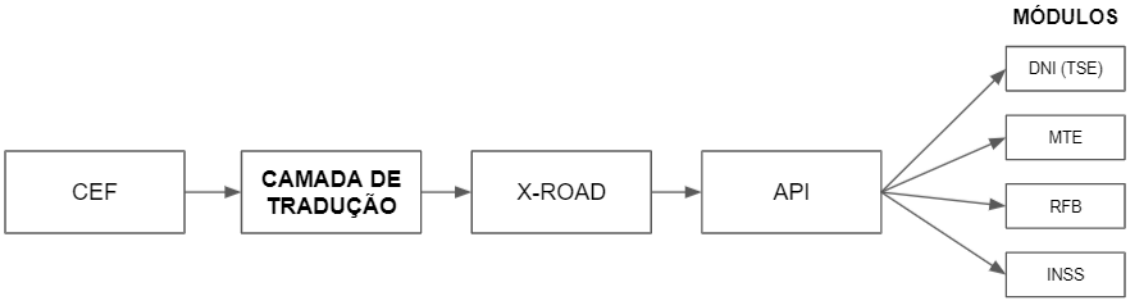

Figura 4.5: Fluxo genérico do protótipo

A Figura 4.5 ilustra o caminho das requisições que partem do Sistema de Informação da Caixa, que gerencia as solicitações de auxílio, até os SIs e bases de dados necessárias para as validações.

No segundo passo se inicia o processo de transformação de dados, seguindo a semântica proposta na nossa arquitetura, implementamos uma camada de tradução, o *GraphQL-Rest-Adapter*. Como pode ser visto no Código Fonte 4.1 foi desenvolvida uma API, na linguagem de programação Java, que recebe as requisições vindas da aplicação que solicita o auxílio e dos respectivos módulos que fazem parte deste ecossistema interoperável.

Código Fonte 4.1: Camada de Tradução GraphQL - REST

```
1  @Service
2  @GraphQLApi
3  public class CamadaTradutoraService {
4
5      private final CamadaTradutoraRepository CamadaTradutoraRepository;
6
7      public CamadaTradutoraService(CamadaTradutoraRepository camadaTradutoraRepository) {
8          this.camadaTradutoraRepository = camadaTradutoraRepository;
9      }
10
11     @GraphQLQuery(name = "solicitacaoAuxilio")
12     public void traducaoRequest(@GraphQLArgument(name = "acao") String acao,
           @GraphQLArgument(name = "nomeCompleto") String nomeCompleto, @GraphQLArgument(name
           = "dni") Integer dni, @GraphQLArgument(name = "dataNascimento") String
           dataNascimento ) {
13
14         //Recebe o request em GraphQL e envia ao X-Road em JSON
15
16         // Exemplo: Requisi  o Inicial:
17         String jsonRequest =  "{'acao':'"+acao+"', 'nome_completo':'"+nomeCompleto+"', 'dni
               ':'"+dni+"', 'data_nascimento':'"+dataNascimento+"'}";
18
19         runScript(jsonRequest);
20
21     }
22
23     //ENVIA O JSON PARA COMPONENTE PYTHON (CEF) RESPONS VEL PELA INTERA  O COM O X-ROAD
24     public void runScript(String jsonRequest){
25        Process process;
26        try{
27              process = Runtime.getRuntime().exec(new String[]{"cef_requests", jsonRequest})
                    ;
28              mProcess = process;
29        }catch(Exception e) {
30           log.error(e.toString());
31        }
32        InputStream stdout = mProcess.getInputStream();
33        BufferedReader reader = new BufferedReader(new InputStreamReader(stdout,
```

```
            StandardCharsets.UTF_8));
34          String line;
35          try{
36             while((line = reader.readLine()) != null){
37                  log.info("stdout: "+ line);
38             }
39          }catch(IOException e){
40                log.error("Exception in reading output"+ e.toString());
41          }
42  }
```

Assim como a tradução da requisição inicial, a camada tradutora, demonstrada acima, pode receber as demais requisições vindas dos outros módulos, sendo necessário apenas os ajustes dos respectivos campos do *GraphQL* para *JSON* recebidos nesta API GraphQL e encaminhadas para o módulo desejado.

Com os dados transformados, nas operações de semântica, módulos foram desenvolvidos para verificar em bancos de dados simulatório se tal dado vindo de um DNI é condizente com o pré-requisito do auxílio, por exemplo, se um CPF que veio de um DNI qualquer está na base de dados da RFB como Microempreendedor Individual.

Com a estrutura básica da camada interoperável instalada e funcional, foi implementada na linguagem de programação *Python* um *RESTful Web Service*, responsável por gerenciar as requisições de troca de dados via barramento dos módulos desse *e-serviço* e servir como API das transações, e os protótipos simulatórios dos outros módulos responsáveis por verificar os bancos de dados governamentais também foram desenvolvidos na mesma linguagem.

Código Fonte 4.2: Encaminhamento das requisições da Caixa para o X-Road

```
1  api = Flask(__name__)
2  def requisicao_atraves_do_xroad(dados):
3      try:
4          headers = {
5              'Content-Type': 'application/json',
6              'X-Road-Client': 'DEMO/GOV/1234/SUB1',
7          }
8
9          data = str(json.dumps(dados))
10
11         if dados["acao"] == "request_dni":
12             response = requests.post('http://10.206.36.253/r1/DEMO/GOV/1234/SUB3/dni-tse/
                   api', headers=headers, data=data)
13
```

```
14          elif dados["acao"] == "request_mte":
15              response = requests.post('http://10.206.36.253/r1/DEMO/GOV/1234/SUB4/mte/api',
                    headers=headers, data=data)
16
17          elif dados["acao"] == "request_rfb_cpf" or dados["acao"] == "request_rfb_mei":
18              response = requests.post('http://10.206.36.253/r1/DEMO/GOV/1234/SUB5/rfb/api',
                    headers=headers, data=data)
19
20          elif dados["acao"] == "request_inss":
21              response = requests.post('http://10.206.36.253/r1/DEMO/GOV/1234/SUB6/inss/api',
                     headers=headers, data=data)
22
23      except (Exception) as e:
24          print(e)
25
26      finally:
27          return response.text
```

O Código Fonte 4.2 possui a função de processar as requisições do SI da Caixa Econômica Federal que fazem as validações nos outros sistemas do *e-gov*. Nesse código é possível perceber que especificamos no cabeçalho da requisição (linha 5) que se trata de um tipo de dado *JavaScript Object Notation* (*JSON*), pois toda comunicação de entrada e saída é feita pelo estilo de arquitetura de software *REST*, para a troca de dados. Além disso, já é especificado que o o cliente das requisições dentro do barramento será o *subsystem* 1 presente no SS1, na linha 6. Da linha 11 à 21 estão presentes as condicionais que recebem as requisições e dependendo do seu tipo as encaminham para o *subsystem* correspondente. Por exemplo, se no corpo do *JSON* existir a variável *acao* com o valor *request_mte*, essa requisição será encaminhada para o subsystem 4 do SS2 e esse enviará para o módulo do Ministério do Trabalho e Emprego.

Código Fonte 4.3: *API REST* para comunicação com o barramento

```
1  api = Flask(__name__)
2
3  @api.route('/api', methods=['POST'])
4  def acao_do_post():
5
6      data = request.json
7
8      response = {}
9
10     if data["acao"] == "request_cef":
11         response = jsonify(recebe_solicitacao_cef(data))
```

```
12      elif data["acao"] == "request_dni":
13          response = jsonify(recebe_solicitacao_validacao_dni(data))
14      elif data["acao"] == "request_mte":
15          response = recebe_solicitacao_checagem_empregados_e_desempregrados(data)
16      elif data["acao"] == "request_rfb_cpf":
17          response = recebe_solicitacao_checagem_verifica_cpf(data)
18      elif data["acao"] == "request_rfb_mei":
19          response = recebe_solicitacao_checagem_verifica_mei(data)
20      elif data["acao"] == "request_inss":
21          response = recebe_solicitacao_checagem_contribuinte_indv_INSS(data)
22
23      return response
```

O Código Fonte 4.3, implementa um básico serviço de API com a micro-*framework Python Flask*, da linha 10 à 22 as condicionais recebem requisições vindas do barramento e as direcionam para o módulo correspondente do protótipo de teste. Por exemplo, se a requisição do tipo *POST*, possuir no seu corpo a variável *acao* com o valor *request_mte*, o *webservice* irá direcionar esse *request* para o módulo do Ministério do Trabalho e Emprego (MTE) e esse módulo vai consultar em seu banco de dados se o CPF (vindo do correspondente DNI) está como desempregado ou empregado.

Código Fonte 4.4: Módulo do Ministério do Trabalho e Emprego

```
1  import requests
2  import json
3  from base_de_dados import base_mte_empregados
4
5  def recebe_solicitacao_checagem_empregados_e_desempregrados(dados):
6
7      empregado = ([i for i in base_mte_empregados() if i["cpf"] == int(dados["cpf"]) and i['
           empregado'] == True] or [None])[0]
8
9      if empregado is not None:
10         return {"empregado": empregado["empregado"]}
11     else:
12         return {"empregado": False}
```

O módulo correspondente ao MTE é visto no Código Fonte 4.4, implementamos uma função simples que apenas verifica se o CPF vindo do DNI está como empregado ou desempregado na base do Ministério. Esse mesmo processo acontece para todos os outros órgãos (módulos da simulação) no processo interoperável da validação do auxílio via *X-Road*.

Código Fonte 4.5: Banco de Dados simulatório

```
1  def base_mte_empregados():
2
3      base = [
4          {"cpf": 123321123,
5          "empregado": False},
6          {"cpf": 648812677,
7          "empregado": False},
8          {"cpf": 648812455,
9          "empregado": True},
10         {"cpf": 648810000,
11         "empregado": True}
12     ]
13
14     return base
```

O código fonte acima, número 5.2, ilustra um trecho de uma base de dados temporária em um *list* simples da linguagem *Python*, em Memória RAM, apenas para fins de testes do protótipo. Foi utilizado o mesmo tipo de método para as demais bases de dados acessadas pelos outros módulos.

4.3 Projeto de Experimentos e Resultados Obtidos

Com a prototipação funcional operante, foram planejados e realizados uma série de experimentos para validação da prova de conceito proposta. Para isso foi criado um conjunto de dados fictícios em uma simulação de banco de dados, como apresentado na seção anterior. A Tabela 5.2 lista os dados pertinentes a cada cidadão fictício que fez a solicitação do auxílio emergencial nos testes e estão presentes na nosso banco de dados dos módulos.

Tabela 4.2: Parte do Banco de Dados do DNI (TSE)

DNI	**Nome Completo**	**Data de Nascimento**	*CPF*	*CTPS*
123321123	Joao Silva	12/12/1955	648812677	78941548484485
123325553	Maria Sampaio	12/12/1955	648812677	78941548484485
123325669	Joana Marques	05/02/1985	648812455	78941548486688
123325784	Alvaro Pereira	04/11/1990	648810000	78941548480078

Serão executados no nosso programa solicitações de auxílios para testar a interoperabilidade entre os módulos (bancos de dados fictícios) e simular a comunicação entre diferentes

órgãos da esfera do *e-gov* brasileiro.

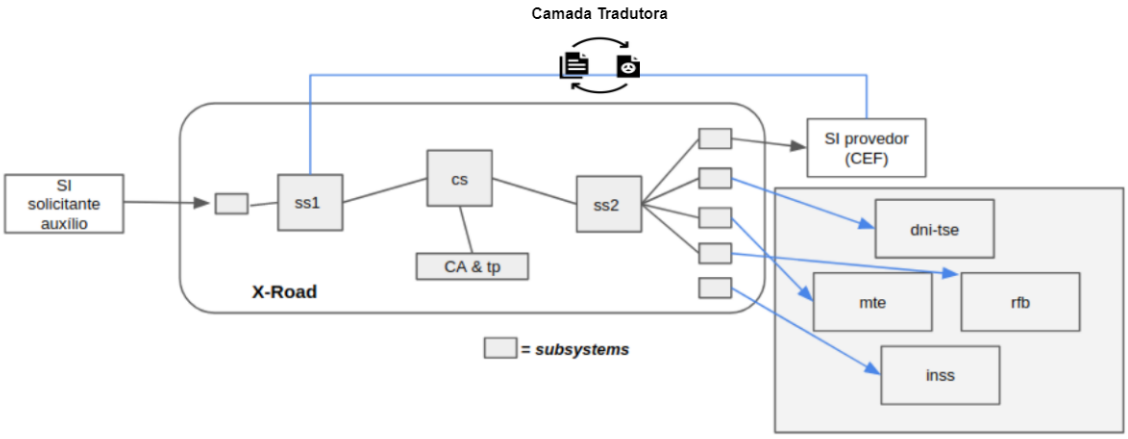

Figura 4.6: Fluxo detalhado do experimento com o *X-Road*

A Figura 4.3 mostra o fluxo mais detalhado de como os dados irão trafegar na nossa camada interoperável. O seguinte passo-a-passo das transações podem trazer um melhor entendimento desse fluxo utilizado no nosso Projeto de Experimentos para a obtenção dos resultados esperados:

- No módulo do *software* de requisição do auxílio foram inseridos os *inputs* necessários para a validação do respectivo DNI na base do TSE (Nome Completo, DNI e Data de Nascimento);
- Este primeira requisição entra no *X-Road* e se dirige ao *subsystem* 1 do Servidor de Segurança 1 (esse será empregado como cliente nas requisições);
- No código-fonte do módulo está implementado que o Cliente *X-Road* é o Servidor de Segurança 1, ou seja, os dados irão entrar por ele, porém o destino da requisição está apontada para o *subsystem* 2, pertencente ao Servidor de Segurança 2 e esse *subsystem* tem como cliente o módulo do SI provedor, que é a própria Caixa Econômica Federal;
- Entretanto, antes que a Caixa forneça o auxílio, ela precisa validar algumas informações com outros Sistemas de Informação do *e-gov* brasileiro. Toda essa interação da CEF com os demais órgãos vai acontecer por meio de comunicação via *GraphQL* para a tradução em *REST* por meio da camada tradutora demonstrada anteriormente;

- Nesse ponto é que se inicia a interoperabilidade necessária para validar esses dados, os fluxos em setas de cor azul demonstram esses passos;

- Como o *subsystem* 1 é o cliente que recebe as requisições, então a CEF irá enviar os pedidos de validação nos outros sistemas a partir dele, mas como destino os *subsystems* do Servidor de Segurança 2 correspondente aos outros órgãos irão receber essas validações;

- Esses passos de validações irão ocorrer pelo *susbsystem* 2 ao módulo *dni-tse* (validar se o DNI fornecido é válido e pertence realmente a pessoa solicitante), *subsystem* 3 ao módulo *mte* (validar se a pessoa está empregada ou desempregada no banco de dados de Ministério do Trabalho e Emprego), *subsystem* 4 ao módulo *rfb* (validar se é Microempreendedor Individual e se o CPF está regular na base da Receita Federal do Brasil) e por fim consultar o *subsystem* 5 ao módulo *inss* (validar se é Contribuinte Individual da Previdência Social na base de dados do INSS);

- Após a checagem dos pré-requisitos nesses módulos e receber meio de mensagem *REST* todas essas informações, nosso protótipo verifica as condicionais de pré-requisitos. O fluxo foi detalhado na Figura 4.3, retorna ao usuário se o benefício foi aprovado ou negado.

Após a análise e implementação desse fluxo, realizamos o primeiro experimento utilizando a cidadã Maria Sampaio de DNI número 123325553 (um dos cidadãos não reais cadastrados no nosso banco de dados simulatório do DNI). Após o a interoperabilidade do barramento realizar todo o cruzamento de dados necessário nos outros órgãos foi retornado o resultado da solicitação do auxílio.

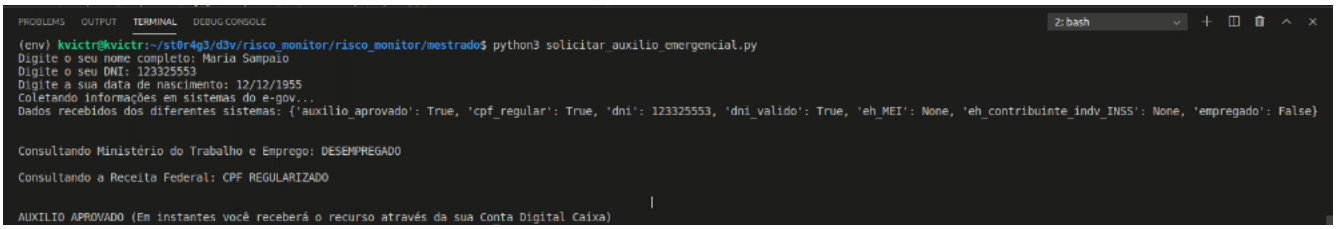

Figura 4.7: *Printscreen* Output do Resultado do Protótipo Funcional

A Figura 4.7 mostra o captura de tela do *output* após os dados trafegarem via barramento, após utilizar os dados da cidadã, Maria Sampaio, nosso protótipo trafegou e cruzou os dados nos diferentes módulos do órgãos e banco de dados e teve como resposta a mensagem *REST/JSON* explicitada no *printscreen*. Com essa mensagem nossa simulação da aplicação de solicitação de auxílio fez as comparações e processamentos necessários, para que, de acordo com os pré-requisitos já mencionados, autorizar ou não o auxílio. Nesse caso podemos perceber que o auxílio foi aprovado.

Na Figura 4.8 ilustramos a execução do Sistema de Informação de solicitação que implementamos. A requisição é enviada ao barramento, o SI provedor, a alusão à Caixa Econômica Federal, busca validar as informações em outras bases dados. No nosso resultado, a nossa cidadã fictícia, Maria Sampaio, teve seu benefício autorizado porque atendia todos os pré-requisitos necessários para a obtenção do auxílio.

Já o cidadão, João da Silva, teve o auxílio negado por não atender a um dos requisitos em alguma das bases de dados verificadas através do barramento. Todo o *output* e resultados dos nossos testes, incluindo dos cidadãos Joana Marques e Álvaro Pereira, estão presentes no repositório do projeto, disponível no endereço **https://github.com/iodatabus/auxilio-emergencial-com-xroad.git**.

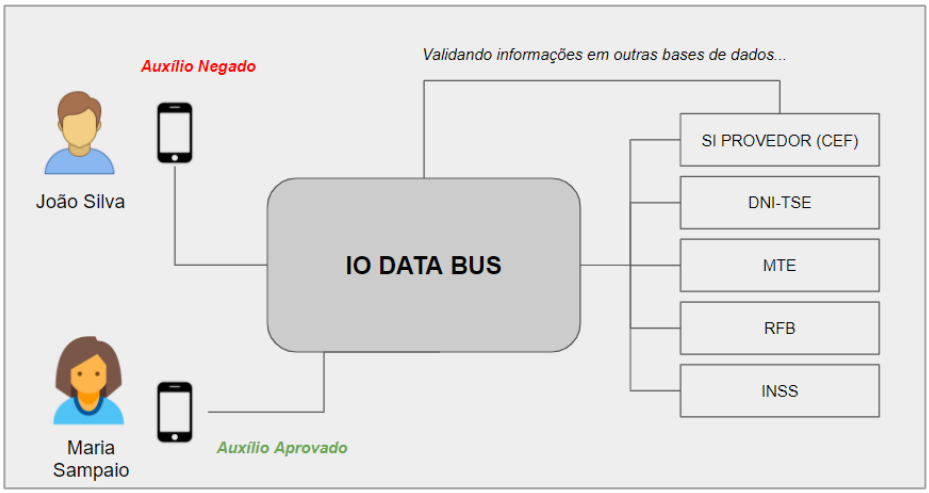

Figura 4.8: Executando a Solicitação do Auxílio no Barramento

4.4 Armazenando valores evidenciais no *Hyperledger Fabric*

Com os experimentos realizados no núcleo *X-Road*, neste momento propomos o armazenamento de valores evidencias dessa solicitações em ambiente de livro-razão distribuído. Essa funcionalidade será diretamente interligada com o nosso núcleo do barramento estoniano demonstrado anteriormente.

4.4.1 Transformando a evidência em *hash*

Visando uma maior privacidade, por questões de segurança, não iremos armazenar o valor real de carimbo de tempo e *id* da solicitação de auxílio emergencial, dados estes que servirão como nossa referência. Transformamos esses valores em função *hash* do tipo SHA-512. O Código Fonte 4.6 é responsável pela conversão desses itens em *hash*.

Código Fonte 4.6: Transformando a evidência da solicitação em hash SHA-512

```
1
2  import hashlib
3
4  def main(idSolicitacao, timestamp):
5      print("Transformando o timestamp e id da solicita  o de aux lio em fun  o hash..."
           )
6      hash_object = hashlib.sha512(bytes(idSolicitacao)+bytes(timestamp.encode("utf-8")))
7      hex_dig = hash_object.hexdigest()
8      print(hex_dig)
9      return hex_dig
10
11
12 if __name__ == "__main__":
13     main(10202202, "2021-07-17 12:03:29,573")
```

4.4.2 Preparação da API

Com o *hash* evidencial pronto, neste momento preparamos o atendimento da funcionalidade de armazenamento de valores evidenciais, descrita no capítulo da proposta, foi implementado um conjunto de códigos fontes, como protótipos funcionais, para atender tal necessidade.

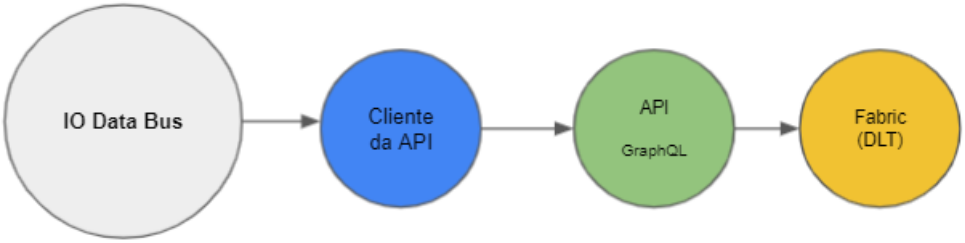

Figura 4.9: Fluxo do envio do *hash* evidencial para a DLT

A Figura 4.9 ilustra o fluxo do nosso experimento através de uma API *GraphQL*, em um primeiro momento implementamos um cliente para essa API, na linguagem de programação *Python*. Esse cliente recebe o valor evidencial vindo do barramento solicitação e faz a requisição HTTP, de uma *Mutation* até a nossa API. O Código Fonte 4.7 demonstra essa implementação.

Código Fonte 4.7: Cliente da API GraphQL enviando o valor do hash

```
1
2  from gql import gql, Client
3  from gql.transport.aiohttp import AIOHTTPTransport
4
5  def apiCliente(hashValue):
6
7      transport = AIOHTTPTransport(url="http://192.168.0.5:8080/")
8
9      client = Client(transport=transport, fetch_schema_from_transport=True)
10
11     query = gql(
12         """
13             mutation {
14                 saveSolicitacao(solicitacao: { value: """+hashValue+""" })
15             }
16     """
17     )
18     result = client.execute(query)
19     print(result)
```

Para receber essa solicitação, foi implementa uma API na linguagem *Java*, utilizando o *Spring Boot Framework*. Essa codificação possui uma classe *Service* principal que vai tratar todas as requisições recebidas. A escolha de uma outra linguagem de programação reforça a importância da transformação dos dados em *GraphQL*, gerando assim uma maior

adaptabilidade, independência de plataforma e padronização. O Código Fonte 4.8 traz a classe dos serviços.

Código Fonte 4.8: API em GraphQL das Solicitações de Auxílio

```
/**
 * IMPORTS
 **/

@Service
@GraphQLApi
public class SolicitacaoService {

    private final SolicitacoesRepository solicitacoesRepository;

    public SolicitacaoService(SolicitacoesRepository solicitacoesRepository) {
        this.solicitacoesRepository = solicitacoesRepository;
    }

    @GraphQLQuery(name = "solicitacoes") // Ler todas
    public List<Food> getSolicitacoes() {
        return solicitacoesRepository.findAll();
    }

    @GraphQLQuery(name = "solicitacoes") // Busca por id
    public Optional<Solicitacao> getSolicitacaoById(@GraphQLArgument(name = "id") Long id)
        {
        return solicitacoesRepository.findById(id);
    }

    @GraphQLMutation(name = "saveSolicitacao") // Armazenar Solicita   o
    public Solicitacao saveSolicitacao(@GraphQLArgument(value = "solicitacao") Solicitacao
        solicitacao) {
        return solicitacoesRepository.save(solicitacao);
    }

}
```

4.4.3 Discussão do Resultado

Para discutir os resultados que tivemos com a nossa prototipação utilizando um barramento de dados interoperável, vamos fazer uma análise de como a DATAPREV faz atualmente o processamento das informações para a disponibilização do Auxílio Emergencial.

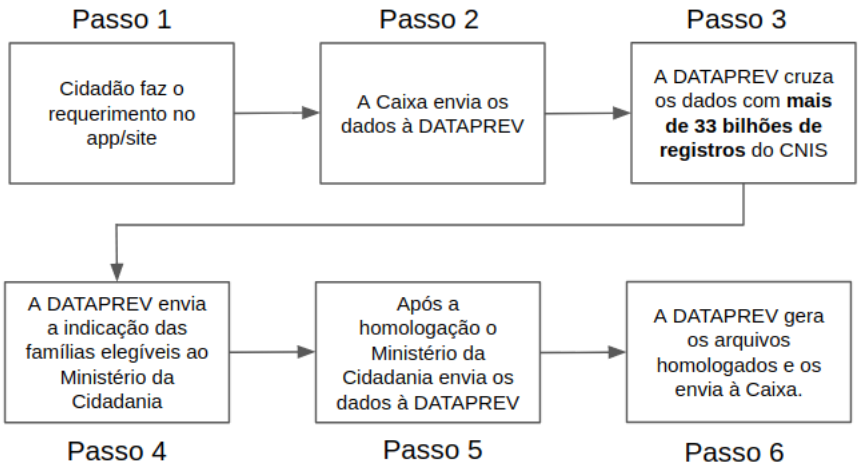

Figura 4.10: Fluxo atual da solicitação do auxílio pela DATAPREV

A Figura 4.10 ilustra o fluxo atual de uma requisição de auxílio processada pela DATAPREV. Como pode ser visto, a DATAPREV recebe a solicitação da Caixa, depois cruza os dados na base do Cadastro Nacional de Informações Sociais (CNIS) com mais de 33 bilhões de registros, após esse cruzamento envia as indicações de famílias elegíveis ao auxílio para o Ministério da Cidadania que devolve a solicitação à DATAPREV após a homologação, só depois de todo esse processo é que a DATAPREV gera os arquivos homologados e os envia à Caixa [Auxílio Emergencial 2020].

Essa solicitação está sendo crucial para o momento de crise econômica causada pela pandemia, contudo, nesse contexto, essas solicitações ainda estão demorando até semanas para serem retornadas. Um dos fatores que poderiam justificar essa longa espera na validação dos dados para o auxílio é a falta de uma interoperabilidade consistente no universo do *e-gov* do Brasil.

De acordo com [RIBEIRO, RIBEIRO e LEITE 2019], o CNIS é um banco de dados do governo federal criado em 1989 que armazena informações trabalhistas e previdenciárias dos trabalhadores brasileiros. Possuindo dados dos vínculos empregatícios desde 1976, as remunerações mensais a partir de 1990 e os recolhimentos dos contribuintes individuais desde 1979.

Nesse molde é possível perceber a massiva centralização de informações em um único banco de dados, podendo gerar um ponto crítico de falha e ocasionar possíveis *overloads* nessa base, visto que milhões de brasileiros estão solicitando esse auxílio diariamente. Com nosso modelo interoperável proposto, existiria uma melhor descentralização de informações

e dados, pois vários órgãos do Governo Eletrônico estariam presentes no barramento e as informações poderiam ser buscadas em bancos de dados descentralizados e autônomos, gerando uma melhor qualidade na busca pela informações e maior velocidade no atendimento ao cidadão.

Capítulo 5

Trabalhos Relacionados

Nesta capítulo descrevemos alguns trabalhos presentes na literatura que possuem relação com a estratégia proposta nessa dissertação, considerando as diferenças e semelhanças com o presente trabalho.

A impulsão da digitalização na entrega de serviços para a sociedade em esferas governamentais vem se fazendo como fator necessário ao decorrer dos anos. Em 2004 foi publicado o *The European Interoperability Framework for Pan-European E-Goverment Services* [Framework 2004]. O surgimento desse guia teve como objetivo definir um conjunto de recomendações para serviços de Governo Eletrônico em que administrações públicas, empresas e cidadãos pudessem interagir além fronteiras no contexto pan-Europeu.

Já em [Srivastava e Teo 2004], os autores trouxeram um outro *framework* de transformação em Governo Eletrônico. Esse guia explorou essa transformação através de facilitadores e marcos nomeados. A utilização de guias de boas práticas em *e-gov* também é utilizada no contexto brasileiro, com o *e-PING*, ou Padrões de Interoperabilidade do Governo Eletrônico [Brasileiro 2018]. Antes da aplicação de qualquer prática tecnológica no barramento realizamos um estudo das práticas utilizadas no guia.

As abordagens descritas anteriormente utilizam-se de guias teóricos para dispor das melhores práticas acerca de interoperabilidade e governo eletrônico, assemelham-se em parte com a nossa proposta pois utilizaremos algumas premissas do *e-PING* para a visão arquitetural do barramento.

Na aliança com essas boas práticas, se fez necessário enxergar artefatos tecnológicos que forneçam um melhor suporte aos modelos interoperáveis de *e-gov*. Segundo [Jaeger e

Thompson 2003], o principal propósito na busca de soluções de interoperabilidade no setor público é a habilitação da troca de dados dentro da organização pela internet. Também é importante conscientizar pessoas entre os cidadãos e funcionários do governo. Isso oferece vantagens na direção a um governo eletrônico. Essa utilização de uma solução tecnológica de interoperabilidade está presente na proposta do nosso trabalho visando uma melhor integração de serviços públicos.

O desenvolvimento de uma plataforma de interoperabilidade de e-Serviços no Brasil para além do *e-PING* foi avaliada por [Barros, Cepik e Canabarro 2010], no artigo os autores, consultores contratados pelo Banco Interamericano de Desenvolvimento (BID), apresentam uma síntese de avaliações para o desenvolvimento de um plano de ações para uma possível construção dessa plataforma, que teria o nome de Plataforma de Integração de Serviços Públicos [Projeto BR-TI066], do BID e da Secretaria de Logística de TI do antigo Ministério do Planejamento, Orçamento e Gestão (SLTI/MPOG).

Essa proposta, de 11 anos atrás, elenca apenas aspectos e possíveis modelos que uma possível solução tecnológica poderia proporcionar na interoperabilidade entre entes governamentais, mas não chega a detalhar essa solução e não utilizam simulações que embasem a proposta na prática. Assemelha-se com o nosso trabalho pois também buscamos, em partes, o mesmo objetivo geral. Entretanto, diferentemente desse trabalho, além de propor tal solução detalhamos toda a visão arquitetural e tecnológica e utilizamos uma prova de conceito para defender a proposta.

Em [Nielsen 2017] são descritos cinco itens considerados como essenciais e pré-requisitos de Governo Eletrônico e que ocasionaram, por exemplo, o sucesso das soluções estonianas no seu modelo de *e-gov*. Esses itens são a infraestrutura, o grau de instrução digital da população, identificação única e digital, registros digitais e troca de dados formal. Dessa forma, apresentaremos uma proposta de Identificação Digital e Única para nossa prova de conceito.

Open-source e chave da digitalização de sucesso entre serviços públicos e privados são características que trouxeram e trazem com o *X-Road* vários benefícios na sua utilização. Argentina, Azerbaidjão, El Salvador, Ilhas Faroe, Finlândia, Islândia, Quirguistão, Namíbia e Palestina são exemplos de países, citados por [Saputro et al. 2020], que reconheceram os conceitos do barramento estoniano e implantaram soluções semelhantes para seus modelos

de *e-gov*. Porém, maiores detalhamentos específicos dessas soluções não foram encontradas em suficiência na literatura.

Através desse fato, pôde ser observado em [Saputro et al. 2020] a análise de pré-requisitos na adoção do *X-Road* em um país, através de um estudo comparativo. Esses pré-requisitos foram validados através de entrevistas qualitativas e identificando alguns pré-requisitos adicionais. Servindo como base para a aplicação na nossa simulação de contexto de *e-gov* no governo brasileiro. Observando assim que somente a implantação de um barramento como *X-Road*, não significará o total benefício e sucesso de uma solução de governo digital, sendo necessário um conjunto itens como capacidade técnica, Políticas de Segurança da Informação, Identificação Única e outros aspectos culturais.

A literatura é vasta nas análises de soluções de Governo Eletrônico e temas que rodeiam o assunto, porém, o *X-Road* se dá como a principal, ou o ponto que norteia e remete inspiração para novas soluções, encontrada acerca da especificidade da construção e proposição de uma camada ou barramento tecnológico para integração de serviços públicos. No Capítulo 2 tal solução será melhor detalhada.

Capítulo 6

Conclusão e Trabalhos Futuros

De acordo com o que foi exposto, desde a fundamentação teórica sobre os aspectos gerais de Governo Eletrônico e principais tecnologias que podem fazer parte desse ecossistema, este trabalho propôs uma camada interoperável para integração de serviços públicos digitais.

A proposta visa que esta camada, a *IO Data Bus*, funcione como uma facilitadora na troca de dados entre sistemas de informação governamentais.

Este barramento, neste momento, tem como núcleo a camada de troca de dados que impulsionou a era digital de sucesso do governo eletrônico estoniano, a *X-Road*.

A defesa consistiu na implantação do sistema da Estônia Digital em ambiente local, dando todo o suporte necessário para a implementação de algumas funcionalidades que pudessem complementar os itens da nossa proposta que conta com os seguintes preceitos básicos:

- Utilização do guia de interoperabilidade do governo eletrônico, o *e-PING*, como referência na aplicação prática de alguns dos seus itens de segmentação;
- Uma arquitetura com o núcleo através do *X-Road*, embasando assim a nossa proposta com uma tecnologia *open-source* consolidada na entrega de interoperabilidade em governos;
- Através deste apoio e referencial tecnológico, listamos os itens de arquitetura presente no nosso barramento;

- Propomos uma política de acesso e utilização para que entidades governamentais participem do barramento;
- Definimos a semântica e a padronização de dados. Detalhando o tratamento e transformação dos dados;
- Por fim, sugerimos o armazenamento de *hashes* de valores evidenciais em Livro-Razão distribuído.

Em nossos experimentos utilizamos um cenário abstrato de um serviço digital que está alta em evidência, atualmente, no *e-gov* brasileiro, o pagamento de um Auxílio Emergencial. Este implementado a partir do momento de crise causada pela pandemia da covid-19.

Nesta prova de conceito buscamos demonstrar o funcionamento prático do barramento proposto, tomando como norte as regras reais para a concessão do auxílio. Em nossas simulações, através de protótipos funcionais da aplicação e possíveis abstrações de sistemas de informação participantes da interoperabilidade do serviço digital em questão, observou-se um fluxo de dados fluindo em tempo real entre as integrações desses *stakeholders* através do barramento.

Como trabalhos futuros, propõe-se a utilização de mais casos de uso na prova de conceito do barramento, testes de *stress* mais completos, rodando em infraestrutura profissional, por exemplo, através de *cloud computing* e com massa de dados mais aproximada de uma simulação da realidade. Implementar e testar a funcionalidade, descrita na proposta, da política de acesso com utilização de certificação digital ICP-Brasil.

Outro ponto de trabalho futuro seria a análise e implementação de estratégia para tratamento de falhas na comunicação via barramento. Por fim, com o amadurecimento de novos estudos e experimentos, espera-se, na posteridade, uma menor dependência do *X-Road* no ambiente operacional de interoperabilidade.

Bibliografia

[Aguair et al. 2010]AGUAIR, E. et al. Padrões tecnológicos-o uso na prestação de serviços públicos e no relacionamento com o governo federal. *Panorama da Interoperabilidade no Brasil. Brasília: Orgs. MP/SLTI*, 2010.

[ALVARENGA 2017]ALVARENGA, D. *Empresas gastam 1.958 horas e R$ 60 bilhões por ano para vencer burocracia tributária, apontam pesquisas*. 2017. Disponível em: <https://g1.globo.com/economia/noticia/empresas-gastam-1958-horas-e-r-60-bilhoes-por-ano-para-vencer-burocracia-tributaria-apontam-pesquisas.ghtml>.

[Araujo, Reinhard e Cunha 2018]ARAUJO, M. H. d.; REINHARD, N.; CUNHA, M. A. Serviços de governo eletrônico no brasil: uma análise a partir das medidas de acesso e competências de uso da internet. *Revista de Administração Pública*, SciELO Brasil, v. 52, p. 676–694, 2018.

[Auxílio Emergencial 2020]AUXíLIO Emergencial. 2020. Disponível em: <http://www.caixa.gov.br/auxilio/PAGINAS/DEFAULT2.ASPX>.

[Barbaglia, Murzilli e Cudini 2017]BARBAGLIA, G.; MURZILLI, S.; CUDINI, S. Definition of rest web services with json schema. *Software: Practice and Experience*, Wiley Online Library, v. 47, n. 6, p. 907–920, 2017.

[BARBOSA, FARIA e PINTO 2004]BARBOSA, A. F.; FARIA, F. d.; PINTO, S. L. Governo eletrônico: um modelo de referência para a sua implementação. In: *Congresso Anual de Tecnologia de Informação (CATI)*. [S.l.: s.n.], 2004.

[Barros, Cepik e Canabarro 2010]BARROS, A.; CEPIK, M. A. C.; CANABARRO, D. R. Para além da e-ping: o desenvolvimento de uma plataforma de interoperabilidade de e-serviços no brasil. *Panorama da interoperabilidade no Brasil. p. 137-157*, 2010.

[Bashir 2018]BASHIR, I. *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. [S.l.]: Packt Publishing Ltd, 2018.

[Bishr 1997]BISHR, Y. *Semantic Aspect if Interoperable GIS (PhD Thesis)*. [S.l.]: Wageningen Agricultural University and ITC, 1997.

[Brasileiro 2018]BRASILEIRO, G. e-ping padrões de interoperabilidade de governo eletrônico. *Comitê Executivo de Governo Eletrônico, May*, 2018.

[Brasileiro e Eletrônico 2018]BRASILEIRO, G.; ELETRÔNICO, C. E. de G. e-ping padrões de interoperabilidade de governo eletrônico. 2018.

[Brito, Mombach e Valente 2019]BRITO, G.; MOMBACH, T.; VALENTE, M. T. Migrating to graphql: A practical assessment. In: IEEE. *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. [S.l.], 2019. p. 140–150.

[Casino, Dasaklis e Patsakis 2019]CASINO, F.; DASAKLIS, T. K.; PATSAKIS, C. A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telematics and informatics*, Elsevier, v. 36, p. 55–81, 2019.

[CertiSign 2019]CERTISIGN. *Certificado Digital: o que é?* 2019. Disponível em: <https://blog.certisign.com.br/o-que-e-certificado-digital/>.

[Choi e Whinston 2000]CHOI, S.-Y.; WHINSTON, A. B. Benefits and requirements for interoperability in the electronic marketplace. *Technology in society*, Elsevier, v. 22, n. 1, p. 33–44, 2000.

[COSTA 2019]COSTA, B. S. d. E-estônia: digitalização dos serviços públicos da estônia. Universidade Federal de Campina Grande, 2019.

[Cruz et al. 2020]CRUZ, M. et al. Interoperabilidade e integração de sistemas e dados para apoio à tomada de decisão pela gestão da prefeitura de volta redonda-rj: Perspectivas e desafios. In: SBC. *Anais do VIII Workshop de Computação Aplicada em Governo Eletrônico*. [S.l.], 2020. p. 148–155.

[Datta 2019]DATTA, A. Blockchain in the government technology fabric. *arXiv preprint arXiv:1905.08517*, 2019.

[DECRETO Nº 9.278 2018]DECRETO Nº 9.278. 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Decreto/D9278.htm>.

[e-Estonia 2020]E-ESTONIA. 2020. Disponível em: <https://e-estonia.com/>.

[Farinelli e Almeida 2014]FARINELLI, F.; ALMEIDA, M. B. Interoperabilidade semântica em sistemas de informação de saúde por meio de ontologias formais e informais: um estudo da norma openehr. *XVII Encontro Nacional de Pesquisa em Ciência da Informaç ao*, v. 17, n. 1, 2014.

[FERRARA 2018]FERRARA, Y. *Visita ao governo da Estônia: o mais moderno do mundo*. 2018. Disponível em: <https://www.ecommercebrasil.com.br/artigos/visita-ao-governo-da-estonia>.

[Fiarresga et al. 2010]FIARRESGA, V. M. C. et al. *Criptografia e matemática*. Tese (Doutorado), 2010.

[Field et al. 2003]FIELD, T. et al. *OECD e-government studies the e-government imperative*. [S.l.]: OECD Publishing, 2003.

[Fielding e Taylor 2000]FIELDING, R. T.; TAYLOR, R. N. *Architectural styles and the design of network-based software architectures*. [S.l.]: University of California, Irvine Irvine, 2000.

[Framework 2004]FRAMEWORK, I. European interoperability framework for pan-european egovernment services. 2004.

[Gao et al. 2021]GAO, Y. et al. The notarial office in e-government: A blockchain-based solution. *IEEE Access*, IEEE, v. 9, p. 44411–44425, 2021.

[Governo Digital 2020]GOVERNO Digital. 2020. Disponível em: <https://www.gov.br/governodigital/pt-br>.

[Hartig e Pérez 2018]HARTIG, O.; PÉREZ, J. Semantics and complexity of graphql. In: *Proceedings of the 2018 World Wide Web Conference*. [S.l.: s.n.], 2018. p. 1155–1164.

[Hou 2017]HOU, H. The application of blockchain technology in e-government in china. In: IEEE. *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. [S.l.], 2017. p. 1–4.

[Isaja e Soldatos 2018]ISAJA, M.; SOLDATOS, J. Distributed ledger technology for decentralization of manufacturing processes. In: IEEE. *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. [S.l.], 2018. p. 696–701.

[Jaeger e Thompson 2003]JAEGER, P. T.; THOMPSON, K. M. E-government around the world: Lessons, challenges, and future directions. *Government information quarterly*, JAI, v. 20, n. 4, p. 389–394, 2003.

[Joshi e Islam 2018]JOSHI, P. R.; ISLAM, S. E-government maturity model for sustainable e-government services from the perspective of developing countries. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 10, n. 6, p. 1882, 2018.

[Karpersky 2014]KARPERSKY. *Hash: o que são e como funcionam*. 2014. Disponível em: <https://www.kaspersky.com.br/blog/hash-o-que-sao-e-como-funcionam/2773/>.

[Kouicem, Bouabdallah e Lakhlef 2018]KOUICEM, D. E.; BOUABDALLAH, A.; LAKHLEF, H. Internet of things security: A top-down survey. *Computer Networks*, Elsevier, v. 141, p. 199–221, 2018.

[Ludwig, Rebelatto e Silva 2020]LUDWIG, L.; REBELATTO, M. G.; SILVA, S. J. R. da. O estado da arte das criptografias modernas: uma revisão sistemática da literatura. *Revista Brasileira de Computação Aplicada*, v. 12, n. 2, p. 46–53, 2020.

[Lyons, Courcelas e Timsit 2018]LYONS, T.; COURCELAS, L.; TIMSIT, K. *Blockchain and the GDPR [Thematic Report]. The European Union Blockchain Observatory & Forum*. 2018.

[MEIRELLES et al. 2010]MEIRELLES, H. L. et al. Direito administrativo brasileiro, 36"ed. *São Paulo: Malheiros*, 2010.

[Mesquita 2020]MESQUITA, K. A evolução do governo eletrônico no brasil e a contribuição das tic na redefinição das relações entre governo e sociedade. *Comunicologia-Revista de Comunicação da Universidade Católica de Brasília, Brasília*, v. 12, n. 2, p. 174–195, 2020.

[Nakamoto 2008]NAKAMOTO, S. *Bitcoin: A peer-to-peer electronic cash system*. [S.l.], 2008.

[Nielsen 2017]NIELSEN, M. M. egovernance and online service delivery in estonia. In: *Proceedings of the 18th Annual International Conference on Digital Government Research*. [S.l.: s.n.], 2017. p. 300–309.

[Oliveira 2012]OLIVEIRA, R. R. Criptografia simétrica e assimétrica-os principais algoritmos de cifragem. *Segurança Digital [Revista online]*, v. 31, p. 11–15, 2012.

[PAVANATI et al. 2017]PAVANATI, A. et al. Documentos digitais na gestão universitária: O certificado digital como garantia de segurança, origem e integridade. 2017.

[Plansky 2014]PLANSKY, R. *Definição, restrições e benefícios do modelo de arquitetura REST*. 2014. Disponível em: <http://www.rplansky.com/definicao-restricoes-e-beneficios-do-modelo-de-arquitetura-rest/>.

[Purificação 2019]PURIFICAÇÃO, C. S. da. A digitalização dos serviços públicos de atendimento no âmbito do governo federal: Um olhar para as mudanças na forma de prestação desses serviços após a adesão dos instrumentos digitais. 2019.

[RedHat 2021]REDHAT. *GraphQL - O que é e para que serve?* 2021. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-is-graphql>.

[RIBEIRO, RIBEIRO e LEITE 2019]RIBEIRO, I. M. V.; RIBEIRO, M. D. M.; LEITE, M. D. M. G. A transformação do cadastro nacional de informações sociais (cnis) em instrumento de exclusão do acesso à previdência social rural. *Sinapse Múltipla*, v. 8, n. 2, p. 95–99, 2019.

[Ribeiro e Mendizabal 2021]RIBEIRO, L.; MENDIZABAL, O. *Introdução à Blockchain e Contratos Inteligentes*. [S.l.], 2021.

[Santos 2010]SANTOS, E. M. Desenvolvimento e implementação da arquitetura e-ping: estratégias adotadas e possíveis implicações. *Panorama da interoperabilidade no Brasil. Brasília: Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação*, 2010.

[Santos 2011]SANTOS, E. M. dos. A adoção da arquitetura e-ping: Um estudo de caso na fiocruz/bahia. 2011.

[Saputro et al. 2020]SAPUTRO, R. et al. Prerequisites for the adoption of the x-road interoperability and data exchange framework: A comparative study. In: IEEE. *2020 Seventh International Conference on eDemocracy & eGovernment (ICEDEG)*. [S.l.], 2020. p. 216–222.

[Sharma e Jain 2019]SHARMA, K.; JAIN, D. Consensus algorithms in blockchain technology: A survey. In: IEEE. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. [S.l.], 2019. p. 1–7.

[Souza 2018]SOUZA, E. F. *GraphQL — Aprendendo na prática*. 2018. Disponível em: <https://medium.com/trainingcenter/graphql-aprendendo-na-pr%C3%A1tica-569a6866065b>.

[Souza, Curi e Nuintin 2019]SOUZA, P. R. R. de; CURI, M. A.; NUINTIN, A. A. Práticas de governo eletrônico nos municípios: Um estudo da mesorregião do sul e sudoeste do estado de minas gerais. *REUNIR Revista de Administração Contabilidade e Sustentabilidade*, v. 9, n. 1, p. 63–72, 2019.

[Srivastava e Teo 2004]SRIVASTAVA, S. C.; TEO, T. S. A framework for electronic government: evolution, enablers and resource drainers. In: *Proceedings of the Eighth Pacific Asia Conference on Information Systems*. [S.l.: s.n.], 2004.

[Surwase 2016]SURWASE, V. Rest api modeling languages-a developer's perspective. *Int. J. Sci. Technol. Eng*, v. 2, n. 10, p. 634–637, 2016.

[Taelman, Sande e Verborgh 2018]TAELMAN, R.; SANDE, M. V.; VERBORGH, R. Graphql-ld: linked data querying with graphql. In: *ISWC2018, the 17th International Semantic Web Conference*. [S.l.: s.n.], 2018. p. 1–4.

[Tapscott e Tapscott 2016]TAPSCOTT, D.; TAPSCOTT, A. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. [S.l.]: Penguin, 2016.

[Tasatanattakool e Techapanupreeda 2018]TASATANATTAKOOL, P.; TECHAPANUPREEDA, C. Blockchain: Challenges and applications. In: IEEE. *2018 International Conference on Information Networking (ICOIN)*. [S.l.], 2018. p. 473–475.

[Wegner 1996]WEGNER, P. Interoperability. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 28, n. 1, p. 285–287, 1996.

[X-Road Architecture 2020]X-ROAD Architecture. 2020. Disponível em: <https://bitbucket.niis.org/projects/X-ROAD/repos/x-road/browse/doc/Architecture/arc-g_x-road_arhitecture.md/>.

[Zhu e Zhou 2016]ZHU, H.; ZHOU, Z. Z. Analysis and outlook of applications of blockchain technology to equity crowdfunding in china. *Financial innovation*, SpringerOpen, v. 2, n. 1, p. 1–11, 2016.