Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Assinatura Digital no Padrão XAdES Como um Serviço REST: Um Estudo de Caso na Universidade de Brasília

Renato Carauta Ribeiro

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientadora
Prof.a Dr.a Edna Dias Canedo

Brasília
2021

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Assinatura Digital no Padrão XAdES Como um Serviço REST: Um Estudo de Caso na Universidade de Brasília

Renato Carauta Ribeiro

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof.a Dr.a Edna Dias Canedo (Orientadora)
CIC/UnB

Prof. Dr. Eduardo Luzeiro Feitosa
IC/UFAM

Prof. Dr. Rafael Timóteo de Sousa Junior
FT/UnB/PPEE

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 25 de junho de 2021

Dedicatória

Dedico este trabalho aos meus pais Marie Elize e Geraldo Magela, pelo incentivo e compreensão neste trajeto. Sem eles não teria superado os obstáculos para o fim desta jornada.

Agradecimentos

À minha orientadora, Professora Dr.a Edna Dias Canedo, agradeço todo auxílio e as orientações que me ajudaram a concluir este estudo.

Aos professores do Mestrado de Computação Aplicada da Universidade de Brasília, pela competência ao transmitir os conteúdos tão relevantes que agregaram conhecimento e ajudaram a enriquecer este trabalho.

Aos colegas da STI/UnB que sempre se dispuseram a colaborar no que fosse necessário para conclusão desta pesquisa.

Agradeço também o apoio técnico e computacional do Laboratório de Tecnologias para Tomada de Decisão - LATITUDE, da Universidade de Brasília, que conta com apoio do CNPq - Conselho Nacional de Pesquisa (Outorgas 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN e 465741/2014-2 INCT em Cibersegurança), da CAPES - Coordenação de Aperfeiçoamento do Pessoal de Nível Superior (Outorgas PROAP PPGEE/UnB, 23038.007604/2014-69 FORTE, e 88887.144009/2017-00 PROBRAL), da FAP-DF - Fundação de Amparo à Pesquisa do Distrito Federal (Outorgas 0193.001366/2016 UIoT e 0193.001365/2016 SSDDC), do Ministério da Economia (Outorgas 005/2016 DIPLA e 083/2016 ENAP), da Secretaria de Segurança Institucional da Presidência da República do Brasil (Outorga ABIN 002/2017), do Conselho Administrativo de Defesa Econômica (Outorga CADE 08700.000047/2019-14), da Advocacia Geral da União (Outorga AGU 697.935/2019),do Departamento Nacional de Auditoria do SUS (Outorga 23106.118410/2020-85), da Procuradoria Geral da Fazenda Nacional (Outorga 23106.148934/2019-67) e dos Decanatos de Pesquisa e Inovação e de Pós-Graduação da Universidade de Brasília (Outorga 23106.067186/2021-37).

E, por fim, à minha mãe Marie Elize Carauta Couto que realizou a correção textual deste estudo.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A assinatura digital de documentos e diplomas é um tema amplamente discutido na Administração Pública Federal. Diversas leis e portarias foram criadas para normatizar a emissão, validação e validade jurídica dos documentos assinados digitalmente em território nacional, como por exemplo as portarias criadas pelo Ministério da Educação (MEC) para normatizar a emissão de diplomas em formato digital. Essas portarias criaram diretrizes e padrões que devem ser adotados pelas Universidades Federais para a assinatura de diplomas em formato digital. O objetivo principal deste trabalho é estudar essas portarias, as principais tecnologias e padrões de assinatura digital utilizadas na literatura para criar um modelo de sistema de assinatura digital para a Universidade de Brasília- UnB, que esteja em conformidade com os padrões do MEC e da ICP-Brasil. Além disso, o modelo deve ser desenvolvido com os principais padrões e tecnologias do mercado, ser aderente à atual arquitetura da UnB, ser de fácil manutenção e atualizações para novos padrões que venham a surgir e, também, ser um projeto de código totalmente aberto. O método utilizado para desenvolver o estudo de caso foi o proposto por Yin[1]. Foi desenvolvido dois protótipos, um utilizando a linguagem Erlang dentro do barramento de serviços utilizado pela UnB e outro utilizando a linguagem Java, com a biblioteca XAdES4j, como um microsserviço intermediado pelo barramento utilizado pela UnB. Os dois protótipos desenvolvidos foram comparados com o atual sistema de assinatura digital utilizado pela UnB, o C3Web. Os testes comparativos entre cada uma das três soluções demonstrou que o atual sistema da UnB não faz a assinatura em conformidade com o atual padrão de assinatura proposto pelo MEC, além de ser um sistema fechado e com uso de tecnologias proprietárias para a execução de assinatura digital. Os testes com o modelo proposto demonstraram que o modelo realiza a assinatura digital em conformidade com o padrão XAdES-T, normativos do MEC e da ICP-Brasil. Além disso, a solução apresentou uma performance e segurança equiparada ao atual sistema utilizado pela UnB. A solução proposta utiliza tecnologias e bibliotecas open-source, sendo de fácil integração, atualização e manutenções futuras, permite uma redução de custos em relação à emissão de diplomas em formato físico e possibilita uma integração do modelo proposto à qualquer arquitetura utilizada por outras Universidades Federais de maneira rápida e menos onerosa.

Palavras-chave: Assinatura Digital, XAdES, REST, RSA, ICP-Brasil

Abstract

The digital signature of documents and diplomas is a topic widely discussed in the Federal Public Administration. They have created several laws and ordinances to standardize the issuance, validation, and legal validity of documents digitally signed in the national territory, such as, for example, the regulations created by the Ministry of Education (MEC) to standardize the issuance of diplomas in digital format. These ordinances created guidelines and standards that Federal Universities must adopt to sign certificates in digital format. The main objective of this work is to study these ordinances, the leading technologies, and digital signature standards used in the literature to create a model of the digital signature system for the University of Brasilia - UnB, which conforms with the standards of MEC and ICP -Brazil. In addition, the model must be developed with the primary criteria and technologies in the market, adhere to the current architecture of UnB, be easy to maintain and update to new standards that may arise, and be a fully open source project. The method used to develop the case study was proposed by Yin [1]. Two prototypes were developed, one using the Erlang language within the service bus used by UnB and the other using the Java language, with the XAdES4j library, as a microservice intermediated by the bus used by UnB. The two prototypes developed were compared with the current digital signature system used by UnB, the C3Web. The comparative tests between each of the three solutions showed that the current UnB system does not subscribe following the current subscription standard proposed by MEC, in addition to being a closed system and using proprietary technologies for the execution of digital signatures. Tests with the proposed model demonstrated that the model performs the digital signature following the XAdES-T standard, normative by MEC and ICP-Brasil. In addition, the solution presented performance and security comparable to the current system used by UnB. The proposed solution uses open-source technologies and libraries, being easy to integrate, update and maintain in the future, allow a cost reduction with the issuance of diplomas in physical format, and integrate the proposed model with any architecture used by other Federal Universities quickly and less costly.

Keywords: Digital Signature, XAdES, REST, RSA, ICP-Brasil

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A segurança da informação é um tema atualmente muito discutido por diversos países os quais tem criado leis cada vez mais rigorosas para proteção e segurança dos dados e informações dos usuários. De maneira geral, a segurança é a capacidade de se proteger contra alguma ameaça ou dano intencional ou não. [14]. A segurança da informação tem por objetivo proteger a confidencialidade, integridade e disponibilidade de informações, seja no armazenamento, processamento ou transmissão. Para se conseguir uma boa segurança é necessária uma política de segurança, educação, treinamento e conscientização das partes envolvidas bem como o uso de tecnologias [14].

Diversos países têm se preocupado com a segurança da informação e com as consequências de possíveis vazamentos de dados. Por esse motivo diversas leis foram aprovadas com intuito de normatizar a proteção das informações. Atualmente, a lei internacional da União Europeia que trata da proteção e segurança dos dados pessoais é a *General Data Protection Regulation* (GDPR) [15]. GDPR é um regulamento do Parlamento Europeu e Conselho da União Europeia que estabelece regras sobre a privacidade e proteção de dados de cidadãos da União Europeia e Espaço Econômico Europeu. De acordo com o regulamento citado, a partir de uma política de transparência e notificação de privacidade, o próprio usuário deve ter controle de quais dados serão utilizados pelas empresas que coletam dados de pessoas.

O Brasil aprovou em 14 de agosto de 2018, a Lei Geral de Proteção de Dados Pessoais (LGPD) [16]. Essa lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. É necessário que os dados pessoais, ou seja, a informação digital, esteja segura e para isso é importante adotar medidas de segurança técnicas e administrativas para proteção dessa informação [16]. A LGPD determina o que pode ou não ser feito em relação à coleta e tratamento de dados no Brasil, prevendo punições para as empresas que desrespeitarem os seus princípios [16].

As aplicações devem estar seguras, ou seja, as informações só podem ser disponibilizadas para quem tem permissão para acessá-las. Para garantir a segurança das aplicações é necessário o uso de autenticação e de autorização. A autenticação é o processo de verificar uma identidade alegada por uma entidade do sistema. Para isso são necessários dois passos [3]:

- **Identificação:** apresentar um identificador ao sistema de segurança para confirmação da identidade.
- **Verificação:** apresentar ou gerar informações de identificação que corroborem o vínculo entre a entidade e o identificador.

A autorização é o processo pelo qual se determina se um usuário autenticado tem permissão para executar uma operação ou acessar alguma informação específica e/ou restrita [3] [17]. Além das aplicações é necessário que as redes sejam seguras para que a mensagem não seja interceptada e/ou alterada durante o trajeto entre o emissor e o receptor. A segurança de rede se baseia em criptografia para transformar as mensagens e torná-las seguras e imunes a ataques. A criptografia pode fornecer diversos aspectos de segurança tais como a confidencialidade, integridade, autenticação e o não-repúdio no envio de informações pela rede [2].

Existem dois tipos de criptografia: simétricas e assimétricas. A criptografia simétrica consiste em transformar um texto claro em um texto cifrado usando apenas uma chave, que serve tanto para transformar um texto claro em cifrado quanto para voltar o texto cifrado para texto claro novamente [18]. Para que a criptografia simétrica seja segura é necessário que seja impraticável decifrar uma mensagem com base no texto cifrado e com o conhecimento do algoritmo de cifração/decifração [3]. Um dos principais problemas da criptografia simétrica é a distribuição da chave, pois para cifrar e decifrar o texto é necessário que o emissor e o receptor tenham conhecimento da chave. O compartilhamento dessa chave por um canal de comunicação inseguro é um dos grandes problemas desse tipo de criptografia [3].

Criada em 1976 por Whitfield Diffie e Martin Hellman, a criptografia assimétrica também chamada de criptografia de chave pública, consiste na utilização de duas chaves distintas, uma pública e a outra privada. A chave pública pode ser divulgada em ambientes não seguros, sem comprometer a decifração da chave privada. Quando um texto é cifrado por uma das chaves, apenas a outra chave é capaz de decifrá-lo. Isso garante a autenticidade das mensagens cifradas [19].

Uma das maiores utilizações da criptografia de chave pública é o uso de assinatura digital. Essa é uma forma de um usuário assinar uma mensagem ou documento com sua chave privada. Outro usuário poderá verificar a veracidade da assinatura utilizando a

chave pública do usuário que assinou a mensagem [3]. A vantagem dessa abordagem é que, uma vez assinado o documento ou mensagem com a chave privada de um usuário, este não pode negar ter assinado o documento ou enviado a mensagem, o que garante uma segurança forte e o princípio do não repúdio [20].

Os certificados digitais foram utilizados para garantir a autenticidade das chaves públicas, ou seja, para assegurar que uma determinada chave pública de fato pertença apenas a uma pessoa ou organização associada a ela. Para essa garantia é necessário um terceiro confiável responsável por verificar a alegação de cada proprietário que gera uma chave privada, chamado de autoridade certificadora (CA). Assim, a forma de garantir a autenticidade se dá com a assinatura da CA na chave privada do proprietário. [21].

1.1 Problema de Pesquisa

A segurança da informação é um tema amplamente discutido e analisado pelo governo brasileiro. A principal lei que trata desse assunto é a Lei Geral de Proteção de Dados (LGPD) [16]. No âmbito das Universidades Federais, o Ministério da Educação (MEC) criou a portaria nº 554 com o objetivo de normatizar a emissão dos diplomas em formato digital. Nessa portaria, são apresentados os formatos dos documentos, bem como o padrão de assinatura que deve ser utilizado pelas Instituições de Ensino Superior (IES) [22].

A Universidade de Brasília (UnB) tem por objetivo a modernização da emissão de diplomas digitais de acordo com o que é normatizado pela portaria do MEC. Em 2018 a UnB teve um gasto de R$ 11 mil com emissão de diplomas para pós-graduação e, no mesmo ano, passou a disponibilizar um sistema para emissão de diplomas digitais, o C3Web [23], [24]. O contrato para manutenção é de R$ 200 mil reais anuais. Esse sistema possui algumas desvantagens como não fazer assinatura de diplomas que não estejam previamente no banco de dados. Além disso, por ser um software proprietário, é cobrada da UnB uma anuidade para sua utilização. Outra desvantagem consiste em não disponibilizar o código fonte para a evolução do sistema, correção de possíveis bugs e adequação desse software a portaria do MEC. Assim, faz-se necessário que a UnB desenvolva um sistema próprio para realizar a assinatura de documentos em formato digital.

Os principais problemas encontrados na solução utilizada pela UnB são a não adequação aos padrões propostos pelo MEC e a dificuldade para manutenção e evoluções futuras devido ao uso de bibliotecas proprietárias. Os diplomas assinados ficam no banco de dados fora do modelo de dados da UnB, o que faz com que sejam replicados os dados necessários para apresentação dos diplomas salvos. Outro problema é que os diplomas só podem ser assinados se anteriormente forem armazenados no banco de dados do sistema, o que impede a assinatura de diplomas externos. O sistema é fechado e não possui

serviços para comunicação entre outras aplicações. O Software Development Kit (SDK) desktop para leitura dos certificados digitais não se integra com nenhuma base de dados, tornando-se necessária a leitura do certificado a cada vez em que for assinado um lote do sistema. Diversos sistemas de assinatura digital utilizam o cadastro dos certificados apenas uma vez para assinar os documentos. O SDK foi desenvolvido em Java, versão 8 e atualmente o sistema está sendo modernizado para uma versão utilizando o framework Spring Boot [25].

Para entender e responder à necessidade da UnB, neste trabalho foram definidas as seguintes questões de pesquisa:

QP.1. O que evidencia a integridade, validade e segurança em um documento em formato digital de acordo com a literatura existente?

QP.2 Quais as principais técnicas, modelos e padrões que têm sido sugeridos na literatura para assinar um documento em formato digital de maneira segura?

1.2 Justificativa

O Ministério da Educação (MEC) instituiu que os diplomas das Instituições de Ensino Superior (IES) devem ser em formato digital com uso de certificado digital e no padrão XAdES. O portal do MEC estipulou diversos prazos para a implantação do diploma em formato digital. O processo teve inicio em 2018 e sua finalização está prevista para 2022 [26]. A partir de 2022, conforme o normativo do MEC, não serão mais impressos diplomas em formato físico, mas somente gerados em formato digital. Apenas em ocasiões especiais e com justificativa plausível será impresso o documento em formato físico. O documento assinado digitalmente deve ser válido em todo o território nacional e deve ser assinado por uma autoridade certificadora subsidiária da ICP-Brasil com certificado A3 ou superior [27].

O MEC disponibilizou diversos normativos e padrões que devem ser seguidos pelas IES para implementação do diploma digital. Por esse motivo surgiu a necessidade de a UnB criar um sistema de assinatura digital até no máximo a data limite estipulada. O sistema atual C3Web não obedece ao padrão determinado pelo MEC, e por ser um sistema proprietário de código fechado torna-se inviável para a própria UnB adaptá-lo ao padrão imposto. Assim, a UnB necessita de um sistema rápido, escalável e que atenda a necessidade de assinatura de documentos em lote, além da possibilidade de se verificar sua validade e autenticidade.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é a criação de um modelo de assinatura digital em conformidade com as leis e portarias do Governo Federal para a emissão dos diplomas da UnB.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

- Realizar uma revisão de literatura sobre as soluções disponíveis para assinatura digital;
- Analisar as principais leis brasileiras e internacionais sobre segurança da informação;
- Selecionar as melhores tecnologias e metodologias utilizadas em sistemas de assinatura digital;
- Apresentar as soluções tecnológicas para assinatura digital aderentes a legislação brasileira e aos processos e tecnologias da UnB;
- Propor um modelo de assinatura digital relevante para a UnB que possa ser mantido e evoluído pela equipe de desenvolvimento de software da UnB;
- Realizar um estudo comparativo com o sistema legado;
- Realizar uma análise de desempenho e segurança da solução proposta.

1.4 Resultados Esperados

Com o desenvolvimento do sistema de assinatura digital espera-se os seguintes resultados:

- Redução dos custos de licença de software;
- Solução aderente as leis e portarias do Governo Federal;
- Solução com maior facilidade de manutenção e evoluções futuras;
- Modelo arquitetural simplificado e aderente aos novos padrões;
- Maior escalabilidade na assinatura de documentos no formato digital.

1.5 Metodologia de Pesquisa

Este trabalho utilizou como metodologia de pesquisa uma revisão da literatura em conjunto com um estudo de caso para a validação da solução por meio de um experimento controlado. A revisão da literatura é utilizada como uma forma de entendimento de diferentes metodologias e soluções para o desenvolvimento da solução proposta permitindo identificar problemas reais que puderam ser resolvidos neste trabalho [1].

Um experimento controlado é um método científico para identificar a relação de causa-efeito. Um dos objetivos dessa metodologia é a comparação de diferentes ferramentas utilizadas para resolução de um problema. O experimento deve ser conduzido de modo que fatores que possam influenciar o resultado devam ser mapeados e controlados para que não se tenha resultados inesperados [28]. O estudo de caso deste trabalho, utilizando um experimento controlado, será realizado com base no modelo proposto por Yin [1], o qual consiste em 6 passos que são apresentados na Figura 1.1:

1. **Planejar o Estudo de Caso:** Esta etapa tem o objetivo de decidir sobre o método a ser utilizado na escolha e na condução do estudo a ser realizado. Devem ser considerados todos os métodos de maneira inclusiva e pluralista para a escolha daquele a ser adotado.

2. **Criar o Design do Estudo de Caso:** Nesta fase devem ser definidos a lógica do estudo; o(s) caso(s) a ser(em) estudado(s), o desenvolvimento da teoria, as proposições e as questões relacionadas para orientar o estudo de caso. Deve ser especificado também qual desenho será utilizado para o caso de uso (casos únicos ou múltiplos, holísticos ou incorporados) e testar o projeto com base em critérios pré-definidos para manter a qualidade do estudo de caso.

3. **Preparar o Estudo de Caso:** Nesta etapa devem ser aprimoradas as habilidades para realizar o estudo de caso, desenvolver o protocolo do estudo de caso, além da realização do treinamento no estudo de caso específico.

4. **Coletar as Evidências do Estudo de Caso:** Esta etapa tem como objetivo a coleta dos dados para o desenvolvimento do problema de pesquisa definido nas etapas anteriores. Poderão ser utilizados experimentos e testes para o levantamento dos dados necessários ao posterior desenvolvimento da pesquisa ou a utilização de perguntas abertas para um melhor entendimento do estudo de caso. Ao reunir os dados é necessária a interpretação das informações que estão sendo coletadas e sua real importância para a pesquisa a ser realizada.

5. **Analisar o Estudo de Caso:** Nesta fase os dados devem ser analisados com o objetivo de descoberta de padrões, ideias ou conceitos promissores. A estratégia a

ser conduzida com a análise dos dados deve seguir um ciclo que envolva as questões de pesquisa, os dados, seu manuseio, a interpretação dos dados e a capacidade de declarar alguma descoberta ou extrair alguma conclusão. Podem ser adotadas quatro estratégias de análise de dados: basear-se em proposições teóricas; trabalhar com os dados desde o início do estudo; desenvolver uma descrição do caso e examinar explicações contrárias.

6. **Compartilhar o Estudo de Caso:** Nesta etapa deve ser relatado o que de relevante o estudo deixou para a comunidade científica. Para isso, podem ser desenvolvidos materiais textuais/visuais; exibição de evidências para o leitor chegar a uma conclusão; revisão e recomposição até que a pesquisa esteja adequada e bem estruturada.

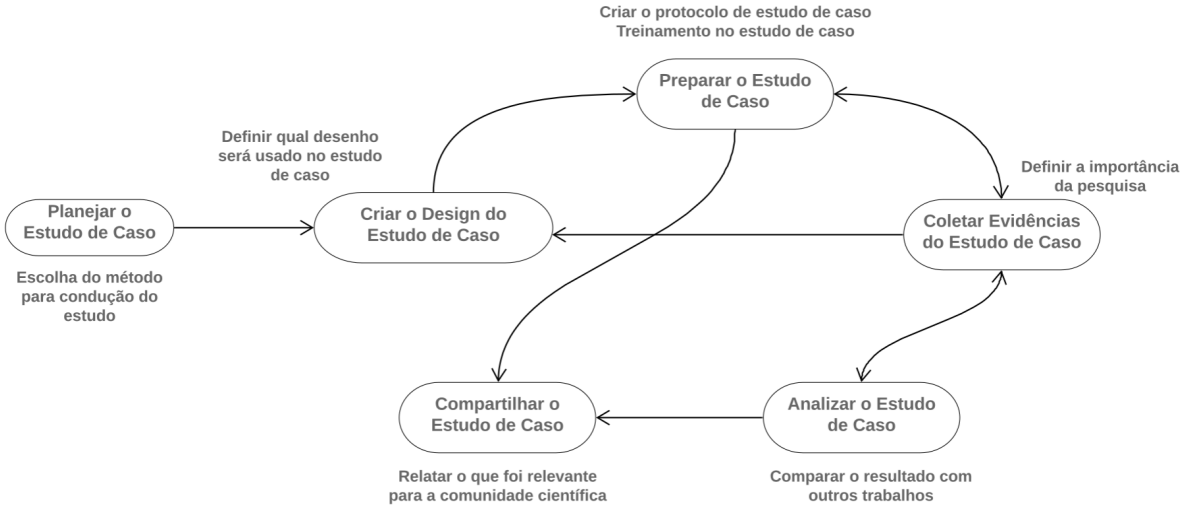


Figura 1.1: Modelo Proposto [1]

Para o desenvolvimento do estudo de caso com um experimento controlado, será utilizado um ambiente controlado com os dados retirados do ambiente de homologação da UnB. Serão realizados testes de desempenho e de eficácia da nova solução proposta para a assinatura digital dos documentos em relação a solução adotada pela UnB.

1.6 Estrutura do Trabalho

Este trabalho está organizado em 7 capítulos, além deste, consistindo em:

- **Capítulo 2:** Apresenta os conceitos fundamentais necessários para entendimento deste estudo.

- **Capítulo 3:** Descreve a metodologia de pesquisa adotada e os resultados encontrados com a revisão de literatura.
- **Capítulo 4:** Apresenta a legislação e documentos governamentais necessários para o modelo de assinatura de diplomas em formato digital. Também é apresentado o modelo base para esse sistema.
- **Capítulo 5:** Apresenta os requisitos funcionais e não funcionais e expõe um modelo arquitetural minimalista para a solução de assinatura digital que se integre à atual arquitetura da UnB.
- **Capítulo 6:** Descreve os testes comparativos entre o atual sistema C3Web e o sistema desenvolvido neste trabalho. São apresentadas as políticas de segurança de controle interno e externo para os diplomas assinados, bem como um comparativo entre a legislação brasileira e a legislação europeia, de maneira a verificar possíveis falhas e tecnologias atualizadas mencionadas pelo governo brasileiro.
- **Capítulo 7:** Realiza uma análise geral do trabalho com a apresentação dos resultados da pesquisa e descreve possíveis trabalhos futuros a serem desenvolvidos.

Capítulo 2

Fundamentação Teórica

A assinatura realizada por meio digital é um mecanismo de assinatura que tem a mesma segurança e validade de uma assinatura em formato manuscrito. O objetivo da assinatura digital é assinar documentos em formato eletrônico, adicionando um selo que garanta a origem e a integridade dos dados [29]. Para garantir a segurança da assinatura digital os dados assinados são adicionados ao documento de maneira cifrada para que a validade e a integridade do documento possam ser comprovados de maneira confiável e livre de falsificações [29]. Na União Europeia (UE) foi criado o regulamento número 910/2014, relativo aos serviços de identificação eletrônica e de confiança para transações eletrônicas dentro da UE. Esse regulamento define as regras de como deve ser uma assinatura eletrônica e sua validade [30].

A *European Telecommunications Standards Institute (ETSI)* e a *World Wide Web Consortium* (W3C) são as entidades que mantém os padrões atuais de assinatura digital: *Extensible Markup Language* (XML) *Advanced Electronic Signatures* (XAdES) e PDF *Advanced Electronic Signatures* (PAdES), que atualmente são os dois padrões mais utilizados para assinatura de documentos em formato digital [29].

No Brasil a entidade que é a Autoridade Certificadora Raiz (AC-Raiz) é a Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil). A ICP-Brasil é a entidade de confiança que viabiliza a emissão de certificados digitais para identificação virtual do cidadão, além de credenciar e descredenciar as demais autoridades certificadoras no âmbito nacional [31].

Existem dois tipos de criptografia, a criptografia simétrica e a criptografia assimétrica, também chamada de criptografia de chave pública. Na criptografia de chaves simétricas a mesma chave é utilizada para cifrar e decifrar uma mensagem. Na criptografia de chave pública ou assimétrica, exitem duas chaves: uma pública e outra privada. Mensagens cifradas com uma das chaves só podem ser decifradas com a outra chave [2].

Para utilização da assinatura digital é usada a criptografia de chave pública para garantir a segurança da transmissão da informação e a autenticidade dos dados transmitidos [3]. Dois dos algoritmos utilizados são o *Rivest-Shamir-Adleman* (RSA), baseado na fatoração de números inteiros, e o *Digital Signature Algorithm* (DSA), baseado em logaritmos discretos [32].

2.1 Tipos de Criptografia

Os algoritmos de criptografia podem ser divididos em dois grupos: os algoritmos de criptografia simétrica, também chamados de chave secreta e algoritmos de criptografia assimétrica ou de chave pública [2]. Os algoritmos de encriptação são baseados em dois princípios: substituição, em que cada elemento no texto claro (bit, letra, grupo de bits ou letras) é mapeado em outro elemento e transposição, em que os elementos do texto claro são rearranjados. Ao fazer estas duas operações nenhuma informação pode ser perdida [3]. Para que um algoritmo de criptografia seja considerado computacionalmente seguro ele deve seguir dois critérios: o custo para quebrar a cifra ultrapassa o valor da informação encriptada e o tempo exigido para quebrar a cifra deve superar o tempo de vida útil da informação [3].

Até a década de 1970 o único tipo de criptografia existente era a criptografia simétrica e até hoje é o tipo de criptografia mais utilizado [3]. Atualmente, na criptografia simétrica os dois algoritmos mais utilizados são o *Data Encryption Standard* (DES) e o *Advanced Encryption Standard* (AES) [33], [3]. Na criptografia assimétrica os algoritmos mais utilizados são o RSA e o DSA [3]. Exite ainda um tipo mais recente de criptografia assimétrica que foi desenvolvido chamado de *Elliptic Curve Cryptography* (ECC) [34].

2.1.1 Criptografia Simétrica

A criptografia simétrica é o modelo de criptografia mais antigo e ainda hoje é o tipo mais utilizado. O modelo básico de encriptação simétrica possui 5 itens básicos [3]:

- **Texto claro:** Mensagem ou dado original, em formato legível, que serve como entrada do algoritmo de encriptação.
- **Algoritmo de encriptação:** Realiza diversas substituições e transformações no texto claro.
- **Chave secreta:** Código utilizado para que o algoritmo efetue as substituições e transformações no texto claro resultando em um texto cifrado.

- **Texto cifrado:** Mensagem embaralhada, produzida pelo algoritmo de cifração. O texto cifrado é o conjunto do texto claro mais a chave secreta.
- **Algoritmo de decifração:** Basicamente o algoritmo de cifração executado de modo inverso utilizando o texto cifrado e a chave secreta. A saída do algoritmo é o texto claro original.

Para que um algoritmo de cifração seja considerado forte, deve ser inviável decifrar uma mensagem apenas com o texto cifrado e o conhecimento do algoritmo, ou seja, o algoritmo é tão forte quanto a força da chave utilizada para cifração do texto claro. A chave deve ser compartilhada por meio de um canal seguro [3]. A Figura 2.1 apresenta, de maneira simplificada, o fluxo para cifrar um texto com algoritmo simétrico [2]:

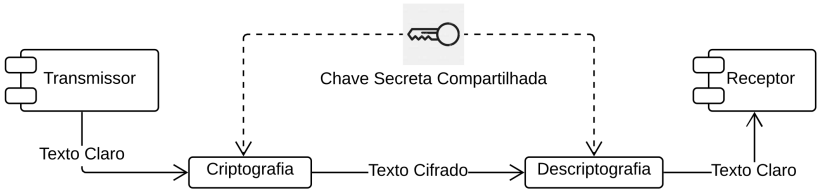


Figura 2.1: Criptografia de chave Simétrica [2]

Os algoritmos podem cifrar um fluxo de dados (um bit ou byte por vez), o que é chamado cifragem de fluxo, ou podem cifrar blocos de bits de uma só vez. Normalmente são usados 64 ou 128 bits por vez a serem cifrados. As principais aplicações da criptografia simétrica são baseadas na utilização das cifras de bloco [3].

2.1.2 Criptografia Assimétrica

A criptografia de chave pública é baseada em funções matemáticas ao invés de substituição e permutação e, diferentemente da criptografia simétrica, a criptografia de chave pública utiliza duas chaves separadas [3]. Pelo uso de duas chaves de criptografias distintas, uma pública e outra privada, foi resolvido o problema encontrado na criptografia simétrica que é o compartilhamento da chave criptográfica entre o emissor e o receptor. Uma das chaves serve para cifração e a outra serve para decifração. As características básicas de um sistema de chave pública são:

- É computacionalmente inviável determinar a chave de decifração dado apenas o conhecimento do algoritmo de criptografia e a chave de encriptação.
- Qualquer uma das duas chaves relacionadas pode ser usada para cifrar e a outra para decifrar.

Na criptografia assimétrica a chave privada deve ser guardada de maneira segura pelo emissor e nunca deve ser compartilhada. A chave pública é uma chave gerada pelo emissor, porém pode ser compartilhada sem comprometer a segurança da comunicação [2], [3].

A Figura 2.2 apresenta o processo utilizado na criptografia assimétrica. Se a chave pública é usada para cifrar a mensagem, apenas a chave privada pode ser usada para decifrá-la. Se uma mensagem for cifrada com a chave privada, apenas a chave pública poderá decifrá-la, ou seja, a mesma chave nunca é usada para cifrar e decifrar uma mensagem [2].

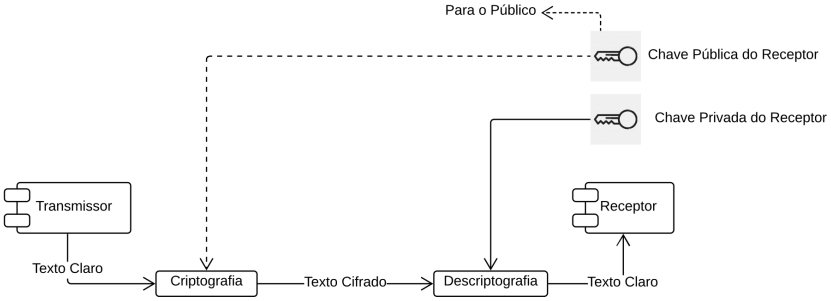


Figura 2.2: Criptografia Assimétrica [2]

Um dos algoritmos mais comuns utilizados é o RSA. Esse algoritmo foi proposto por Rivest et al. [35]. É um algoritmo de chave pública utilizado para gerar sistema de assinatura assimétrica seguro [36]. Para criação das chaves públicas e privadas esse algoritmo utiliza dois números primos muito grandes, p e q [2], e devem obedecer às seguintes etapas conforme apresentado na Figura 2.3:

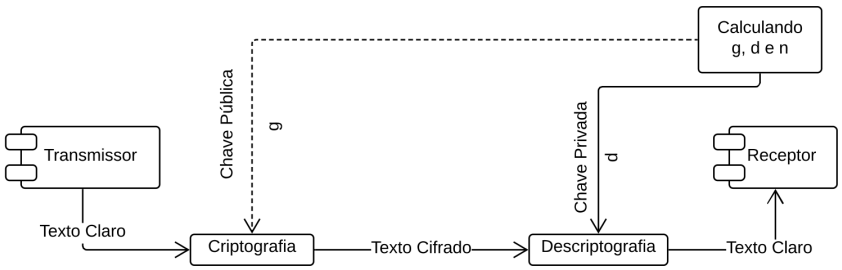


Figura 2.3: RSA [2]

- Multiplicam-se os dois números primos escolhidos para descobrir n, que é o módulo para cifrar e decifrar, $n = p\ x\ q$.
- Calcula-se outro número z = (p - 1) x (q - 1).
- Escolhe-se um inteiro aleatório g. Em seguida, calcula-se d de modo que d é = mod z.

- Anuncia-se g e n para o público, mantendo-se z e d secretos.

Para se cifrar uma mensagem utilizando o algoritmo RSA é necessário calcular o texto claro com os valores g e n para gerar o texto cifrado a ser enviado. A fórmula utilizada para cifrar um texto claro é a seguinte [2]:

$$C = P^g (mod\ n)$$

Para decifrar uma mensagem é necessário que os valores de z e d sejam privados e ao receber um texto cifrado é utilizada a chave privada d para decifrar a mensagem [2]:

$$P = C^d (mod\ n)$$

2.2 Funções Hash

As funções hash têm por objetivo manter a integridade dos dados. Uma pequena mudança em qualquer bit de uma mensagem resulta, com alta probabilidade, em uma mudança no código do hash [3]. As funções hash são algoritmos que transformam um texto em um hash pré-especificado que nunca se repete. Deve ser inviável descobrir o texto original apenas com o resultado da função hash [3], [37]. Conforme apresentado na Figura 2.4, a mensagem original hash é um bloco de dados de tamanho variável M e a saída é sempre um bloco de dados de tamanho fixo h, o que garante maior agilidade para geração do hash [3].

$$h = H(M)$$

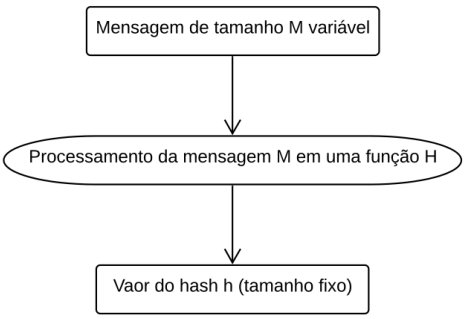


Figura 2.4: Função hash [3]

2.2.1 Aplicações de Funções Hash

Uma das principais aplicações das funções hash está na autenticação de mensagens, por ser um mecanismo que serve para verificar a integridade da mensagem. Ela garante que os dados recebidos sejam exatamente os mesmo que foram enviados. A autenticação é utilizada da seguinte maneira: o emissor calcula um valor hash com a função dos bits da mensagem e transmite esse valor. O receptor realiza o mesmo cálculo de hash sobre a mensagem recebida e compara o valor do hash calculado com o valor do hash recebido na mensagem, se houver divergência, o receptor sabe que houve alteração na mensagem recebida [3].

Outra aplicação importante das funções hash consiste nas assinaturas digitais. O valor hash da mensagem é encriptado com a chave privada do usuário. Qualquer um que tiver a chave pública desse usuário pode verificar a integridade da mensagem associada à assinatura digital conforme demonstrado na imagem 2.5. Com isso o invasor que quiser alterar a mensagem precisará conhecer a chave privada do usuário [3].

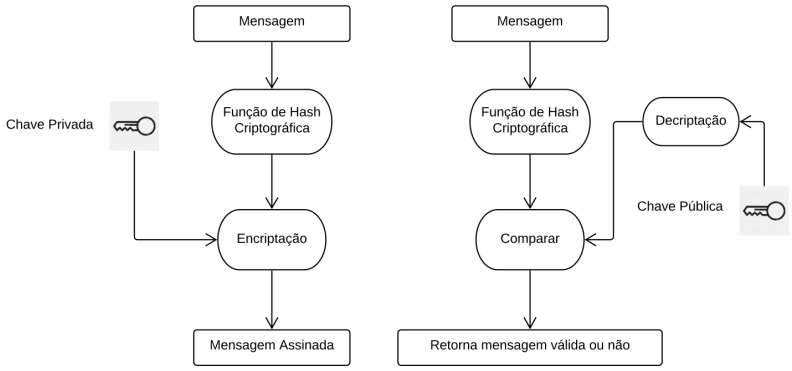


Figura 2.5: Processo de Assinatura [3]

Outra utilização muito comum das funções hash é a criação de senhas de mão única, desse modo a senha real não pode ser recuperada pelo hacker que conseguir acessar a senha armazenada no formato hash. Para verificar se as senhas são iguais para autenticação do usuário, a senha digitada é convertida para a função hash e comparada com a função hash já armazenada, se forem iguais o acesso é permitido, se forem diferentes o acesso é negado [3]. É importante uma visão geral sobre a matemática para cifrar e decifrar uma assinatura digital com intuito de demonstrar a segurança que é envolvida nesse processo com o RSA e a criação das funções hash.

2.3 Aritmética Modular

A congruência estabelece uma relação de equivalência entre certos números. Por exemplo, um conjunto de números que se divididos por 7 darão os resultados do resto da divisão entre 0 e 6. Com isso pode-se representar todos os números inteiros sendo congruente a algum número entre 0 e 6 [38].

$$8 = 1 \ (mod\ 7)$$
$$9 = 2 \ (mod\ 7)$$
$$10 = 3 \ (mod\ 7)$$
$$11 = 4 \ (mod\ 7)$$
$$12 = 5 \ (mod\ 7)$$
$$13 = 6 \ (mod\ 7)$$
$$14 = 0 \ (mod\ 7)$$

A aritmética modular ou relação de congruência é definida da seguinte forma: sendo a,b,n inteiros com n <> 0, é representado como a = b (mod n). Essa fórmula significa que se a diferença entre a - b é um inteiro múltiplo de n então o número é chamado de módulo da congruência [38]. Existem quatro proposições básicas na aritmética modular. Assumindo que a,b,c,n são inteiros e que n <> 0 [38]:

1. a = 0 (mod n) se e apenas se n/a.
2. a = a (mod n).
3. a = b (mod n) se e apenas se b = a (mod n).
4. Se a = b e b = c (mod n) então a = c (mod n).

2.4 Assinatura Digital

A assinatura digital utiliza como base a criptografia de chave pública. O sistema de criptografia de chave pública protege a mensagem na transmissão contra um terceiro qualquer. Porém, ela não protege as duas partes uma da outra [3]. Suponha que Bob e Alice troquem mensagens utilizando o esquema de chave pública, alguns problema podem surgir:

- Alice pode forjar uma mensagem diferente e reivindicar que ela veio de Bob.

- Bob pode negar o envio da mensagem para Alice.

Quando não se tem confiança completa entre o emissor e o receptor, é necessário algo mais que a autenticação de chave pública para a troca das mensagens. Para resolver esse problema é utilizada a assinatura digital. A assinatura digital precisa ter as seguintes características [3]:

- Verificar o autor e a data da assinatura.
- Autenticar o conteúdo no momento da assinatura.
- Ser verificável por terceiros.

A assinatura digital pode ser baseada em diversos algoritmos. O esquema usado neste trabalho utiliza como base o algoritmo RSA. O algoritmo RSA é bastante empregado em diversas aplicações, incluindo financeiras, por isso é importante que esse esquema seja considerado seguro. A garantia da segurança da função RSA está expressa na operação de geração de assinatura empregada para embutir o valor hash e na verificação da consistência da mensagem. Junto com o RSA é utilizada a técnica *Probabilistic Signature Scheme* (PSS) [3], [39].

O esquema RSA utiliza expressões com exponenciais. O texto claro é encriptado em blocos, com cada bloco sendo menor que a mensagem n. Tanto o emissor quanto o receptor precisam conhecer o valor de n. O emissor conhece o valor de e, e somente o receptor sabe do valor de d. O RSA possui duas chaves: uma pública com PU = e,n e uma privada PR = d,n. Para que a criptografia seja eficiente é necessário que seja impossível determinar d a partir de e e n [3]. Para se cifrar uma mensagem utilizando o RSA emprega-se a seguinte função:

$$m^e = c \ mod \ n$$

Para decifrar uma mensagem cifrada com RSA aplica-se a seguinte função:

$$c^d = n \ mod \ m$$

Para introduzir maior segurança na criptografia RSA foi adicionado um esquema de padding PSS. Um esquema de preenchimento comprovadamente seguro para a produção de assinaturas é normalmente utilizado como um esquema de criptografia segura. A função PSS usa duas funções além do hash para segurança. Dados aleatórios são inseridos na

mensagem M antes de ser executado o algoritmo de hash [39]. O PSS tem como principal objetivo garantir a comprovação matemática da real segurança do algoritmo RSA para a criptografia [40]. A Figura 2.6 apresenta o fluxo RSA.

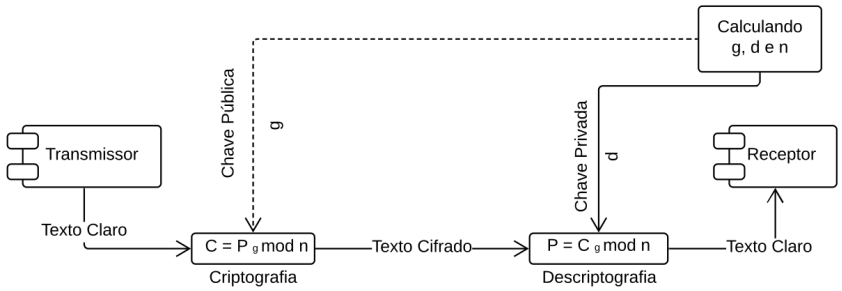


Figura 2.6: Fluxo RSA [2]

Para se criptografar uma mensagem com o algoritmo RSA os seguintes passos são executados conforme apresentado na Figura 2.6:

- Escolher dois números primos muito grandes, p e q.
- Multiplicar os dois primos escolhidos para descobrir n, que é o módulo para cifrar e decifrar.
- Escolher um inteiro aleatório g. Em seguida calcular o valor de d.
- Anunciar os valor g e n para o público. Manter o valor de g e d secretos.

2.5 Certificados X.509

A recomendação X.509 da International Telecomunication Union (ITU-T) [41] faz parte de uma série X.500 de recomendações que definem serviços de diretórios de banco de dados que mantêm informações sobre os usuários. A recomendação X.509 define uma estrutura para provisão de serviços de autenticação para os usuários. Esses diretórios podem servir como repositórios de chaves públicas [3]. O X.509 é um padrão importante e amplamente usado, baseado na utilização de criptografia de chave pública e assinatura digital, sendo recomendado o uso de RSA. Diversas informações são inseridas em um esquema X.509, porém, a principal é o certificado de chave pública associado a cada usuário. Os certificados presentes no serviço de diretórios X.509 são considerados como sendo criados por alguma autoridade certificadora (CA) [3].

As CAs tem uma hierarquia de certificados validados que seguem a estrutura de armazenamento X.509. Com isso é possível verificar se o certificado é válido pela hierarquia de CAs ou não, garantindo uma maior segurança. Os certificados revogados são emitidos

em uma *Certification Revocation List* (CLR), onde aparecem os certificados revogados pela CA. É necessário que esse repositório seja público para que seja feita a consulta da validade do certificado [42].

2.6 Assinatura Padrão XML Advanced Electronic Signature (XAdES)

As portarias do MEC e a nota técnica normatizam que os documentos devem ser assinados no formato XML Advanced Electronic Signature (XAdES). Esse padrão de assinatura é feito em documentos em formato XML com o objetivo de garantir autenticidade, integridade e não repúdio, definindo formas eletrônicas que assegurem sua validade por um longo período de tempo [43].

A Assinatura utilizando o padrão XAdES fornece uma autenticação básica, proteção de integridade e atende aos requisitos legais para assinaturas avançadas, conforme definido na Diretiva Europeia [EU-DIR-ESIG], porém nem todos os tipos de assinatura feitas no padrão XAdES garantem totalmente o não repúdio[43].

Existem diversos tipos de assinaturas que podem ser feitos utilizando o padrão XAdES [44]:

- XAdES-BES: forma básica de assinatura de documentos em formato XML, onde são garantidas a autenticidade e a integridade desses registros, entretanto não proporciona o não repúdio.
- XAdES-T: há a adição do carimbo de tempo na assinatura básica XadES-BES, o que garante também o não repúdio.
- XAdES-C: adiciona ao XAdES-T referências a um conjunto de dados que suportam a validação da assinatura eletrônica. Informações como referência ao caminho de certificação e informações de status e revogação do certificado. Esse tipo de assinatura é importante quando se for arquivar as informações de maneira externa.
- XAdES-X: baseia-se no XAdES-C adicionando carimbo de data e hora para proteção contra o risco de qualquer chave usada na cadeia de certificados ou nas informações de status de revogação que podem estar comprometidas.
- XAdES-X-L: adição ao XAdES-X acrescentando mais dados de validação, tais como valores de revogação dos certificados, para o caso das informações de revogação não serem armazenadas em outro local externo.

- XAdES-A: baseia-se no XAdES-X-L adicionando um campo de timestamp para arquivamento dos documentos assinados.

2.7 Arquitetura Microsserviços

O estilo arquitetural de microsserviços desenvolve uma estratégia de software que permite a construção de sistemas baseado na comunicação dos microsserviços, que são construídos de maneira independente. Hoje em dia ainda existe uma grande quantidade de sistemas corporativos que foram, ou ainda são, desenvolvidos utilizando a arquitetura monolítica, que tem a desvantagem de qualquer mudança na aplicação exigir uma nova compilação do código e a implantação do sistema como um todo [45].

O centro da arquitetura microsserviço são os microsserviços. Os microsserviços seguem o princípio da responsabilidade única. Os microsserviços têm um escopo bem definido e têm um limite explícito evitando seu crescimento desenfreado, assim não podendo ser nem muito grandes nem muito pequenos, pois quanto menor o serviço maior a maximização das vantagens e desvantagens da arquitetura de microsserviços [46].

Os microsserviços são unidades separadas, o que significa que cada um deles pode ser implantado como um serviço isolado em uma plataforma de maneira independente e autônoma. A comunicação entre os serviços é feita por meio de chamadas de rede, isso evita que haja um acoplamento rígido entre os serviços. Os serviços devem ter a capacidade de ser mutáveis sem interferir no funcionamento geral da arquitetura [46], [47].

A comunicação entre um microsserviço que provê a funcionalidade e um outro que consome a funcionalidade é feito por meio da uma interface que o microsserviço provedor disponibiliza para ser consumida por outros serviços. A comunicação deve ser feita de maneira que não seja dependente de tecnologia específica para comunicação, ou seja, o microsserviço provedor e o consumidor podem ser desenvolvidos em tecnologias distintas. Um dos mecanismos amplamente empregados para a comunicação entre os serviços é a comunicação utilizando o protocolo HTTP [46], [48]. Uma das principais vantagens com o uso de microsserviços é a escalabilidade. A escalabilidade significa que o microsserviço pode aumentar ou diminuir seu desempenho de acordo com a carga exigida no momento. Além disso a escalabilidade garante disponibilidade e tolerância a falhas [46], [49], [50].

A distribuição é uma outra caraterística importante dos microsserviços. Cada um dos microsserviços pode ser implantado separadamente em diferentes hosts. Isso ajuda na distribuição de carga, torna o sistema mais escalável e com melhor desempenho do que um sistemas com arquitetura monolítica. A distribuição dos serviços garante também uma alta disponibilidade, pelo fato de que uma falha em um microsserviço não afeta ne-

cessariamente os demais microsserviços [49]. Os microsserviços devem ter elasticidade, ou seja, que escalem dinamicamente de acordo com a carga requerida para o microsserviço. Devido a esse modelo, os microsserviços podem ser implantados em cluster utilizando os recursos computacionais de forma mais eficiente [49]. Uma grande vantagem dos microsserviços é a possibilidade de se utilizar diferentes tecnologias para cada serviço. Isso gera flexibilidade da escolha da melhor tecnologia para cada funcionalidade. Com isso é possível optar por diferentes maneiras de se armazenar cada um dos serviços, o que garante uma maior separação e independência entre eles [46], [49], [47].

De acordo com Newman [51], os microsserviços se baseiam no princípio de um modelo em torno de conceitos de negócios, adotando uma cultura de automação, ocultando os detalhes da implementação interna, descentralizando tudo, isolando as falhas nos serviços e com implantação de forma autônoma [51]. A arquitetura dos microsserviços permite que sejam utilizados para compor diversas aplicações diferentes, o que proporciona sistemas descentralizados, mais reconfiguráveis e flexíveis [52].

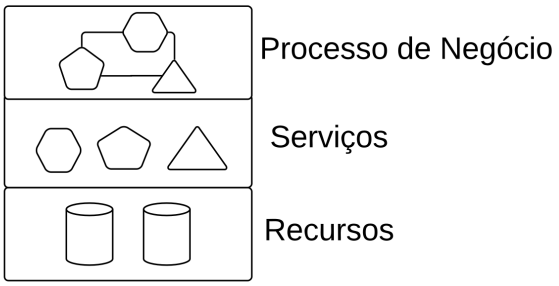


Figura 2.7: Arquitetura Microsserviços [4]

A Figura 2.7 apresenta as camadas básicas de uma arquitetura com microsserviços. Exitem três camadas e cada uma delas demonstra de maneira simplificada como funciona a integração entre os serviços [4]:

- A camada de recursos é onde ficam os recursos necessários, como banco de dados, que são utilizados pelos serviços.
- A camada de serviço fornece os serviços que abstraem e disponibilizam alguma parte de uma regra de negócio.
- A camada de processo de negócio consiste em uma série de operações executadas em uma sequência ordenada onde os serviços em conjunto fornecem uma funcionalidade de real valor para o negócio.

Atualmente, os microsserviços estão sendo migrados para o ambiente de nuvem, usando tecnologias leves de contêineres, seguindo práticas de DevOps com integração de software

totalmente automatizada. Isso permite a rápida implantação de microsserviços em diversos ambientes de maneira célere e padronizada [53].

2.8 Java

Hoje em dia é uma das maiores linguagens de programação, usada para desenvolver aplicações em desktop, web, aplicações enterprise e também aplicações para dispositivos móveis. Uma das vantagens da linguagem Java é que ela usa uma Máquina Virtual Java (JVM), que faz com que a linguagem possa ser desenvolvida, compilada em um sistema operacional e rodada em qualquer outro sistema que tenha essa máquina virtual instalada [5].

A linguagem Java inicialmente foi criada pela empresa SUN para ser uma linguagem para desenvolvimento e execução de programas desde televisores, dispositivos móveis, ou até mesmo torradeiras, com o objetivo de ser executada em diversos dispositivos sem necessidade de compilar em cada um deles para rodar [5].

De acordo com a Oracle, atual mantenedora do Java, a linguagem Java é uma linguagem de programação com o objetivo de reduzir custos, diminuir prazos de desenvolvimento, impulsionar a inovação e aprimorar serviços de aplicativos. Existem mais de 51 bilhões de aplicações executando em Java Virtual Machines em todo o mundo. Essa plataforma ainda hoje é a mais usada por empresas e desenvolvedores [54].

As principais características da linguagem Java são [5]:

- **Simples:** Comparada a linguagem C++, que é uma linguagem complexa, a qual faz uso de ponteiros. A vantagem do Java, além de poder ser executada em diversos dispositivos com uma única programação, não permite manipular ponteiros de maneira manual conforme no C++, o que torna a linguagem consideravelmente mais simples [5].
- **Orientada a Objetos:** Atualmente pode não parecer uma grande vantagem, pois a orientação a objetos hoje é o padrão de programação, porém, na década de 90 quando foi criado o Java, não existiam muitas linguagens que utilizassem esse método, assim provocou uma grande evolução na programação, tornando-a consideravelmente popular e permitindo sua utilização até os dias atuais [5].
- **Distribuída:** A linguagem Java tem características para criação de sistemas distribuídos. Isso proporciona o armazenamento em servidores e compatibilidade com os principais protocolos, o que possibilita a comunicação entre aplicações [5].

- **Robustez:** Java é uma linguagem robusta, ou seja, não tem comportamento imprevisível, tampouco uma quantidade de erros significativos. Possui também um gerenciamento de memória automático pela JVM.
- **Segurança:** Os sistemas integrados a JVM e ao Java garantem que, uma vez compilado o código, seja difícil sua alteração, o que garante uma estabilidade e uma segurança na linguagem [5].
- **Arquitetura Neutra:** Uma das grandes vantagens da linguagem Java é poder ser desenvolvida e compilada uma vez e ser rodada em qualquer dispositivo que contenha a JVM. O Java, diferente das linguagens que a antecederam, compila o código para *byte code* e posteriormente o *byte code* pode ser executado em qualquer dispositivo que tenha a JVM [5].

A portabilidade, uma das vantagens da linguagem java, pode ser demonstrada na figura 2.8:

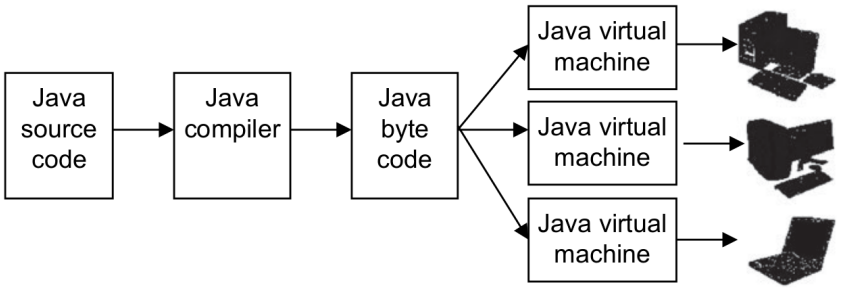


Figura 2.8: Código Distribuído [5]

2.9 Trabalhos Correlatos

A segurança da informação e os sistemas que garantem a autenticidade, privacidade e o não repúdio nos documentos e mensagens que irão trafegar pela internet é atualmente uma das prioridades do atual governo brasileiro. Diversas iniciativas na área têm sido tomadas para se obter documentos e sistemas mais seguros em formato digital. No contexto educacional, o MEC tem criado diretrizes e padrões para o desenvolvimento de documentos e diplomas no formato digital que sejam válidos e confiáveis. Atualmente, é a prioridade das Universidades Federais o desenvolvimento de soluções para assinatura de diplomas e documentos no formato digital. O MEC impôs que estes documentos sejam assinados por meio de um certificado digital no formado XAdES. Os documentos e diplomas assinados por essa solução devem ter o padrão de assinatura da ICP-Brasil [55],

[56] e ter validade em todo o território nacional. A UnB tem por objetivo desenvolver um sistema de assinatura digital em concordância com os padrões do MEC e da ICP-Brasil. O sistema deve utilizar as tecnologias e padrões arquiteturais atuais. Com esse objetivo foram pesquisados os principais artigos relevantes para o desenvolvimento deste trabalho.

O trabalho realizado por Boneh et al. [57], apresenta um conceito de assinatura agregada, demostrando a segurança que a agregação de assinaturas proporciona em vários aplicativos. Segundo os autores, a agregação de assinaturas são úteis para reduzir o tamanho da cadeia de certificados e para reduzir as mensagens em protocolos de roteamento seguro. As assinaturas agregadas, assim como a assinatura única, são criptografadas de forma verificável. Uma das aplicações práticas da agregação de assinaturas está na cadeia de certificados X.509 que pode ser agregada em uma simples assinatura.

Crampton [58] apresentou a necessidade de se utilizar criptografia e assinatura digital em documentos no formato XML para proteger a confidencialidade e fornecer garantias sobre a integridade desses documentos transmitidos através de meios inseguros. O autor focou em como algumas políticas de controle e acesso a documentos XML podem ser aplicadas utilizando criptografia. Foram também apresentadas técnicas para atribuir chaves a uma rede de segurança que podem ser adaptadas para minimizar o número de chaves distribuídas aos usuários. O autor propôs uma nova política de controle de acesso a documentos XML.

Kumar et al. [59] investigaram a necessidade de criptografia para imagens com o objetivo de garantir a segurança contra ataques. Os autores propuseram um novo algoritmo para segurança de imagem usando *Elliptic Curve Cryptography* (ECC). Esse algoritmo codifica primeiro a imagem RGB utilizando a codificação de DNA, seguida por criptografia assimétrica baseada na *Elliptic Curve Diffie – Hellman* (ECDHE). O algoritmo foi aplicado em imagens de testes para analisar os padrões. Os resultados desse estudo concluíram que o algoritmo proposto pode resistir a ataques exaustivos e é adequado para aplicações práticas.

Perin et al. [60] apresentaram uma comparação de duas arquiteturas de multiplicação modular. Uma matriz totalmente sistólica e uma implementação paralela. A multiplicação modular é empregada no processo de exponenciação modular. Um dos algoritmos criptográficos mais importantes é o esquema de criptografia RSA. A arquitetura de matriz totalmente sistólica apresenta uma implementação de alta raiz com propagação de transporte entre os elementos de processamento. A implementação paralela é composta por blocos multiplicadores em paralelo com os elementos de processamento e fornecendo um modo de operação em pipeline. Foi comparada a eficiência de tempo entre as duas arquiteturas utilizando o RSA. O tempo da arquitetura sistólica para descriptografar 1024 bits utilizando RSA foi de 3.4 ms e da arquitetura paralela foi de 6 ms. Foi identificado

um desempenho competitivo entre as duas arquiteturas.

Engelbertz et al. [61] avaliaram a lógica de validação da biblioteca de software de código aberto para criar e validar documentos assinados – fornecida pela *Connectiong Europe Facility* (CEF) chamada de *Digital Signature Service* (DSS) –, em relação a ataques baseados em XML. As vulnerabilidades descobertas permitiram ler arquivos do servidor e ignorar a proteção feita pelo XML *Advanced Eletronic Signature* (XAdES). Os autores concluíram que existe uma necessidade urgente de documentos de boas práticas de segurança e ferramentas automáticas de avaliação de segurança para apoiar o desenvolvimento de implementações relevantes à segurança.

Krylovskiy et al. [62] demonstraram a crescente evolução da internet, que vem com a internet das coisas (IoT), como uma nova visão para as Cidades Inteligentes. Existem esforços para padronizar o ambiente da web e as tecnologias de computação em nuvem. Em resposta aos desafios na criação de aplicativos em plataformas distribuídas na web, surgiu o estilo arquitetural de microsserviços. Os autores compartilharam a utilização de microsserviços para projetar uma plataforma IoT de cidade inteligente. Esse estilo arquitetural é comparado com a Arquitetura Orientada a Serviços (SOA). Foi demonstrado que a arquitetura de microsserviços proporciona uma abordagem que permite que as partes de um sistema sejam feitas de maneira altamente independentes e cada parte pode ter uma tecnologia diferente. Uma desvantagem da arquitetura de microsserviços é que, apesar de simplificar o design e a implementação de serviços individuais, tem uma complexidade maior na integração de sistemas distribuídos.

Dragoni et al. [63] analisaram a história da arquitetura de software, os motivos que levaram a difusão dos serviços e a criação dos microsserviços. Os autores demonstraram os desafios futuros e os problemas em aberto. Além disso, apresentaram o ponto de vista acadêmico e algumas questões práticas. Foi demonstrado que com o passar do tempo a arquitetura de microsserviços vem ganhando cada vez mais importância e popularidade, tanto na academia quanto no mundo industrial. A mudança para os microsserviços é um processo sensível, porém de suma importância.

Rahaman et al. [64] criaram uma solução para redução de falsos positivos que acontecem em APIs criptográficas. Os autores propuseram um conjunto de algoritmos chamado de CryptoGuard. Com esses algoritmos tem-se a redução de falsos alertas de 76% a 80%. Essa ferramenta é utilizada em projetos Apache de grande escala e de alto impacto. Forma anualizados 6.181 aplicativos em android que geraram insights de segurança. Foram feitas análises em projetos de grande porte em Java.

Ameya Ketkar et al. [65] descreveram a dificuldade de atualizações de um sistema quando uma linguagem atualiza suas funções. As linguagens não possuem uma forma automática de mudar uma aplicação de uma versão para outra. Os autores apresentaram

um estudo empírico sobre as mudanças que ocorreram no Java. Com ferramentas de mineração de dados foi possível verificar que 297.543 alterações ocorreram na linguagem Java. Os autores notaram com isso que as mudanças de tipo de dados são mais comuns que a renomeação de atributos e métodos.

Lepiane et al. [66] apresentaram uma visão sobre o problema da falsificação de diplomas físicos que ocorre no Brasil e como o certificado no formato digital pode ajudar a resolver essa questão. Os certificados devem ser seguros e legalmente aceitáveis em todo o território brasileiro. O trabalho dos autores propõe uma especificação técnica para que as instituições de ensino superior possam emitir certificados de acordo com as políticas adotadas pelo Governo Federal.

Diferentemente do trabalho de Lepiane et al. [66], este trabalho apresenta um modelo para a assinatura de diplomas e documentos no formato digital e não apenas uma especificação técnica, o que permitirá às instituições de ensino emitirem diplomas em conformidade com as políticas adotadas pelo Governo Federal. Dessa forma é apresenta uma análise e um modelo inicial de assinatura digital de documentos e diplomas para a Universidade de Brasília (UnB) nos padrões do MEC e da ICP-Brasil, com o objetivo de minimizar custos e adequar às especificações do MEC no prazo estipulado por este órgão. Embora essa solução tenha sido desenvolvida para a UnB, ela poderá ser utilizada por qualquer outra instituição que necessite realizar a assinatura digital dos seus diplomas e documentos.

O trabalho proposto por Oliveira Fernandes el al. [67], teve como objetivo um estudo técnico para a viabilidade da utilização de Blockchain nos diplomas em formato digital que devem ser compatíveis com os normativos do MEC e da ICP-Brasil. O estudo concluiu que as características do Blockchain mostram que é possível utilizá-lo para que os diplomas digitais sejam acessados de maneira segura e não somente para efetuar a assinatura dos diplomas digitais.

Um estudo em caráter exploratório desenvolvido pela Universidade Estadual de Goiás, teve como objetivo demonstrar uma alternativa ao uso da assinatura digital de acordo com as portarias do MEC. Os autores mencionaram que uma das alternativas viáveis para assinatura de maneira descentralizada é o uso de Blockchain. Os estudos revelam que o padrão estabelecido pelo MEC para assinatura digital tem diversas similaridades com a tecnologia Blockchain, podendo também ser empregado para garantir a segurança dos diplomas em formato digital [67].

Uma pesquisa realizada por Costa et al. [68] na Universidade Federal da Paraíba (UFPB) teve como objetivo a criação de uma rede Blockchain onde as Universidades Federais pudessem registrar os diplomas e certificados digitais. Essa rede permite que outras instituições possam verificar a veracidade de um documento assinado nessa cadeia.

O trabalho de Alexandre Bubrowsky [13] desenvolvido pela FGV para criação de um sistema de armazenamento e distribuição de diplomas em formato digital utilizando uma cadeia de Blockchain, teve como objetivo facilitar o armazenamento, a validação e o compartilhamento seguro do diploma assinado digitalmente. Um dos grandes desafios para criação de um Blockchain de diplomas digitais é a necessidade de integração de diversas entidades da rede pública e privada para a criação de uma rede de diplomas digitais tendo o MEC como entidade central dessa rede. O trabalho demonstra que é viável a criação de um Blockchain para compartilhamento seguro dos diplomas digitais, sem anular a necessidade da assinatura dos diplomas com o padrão XAdDES imposta nos normativos do MEC.

Ibarz e Cruellas [69] mencionaram uma nova forma de assinatura digital usando o Json Web Token (JWT) para assinatura digital no padrão JAdES. Esse padrão deve ter a mesma segurança e validade dos demais formatos de assinatura da família ETSI AdES (CAdES, PAdES e XAdES). O padrão JAdES tem ênfase especial na assinatura com o JWT e no carimbo de tempo. Essa assinatura deve ser válida com o passar do tempo nos mesmos moldes dos outros padrões. Ao final, os autores resumem os resultados obtidos por um programa desenvolvido para verificar a exatidão das abordagens técnicas adotadas e que servem como prova de conceito. Este trabalho serviu como ponto de partida para a construção de uma proposta formal de produção de um ETSI (European Telecommunications Standards Institute) com o uso de JWT [69].

O estudo desenvolvido por Palma et al. [70] teve como objetivo o uso de uma rede de Blockchain para validação de créditos acadêmicos e para a integração dos documentos assinados digitalmente com o padrão do MEC e da ICP-Brasil. Com essa rede, objetivou-se garantir uma maior segurança dos diplomas. Os autores desenvolveram um contrato inteligente para que cada estudante possa ter seu histórico da graduação e o diploma de conclusão de curso de maneira segura e livre de fraudes.

Diferente dos outros trabalhos, este trabalho tem o foco específico no desenvolvimento de um modelo arquitetural e criação de uma solução inicial para assinatura dos diplomas no padrão imposto pelos normativos do MEC e em conformidade com a atual arquitetura da UnB, bem como com a arquitetura de microsserviços utilizada atualmente em diversas organizações. Com o diploma assinado de acordo com esses padrões é possível ser feita a validação desse diploma nos validadores da ICP-Brasil.

2.10 Síntese do Capítulo

Neste capítulo foram apresentados os conceitos básicos necessários para o entendimento deste trabalho. Foram explanados os conceitos dos dois tipos de criptografia: simétrica e

assimétrica, as características de cada uma delas, o funcionamento da assinatura digital, as funções hash, a arquitetura microsserviços (que será utilizada para a implementação do sistema de assinatura digital) e a linguagem de programação Java.

Ao final do capítulo são apresentados os principais artigos relacionados com cada um dos assuntos apresentados bem como os mais relevantes trabalhos desenvolvidos por outras Universidades Federais Brasileiras para soluções de assinatura de diplomas e documentos no formato digital de acordo com os normativos do MEC e da ICP-Brasil. Pode-se concluir que as Universidades Federais estão desenvolvendo trabalhos de arquivamento e compartilhamento seguros dos diplomas digitais por meio da tecnologia de Blockchain, porém é necessário que esses diplomas estejam assinados com, no mínimo, o certificado A3 ou superior e com o padrão XAdES-T, que é um dos padrões do protocolo XAdES proposto nos dois normativos do MEC e na ICP-Brasil.

Capítulo 3

Metodologia

Este capítulo apresenta o método de pesquisa de estudo de caso proposto por Yin. De acordo com esse autor, o estudo de caso é um método de pesquisa focada no fenômeno, isto é, muita das vezes não se pode classificar o estudo de maneira qualitativa ou quantitativa o que requer múltiplos métodos para descrever o fenômeno estudado e os seus efeitos [1].

O estudo de caso é um método abrangente com um objetivo principal de analisar um fenômeno definido, examinando diversos dados e informações para chegar a uma conclusão sobre a eficácia ou não do fenômeno estudado [1]. É necessário também a definição das questões centrais de pesquisa, ou seja, qual o objetivo central que o trabalho tenta resolver e estabelecer todas as etapas sobre o estudo de caso de uso. As fases para o desenvolvimento do estudo de caso são apresentadas na Figura 1.1. O método segundo Yin, é dividido em seis passos [1]:

1. Planejar o estudo de caso.
2. Criar o estudo de caso.
3. Preparar o estudo de caso.
4. Coletar as evidências do estudo de caso.
5. Analisar as evidências do estudo de caso.
6. Compartilhar o estudo de caso.

Este capítulo, além de apresentar a visão geral do método de estudo de caso proposto por Yin [1], demonstra o que foi feito em cada uma das etapas para o desenvolvimento deste trabalho, desde a revisão da literatura abordando os principais artigos, as questões de pesquisa, o planejamento da arquitetura, o co-envolvimento até o compartilhamento das contribuições encontradas.

O estudo de caso desenvolvido neste trabalho é do tipo exploratório [1], onde será feita uma comparação entre a atual situação da UnB sobre assinatura de sistemas em formato digital, o que se pretende melhorar e quais as vantagens que este estudo tem em relação às atuais soluções utilizadas pela UnB. Foi feita uma análise qualitativa para verificar os problemas relativos a segurança e cifragem de diplomas em formato digital.

Nas seções a seguir serão demonstradas cada uma das seis etapas propostas para o desenvolvimento do estudo de caso e quais as técnicas usadas em cada uma dessas etapas.

3.1 Planejar o estudo de caso

Na fase de planejamento é necessário fazer a revisão de literatura para definir o problema a ser estudado. É importante a definição do método de pesquisa a ser utilizado para o desenvolvimento do trabalho [1]. Assim, a primeira atividade foi a realização de uma revisão da literatura com o objetivo de identificar as principais referências na área e as principais soluções propostas para o problema a ser resolvido por este estudo. A revisão da literatura foi apresentada no capítulo 2 e utilizou-se as principais bases de dados: Scopus, Web of Science, ACM Digital Library and IEEE Xplore Digital Library.

3.2 Criar o Estudo de Caso

Nesta etapa foi feita a compreensão dos conceitos básicos para o entendimento deste trabalho e dos pontos relevantes para este estudo. Foi definida a importância do estudo feito, quais as evidências que foram procuradas, quais as demais alternativas para a resolução dos problemas de pesquisa adotadas e porque a alternativa adotada foi a melhor solução para este estudo [1]. Na coleta dos dados e informações relevantes deve-se seguir um protocolo formal. Ao coletar essas evidências para o estudo de caso, deve ser questionado constantemente por que esses eventos aparecem de determinada forma. As informações quando consideradas relevantes ajudam a embasar o estudo de caso [1].

Nesta etapa devem ser levantados e analisados os trabalhos dos principais pesquisadores, salientadas e comparadas as questões definidas por eles e como responderam a essas questões, ou seja, quais as principais soluções encontradas para os problemas levantados. Nesta fase apresenta-se ainda a inovação trazida com o desenvolvimento do presente trabalho bem como em relação aos demais trabalhos referidos [1]. Ao responder as questões de pesquisa e verificar qual as principais técnicas e metodologias utilizadas para a assinatura digital e desenvolvimento de serviços web para esse objetivo, é possível a criação, nas próximas etapas, do modelo arquitetural de sistemas de maneira integrada à atual

arquitetura da Universidade e Brasília (UnB) e às melhores metodologias sugeridas pela literatura atual.

3.3 Preparar o Estudo de Caso

Nesta etapa é necessária a estruturação dos dados, o entendimento dos requisitos funcionais e não funcionais e o comparativo entre a situação atual dos dados e a situação futura demonstrada neste estudo. Foram feitas as análises, de preferência quantitativas, dos dados recuperados na seção anterior [1]. A estruturação dos dados e das informações necessárias podem ser obtidas de diversas formas. Neste trabalho os dados foram estruturados utilizando o comparativo do que já existe atualmente na STI da UnB (como dados de testes gerados), o que permitiu um comparativo para se obter a validação da solução mais promissora e que atinja os melhores resultados. [1]. Os dados foram recuperados dos estudos identificados na literatura e dos dados internos da UnB, assim foi possível obter as informações estruturadas para o desenvolvimento e o compartilhamento das inovações e contribuições científicas feitas por este trabalho [1].

A partir das informações coletadas pode-se definir os requisitos funcionais e não funcionais necessários para uma solução de assinatura digital. Nesta etapa também é demonstrada a arquitetura geral da UnB e qual a melhor solução para integrar de maneira efetiva a atual arquitetura utilizada por essa universidade. Com essas definições, foi delineada uma arquitetura minimalista, feita dentro da estrutura atual de segurança da UnB, para que os documentos sejam assinados de maneira segura e com validade por um longo período, com facilidade de evoluções e mudanças de padrões quando do surgimento de novas tecnologias de segurança e assinatura digital.

A solução desenvolvida para assinatura digital foi realizada utilizando a linguagem Java como um serviço independente e integrado à arquitetura da UnB. O objetivo é integrar de maneira fácil, com um bom desempenho e criar uma nova solução para assinatura digital no padrão dos normativos do MEC e da ICP-Brasil desenvolvidos na linguagem Java. Essa linguagem tem a vantagem de proporcionar um bom desempenho na assinatura de diversos documentos, com uma boa disponibilidade e segurança, além de ser uma linguagem robusta e massivamente usada pelas empresas. Nesta etapa, foi definida a arquitetura e a linguagem para o desenvolvimento da solução de assinatura digital. Nos próximos passos foi realizado o desenvolvimento do modelo de sistema, coletados os dados de teste e feitos os comparativos entre o sistema atual e a nova solução de assinatura desenvolvida neste trabalho.

3.4 Coletar as Evidências do Estudo de Caso

Nesta fase é necessário definir as formas de coletar as evidências para o estudo de caso. Existem diversas fontes e maneiras de coleta das evidências imprescindíveis para o desenvolvimento da solução esperada [1]. Neste trabalho foi utilizado o uso de documentos e artefatos físicos para o levantamento de dados e comparativo da situação atual e da situação futura proposta no estudo.

Os dados coletados têm por objetivo principal obter evidências sobre o estudo desenvolvido. Não é recomendada a utilização de apenas uma fonte de dados como evidência para o estudo. Neste trabalho foram adotadas diversas fontes de dados diferentes, para possibilitar uma ampla revisão dos principais dados e informações relevantes para o desenvolvimento do estudo de caso [1].

É necessário que um observador externo siga a origem das evidências dispostas no trabalho, isto é, seja capaz de mapear desde as questões iniciais até as conclusões do estudo de caso, de maneira que possa ser evidenciada a fonte de cada uma das informações apresentadas nesta pesquisa [1]. Para este estudo foi realizado um levantamento dos dados da UnB por meio de uma pesquisa documental e realização de análises implícitas para ambas as soluções em estudo. Por ser uma solução a um problema interno da UnB, todos os dados para análise são de domínio da própria Universidade.

A maior parte dos dados foi empregada para fazer a análise da atual situação de emissão de diplomas em formato digital, da quantidade ainda existente de diplomas em formato físico que são emitidos, da segurança da atual solução utilizada para a UnB, e da aderência dessa solução ao padrão exigido pelo MEC e pela ICP-Brasil. Com esse cenário a maior parte dos dados foi obtida por pesquisa documental e pela análise de testes e comparativo entre a atual solução e a solução proposta neste estudo.

A partir desta etapa se tem um modelo arquitetural básico da solução de assinatura digital consolidada, os requisitos funcionais e não funcionais definidos e as principais funções e modelos de classe já consolidados. Nesta etapa também foram feitos testes para verificar se a arquitetura, os requisitos e os modelos definidos estão seguros conforme as principais metodologias de verificação de segurança em aplicações REST.

3.5 Analisar as Evidências do Estudo de Caso

Nesta fase já se alcançou as evidências, já se realizou todos os testes e se obteve os dados necessários para definir a conclusão do trabalho. É necessário fazer a análise dos dados para tomar decisões seguras sobre as proposições iniciais estipuladas no princípio do estudo [1].

Neste trabalho foram feitas as análises comparativas entre as atuais soluções para a assinatura e diplomas em formato digital existentes na UnB e a solução proposta neste estudo. É relevante o comparativo de todos os testes feitos para a possibilidade de se desenvolver uma solução adequada em relação à segurança, à evoluções futuras e ao desempenho das soluções a serem comparadas. Feitas essas análises, é possível definir a conclusão mais viável e factível para o desenvolvimento de uma solução institucional de assinatura de diplomas em formato digital para a UnB. O objetivo principal nesta etapa, além de validar a nova solução de assinatura dos diplomas em formato digital, é também uma abordagem descritiva da nova solução apresentada e como esta será desenvolvida para se atingir uma segurança aceitável com um bom desempenho.

Segundo Yin [1], neste estágio é necessário ter certeza de que a análise feita é de alta qualidade, ou seja, se todas as evidências foram embasadas e se todas as metodologias concorrentes e os principais trabalhos e soluções encontrados para o mesmo tema foram examinados cuidadosamente [1]. Nesta etapa também foram apresentados os problemas que surgiram durante o desenvolvimento, os testes da solução para assinatura digital, os resultados encontrados e a conclusão do estudo, e, principalmente o que foi desenvolvido para o padrão XAdES conforme pressuposto pelas portarias do MEC e da ICP-Brasil. Também devem ser estruturados futuros trabalhos necessários para complementar este estudo e atender a todos os requerimentos da ICP-Brasil e do MEC.

Este estudo aborda apenas o modelo arquitetural do sistema de assinatura digital dos diplomas, faltando ainda outros requisitos que não foram abordados neste trabalho, tais como os requisitos para armazenamento, validação, infraestrutura e visualização. Para validar o modelo proposto foi desenvolvido um protótipo inicial e funcional de assinatura digital para garantir que a solução deste projeto fosse corroborada. Esse protótipo pressupõe uma biblioteca de assinatura digital na linguagem Java, considerando-se que atualmente a referida linguagem apresenta vantagens como sua facilidade de desenvolvimento, utilização de bibliotecas open source e a larga aplicação em sistemas corporativos. Além disso, foi constatado em pesquisa que ainda não existe uma solução de assinatura digital XAdES no padrão do MEC e da ICP-Brasil desenvolvida na linguagem Java.

Ao realizar os testes de segurança e performance, com o objetivo de validar o modelo proposto em relação a outras soluções, foi possível verificar a segurança e o desempenho da solução apresentada e traçar um comparativo entre as outras, com o objetivo de demonstrar as vantagens e desvantagens de cada possível solução. Dessa forma foi selecionada a melhor dentre as soluções de assinatura digital para a UnB.

3.6 Compartilhar o estudo de caso

Nesta etapa é necessário fazer o levantamento das inovações e o que este trabalho possibilita acrescentar para a comunidade científica, assim, foram publicados artigos demonstrado os pontos inovadores e que agregam para a comunidade. É importante relatar os desafios, dificuldades, vantagens e desvantagens que essa nova solução irá proporcionar [1].

3.7 Resultados

Nesta seção são descritos os resultados obtidos de acordo com as publicações selecionadas. A primeira questão de pesquisa foi respondida analisando os principais artigos que validam a segurança e eficácia da assinatura em formato digital. Para responder a segunda questão de pesquisa foi analisado os principais artigos, modelos e padrões estudados que utilizam a assinatura de documentos em formato digital.

3.7.1 Análise Relacionada à Primeira Questão de Pesquisa

QP.1. O que evidencia a integridade, validade e segurança em um documento em formato digital de acordo com a literatura existente?

Para responder essa questão, buscou-se verificar os principais trabalhos que justifiquem a integridade, validade e segurança em documentos digitais de acordo com o padrão XAdES, que é o que foi mencionado nas portarias do MEC e utilizado nesta pesquisa. A Tabela 3.1 apresenta os artigos identificados na literatura que abordam o assunto.

De acordo com a *Electronic Signatures and Infrastructures* [71], o XML tem o dado e o metadado para garantir diversas características de segurança do documento assinado no padrão XAdES. Algumas tags são colocadas dentro desse padrão em combinação com algumas tecnologias, como o uso de hash e do algoritmo RSA, que garantem a integridade e a segurança dos dados do padrão XAdES:

- O elemento principal para garantir a integridade é o elemento <ds:Reference>. Dentro desse elemento existe duas tags: a chamada <ds:DigestMethod> que é o local onde é colocado o método hash para garantia da integridade. Logo abaixo é colocada a tag <ds:DigestValue> contendo o valor hash em base64 do XML assinado, com isso a integridade do documento é garantida.

Outra característica importante que deve apresentar um documento XML é que ele seja seguro e tenha validade em todo território nacional. Para garantir essas características o

Tabela 3.1: Lista dos estudos identificados

ID	Título	Referência	Ano
S1	Applying Hierarchical and Role-Based Access Control to XML Documents	[58]	2004
S2	Montgomery modular multiplication on reconfigurable hardware: Fully systolic array vs parallel implementation	[60]	2010
S3	Network and internetwork security: principles and practice	[2]	2012
S4	Network and internetwork security: principles and practice	[3]	2014
S5	Electronic Signatures and Infrastructures (ESI); XAdES digital signatures;	[71]	2016
S6	Europass framework for digitallysigned credentials.	[72]	2018
S7	Development of e-diploma system model with digital signature authentication.	[73]	2018
S8	Security analysis of xades validation in the CEF digital signature services (DSS)	[61]	2019
S9	Bringing JSON signatures to ETSI AdES framework: Meet JAdES signatures.	[69]	2020

padrão XAdES é assegurado pela força da chave que vai assinar o documento em formato XML:

- O elemento <ds:SignatureValue> é o elemento que armazena o valor atual – utilizando a tecnologia RSA –, da assinatura digital realizada convertida para base64.
- A tag <ds:X509Certificate> armazena o valor do certificado X.509, com isso é possível verificar a veracidade da chave pública, o que garante a validade do documento.

Além desses elementos que garantem a integridade e a validade do certificado, outra importante característica para afiançar que o documento foi assinado no tempo que o certificado digital era válido é o carimbo de tempo. É importante verificar a data que o certificado foi assinado. Algumas características do carimbo de tempo:

- O carimbo de tempo é um elemento <ds:TimeStampType> onde dentro dele se encontra o Hash da data e hora que o certificado foi assinado.

Algumas das principais utilizações da assinatura digital no padrão XAdES são relatadas na literatura. Crampton [58] demonstrou a necessidade de se utilizar criptografia e assinatura digital em documentos no formato XML para proteger a confidencialidade e fornecer garantias sobre a integridade desses documentos transmitidos através de meios inseguros. O autor criou uma política de controle de acesso para os documentos em XML, utilizando características como integridade e validade desses documentos.

Em alguns outros países já é usado o diploma assinado digitalmente. Um exemplo disso é o que está sendo desenvolvido na Indonésia. Finandhita e Afrianto [73] demonstraram a importância da assinatura do diploma em formato digital e apresentaram as vantagens dessa assinatura. A solução dos autores gera o documento tanto em pdf quanto em formato XML, ambos válidos. A União Europeia também está utilizando a assinatura digital no padrão XAdES, o chamado Europass, que é um certificado digital emitido em cartão para cada estudante com todos os certificados e diplomas pra formar um currículo válido digitalmente [72].

3.7.2 Análise Relacionada à Segunda Questão de Pesquisa

QP.2. Quais as principais técnicas, modelos e padrões que têm sido sugeridos na literatura para assinar um documento em formato digital de maneira segura?

Tanto na União EuroOpeia, que está criando um currículo digital para seus estudantes, quando no Brasil, que está criando diplomas em formato digital, é utilizado o padrão de assinatura XAdES [72], [7]. Os principais padrões de assinatura digital são:

- **CAdES:** É um padrão de assinatura digital que faz a assinatura em diversos formatos de arquivos. Os arquivos são assinados em formato binário, gerando um arquivo p7s que contém a assinatura digital. O problema dessa assinatura é a dificuldade de leitura [74].
- **XAdES:** Esse padrão faz uma assinatura diretamente em um documento em formato XML. A vantagem é que os dados e os metadados ficam no mesmo arquivo. Esse arquivo é de fácil leitura, escrita e integração entre as aplicações [71].
- **PAdES:** É uma assinatura também em formato binário semelhante ao padrão CAdES, porém focado apenas em arquivos em formato pdf. Utiliza o padrão ISO 32000-1 e faz a assinatura diretamente no pdf. É um padrão de leitura mais complicado que o XAdES [75].

Pela facilidade de leitura e padronização que o XML oferece e a fácil integração desses arquivos assinados com outras aplicações, como leitura de algum campo ou outro tipo de processamento, foi escolhido nesta proposta o uso do padrão XAdES.

3.8 Síntese do Capítulo

Neste Capítulo foi apresentada a metodologia utilizada para o desenvolvimento do trabalho, a qual compreende as seis etapas de evolução de um estudo de caso, de acordo com

Yin [1]. Na primeira etapa, que consiste em planejar o estudo de caso, foi realizada uma revisão da literatura para identificação das soluções propostas. Foi verificado também quais as principais arquiteturas para o desenvolvimento de sistemas. Após planejar o estudo de caso, seguiu-se a etapa da criação do referido estudo e nesta etapa foi determinado um modelo para o desenvolvimento da solução, definindo-se assim os processos para a assinatura de documentos em formato digital. Nos próximos capítulos será preparado o estudo de caso e apresentada a arquitetura de suporte para a implantação do modelo de assinatura de documentos no formato digital. Além disso, será apresentado o funcionamento da atual arquitetura da UnB e como se pode integrar novas soluções à arquitetura atual dessa instituição.

Capítulo 4

Legislação e Modelo Base Para Assinatura Digital

Este capítulo apresenta a atual legislação aprovada no Brasil sobre assinatura de diplomas no formato digital e os padrões preconizados pela ICP-Brasil, ou seja, os modelos e padrões que o governo brasileiro estipulou para que seja efetivada a criação de um modelo de assinatura digital. Além disso são apresentados ainda, quais tags devem estar presentes para a assinatura digital no formato XAdES. Por fim é exposta a arquitetura de assinatura digital sugerida pelo MEC e pela ICP-Brasil.

4.1 Legislação Sobre Assinatura Digital

Recentemente o Ministério da Educação (MEC) criou algumas portarias para que as Universidades Federais Brasileiras, tanto da rede pública quanto privada, possam emitir documentos válidos em formato digital. Uma dela é a portaria nº 330, de 5 de abril de 2018. Por meio dessa portaria, foi instituído o diploma digital no âmbito das Universidades Públicas e Privadas pertencentes ao Sistema Federal de Ensino. De acordo com essa portaria, o diploma digital deve abranger o registro e o histórico escolar do estudante. Apenas as instituições de ensino que tem prerrogativa para emissão e registro de diplomas podem emitir em formato digital. É ressaltado nessa portaria que os diplomas assinados digitalmente devem atender as diretrizes de certificação da ICP-Brasil, o que garante a validade em todo território nacional. Por fim, a referida portaria estipula o prazo de 24 meses para a implantação da solução de assinatura digital pelas instituições de ensino superior federais [76].

A portaria nº 554, de 11 de março de 2019, tem o objetivo de normatizar a obrigatoriedade da emissão de diplomas de graduação em formato digital pelas instituições federais de ensino superior. Essa portaria menciona que o diploma emitido em formato

digital deve ser preservado em ambiente computacional que garanta: a) validade a qualquer tempo; b) interoperabilidade entre sistemas; c) atualização tecnológica da segurança; e d) possibilidade de múltiplas assinaturas em um mesmo documento [27].

Os signatários do diploma digital devem ser os mesmos signatários do diploma em formato físico, exigindo-se a assinatura com o certificado digital da ICP-Brasil tipo A3 ou superior. O certificado deve ser um certificado oficial da instituição de ensino superior que assina o diploma no formato digital. O diploma deve ser assinado em formato XML com a assinatura no formato XML Advanced Electronic Signature - XAdES e fornecida uma URL para validação e disponibilização do diploma assinado digitalmente. O diploma deve ser assinado seguindo o Padrão Brasileiro de Assinatura Digital - PBAD. Além da assinatura em formato XML, a portaria nº 554 determina que o diploma deve estar disponível de maneira visual e legível [27].

Outro recurso que a portaria nº 554 preconiza é que, além da assinatura do diploma em formato XML, deve ser disponibilizado um QRCode no diploma em formato visual como uma forma segura e válida de acessar o diploma em formato XML. A consulta do diploma, tanto no formato visual quanto no formato XML, deve ser disponibilizada no site da instituição federal de ensino. Todos os diplomas assinados digitalmente devem ser encaminhados para o Ministério da Educação para que conste como parte do acervo acadêmico do estudante [27].

As portarias nº 330 e nº 554 normatizam como disponibilizar um diploma em formato digital que seja aceito em todo território nacional e de acordo com as atuais diretrizes do MEC e da ICP-Brasil. Além dessas duas portarias, o Ministério da Educação disponibiliza os layouts, padrões e a Nota Técnica Número 13/2019/DIFES/SESU/SESU (atual INSTRUÇÃO NORMATIVA Nº 1, DE 15 DE DEZEMBRO DE 2020) que tem por objetivo proporcionar maior especificidade técnica de cada critério envolvido na emissão e no registro do diploma digital [7] [77].

De acordo com a nota técnica nº 13/2019/DIFES/SESU/SESU (atual INSTRUÇÃO NORMATIVA Nº 1, DE 15 DE DEZEMBRO DE 2020), é necessário o armazenamento inteiramente no meio digital, e cuja validade jurídica é presumida mediante a assinatura com certificação digital e carimbo de tempo na Infraestrutura de Chaves Públicas Brasileira - ICP-Brasil, conforme os parâmetros do Padrão Brasileiro de Assinaturas Digitais - PBAD. Apenas as assinaturas que seguirem os padrões preconizados pelas portarias, normas técnicas e pelo XML Schema Definition - (XSD) disponibilizado pelo MEC serão consideradas válidas [7], [77].

O diploma digital é nato-digital, ou seja, é emitido e armazenado inteiramente em meio digital. O diploma digital deve ser armazenado de acordo com os procedimentos e tecnologias que permitam a validação ao longo do tempo. Deve ser observada a evolu-

ção tecnológica, podendo ser mudado o formato do documento armazenado para garantir sua autenticidade, integridade, confiabilidade, disponibilidade, rastreabilidade, irretratabilidade, tempestividade, privacidade, legalidade, interoperabilidade e validade jurídica nacional [7]. O diploma emitido em formato digital deve ter a mesma validade e seguir a mesma legislação federal vigente que regula a emissão e o registro do diploma físico. É de responsabilidade das Instituições de Ensino Superior seguir a legislação, normativos e o fluxo interno para a emissão de diploma em formato digital [7], [77].

4.1.1 Extensible Markup Language - XML

A nota técnica nº 13/2019/DIFES/SESU/SESU (atual INSTRUÇÃO NORMATIVA Nº 1, DE 15 DE DEZEMBRO DE 2020) especifica o XML como padrão a ser adotado para os diplomas assinados em formato digital. De acordo com essa nota técnica, o XML é uma linguagem de marcação extensível que consegue armazenar todos os tipos de dados, e informações de forma estruturada e de fácil leitura por pessoas e por sistemas de computadores. É uma linguagem que independe de tecnologia, sistema operacional ou uma linguagem específica para ser manipulada [7], [77], [78].

A estrutura de um documento XML é formada por marcações, que consistem em elementos. Por sua vez, um elemento consiste em uma tag de início e de fim da informação. O XML deve ser bem formado, ou seja, respeitar as regras de formação de um XML. Um exemplo do uso de XML é apresentado na Figura 4.1, bem como as regras para se ter um XML bem formado [6]:

```
<?xml version="1.0" ?>
<w3resource>
    <design>
        html xhtml css svg xml
    </design>
    <programming>
        php mysql
    </programming>
</w3resource>
```

Figura 4.1: Exemplo XML bem formado [6]

- O documento XML deve conter apenas um elemento Raiz e todos os outros elementos devem estar dentro do elemento raiz;
- Todos os elementos devem ter uma tag de abertura e uma tag de fechamento. Os elementos que não têm conteúdo entre eles podem ter a tag de abertura e fechamento na mesma tag, como por exemplo: <design />;

- Os elementos devem estar corretamente aninhados. A tag de abertura e de fechamento devem estar no mesmo nível;
- Os valores dos atributos devem estar entre aspas ou apóstrofes. O mesmo atributo não pode estar contido no mesmo elemento;
- O XML diferencia caracteres maiúsculos de minúsculos (case sensitivo).
- Cada arquivo XML somente poderá ter uma única declaração <?xml version = "1.0"?>. Essa declaração garante que o XML segue a especificação XML 1.0 definida pela W3C.
- O XML deve ter também uma única declaração do namespace no elemento raiz de acordo com o que é mostrado no XSD disponibilizado pelo MEC. Não se pode usar prefixos nos namespaces.

4.1.2 XML Schema Definition - XSD

Para garantir a integridade e a correta formatação do XML é necessário gerar o XML baseado em um Schema Definition (XSD) definido pelo Ministério da Educação. O XSD é um documento baseado no padrão XML que orienta a definição de um documento XML e serve também para verificar sua validade. Os documentos XSD serão sempre mantidos e atualizados para atender aos padrões necessários para assinatura de diplomas em formato digital [7], [77].

Os arquivos XSD devem ser constantemente atualizados a fim de estabelecer correspondência entre a legislação federal vigente e os atuais padrões tecnológicos. O XSD atualmente disponibilizado pelo MEC para a criação de um diploma no formato digital está presente no portal do diploma digital disposto pelo MEC [7], [26], [77]. O XSD contém as definições de tipo, tamanho, ocorrência e regras de preenchimento dos elementos que devem compor o documento XML [7]. O MEC apresenta alguns arquivos XSDs para especificação do diploma digital e como deverá ser feita a assinatura [7]:

- **DiplomaDigital_v1.00.xsd:** define a estrutura sintática do XML do diplomado.
- **Documentação Acadêmica Registro do Diploma Digital_v1.090.xsd:** define a estrutura sintática da documentação acadêmica para emissão e registro.
- **Tipos Básicos_v1.00.xsd:** responsável pelo controle sintático de todos os tipos utilizados dentro de outros arquivos desta especificação técnica do diploma digital. Serve para garantir a integridade sintática dos campos utilizados no diploma digital.
- **Leiaute do Diploma Digital_v1.00.xsd:** Responsável pelo controle sintático da estrutura do XML do diplomado.

- **Leiaute da Documentação Acadêmica do Registro do Diploma Digital_v1.00:** Responsável pelo controle sintático da estrutura XML da documentação acadêmica para emissão e registro.
- **xmldsig-core-schema_v1.1.xsd:** responsável pelo controle sintático da estrutura do XML da assinatura digital.

4.1.3 Representação Visual do Diploma Digital

A representação visual do diploma digital é apenas um dispositivo para facilitar a visualização do diploma, não podendo ser substituído pelo diploma digital no formato XML, apenas é uma outra representação complementar ao diploma digital assinado no padrão XAdES. O diploma visual em formato digital deve adotar o mesmo modelo dos diplomas emitidos em formato físico. Na representação visual do diploma digital podem ser inseridas imagens de assinaturas físicas, desde que assegurada a sua validade jurídica e os requisitos de segurança da portaria nº 544/2019 [7], [77].

O diploma digital é um documento nato-digital que necessita que sua representação visual disponha de mecanismos de acesso ao XML do diploma digital assinado. A representação visual do diploma digital deve conter um Quick Response Code (QR Code), que é capaz de fazer o link entre a representação visual do documento digital e o documento digital em formato XML [7]. O QR Code deve conter um Resource Locator (URL) único do diploma em formato digital, com um apontamento direto para o local onde os dados podem ser acessados. O QR Code deve ser posicionado no canto inferior direito, com as dimensões e qualidade que permitam a leitura por um aplicativo de leitura de QR Code [7], [77].

A URL única que deve ser disponibilizada pela Instituição de Ensino Superior (IES) deve seguir o seguinte formato: http://<URL Institucional do Diploma>/<CodigoValidacao> e deve ser feito o uso de proteção contra robôs de busca por meio de CAPCHAS para repetidas consultas à mesma URL. Além disso, deve ser disponibilizado nessa URL o XML do diploma digital para download, a representação visual do diploma digital e o status do diploma indicando, ou seja, se este se encontra ativo ou anulado [7], [77].

4.1.4 Código de Validação de um Diploma Assinado

Todo diploma assinado digitalmente no padrão XAdES deve gerar um código de verificação. Esse código permite validar o estado atual do diploma digital registrado pela IES. O diploma será considerado válido e possuidor de integridade através da validação das assinaturas digitais nele aplicadas [7], [77]. A composição do código será formulada por três grupos de dados: Código do e-MEC da IES Emissora, Código e-MEC da IES

Registradora e Código de localização do diploma, e deve ser construído de acordo com a Figura 4.2:

Figura 4.2: Estrutura do código de validação [7]

A geração do código de localização do diploma digital deve ser formada por, no mínimo, os primeiros 12 caracteres do hash calculado pelo algoritmo SHA256 [7], [3], [79], [77], gerado a partir da concatenação do CPF do diplomado com o CNPJ da emissora. Para a criação do código de validação do diploma digital, os caracteres devem ser convertidos para string UTF-8 antes de se efetuar a concatenação. O código de validação do diploma digital deve ser incluído dentro do XML assinado [7], [77].

4.1.5 Documentação e Padrões para Emissão do Diploma Digital

As IES devem possuir permissão para emissão de diplomas com validade em território nacional. As IES que não possuem prerrogativas para registro de diplomas são chamadas de IES Emissoras e as IES que possuem direito de registrar diploma são chamadas de IES Registradoras. As primeiras devem encaminhar para uma IES Registradora as informações referentes ao diplomado, assinadas digitalmente, cumprindo a legislação e demais normas internas da IES Registradora. Fica a cargo desta última a criação do diploma e a assinatura do XML nos padrões impostos pelo MEC [7], [77].

Nas assinaturas digitais usadas no diploma digital, o MEC determina a utilização de um carimbo de tempo. O carimbo de tempo é um documento eletrônico emitido por uma parte confiável, a Autoridade Certificadora do Tempo (ACT), que serve como evidência de que uma informação digital existia em uma determina data e hora. O carimbo de tempo oferece a informação de data e hora do registro do documento, isto é, quando este chegou a ACT e não a data de criação do documento [7], [77]. A validação de um diploma digital deve sempre ocorrer em sua versão XML, sendo a versão visual apenas uma visualização gráfica das informações no XML. A primeira parte da validação de um diploma digital é a verificação das assinaturas digitais do diploma assinado, com o objetivo de averiguar a validade e eficácia jurídica [7], [77].

Após a verificação das assinaturas presentes no diploma assinado é necessário verificar a conformidade do XML assinado com a versão do Schema XSD disponibilizado pelo

MEC e em vigência no seu período de emissão. Os diplomas devem sempre ser emitidos utilizando a atual versão do Schema XSD disponibilizado pelo MEC. O último passo é a verificação da URL única do diploma a fim de verificar se seu status se encontra ativo ou inativo [7], [77].

4.1.6 Conformidade com a Lei Geral de Proteção de Dados - LGPD

O atual diploma digital disponibilizado pelos normativos do MEC atende a todas as especificações da LGPD, em especial no que tange à publicação dos dados privados do usuário de maneira segura. Com a assinatura do diploma digital em formato XML, os dados que não são considerados públicos ficam apenas de posse das Instituições de Ensino Superior, das IES Emissoras e das IES Registradoras, para fins do processo de registro e de acervo. O diploma deve ser guardado podendo ser solicitada a sua invalidação pelo usuário e a emissão de uma nova via quando este considerar que algum dado foi exposto e fere a sua privacidade. Quando um diploma é invalidado os dados pessoais pertencentes não devem ser mais expostos [7],[77].

4.1.7 Partes Importantes da especificação XSD

De acordo com o documento Diploma Digital_v1.00.xsd, um diploma digital válido é aquele cujo elemento raiz é chamado de Diploma e atende as especificações determinadas pela legislação, conforme apresentado na Figura 4.3.

```
<xs:complexType name="TDiploma">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="infDiploma"
        type="TInfDiploma"/>
    <xs:element ref="ds:Signature"/>
  </xs:sequence>
</xs:complexType>
```

Figura 4.3: XSD para Elemento raiz [7]

O elemento infDiploma é um elemento obrigatório que deve estar presente no XML do diploma digital. O elemento deve conter os seguintes itens: dados do diploma, dados do registro e a versão do diploma, conforme apresentado na Figura 4.4 [7], [77].

Os dados do diplomado do curso da IES Emissora e da assinatura devem seguir a especificação do XSD do diploma digital com o objetivo de determinar os dados obrigatórios para conter no XML da assinatura digital, conforme apresentado na Figura 4.5 [7], [77].

```
<xs:complexType name="TInfDiploma">
  <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="DadosDiploma"
type="TDadosDiploma" />
     <xs:element minOccurs="1" maxOccurs="1" name="DadosRegistro"
type="TDadosRegistro" />
  </xs:sequence>
  <xs:attribute name="versao" type="TVersao" use="required">
  </xs:attribute>
  <xs:attribute name="id" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:ID">
        <xs:pattern value="VDip[0-9]{44}" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Figura 4.4: Informações do diploma [7]

```
<xs:complexType name="TDadosDiploma">
  <xs:sequence>
    <xs:element name="Diplomado" type="TDadosDiplomado"/>
    <xs:element name="DadosCurso" type="TDadosCurso"/>
    <xs:element name="IesEmissora" type="TDadosIesEmissora"/>
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element ref="ds:Signature" />
    </xs:sequence>

  </xs:sequence>
  <xs:attribute name="id" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:ID">
        <xs:pattern value="Dip[0-9]{44}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Figura 4.5: Dados Presentes no Diploma [7]

A tag de dados do diplomado deve estar em conformidade com a especificação definida no XSD com os seguintes elementos: id, pessoa, nacionalidade, naturalidade, CPF, RG e data de nascimento, conforme apresentado na Figura 4.6 [7].

```
<xs:complexType name="TDadosDiplomado" >
  <xs:sequence>
    <xs:element name="ID" type="TId"/>
    <xs:group ref="GPessoa"/>
    <xs:element name="Nacionalidade" type="TNacionalidade"/>
    <xs:element name="Naturalidade" type="TNaturalidade"/>
    <xs:element name="CPF" type="TCpf"/>
    <xs:element name="RG" type="TRg"/>
    <xs:element name="DataNascimento" type="TData"/>
  </xs:sequence>
</xs:complexType>
```

Figura 4.6: Dados do Diplomado [7]

Após a definição dos dados do diplomado, os dados da IES emissora também devem ser definidos, conforme a especificação do XSD disponibilizado pelo MEC. A presença de

cada campo deve ser verificada para garantir que o diploma digital esteja em conformidade com o pressuposto pelo MEC, conforme apresentado na Figura 4.7.

```
<xs:complexType name="TDadosIesEmissora">
  <xs:sequence>
    <xs:element name="Nome" type="TNomeIES"/>
    <xs:element name="CodigoMEC" type="TCodIESMEC"/>
    <xs:element name="CNPJ" type="TCnpj"/>
    <xs:element name="Endereco" type="TEndereco"/>
    <xs:element name="Credenciamento" type="TAtoRegulatorio" />
    <xs:element name="Recredenciamento" type="TAtoRegulatorio" />
    <xs:element minOccurs="0"  name="RenovacaoDeRecredenciamento"
        type="TAtoRegulatorio" />
    <xs:element minOccurs="0" name="Mantenedora">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="RazaoSocial" type="TRazaoSocial"/>
          <xs:element name="CNPJ" type="TCnpj"/>
          <xs:element name="Endereco" type="TEndereco"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Figura 4.7: Dados da IFES Emissora [7]

Outra parte do XSD disponibilizado pelo MEC é a inclusão dos dados do curso do diplomado. É importante que seja inserido o nome, o código do curso cadastrado no MEC, dentre outros atributos que são informados no documento XSD, conforme apresentado na Figura 4.8.

```
<xs:complexType name="TDadosCurso">
  <xs:annotation>
    <xs:documentation>Dados do curso</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="NomeCurso" type="TString"/>
    <xs:element name="CodigoCursoEMEC" type="TCodCursoMEC"/>
    <xs:element name="NomeHabilitacao" type="TString"/>
    <xs:element name="Modalidade" type="TModalidadeCurso"/>
    <xs:element name="TituloConferido" type="TTituloConferido"/>
    <xs:element name="GrauConferido" type="TGrauConferido"/>
    <xs:element name="EnderecoCurso" type="TEndereco"/>
    <xs:element minOccurs="0" name="Polo" type="TPolo"/>

    <xs:element name="Autorizacao" type="TAtoRegulatorio"/>
    <xs:element name="Reconhecimento" type="TAtoRegulatorio"/>
    <xs:element minOccurs="0" name="RenovacaoReconhecimento"
        type="TAtoRegulatorio"/>

  </xs:sequence>
</xs:complexType>
```

Figura 4.8: Dados do Curso [7]

A próxima etapa a ser definida pelo diploma digital são os dados do registro. Esse campo define a estrutura dos dados referentes ao registro do diploma e deve estar de acordo com a especificação do XSD, conforme apresentado na Figura 4.9.

```
<xs:complexType name="TDadosRegistro">
  <xs:sequence>
    <xs:element name="IesRegistradora"
        type="TDadosIesRegistradora"/>
    <xs:element name="LivroRegistro" type="TLivroRegistro"/>
    <xs:element name="IdDocumentacaoAcademica">
      <xs:simpleType>
        <xs:restriction base="xs:ID">
          <xs:pattern value="ReqDip[0-9]{44}" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Seguranca" type="TSeguranca"/>
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element ref="ds:Signature" />
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="id" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:ID">
        <xs:pattern value="RDip[0-9]{44}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Figura 4.9: Dados do Registro do Diploma Digital [7]

Além dos dados da IES Emissora, é necessário também que o diploma digital tenha os dados da IES Registradora que devem estar em conformidade com o XSD disponibilizado pelo MEC, conforme apresentado na Figura 4.10.

```
<xs:complexType name="TDadosIesRegistradora">
  <xs:sequence>
    <xs:element name="Nome" type="TNomeIES"/>
    <xs:element name="CodigoMEC" type="TCodIESMEC"/>
    <xs:element name="CNPJ" type="TCnpj"/>
    <xs:element name="Endereco" type="TEndereco"/>
    <xs:element name="Credenciamento" type="TAtoRegulatorio"/>
    <xs:element minOccurs="0" name="Recredenciamento"
        type="TAtoRegulatorio"/>
    <xs:element minOccurs="0" name="RenovacaoDeRecredenciamento"
        type="TAtoRegulatorio"/>

    <xs:element name="Mantenedora">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="RazaoSocial" type="TRazaoSocial"/>
          <xs:element name="CNPJ" type="TCnpj"/>
          <xs:element name="Endereco" type="TEndereco"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Figura 4.10: Dados da Registradora [7]

Além dos dados do Diplomado da IES Emissora, da IES Registradora, dos dados referentes ao diploma e dos dados do curso, devem também constar os dados dos responsáveis pelo armazenamento das informações, de localização, do registro e do livro no qual o

diploma foi registrado, devendo estar em conformidade com o XSD, de acordo com o apresentado na Figura 4.11:

```
<xs:complexType name="TLivroRegistro">
  <xs:sequence>
    <xs:element name="LivroRegistro" type="TCodLivroRegistro"/>
    <xs:element name="NumeroFolhaDoDiploma"
        type="TNumFolhaDoDiploma"/>
    <xs:element name="NumeroSequenciaDoDiploma"
        type="TNumSequenciaDiploma"/>
    <xs:element name="ProcessoDoDiploma"
        type="TCodProcessoDoDiploma"/>
    <xs:element name="DataColacaoGrau" type="TData"/>
    <xs:element name="DataExpedicaoDiploma" type="TData"/>
    <xs:element name="DataRegistroDiploma" type="TData"/>
    <xs:element name="ResponsavelRegistro">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Nome" type="TNome"/>
          <xs:element name="CPF" type="TCpf"/>
          <xs:element name="IDouNumeroMatricula" type="TId"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Figura 4.11: Dados Livro de Registro [7]

É necessário também a adição do código de validação que deve ficar dentro da tag de segurança de acordo com o XSD, conforme apresentado na Figura 4.12:

```
<xs:complexType name="TSeguranca">
  <xs:sequence>
    <xs:element name="CodigoValidacao" type="TCodigoValidacao"/>
  </xs:sequence>
</xs:complexType>
```

Figura 4.12: Dados do código e Validação [7]

Segundo a norma técnica disponibilizada pelo MEC, o elemento <signature> é um elemento de inclusão obrigatório para que se obtenha o diploma digital assinado conforme o padrão XAdES. A assinatura deve ser feita com todos os elementos dentro da tag <diploma> e armazenada dentro dessa mesma tag [7], [77].

4.2 Padrão de Assinatura de Diplomas em Formato XML

Nesta seção será explicado como deve ser feita a assinatura dos diplomas assinados em formato digital, quais tags devem estar presentes na assinatura, qual o padrão mínimo XAdES deve ser usado para a assinatura digital e uma visão geral do processo de assinatura digital de maneira integrada à atual arquitetura da UnB.

O XSD disponibilizado pelo MEC para fazer a autenticação no padrão XAdES é o XSD padrão para assinatura digital de documentos em formato XML disponibilizado pelo *Massachusetts Institute of Technology* (MIT) [1]. No site do MEC sobre assinatura de diplomas digitais pode ser encontrado esse mesmo XSD para possibilitar assinar um documento seguindo esse padrão [26], [77].

4.2.1 XML com base no XSD Disponibilizado pelo MEC

Para a criação de um documento XML básico, visando a geração da arquitetura desenvolvida neste trabalho, foi utilizado como base o XSD disponibilizado pelo MEC, o qual gera um XML, de acordo com o apresentado na Figura 4.13.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Created with Liquid Technologies Online Tools 1.0 (https://www.liquid-technologies.com) -->
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#"
           xsi:schemaLocation="http://www.w3.org/2000/09/xmldsig# schema.xsd"
           Id="AAAAA"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="https://www.liquid-technologies.com" />
    <SignatureMethod Algorithm="https://www.liquid-technologies.com" />
    <Reference>
      <DigestMethod Algorithm="https://www.liquid-technologies.com" />
      <DigestValue>YTM0NZomIzI2OTsmIzM0NTueYQ==</DigestValue>
    </Reference>
    <Reference>
      <DigestMethod Algorithm="https://www.liquid-technologies.com" />
      <DigestValue>YTM0NZomIzI2OTsmIzM0NTueYQ==</DigestValue>
    </Reference>
    <Reference>
      <DigestMethod Algorithm="https://www.liquid-technologies.com" />
      <DigestValue>YTM0NZomIzI2OTsmIzM0NTueYQ==</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue Id="AAAAB">YTM0NZomIzI2OTsmIzM0NTueYQ==</SignatureValue>
  <KeyInfo Id="AAAAC">
    <MgmtData>string</MgmtData>
    <SPKIData>
      <SPKISexp>YTM0NZomIzI2OTsmIzM0NTueYQ==</SPKISexp>
      <SPKISexp>YTM0NZomIzI2OTsmIzM0NTueYQ==</SPKISexp>
      <SPKISexp>YTM0NZomIzI2OTsmIzM0NTueYQ==</SPKISexp>
    </SPKIData>
    <KeyName>string</KeyName>
  </KeyInfo>
  <Object Encoding="https://www.liquid-technologies.com" />
  <Object Id="AAAAD" Encoding="https://www.liquid-technologies.com" />
  <Object MimeType="string" Encoding="https://www.liquid-technologies.com" />
</Signature>
```

Figura 4.13: XML exemplo gerado a partir do XSD

Os elementos da Figura 4.13 são os elementos para a assinatura digital de documentos XML no padrão XAdES. Todo documento XML deve começar com a declaração <?xml version="1.0"encoding="utf-8"?>. Essa declaração mostra a versão do XML que é usada (atualmente é recomendado que se use a versão 1.0) e, adicionalmente, pode ser inserida a codificação que é utilizada no XML [80]. De acordo com o XSD, após a declaração inicial, o elemento principal da assinatura é o elemento <Signature>. Todos os outros elementos utilizados na assinatura XAdES devem estar dentro desse elemento, assim esse elemento pode conter todos os XML Namespaces (xmlns) utilizados na assinatura. [43], [81].

[1] http://www.w3.org/2000/09/xmldsig#

O elemento posterior ao Signature é o <SignedInfo>. Este elemento inclui o algoritmo de canonização, o algoritmo de assinatura e uma ou mais referências. Também pode conter um elemento opcional ID, o qual permite que esse elemento seja referenciado por outras assinaturas e objetos [81]. Um dos elementos incluídos dentro de <SignedInfo> é o elemento <CanonicalizationMethod>. Esse elemento é obrigatório e especifica o algoritmo de canonização aplicado ao elemento SignedInfo antes da execução do cálculo da assinatura. O algoritmo de canonização preconizado tanto pelo MEC quanto pela ICP-Brasil é apresentado na Figura 4.14. O método canônico define como são validados os caracteres presentes no documento XML [81].

```
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
```

Figura 4.14: Elemento para definir o método canonical utilizado

Outro elemento interno ao SignedInfo é o elemento SignatureMethod. Esse elemento é obrigatório e especifica o algoritmo usado para geração e validação das assinaturas. Esse algoritmo identifica todas as funções criptográficas envolvidas na operação de assinatura. O método de assinatura preconizado pelo MEC e pala ICP-Brasil é o SHA256. O elemento deve ser adicionado conforme apresentado na Figura 4.15 [81].

```
<SignatureMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
```

Figura 4.15: Elemento signature Method a ser usado na assinatura do XML

O próximo elemento é o elemento Reference. Esse elemento pode estar presente uma ou mais vezes dentro do elemento SignedInfo. Ele especifica o algoritmo de digest usado e o valor desse algoritmo, e pode opcionalmente identificar o objeto que está sendo assinado [81]. Dentro do elemento Reference existem dois outros elementos, o DigestMethod e o DigesteValue. O elemento DigestMethod identifica o algoritmo de digest a ser aplicado ao objeto assinado. Esse elemento utiliza a codificação SHA1 ou a SHA256. Atualmente, é recomendado pela ICP-Brasil e as portarias do MEC, o uso da codificação SHA256. O elemento DigestValue contém o valor codificado em base64 da codificação do algoritmo do DigestMethod usado na assinatura XAdES [81]. Após o elemento SignedInfo é necessário que seja incluído o valor real da assinatura digital. Esse valor é codificado em base64 e é adicionado no elemento SignatureValue [81].

O próximo elemento é o KeyInfo. Esse elemento é um elemento opcional que permite ao destinatário obter a chave necessária para validar a assinatura. Dentro dele podem estar contidos nomes, chaves, certificados e outras informações para gerenciamento das chaves públicas [81]. O último elemento para que seja assinado o XML no padrão XAdES é o elemento Object. Esse elemento é um elemento opcional que pode aparecer diversas

vezes na assinatura digital de um documento XML. Esse elemento pode conter qualquer dado necessário para complementar a autenticação XAdES. A ICP-Brasil recomenda que nele estejam contidos os dados da autoridade certificadora, o nome do dono do certificado, assim como o número e o identificador de cada certificado que assinou o XML [81], [31].

4.2.2 Visão Geral do Processo de Assinatura Digital

O sistema para assinatura de diplomas digitais deve seguir todos os passos para que seja feita a assinatura no padrão XAdES e no padrão recomendado pelo MEC e pela ICP-Brasil. Para efetuar a assinatura é necessário primeiro que o documento XML esteja no formato canônico isto é, um formato específico para formatação padrão de documentos em formato XML. A formatação utilizada é a c14n [82]. Essa formatação pode aceitar um XML com comentários ou sem comentários. O sistema de assinatura digital criado deve transformar a formatação do XML para o padrão c14n, ou seja, transformar a solução de assinatura de diplomas em formato digital da codificação do UTF-8 pra a c14n [71], [83], [7], [77].

De acordo com o MEC e com a ICP-Brasil o diploma em formato XML deve estar no formato UTF-8 e, como dito acima, para que seja assinado de acordo com o padrão XAdES, o documento deve ser transformado para padrão c14n de formatação. Esse processo tem por objetivo padronizar os caracteres e espaçamentos de um documento XML. Algumas regras devem ser seguidas para este padrão de formatação:

- Todos os comentários do XML original são removidos.
- O documento XML deve ser transformado para a codificação UTF-8 caso esteja em outra codificação.
- Os documentos em formato XML criados no Windows são produzidos com uma formatação diferente dos documentos criados no Unix, o que causa uma diferença significativa na hora de assinar o documento XML. A codificação deve ser padronizada para o padrão c14n.
- Todos os espaços em branco entre os caracteres devem ser mantidos.
- Elementos vazios são convertidos em elementos em pares. Por exemplo o elemento <foo /> deve ser convertido para <foo></foo>.
- O valor dos atributos dos elementos devem ser normalizados. Os delimitadores devem ser definidos com aspas duplas. O espaço em branco entre os elementos deve ser normalizado.
- Os espaços excedentes entre os elementos devem ser removidos.

- Os valores default dos atributos de cada um dos elementos dentro do XML devem ser adicionados ao documento canonizado.

Para ser feita a transformação de UTF-8 para o padrão c14n e efetuar a assinatura digital é usada a biblioteca do Java XAdES4j. Essa biblioteca possui os métodos necessários para assinatura digital nos padrões XAdES com o carimbo de tempo (XAdES-T), com isso é necessário apenas o certificado para assinatura e o diploma a ser assinado, para que seja feita a assinatura nos padrões XAdES de acordo com os XSD do MEC.

Para a assinatura digital o diploma já deve vir com todos os dados do diplomado, da IES emissora, os dados do curso, os dados da IES registradora, as informações no livro de registro, o código do e-MEC formado pelo código da IES emissora, o código da IES registradora além do código da localização do diploma digital. Caso venha faltando alguns desses elementos no diploma, deve ser enviado um erro e não deve ser efetuada a assinatura desse diploma. No modelo é necessário também a validação do diploma para verificar se estão presentes todas as tags necessárias para se efetuar a assinatura digital.

Sendo válido o diploma é necessário fazer a assinatura do diploma em formato XAdES de acordo com o padrão da ICP-Brasil e o Padrão Brasileiro de Assinatura Digital - PBAD. O padrão PBAD tem por objetivo impor regras de validação e de criação de assinaturas digitais para que os documentos tenham a segurança necessária e possam ser validados por instituições nacionais e internacionais [7], [77]. De acordo com o documento da ICP-Brasil para assinatura digital a assinatura XAdES deve ser embasada na assinatura XMLDsig. Além dos elementos obrigatórios da assinatura XAdES devem estar contidos os seguintes elementos: DataObjectFormat (para assinaturas do tipo detached), SigningCertificate e SignaturePolicyIdentifier [8]. Um certificado, de acordo com a ICP-Brasil, deve ter pelo menos os seguintes atributos para sua validade, conforme apresentado na Figura 4.16 [8].

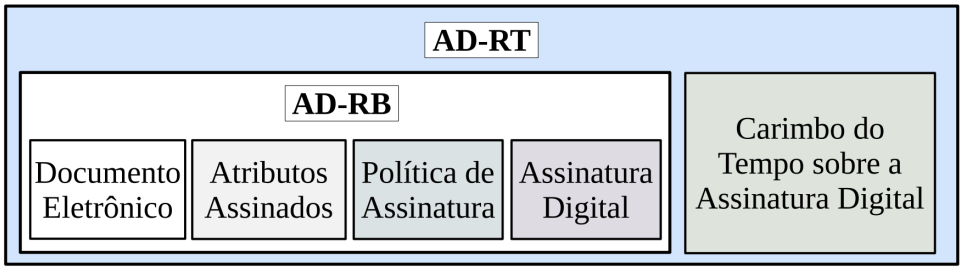


Figura 4.16: Assinatura digital com Referência de Tempo [8]

Esse tipo de assinatura deve ser utilizado nos processos de negócio nos quais a assinatura digital necessita de segurança em relação à irretratabilidade do momento de sua geração. Essa validação não contempla as informações de referência da lista de certificados revogados importantes para validações posteriores, porém, essas informações podem

ser conseguidas de forma externa e a validação poderá ser feita por meio do validador oficial da ICP-Brasil [84].

De acordo com a atual estrutura que a ICP-Brasil fornece para realizar a validação, é seguro que se use o padrão de assinatura com referência de tempo (AD-RT), conforme apresentado na Figura 4.16. Após validar o diploma é necessário verificar se o elemento principal Signature já possui um atributo ID. Caso possua o ID, esse elemento deve ser usado, caso contrário, deve ser gerado um atributo ID aleatório no elemento Signature. Esse atributo serve para identificar a assinatura, uma vez que um diploma pode ter mais de uma assinatura.

Após a adição do atributo ID no elemento Signature deve-se acrescentar a função hash inicial do diploma em formato XML. O formato do Hash definido pela nota técnica do MEC é o formato SHA256 e o elemento que deve ser usado para gerar o hash é o elemento Diploma. O hash desse elemento deve ser gerado após a transformação para o formato c14n [7]. Depois de gerado o hash para o elemento Diploma é criada a assinatura seguindo o padrão xmldsig. Em seguida são incluídos no elemento Object os elementos necessários para que a assinatura seja válida de acordo com o padrão da ICP-Brasil. Além disso, é necessário adicionar os elementos referentes ao tipo de assinatura utilizada. Dentro do elemento Signature é acrescentado o atributo xmlns:ds, o qual representa o padrão de assinatura usada, que deve ser o XMLDsig. No elemento SignatureValue é adicionado o valor da assinatura efetuada transformada para a base64 [81].

É necessário também colocar o valor da chave pública que assinou o certificado. Esse valor é adicionado dentro do elemento X509 Certificate. Com esse valor é possível garantir a autenticidade da assinatura e a verificação de sua validade [81]. De acordo com o normativo do MEC e das políticas de assinatura da ICP-Brasil é necessária a adição de outras informações dentro da assinatura para que esteja no padrão brasileiro. Essas informações devem ser adicionadas dentro do elemento Object conforme determina o padrão XAdES [81]. Após a assinatura do diploma com o padrão XAdES é necessário a adição do carimbo de tempo. A necessidade da utilização desse carimbo de tempo é para evidenciar o exato período de tempo em que uma determinada assinatura foi criada. Quando o certificado é revogado o carimbo de tempo serve para verificar se o diploma foi assinado no período em que a assinatura era válida [85].

Para que o diploma esteja no padrão de assinatura do MEC devem ser assinadas partes do diploma, assim como ele por completo. No caso do padrão do MEC é necessário fazer 3 assinaturas no mesmo documento, entretanto, é possível fazer diversas assinaturas em um mesmo diploma [84].

4.3 Síntese do Capítulo

Este capítulo apresentou resumidamente a portaria nº 330 de 5 de abril de 2018 e a portaria nº 554 de 11 de março de 2019 que autorizam e normatizam a emissão de diplomas em formato digital pelas Instituições de Ensino Superior que têm autorização para emissão de diplomas. Além dessas duas portarias, este capítulo apresenta a nota técnica número 13/2019/DIFES/SESU/SEUS. Essa nota especifica que o diploma deve ser emitido no formato XML com assinatura no padrão XAdES. Foi apresentado também o XSDs para a validação dos campos necessários para a criação de um diploma no formato XML. Após a apresentação dos padrões para realizar a assinatura foi exposto o padrão de assinatura digital disponibilizado pelo ICP-Brasil. Esse padrão consiste em um grupo de normas que formam o Padrão Brasileiro de Assinatura Digital – PBAD. O diploma deve ser assinado seguindo esse padrão, pois com o seu uso, o diploma tem validade jurídica em todo o território nacional. Ao final foi apresentado o processo para fazer a assinatura digital de um diploma em formato digital.

Capítulo 5

Arquitetura e Requisitos do Sistema de Assinatura Digital

Este Capítulo apresenta o modelo arquitetural minimalista do sistema de assinatura de diplomas e documentos no formato digital. São apresentados os principais módulos de assinatura, os requisitos funcionais e não funcionais para o desenvolvimento da referida solução e ainda, o diagrama de classes e pacotes para demonstrar como funciona a relação entre cada módulo do sistema de assinatura digital. Ao final deste capítulo é explanado como é feita a interação do sistema de assinatura digital com o barramento utilizado na UnB. Também é apresentado como esse sistema pode ser usado por outras soluções desacopladas do barramento da UnB. É necessário que esse sistema seja um serviço independente desenvolvido em Java que se integre facilmente com o barramento de serviços ou qualquer outra solução e que não acarrete ônus desnecessário para a referida universidade.

5.1 Arquitetura dos Sistemas da Universidade de Brasília

A arquitetura de desenvolvimento das aplicações fornecidas pela UnB sofreu mudanças para atualizá-las e modernizar o processo de desenvolvimento e manutenção dos sistemas utilizados na UnB. Para evoluir no desenvolvimento e modernizar os sistemas foi utilizado o modelo arquitetural aderente ao padrão SOA [10].

A arquitetura desenvolvida para os sistemas da UnB faz uso de algumas práticas de design do modelo Domain-Driven Design (DDD) tais como: a divisão de um domínio maior em domínios menores e mais coesos e a utilização de uma linguagem ubíqua, que é focada em uma linguagem centrada nos domínios do negócio para auxiliar a equipe a raciocinar

a camada dos serviços a serem desenvolvidos e o conjunto de domínios necessários para o desenvolvimento de uma aplicação [10].

Os atuais serviços baseados no uso do novo modelo de desenvolvimento de software para a UnB são concebidos a partir da especificação dos modelos de domínio de negócio em conformidade com o que é esperado pela camada de apresentação desses serviços [10]. Cada serviço desenvolvido deve tratar apenas de uma pequena parte do domínio do negócio de maneira bem coesa para facilitar o desenvolvimento e futuras manutenções no serviço. Os serviços devem ser bem documentados e disponibilizados em um catálogo de serviços para que não sejam feitos de maneira duplicada. O foco desse novo modelo de desenvolvimento está na criação e também na reutilização dos serviços criados pela camada de visão [10].

A camada de visão da arquitetura dos sistemas da UnB funciona com a abordagem do padrão arquitetural Single Page Application (SPA). O SPA é um padrão de aplicações na web que possui apenas uma única página a qual é atualizada com o conteúdo necessário para cada serviço solicitado utilizando a tecnologia AJAX, conforme apresentado na Figura 5.1 [9].

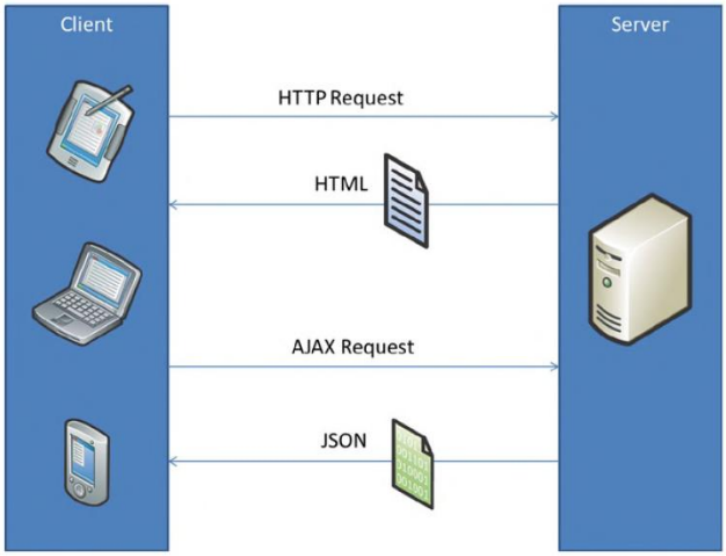


Figura 5.1: Padrão SPA [9]

A primeira requisição, HTTP Request, retorna à página HTML com todo o conteúdo necessário para que a aplicação funcione. As demais requisições utilizam o Ajax Request. Uma requisição Ajax Request só retorna o conteúdo essencial (em formato JSON) para a atualização da página gerada [9], conforme apresentado na Figura 5.1. As atuais aplicações da UnB desenvolvidas na camada de visão, também chamadas de aplicações frontend, estão operando na linguagem Angular e atualmente encontram-se na versão 10 desse framework. As aplicações frontend se comunicam com os serviços em backend através

de um barramento de serviços. O barramento de serviços utilizado foi desenvolvido na linguagem Erlang pela própria UnB e tem por objetivo o controle da comunicação entre o frontend e o backend por meio do modelo REST.

O Barramento proporciona o controle de autenticação e autorização nas chamadas REST utilizando o protocolo Oauth2 [86]. O barramento também oferece suporte ao protocolo LDAP, utilizado em algumas aplicações da UnB para controle de autenticação [86].

5.2 Arquitetura do Barramento de Serviços

O barramento de serviços utilizado pela UnB foi desenvolvido na linguagem Erlang [11, 10]. Essa linguagem tem um modelo de simultaneidade baseado em processo com passagem de mensagens assíncronas. Os mecanismos de simultaneidade em Erlang são leves, ou seja, os processos requerem pouca memória e a criação/exclusão de processos e a passagem de mensagens exigem pouco esforço computacional. O Erlang é utilizado para sistemas de programação de software em tempo real onde é necessário tempo de resposta na faixa de milissegundos [87].

A linguagem Erlang tem grande capacidade em sistemas que não podem ser interrompidos. Uma das principais características dessa linguagem é que ela não possui memória compartilhada, toda a interação entre processos é feita por passagem de mensagem [87]. O barramento de serviços foi desenvolvido utilizando a arquitetura de microsserviços [11, 10]. Cada processo do barramento é um serviço separado que é integrado por meio de um orquestrador de serviços central, chamado de dispatcher [10, 11].

O processo principal dispacher chama o serviço específico requisitado pelo frontend. O barramento de serviços da UnB possui o serviço de intermediador entre as requisições do frontend e os serviços do backend. Possui também o serviço de autenticação pelo protocolo LDAP e o serviço de autenticação e autorização pelo protocolo OAuth2 [10, 11], [86].

A Figura 5.2 apresenta como funciona a interação das aplicações frontend com o barramento utilizado pela STI/UnB. É demonstrado como são feitas as chamadas do frontend para o processo central do barramento (o dispatcher) e a interação entre o processo central e os outros serviços presentes no barramento [10].

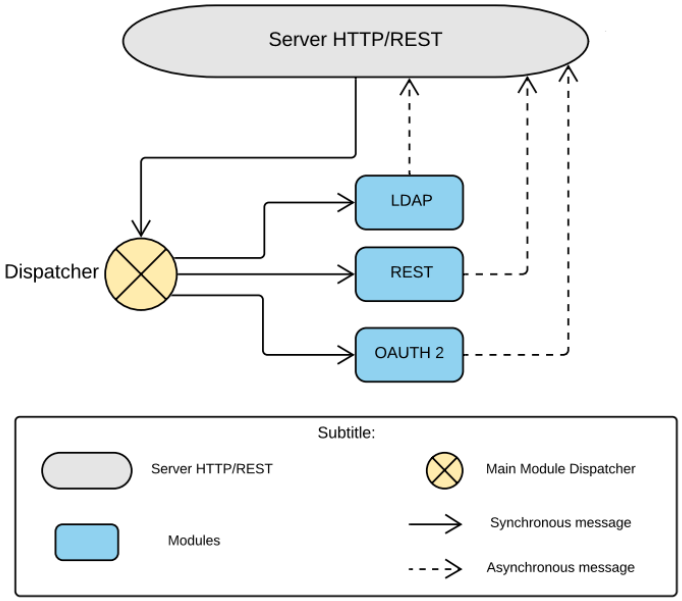


Figura 5.2: Arquitetura do barramento de serviços [10, 11]

5.3 Alternativas Para Criação de uma Solução de Assinatura Digital

Com essa nova arquitetura sendo utilizada pela STI/UnB para os sistemas novos, e com o barramento servindo como um serviço tanto para a parte de segurança – utilizando os protocolos LDAP e OAuth2 – quanto para a comunicação entre as aplicações frontend e os serviços no backend, foi demonstrado que a melhor alternativa para assinatura de documentos e diplomas no formato digital é a criação de um serviço backend em java com a comunicação intermediada pelo barramento de serviço para fornecer o serviço para o frontend. Outras alternativas foram levantadas para a criação de uma solução de assinatura digital como por exemplo a utilização de uma solução proprietária. A principal desvantagem dessa solução consiste em ser mais demorada para se obter futuras evoluções que serão necessárias e, além disso, gera uma dependência da solução utilizada, o que dificultaria quando necessário a troca futura para uma nova solução.

Foram encontradas poucas empresas com soluções para assinatura de diplomas em formato digital de acordo com o padrão do MEC e da ICP-Brasil. Um dos frameworks para assinatura desenvolvido pelo Serviço Federal de Processamento de Dados (SERPRO) é o Singer. Essa solução proposta pelo SERPRO tem por objetivo a assinatura digital em diversos padrões de acordo com os padrões do MEC e da ICP-Brasil [88].

Um dos principais problemas do uso do framework proposto pelo SERPRO, é que até o momento, não existe uma maneira de fazer assinatura no padrão XAdES, uma solução para isso ainda se encontra em desenvolvimento. Outras empresas que trabalham

com uma assinatura no padrão XAdES de acordo com a ICP-Brasil foram pesquisadas, entretanto não foi encontrada nenhuma solução totalmente compatível com os padrões de assinatura impostos pelo MEC [88]. Outra alternativa seria transformar o atual sistema em serviços e fazer uma intermediação deles com o barramento. O problema principal dessa solução é que o sistema de assinatura atualmente utilizado pela UnB, além de não estar no padrão do MEC e da ICP-Brasil, realiza a assinatura em um sistema em desktop e não via web.

Outra opção seria a criação de um serviço interno ao barramento de serviços Erlangms. O problema dessa solução é que não existe muitas bibliotecas que facilitem a assinatura de documentos digitais no padrão XAdES e que proporcionem uma real vantagem tanto em segurança quanto em performance (vide Capítulo 6), onde é possível verificar o comparativo de segurança e performance entre as diferentes soluções estudadas.

Com base nos testes realizados, a solução adotada neste trabalho foi a criação de uma solução desenvolvida no backend, utilizando a linguagem Java para assinatura digital e o barramento Erlangms como um intermediador para a chamada desse serviço. Foram feitas pesquisas para identificar as bibliotecas de assinatura digital em Java tendo sido encontrada a biblioteca open source XAdES4j, que veio facilitar muito a criação de uma solução de assinatura aderente às necessidades da UnB e de acordo com os normativos do MEC e da ICP-Brasil.

5.4 Requisitos do Sistema de Assinatura Digital

O Ministério da Educação (MEC) disponibilizou a portaria nº 330, a portaria nº 554 e a nota técnica de 19/2019 que autorizam as Universidades Federais emitirem diploma em formato digital. Dentro de cada uma dessas normas é exigido que sejam seguidas algumas diretrizes. O sistema de assinatura digital desenvolvido neste trabalho deve acatar o que é colocado em cada um desses normativos do MEC. Nas Tabelas 5.1, 5.2 e 5.3 são mostrados cada um dos requisitos exigidos pelo MEC para a solução de assinatura de diplomas no formato digital:

5.5 Arquitetura do Sistema de Assinatura Digital

A solução de assinatura digital deve ser feita em duas etapas. Na primeira etapa é necessário a utilização de um módulo em desktop que faz a leitura do certificado, tanto A1 como A3. A leitura do certificado é feita por meio de uma aplicação em Java usando um template básico em thymeleaf, com isso o usuário ao iniciar o SDK aciona uma tela

Tabela 5.1: Histórias de Usuários

1	Como um usuário do sistema eu quero poder assinar os diplomas de maneira digital para que seja melhorado a disponibilidade, segurança e redução de custos com emissão de diplomas.
2	Como um usuário do sistema eu quero poder validar os diplomas de maneira digital para que seja garantido a segurança e a adequação aos padrões exigidos.
3	Como usuário eu quero que seja possível a disponibilização e armazenamento do diploma a qualquer tempo para que o diploma fique disponível a todo tempo e armazenado de maneira segura por um longo período.
4	Como gestor eu quero que os diplomas assinados tenham validade jurídica em todo o território nacional.

Tabela 5.2: Requisitos Funcionais

1	Assinatura de diplomas em formato digital.
2	O diploma digital deve ter validade jurídica nacional.
3	O diploma digital deve seguir os padrões do MEC e da ICP-Brasil.
4	O diploma deve ser validado em validadores nacionais.
5	O diploma digital deve ser assinado no padrão XAdES-T.
6	O diploma deve ser armazenado de maneira segura por um longo período de tempo.
7	O diploma deve ser disponibilizado de maneira segura digitalmente.
8	O diploma deve ser válido à longo prazo.

Tabela 5.3: Requisitos Não Funcionais

1	O diploma digital deve estar em formato XML.
2	O XML do diploma digital deve ser assinado no padrão XAdES-T.
3	Deve seguir as diretrizes do MEC descritas nas duas portarias e na norma técnica.
4	O diploma deve seguir os parões da ICP-Brasil.
6	Deve ser assinado um lote de 200 a 500 diplomas em no máximo 30 minutos.

no browser para que ali ele faça localmente o upload do tipo de certificado que será armazenado em um sistema de arquivo no servidor [73].

O fluxo para leitura do certificado, tanto A1 quanto A3, é realizado baixando e executando o SDK feito em Java com spring boot com um template web local. Depois de fazer a execução do SDK é aberto um layout no browser com uma tela de login, onde será feita a autenticação via LDAP. Após isso será apresentada uma tela onde será feito o upload do certificado A1 ou A3 com a respectiva senha. Para ser feito o upload do certificado A3 é necessário a instalação das dlls da certificadora e a compatibilidade dessas dlls com

o SDK. Feito isso, o SDK salva o certificado em um banco de dados para que o diploma possa ser posteriormente assinado com esse certificado salvo [73].

É importante que o sistema de assinatura digital seja tão seguro quanto o sistema atual utilizado pela UnB e faça a assinatura de um lote de documentos de forma efetiva e nos padrões do MEC e da ICP-Brasil. O módulo de assinatura digital deve ser um módulo minimalista que utilize a biblioteca open source XAdES4j para assinatura de documentos no formato XAdES [89], [12].

A biblioteca XAdES4j é uma implementação Java configurável e extensível para assinar documentos em formato XML no padrão XAdES. Essa biblioteca tem a capacidade de fazer assinatura nos formatos XAdES-BES, XAdES-EPES, XAdES-T e XAdES-C. Essa API fornece um alto nível de abstração, tratando de todos os detalhes estruturais do XAdES [89]. A biblioteca XAdES4j tem por objetivo a verificação de assinaturas o mais simples possível, do mesmo modo, é flexível o suficiente para abranger diversos cenários diferentes [12].

Nessa biblioteca exitem dois fluxos básicos, o fluxo de assinatura digital e o fluxo de validação da informação, que são realizadas por provedores de serviços [12]. Um provedor de serviços tem uma interface específica e é usado durante a produção de assinaturas e verificação de algum dado ou funcionalidade. Os provedores permitem configurações independentes, o que garante que a substituição de um provedor de serviço por outro não interfere no processo de assinatura, o que faz com que possam ser adicionadas novas funcionalidades de maneira facilitada [12].

Essa biblioteca tem como base uma extensibilidade da biblioteca do Apache XML e da própria biblioteca javax.xml da linguagem Java, o que fornece uma organização no modelo, conforme apresentado na Figura 5.3:

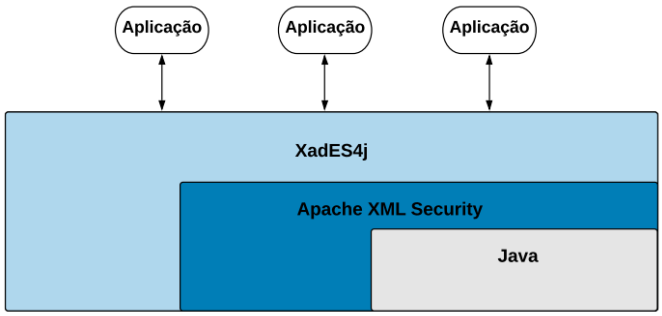


Figura 5.3: Visão geral da Arquitetura XAdES4j [12]

O diagrama de classe representando a forma correta de assinar um documento no formato XAdES é feito por meio de um conjunto de propriedades obrigatórias dentro de cada formato de XAdES. Por exemplo, no XAdES-C todos os dados de validação devem estar disponíveis, o que não ocorre no XAdES-BES. Portanto, existe uma classe abstrata

chamada XadesSignignProfile, onde as suas classes filhas vão definir as propriedades de cada tipo de assinatura XAdES. Para efetuar de fato a assinatura é utilizada a classe abstrata XadesSigner. Essa classe tem subclasses que utilizam as propriedades específicas e fazem a assinatura de acordo com o padrão estipulado. Esse modelo é demonstrado no diagrama 5.4 [12].

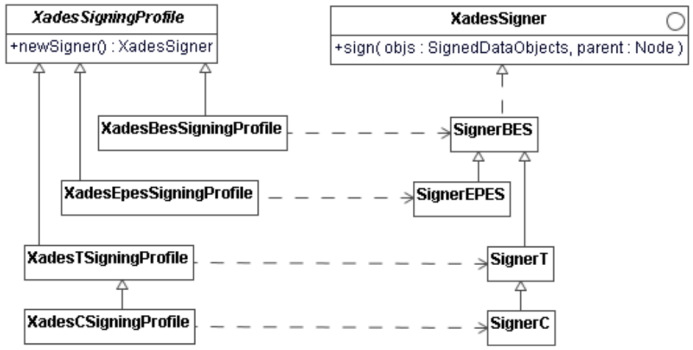


Figura 5.4: Diagrama de Classe XaDES4j [12]

Para ser feita a assinatura digital no padrão XAdES utilizando a biblioteca XAdES4j é necessário primeiro que seja realizada a leitura do certificado A1 ou A3 e, após isso, ser passado todos os diplomas em formato XML a serem assinados em formato zip. Após a leitura são verificados os elementos do diploma a ser assinado, observando se existe o elemento principal <Diploma>. Depois é feita uma assinatura e salvo no banco de dados cada um dos diplomas assinados. Essa biblioteca fornece uma maneira simples para fazer assinatura no padrão XAdES-T, que é o tipo de assinatura que será desenvolvida neste trabalho. O diagrama 5.5 apresenta o fluxo completo da assinatura digital, desde o armazenamento até salvar e disponibilizar o diploma assinado no banco de dados.

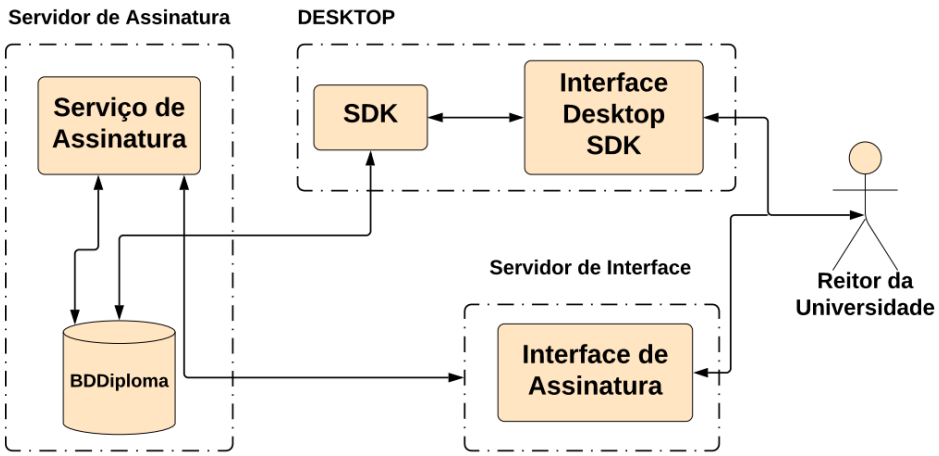


Figura 5.5: Componentes da assinatura digital

É possível observar no diagrama 5.5 da assinatura digital que antes de ser feita a assinatura digital é necessário que seja efetuado o upload do certificado (tanto A1 quanto A3). Isso é conseguido utilizando um SDK, instalado no desktop, que apresenta uma interface web local. É executado o login via LDAP e posteriormente é possível ser feito o upload do arquivo. No caso do certificado A3 é necessário que se tenha as dlls para leitura do certificado local. Apenas o reitor da UnB pode assinar os certificados, portanto apenas este IP tem permissão de acesso para o sistema e para salvar o certificado. Tanto o SDK quanto o serviço de assinatura não devem ter acesso à internet, o sistema deve ficar em servidores sem acesso a internet e apenas com acesso a IPs específicos via intranet. Toda a comunicação deve ser criptografada via ssl de ponta a ponta. Esse modelo garante que seja realmente seguro assinar um lote de diploma digital.

Os diplomas armazenados no banco de dados ficam acessíveis apenas para visualização por outros sistemas que disponibilizem, de maneira online, o diploma assinado de maneira confiável e segura. Os certificados que estão salvos no banco de dados só podem ser assinados utilizando a senha do proprietário do certificado. Toda vez que for assinado um lote de diplomas é necessário que seja passada a senha do certificado que irá assinar o diploma. Todo este processo acontece sem acesso à internet, o que garante segurança externa do modelo proposto e, mesmo que o certificado esteja armazenado no banco é necessária a senha para sua assinatura. Dessa forma, em nenhum momento, o certificado passa por um fluxo com acesso a internet.

O fluxo principal para a assinatura digital para os diplomas e documentos da UnB é apresentado na Figura 5.6:

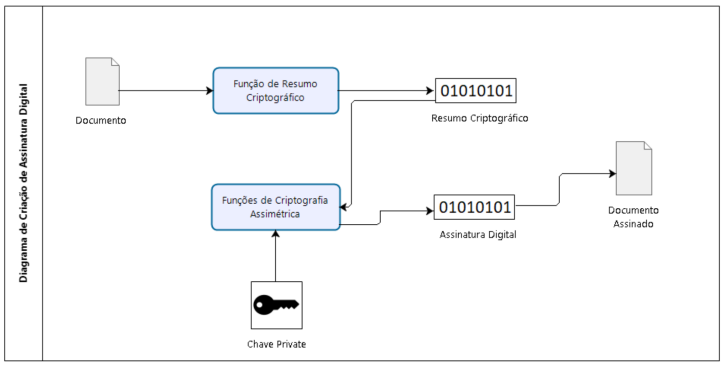


Figura 5.6: Fluxo de Assinatura Digital

Conforme apresentado no diagrama 5.6, o documento passa por uma função que vai gerar o resumo do documento, ou seja o seu hash. Após a geração do hash é utilizada a chave privada para realizar a assinatura assimétrica do documento a ser assinado, dessa forma, é gerado o documento com o hash e a assinatura digital feita nesse documento.

Esse é um processo simples onde a segurança do diploma assinado em formato digital é garantida com o uso de uma função de criptografia segura, no caso o algoritmo RSA e a chave privada sendo segura e confiável. Com base no modelo apresentado na Figura 5.6, foi criada uma arquitetura base para o desenvolvimento do sistema de assinatura. Essa arquitetura deve ser aderente ao modelo de assinatura e ao padrão de arquitetura atualmente utilizado pela STI/UnB.

O módulo de assinatura digital é apenas mais um serviço em java que é fornecido com intermédio do barramento de serviços Erlangms da STI/UnB. Esse serviço deve ser isolado e seu único ponto de acesso deve ser pela URL de assinatura utilizando o método POST. O barramento faz a intermediação com o método de acesso para iniciar o fluxo de assinatura dentro do módulo de assinatura desenvolvido em Java. O sistema de assinatura digital deve ser independente e ter a capacidade de se integrar de maneira fácil por meio do modelo principal de assinatura. É disponibilizada uma interface pública de comunicação para fácil integração dessa solução com qualquer outro sistema que venha a consumir esse serviço. A Figura 5.7 apresenta o resumo da atual arquitetura dos serviços de autenticação propostos por este trabalho e como é feita a comunicação e a integração entre o módulo em backend e o barramento.

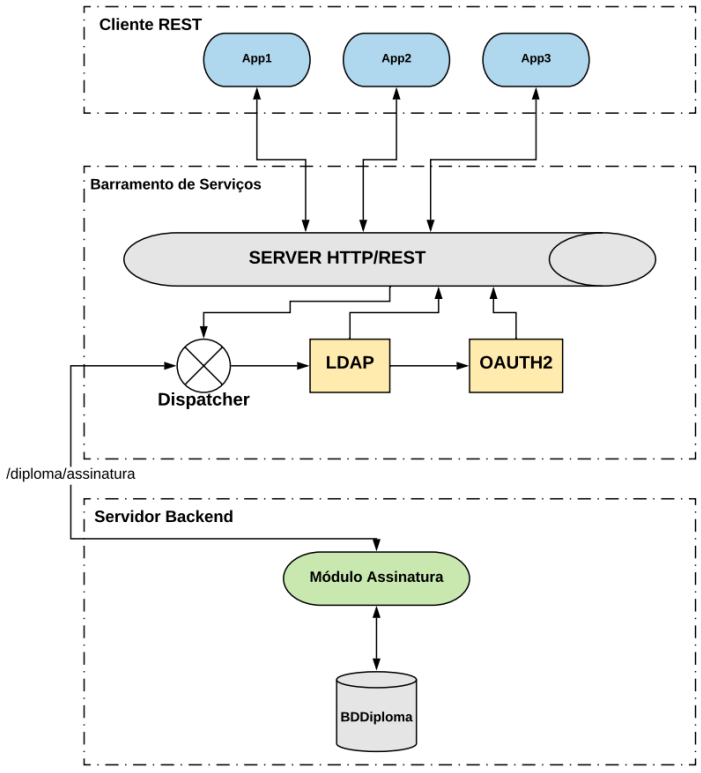


Figura 5.7: Diagrama Barramento de Serviços

Na Figura 5.7 são apresentadas as aplicações que se comunicam via REST para se autenticar utilizando os protocolos LDAP e Oauth2. O serviço de assinatura deve ser

um módulo Java que deve estar em um servidor sem acesso a internet. Esse módulo é utilizado pelo barramento da STI/UnB como intermediador de acesso ao serviço de assinatura. O serviço de assinatura digital possui três módulos básicos. Na Figura 5.7 é apresentado o fluxo de comunicação desde uma aplicação frontend até o módulo de assinatura e o banco de dados para armazenamento dos diplomas e certificados. Para mostrar a arquitetura interna do serviço de assinatura digital é apresentado o diagrama de classe feito – utilizando a notação UML – com as principais classes do sistema de assinatura digital no modelo 5.8.

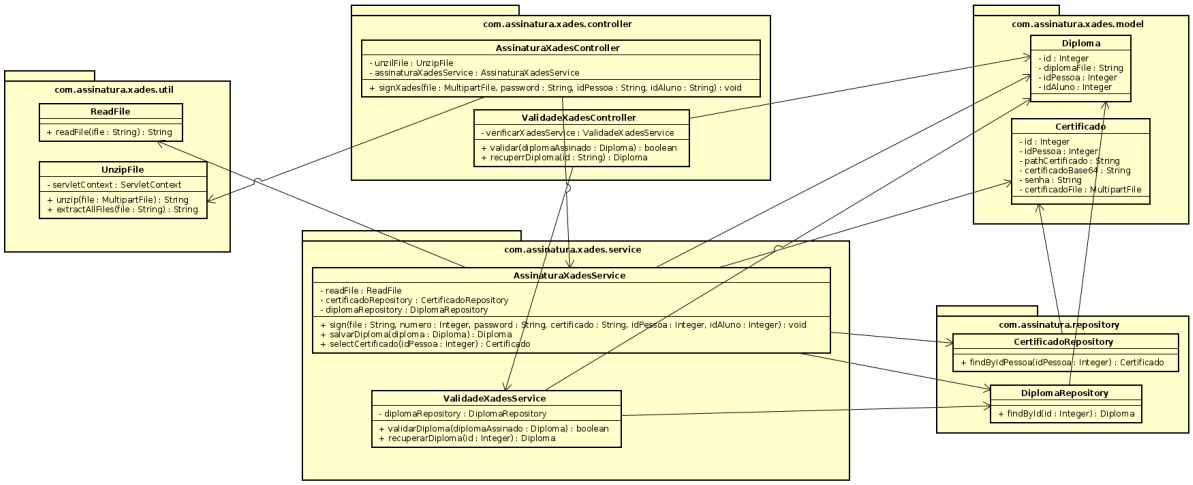


Figura 5.8: Diagrama de Classe do Modelo de Assinatura

No diagrama de classe 5.8 é possível observar a estrutura de pacotes, classes e suas interações. É necessário demonstrar o que cada classe faz e sua importância nesse modelo:

- **AssinaturaXadesController:** É a classe inicial que tem a interface de comunicação com o barramento Erlangms. O método signXades é o método principal para a assinatura no padrão XAdES. Como parâmetro para isso são passados os diplomas compactados em formato zip, a senha do certificado e o idPessoa para buscar o certificado.
- **ValidadeXadesController:** É a classe inicial que faz a interface de comunicação para validade de um diploma assinado. O método principal é o método validar, que verifica se o diploma assinado está de acordo com o padrão do MEC, conforme os xsd para validação do diploma.
- **AssinaturaXadesService:** É a classe de serviço que tem de fato a função de assinar cada documento que está dentro do arquivo zip de assinatura de diplomas digitais. O principal método dessa classe é o método sign que faz a assinatura de um diploma para o padrão XAdES e dois métodos auxiliares, um para salvar o diploma

assinado no banco de dados e outro para buscar o certificado salvo pelo sdk no banco de dados para efetuar a assinatura de fato.

- **ValidadeXadesService:** É a classe de serviço que utiliza os arquivos xsds para validar o diploma assinado no padrão do MEC. O método validarDiploma é o método principal que informa se o diploma está válido ou não conforme o padrão especificado no xsd.
- **DiplomaRepository:** É a classe que salva o diploma assinado no banco de dados e relaciona o diploma salvo com o idPessoa que o assinou e o idPessoa do discente do diploma.
- **CertificadoRepository:** Recupera o certificado a partir do idPessoa para ser usado na hora de assinar um lote de diplomas.
- **Certificado:** É uma entidade que tem todos os parâmetros do banco de dados. São todos os dados que serão recuperados na tabela de certificado.
- **Diploma:** É a entidade que estabelece todos os parâmetros para salvar um diploma. São todos os dados que serão salvos na tabela de diplomas

A estrutura de pacotes e a organização do projeto é apresentado na Figura 5.9:

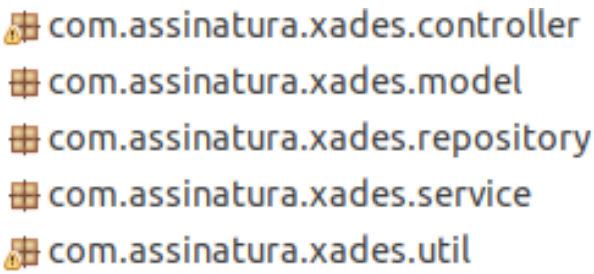


Figura 5.9: Estrutura de pacotes do modelo de assinatura

Como pode ser visto na estrutura de pacotes apresentado na Figura 5.9, o sistema foi dividido em uma arquitetura em camadas para facilitar futuras evoluções e manutenções de sistemas. Cada um dos pacotes criados tem o objetivo de agrupar classes que tenham a mesma função semântica dentro do modelo proposto neste trabalho. Os pacotes estão divididos da seguinte forma:

- **com.assinatura.xades.controller:** É o pacote onde ficam as classes principais de comunicação com o barramento de serviços. Nela se encontram as interfaces de comunicação e os métodos para acesso dos serviços.

- **com.assinatura.xades.service:** É o pacote onde ficam as classes com a lógica de negócio da aplicação de assinatura digital. Nessas classes tem-se a comunicação entre as classes de interface de comunicação e as classes de infraestrutura, que fazem as operações no banco de dados. Dentro dessas classes estão os métodos com as regras de negócio do sistema.
- **com.assinatura.xades.repository:** É o pacote que reúne as classes que fazem as operações no banco de dados. Essas classes se comunicam com as classes de serviço e com as classes de entidade. Dentro delas existem os métodos para as operações necessárias no banco de dados para salvar um diploma assinado e recuperar o certificado salvo pelo SDK.
- **com.assinatura.xades.model:** Esse pacote reúne todas as classes de modelo do sistema, ou seja, as classes que refletem as tabelas do banco de dados, também chamadas de entidades. Essas classes servem como modelo para serem inseridos os dados que serão salvos ou recuperados dentro do banco de dados da aplicação de assinatura digital.
- **com.assinatura.xades.util:** Esse pacote reúne todas as classes utilitárias para o sistema de assinatura. As classes como ReadFile, que faz a leitura de um arquivo, a classe UnzipFile, que descompacta um arquivo zip para a leitura dos diplomas pelo sistema. Nesse pacote serão colocadas classes que são de apoio para a estrutura principal do modelo proposto.

O diagrama do processo do modelo de arquitetura vai desde o frontend até salvar o diploma assinado em formato digital no padrão XAdES. O serviço que o barramento fornece para salvar o diploma é feito no modelo REST com o método POST. O serviço principal é disponibilizado pela uri/diploma/assinatura. No corpo da requisição REST deve ser passado o lote de diploma em formato zip, a senha do certificado e o idPessoa. Esse serviço é disponibilizado apenas na intranet, nenhum serviço de assinatura deve ser disponibilizado pela internet, apenas consultas a diplomas já assinados podem ser visualizados pela internet. Da mesma forma, nenhum comando que altere o estado do banco de dados deve estar disponível na internet.

Até o momento, este trabalho focou em como são as tags do sistema de assinatura digital, como esse módulo de execução de assinatura é inserido dentro da atual arquitetura da UnB e, durante o desenvolvimento do modelo, demonstrado o porquê do uso de um modelo em desktop para a criação do SDK e outro para a execução da assinatura digital no padrão XAdES de acordo com as diretrizes do MEC e da ICP-Brasil. O certificado digital para assinar os diplomas em formato digital é realizado via web. No caso do certificado A1, que é utilizado neste protótipo, a assinatura digital é feita através de uma

aplicação web frontend que envia o certificado para armazenamento no servidor, com isso fica facilitada a performance da assinatura digital no momento de sua efetivação. O formato padrão que deve ser aceito para armazenamento do certificado deve ser o formato .pfx.

Os certificados A3 são armazenados através de pendrives ou cartões de memória, que são inseridos nos computadores e só podem ser lidos localmente, não há possibilidade de serem lidos diretamente pelo browser, diferentemente do certificado A1. Assim, seria necessitaria uma solução em desktop para acesso, validação e armazenamento do certificado A3 no servidor para futuras assinaturas [90].

Para adição de um certificado do nível A3 no servidor para futuras assinaturas é necessário que seja feito localmente por um SDK que fará a leitura do token, a validação do token e o salvamento desse token no servidor específico para a assinatura de diplomas em formato digital. Esse módulo deve ter um componente desktop com o objetivo de validar e integrar o certificado ao servidor para que possam ser assinados futuramente os lotes de certificados através de uma aplicação frontend.

Para o SDK é necessário adicionar as dlls específicas de cada uma das certificadoras para que o módulo SDK possa ler esse certificado de maneira correta e armazená-lo no servidor para que seja efetuada de fato a assinatura dos lotes de documentos. Todos os certificados são adicionados via banco de dados, sendo apenas alguns IPS específicos que têm a permissão para salvar os certificados no servidor exclusivo para esse armazenamento. É necessária uma boa segurança para garantir que apenas pessoas específicas tenham acesso ao servidor e a tudo que for executado dentro dele, o que deve ser feito pelo usuário que estiver logado. O modelo geral da arquitetura, desde a leitura do certificado até a assinatura dos lotes de diplomas em formato digital, é apresentado na Figura 5.10:

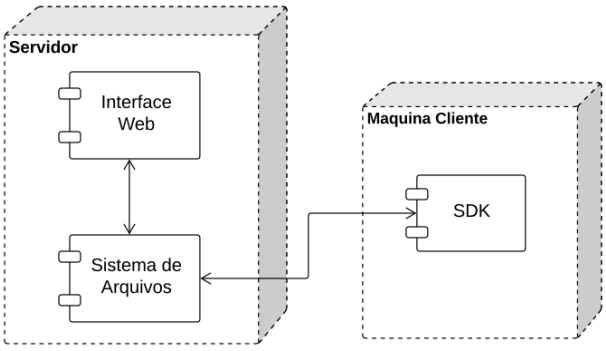


Figura 5.10: Diagrama de Componentes da Solução de Assinatura

5.6 Carimbo de Tempo

Uma propriedade fundamental para um diploma assinado digitalmente por um certificado válido, deve ser a de que sua validade continue mesmo depois que o certificado prescreva. De acordo com o MEC, um diploma digital deve ser válido a qualquer tempo. [7], [91].

O carimbo de tempo ou *time-stamp* é um elemento que deve estar presente dentro de uma assinatura XAdES para garantir que o diploma tenha sido assinado em uma determinada data, isso assegura a validade a qualquer tempo e o não repúdio em caso de comprometimento de uma chave [91]. Com o objetivo de prolongar a validade de um diploma assinado no padrão XAdES deve ser aplicado sobre a assinatura um carimbo de tempo. O carimbo de tempo deve ter referência da assinatura para provar que a assinatura foi feita antes do carimbo de tempo [91].

A Figura 5.11 apresenta o xsd do time-stamp. O elemento HashDataInfo contém um uriattribute referenciando um objeto de dados e um elemento ds:Transform que indica as transformações a serem feitas neste objeto de dados. Esse HashDataInfo será usado para produzir a entrada do processo para geração do hash com o resultado incluído no carimbo de tempo [91].

```
<xsd:complexType name="TimeStampType">
  <xsd:sequence>
    <xsd:element name="HashDataInfo" type="HashDataInfoType"
     maxOccurs="unbounded"/>
    <xsd:choice>
      <xsd:element name="EncapsulatedTimeStamp"
       type="EncapsulatedPKIDataType"/>
      <xsd:element name="XMLTimeStamp" type="AnyType"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="HashDataInfoType">
  <xsd:sequence>
    <xsd:element name="Transforms" type="ds:TransformsType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="uri" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

Figura 5.11: XSD do TimeStamp

5.7 Síntese do Capítulo

Este capítulo teve por objetivo apresentar um resumo da arquitetura de autenticação, autorização e segurança da UnB, mostrando como as aplicações frontend se comunicam com os serviços de segurança presentes no barramento e com os serviços que estão presentes no backend. Os serviços se comunicam por meio do protocolo REST. Atualmente, a UnB dispõe dos serviços de autenticação e autorização com os protocolos LDAP e Oauth2. Todas as comunicações das aplicações da UnB são feitas com a utilização desses

protocolos. Após a definição da arquitetura da UnB foi necessário definir os principais requisitos funcionais e não funcionais exigidos pelo MEC e pela ICP-Brasil. Os requisitos foram definidos com base nas portarias números 330 e número 554 e na Nota Técnica No. 13/2019/DIFES/SESU/SESU. Todos esses normativos definem como deve ser o sistema de assinatura digital no padrão XAdES. Por último, foi definida a arquitetura interna do sistema de assinatura digital desenvolvida na linguagem Java compatível com a atual arquitetura da UnB. A arquitetura utiliza todos os recursos da linguagem Java e da biblioteca XAdES4j, que dispõe de funcionalidades para a assinatura digital no padrão XAdES. O sistema deve ser de fácil manutenção, evolução e integração com qualquer outra aplicação que se comunique através do modelo de serviços.

Capítulo 6

Testes, Comparativos e Políticas de Segurança dos Diplomas Digitais

Este capítulo objetivou analisar comparativamente as diferentes soluções abordadas neste trabalho para assinatura digital e o atual sistema utilizado pela UnB. Foram analisados a segurança e o desempenho dos modelos para comprovar aquele que fornece melhores resultados. Foram feitos os testes baseados nos principais problemas de segurança presentes nos sistemas em geral documentados pela organização Projeto Aberto de Segurança em Aplicações Web (OWASP). A análise deve ser focada nos seguintes princípios: confiabilidade, integridade, disponibilidade, autenticação e autorização. Foi verificado qual dos sistemas tem maior compatibilidade com os normativos do MEC e da ICP-Brasil e qual das soluções tem maior possibilidade de evoluções futuras.

Foram feitos testes estáticos utilizando o SonarQube [92] para verificar possíveis defeitos existentes nas soluções e averiguar a cobertura de testes nos modelos. Com isso foi possível analisar quais dos sistemas tem maior probabilidade de conter uma possível falha crítica no código. Para testes dinâmicos foi utilizado o software OWASP ZAP. Esse sistema tem por objetivo identificar cada um dos principais defeitos de segurança relatadas pela OWASP [93], assim como averiguar se nas soluções comparadas existem algumas desses defeitos, além de analisar a gravidade de cada um deles. Com isso pôde-se fazer uma tabela comparativa para examinar possíveis defeitos encontrados em cada uma das soluções.

No decorrer do capítulo foi demonstrado como foi feita a segurança interna e externa dos diplomas digitais assinados, pois é de considerável importância demonstrar a política de segurança e como esta será mantida para que os diplomas não sejam alterados tanto externa quando internamente. Foi apresentado no capítulo 2 os principais trabalhos realizados até agora pelas Universidades Federais que envolvem o desenvolvimento e adaptação de sistemas de assinatura de diplomas digitais para os padrões do MEC e da ICP-Brasil.

Por último, foi realizado um comparativo entre os normativos e as soluções propostas pelo Governo Federal Brasileiro e as soluções e normativos da União Europeia com relação a assinatura de documentos e diplomas em formato digital. Essas soluções foram comparadas com as técnicas mais recentes e utilizadas atualmente para assinatura e segurança digital, possibilitando demonstrar as falhas do modelo e das tecnologias que o Governo Federal Brasileiro tem adotado.

6.1 Testes de Segurança e Desempenho

Anterior a definição do modelo sugerido neste estudo, foi utilizado uma abordagem para verificar a existência de erros de segurança nos próprios requisitos. Essa estratégia garante uma futura solução mais adequada e maior segurança, pois os erros críticos de segurança já são tratados no começo do desenvolvimento [94].

Após a definição dos requisitos básicos e a verificação de sua segurança foram realizados três testes. O primeiro foi um teste estático, importante para ajudar na descoberta de vulnerabilidades da segurança em aplicações e serviços. A análise estática auxilia na identificação de brechas de segurança e qualidade do código, que muitas das vezes não são considerados durante o desenvolvimento de software [95]. Esse primeiro teste foi feito no atual sistema de assinatura digital da UnB, o C3Web, comparando-o com o protótipo desenvolvido internamente no barramento de serviços Erlangms e outro protótipo utilizando a linguagem Java com a biblioteca XAdES4j.

Além dos testes estáticos foram necessários outros tipos de testes, como os testes de segurança em tempo de execução. Uma das ferramentas mais recomendadas e importantes para realizar esse teste é o OWASP ZAP. Essa ferramenta faz testes de acordo com as principais vulnerabilidades de segurança de software documentadas pela OWASP [96]. É uma ferramenta de fácil utilização para descoberta de possíveis vulnerabilidades. Sendo assim, torna-se importante comparar as soluções em tempo de execução [97].

Foi relevante também verificar se as soluções tiveram a capacidade de suprir a demanda de maneira eficiente, ou seja, se foi possível assinar a quantidade de diplomas pretendidos no tempo esperado. A demanda de assinatura da UnB é de cerca de 200 a 500 diplomas por lote de assinatura, com um total de 2 a 5 mil diplomas assinados por semestre. É necessário que as soluções sejam seguras contra o ataque de *distributed-denial-of-service* (DDoS) e para isso será utilizado o Loctus. Essa aplicação é uma aplicação para testes funcionais voltados para mensurar a carga e a performance de uma aplicação [98].

6.2 Resultados Comparativos de Segurança e Desempenho

6.2.1 Testes de Segurança

Com a utilização do SonarQube foi realizada a análise do código do projeto C3Web atualmente utilizado pela UnB. Foi constatado que esse sistema possui uma quantidade de 5 bugs, com nenhuma vulnerabilidade e nenhum code smell, conforme pode ser visualizado na Figura 6.1. Assim, pode-se concluir esse sistema sistema não possui problemas de segurança, de acordo com a análise estática de código.

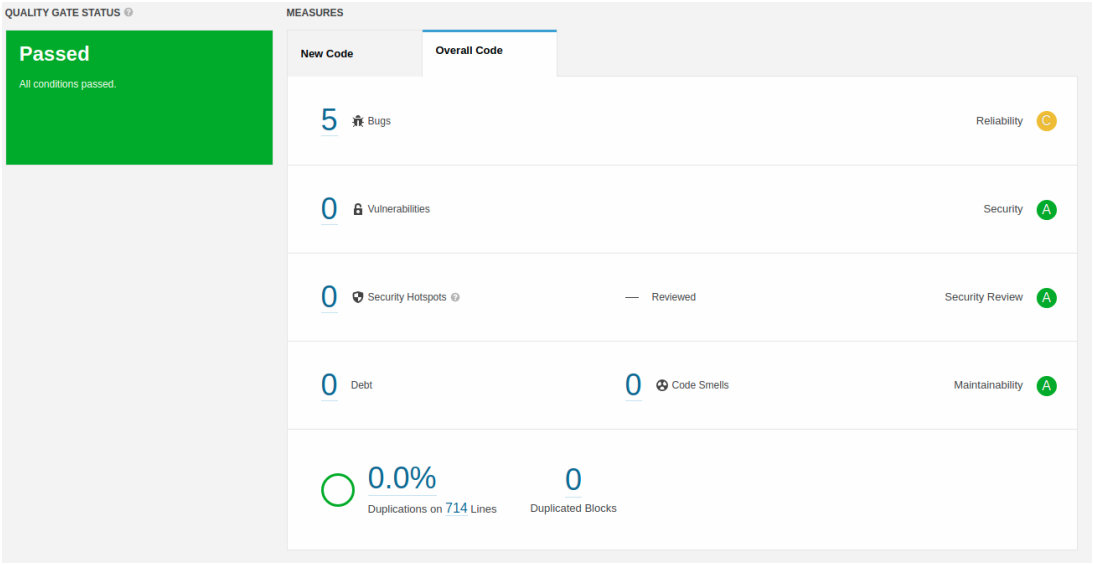


Figura 6.1: Resultado dos testes estáticos do Sistema C3Web

Apesar de a análise do sistema C3Web demonstrar 5 potenciais bugs que poderiam ocasionar algum problema segundo a ferramenta SonarQube, esses bugs não têm uma vulnerabilidade alta na aplicação, o que garante que o sistema C3Web, apesar de ser um sistema que utiliza uma biblioteca proprietária para executar a assinatura, é seguro e cofiável para assinaturas no padrão XAdES-BES. Até o momento, este sistema não utiliza o XAdES-T, que é o padrão recomentado pelo MEC.

Os resultados do SonarQube demonstraram que a prova de conceito desenvolvida em Erlang passou nas análises estáticas e que não foi encontrado nenhum problema relacionado a segurança. Foi feito o teste no módulo de assinatura digital e o resultado é apresentado na Figura 6.2.

No teste estático 6.2 é possível notar que a análise estática do código fonte do modelo proposto em Erlang demonstrou igualmente um total de 5 bugs encontrados que sugerem falhas que poderiam ocasionar vulnerabilidades na aplicação. Outro problema encontrado

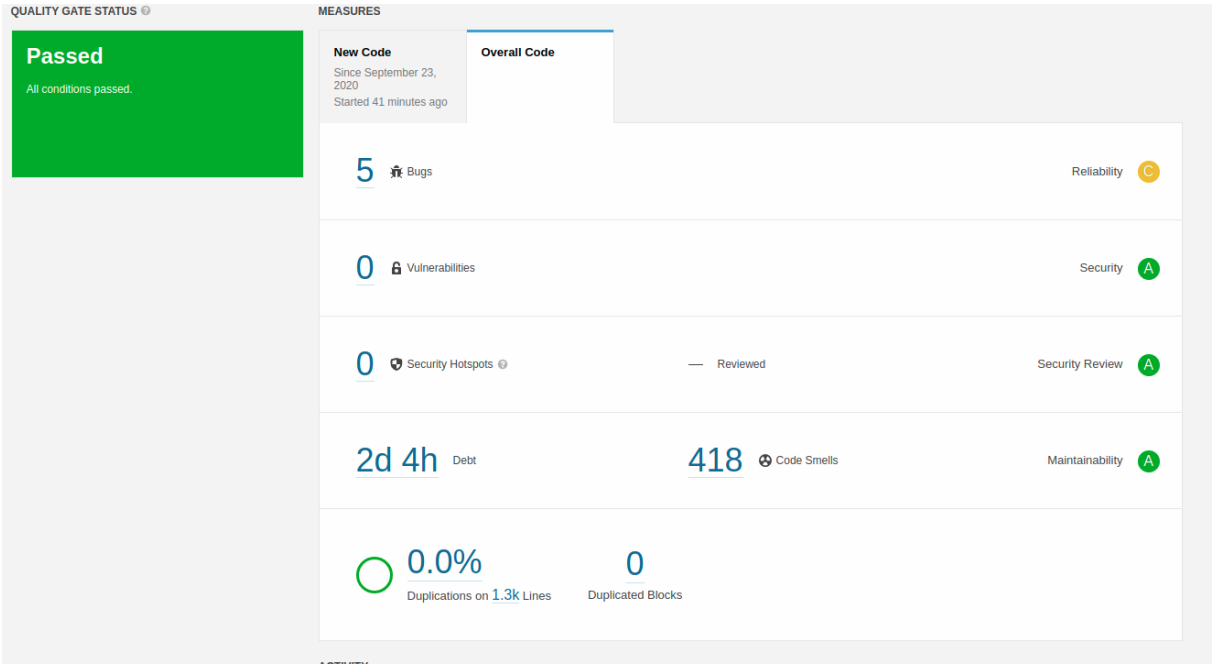


Figura 6.2: Resultado dos testes estáticos do modelo usando Erlang

no relatório do SonarQube consiste em possuir 418 possíveis códigos que podem significar algum erro ou código mal feito dentro da aplicação. Observou-se que as duas soluções apresentaram uma mesma quantidade de bugs, porém o C3Web não tem nenhum code smell, o que faz com que seja uma solução mais segura.

Outra deficiência demonstrada pelo teste estático 6.2 do SonarQube é a de que existe uma grande quantidade de débitos técnicos no protótipo desenvolvido em Erlang. Assim, esse código fica com uma difícil manutenção pelo motivo relatado, o que faz com que essa solução apresentada seja verbosa e, além disso, passível de erros. Nessa solução não exitem bibliotecas que facilitem a assinatura no padrão XAdES.

O último projeto que foi analisado foi o projeto proposto neste trabalho. Foi utilizada a linguagem Java a biblioteca XAdES4j para implementar o sistema de assinatura de lotes de diplomas, de acordo com o formato do MEC e da ICP-Brasil. O teste estático foi feito com o SonarQube nos mesmos moldes das outras duas soluções apresentadas. O resultado pode ser verificado na Figura 6.3.

O que pode ser evidenciado nesse teste é que o modelo proposto e o protótipo desenvolvido utilizando a biblioteca XAdES4j comprovou a viabilidade da criação de uma solução simples, minimalista e sem nenhum bug, vulnerabilidade ou duplicidade de código. O que garante que, estaticamente, o modelo apresentado tem a mesma segurança da atual solução utilizada pela UnB, com uma menor quantidade de bugs e, diferente do modelo em Erlang, não possui nenhum código duplicado.

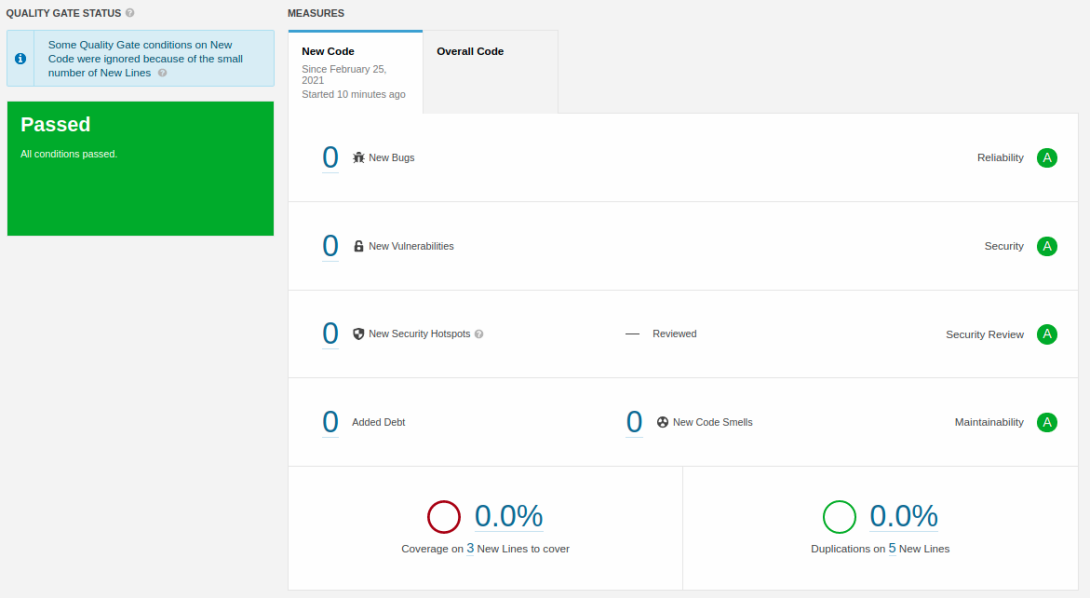


Figura 6.3: Resultado dos testes estáticos da solução proposta

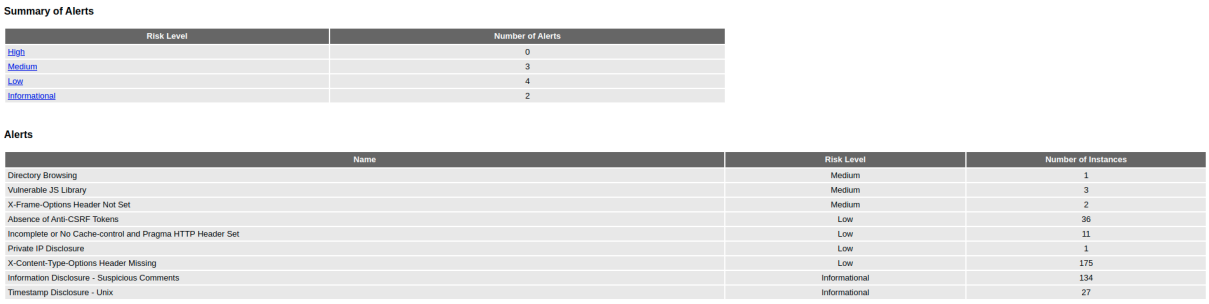
Outra vantagem desse modelo é que é todo composto por microsserviços que se integram para assinar um diploma digital conforme determina o MEC e a ICP-Brasil. Pode ser notado que possui uma quantidade menor de linhas e código e, ao mesmo tempo, uma menor complexidade para manutenções futuras, haja vista os microsserviços terem uma complexidade menor e, se for preciso futuramente a inclusão de novos serviços, será feita de maneira independente a não impactar no serviço principal de assinatura, como acontece com o atual sistema utilizado pelo UnB, o C3Web, o qual é um projeto único.

Para validar a segurança foi necessário, além dos testes estáticos, efetuar testes dinâmicos de segurança. Um dos padrões mais utilizados para validar a segurança dos serviços é da OWASP. A OWASP é uma organização sem fins lucrativos com o objetivo de melhorar a segurança dos softwares. A principal ferramenta que utiliza as definições das principais vulnerabilidades de segurança documentadas pela OWASP é o software OWASP ZAP. Essa ferramenta é considerada uma ótima solução para verificação de problemas em aplicações web e serviços [97]. Esse software faz uma varredura em tempo de execução para verificar se o software é seguro ou não.

Após o comparativo realizado com o código estático utilizando o SonarQube, foi feito o teste dinâmico de segurança das três soluções utilizando a ferramenta a OWASP ZAP [97]. Com isso foi possível verificar em tempo de execução se as aplicações tinham os padrões de segurança aceitáveis para assinatura de diplomas em formato digital [99].

O atual sistema utilizado pela UnB, o C3Web, faz a assinatura de diplomas via desktop e tem um portal de armazenamento de certificados, diplomas e padrões para assinatura e manutenção dos sistemas. Os testes de segurança foram realizados neste portal de

administração para verificar a segurança do armazenamento dos diplomas, certificados e da assinatura. Os resultados são apresentados na Figura 6.4.



Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	4
Informational	2

Alerts

Name	Risk Level	Number of Instances
Directory Browsing	Medium	1
Vulnerable JS Library	Medium	3
X-Frame-Options Header Not Set	Medium	2
Absence of Anti-CSRF Tokens	Low	36
Incomplete or No Cache-control and Pragma HTTP Header Set	Low	11
Private IP Disclosure	Low	1
X-Content-Type-Options Header Missing	Low	175
Information Disclosure - Suspicious Comments	Informational	134
Timestamp Disclosure - Unix	Informational	27

Figura 6.4: Teste de segurança com o sistema C3Web

Os testes realizados no sistema C3Web demonstraram os seguintes erros:

- **Directory Browsing:** Essa vulnerabilidade mostra que é possível acessar os diretórios a partir do browser, ou seja, tem uma árvore de diretórios visíveis pelo browser, o que pode significar uma vulnerabilidade de segurança de nível médio na aplicação.
- **Vulnerable JS Library:** Essa falha é ressaltada quando é utilizado uma ou mais bibliotecas que tenham alguma vulnerabilidade e que foram relatadas nas versões dessas bibliotecas. Essas vulnerabilidades podem identificar uma falha de segurança na aplicação.
- **X-Frame Options Header Not Set:** Essa vulnerabilidade significa que teve algum cabeçalho na requisição Options ausente, o que pode significar que esse sistema está vulnerável a um ataque de clickjacking. Esse tipo de ataque engana o usuário fazendo-o clicar em elementos invisíveis da página da web disfarçado de outro elemento, o que pode fazer com que o usuário baixe malwares indevidos.
- **Absence of Anti-CSRF Tokens:** Esse tipo de vulnerabilidade está relacionada a falta de um token anti-CSRF. Esse token evita ataques de falsificação de solicitação entre sites. Esse tipo de vulnerabilidade faz com que o atacante possa redirecionar o cliente para um outro sistema, que pode ser um sistema malicioso.
- **Incomplete or No Cache-control and Pragma HTTP Header Set:** O cache-control e o cabeçalho HTTP pragma não foram configurados corretamente ou estão ausentes, permitindo que os navegadores e os proxies armazenem conteúdo em cache. É recomendado pela OWASP que, sempre que possível, as requisições HTTP sejam enviadas com o cabeçalho no-cache, no-store ou must-relevant.

- **Private IP Disclosure:** Essa vulnerabilidade foi encontrada no corpo de retorno da resposta HTTP em um IP privado. Essa vulnerabilidade pode possibilitar ataques a este IP exposto.
- **X-Content-Type-Options Header Missing:** Esse erro significa que não é retornado o cabeçalho completo do OPTIONS, sendo feito apenas o retorno do cabeçalho completo do método GET.
- **Information Disclosure - Suspicious Comments:** A resposta de algumas requisições feitas retornam algum comentário que podem ajudar algum invasor a perceber alguma vulnerabilidade exposta por esses comentários.
- **Timestamp Disclosure - Unix:** Um carimbo de data/hora divulgado pelo servidor que pode ser utilizado pelo invasor para se obter outras informações confidenciais, como por exemplo, um salt ou um tolen durante a autenticação ou criptografia.

O segundo teste de segurança em tempo de execução foi realizado utilizando um protótipo desenvolvido dentro do barramento da UnB. A linguagem utilizada para o desenvolvimento foi a Erlang, que é a mesma do barramento. Pode-se notar que existe uma quantidade de falhas menores que no sistema C3Web. O barramento de serviços apresenta um problema com a liberação geral de acesso, ou seja, não tem restrições, o que é uma maneira performática porém, pouco segura, sendo impossível limitar os IPs dos seus serviços interno o que impediria o não acesso via internet ao serviço de assinatura. Os resultados desse teste, podem ser visualizados na Figura 6.5.

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	0
Informational	2

Alerts

Name	Risk Level	Number of Instances
Cross-Domain Misconfiguration	Medium	5
Content-Type Header Missing	Informational	2
Timestamp Disclosure - Unix	Informational	5

Figura 6.5: Teste de segurança no protótipo desenvolvido em Erlang

Nos testes de segurança realizados com a solução em Erlang foi evidenciado um erro com complexidade média e dois erros com complexidade apenas de informação. Os erros encontrados foram:

- **Cross-Domain Misconfiguration:** Este problema significa que não existe um controle de IPs ou outra autenticação para identificar quais clientes podem acessar o servidor do barramento. Isso é um problema complicado, porque o barramento de serviços deve ser liberado para qualquer cliente que queira acessar e, ao mesmo tempo, o processo de assinatura de diplomas digitais deve ser limitado para alguns IPs específicos e, para garantir uma maior segurança, nenhum dos serviços construídos deve ser aberto para internet, apenas para as máquinas dos usuários que têm permissão para assinar um documento. Esse problema não pode ser resolvido utilizando o barramento, porque todos os serviços do barramento devem ter acesso à internet.
- **Content-Type Header Missing:** O modelo em Erlang, em alguns momentos, retorna o conteúdo e não retorna o cabeçalho associado. Isso não é visto como um erro grave e não impacta em nenhum problema de segurança.
- **Timestamp Disclosure - Unix:** O barramento de serviços divulga o carimbo de tempo no momento que o serviço termina. Essa é apenas uma informação repassada pelo barramento, o que não caracteriza uma falha de segurança.

Ao verificar os testes, o principal problema de segurança relatado foi a demonstração de que o barramento não é um bom local para o desenvolvimento de um sistema de assinatura digital. A linguagem Erlang, não tem bibliotecas que facilitem a construção de um serviço de assinatura, já que acarretaria muito trabalho e dificultaria a manutenção e a criação de uma solução agnóstica. Além disso, o barramento é um local onde todos os serviços devem ter acesso à internet. A solução de assinatura digital proposta neste projeto não deve, em nenhum momento, disponibilizar qualquer informação via internet, o que irá garantir uma alta segurança em todo o processo de assinatura.

Por último foi realizado o teste no protótipo desenvolvido neste trabalho. O que é possível verificar na figura 6.6.

O resultado dos testes realizados com o OWASP ZAP executados no protótipo desenvolvido para validar o modelo sugerido neste trabalho, demonstrou que não foi encontrado nenhum risco alto, houve apenas um risco de nível intermediário que não tem impacto relevante na segurança do modelo proposto. As falhas encontradas foram:

- **Estouro de Buffer:** Em algumas situações de envio de dados em grande quantidade foi verificado o estouro de buffer. Esse problema não irá ocorrer quando se fizer um controle de quantidade de dados com um balanceador, que possa limitar a quantidade de dados. Não é considerado um problema grave e tem uma solução simples ao ser desenvolvido e implantado no modelo apresentado neste trabalho.

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	3
Informational	0

Alerts

Name	Risk Level	Number of Instances
Estouro de Buffer	Medium	3
Application Error Disclosure	Low	2
Information Disclosure - Debug Error Messages	Low	2
X-Content-Type-Options Header Missing	Low	1

Figura 6.6: Teste de segurança realizado no protótipo desenvolvido neste projeto

- **Application Error Disclosure:** Esse erro ocorre quando alguma exceção não é tratada na aplicação. No protótipo os erros estão sendo apenas jogados e não tratados, o que justifica esse alerta, porém não gera um risco real para o modelo, assim, não é relevante.
- **Information Disclosure - Debug Error Messages:** Esse erro significa que algumas informações e debug estão indo no retorno do serviço. Em algumas situações, quando a chamada for errada, o protótipo não tem um tratamento de erro,porém, é um erro que não gera nenhuma falha de segurança.
- **X-Content-Type-Options Header Missing:** Esse microsserviço não retorna o cabeçalho completo do OPTIONS, sendo feito apenas o retorno do cabeçalho completo do método GET. Esse erro é semelhante ao encontrado no sistema utilizado atualmente pela UnB.

6.2.2 Testes de Desempenho

Uma das principais preocupações da UnB é a assinatura de diplomas em formato digital de maneira correta, segura e em um tempo adequado. O atual sistema utilizado pela UnB tem um tempo razoável de assinatura mas, frequentemente, ocorrem falhas de assinatura em determinados documentos, o que faz com que documentos que tiveram falhas na assinatura sejam assinados novamente. Além disso, está totalmente dependente das bibliotecas do e-Sec, dificultando a manutenção ou adaptação ou, até mesmo, correções de possíveis erros.

Nesta etapa foi realizado um comparativo de desempenho entre o atual sistema da UnB, o C3Web, uma solução desenvolvida em Erlang, e com o modelo proposto por este trabalho, desenvolvido com a linguagem Java. Foi utilizado o software de testes de

desempenho locust. Essa ferramenta de teste tem um desempenho aceitável, um tempo de resposta considerado bom e também uma interface fácil e resumida para escalonamento de usuários, requisições e geração de gráficos para verificar o desempenho dos serviços analisados [100].

O primeiro teste foi realizado no sistema C3Web, posteriormente no protótipo desenvolvido em Erlang e por fim no protótipo utilizando a linguagem Java e a biblioteca XAdES4j.

A biblioteca XAdES4j tem a facilidade de assinar um documento utilizando o padrão XAdES-T, que é um dos padrões recomendados pelo MEC e pela ICP-Brasil como seguros para um documento assinado. O teste foi realizado com um lote de diplomas com 200 certificados empacotados em um arquivo em formato zip.

Na Figura 6.7 é demonstrado o comparativo de tempo de assinatura de um lote de 200 diplomas no padrão XAdES dos três diferentes modelos apresentado: o atual sistema utilizado C3Web, o protótipo desenvolvido no barramento em Erlang o protótipo desenvolvido em Java.

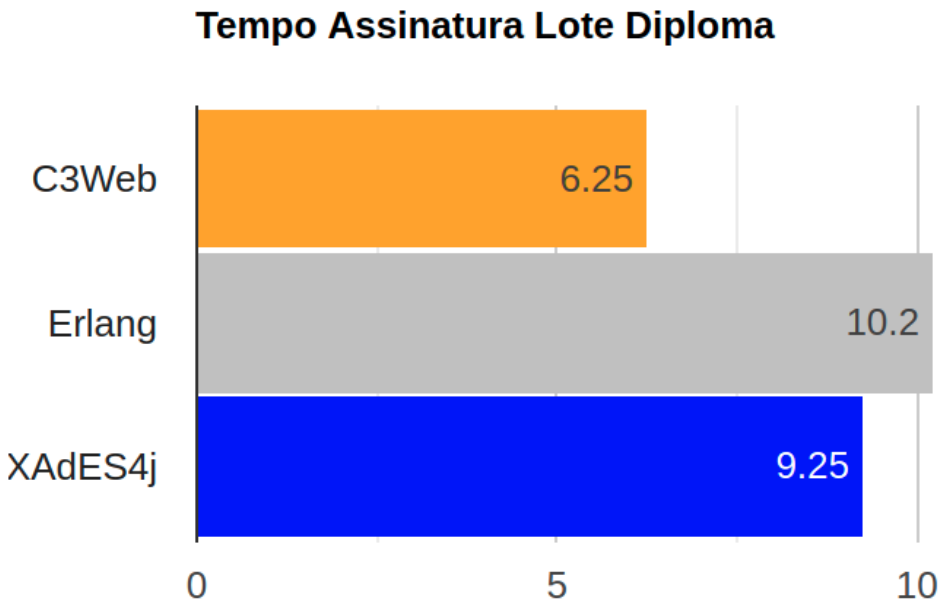


Figura 6.7: Comparativo do tempo gasto para realizar a assinatura em cada solução

É possível visualizar na Figura 6.7 que o desempenho em minutos para assinatura de 200 diplomas do atual sistema de assinatura utilizado pela UnB, o C3Web, foi de 6 minutos e 46 segundos, o protótipo de assinatura em Erlang demorou 10 minutos e 20 segundos para realizar a mesma tarefa e o protótipo proposto por este trabalho desenvolvido em Java e com a biblioteca XAdES4j tem o desempenho total de 9 minutos e 25 segundos para assinatura de 200 diplomas. A solução em Erlang tem um limite de assinatura de no máximo 3683 diplomas. Acima desse total, o sistema fica travado e não mais

realiza a assinatura dos diplomas. Pode-se notar que o sistema atual da UnB tem uma performance ligeiramente menor, porém não se configura em um comparativo real porque esse sistema faz a assinatura XAdES-BES sem o carimbo de tempo. Já o sistema de assinatura desenvolvido neste trabalho faz a assinatura dos diplomas no padrão XAdES-T com carimbo de tempo e no padrão atual do MEC e da ICP-Brasil, o que mostra que são protocolos diferentes, que têm características diferentes, consequentemente tempos diferentes de assinatura.

É importante também observar que o tempo de assinatura de ambas as soluções estudadas, utilizando o XAdES4j em Java e a solução utilizando o Erlang dentro do barramento ,ambas com a assinatura no padrão do MEC e da ICP-Brasil e no padrão XAdES-T, tiveram um tempo muito próximo de assinatura. Esse resultado permitiu concluir que o padrão XAdES-T tem uma limitação de tempo independente da solução utilizada. Assim, uma assinatura com carimbo de tempo vai ter um tempo diferente de assinatura do documento assinado no padrão XAdES-BES. Além disso, com esta assinatura é garantido o não repúdio, ou seja, um usuário não pode dizer que não assinou um diploma [101].

Os testes foram todos realizados em um ambiente real da UnB para que fosse possível verificar a performance genuína de cada uma das soluções testadas. Todas as máquinas possuem 4 núcleos de processamento, 4 gigas de memória e 120 gigas de HD. Após os testes realizados foi possível verificar que os sistemas que fazem a assinatura utilizando o padrão XAdES-T têm um desempenho semelhante, mas inferior ao sistema utilizando o padrão XAdES-BES que demonstrou uma performance melhor, porém sem a utilização do carimbo de tempo. O que se pode notar é que o tempo depende mais do protocolo utilizado do que da tecnologia adotada para executar a assinatura digital. O padrão do MEC exige pelo menos o XAdES-T como o mínimo para segurança de um diploma assinado digitalmente.

6.3 Diretrizes de Segurança de Armazenamento de Longo Prazo dos Diplomas Assinadas

De acordo com os trabalhos que estão sendo desenvolvidos nas Universidades Federais, para se obter uma melhor segurança dos diplomas assinados por um longo período de tempo, está sendo utilizado Blockchain [102], [103], [104], [105], [70]. A universidade Federal da Paraíba (UFPB) realizou um estudo com o objetivo de criar uma rede de Blockchain entre as Universidades Federais Brasileiras para que os diplomas assinados com o padrão XAdES possam ter uma nova forma de segurança que garanta que não possam ser alterados internamente nem externamente sem a invalidação dos diplomas já inseridos nessa rede [68].

Um outro trabalho desenvolvido pela Fundação Getúlio Vargas (FGV) tem uma proposta de criação de uma rede de Blockchain privada para que as Universidades Federais compartilhem seus diplomas assinados no padrão XAdES, garantindo a segurança interna e externa desses documentos. Essa rede privada teria o MEC como órgão central e as Universidades Federais como compartilhadoras dos diplomas dentro da rede Blockchain, onde eles seriam assinados e sua segurança ficaria reforçada. A Figura 6.8, apresenta a rede Blockchain proposta pela FGV [13].

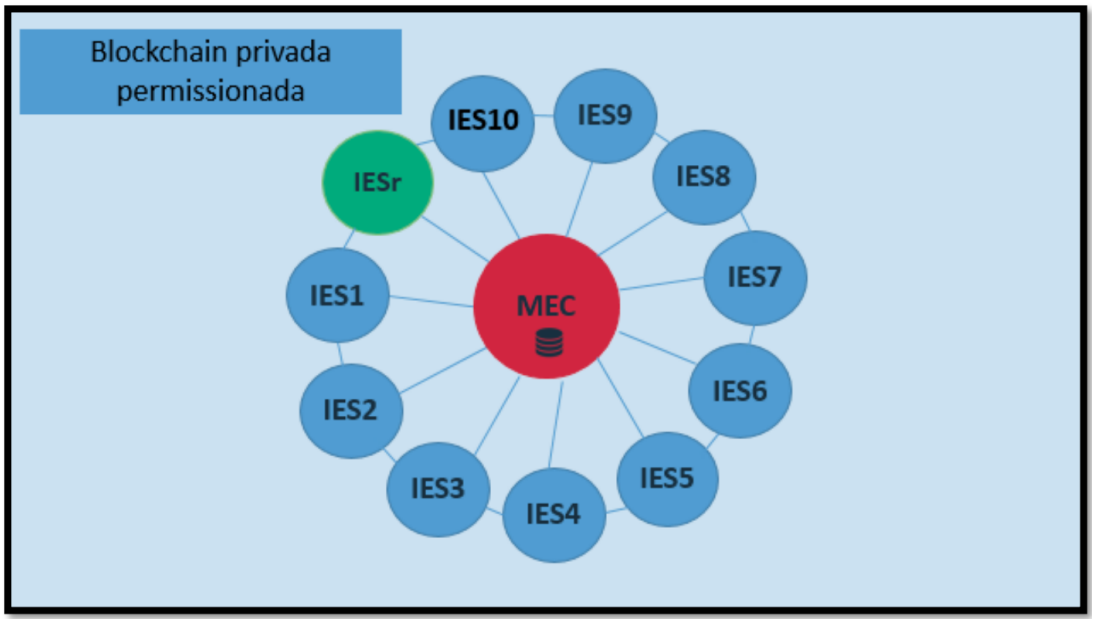


Figura 6.8: Blockchain privada permissionada [13]

Com a proposta de desenvolvimento de uma nova rede para Blockchain e com a obrigatoriedade dos diplomas serem assinados seguindo os normativos do MEC e da IPC-Brasil, é necessário que os diplomas sejam assinados primeiramente no padrão XAdES com carimbo de tempo, podendo posteriormente ser inseridos na rede Blockchain e assinados para que não possam ser modificados interna ou externamente, o que quebraria a cadeia de armazenamento Blockchain. Essa estratégia garante que nenhum dado de um diploma já assinado possa ser alterado.

De acordo com a política de preservação de documentos nato digital do Arquivo Nacional, a entidade deve ser capaz de manter a autenticidade dos documentos armazenados digitalmente pela Instituição. Assim, devem ser utilizadas políticas e métodos para assegurar que o documento não seja manipulado, alterado ou falsificado [106]. Todas as cópias feitas do documento digital armazenado devem ser autênticas, todas as alterações feitas devem ser mapeadas e todos os documentos digitais armazenados deverão ser identificados, classificados e catalogados para que se tenha o registro dos documentos digitais que estão armazenados [106].

Todos os dados presentes nos documentos em formato digital devem ser mantidos de maneira íntegra e sem alterações desde o momento em que o documento é considerado válido [106]. No âmbito da UnB um diploma é considerado válido no momento que ele foi assinado de acordo com o padrão XAdES. Desse momento em diante os dados presentes no diploma digital devem ser preservados de maneira íntegra.

De acordo com o que pressupõe o Arquivo Nacional, o repositório de armazenamento de documentos digitais deve ter a capacidade de expansão minimamente para três anos [106]. Já a nota técnica do MEC não estabelece um período mínimo para validade e tempo de armazenamento dos diplomas assinados digitalmente, porém é mencionado que um diploma assinado digitalmente deve ser atualizado, ter validade a qualquer tempo e ser armazenado por um longo período de tempo [7].

Para garantir a disponibilidade, integridade, autenticidade e validade do diploma digital a qualquer tempo sem que haja fraudes, além de todas as medidas apresentadas acima, é necessário que o diploma só possa ser assinado uma vez. Caso seja necessário alterar alguma informação do diploma, os normativos do MEC mencionam que deve ser invalidado o diploma antigo e assinado um diploma novo [7]. Toda vez que for invalidado um diploma, deve-se salvar em um log permanente qual foi o funcionário que efetuou a invalidação do diploma. Essa é uma das garantias, além do Blockchain, que impede que aconteçam fraudes internas.

A nota técnica do MEC menciona que o diploma digital deve ser armazenado em ambiente digital que garanta a validação a qualquer tempo, interoperabilidade entre sistemas, atualização tecnológica de segurança e possibilidade de múltiplas assinaturas no mesmo diploma [7]. Tendo isso em vista, a UnB deve manter os diplomas assinados digitalmente, armazenados localmente em servidores da UnB e ainda, quando a rede de Blockchain estiver ativa, armazenar esses diplomas nessa rede.

Utilizando a técnica de blockchain e de assinatura com o padrão XAdES é possível manter a segurança dos diplomas em formato digital. Com o compartilhamento do diploma nessa futura rede, será possível ter uma alta disponibilidade para a verificação da validade do diploma a qualquer tempo. Outro ponto importante, abordado tanto pela nota técnica do MEC quanto pelo Arquivo Nacional, é o fato de que os diplomas manterão sua validade por um longo período de tempo.

Para resolver este problema foi proposta a adição dos diplomas assinados na rede de Blockchain. Além disso, os documentos podem ficar armazenados em uma Trusted Archival Services (TAS). Esse é um local de armazenamento seguro, onde os documentos armazenados nesse repositório de arquivos ficam disponíveis de maneira segura. Caso haja uma evolução tecnológica, apenas o servidor central de arquivos deve ser atualizado para manter os documentos armazenados e já assinados de maneira segura e válida [107].

Com a adição do diploma assinado na rede de Blockchain que será criada pelo MEC e com o armazenamento interno em um TAS os diplomas terão uma garantia de segurança a longo prazo, independente das mudanças tecnológicas e de assinatura de documentos que poderão ocorrer futuramente.

6.4 Políticas de Segurança Interna dos Diplomas Armazenados

Nesta seção será apresentado um resumo das políticas de segurança interna que são necessárias para que se mantenha os diplomas armazenados pela UnB seguros, além de garantir que a solução de assinatura proposta neste trabalho esteja disponível quando necessária e não sofra com ataques internos que invalidem futuros diplomas a serem assinados por essa biblioteca.

O modelo proposto inicialmente foi validado e desenvolvido como uma solução open-source, adicionada no github e mantido pela UnB. Apesar de ser um projeto aberto para uso, apenas as pessoas autorizadas e responsáveis pela manutenção da biblioteca de assinatura digital poderão alterar e manter a solução atual.

O Arquivo Nacional Brasileiro assume o compromisso pela preservação e acesso aos documentos digitais sob sua custódia, assim como os metadados associados a ele. Aos documentos que estão sob responsabilidade do Arquivo Nacional devem ser garantidas a recuperação, inteligibilidade e autenticidade, para que possam servir de fonte de prova e informação [106]. Para promover as estratégias de preservação, o Arquivo Nacional se baseia nos seguintes procedimentos [106]:

- normalização dos formatos de arquivo no momento da inserção dos documentos no repositório;
- monitoramento dos formatos em uso;
- migração (atualização de versões e de formatos de arquivo) sempre que verificada a necessidade, devido ao avanço da tecnologia e a decorrente obsolescência tecnológica.

Tendo o Arquivo Nacional também como arquivo central de documentos na esfera federal, uma das estratégias é enviar o documento para o Arquivo Nacional, garantindo assim que a segurança dos diplomas assinados sejam válidas, mesmo se forem alterados internamente na UnB. O Arquivo Nacional em sua diretriz para preservação de documentos em formato digital recomenda que se tenha uma cópia oficial e válida preservada digitalmente durante todo o período em que o documento deve ser preservado. Com isso fica garantida uma cópia válida, íntegra e autêntica [108].

Outro ponto importante abordado pelo Arquivo Nacional é que não se pode assegurar a preservação de documentos em formato digital a longo prazo, se estes não estiverem corretamente mantidos e protegidos dentro do sistema de manutenção ou preservação que os contêm. As principais estratégias para manutenção a longo prazo são [108]:

- Distribuição de responsabilidades.
- Provisão de infraestrutura técnica adequada.
- Implementação de um plano de manutenção, suporte e substituição do sistema.
- Implementação de um plano para a transferência regular de documentos arquivísticos para novas mídias de armazenamento.
- Adesão a condições adequadas de armazenamento e manuseio voltada para mídias digitais.
- Redundância e backup regular das entidades digitais.
- Estabelecimento de um sistema de segurança.
- Planejamento para situação de emergência.

O Arquivo Nacional questiona o uso apenas da certificação e do carimbo de tempo como verificador de validade e autenticidade de um documento. O carimbo de tempo pode ser modificado se o relógio do sistema for ajustado, o que pode dificultar a garantia de validade de um documento em formato digital [108]. É necessário que os documentos preservados digitalmente sejam monitorados periodicamente para garantir sua autenticidade, integridade e adequação a futuras tecnologias de assinatura digital que venham a surgir [108].

As cópias devem ser identificadas como cópias, ou seja, não se pode ter duplicatas de documentos originais, deve se manter apenas um documento original e na transferência, tanto das cópias quanto do documento original, estes devem estar claramente identificados e marcados, a fim de se evitar transferências acidentais [108].

Outra preocupação do Arquivo Nacional é explicar quais os requisitos técnicos mínimos necessários para se ter acesso ao documento em formato digital. Devem ser explicitados como é feito o acesso, quais plataformas devem ser usadas e quais os requisitos de autenticação e autorização para o documento a ser acessado [108]. Além disso, pode ser feito um comparativo entre o diploma armazenado localmente na UnB com o diploma armazenado no futuro sistema de Blockchain e com o armazenamento no Arquivo Nacional. Assim, pode-se obter uma garantia maior da disponibilidade do diploma e segurança quanto a

alteração localmente do diploma, pressupondo que o diploma armazenado no Arquivo Nacional e na rede de Blockchain estejam sempre válidos, conforme determina as políticas de armazenamento do Arquivo Nacional.

Além do Arquivo Nacional, a LGPD menciona que são necessárias medidas de segurança para que seja assegurada a integridade dos dados pessoais, minimizando os riscos de perda ou vazamento de dados. Assim, são fundamentais políticas para garantir que a informação não seja acessível e nem modificada por pessoas que não tenham permissão para isso. Além da segurança informada, das distribuições e controles mencionados pelo Arquivo Nacional, é necessária a implementação de mecanismos tecnológicos que visem garantir a segurança interna do acesso e modificação desses dados [16].

A melhor maneira para garantir a segurança interna é a criptografia de ponta a ponta, ou seja, toda a comunicação do momento inicial do acesso ao serviço de assinatura até a comunicação com o banco de dados deve ser criptografada via ssl. Para garantir a segurança é necessário manter a versão mais recente desse protocolo [109].

Conforme detalhado no capítulo 4, toda a comunicação, inclusive o SDK da máquina do usuário, deve estar criptografada com o protocolo ssl. Além disso, os acessos aos dados devem ser limitados apenas aos servidores que tenham permissão para alterar ou visualizar esses dados. Com isso é garantido que os dados não fiquem expostos tanto externa quanto internamente.

6.5 Diferenças entre as diretrizes do MEC e o que Recomenda a Academia

Para assinatura digital as diretrizes do MEC recomendam que seja feita a assinatura utilizando o padrão XAdES com o carimbo de tempo. O diploma deve ser assinado com, no mínimo, um certificado digital A3 ou superior. Além do XML que deve ser assinado, também é necessário que seja disponibilizado um diploma em formato visual. Já os trabalhos identificados na literatura sobre assinatura digital estão utilizando o Blockchain, tanto para assinar um documento, quanto para manter o documento válido por um longo período. O Blockchain é uma espécie de assinatura digital onde diversos formatos de assinatura digital podem ser realizados nos documentos:

- Assinatura Agregada: É uma espécie de assinatura que recebe n assinaturas diferentes de n usuários diferentes e é possível resumir todas essas assinaturas em uma única assinatura curta. Em resumo, garante que os n usuários realmente assinaram o documento. A vantagem dessa técnica é que reduz muito o poder computacional

para o armazenamento e verificação da assinatura. Essa solução é recomendada em locais com baixa largura de banda e espaço de armazenamento limitado [110].

- Assinatura em Anel: Esse esquema pressupõe uma chave pública para todos os usuários e uma chave privada para cada um dos usuários. Esse tipo de assinatura é utilizada em documentos que precisam de rastreabilidade e de proteção a longo prazo. Essa abordagem elimina a necessidade de um terceiro para validar a assinatura no documento [110].

O Blockchain é uma forma de assinatura peer-to-peer onde não há um ponto central de controle. Para garantir a segurança, uma vez gravada uma assinatura de Blockchain, esse documento não pode ser alterado, porém, deve aceitar novas transações. Para se adicionar novos elementos à cadeia de Blockchain é necessário colocar uma referência hash do documento anterior no atual documento adicionado. Ao fazer isso, a tentativa de quebra em um documento impacta em todos os seus documentos anteriores. Com isso é possível adicionar novas transações, mas é quase impossível alterar os dados que foram adicionados no passado. Qualquer tentativa de alteração faz uma quebra nas referências de hash dos documentos [111].

A União Europeia tem um documento com as diretrizes para implementação de um histórico escolar utilizando certificação digital. O Europass Framework for Digitally-Signed Credentials demonstra as vantagens do uso de Blockchain em conjunto com a estrutura de assinaturas de chave pública PKI [72]. Essa tecnologia possibilita que a autoridade de verificação seja descentralizada, além de fornecer um controle das transações executadas nesse documento assinado. Evita também tentativas de fraudes que podem gerar um documento com assinatura falsa [72] invalidando a cadeia completa de certificados, o que facilita a identificação do documento fraudado.

6.6 Problemas do Padrão de Assinatura de Diplomas Digitais do MEC

Os principais trabalhos identificados na literatura em relação ao desenvolvimento de soluções para assinatura e diplomas e documentos em formato digital mencionam o uso de Blockchain, que é uma das soluções mais seguras para garantir a autenticidade de uma informação ou documento.

As duas portarias do MEC e a nota técnica do MEC, utilizam o padrão XAdES para assinatura digital de diplomas em formato digital. Em nenhum momento é mencionado a criação e armazenamento desses diplomas assinado em uma rede de Blockchain, que

atualmente é o padrão mais seguro para armazenamento de informações. Dessa forma considera-se uma falha da solução proposta pelo MEC deixar de mencionar essa tecnologia.

Existem estudos iniciais na ICP-Brasil e nas Universidades sobre Blockchain [112], [67], mas ainda não existe nenhuma iniciativa concreta na área educacional para implementação de uma rede de Blockchain.

Outro problema encontrado, é o fato de que os dados exigidos pelo MEC para a geração do diploma em formato digital são apenas os dados mínimos necessários para que seja garantida a conformidade com a Lei Geral de Proteção de Dados Pessoais (LGPD), por exemplo, no diploma são exigidos dados pessoais, tais como CPF e RG, que poderiam ser mantidos em formato criptografado e não de maneira explícita no documento.

O padrão europeu de assinatura de diplomas digitais pressupõe que cada estudante tenha um Europass. O Europass é o certificado digital emitido para cada estudante em formato de cartão com todos os certificados e diplomas que o estudante tem em seu currículo. Assim, o Europass funciona como um currículo digital validado com certificação digital. A assinatura dos certificados e diplomas são realizadas pela instituição e, também, pelo próprio estudante [72]. Já no Brasil, apenas o diploma de universidades está sendo validado com certificado digital e apenas pela instituição, pois o aluno não terá um cartão digital como na Europa.

Com o Europass os documentos do estudante podem ser validados sem a necessidade de um terceiro intermediário para validar os dados ou a identidade do estudante. Além disso, a União Europeia demonstra preocupação com a emissão de certificados em formato físico. O diploma assinado digitalmente deve ter uma maior segurança que um diploma emitido em formato físico [72].

Os diplomas e certificados assinados em território europeu devem ser válidos dentro e fora da Europa [72]. Já no Brasil, o Ministério da Educação com uso da ICP-Brasil garante que os diplomas assinados tenham validade apenas em território nacional [7].

Tanto no Brasil, com as portarias do MEC e as regras da ICP-Brasil, como na Europa com o European Digital Credentials Infrastructure (EDCI) é utilizada a assinatura dos diplomas em formato XML [72]. O padrão europeu de assinatura usa o framework stands for Electronic Identification (eID) and Trust Services (AS) (eIDAS). Nesse framework é criada uma identidade eletrônica a qual contém os seguintes itens [113]:

- eSignature: Assinatura eletrônica para documentos e certificados.
- eSeal: Serve para criar um carimbo de tempo para garantir a origem do documento assinado.
- eTimestamp: Tem por objetivo adicionar um carimbo de tempo para assegurar que o documento tenha sido assinado com um certificado válido naquele período.

- eDelivery: Um serviço seguro que permite a transferência de documentos por meio eletrônico entre empresas, administração pública e o cidadão. É fornecida prova de envio e recebimento dos documentos e dados, protegendo contra perda, roubo, dano ou alterações não autorizadas.
- WACs: Certificados para gerenciar e garantir a autenticidade de um site no território europeu.

Nas portarias e no normativo do MEC, não se discute em detalhes como fazer o compartilhamento seguro dos diplomas digitais assinados e tampouco o armazenamento a longo prazo; já no padrão europeu pode-se ver que há diversas soluções de assinatura e camadas para garantir a segurança e o compartilhamento seguro. Pela legislação do MEC, o diploma, diferentemente do europeu, não fica em um portfólio acadêmico exclusivo, sendo disponibilizado sem o histórico completo [7], [72], [113].

6.7 Síntese do Capítulo

Este capítulo apresentou os testes realizados nos estudos abordados neste trabalho. A primeira solução analisada é o atual sistema utilizado pela UnB, o C3Web, o qual apresentou um bom desempenho para assinatura do lote de 200 diplomas em 6 minutos e 45 segundos, porém algumas pequenas possíveis falhas de segurança, além de ser uma solução fechada que utiliza uma biblioteca fechada a e-sec. Por este motivo é uma solução que inviabiliza futuras evoluções e manutenções em um tempo adequado. O segundo teste foi feito utilizando o protótipo desenvolvido dentro do barramento Erlangms com um tempo de 10 minutos e 20 segundos para assinar o mesmo lote de diplomas. Essa solução foi a que apresentou a menor quantidade de possíveis falhas de segurança, porém tem o pior desempenho e, também, é a solução com maior dificuldade de manutenção e evoluções futuras. Não existe uma biblioteca que facilite a criação de uma solução em XAdES, além disso por ser desenvolvido na linguagem Erlang, que é uma linguagem de desenvolvimento mais complexa e de difícil manutenção pela STI/UnB. O ultimo teste foi feito utilizando o protótipo proposto por este estudo é a solução desenvolvida em Java com a biblioteca XAdES4j. Esse modelo apresentou o código com a menor quantidade de erros, nenhuma falha de segurança e um desempenho de 9 minutos e 25 segundos para assinar o mesmo lote. Esse tempo está dentro do esperado pela UnB. A vantagem é que essa solução é desenvolvida de maneira independente, por meio de microsserviços, o que permite que possa ser facilmente integrado, de fácil manutenção e evolução. É totalmente customizável para os padrões do MEC e da ICP-Brasil e para futuras tecnologias e padrões que possam surgir.

Foi também mostrado como executar a política de segurança interna e externa. A segurança externa é feita pela própria assinatura no padrão XAdES e os normativos do MEC e da ICP-Brasil. Além da segurança externa é necessário manter os diplomas assinados digitalmente íntegros, válidos e autênticos. Para isso é fundamental que apenas pessoas autorizadas tenham acesso ao sistema e que futuramente os diplomas sejam inseridos na rede de Blockchain das universidades, assim que essa rede esteja efetivada. É necessário que seja mantido um registro de todas as pessoas que assinarem um diploma ou invalidarem um diploma previamente assinado, assim como o log das pessoas que fizerem essas mudanças.

É importante ressaltar o comparativo feito entre a atual legislação do MEC e da ICP-Brasil com a atual legislação da Europa no que tange a assinatura digita, foi verificado que o Brasil tem algumas diferenças, uma delas é não mencionar o uso de Blockchain para garantir uma maior segurança. Ambos utilizam o padrão XAdES-T, porém a Europa disponibiliza um cartão digital para cada estudante, o que garante, também, uma segurança de alto nível para os diplomas e para o currículo Europeu.

Ao final é demonstrado os problemas encontrado nas legislações de assinatura digital do MEC e da ICP-Brasil. É verificado que o padrão principal para assinatura de diplomas em formato digital utilizando Blockchain. Essa tecnologia pode ser integrada com a assinatura XAdES. Por fim foi possível validar a legislação brasileira, validar o modelo desenvolvido neste estudo e as evoluções futuras, tanto no modelo proposto quando na legislação brasileira atualmente adotada.

Capítulo 7

Conclusões

O objetivo deste trabalho foi a criação de um modelo para a concepção de um sistema de assinatura de diplomas em formato digital que seja aderente aos normativos do MEC e aos padrões da ICP-Brasil e à arquitetura tecnológica atualmente utilizada pela STI/UnB, haja vista não existir ainda uma solução para assinatura digital no formato XAdES conforme os referidos normativos e a atual necessidade da UnB. O modelo desenvolvido possui código aberto e fácil implementação por qualquer outra universidade que venha a utilizar a solução proposta.

Foi realizada uma revisão da literatura para evidenciar as principais referências nas áreas de criptografia, certificação digital, segurança e microsserviços. Neste trabalho foi realizado um estudo de caso utilizando a metodologia proposta por Yin[1]. Foi desenvolvido um protótipo baseando-se em toda a legislação referente a assinatura de diplomas em formato digital, tanto do MEC quanto da ICP-Brasil. A solução proposta abordou apenas uma parte dos requisitos exigidos pelo MEC para a criação, validação, armazenamento e descarte dos diplomas assinados em formato digital, utilizando o padrão XAdES. Para atender a todos os requisitos seria necessário observar a forma de armazenamento que é regulamentada nos normativos do MEC, o tempo de armazenamento de um diploma em formato digital e como deve ser disponibilizado o diploma, dentre outros requisitos.

Foram realizados testes comparativos, um utilizando o atual sistema da UnB, o C3Web, outro utilizando um protótipo desenvolvido na linguagem Erlang e um terceiro um protótipo desenvolvido em Java utilizando a biblioteca XAdES4j. Os testes demonstraram que a solução proposta tem uma segurança equiparada ao sistema atual da UnB, excelente escalabilidade e a possibilidade de assinar dezenas de milhares de documentos. Além disso, não possui vulnerabilidades de segurança e tem a capacidade de assinar e validar documentos externos. Entretanto, embora a segurança da arquitetura proposta seja a mesma encontrada no atual sistema da UnB, a principal vantagem do modelo proposto é o fato de ser código aberto, ter a possibilidade de validação no formato da ICP-Brasil e

de ser um modelo de fácil manutenção e evoluções futuras.

Com o uso do modelo proposto é possível a Universidade de Brasília desenvolver seu próprio sistema de assinatura digital aderente aos padrões do MEC e da ICP-Brasil reduzindo uma quantidade considerável de custos com a emissão de diplomas em formato físico e com contratos para criação, manutenção e evolução do sistema de assinatura digital e que tenha a possibilidade de ser usado por diversas outras Universidades Federais.

Referências

[1] Yin, Robert K.: *Case Study Research: Design and Methods (Applied Social Research Methods)*. Sage Publications, fourth edition. edição, 2008, ISBN 1412960991. vi, viii, xii, 6, 7, 28, 29, 30, 31, 32, 33, 36, 90

[2] Forouzan, Behrouz, Catherine Coombs e Sophia Chung Fegan: *Introduction to data communications and networking*. McGraw-Hill, Inc., 2012. xii, 2, 9, 10, 11, 12, 13, 17, 34

[3] Sklavos, Nicolas: *Book review: Stallings, w.* Cryptography and Network Security: Principles and Practice - *6th ed. upper saddle river, NJ: prentice hall, 2013, 752p., $142.40. ISBN: 13: 978-0133354690*. Inf. Secur. J. A Glob. Perspect., 23(1-2):49–50, 2014. https://doi.org/10.1080/19393555.2014.900834. xii, 2, 3, 10, 11, 12, 13, 14, 15, 16, 17, 34, 42

[4] Rosen, Michael, Boris Lublinsky, Kevin T Smith e Marc J Balcer: *Applied SOA: service-oriented architecture and design strategies*. John Wiley & Sons, 2012. xii, 20

[5] Parsons, David: *The java story*. Em *Foundational Java*, páginas 1–10. Springer, 2020. xii, 21, 22

[6] *W3C: extensible markup language (XML) 1.0 - first edition*, 1998. https://www.w3.org/TR/xml/. xii, 39

[7] *Nota técnica no. 13/2019/difes/sesu/sesu*, 2019. https://abmes.org.br/arquivos/documentos/nota-tecnica-%20diploma-digital-11122019.pdf. xii, 35, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 52, 68, 82, 87, 88

[8] DE, GERAÇÃO E VERIFICAÇÃO: *Requisitos mínimos para geração e verificação de assinaturas digitais na icp-brasil doc-icp-15.01 versão 1.0*. ICP-Brasil, 2008. xii, 51

[9] Fink, Gil, Ido Flatow, SELA Group *et al.*: *Pro Single Page Application Development: Using Backbone. Js and ASP. Net*. Apress, 2014. xii, 55

[10] Agilar, Everton de Vargas: *Uma abordagem orientada a serviços para a modernização de sistemas legados*. Universidade de Brasília, 2016. xii, 54, 55, 56, 57

[11] Agilar, Everton, Rodrigo Almeida e Edna Dias Canedo: *A systematic mapping study on legacy system modernization*. Em *SEKE*, páginas 345–350. KSI Research Inc. and Knowledge Systems Institute Graduate School, 2016. xii, 56, 57

[12] Gonçalves, Luís Filipe dos Santos: *Xades4j: a java library for xades signature services*. github, 2010. xiii, 60, 61

[13] Dubrowsky, Alexander: *Transformação digital nas instituições privadas de ensino superior brasileiras: proposta para autenticação de diplomas digitais de graduação por meio de blockchain*. Fundação Getúlio Vargas, 2019. xiii, 26, 81

[14] Whitman, Michael E e Herbert J Mattord: *Principles of information security*. Cengage Learning, 2011. 1

[15] Regulation, General Data Protection: *Eu data protection rules*, 2018. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en. 1

[16] República, Presidência da: *Lei geral de proteção de dados pessoais (lgpd)*, 2018. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. 1, 3, 85

[17] Neuman, B. C. e T. Ts'o: *Kerberos: an authentication service for computer networks*. IEEE Communications Magazine, 32(9):33–38, 1994. 2

[18] Bellare, Mihir, Kenneth G. Paterson e Phillip Rogaway: *Security of symmetric encryption against mass surveillance*. Em Garay, Juan A. e Rosario Gennaro (editores): *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 de *Lecture Notes in Computer Science*, páginas 1–19. Springer, 2014. https://doi.org/10.1007/978-3-662-44371-2_1. 2

[19] Diffie, Whitfield e Martin E. Hellman: *New directions in cryptography*. IEEE Trans. Inf. Theory, 22(6):644–654, 1976. https://doi.org/10.1109/TIT.1976.1055638. 2

[20] Rivest, Ronald L., Adi Shamir e Leonard M. Adleman: *A method for obtaining digital signatures and public-key cryptosystems (reprint)*. Commun. ACM, 26(1):96–99, 1983. https://doi.org/10.1145/357980.358017. 3

[21] Tycksen Jr, Frank A e Charles W Jennings: *Digital certificate*, 2001. US Patent 6,189,097. 3

[22] Educação, Ministério da: *Portaria nº 554, de 11 de marÇo de 2019*, 2019. http://abmes.org.br/arquivos/legislacoes/Portaria-MEC-554-2019-03-11.pdf. 3

[23] *Unb diploma. sistema de emissão de diplomas digitais*, 2018. http://servicos.unb.br/diploma. 3

[24] *Emissão de diplomas digitais*, 2019. https://www.noticias.unb.br/67-ensino/2723-emissao-de-diplomas-digitais-chega-a-pos-graduacao. 3

[25] Oracle: *Java se development kit 8*. Java, 2019. https://www.oracle.com/br/java/technologies/javase/javase-jdk8-downloads.html. 4

[26] *Diploma digital*, 2020. http://portal.mec.gov.br/diplomadigital/. 4, 40, 48

[27] *Portaria n 554, de 11 de março de 2019*, março 2019. http://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/66544171/do1-2019-03-12-portaria-n-554-de-11-de-marco-de-2019-66543842. 4, 38

[28] Sjøberg, Dag I. K., Jo Erskine Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg e Anette C. Rekdal: *A survey of controlled experiments in software engineering*. IEEE Trans. Software Eng., 31(9):733–753, 2005. https://doi.org/10.1109/TSE.2005.97. 6

[29] *European telecommunications standards institute (etsi)*, 2019. https://www.etsi.org/technologies/digital-signature. 9

[30] *Regulamento (ue) n.o 910/2014 do parlamento europeu e do conselho*, julho 2014. 9

[31] *Icp-brasil*, 2019. https://www.iti.gov.br/icp-brasil. 9, 50

[32] Khalique, Aqeel, Kuldip Singh e Sandeep Sood: *Implementation of elliptic curve digital signature algorithm*. International journal of computer applications, 2(2):21–27, 2010. 10

[33] Bellare, Mihir, Anand Desai, E. Jokipii e Phillip Rogaway: *A concrete security treatment of symmetric encryption*. Em *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, páginas 394–403. IEEE Computer Society, 1997. https://doi.org/10.1109/SFCS.1997.646128. 10

[34] Szerwinski, Robert e Tim Güneysu: *Exploiting the power of gpus for asymmetric cryptography*. Em Oswald, Elisabeth e Pankaj Rohatgi (editores): *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 de *Lecture Notes in Computer Science*, páginas 79–99. Springer, 2008. https://doi.org/10.1007/978-3-540-85053-3_6. 10

[35] Rivest, Ronald L., Adi Shamir e Leonard M. Adleman: *A method for obtaining digital signatures and public-key cryptosystems*. Commun. ACM, 21(2):120–126, 1978. http://doi.acm.org/10.1145/359340.359342. 12

[36] Bellare, Mihir e Phillip Rogaway: *The exact security of digital signatures - how to sign with RSA and rabin*. Em Maurer, Ueli M. (editor): *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 de *Lecture Notes in Computer Science*, páginas 399–416. Springer, 1996. https://doi.org/10.1007/3-540-68339-9_34. 12

[37] Rogaway, Phillip e Thomas Shrimpton: *Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance*. Em Roy, Bimal K. e Willi Meier (editores): *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February*

5-7, 2004, Revised Papers, volume 3017 de *Lecture Notes in Computer Science*, páginas 371–388. Springer, 2004. https://doi.org/10.1007/978-3-540-25937-4_24. 13

[38] Trappe, Wade: *Introduction to cryptography with coding theory*. Pearson Education India, 2006. 15

[39] Coron, Jean Sébastien: *Optimal security proofs for pss and other signature schemes*. Em Knudsen, Lars R. (editor): *Advances in Cryptology — EUROCRYPT 2002*, páginas 272–287, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg, ISBN 978-3-540-46035-0. 16, 17

[40] Schaad, Jim: *Use of the RSASSA-PSS signature algorithm in cryptographic message syntax (CMS)*. RFC, 4056:1–6, 2005. https://doi.org/10.17487/RFC4056. 17

[41] *Itu-t recommendations, iuti-t x.509*, outubro 2019. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509. 17

[42] Housley, Russell, Warwick Ford, William Polk e David Solo: *Internet x. 509 public key infrastructure certificate and crl profile*. Relatório Técnico, RFC 2459, January, 1999. 18

[43] ETSI, TS: *101 903 xml advanced electronic signatures (xades), v1. 3.2*. European Telecommunications Standards Institute (ETSI), 2006. 18, 48

[44] Brzica, Hrvoje, Boris Herceg e Hrvoje Stančić: *Long-term preservation of validity of electronically signed records*. Em *INFuture2013*, novembro 2013. 18

[45] Luz, Welder Pinheiro, Everton Agilar, Marcos César de Oliveira, Carlos Eduardo R. de Melo, Gustavo Pinto e Rodrigo Bonifácio: *An experience report on the adoption of microservices in three brazilian government institutions*. Em Kulesza, Uirá (editor): *Proceedings of the XXXII Brazilian Symposium on Software Engineering, SBES 2018, Sao Carlos, Brazil, September 17-21, 2018*, páginas 32–41. ACM, 2018. https://doi.org/10.1145/3266237.3266262. 19

[46] Newman, Sam: *Building microservices - designing fine-grained systems, 1st Edition*. O'Reilly, 2015, ISBN 9781491950357. http://www.worldcat.org/oclc/904463848. 19, 20

[47] Carvalho, Luiz, Alessandro Garcia, Wesley K. G. Assunção, Rodrigo Bonifácio, Leonardo P. Tizzei e Thelma Elita Colanzi: *Extraction of configurable and reusable microservices from legacy systems: An exploratory study*. Em *Proceedings of the 23rd International Systems and Software Product Line Conference - Volume A*, SPLC '19, página 26–31, New York, NY, USA, 2019. Association for Computing Machinery, ISBN 9781450371384. https://doi.org/10.1145/3336294.3336319. 19, 20

[48] Bogner, J. e A. Zimmermann: *Towards integrating microservices with adaptable enterprise architecture*. Em *2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW)*, páginas 1–6, Sep. 2016. 19

[49] Dragoni, Nicola, Ivan Lanese, Stephan Thordal Larsen, Manuel Mazzara, Ruslan Mustafin e Larisa Safina: *Microservices: How to make your application scale*. Em Petrenko, Alexander K. e Andrei Voronkov (editores): *Perspectives of System Informatics - 11th International Andrei P. Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27-29, 2017, Revised Selected Papers*, volume 10742 de *Lecture Notes in Computer Science*, páginas 95–104. Springer, 2017. https://doi.org/10.1007/978-3-319-74313-4_8. 19, 20

[50] Balalaie, Armin, Abbas Heydarnoori e Pooyan Jamshidi: *Microservices architecture enables devops: Migration to a cloud-native architecture*. IEEE Softw., 33(3):42–52, 2016. https://doi.org/10.1109/MS.2016.64. 19

[51] Yarygina, T. e A. H. Bagge: *Overcoming security challenges in microservice architectures*. Em *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, páginas 11–20, March 2018. 20

[52] Morgan, Jeff e Garret E. O'Donnell: *Enabling a ubiquitous and cloud manufacturing foundation with field-level service-oriented architecture*. Int. J. Comput. Integr. Manuf., 30(4-5):442–458, 2017. https://doi.org/10.1080/0951192X.2015.1032355. 20

[53] Jamshidi, P., C. Pahl, N. C. Mendonça, J. Lewis e S. Tilkov: *Microservices: The journey so far and challenges ahead*. IEEE Software, 35(3):24–35, May 2018, ISSN 1937-4194. 21

[54] Java, Oracle: *Oracle java*. Java, 2021. https://www.oracle.com/br/java/. 21

[55] *VisÃo geral sobre assinaturas digitais na icp-brasil*, 2016. https://www.iti.gov.br/images/repositorio/legislacao/documentos-principais/15/DOC-ICP-15_v.1.0.pdf. 22

[56] *Requisitos das polÍticas de assinatura digital na icp-brasil*, 2016. http://www.iti.gov.br/images/repositorio/consulta-publica/encerradas/DOC-ICP-15.03_-_Versao_7.1_Consulta_Publica.pdf. 23

[57] Boneh, Dan, Craig Gentry, Ben Lynn e Hovav Shacham: *Aggregate and verifiably encrypted signatures from bilinear maps*. Em Biham, Eli (editor): *Advances in Cryptology — EUROCRYPT 2003*, páginas 416–432, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg, ISBN 978-3-540-39200-2. 23

[58] Crampton, Jason: *Applying hierarchical and role-based access control to xml documents*. Em *Proceedings of the 2004 Workshop on Secure Web Service*, SWS '04, página 37–46, New York, NY, USA, 2004. Association for Computing Machinery, ISBN 158113973X. https://doi.org/10.1145/1111348.1111353. 23, 34

[59] Kumar, Manish, Akhlad Iqbal e Pranjal Kumar: *A new rgb image encryption algorithm based on dna encoding and elliptic curve diffie–hellman cryptography*. Signal Processing, 125:187 – 202, 2016, ISSN 0165-1684. http://www.sciencedirect.com/science/article/pii/S0165168416000347. 23

[60] Perin, G., D. G. Mesquita, F. L. Herrmann e J. B. Martins: *Montgomery modular multiplication on reconfigurable hardware: Fully systolic array vs parallel implementation*. Em *2010 VI Southern Programmable Logic Conference (SPL)*, páginas 61–66, March 2010. 23, 34

[61] Engelbertz, Nils, Vladislav Mladenov, Juraj Somorovsky, David Herring, Nurullah Erinola e Jörg Schwenk: *Security analysis of xades validation in the CEF digital signature services (DSS)*. Em Roßnagel, Heiko, Sven Wagner e Detlef Hühnlein (editores): *Open Identity Summit 2019, OID 2019, Garmisch-Partenkirchen, Germany, March 28-29, 2019*, volume P-293 de *LNI*, páginas 95–106. GI, 2019. https://dl.gi.de/20.500.12116/20997. 24, 34

[62] Krylovskiy, A., M. Jahn e E. Patti: *Designing a smart city internet of things platform with microservice architecture*. Em *2015 3rd International Conference on Future Internet of Things and Cloud*, páginas 25–30, Aug 2015. 24

[63] Dragoni, Nicola, Saverio Giallorenzo, Alberto Lluch-Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin e Larisa Safina: *Microservices: Yesterday, today, and tomorrow*. Em Mazzara, Manuel e Bertrand Meyer (editores): *Present and Ulterior Software Engineering*, páginas 195–216. Springer, 2017. https://doi.org/10.1007/978-3-319-67425-4_12. 24

[64] Rahaman, Sazzadur, Ya Xiao, Sharmin Afrose, Fahad Shaon, Ke Tian, Miles Frantz, Murat Kantarcioglu e Danfeng (Daphne) Yao: *Cryptoguard: High precision detection of cryptographic vulnerabilities in massive-sized java projects*. Em *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, página 2455–2472, New York, NY, USA, 2019. Association for Computing Machinery, ISBN 9781450367479. https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3319535.3345659. 24

[65] Ketkar, Ameya, Nikolaos Tsantalis e Danny Dig: *Understanding type changes in java*. Em *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, página 629–641, New York, NY, USA, 2020. Association for Computing Machinery, ISBN 9781450370431. https://doi.org/10.1145/3368089.3409725. 24

[66] Lepiane, Cristiane Dias, Fernando Lauro Pereira, Giovani Pieri, Douglas Marcelino Beppler Martins, Jean Everson Martina e Mauro Luiz Rabelo: *Digital degree certificates for higher education in brazil: A technical policy specification*. Em Schimmler, Sonja e Uwe M. Borghoff (editores): *Proceedings of the ACM Symposium on Document Engineering 2019, Berlin, Germany, September 23-26, 2019*, páginas 7:1–7:10. ACM, 2019. https://doi.org/10.1145/3342558.3345398. 25

[67] Oliveira Fernandes, Eliézer de e José Leonardo Oliveira Lima: *Estudo da tecnologia blockchain para aplicação na emissão, arquivamento e validação de diplomas de graduação em formato digital*. Anais do Simpósio de Tecnologia da Informação e da Semana de Iniciação Científica do Curso de Sistemas de Informação (ISSN em fase de registro), páginas 36–40, 2019. 25, 87

[68] Costa, Rostand, Daniel Faustino, Guido Lemos, Ademir Queiroga, Cláudio Djohnnatha, Felipe Alves, Jordan Lira e Mateus Pires: *Uso não financeiro de blockchain: Um estudo de caso sobre o registro, autenticação e preservação de documentos digitais acadêmicos*. Em *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações*, Porto Alegre, RS, Brasil, 2018. SBC. https://portaldeconteudo.sbc.org.br/index.php/wblockchain/article/view/2356. 25, 80

[69] Ibarz, Juan Carlos Cruellas: *Bringing json signatures to etsi ades framework: Meet jades signatures*. Computer Standards & Interfaces, 71:103434, 2020, ISSN 0920-5489. https://www.sciencedirect.com/science/article/pii/S0920548919300960. 26, 34

[70] Palma, Lucas M., Martín A. Gagliotti Vigil, Fernando Lauro Pereira e Jean Everson Martina: *Blockchain and smart contracts for higher education registry in brazil*. Int. J. Netw. Manag., 29(3), 2019. https://doi.org/10.1002/nem.2061. 26, 80

[71] *Electronic signatures and infrastructures (esi); xades digital signatures;*, 2016. https://www.etsi.org/deliver/etsi_en/319100_319199/31913201/01.01.00_30/en_31913201v010100v.pdf. 33, 34, 35, 50

[72] *Europass framework for digitallysigned credentials*, 2018. https://ec.europa.eu/futurium/en/system/files/ged/europass_background-info_framework-digitally-signed-credentials.pdf. 34, 35, 86, 87, 88

[73] Finandhita, A e I Afrianto: *Development of e-diploma system model with digital signature authentication*. Em *IOP Conference Series: Materials Science and Engineering*, volume 407, página 012109, 2018. 34, 35, 59, 60

[74] Pinkas, D, N Pope e J Ross: *Rfc 5126-cms advanced electronic signatures (cades)*. Internet Engineering Task Force (IETF), 2008. 35

[75] Funk, Alexander, Luigi Rizzo, Juan Carlos Cruellas Ibarz, Andrea Röck e Andrea Caccia: *Etsi en 319 142-1 v1. 1.1. pades digital signatures. part 1: Building blocks and pades baseline signatures*. EUROPEAN STANDARD, 2016. 35

[76] Educação, Ministério da: *Portaria nº 330, de 05 de abril de 2018*, 2018. http://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/9365055/do1-2018-04-06-portaria-n-330-de-5-de-abril-de-2018-9365051. 37

[77] *InstruÇÃo normativa nº 1, de 15 de dezembro de 2020*, 2020. http://portal.mec.gov.br/diplomadigital/arquivos/in_01_15122020.pdf. 38, 39, 40, 41, 42, 43, 47, 48, 50, 51

[78] Gu, Yue-sheng, Meng-tao Ye e Yong Gan: *Web services security based on XML signature and XML encryption*. J. Networks, 5(9):1092–1097, 2010. https://doi.org/10.4304/jnw.5.9.1092-1097. 39

[79] Mendel, Florian, Norbert Pramstaller, Christian Rechberger e Vincent Rijmen: *Analysis of step-reduced SHA-256*. IACR Cryptology ePrint Archive, 2008:130, 2008. http://eprint.iacr.org/2008/130. 42

[80] Goldberg, Kevin Howard: *XML: Visual quickstart guide*. Peachpit Press, 2010. 48

[81] Bartel, Mark, John Boyer, Barb Fox, Brian LaMacchia e Ed Simon: *Xml signature syntax and processing version 1.1*. Signature, 6:1, 2013. 48, 49, 50, 52

[82] *Exclusive xml canonicalization version 1.0*, 2002. `https://www.w3.org/TR/xml-exc-c14n/`. 50

[83] *Xml advanced electronic signatures (xades)*, 2003. `https://www.w3.org/TR/XAdES/`. 50

[84] ICP-BRASIL, ASSINATURAS DIGITAIS NA: *Visão geral sobre assinaturas digitais na icp-brasil*. ICP-Brasil, 2015. 52

[85] Adams, Carlisle, Pat Cain, Denis Pinkas e Robert J. Zuccherato: *Internet X.509 public key infrastructure time-stamp protocol (TSP)*. RFC, 3161:1–26, 2001. `https://doi.org/10.17487/RFC3161`. 52

[86] Sousa Ribeiro, Alysson de, Edna Dias Canedo e Sérgio Antônio Andrade de Freitas: *An implementation of the oauth 2.0 for an enterprise service bus*. Em *ICCSA (1)*, volume 10960 de *Lecture Notes in Computer Science*, páginas 469–484. Springer, 2018. 56

[87] Armstrong, Joe, Robert Virding e Mike Williams: *Concurrent programming in ERLANG*. Prentice Hall, 1993, ISBN 978-0-13-285792-5. 56

[88] *Demoiselle signer*, 2020. `https://www.frameworkdemoiselle.gov.br/v3/signer/index.html`. 57, 58

[89] luisgoncalves: *Xades 4 java*. xades4j, 2020. `https://github.com/luisgoncalves/xades4j`. 60

[90] Cooper, David, Stefan Santesson, S Farrell, Sharon Boeyen, Rusell Housley e W Polk: *Rfc 5280: Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile*. IETF, May, 2008. 67

[91] *Xml advanced electronic signatures (xades)*, 2003. `https://www.w3.org/TR/XAdES/#Syntax_for_XAdES_T_form_The_SignatureTimeStamp_element`. 68

[92] Marcilio, Diego, Rodrigo Bonifácio, Eduardo Monteiro, Edna Dias Canedo, Welder Pinheiro Luz e Gustavo Pinto: *Are static analysis violations really fixed?: a closer look at realistic usage of sonarqube*. Em *ICPC*, páginas 209–219. IEEE / ACM, 2019. 70

[93] Marchand-Melsom, Alexander e Duong Bao Nguyen Mai: *Automatic repair of OWASP top 10 security vulnerabilities: A survey*. Em *ICSE (Workshops)*, páginas 23–30. ACM, 2020. 70

[94] Villamizar, Hugo, Amadeu Anderlin Neto, Marcos Kalinowski, Alessandro Garcia e Daniel Méndez: *An approach for reviewing security-related aspects in agile requirements specifications of web applications*. Em Damian, Daniela E., Anna Perini e

Seok-Won Lee (editores): *27th IEEE International Requirements Engineering Conference, RE 2019, Jeju Island, Korea (South), September 23-27, 2019*, páginas 86–97. IEEE, 2019. https://doi.org/10.1109/RE.2019.00020. 71

[95] Guamán, D., F. Guamán, D. Jaramillo e M. Sucunuta: *Implementation of techniques and owasp security recommendations to avoid sql and xss attacks using j2ee and ws-security.* Em *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, páginas 1–7, 2017. 71

[96] *Owasp zed attack proxy (zap)*, 2020. https://www.zaproxy.org/. 71

[97] Riadi, Imam e Pradana Ananda Raharja: *Vulnerability analysis of e-voting application using open web application security project (owasp) framework.* 71, 74

[98] Cika, P. e V. Clupek: *Stress tester and network emulator in apache jmeter.* Em *2019 PhotonIcs Electromagnetics Research Symposium - Spring (PIERS-Spring)*, páginas 3722–3726, 2019. 71

[99] Mishra, Shailendra, Majed A Alowaidi e Sunil Kumar Sharma: *Impact of security standards and policies on the credibility of e-government.* Journal of Ambient Intelligence and Humanized Computing, páginas 1–12, 2021. 74

[100] Ibrahim, Omer Imad Taj aldeen *et al.*: *A Comparative Study of Web Service Testing Tools: Apache JMeter & Locust.* Tese de Doutoramento, University of Science and Technology, 2018. 79

[101] Brzica, Hrvoje, Boris Herceg e Hrvoje Stančić: *Long-term preservation of validity of electronically signed records.* 2013. 80

[102] Sudaryono, Sudaryono, Qurotul Aini, Ninda Lutfiani, Firman Hanafi e Untung Rahardja: *Application of blockchain technology for ilearning student assessment.* IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 14(2):209–218. 80

[103] Panachev, Anton, Vladislav Shcherbitsky e Maxim A Medvedev: *Application of blockchain technologies and game approach in the educational process of universities.* Em *AIP Conference Proceedings*, volume 2333, página 100004. AIP Publishing LLC, 2021. 80

[104] Liang, Xiao e Shiyuan Xu: *Student performance protection based on blockchain technology.* Em *Journal of Physics: Conference Series*, volume 1748, página 022006. IOP Publishing, 2021. 80

[105] Meng, Nan e Shunxiang Zhang: *University education resource sharing based on blockchain and ipfs.* Em Atiquzzaman, Mohammed, Neil Yen e Zheng Xu (editores): *Big Data Analytics for Cyber-Physical System in Smart City*, páginas 1808–1813, Singapore, 2021. Springer Singapore, ISBN 978-981-33-4572-0. 80

[106] *An digital política de preservação digital*, 2012. http://siga.arquivonacional.gov.br/images/an_digital/and_politica_preservacao_digital_v1.pdf. 81, 82, 83

[107] Stančić, Hrvoje: *Long-term preservation of digital signatures*. Em *Tehnični in vsebinski problemi klasičnega in elektronskega arhiviranja*, 2016. 82

[108] *A preservaÇÃo de documentos arquivÍsticos digitais: Diretrizes para organizaÇÕes*, 2017. http://siga.arquivonacional.gov.br/images/publicacoes/diretrizes_preservador_digital.pdf. 83, 84

[109] Ranjbar, Alireza, Miika Komu, Patrik Salmela e Tuomas Aura: *An sdn-based approach to enhance the end-to-end security: SSL/TLS case study*. Em Oktug, Sema, Mehmet Ulema, Cicek Cavdar, Lisandro Zambenedetti Granville e Carlos Raniery Paula dos Santos (editores): *2016 IEEE/IFIP Network Operations and Management Symposium, NOMS 2016, Istanbul, Turkey, April 25-29, 2016*, páginas 281–288. IEEE, 2016. https://doi.org/10.1109/NOMS.2016.7502823. 85

[110] Fang, Weidong, Wei Chen, Wuxiong Zhang, Jun Pei, Weiwei Gao e Guohui Wang: *Digital signature scheme for information non-repudiation in blockchain: a state of the art review*. EURASIP J. Wirel. Commun. Netw., 2020(1):56, 2020. https://doi.org/10.1186/s13638-020-01665-w. 86

[111] Drescher, Daniel: *Protecting the Data Store*, páginas 135–143. Apress, Berkeley, CA, 2017, ISBN 978-1-4842-2604-9. https://doi.org/10.1007/978-1-4842-2604-9_16. 86

[112] Palma, Lucas M, Martín AG Vigil, Fernando L Pereira e Jean E Martina: *Blockchain and smart contracts for higher education registry in brazil*. International Journal of Network Management, 29(3):e2061, 2019. 87

[113] *eidas made easy!*, 2017. https://ec.europa.eu/digital-single-market/en/news/quickstart-guide-eidas-made-easy. 87, 88