CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Felipe Dias da Silva Cabral

PROTÓTIPO DE UM MÓDULO DE ASSINATURA DIGITAL PARA O PROTOCOLADOR DIGITAL DE DOCUMENTOS ELETRÔNICOS DO CEULP/ULBRA

Palmas – TO

2018

Felipe Dias da Silva Cabral

PROTÓTIPO DE UM MÓDULO DE ASSINATURA DIGITAL PARA O PROTOCOLADOR DIGITAL DE DOCUMENTOS ELETRÔNICOS DO CEULP/ULBRA

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Jackson Gomes de Souza.

Palmas – TO

2018

Felipe Dias da Silva Cabral

PROTÓTIPO DE UM MÓDULO DE ASSINATURA DIGITAL PARA O PROTOCOLADOR DIGITAL DE DOCUMENTOS ELETRÔNICOS DO CEULP/ULBRA

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Jackson Gomes de Souza.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Prof. M.e Jackson Gomes de Souza

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof.a M.e Cristina D'Ornellas Filipakis Souza

Centro Universitário Luterano de Palmas – CEULP

_____

Prof.a M.e Madianita Bogo Marioti

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2018

AGRADECIMENTOS

Agradeço a Deus pois o seu amor me fez mais forte.

Agradeço minha família, em especial minha mãe (Paulina) que fez de tudo para que eu conseguisse alcançar meus objetivos.

Agradeço meus amigos e colegas de trabalho, por todo esforço e dedicação e auxilio nas horas difíceis deste trabalho e na reta final da faculdade.

Agradeço até minha cachorra, por me acompanhar nas noites em claras enquanto estudava.

Agradeço a toda equipe de professores desta universidade, por todo empenho e sabedoria.

Fim..

“Que Deus lhe dê em dobro, tudo que me desejas”.

(Autor Desconhecido).

RESUMO

CABRAL, Felipe Dias da Silva. **Protótipo de um módulo de assinatura digital para o protocolador digital de documentos eletrônicos do CEULP/ULBRA**. 2018. 65 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistema de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2018.

Este trabalho apresenta uma proposta para desenvolvimento de um módulo de assinatura digital para o projeto de um Protocolador Digital de Documentos Eletrônicos (PDDE) para o CEULP/ULBRA, que tem o objetivo de garantir segurança no que tange ao processo de protocolação de documentos por meio da utilização de recursos de criptografia, certificados digitais e assinatura digital dos documentos. Este trabalho apresenta conceitos do processo de protocolação, criptografia, assinatura digital e certificado digital. A protótipo desenvolvido visa permitir a assinatura do documento utilizando certificado digitais e foi implementada com o uso de ferramentas para desenvolvimento web como Laravel, PostgreSQL, HTML e CSS e a API da Lacuna, a empresa que fornece os mecanismos de obtenção do certificado instalado no sistema operacional. O desenvolvimento do trabalho envolveu as etapas de coleta de requisitos para obtenção das funcionalidades do sistema, criação de uma documentação básica do sistema, design do projeto e finalmente a implementação utilizando as ferramentas previamente indicadas. Foi elaborado uma visão geral para entender o funcionamento da comunicação da aplicação com a API da Lacuna e assim foi desenvolvida a aplicação SigDoc, que permite o usuário efetuar o seu cadastro, cadastrar um documento e assinar o documento utilizando o certificado digital. A aplicação permite também a validação do usuário desde da assinatura do documento até a validação do documento que foi impresso, pois a aplicação permite validação acessando o sistema e digitando código de verificação.

Palavras-chave: Protocolação, Criptografia, Certificado Digital, Assinatura Digital

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CA	Autoridade Certificadora
CEULP	Centro Universitário Luterano de Palmas
GED	Gerenciador Eletrônico de Documento
HTML	HyperText Markup Language
IPSEC	IP Security Protocol
MD5	*Message-Digest algorithm 5*
ORM	*Object-relational mapping*
PDDE	Protocolador Digital de DocumentosEletrônicos
PGP	*Pretty Good Privacy*
PHP	*Hypertext Pre processor*
RSA	*Rivest-Shamir-Adleman*
SHA	*Secure Hash Algorithm*
SSL	*Secure Sockets Layer*
SSH	*Secure Shell*
S/MIME	*Secure/Multipurpose Internet Mail Extensions*
TLS	*Transport Layer Security*
ULBRA	Universidade Luterana do Brasil
URL	Uniform Resource Locator

SUMÁRIO

1 INTRODUÇÃO

Todas as organizações governamentais, as instituições públicas e as privadas utilizam documentos em seus processos administrativos. Existem diversas formas para protocolação de documentos, uma vez que estas vão de acordo com a estrutura interna de cada organização. Uma forma de gerenciar e ter controle de todos os documentos registrados é por meio de um Gerenciador Eletrônico de Documento (GED). Segundo Setti (2008), o GED é um sistema que agiliza o acesso a informação e soluciona os problemas de organização, acúmulo de papel e perda de documentos.

O GED possui inúmeras vantagens, mas destaca-se a redução de custo, o aumento da produtividade do trabalho e diminuição do volume de documentos físicos. Entretanto, existem algumas desvantagens devido as constantes mudanças de mídia, que acarretam a não garantia da autoria do documento e a ausência do não-repúdio. Ou seja, não visa garantir que o autor negue ter criado e assinado o documento, possibilitando com que este possa sofrer alterações sem conhecimento do autor. Portanto, para Macedo (2003), a finalidade do GED é gerenciar o ciclo de vida de um documento desde sua constituição até o seu arquivamento.

Em uma Instituição de Ensino, como o CEULP/ULBRA, setores lidam frequentemente com a necessidade de receber documentos, registrar o recebimento e, por vezes, entregar um recibo de tal ocorrência para quem encaminhou o documento. Procedimentos deste tipo podem ocorrer de forma manual (uma pessoa entregar um documento pessoalmente) ou digital (uma pessoa enviar um documento via e-mail, sistemas institucionais, e etc). Embora estes procedimentos funcionem a certo nível de aceitação, eles não conseguem fornecer garantias de verificações como:

a) quem entregou o documento é realmente quem diz ser?
b) quem registrou o documento é realmente quem diz ser?
c) quem registrou o documento pode afirmar que é possível garantir a integridade do documento?

No contexto do CEULP/ULBRA essa é uma realidade diária, evidente em documentos impressos e eletrônicos. Devido à maior disponibilidade de canais de distribuição (como e-mail, compartilhamento de arquivos etc.) para documentos eletrônicos, estes podem demandar ainda mais necessidade de garantias de segurança. Frente a este contexto, foi pensando em como desenvolver um módulo de assinatura do Protocolador Digital de Documentos Eletrônicos (PDDE) para o CEULP/ULBRA utilizando certificado digitais e criptografia assimétrica.

Um PDDE permite aprimorar a organização geral corporativa, aumentar a produtividade dos funcionários, economizar tempo, além de manter e garantir quem é o emissor e o destinatário, a data e horário de emissão, a validade do documento e que não sofreu o não-repúdio. Mas para tal efeito, visando garantir todos esses benefícios, é necessário introduzir um módulo de segurança que utiliza certificados digitais, e algoritmos criptográficos para geração de assinatura digitais, no PDDE do CEULP/ULBRA.

Tendo em vista isso, é necessário entender o que é o certificado digital. Segundo o Instituto Nacional de Tecnologia da Informação ITI (2018), os certificados digitais funcionam como uma identidade virtual que permitem a identificação segura e inequívoca do autor de uma transação feita por meio eletrônico. Isto é, essa tecnologia, além de agregar mais segurança no processo de protocolação, aumenta a organização, a eficiência e reduz o tempo de assinatura e protocolação dos documentos. Consequentemente, a adoção dessa tecnologia em um PDDE aumenta a segurança no processo, pois garante os autores, as datas, o não-repúdio e dentre outras características que vão ser abordadas neste trabalho.

Há um projeto de um PDDE em desenvolvimento no CEULP/ULBRA. Assim, este trabalho tem como objetivo geral fornece uma extensão desse projeto, adicionando recursos de certificados digitais e algoritmos criptográficos para assinar documentos protocolados no PDDE. Para contemplar o objetivo geral, utilizou-se um mecanismo para comunicação com o *browser* para obtenção do certificado do usuário. Com isso, foi possível gerar assinatura digital de um documento eletrônico protocolado pelo usuário. Após esse processo pode-se também analisar a validação do documento eletrônico e da sua assinatura e por fim integrar o módulo de assinatura eletrônica no PDDE do CEULP/ULBRA.

À vista disso, é válido ressaltar que o uso da assinatura digital em documentos fornece diversas vantagens, tais como: a verificação do documento em qualquer lugar; redução de custos administrativos em decorrência da eliminação de documentos em papel; torna-se desnecessário colher assinatura manual e a verificação da mesma. Todavia, esses benefícios somente poderão ser alcançados com a implementação do módulo de assinatura digital no PDDE.

Esta monografia apresenta a seguinte estrutura: a seção 2 contém o referencial teórico, uma revisão sobre documentos eletrônicos, protocolação digital, criptografia e certificados digitais, conceitos e técnicas que envolvem todo o processo deste trabalho, a seção 3 dispõe os materiais e métodos para o trabalho. Os materiais são os recursos técnicos utilizados para o desenvolvimento do trabalho. Já os métodos contemplam a metodologia utilizada para desenvolvimento do módulo de assinatura utilizando certificados digitais. A seção 4 apresenta

os resultados, desafios e discussões sobre as análises realizadas e tendo como resultado maior um protótipo funcional da aplicação, por fim, são apresentadas as considerações finais sobre o desenvolvimento do trabalho e possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1. DOCUMENTOS ELETRÔNICOS

Documentos eletrônicos são criados ou armazenadas de forma digital, ou seja, por máquinas eletrônicas, que possuem software para elaboração de documentos. Esmec (2009) afirma que documento eletrônico é todo registro físico de um documento por meio de um suporte eletrônico. Diante dessa afirmação, é importante ressaltar que o documento eletrônico deve possuir a capacidade de armazenamento de informações de forma que impeça alteração do documento sem autorização da origem.

Segundo a CONARQ (2018), todo documento eletrônico não é digital, mas um documento digital é eletrônico. Isto é, o documento digital é composto por bits (dados binários) e é acessado por meio de um sistema computacional. Todavia, um documento eletrônico pode ser acessado e interpretado por meio de um sistema computacional, mas também por filmadoras e aparelhos de leitura de vídeos, entre outros. Um exemplo que pode ser citado para melhor entendimento sobre essa afirmação é: um filme em VHS ou uma música em fita cassete são exemplos de documento eletrônico; um texto em PDF e um vídeo em AVI são exemplos de documentos digitais. Sendo assim, todo documento digital é eletrônico, mas nem todo documento eletrônico é digital.

No contexto de uma organização (pública ou privada, secretaria de governo estadual ou uma instituição de ensino) documentos eletrônicos são utilizados de formas específicas e, geralmente, há um processo de protocolação que registra os trâmites de um documento (entrega, devolução, repasse a outros setores, por exemplo). Pode-se afirmar que a protocolação fornece certas garantias e proteções ao documento e às pessoas envolvidas.

2.1.1 Segurança em Documentos Eletrônicos

Os documentos eletrônicos devem garantir a mesma segurança que os documentos tradicionais ou documentos em papéis possuem, tais como autenticidade, integridade, tempestividade que são adquiridas no processo criptográfico do documento.

Esse processo criptográfico é usado na assinatura digital e permite verificar se o autor foi alterado, garantindo a autenticidade. Consequentemente, a integridade é garantida com a possibilidade de verificação se o conteúdo do documento foi alterado. Sendo assim, neste processo a tempestividade é garantida a partir da identificação e preservação da data em que o documento foi declarado e/ou assinado.

Carmo e Santos (2009) afirmam que há três pilares na segurança de documentos eletrônicos: **assinatura eletrônica, certificado digital (acesso seguro) e protocolação digital**, conforme apresentado na Figura 1:

Figura 1 - Estrutura de segurança na internet, baseada no tripé: acesso seguro, protocolação digital e assinatura

Fonte: Nobre (2007)

Conforme ilustra a Figura 1 à cada pilar está relacionado com os conceitos de segurança:

a) **Assinatura eletrônica**: garante a autoria do documento;
b) **Acesso seguro**: o conceito de acesso seguro está relacionado a confidencialidade e sigilo do documento.
c) **Protocolação digital**: é garantido todo o processo de autenticidade, integridade, irretratabilidade, irrefutabilidade, tempestividade e irretroatividade:

Para entendimento destes conceitos de segurança ilustrados na Figura 1, o restante desta seção apresenta e exemplifica estes conceitos.

Carmo e Santos (2009) afirmam que **assinatura eletrônica** é responsável pelo algoritmo que garante a autenticidade, a autoria do documento utilizando os conceitos de chaves assimétricas. Resumidamente, este conceito requer duas chaves sendo uma secreta e uma pública, utilizadas no processo de criptografia.

Segundo Stallings (2015), a **autenticidade** é a confiança na validação do processo que foi transmitido, isto é, garantir a veracidade ou legitimidade da fonte da informação e assegurar que o usuário é quem diz ser. Um exemplo de autenticidade, no mundo físico, para garantir a autenticidade de um documento, é necessário um visto em todas as páginas e uma assinatura rubrica reconhecida em cartório.

A **confidencialidade** e o **sigilo** garantem o **acesso seguro**. Segundo Stallings (2015), o controle de acesso à informação somente para pessoas autorizadas garante o sigilo e a confidencialidade. Isto é, em um sistema contendo um mecanismo de controle de acesso a

usuários com permissões de acesso, a confidencialidade garante que alguma informação não autorizada não seja divulgada para outros usuários do sistema.

O processo de **protocolação digital** inclui diversos conceitos de segurança em documentos eletrônicos, tais como:

- **integridade:** Segundo Stallings (2015), garante que o documento e os programas sejam modificados somente de forma específica e autorizada, ou seja, garante que o documento não foi alterado;
- **irretratabilidade:** Segundo Stallings (2015), impede que o remetente ou o destinatário neguem uma mensagem transmitida;
- **irrefutabilidade:** É a capacidade dos envolvidos em uma transação de provar posteriormente se algo aconteceu ou não, como nos casos de um empresário negar-se a comprovar alguma acusação. (ANDERSON, 2008).
- **tempestividade:** garante que um determinado evento eletrônico ocorreu em um determinado instante (CARMO E SANTOS, 2009).
- **irretroatividade:** o princípio da irretroatividade é que uma lei não pode ser aplicada em crimes que não foram cometidos antes dessa lei existir. (BOUCHET-SAULNIER, 2007). No caso, a irretroatividade, é o que não pode retroagir, ou seja, um documento ele não pode ser datado ou protocolado com data retroativa.

Portanto, um documento possui uma origem, uma data e hora de criação, autores e dentre outros fatores e características. Os conceitos apresentados garantem que todas essas informações estejam presentes no documento e que sejam válidas conforme as propriedades dos conceitos.

2.2 PROTOCOLAÇÃO DIGITAL DE DOCUMENTOS ELETRÔNICO

Ao serem utilizados documentos eletrônicos, uma organização pode adotar um sistema computacional para realizar a protocolação. Para isto, é necessário a utilização do PDDE que, conforme Fabro (2005), é composto por um software, uma base de dados para armazenamentos das chaves e/ou informações referentes ao usuário, e um software para fazer a interação necessária para validações dos documentos. Não compete ao escopo deste tópico explicar o processo interno do PDDE.

Para tanto, tem-se como exemplo o PDDE fabricado pela Bry, que é descrito como:

> (...) um servidor que gera protocolos digitais com o registro da data e hora em que foram emitidos procedente de uma fonte de tempo confiável. Estes recibos possibilitam comprovar o recebimento de documentos eletrônicos, sua integridade e o ordenamento cronológico do recebimento dos documentos, o que garante a tempestividade e confere validade jurídica aos processos digitais. (BRY, 2018)

Para entender o que é a protocolação, utiliza-se o contexto de como é realizado o protocolo dos documentos emitidos dentro da instituição do CEULP/ULBRA. Sendo assim,

no departamento de apoio ao aluno, ao solicitar um documento de Importo de Rende IR, é emitido um recibo contendo a data de solicitação do documento, o nome do funcionário e os dados do requerido. Neste recibo também é expressa a data em que o documento pode ficar pronto e isso tudo de forma impressa e manual, realizado pelos funcionários.

A ideia da protocolação digital é ter um serviço digital que aplique o carimbo de tempo. Anjo (2016) afirma que carimbo de tempo é um mecanismo que fornece a um documento o exato momento em que o evento ocorreu, baseando-se na hora legal brasileira, que é fornecido pelo Observatório Nacional.

Então, o carimbo de tempo, garante que o documento foi registrado e que possa ser confirmada por terceiros e também garanta que o documento e o registro tenham existido em um determinado período de tempo. Entretanto, além do carimbo de tempo, na protocolação deve conter o requisito de tempestividade. Costa (2003) afirma que tempestividade é a identificação e preservação da data do documento em que foram declarados. Por sua vez, a tempestividade comprova a entrada e saída de documentos no PDDE. Portanto, esse requisito é obtido através do processo de datação do documento. É válido ressaltar que o processo de datação não será abordado neste tópico, mas será abordado no tópico 2.2.2.

Consequentemente, como o PDDE funciona em um servidor que possui data e hora, os documentos vão ser registrados conforme os horários do servidor. Sendo assim, o carimbo de tempo ou datação eletrônica é uma forma segura de agregar a tempestividade e registrar a data e hora em que um documento que foi utilizado no PDDE (ITI, 2012).

2.2.1 Protocolação

A protocolação é o termo usado referente ao protocolo, que é um recibo que contém uma informação, data e hora de um documento que foi registrado em um determinado local. Moraes (2009) afirma que o serviço de protocolação garante a confiança e garante a segurança nas transações realizadas, assegurando que o documento se mantém incorruptível. Neste trabalho, a protocolação digital visa atender a necessidade de ter uma segurança para que um documento seja capaz de ser datado e assinado digitalmente de forma segura.

Sendo assim, a protocolação digital é um recurso que é utilizado por diversos órgãos e empresas, e também há muitos acadêmicos que criaram plataformas ou sistemas que atende esse recurso, como é o caso de Poffo (2010) que, em seu trabalho de conclusão de curso, criou um serviço de protocolação digital de documentos eletrônicos. Poffo (2010) afirma que a protocolação digital impossibilita protocolizar um documento com data retroativa com relação ao tempo (os mecanismos de datação serão apresentados na próxima seção).

O objetivo da protocolação digital é assegurar a existência de um determinado documento eletrônico em uma determinada data e hora.

Portanto, o PDDE visa também assegurar que determinado documento foi protocolado e utiliza recursos adicionais no protocolo utilizando técnicas de criptografia. Costa (2003) afirma que a importância da protocolação de documentos se torna evidente quando existe a necessidade de utilizar documentos eletrônicos por um longo período de tempo. Isto é, em uma organização que há muitos processos com papeis, entrada e saída de documentos, é possível o uso da protocolação digital devido a organização e controle de todos os documentos protocolados no PDDE.

No PDDE, o recurso de protocolação é indispensável porque não se pode confiar em um documento eletrônico, sem saber a sua origem e como foi a sua criação e protocolação antes da assinatura do documento. Ressalte-se que neste tópico não será abordado os conceitos de assinatura de documentos no PDDE, mas para que haja um entendimento sobre o assunto, pode-se entender que um simples processo de gerar um número com a data, horário e usuário pode ser uma assinatura referente ao documento inserido no PDDE, ou o simples fato de adicionar uma assinatura digitalizada juntamente com o documento no PDDE é uma forma de assinatura também.

2.2.2 **Datação**

Na protocolação digital de documentos estão inseridos dentro do contexto as formas de datação, sendo elas datação absoluta, que tem o tempo conforme a fonte confiável utilizada, e a datação relativa, que são vários documentos encadeados que são protocolados por ordem de chegada.

Moraes (2009) afirma que o objetivo principal da datação é assegurar que determinado documento exista em uma determinada data. Se a data de um documento condiz com o protocolo, isso pode garantir que o documento existiu em um determinado tempo.

De toda forma, adicionar data e hora em um documento não é uma tarefa trivial, pois o processo de datação deve atender os requisitos de segurança tais como: integridade, irretratabilidade, irrefutabilidade, tempestividade, irretroatividade, para ter um processo válido. Ninguém além do cliente poderá ter acesso ao conteúdo do documento eletrônico, pois deve garantir a integridade dos dados do documento além da operação ininterrupta do servidor de datação, que deve garantir o anonimato do cliente, e assegurar a datação em data e hora correta.

Há dois processos de datação: métodos de encadeamento linear e método da árvore sincronizada.

Costa (2003) explica que o encadeamento linear encadeia os resumos dos documentos recebidos do usuário junto com os recibos de datação e encaminha para um servidor e/ou banco de dados local para posterior auditoria. Ou seja, não se baseia na data e hora da aplicação ou do servidor, mas sim na relação entre os documentos. Portanto, a vantagem desse método é ser simples, mas tem uma deficiência em relação ao tempo necessário para verificar o tamanho da cadeia de encadeamento.

A árvore sincronizada também usa o conceito de encadeamento, mas em ordem temporal. No entanto, Costa (2003) afirma que a árvore sincronizada utiliza o conceito de não sequencial, que tem como o objetivo reduzir o tempo de busca dos documentos, além de reduzir o tempo de comparação entre os documentos. Portanto, a vantagem desse método em relação ao anterior é que esse permite agilizar o processo por não ser sequencial e permitir saltos ou rodadas, que é um período de tempo definido ou a quantidade máxima de solicitações de protocolização.

2.2.3 **Auditoria**

Para que se tenha um bom controle relativo do sistema do PDDE e uma validação referente aos processos envolvidos, o PDDE deve conter uma auditoria. Segundo João (2014), uma auditoria identifica todos os controles que governam o sistema e avalia sua efetividade. Além disso, listam e classificam todos os pontos fracos do sistema que está sendo auditado.

No processo do PDDE, a auditoria é em relação ao processo de recibo logo após a datação do documento, é valido ressaltar que o recibo contém as informações de protocolo do documento como data e hora do recebimento.

Pires e Dias (2013) afirmam que a auditoria revisa e avalia os controles internos informatizados e que tem como objetivo verificar a eficiência, constatar a eficácia e atestar a segurança. Conforme essa afirmação é possível notar que a auditoria tem como objetivo atestar a segurança do processo no PDDE, pois há uma verificação do uso dos recursos do sistema, isto é, verificar completamente todas as operações realizadas no PDDE.

Também é necessário compreender toda a avaliação dos resultados gerados verificando a validade dos recibos, os números de sequência, analisando se não há um caminho malicioso.

Consequentemente a forma de auditoria depende muito da metodologia utilizada no processo de datação e protocolação. Para cada metodologia, é utilizada uma forma de

auditoria diferente que são a auditoria para o método de encadeamento linear e para árvore sincronizada (PIRES; DIAS, 2013).

A exemplo disso, Pires e Dias (2013) criaram um sistema de auditoria para o PDDE. Como o PDDE possui dois métodos de datação, os autores então desenvolveram dois métodos de auditoria para cada método de datação sendo eles: encadeamento linear e arvore sincronizada.

- **Auditoria para o método de encadeamento linear:** são realizados 3 tipos de verificações:
    - **Verificar a qualidade de um recibo:** em cada link do encadeamento é verificado se o resumo do documento está presente na lista. Como o método de encadeamento linear tende a ser lento, se a lista for muito grande o retorno dessa verificação pode demorar muito tempo, causando lentidão no processo;
    - **Resolver disputa entre dois recibos:** Essa disputa entre dois recibos é necessária para garantir que realmente existe um documento primário. Ou seja, é feito um percurso no encadeamento verificando em qual posição está cada documento, com posse das posições é necessário fazer apenas a comparação deles para saber qual foi o primeiro documento protocolado;
    - **Verificar a autenticidade do Banco de Dados:** são verificados se os *links* calculados são iguais aos *links* contidos no recibo;
- **Auditoria para o método da árvore sincronizada:** nesta etapa são realizados 3 tipos de verificações:
    - **Verificar a validade de um recibo:** é feito o recálculo das informações do recibo, checa a assinatura digital e o seu ponto de confiança e consulta os certificados revogados;
    - **Resolver disputa entre dois recibos:** essa verificação é necessária para saber qual foi o primeiro recibo a ser protocolado no PDDE.
    - **Verificar a autenticidade do banco de dados:** consiste em explorar o banco de dados do PDDE verificando os recibos;

As operações realizadas no processo de auditoria no PDDE diferem conforme o método de datação do mesmo. Entretanto, o processo de verificação da auditoria é o mesmo para ambos métodos de datação.

2.3 CRIPTOGRAFIA

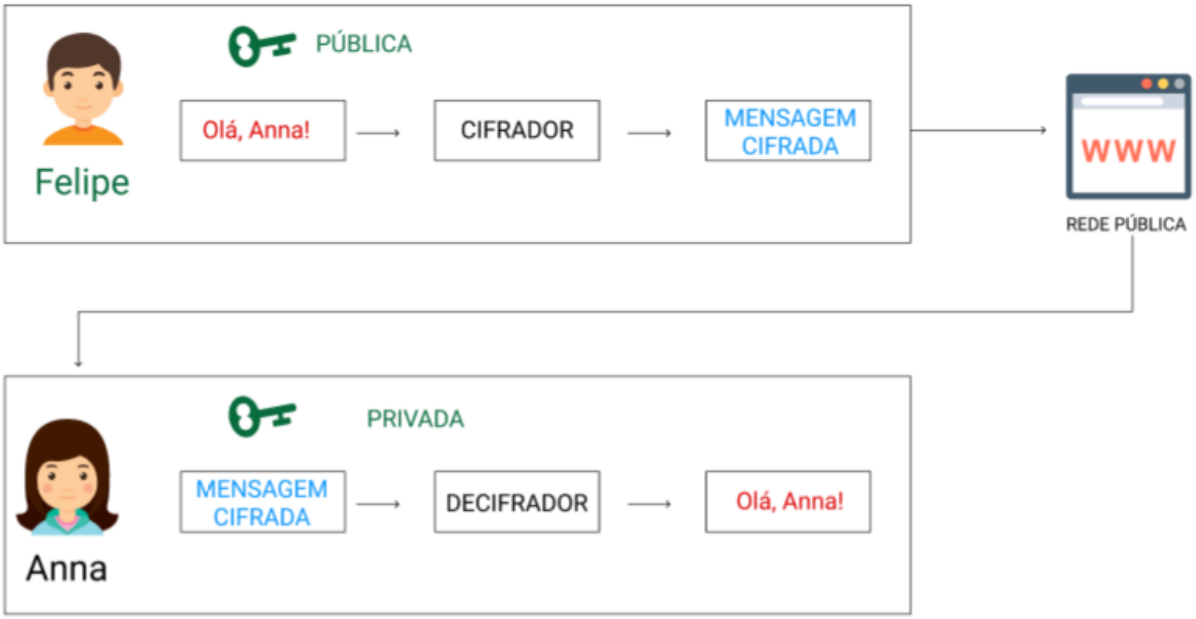
Segundo Galvão (2018), o termo criptografia trata de um conjunto de especificações que objetiva a codificação das informações do emissor e torna a informação ilegível e assim somente o receptor consiga decifrá-la. Com base nessa informação, para tornar a informação legível novamente só será possível se alguém possuir a chave de decifração.

No processo de criptografia, segundo Galvão (2018) são seguidas as etapas a seguir:

- **Elaboração do texto claro:** é a criação da mensagem a ser enviada ou criar um documento;
- **Cifração:** é o processo de conversão do texto claro para um texto ilegível e assim a mensagem não consegue ser lida por um usuário que não possui a chave de acesso;
- **Decifração:** é o processo de traduzir a cifra para obter o texto claro original, sendo assim, é necessário que usuário tenha a chave de acesso.

A figura 2 ilustra o processo de criptografia:

Figura 2- Processo de criptografia assimétrica



Na figura 2, nota-se que Felipe elabora um **texto claro**, criando uma mensagem para enviar para a Anna, para isto, utiliza uma chave para **cifrar** o documento, tornando ele ilegível para demais usuários ou terceiros que possam receber essa mensagem. A Anna ao receber a mensagem de Felipe, fará o processo de **decifração**, que utilizará uma chave acesso para poder ler a mensagem que Felipe enviou.

Stallings (2015) afirma que a criptografia tem quatro objetivos principais: confidencialidade, integridade, autenticação e não-repúdio ou irretratabilidade. As especificações de cada objetivo serão tratadas a seguir:

- **Confidencialidade:** apenas o destinatário é capaz de reproduzir ou retirar o conteúdo da mensagem cifrada;
- **Integridade:** verificar se a mensagem foi alterada durante o seu trâmite, sem que haja alguma interferência para comprometer a mensagem**;**
- **Autenticação:** garante que o emissor é quem diz ser;
- **Não-repúdio ou irretratabilidade:** garante que o emissor não negue a autoria de sua mensagem.

De acordo com Carmo e Santos (2009), na criptografia existem duas técnicas (ou categorias) denominadas simétrica e assimétrica. Apesar de serem técnicas diferentes, elas possuem um objetivo semelhante, mas com diferenças em sua composição e funcionamento. Este trabalho tem foco na Criptografia Assimétrica, como apresenta a próxima seção.

2.3.1 Criptografia Assimétrica

A criptografia assimétrica possui dois tipos de chaves, a pública e a privada. A chave privada é utilizada para decifrar (desencriptar) mensagens e a pública para cifrar (encriptar) uma mensagem ou conteúdo. Sendo assim, para uma pessoa enviar uma mensagem para alguém ela necessita apenas da chave pública do destinatário, que irá usar a chave privada para decifrar o conteúdo. Stallings (2015) afirma que as chaves são matematicamente associadas utilizando funções unidirecionais para codificar as informações.

Segundo Stallings (2015), os principais algoritmos de criptografia assimétrica são:

- **Diffie-Hellman:** criado em 1976, com o surgimento da criptografia assimétrica, é fundamentado em logaritmo discreto. A finalidade é permitir que os usuários troquem uma chave que pode ser utilizada subsequente na criptografia das mensagens.
- **ElGAmal:** criado em 1984 por Taher Elgamal, o algoritmo também é fundamentado em problema de logaritmo discreto, baseado no algoritmo de Diffie-Hellman. Hoje vulnerável a ataques, pois a segurança é baseada na dificuldade de calcular os logaritmos discretos.
- **Curvas Elípticas:** Criado em 1985 por NealKoblitz e V. S. Miller para ser usado com algoritmos Diffie-Hellman e ElGAmal. O cálculo criptografo é feito utilizando logaritmo discreto com fatorial de números primos. A vantagem que existe neste algoritmo é que o tamanho da chave é menor, mais curta, do que a chave gerada pelo RSA. Segundo o autor, ele está sendo mais utilizado para aplicações de assinatura digital.
- **RSA:** Criado em 1978 no MIT, desenvolvido por Ron Rivest, Adi Shamir e LenAdleman. O algoritmo é baseado na dificuldade de resolver o fatorial de dois

números primos. É hoje o mais utilizado em todo o mundo e possibilita a assinatura digital.

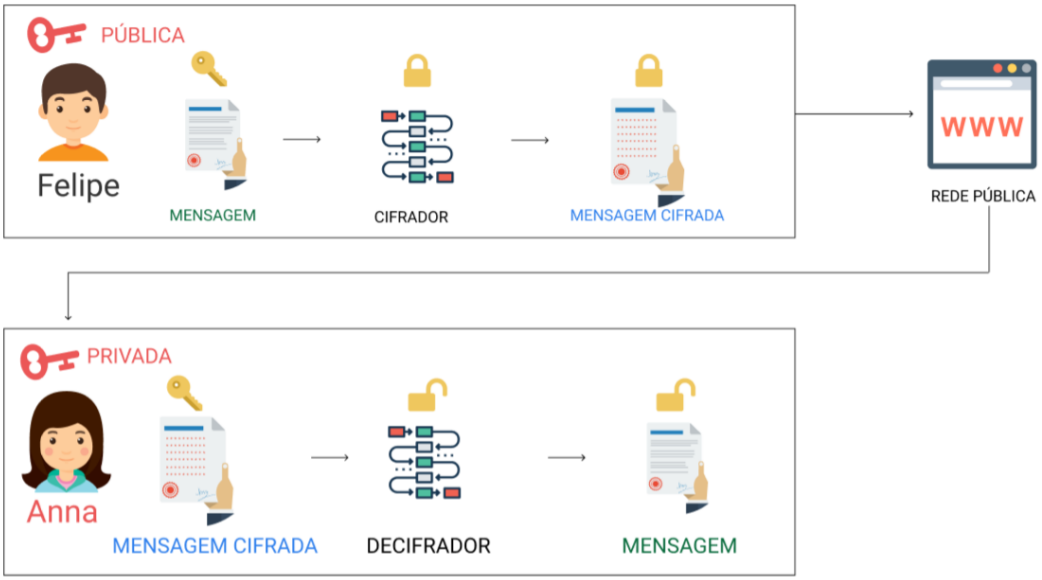
Na criptografia assimétrica é inviável descobrir a chave de decriptação tendo como conhecimento do algoritmo utilizado e a chave de encriptação. Stallings (2015) apresenta cinco elementos que são importantes nesse contexto:

- **Texto Claro:** mensagem legível utilizada como entrada no algoritmo;
- **Algoritmo de encriptação**: realiza variadas alterações no texto claro;
- **Chave pública e privada**: par de chaves selecionado, de forma que se uma é utilizada para encriptar a outra deve ser utilizada para decriptar;
- **Texto cifrado:** é a informação de forma ilegível, resultado da saída do algoritmo de encriptação; o resultado da transformação depende diretamente da chave fornecida no processo de encriptação;
- **Algoritmo de decriptação:** recebe o texto cifrada e a chave inversa que foi utilizado na cifragem e gera o texto claro.

Para garantir a confidencialidade no processo de criptografia assimétrica, a mensagem é assinada com a chave pública e a chave privada deve permanecer secreta com o remente e é somente utilizada para decifrar a mensagem cifrada. Também é possível utilizar a chave privada para cifrar uma mensagem e, neste caso, é garantida a autenticidade, visto que o destinatário irá utilizar a chave pública do remetente para decifrar a mensagem recebida.

A figura 3 ilustra como funciona o processo, com a utilização das duas chaves.

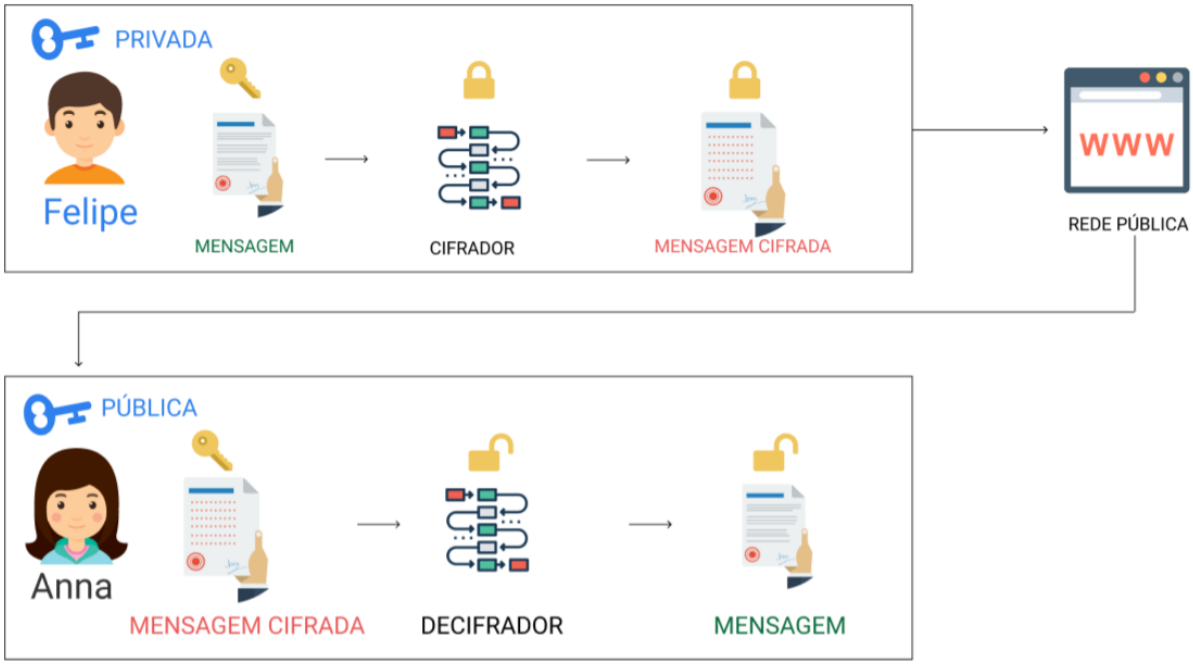
Figura 3 - Processo de criptografia assimétrica



No exemplo da Figura 3, nota-se que Felipe deseja enviar uma mensagem a Anna, portanto, o processo de cifra da mensagem será com a chave pública de Anna. Isso garante a confidencialidade do processo, visto que Anna só poderá decifrar a mensagem recebida com o uso da sua chave privada. Portanto, nenhuma outra chave poderá decifrar a mensagem assinada com a chave pública de Anna feita por Felipe.

Em outro caso, pensando na garantia da autenticidade, Felipe cifra a mensagem com a sua chave privada e envia para Anna. Para Anna decifrar a mensagem é necessária a chave pública de Felipe, portanto, Anna tem a certeza de quem foi o remetente da mensagem recebida. A figura 4 ilustra como funciona o processo, com a utilização das duas chaves.

Figura 4 - Processo de criptografia assimétrica garantindo a autenticidade



No exemplo da Figura 4, para Felipe garantir a sua autenticidade ele cifra a mensagem com sua chave privada e envia para a Anna. Sendo assim, a Anna só consegue visualizar a mensagem, no caso decifrar, com a chave pública de Felipe. Esse processo garante a autenticidade de origem, no qual, quem enviou a mensagem.

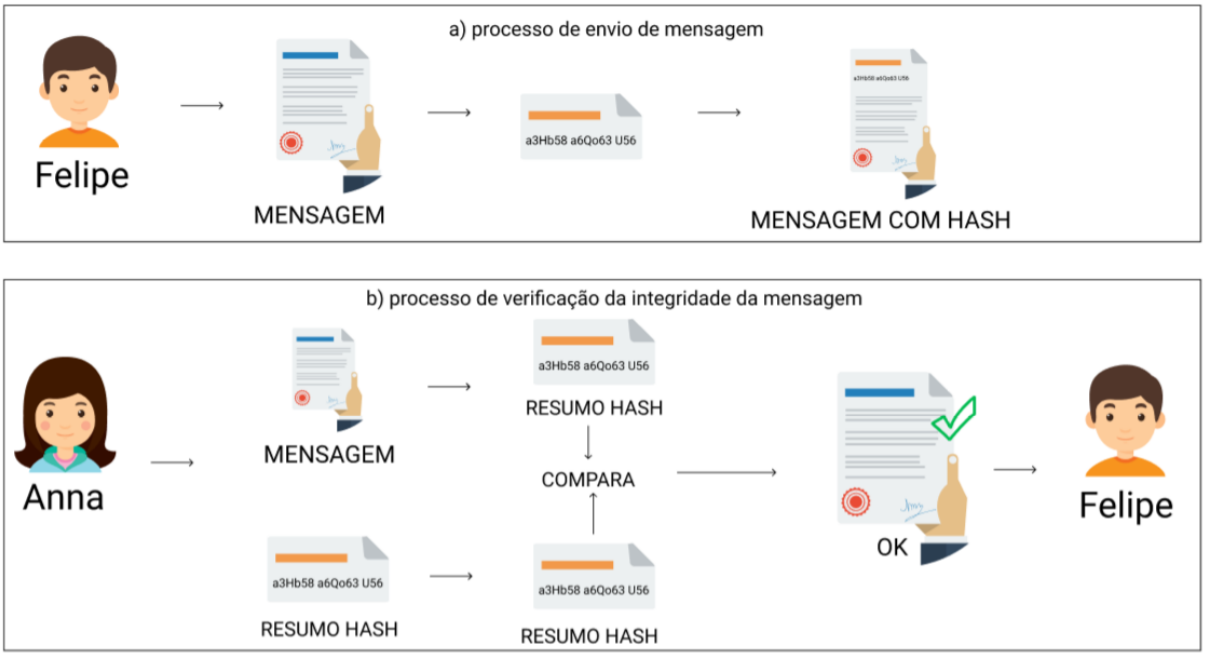
2.3.2 Funções Hash Criptográficas

Função *Hash* Criptográfica é uma função unidirecional criptográfica para resumir um conjunto de informações de entrada, ou seja, recebe uma entrada como *string* de tamanho variável e transforma em uma *string* de tamanho fixo (ARAÚJO et al., 2013). Segundo Núñez

(2014) se os valores de *hash* de duas mensagens são idênticos, espera-se que as duas mensagens sejam iguais.

O objetivo principal das funções *hash* de criptografia é a de manter a integridade dos dados, uma qualquer mudança no tamanho da mensagem, resulta em alta probabilidade uma mudança no código *hash*. A Figura 5 ilustra o conceito de integridade.

Figura 5 - Processo de envio e verificação da integridade



A Figura 5 apresenta duas etapas: a) processo de envio de mensagem e b) processo de verificação da integridade da mensagem. Para ficar claro o que ocorre nesses processos, as etapas serão apresentadas a seguir. Na primeira etapa ocorre o processo de envio de mensagem, onde Felipe envia mensagens para Anna:

1. Felipe produz a mensagem que deseja transmitir;
2. Felipe gera um *hash* da mensagem;
3. Felipe associa o *hash* da mensagem junto com a mensagem e envia como anexo para a Anna;

Na segunda etapa ocorre o processo de verificação de integridade, onde Anna verifica os dados recebidos com o de Felipe:

1. Anna recebe a mensagem por e-mail de Felipe e separa *hash* da mensagem;
2. Anna gera novamente o hash da mensagem de Felipe

3. Anna compara o hash recebido no e-mail com o hash gerado. Se caso os hash forem iguais, as mensagens não foram alteradas, garantindo a integridade, caso estejam diferentes ocorreu uma falha de integridade na mensagem;

Stallings (2015) afirma que o *SecureHash Algorithm* (SHA) é o mais importante e mais utilizado em funções *hash* criptográficas e que o MD5 é uma função que possui semelhanças com SHA-1. De acordo com Núñez (2014), o SHA-1 é uma função mais segura que o MD5. A seguir, algumas características desses algoritmos:

- **MD5:** foi criado por Ron Rivest (co-inventor do RSA) e produz 128 bits de hash.
- **SHA-1:** é baseado no MD5 e foi projetado pelo *NationalInstituteof Standards and Technology* (NIST) e pelo *National Security Agency* (NSA) para uso com o DSS (*Digital SignatureStandard*) produz 160 bits de hash.

H*ash*, conforme a definição da TecMundo (2018), é um algoritmo que transforma dados grandes em uma variável pequena com sequência de números e letras geradas a partir de uma função matemática. Segundo Stallings (2015) o SHA-1 foi lançado em 1995 onde seu algoritmo produz um *hash* de 160 bits e utiliza aritmética modular e operações binárias lógicas no seu processamento. Esses 160 bits é o tamanho do resumo da mensagem que tem 40 caracteres alfanuméricos e o tamanho do bloco tem 512 bits.O SHA-1 dá suporte a vários tipos de protocolo, tais como: TLS, SSL, PGP, SSH, S/MIME e IPSec e é considerado quase idêntico ao MD5 mas o SHA-1 gera uma quantidade maior de caracteres.

Oliveira (2002) explica que o MD5 é um algoritmo que foi projeto para ser rápido, simples e seguro. Stallings (2015) elucida que o MD5 tem a capacidade de gerar um hash com 128 bits de tamanho com 32 caracteres. Visto, que o SHA-1 possui 160 bits, o MD5 acabou perdendo muito o mercado. Oliveira (2002) enfatiza que pelo fato do MD5 ter uma menor quantidade de bits, o mercado tenha preferência por usar outros algoritmos criptográficos, mas era bastante utilizado em sistemas de login.

Há outros algoritmos da família SHA, tais como: SHA-2, SHA-3, SHA-512, mas que não são relevantes para o contexto deste trabalho. Na próxima seção são apresentados o conceito e as características de assinatura digital.

2.4 ASSINATURA DIGITAL

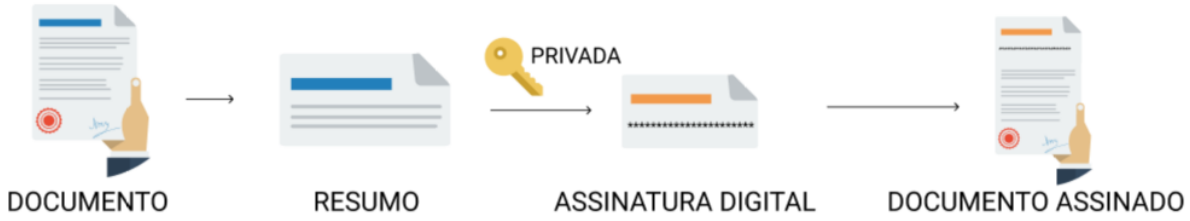
A Assinatura digital é gerada a partir de uma função hash em conjunto com a criptografia assimétrica. O resultado permite verificar, ao mesmo tempo, a identidade do assinante e a integridade da mensagem. Conforme Moraes (2009, p. 28), um documento assinado digitalmente garante ao destinatário que o documento não foi alterado ao ser enviado

(integridade) e ainda comprova a autoria do remetente (autenticidade). Isto é, a assinatura digital é um instrumento que dá uma certeza que o documento digital tem força probante, ou melhor, fornece credibilidade ao documento, pois permite a conferência da autoria e a integridade do documento.

ITI (2009) explica o processo simplificado de criação da assinatura digital, ilustrado pela figura 6:

1. é calculado o resumo da mensagem (documento) através de algoritmos criptográficos;
2. cifra este resumo na chave privada do emissor; o
3. resultado final é então concatenado com a mensagem original (documento);

Figura 6 - Processo de assinatura digital

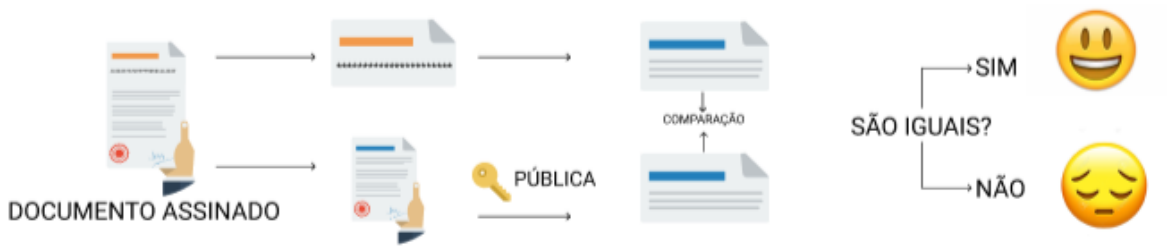


Fonte: Adaptado de ITI (2009)

O resultado desse processo gera a assinatura digital que concatenado ao documento original, produz o documento assinado digitalmente.

Por sua vez, de posse ao documento assinado o destinatário decifra o documento, que verifica o processo de assinatura e se é válido ou foi alterado. O processo é ilustrado na figura 7:

Figura 7 - Processo de verificação assinatura digital



Fonte: Adaptado de ITI (2009)

Conforme a Figura 7, o processo de verificação é composto pelos seguintes passos:

1. recebe o documento com assinatura digital concatenada;
2. faz a separação do documento e da assinatura digital;
3. gera o *hash* do documento;
4. decifra a assinatura digital utilizando a chave pública do usuário;
5. compara o *hash* do documento com o resultado da decifragem da assinatura digital;
6. se forem iguais a assinatura digital é válido, do contrário inválida, ou seja o documento não é o mesmo.

Então para verificar o processo de assinatura digital e se compõe o requisito de integridade, deve-se comparar o *hash* do documento original com o *hash* do documento que foi assinado. Para isso, utiliza-se a chave pública do usuário para fazer a decifragem do documento. Após isso, se o *hash* forem iguais, a assinatura está íntegra, se não, a assinatura está comprometida, ou seja, invalida.

2.5 CERTIFICADOS DIGITAIS

Os certificados digitais são documentos que contêm uma chave pública e são assinados por uma autoridade certificadora. Para ser mais específico o certificado digital é a identificação de uma pessoa física ou jurídica no âmbito digital garantindo os princípios de autenticidade, integridade, confiabilidade e não-repúdio a todos processos autenticados com o mesmo.

De acordo com (ITI, 2018)

> A certificação digital é uma ferramenta que permite que aplicações como comércio eletrônico, assinatura de contratos digitais, operações bancárias virtuais, iniciativas de governo eletrônico, entre outras, sejam realizadas. São transações feitas de forma virtual, ou seja, sem a presença física do interessado, mas que demandam identificação clara da pessoa que a está realizando pela internet**.**

No contexto deste trabalho o certificado digital vai proporcionar a assinatura digital de um documento para garantir no documento assinado os princípios de segurança que o certificado digital possui.

2.5.1 Autoridade Certificadora

Anjos (2016) redige um pouco sobre autoridade certificadora em seu trabalho de conclusão de curso, que tem o objetivo de incentivar o uso de criptografia e certificados digitais no escopo de desenvolvimento de sistemas. Anjos (2016) afirma que uma AC autoridade certificadora administra todo o ciclo de chaves públicas, mas que também uma

empresa pode ter sua própria AC, que podem emitir certificados digitais vinculando pares de chaves criptográficas ao respectivo titular.

Portanto, o objetivo da AC é emitir, distribuir, renovar, revogar e gerenciar certificados digitais, além de ser responsável pelo agendamento da data de expedição de um determinado certificado, assim sendo o seu dever providenciar a publicação de certificados revogados.

Sendo assim, uma autoridade certificadora ou autoridade de confiança é uma entidade pública ou privada, subordinada à hierarquia da ICP-BRASIL. Essa entidade representa a confiança do emissor e receptor de comunicação. A AC é responsável por criar e assinar digitalmente o certificado digital do assinante, onde o certificado emitido pela AC representa a identificação do titular, que possui um par único de chaves sendo ela pública e privada (ITI, 2018).

2.5.2 Formatos de Certificados Digitais

O certificado digital identifica um usuário virtualmente permitindo realizar transações eletrônicas. ITI (2018) afirma que com o certificado digital é possível realizar transações no comércio eletrônico, assinatura de contratos digitais, operações em bancos, autenticar aplicações do governo eletrônico. Isto é, são transações que são realizadas sem ter o usuário presente, mas há uma identidade virtual da pessoa e de forma segura.

Para tanto, para isto ocorrer, o mercado possui dois formatos de certificados digitais o A1 e o A3, conforme ilustrado na figura 8 e 9:

Figura 8 – Arquivos do certificado modelo A1

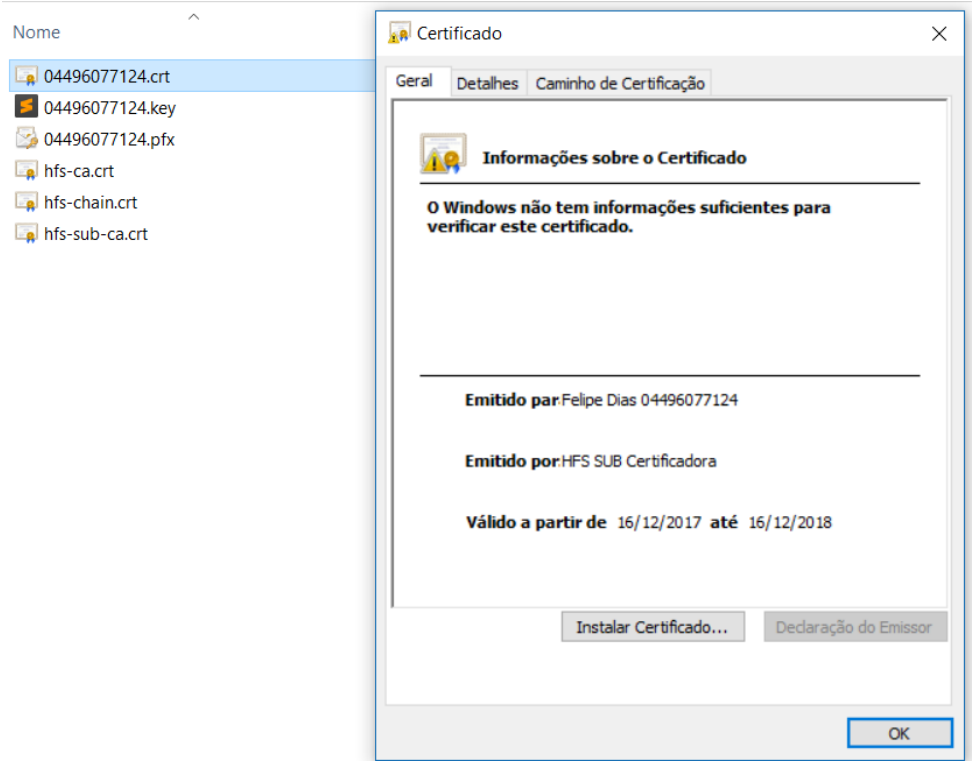


Figura 9 – Formato do modelo de certificado digital A3

Fonte: Site da Certisign (2018)

O modelo A1 é um arquivo digital instalado no computador e tem validade de 1 ano. Conforme o Serasa (2018), o modelo A1 pode ser utilizado para registrar livros fiscais, emitir nota fiscal, consultas nos sites do governo. A vantagem deste tipo de certificado digital é a possibilidade de instalação em várias máquinas.

Por outro lado, tem-se o modelo A3 que possui um modelo de token ou cartão e com as mesmas características e funcionalidades do modelo A1, mas a validade de uso do A3 é de

3 anos. A desvantagem é que não poderá fazer uso simultâneo em vários computadores, pois o modelo de cartão ou token não permite, devido está conectado em uma única máquina.

Para melhor entendimento das diferenças, a tabela abaixo apresenta as vantagens e desvantagens dos modelos de certificados digitais, conforme Sage (2018):

Tabela 1– Comparação das vantagens e desvantagens dos certificados A1 e A3

Características	**Certificado A1**	**Certificado A3**
Instalação e utilização simultânea em vários computadores	Sim	Não
Backup de Segurança	Sim	Não
Não necessita de software específico para instalação	Sim	Não
Custo	Baixo	Alto
Prazo de Validade	Curto	Médio
Rapidez nas transações	Sim	Não

Fonte: Sage (2018)

Na escolha do modelo do certificado devem ser levadas em consideração a usabilidade e a necessidade do usuário ou empresa, tendo em vista que o A1 é mais flexível e o A3 tem o custo alto, mas um prazo de validade maior, além de depender de algum software para instalação.

Além dos modelos disponíveis no mercado, existem os padrões de formatos de certificados digitais gerados na CA: X.509, PEM e PKCS.

Segundo a Amazon (2018), **Certificados X.509** usam uma chave pública para associar a uma identidade composta em um certificado. Conforme a Microsoft (2018) o certificado X.509 segue o padrão internacional ITU-T e todo certificado tem os seguintes dados:

- **Número da versão:** identifica o padrão da versão do X.509;
- **Número de série:** número que diferencia o certificado de outros usuários e que o identifica nos repositórios;
- **Chave pública proprietário:** identifica o sistema de criptografia e quais os parâmetros associados ao proprietário;
- **Identificação única do proprietário (DN):** Nome único para identificação do proprietário na internet;
- **Validade do certificado:** data de início e fim de validade;
- **Nome único do emissor:** Nome da Autoridade Certificadora (CA);
- **Assinatura digital do emissor:** assinatura com chave privada da CA; e
- **Identificação do algoritmo de assinatura:** a identificação do algoritmo utilizado pela CA.

Segundo a IBM (2018), **Certificados PEM** possuem as instruções "*BEGIN CERTIFICATE"* no começo e no final "*END CERTIFICATE"* e suporta vários certificados digitais, incluindo a chave privada de vários em um único arquivo. Os certificados PEM são codificados em Base64 e possuem extensões como .pem, .crt .cer, .key.

PKCS (*Public Key Cryptography Standards*) é um conjunto de padrões de criptografia de chave pública criado pela RSA. Existem doze padrões de certificados PKCS, alguns são codificados em Base64 com extensões p7b, p7c, outros em binário com extensões pfx, p12. Para exemplificar e ter um melhor entendimento sobre os doze padrões de certificados PKCS, Muzzi e Tamae (2013), esboçaram uma tabela a respeito das especificações de cada certificado conforme apresenta a Figura 10:

Figura 10 – Especificações dos padrões do formado de certificado PKCS

Número	Tema
1	• PKCS #1 – Como cifrar e assinar usando sistemas criptográficos RSA
3	• PKCS #3 – Padrão de Normalização de chave Diffie-Hellman
5	• PKCS#5 – Como cifrar com chaves secretas derivadas de um password
6	• PKCS#7 – Sintaxe de mensagens cifradas contendo assinaturas digitais
7	• PKCS #8 – Formato da informação de uma chave privada
8	• PKCS #9 – Tipos de atributos e sua utilização nas normas PKCS
9	• PKCS #10 – Requisição de certificados
10	• PKCS #11 – Define API de criptografia (Criptoki)
11	• PKCS #12 – Formato portável para armazenamento ou transporte (exportação/importação de certificados)
13	• PKCS #13 – Como cifrar e assinar com criptografia de curva elíptica
14	• PKCS #14 – Padrão para geração de números pseudo-random
15	• PKCS #15 – (está ainda em desenvolvimento... Tem em vista propor uma norma para armazenamento de credenciais em "token-based devices" (incluindo smart cards)

Fonte: Muzzi e Tamae (2013)

Segundo Muzzi e Tamae (2013), essas especificações são responsáveis por padronizar a criptografia utilizada no processo de criptografia RSA. Por fim, este modelo de certificado é a principal referência para criação de módulos de segurança fundamentado em criptografia assimétrica. Na próxima seção será apresentada a metodologia utilizada neste trabalho.

3 METODOLOGIA

Neste capítulo são descritos os instrumentos de apoio (materiais) e os procedimentos (métodos) utilizados para o desenvolvimento deste trabalho.

3.1 MATERIAIS

Os materiais utilizados no desenvolvimento deste projeto são divididos em duas etapas, que são: tecnologias e fontes bibliográficas.

3.1.1 Tecnologias

Para o desenvolvimento deste projeto na implementação do protótipo foi utilizado o framework Laravel, podendo ser agregado ao processo o framework front-end VueJs, o banco de dados PostgreSQL e as bibliotecas Web PKI, e REST PKI.

O Laravel é um framework em PHP, livre e de código aberto. Tendo como objetivo um trabalho de forma rápida e estruturada (OTWELL, 2018), o diferencial do Laravel está no sistema de rotas, as *blades*, o *eloquent* e *querybuilder*. O sistema de rotas é bastante rico, permite trabalhar com restrições de parâmetros com expressões regulares e agrupamentos de rotas.

O Laravel também conta com o *template engine* ou blades que é um sistema de *templates*, de views, que visa reduzir a quantidade de código PHP inseridos no HTML. E um dos principais benefícios da blade é o uso da herança e as seções, reuso de páginas. Por outro lado, temos o *eloquent,* o ORM do laravel, que faz a comunicação das tabelas do banco que são representadas através dos *models.*

Para armazenamento de dados na aplicação que será desenvolvida será utilizado o PostgreSQL, que é um banco de dados gratuito e de código-aberto, 100% comunitário e muito avançado e que fornece suporte ao modelo objeto-relacional.

Para leitura do certificados digitais e assinatura dos documentos são utilizadas as bibliotecas WEB PKI e REST PKI. Conforme a Lacuna (2018), a WEB PKI é um *plugin* para navegadores que permite o acesso aos certificados digitais do usuário a partir do *browser.* Além de possuir extensões que possibilitam a assinatura local e geração de chaves RSA. Já a REST PKI é um serviço em nuvem para realizar assinaturas digitais em qualquer tipo de linguagem de programação.

3.1.2 Fontes Bibliográficas

Os materiais utilizados para a elaboração e criação da produção bibliográfica foram de fontes como: artigos, dissertações, livros, publicações científicas, teses e textos técnicos referente ao assunto sobre criptografia, certificado digital e protocolador digital de documentos eletrônicos. Grande parte do material foi coletado no meio eletrônico.

3.2 MÉTODOS

O desenvolvimento do SigDoc compõe as etapas de Coleta de Requisitos, Documentação, Design do Projeto, Implementação e Entrega. Para definição dos elementos da plataforma, bem como entendimento geral de como é a comunicação de cada parte, foi necessário um modelo da visão geral do SigDoc, que será descrita ao longo do próximo tópico deste projeto. As etapas do trabalho podem ser vistas na Figura a seguir:

Figura 11 – Fluxograma das etapas de desenvolvimento do projeto

Conforme ilustra a Figura 11 o desenvolvimento foi realizado por meio das etapas:

1. **Coleta de Requisitos**: o primeiro passo no planejamento consistiu em definir as necessidades da parte interessada para atingir o objetivo deste projeto. Sendo assim, as entrevistas foram realizadas com o orientador na instituição do CEULP/ULBRA.
2. **Documentação**: a etapa de documentação consistiu no desenvolvimento de uma documentação técnica, contendo informações de todo escopo do projeto, diagramas de caso de uso e diagramas de classe para melhorar o entendimento da aplicação.
3. **Design do Projeto**: nesta etapa foi criado um *mockup* referente a aplicação, atendendo os requisitos coletados na etapa de planejamento, com o design do

projeto e com a documentação, a etapa de implementação fica mais clara sobre o funcionamento do projeto, evitando falhas ou falta de requisitos.

4. **Implementação do software web:** nesta etapa foi iniciada a implementação do software web para protocolação, assinatura e validação de documentos eletrônicos. A aplicação será desenvolvida utilizando o *framework Lavarel*.

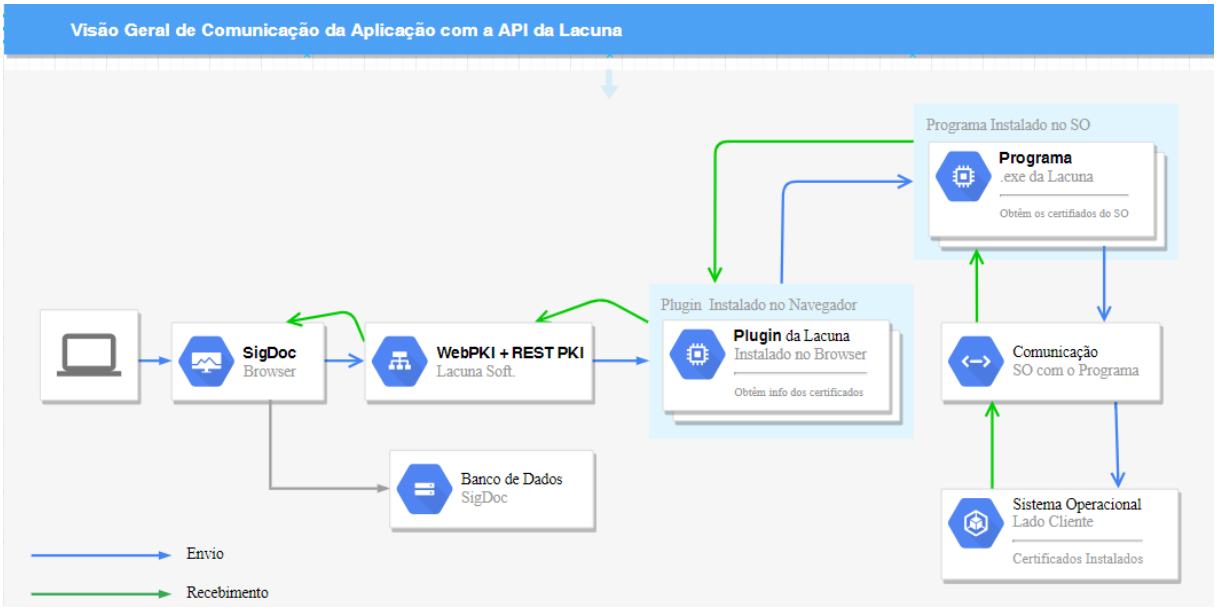
4 RESULTADOS

Este capítulo contém informações sobre os resultados obtidos na realização deste trabalho. Para tanto, neste é apresentada uma visão geral do funcionamento do protótipo desenvolvido, as dificuldades e as soluções adotadas. Na sequência, será abordado o processo de criação do protótipo, bem como são apresentadas explicações das funcionalidades do protótipo.

4.1 VISÃO GERAL

Considerando as informações fornecidas na seção do referencial teórico, materiais e a metodologia apresentada nas seções anteriores, foi elaborada uma visão geral do protótipo em comunicação com a Web PKI que fornece mecanismo para obtenção do certificado instalado na máquina do usuário pelo *browser*, visto que isso não é nativo dos navegadores. A visão geral do protótipo é apresentada na Figura 12.

Figura 12 –Visão Geral de comunicação do Protótipo com a API da Lacuna Web PKI



Na solução ilustrada pela Figura 12 é apresentada a visão geral da comunicação do protótipo com a Web PKI e REST PKI. Quando o usuário do SigDoc solicitar o carregamento dos certificados, é realizada uma consulta por meio da Web PKI que envia uma solicitação ao plugin instalado no navegador. A Web PKI recupera os certificados disponíveis como A1 e A3, obtêm os atributos públicos do certificado, faz a leitura da codificação binária de um certificado e é capaz de assinar dados com um certificado (Lacuna, 2018). O plugin instalado

no navegador tem a funcionalidade de se comunicar com o programa que está instalado na máquina do cliente no sistema operacional para obtenção dos certificados digitais.

Como o uso da Web PKI é de uma empresa privada, é necessária a instalação do plugin e do sistema no sistema operacional, pois não há outro mecanismo para obter os certificados digitais instalados na máquina, seja ele A1 ou A3, pois os navegadores ainda não possuem suporte para acessar dados ou fontes instaladas no sistema operacional. Sendo assim, o plugin facilita essa intermediação do sistema operacional com a aplicação, fornecendo os dados do certificado digital para uso no protótipo.

As próximas seções apresentarão os resultados obtidos no desenvolvimento deste trabalho.

4.2 ATIVIDADES

As seções a seguir apresentam os artefatos gerados no desenvolvimento do SigDoc, bem como sua estrutura e o software.

4.2.1 Artefatos

Nesta seção serão apresentados os artefatos gerados para a implementação do protótipo desenvolvido neste trabalho.

A criação dos artefatos seguiu o modelo UML (para diagramas) utilizando-o como base para a estimativa e para o desenvolvimento de um sistema. Especificamente, os artefatos desenvolvidos foram: levantamento de requisitos, diagrama de caso de uso, modelo conceitual, diagrama de classes e mockup do projeto. As seções seguintes apresentam os artefatos.

4.2.1.1 Levantamento de Requisitos

Nesse processo de análise de requisitos, todas as especificações foram repassadas pela Fábrica de Software do CEULP/ULBRA, identificando os aspectos imprescindíveis para o desenvolvimento da solução. Neste caso, foram utilizadas os *Users Stories* (Histórias de Usuário) como uma maneira no qual fique mais claro o que o software precisa fazer, bem como, melhor compreender as suas necessidades, dificuldades e anseios. A lista abaixo, representa os requisitos em formato de *users stories.*

Tabela 2 – Lista de Requisitos

Módulo Usuário

RF001	Como usuário eu quero cadastrar uma conta no sistema
RF002	Como usuário quero entrar no sistema

Módulo Documento

RF001	Como usuário quero cadastrar um documento
RF002	Como usuário quero consultar a lista de documentos
RF003	Como usuário quero fazer o download dos documentos

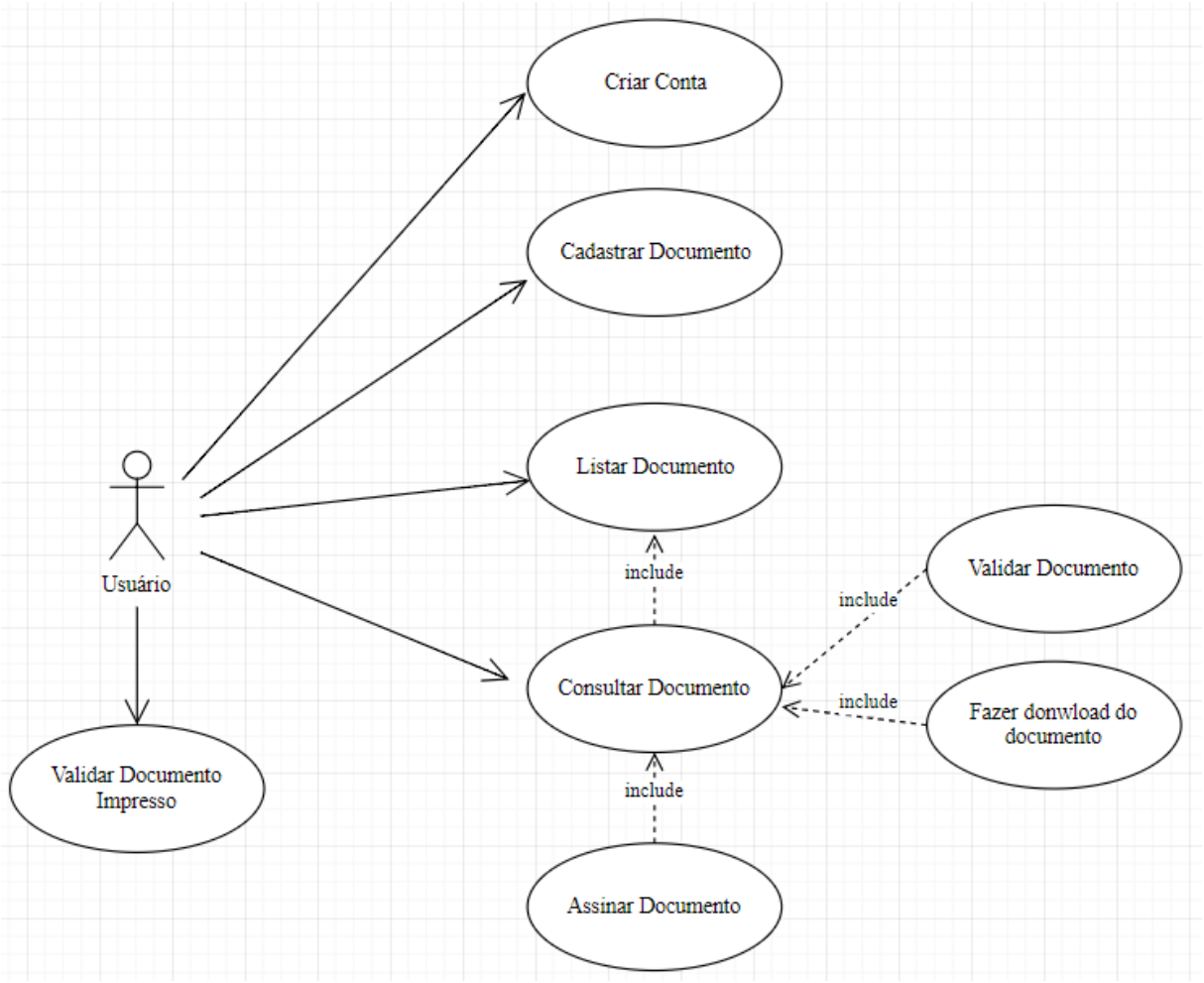
Módulo Assinatura

RF001	Como usuário quero assinar o documento selecionando certificado digital
RF002	Como usuário quero fazer o download do documento assinado
RF003	Como usuário quero validar o documento assinado

4.2.1.2 Diagrama de Caso de Uso

O diagrama de caso de uso descreve as funcionalidades que compõe o SigDoc e as interações dessas funcionalidades com os usuários. O diagrama é composto pelos casos de uso a seguir: Criar Conta, Cadastrar Documento, Listar Documento, Consultar Documento, Assinar Documento, Validar Documento Impresso, Validar Documento e Fazer download do Documento. As especificações de cada caso de uso, bem como suas interações são apresentadas na Figura 13.

Figura 13 – Diagrama de Caso de Uso



Caso de uso: Criar Conta

Tipo: Primário

Atores: Usuário

Descrição: A aplicação deve permitir que o usuário possa efetuar seu cadastro na aplicação. Esse cadastro de usuário deve solicitar ao usuário informe os seguintes campos: Nome, E-mail e Senha.

Fluxo: O Usuário acessa a área pública de cadastro, insere as informações exigidas para o cadastro e confirma a ação.

Caso de uso: Cadastrar Documento

Tipo: Primário

Atores: Usuário

Descrição: A aplicação deve permitir que o usuário possa efetuar o cadastro de documento na aplicação. Esse cadastro de documento deve solicitar ao usuário que informe os seguintes campos: Título, Descrição, Data de Publicação e Arquivo para upload.

Fluxo: O usuário logado e presente na página inicial da aplicação, navegará até o menu de cadastro. Ao escolher a opção de cadastro é redirecionado a tela de cadastro de documento, onde irá preencher os campos necessários e confirmar a ação.

Caso de uso: Listar Documento
Tipo: Primário
Atores: Usuário
Descrição: A aplicação deve permitir que o usuário possa listar os documentos na aplicação. A lista deve ter os seguintes campos: Título, Descrição, Data de Publicação.
Fluxo: O usuário logado, ao entrar na página inicial da aplicação será exibido a lista de documentos

Caso de uso: Consultar Documento
Tipo: Primário
Atores: Usuário
Descrição: A partir da lista de documentos, o usuário pode consultar um documento específico de sua escolha. Na exibição da consulta, irá mostrar os dados do documento, tais como: Título, Descrição, Data de Publicação e o Arquivo. Além disso, terá a opção para assinar o documento.
Fluxo: O usuário na lista de documentos, clica no botão de assinar/consultar. Os dados do documento são exibidos ao usuário e é apresentada a opção para assinatura.

Caso de uso: Assinar Documento
Tipo: Primário
Atores: Usuário
Descrição: A partir da consulta do documento, o usuário terá a opção para selecionar o certificado desejado para assinatura do documento.
Fluxo: No resultado da tela de consulta, o usuário seleciona o certificado desejado e clica em assinar documento.

Caso de uso: Validar Documento
Tipo: Primário
Atores: Usuário

Descrição: Na tela de resultado da assinatura, o usuário poderá clicar na opção de abrir o certificado e validar a assinatura.
Fluxo: Após a tela de assinatura, no retorno da mensagem de sucesso, o usuário terá uma opção de ação para validar a assinatura, então o usuário clica no link e é redirecionado para a validação.

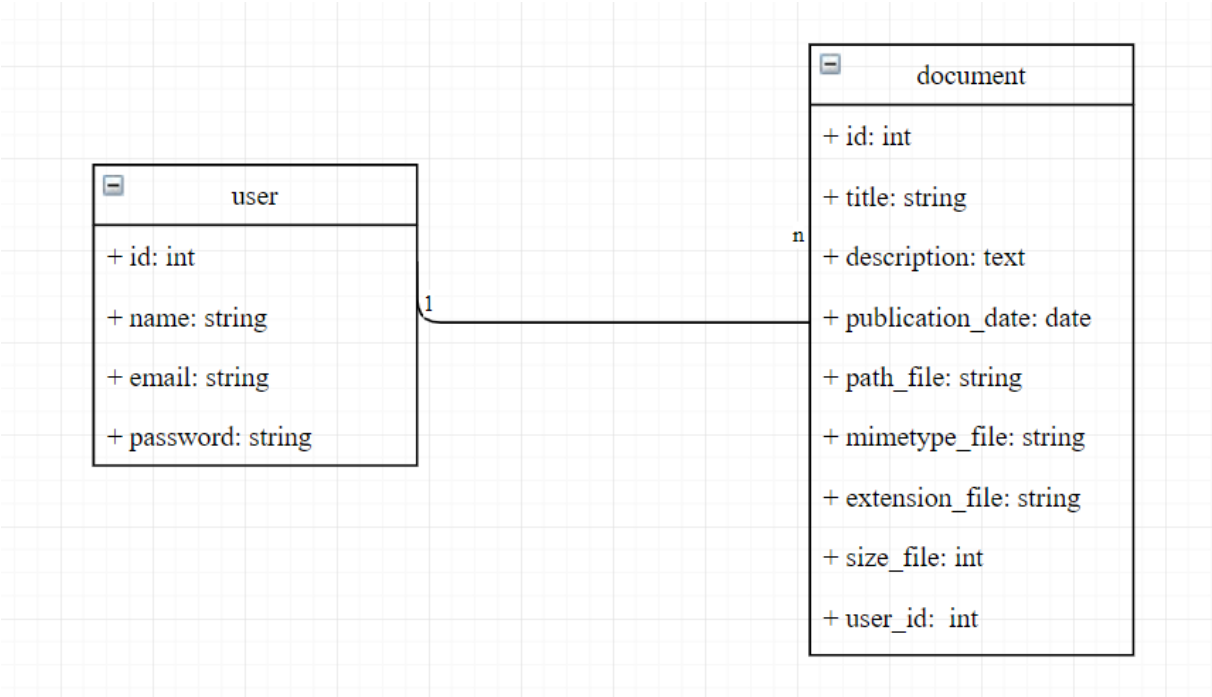
Caso de uso: Fazer Download Documento
Tipo: Primário
Atores: Usuário
Descrição: Na tela de resultado da assinatura, o usuário poderá clicar na opção de download do documento.
Fluxo: Após a tela de assinatura, no retorno da mensagem de sucesso, o usuário terá uma opção de ação para download do documento, então o usuário clica no link e é redirecionado para o documento assinado.

Caso de uso: Validar Documento Impresso
Tipo: Primário
Atores: Usuário
Descrição: O usuário ao receber o documento impresso, poderá acessar por meio de um link e inserir o código de verificação do documento.
Fluxo: O usuário recebe o documento impresso, digita no navegador a URL informada no documento, e logo após é apresentada ao usuário a validação do documento.

4.2.1.3 Modelo de Dados

O sistema proposto possui um modelo de dados composto por 2 entidades, sendo elas: `User`, que são os usuários do sistema; e `Document`, que contém as informações do documento inseridos no sistema. A figura 14 apresenta o modelo de dados do sistema.

Figura 14 – Modelo de Dados do SigDoc



Conforme ilustra a Figura 14, a entidade **`User`** apresenta os atributos: *id*, um identificador único do usuário, que possui o formato *int* (inteiro) que aceita apenas números inteiros; e os demais atributos como *name*, *e-mail* e *password* são do tipo *string*, que aceita uma cadeia de caracteres, sendo ela textos, números.

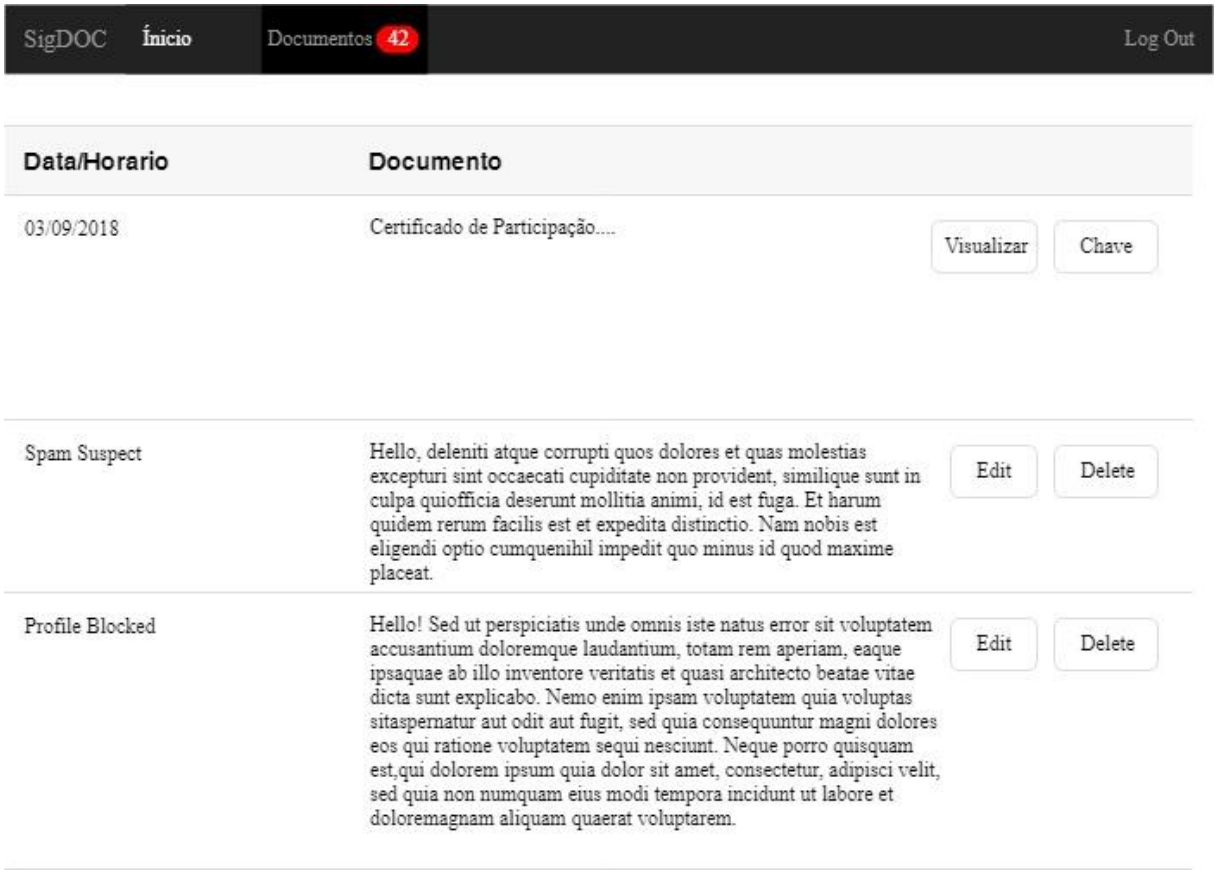
A entidade **`Document`** possui o atributo de identificação *id*, que recebe apenas número inteiros. Além do *id*, tem os atributos *size_file* e *user_id*, que são do tipo inteiro também, os atributos de tipo string e text, que aceita uma cadeia de caracteres, números, textos, os atributos são *title*, *description*, *path_file*, *mimetype_file*, *extension_file*, *o* atributo *publication_date* é do tipo date, aceita apenas valores no formato de data.

O relacionamento da entidade **`User`** para **`Document`** é de um para muitos, ou seja, um usuário pode ter zero ou mais documentos associados a ele.

4.2.1.4 Mockup do Projeto

A elaboração do *mockup* do projeto foi para auxiliar na compreensão do funcionamento do SigDoc, bem como montar um modelo para seguir no desenvolvimento do projeto. A tela inicial possui o elemento de listagem de documentos e ações conforme apresentado na Figura 15.

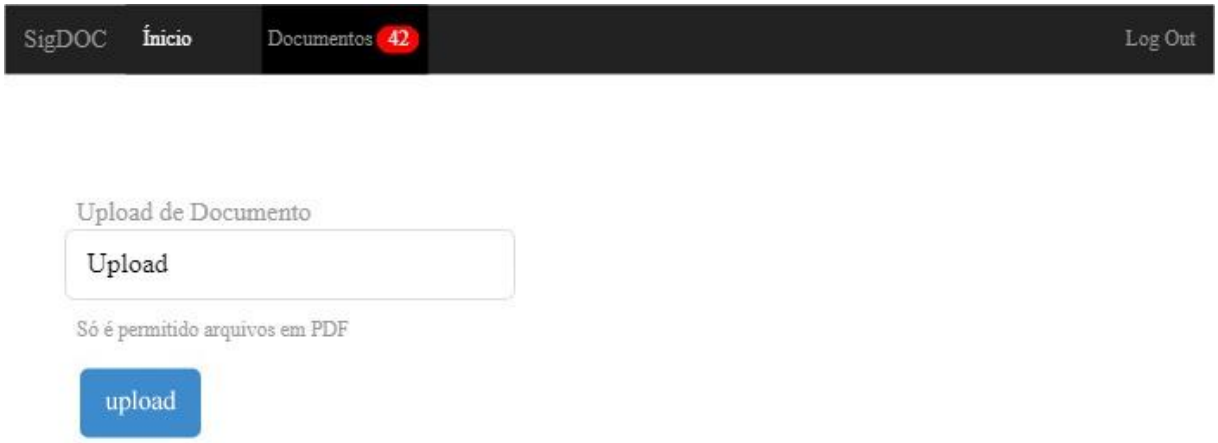
Figura 15 – *Mockup* da tela de listagem de documentos do protótipo



Na Figura 15 é apresentada uma tela com menu superior e uma lista de documentos, direcionando o usuário para pontos chaves dentro da estrutura do protótipo. O menu superior possui 3 funcionalidades, a listagem dos documentos, a criação (*upload* do documento) em documetnos e o *logout*, considerando que o usuário já esteja logado.

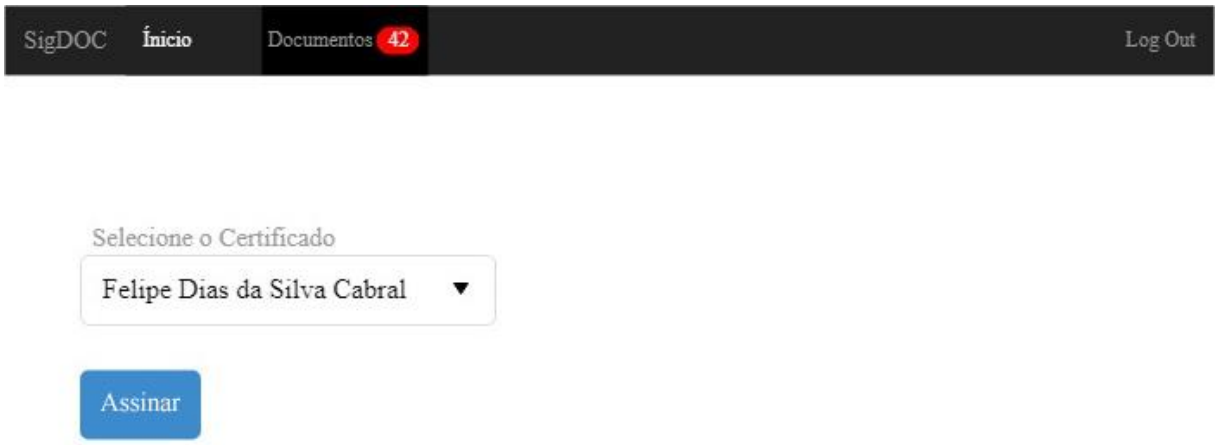
A Figura 16 ilustra uma tela que foi elaborada para o usuário efetuar o *upload* do documento para ser possível registrar a entrada do documento e salvá-lo em uma base de dados.

Figura 16 – *Mockup* da tela de cadastro de documentos



A seguir é apresentada a tela para selecionar o certificado, Figura 17, onde o usuário tem a opção de escolher o certificado para assinatura do documento submetido no protótipo.

Figura 17 – *Mockup* da tela de seleção de certificados



Cada uma das etapas apresentadas na Seção 4.2.1.5, que foram descritas em forma de protótipo de tela nos parágrafos anteriores, serão explicadas com mais detalhes na Seção 4.4.

4.3 ESTRUTURA DO SOFTWARE

Neste projeto é utilizado um framework MVC e isso faz com que o mesmo possua uma estrutura de pastas que contém os arquivos necessários para a sua execução. Nesta primeira

etapa são apresentadas as principais pastas do projeto e, na sequência, a seção é dívida em *front-end* e *back-end*, explicando as pastas e arquivos que envolvem as duas partes e o seu funcionamento.

As principais pastas e os principais arquivos são:

- **app**: contém os controllers, models e classes da aplicação.
- **database**: é um diretório do banco de dados, onde possui as *migrations*[1] e os modelos de dados.
- **filepdf**: armazena os arquivos PDF que são enviados pelos usuários e os arquivos PDF assinados
- **public**: contém imagens e arquivos estáticos utilizados na disponibilização do software
- **resources**: contém os *assets*, que são os arquivos em javascript e SASS (CSS), o diretório **lang,** que possui os arquivos de idiomas, e o diretório **views,** que possui os arquivos html.
- **routes:** é o diretório que contém todas as rotas da aplicação
- **vendor**: contém as dependências e pacotes usados do *back-end*
- **composer.json**: contém a lista das dependências
- **.env**: variáveis de configuração da aplicação

4.3.1 Front-end

Nesta seção são apresentados os recursos desenvolvidos como *front-end*, que se encontram no diretório "*resources*" do projeto. Esta pasta possui os *assets*, arquivos estáticos de *JavaScript* e CSS, e os arquivos de *views*, arquivos em html em formato *blade*, um conceito de templates do framework *Lavarel*.

Dentre os arquivos *JavaScript* há os que representam importações de bibliotecas (*Bootstrap, Datepicker, Jquery.blockUI, FileInput, SignatureForm, Utils*) e o arquivo responsável pela comunicação com a API da Lacuna: "resources/assets/js/signature-form.js". Este arquivo é distribuído pela Lacuna e livre para uso, podendo criar novas funções ou métodos para uso na aplicação. O código desse arquivo cria uma instância de "LacunaWebPKI", uma classe fornecida para obter dados do certificado instalado na máquina

[1] As migrações são como o controle de versão do banco de dados, permitindo que sua equipe modifique e compartilhe facilmente o esquema do banco de dados do aplicativo. Disponível em: < https://laravel.com/docs/5.7/migrations> Acesso em: 03, dez, 2018

do usuário, e isso ocorre através da comunicação da API com a extensão instalada no navegador do usuário.

No arquivo "signature-form.js" há a função *loadCertificades* que efetua uma chamada na WebPKI para listar todos os certificados instaladas na máquina do usuário. A função *sign()* é uma chamada para assinatura digital, que tem como objetivo fazer uma chamada a WebPKI e repassa o *token* gerado do documento e o a chave privada do certificado para a RestPKI, para iniciar o processo de assinatura do documento. Essa etapa de assinatura é explicada com mais detalhes na seção 4.4.5.

Ainda na pasta de "*resources/assets"* há arquivos como:

- *Datepicker.js*, uma biblioteca *JavaScript* para uso do componente de calendário na aplicação;
- *FileInput*.js, um arquivo responsável para upload dos documentos e fornece a opção de arrastar e soltar; e
- *Jquery.blockUI*, que permite, ao usar o AJAX, que o comportamento da função seja síncrono, sem o bloqueio do navegador.

As *views* estão na pasta *"resources//views"* em formatos blades templates um formato que possibilita o reuso de código e simplifica a inserção de código PHP em páginas HTML. A pasta *"resources/views/auth"* contém as *views* de *login e register*, responsável para renderização das telas de entrar no sistema e de criar conta. Já na pasta *"resources//views/document"* há arquivos das *views* referente ao documento:

- *check.blade.php* apresenta se a assinatura é válida ou não referente ao documento;
- *create.blade.php* apresenta os dados do documento e a opção para assinar o documento;
- *index.blade.php* é apresentado a lista dos documentos cadastrados no sistema;
- *list.blade.php* é o arquivo que apresentada o sucesso da assinatura do documento e bem como as opções de download do arquivo, e opções para validação da assinatura;
- *signatureExisting.blade.php* é responsável por apresentar a validação de um determinado documento a partir do seu código de validação; e
- *upload.blade.php*, é responsável por renderizar os campos de cadastro do documento e de upload do mesmo.

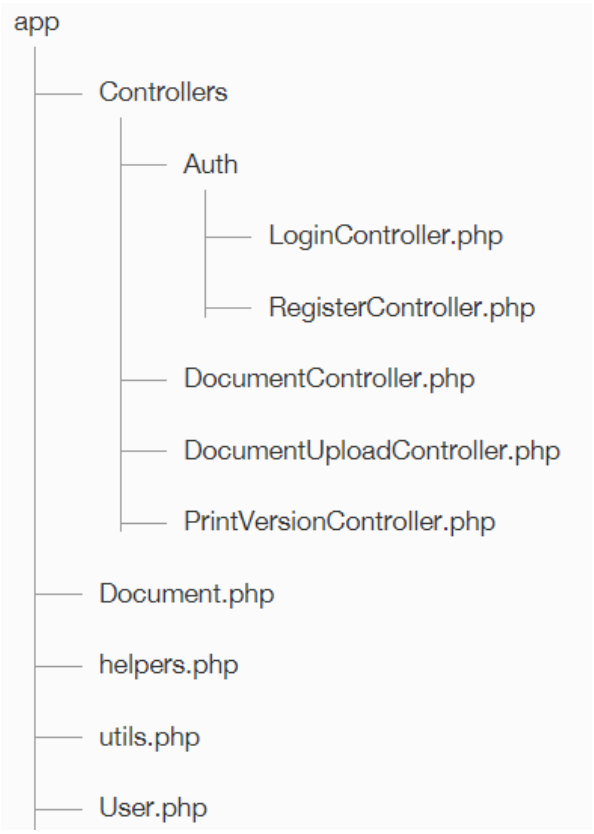
Por fim, a pasta *"resources/views/layouts"* tem o arquivo *app*, que é o principal da aplicação, pois contém o carregamento dos scripts JavaScript, bem como as bibliotecas CSS.

É valido ressaltar que todos os arquivos da pasta "*resources /assets/"* em *JavaScript* e *CSS* são compilados utilizando o WebPack, que empacota todos os assets e gera dois arquivos: *app.js* e *css.css*, que são incluídos no arquivo principal app.blade.php (presente na pasta *"resources/views/layouts"*).

4.3.2 Back-end

A Figura 18 ilustra a estrutura de pastas e arquivos principais que compõem o back-end.

Figura 18 – Estrutura de Pastas do projeto SigDoc

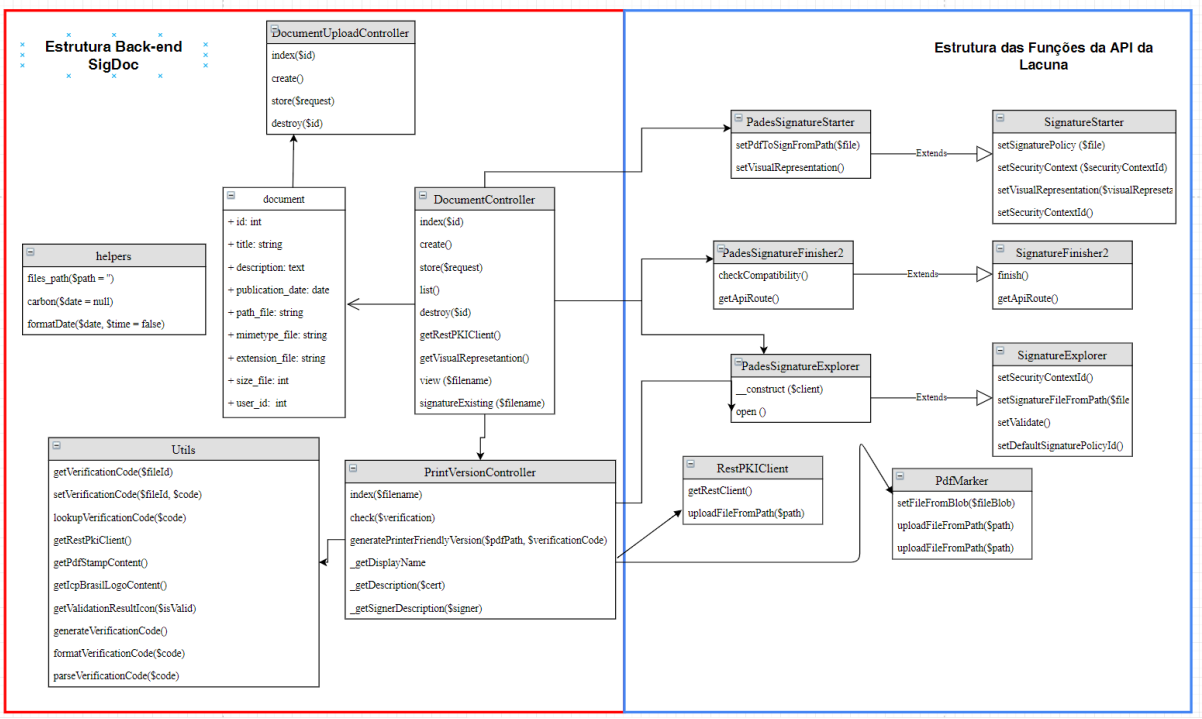


Como pode-se observar na Figura 18, na pasta *"app/Controllers/Auth"* estão os *controllers* responsáveis pelo Login e Registro do usuário. Ao utilizar *laravel* para criar um projeto de software web, o login e registro já são disponibilizados, ou seja, não é necessário alterá-los para que funcionem.

Os arquivos *DocumentController*, *DocumentUploadController* e *PrintVersionController* representam *controllers* responsáveis pela parte mais específica do software, além das funcionalidades padrão fornecidas pelo Laravel, e estão na pasta *"app/Controllers/"*. A Figura 19 apresenta uma visão geral da estrutura do *back-end*, bem

como a comunicação com as funcionalidades da API, que é descrita de forma mais detalhada a seguir.

Figura 19 – Visão geral do Back-end com as funções da API



O *DocumentController* possui funções responsáveis para o controle da aplicação: *index()* e *store()*. A função *index($id)* inicia uma assinatura PAdES usando a REST PKI e renderiza a página de assinatura. Sendo assim, a função index() opera da seguinte forma:

- recebe como parâmetro o *id* de um documento que já foi enviado para o software;
- cria uma instância da classe *PadesSignatureStarter*, responsável por receber os elementos da assinatura e iniciar a assinatura. Com essa instância, é definida a política de assinatura e as medidas utilizadas para editar as marcas do PDF;
- chama a função *PadesSignatureStarter.getVisualRepresetation($token)*, responsável por gerar a marca da assinatura inserida no arquivo PDF. Essa função recebe o token da Rest PKI, para conexão e registro na API da criação de um novo documento; e
- utiliza a função *PadesSignatureStarter.startWithWebPki()* para obter o token correspondente à assinatura e a envia para a view *create.blade.php* responsável para obter o certificado.

A função *store($request)* recebe como parâmetro o token gerado na função *index* e cria uma instância da classe *PadesSignatureFinisher2,* responsável por completar o processo de assinatura. Sendo assim, a função store() opera da seguinte forma:

- recebe como parâmetro o token do documento enviado para o software;
- cria uma instancia da classe *PadesSignatureFinisher2 passando o token de acesso a RestPKI da Lacuna;*
- *PadesSignatureFinisher2.token* recebe o token do documento e logo após chama a função *PadesSignatureFinisher2.finish()* para finalizar o processo de assinatura que é realizado pela RestPKI
- O resultado é inserido na variável *signatureResult* que possui os dados do certificado que foi usado na assinatura e pode ser utilizado para apresentar dados do certificado e da assinatura na view para o usuário; e
- em seguida foi utilizada a função *writeToFile* para salvar o documento assinado localmente (em disco).

A função *signatureExisting($filename)* recebe como parâmetro o nome do arquivo para poder obter o local de armazenamento do mesmo e cria uma instância da classe *PadesSignatureExplorer,* que é responsável para validar PDF já assinados. Depois o código utiliza a função *PadesSignatureExplorer.setSignatureFileFromPath(),* responsável por disponibilizar o caminho do arquivo a ser inspecionado, visto que vai verificar a assinatura existente.

Ainda sobre a instancia da classe *PadesSignatureExplorer,* o código define como *default* o padrão de assinatura PAdES no método *defaultSignaturePolicy* e no método *securityContext* é especificado o contexto de segurança utilizado. Por fim, é chamado o método *open(),* que retorna as informações do arquivo de assinatura.

O *DocumentUploadController* possui a função *index(),* que obtém os documentos através do *model* e redireciona para a view de documentos, para a listagem de todos os documentos existentes no banco. A função *store()* é responsável por receber a requisição que contém o arquivo, tratar os dados e adicionar o documento na pasta *"app/filespdf/upload",* depois isso redireciona para view de listagem dos documentos.

Por fim, o *PrintVersionController* possui funções para criação do documento assinado para versão de impressão. Sendo assim, o processo acontece da seguinte forma:

- inicialmente, é definido o nome do site ou frase que irá apresentar no documento, isso é inserido na variável *verificationSiteNameWithArticle*;

- a seguir define-se a URL do site que o usuário será redirecionado ao digitar a URL no browser, esse valor é inserido na variável *verificationSite;*
- em normalFontSize é definido o tamanho da fonte do documento a ser gerado;
- em *dateFormat* é definido o formato da data de apresentação; e
- na variável *timeZoneDisplayName* é definido o nome do fuso horário.

Essas variáveis são globais e são utilizadas na composição da página da assinatura, no qual a função *generatePrinterFriendlyVersion()* é responsável pela criação do documento, bem como, a adição da assinatura para o formato de impressão.

A função *index($filename)* recebe como parâmetro o nome do arquivo, para localização do mesmo nos diretórios do projeto. Portanto, a função *index()* opera da seguinte forma:

- é verificado se já foi gerado o código de verificação e, caso não tenha sido gerado, a função *generateVerificationCode* gera um código de verificação para o documento;
- em seguida, é chamada a função *generatePrinterFriendlyVersion($filePath, $verificationCode)* que tem como parâmetros o local de armazenamento do documento e o código de verificação gerado; e
- ao final da execução da função *generatePrinterFriendlyVersion(),* é redirecionado o documento na versão de impressão e renderizado para o usuário.

A função *generatePrinterFriendlyVersion()* é responsável por criar um novo documento, inspecionar o documento recebido, verificar se a assinatura dele é válida, adicionar a política de segurança do PAdES e depois começam as funções de edição do documento, visto que é adicionado um resumo da assinatura na margem inferior e direita de cada página. Portanto, o restante da função é apenas para configuração do arquivo que será gerado para impressão.

Em seguida, há uma função *check($verification)* a qual tem a função de verificar se o documento é válido a partir do código gerado referente ao documento que foi impresso. A função *check($verification)* funciona da seguinte forma:

- a função recebe por parâmetro o código de verificação, e com isso é localizado o documento armazenado com o código de verificação.
- após isso com a instância de *PadesSignatureExplorer* é verificado a validade da assinatura utilizando o Rest PKI.

- com a confirmação da assinatura é invocada a função *PadesSignatureExplorer.open()* que retorna os dados da assinatura para o usuário na view de verificação.

As *Moldes* utilizadas no sistema são *Document* e *User* que são geradas de acordo com o modelo de dados do sistema. Os métodos de cada model são gerados automaticamente pelo framework. Caso tenha alguma alteração ou personalização de alguma função referente ao model, é adicionado nas suas respectivas classes.

No diretório *app* possui o arquivo *helpers.php*, um arquivo que contém funções globais para utilização no sistema. Neste arquivo possui uma função *files_path* para localização do arquivo no diretório especificado no projeto, tem a função *carbon* responsável por tratar a data e a função *formateDate* responsável por retornar a data em um formato especifico.

Por fim, o último arquivo do diretório é o arquivo *Utils* que possui funções que a *PrintVersionController* utiliza para montagem do arquivo. A função *getVerificationCode($fileId)* recebe como parâmetro o id do documento e retorna o código de verificação associado ao documento fornecido.

A função *setVerificationCode()* é responsável por registrar o código de verificação para um determinado documento, para isto, recebe como parâmetro a identificação do documento e código. Nesta *controller*, há uma função para geração do código sendo possível gerar ele em 12, 16, 20, 24 caracteres e a função responsável é a *generateVerificationCode()*. Para o código ficar mais legível para o usuário e fácil digitação, é feito uma formatação conforme o tamanho de caracteres, a função responsável para isto é a *formatVerificationCode()*.

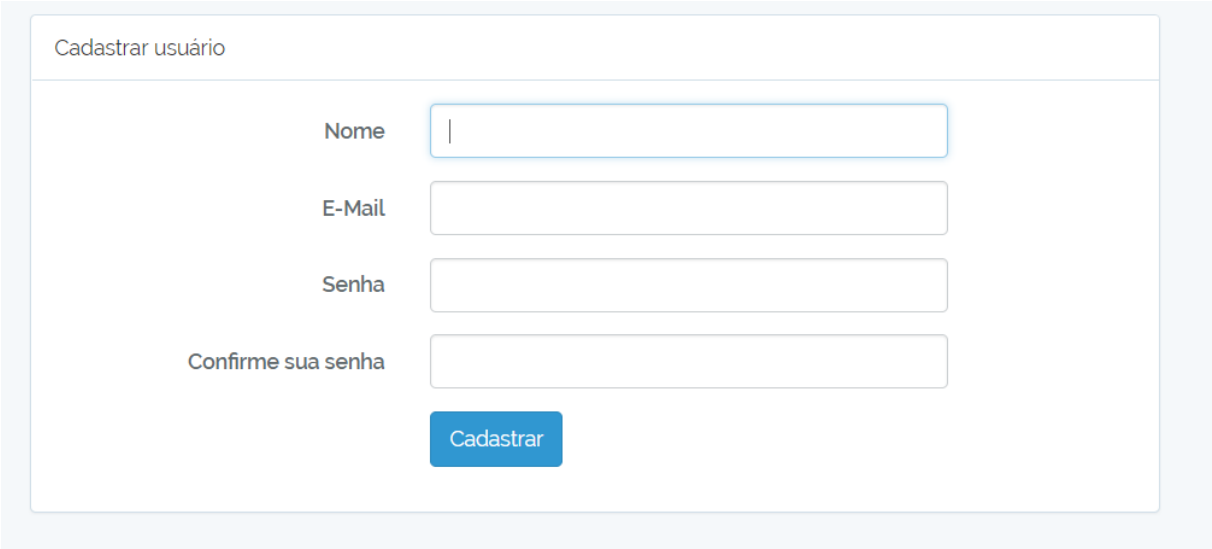
4.4 SOFTWARE

Nesta seção são apresentadas as funcionalidades do *SigDoc*, demonstrando como é o seu funcionamento.

4.4.1 Criar Conta

Para acesso ao sistema, o usuário deve efetuar o seu cadastro informando o nome, e-mail e senha, como ilustra a Figura 20.

Figura 20 – Tela de Criar Conta



A Figura 20 demonstra que a tela de cadastrar usuário apresenta um formulário contendo os campos:

- Nome (texto)
- E-mail (endereço de e-mail)
- Senha e Confirmação de senha

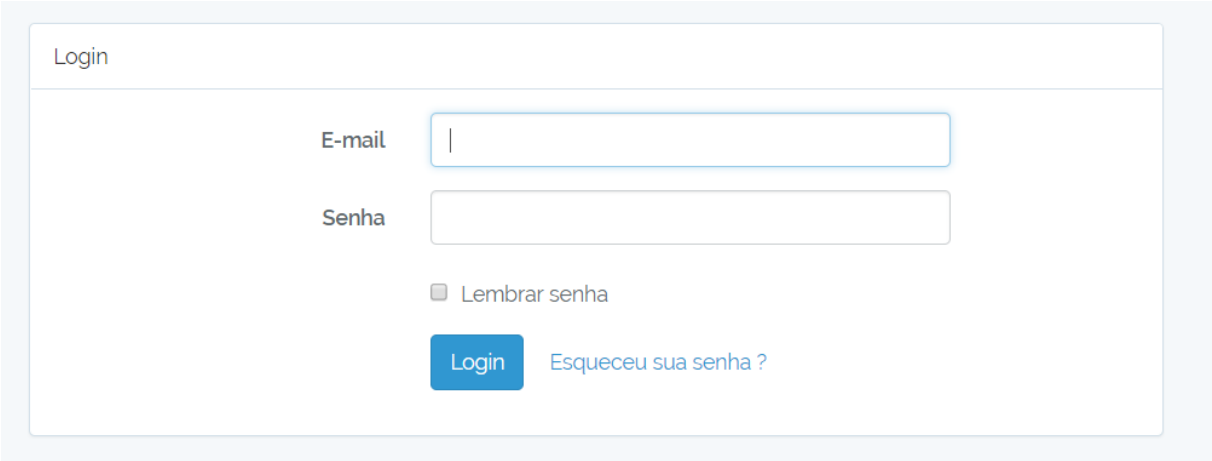
O botão "Cadastrar" faz com que os dados sejam validados e, se estiverem corretos, o cadastro do usuário é efetivado e o software apresenta uma mensagem de confirmação; caso contrário, é apresentada uma mensagem de erro informando o que ocorreu.

A tela cadastrar usuário representa a *view* do *controller RegisterController*, responsável por receber os dados do formulário de login da *view*, bem como tratar as exceções e retornar uma mensagem ao usuário.

4.4.2 Entrar no Sistema

Para acessar as funcionalidades do software, o usuário precisa utilizar a funcionalidade de entrar no sistema. A figura 21, apresenta os campos de e-mail e senha para o usuário entrar no sistema, após ter efetuado seu cadastro.

Figura 21 – Tela de Entrar no Sistema

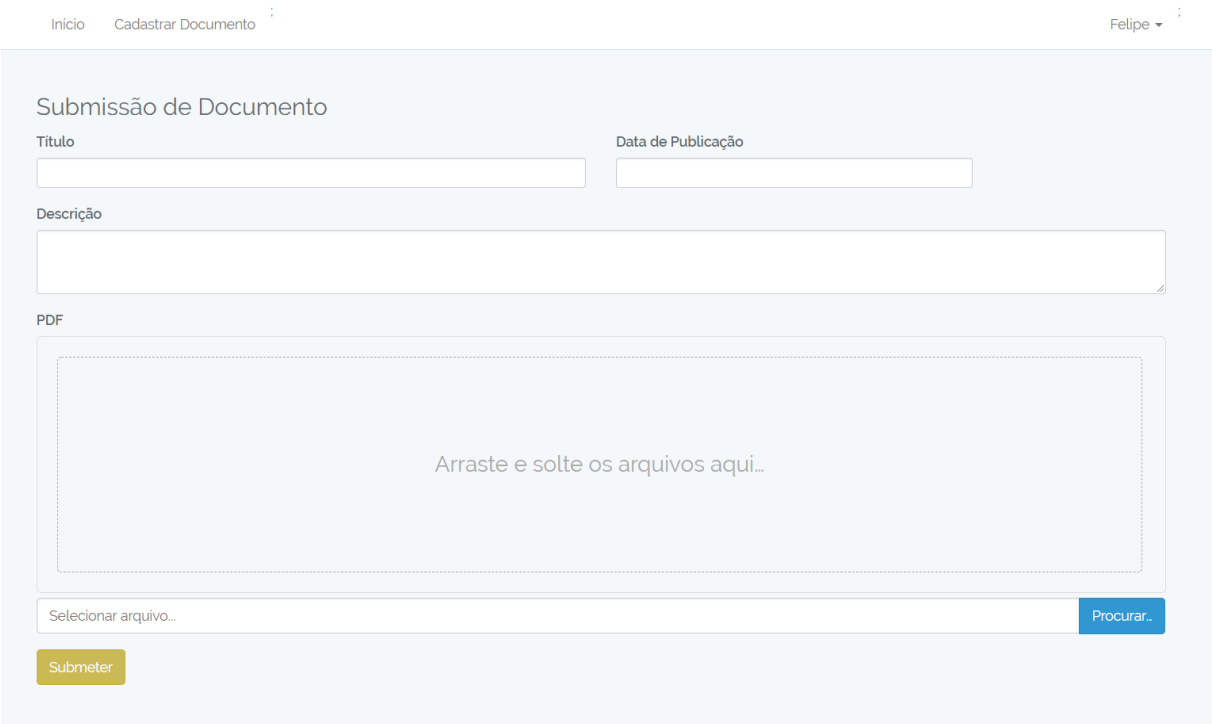


A sessão de login é similar com a sessão de iniciar o Windows. O usuário deve preencher o campo de e-mail e senha para poder entrar no sistema. Após o usuário entrar no sistema, é apresentada a tela de início, uma tela com a listagem dos documentos e uma opção para cadastro de documentos. Esta tela está vinculada a *controller LoginController*, responsável por receber os dados, tratar as exceções e retornar uma mensagem ao usuário.

4.4.3 Cadastrar Documentos

Para efetuar o cadastro de documento no *SigDoc*, o usuário deve utilizar a funcionalidade de cadastrar documentos. A Figura 22 abaixo apresenta a funcionalidade de cadastro do documento.

Figura 22 – Tela de cadastro de documento



Conforme demostrado na figura 22 a tela de cadastro de documento permite ao usuário realizar o *upload* de um arquivo, bem como, informar o título, a data de publicação e a descrição do documento. Ao fazer o upload do documento é definida uma política de assinatura, a qual utiliza PAdES, um conjunto de características, extensões do PDF e restrições (HIRATA, 2018). O PDF é normatizado pela ISSO 32.000-1 (2008), o que torna adequado o uso do PAdES para assinatura digital.

Após a submissão do documento, o usuário vai para a tela da lista dos documentos submetidos conforme apresentado na Figura 22. A tela de cadastro de documentos está vinculada a *controller DocumentUploadController*, que contém a função *store()* responsável pelo upload do documento.

4.4.4 Listar Documentos

Para o usuário ver os documentos listados no *SigDoc*, é necessário acessar a funcionalidade listar documentos na opção início no menu. A Figura 23, apresenta os documentos submetidos pelo usuário e tem as opções de assinar o documento específico ou excluir o documento.

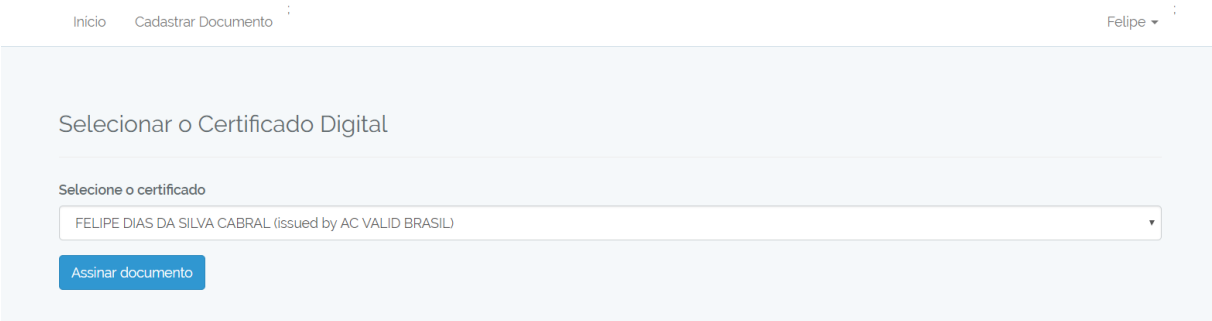
Figura 23 – Tela de lista de documentos

Se o usuário optar por assinar o documento, ele é redirecionado para a tela de seleção do certificado, conforme a figura 24 apresentada na próxima seção. Caso escolha a opção de excluir, o documento é deletado dos registros do usuário. Todavia, não permanente, apenas para visão do usuário. A tela de listagem dos documentos está vinculada a *controller DocumentUploadController*, no qual a função *index()* consulta todos os documentos existentes e retorna para a view *index.blade.php.*

4.4.5 Assinar Documento

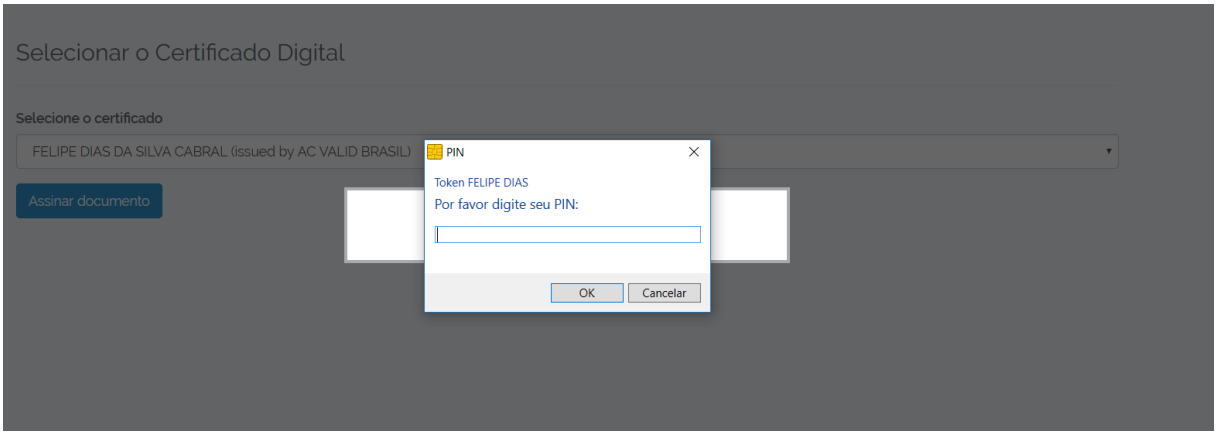
Nesta etapa, são apresentados os dados do documento que o usuário selecionou, assim como, a opção para selecionar o certificado digital para iniciar o processo de assinatura do documento.

Figura 24 – Tela de assinar o documentos



Conforme apresentado na Figura 24, o usuário pode selecionar o certificado da sua preferência e após isso será feito o processo de assinatura do documento. Após o usuário clicar em assinar documento, é solicitada a senha do token do certificado conforme apresentado na Figura 25 abaixo:

Figura 25 – Tela de solicitação de senha do token



Neste processo, a aplicação se comunica com a WEB PKI, onde o principal objetivo é realizar o processo de *client-side* necessário em operações com certificados digitais.

Em linhas gerais o processo ocorre da seguinte forma, tendo em vista uma visão geral do processo:

1) O *backend* inicia a assinatura, gerando os *bytes* que precisam ser utilizados como entrada do algoritmo de assinatura utilizando a chave privada do usuário (chamados de "*to-sign-data*")
2) O "*to-sign-data*" é transmitido para o lado do cliente
3) O algoritmo de assinatura é realizado no *frontend* com a chave privada do certificado do usuário

4) O resultado do algoritmo de assinatura é transmitido de volta ao servidor
5) O *backend* finaliza o processo de assinatura

A assinatura de um documento pode ser categorizada em diferentes níveis de garantia, neste protótipo se atende a assinatura digital, uma implementação de assinatura eletrônica que por sua vez utiliza certificados de chave pública X.509 e aderindo ao padrão de assinatura PAdES.

Após esse processo, é apresentada a tela de sucesso da assinatura e opções para o usuário efetuar o download do documento em formato digital, download para impressão e validação da assinatura, conforme é apresentado na Figura 25. A tela de assinar documento está vinculada a *controller DocumentController* e usa as funções da *index()* e *store()* para fazer os procedimentos da assinatura.

4.4.6 Download do Documento e Validação

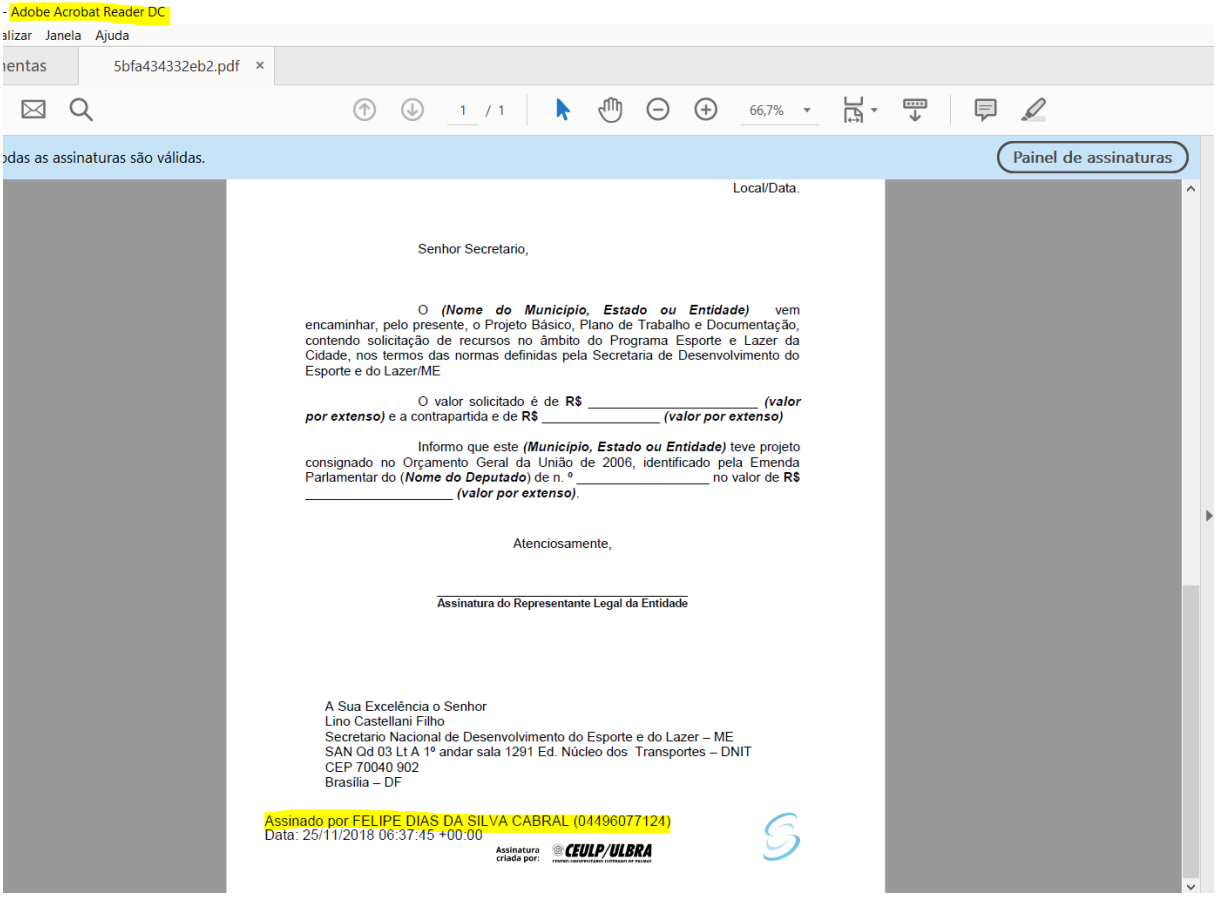
A Figura 26 apresenta a tela de sucesso e as opções para download e validação do documento.

Figura 26 – Tela de sucesso da assinatura do documento

Conforme a Figura 26, a primeira opção é o download do documento assinado. Sendo assim, o usuário pode abrir o documento no software *Abode Acrobat Reader*, para validar as propriedades da assinatura digital. A validação é determinada verificando a autenticidade do status do certificado de ID digital da assinatura e a integridade do documento. Na validade da assinatura é confirmado que o certificado assinante existente na lista do validador da entidade. A validação da integridade, verifica se o documento foi alterado após a assinatura do documento. A integridade garante que o conteúdo não foi alterado pelo signatário.

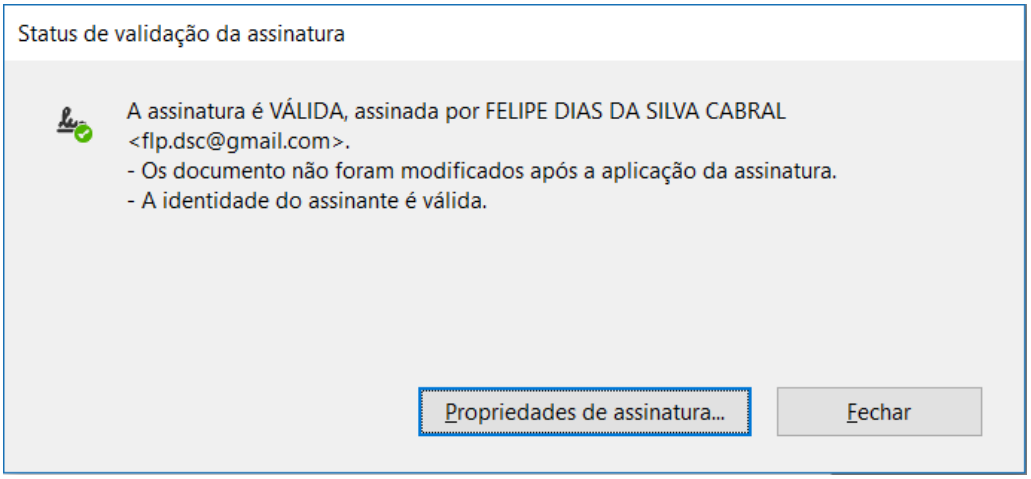
Na Figura 27 é apresentado o documento assinado e executado no software Abode Acrobat Reader.

Figura 27 – Tela de documento assinado executado no Acrobat Reader



Conforme a Figura 27, ao clicar na assinatura em destaque amarelo, é apresentado uma janela do status da validação da assinatura conforme apresentada na Figura 28:

Figura 28 – Tela de status da validação da assinatura



Observa-se na Figura 28, a verificação da autenticidade e de integridade foi validada. Em seguida, o usuário pode visualizar todas as propriedades da assinatura, como em destaque

na Figura 28. Essa tela de sucesso da assinatura e download do documento está associada a *controller DocumentController* que possui as funções *view()* e *store()* responsáveis por disponibilizar o arquivo para download, visto que é salvo em disco e o resultado da assinatura é obtido no retorno da função *store().*

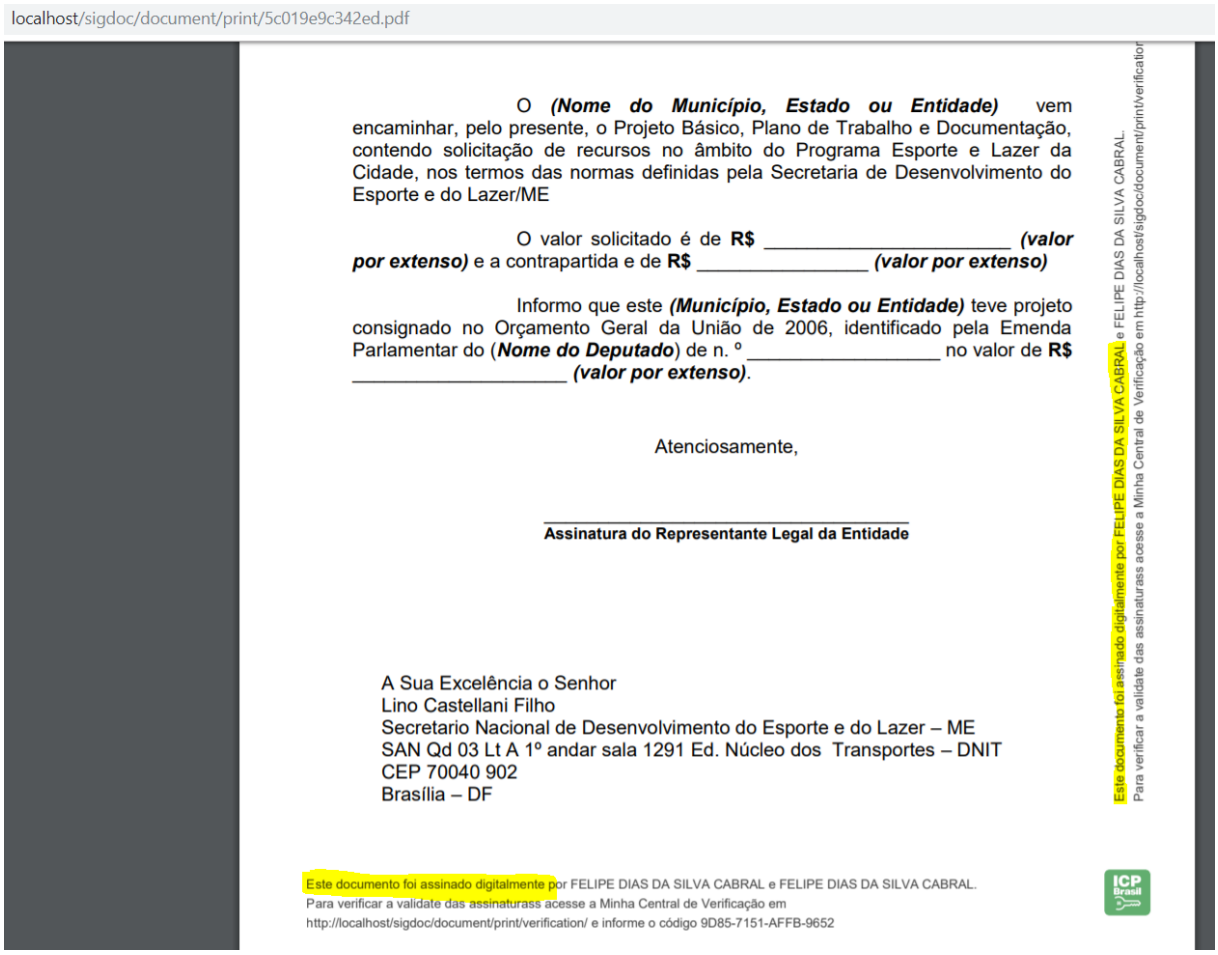
4.4.7 Validar Documento Impressora

O usuário possui a possibilidade de efetuar o download do documento para impressão, neste caso, é gerado um formato em que a assinatura pode ser validada digitando código de verificação do documento.

Figura 29 – Tela da assinatura na versão do documento impresso

No documento gerado para impressão, conforme a Figura 29 acima, na última página, contém as informações da assinatura eletrônica. Além de ter informações nas margens direita e inferior, conforme apresentado na Figura 30:

Figura 30 – Tela de informações da assinatura na versão impressa



O usuário pode validar o documento, acessando a *URL* informada no documento. Posto isto, o usuário pode conferir a validade do documento na tela do sistema, contendo todas as informações conforme a Figura 31:

Figura 31 – Tela da validação da assinatura

Como pode ser observado na Figura 31, a validação do documento está correta, foi encontrada uma assinatura existente no documento e ela está caracterizada como válida. A Figura 31, apresenta os dados da assinatura e do certificado para o usuário que estiver validando o documento via *URL*, disposta no documento impresso.

5 CONSIDERAÇÕES FINAIS

Este trabalho consistiu no desenvolvimento do protótipo de um módulo de assinatura digital utilizando certificados digitais, com objetivo trazer melhorias de segurança para o atual processo de protocolação de documentos existente no CEULP/ULBRA. Desta forma, o protótipo garante a legitimidade de documentos através de assinaturas com o uso de certificados digitais.

O sistema desenvolvido utiliza a API fornecida pela empresa Lacuna para assinatura do documento utilizando certificado digital, isso garante a segurança do processo de protocolação, bem como, confirma que o documento possui validade jurídica, por meio de autenticidade, integridade, irrefutabilidade e tempestividade, visto que são assinados com certificados digitais.

A aplicação deste protótipo permite manter e garantir quem é o emissor e quem é o destinatário, data e horário de emissão, validade do documento e que o mesmo não sofreu não-repúdio. Além disso, aprimorar os processos de verificação do documento em qualquer lugar; redução de custos administrativos em decorrência do descarte de documentos em papel; torna-se desnecessário colher assinatura manual e a verificação da mesma.

O desenvolvimento da aplicação exigiu muito esforço, visto que há várias tecnologias envolvidas. Tecnologias de front-end, back-end, além de comunicação com API. Contudo, a conclusão da aplicação foi satisfatória, visto que atendeu todos os requisitos e a ferramenta pode ser aplicada em diversos contextos.

Para implantação da ferramenta no CEULP/ULBRA deve passar por alguns ajustes, como, a conexão com a base de dados dos alunos, para acessarem com o CGU, e adequar aos padrões da fábrica. A ferramenta não necessita de um treinamento complexo, apenas uma breve introdução a respeito do seu funcionamento e ela poderá ser utilizada.

O que se pode registrar de mais significativo nesse trabalho foi as tecnologias utilizadas e as técnicas para assinatura do documento, bem como, o que ela pode atender e a oferecer de melhorarias aos setores administrativos de algumas empresas. Inclusive, a análise feita nesse trabalho, visam servir como instrumento para mudança de processos que podem ser melhorados utilizando aplicações como esta.

Por fim, fica para trabalhos futuros o desenvolvimento de ferramentas para leitura de QRCode impresso no documento, com objetivo de facilitar a leitura de informações de assinatura e protocolo do documento. Ainda pensando em melhorias, o desenvolvimento de um plugin para integração com editores de textos, para assinatura dos documentos salvos, assim diminuiria o uso de uma ferramenta apenas para assinar.

6 REFERÊNCIAS

ARAÚJO, Bruno Gomes de et al. PROCESSO DE CERTIFICAÇÃO DE SISTEMAS DE REGISTRO ELETRÔNICO DE SAÚDE NO BRASIL: UMA ABORDAGEM ABRANGENTE E OS PRINCIPAIS DESAFIOS. **Revista Brasileira de Inovação Tecnológica em Saúde Issn**: 2236-1103, [s.l.], v. 3, n. 3, p.1-16, 19 dez. 2013. Revista AMAZON. **Certificados X.509.** 2018. Disponível em: <https://docs.aws.amazon.com/pt_br/iot/latest/developerguide/x509-certs.html>. Acesso em: 13 jun. 2018.

ANDERSON, Ross J. Security Engineering: **A Guide to Building Dependable Distributed Systems**. 2. ed. [s.l.]: John Wiley & Sons, 2008. 1040 p. Disponível em: <https://books.google.com.br/books?id=eo4Otm_TcW8C>. Acesso em: 01 dez. 2018.

ANJOS, Leandro Terra Versiani dos. **Segurança da Informação na Programação com uso de Criptografia e Certificação Digital.** 2016. 97 f. TCC (Graduação) - Curso de Tecnologia em Sistemas de Computação, Universidade Federal Fluminense, Niterói, 2016. Disponível em: <https://app.uff.br/riuff/bitstream/1/5696/1/TCC_LEANDRO_TERRA_VERSIANI_DOS_ANJOS.pdf>. Acesso em: 23 maio 2018.

Brasileira de Inovação Tecnológica em Saúde (R-BITS). http://dx.doi.org/10.18816/r-bits.v3i3.3626.

BRY. **PDDE:** O que é. 2018. Disponível em: <https://www.bry.com.br/bry-pdde/#o-que>. Acesso em: 03 jun. 2018.

BOUCHET-SAULNIER, Françoise. **The Practical Guide to Humanitarian Law.** [s.l.]: Rowman & Littlefield, 2007. 555 p. Disponível em: <https://books.google.com.br/books?id=ZvnerYYLKjsC&dq>. Acesso em: 01 dez. 2018.

CARMO, Dyego Souza do; SANTOS, KeitiTakeuti dos. **PROTOCOLADORA DIGITAL DE DOCUMENTOS ELETRÔNICOS.**2009. 66 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Positivo Núcleo de Ciências Exatas e Tecnológicas, Curitiba, 2009. Disponível em: <http://www.up.edu.br/blogs/engenharia-da-computacao/wp-content/uploads/sites/6/2015/06/2009.4.pdf>. Acesso em: 11 jun. 2018.

COSTA, Vanessa. **Um Estudo da Confiabilidade do Sistema de Protocolação Digital de Documentos Eletrônicos.** 2003. 161 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2003. Disponível em: <https://repositorio.ufsc.br/xmlui/handle/123456789/86203>. Acesso em: 02 abr. 2018.

CONARQ. **DOCUMENTOS ELETRÔNICOS - CTDE:** PERGUNTAS MAIS FREQUENTES. 2018. Disponível em: <http://www.conarq.arquivonacional.gov.br/documentos-eletronicos-ctde/perguntas-mais-frequentes.html>. Acesso em: 03 abr. 2018.

DEVMEDIA. **Introdução ao PostgreSQL:** O que é o PostgreSQL?. 2018. Disponível em: <https://www.devmedia.com.br/introducao-ao-postgresql/6390>. Acesso em: 23 maio 2018.

DICIO, DICIONÁRIO ONLINE DE PORTUGUÊS. **Irrefutabilidade.** 2018. Disponível em: <https://www.dicio.com.br/irrefutabilidade/>. Acesso em: 23maio. 2018.

DICIO, DICIONÁRIO ONLINE DE PORTUGUÊS. I**rretroatividade.** 2018. Disponível em: <https://www.dicio.com.br/irretroatividade />. Acesso em: 23maio. 2018.

ESMEC. **Documentos Eletrônicos.** 2009. Disponível em: <http://esmec.tjce.jus.br/wp-content/uploads/2009/05/3-documento-eletronico-esmec.pdf>. Acesso em: 03 jun. 2018.

FABRO, Bry Gustavo Lückmann. **Protocolação Digital de Documentos Eletrônicos ou "Certificação Digital":** O que é. 2018. Disponível em: <http://www.inf.ufsc.br/~aldo.vw/InfoMed/2005/seguranca.html>. Acesso em: 03 jun. 2018.

GALVÃO, Michele da Costa. **Fundamentos em segurança da informação.** São Paulo: Pearson Education do Brasil, 2015. Disponível em: <http://ulbra.bv3.digitalpages.com.br/users/publications/9788543009452/pages/-6>. Acesso em: 11 abr. 2018.

HIRATA, Wilson R. **PAdES – Novo Padrão Brasileiro de Assinatura Digital.** 2018. Disponível em: <http://certforum.iti.gov.br/2015/brasilia/wp-content/uploads/2014/10/apresentacao_hirata.pdf>. Acesso em: 13 nov. 2018.

IBM. **Formatos de certificado e chave.** 2018. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSRMWJ_6.0.0.3/com.ibm.itim_pim.doc/wadpwdsync/install_config/c_pwd_sslcfg_cert_key_formats.htm>. Acesso em: 13 jun. 2018.

ITI. **Certificado Digital.** http://iti.gov.br/certificado-digital. Disponível em: <http://iti.gov.br/certificado-digital>. Acesso em: 02 maio 2018.

ITI. **ICP-BRASIL.** Disponível em: <http://www.iti.gov.br/icp-brasil>. Acesso em: 02 maio 2018.

ITI. **ASSINATURAS DIGITAIS NA ICP BRASIL:** DOC ICP 15. 2008. Disponível em: <http://www.iti.gov.br/images/repositorio/consulta-publica/encerradas/DOC-ICP-15-Assinaturas_digitais_na_ICP-Brasil.pdf>. Acesso em: 20 maio 2018.

JOÃO, Belmiro N..**Sistemas Computacionais.** São Paulo: Pearson Education do Brasil, 2014. Disponível em: <http://ulbra.bv3.digitalpages.com.br/users/publications/9788543005621/pages/-8>. Acesso em: 11 abr. 2018.

LACUNA. **Documentação Lacuna Software.** 2018. Disponível em: <https://lacunasoftware.com/>. Acesso em: 26 set. 2018.

MACEDO, Geraldo Majela Ferreira de. **BASES PARA A IMPLANTAÇÃO DE UM SISTEMA DE GERENCIAMENTO ELETRÔNICO DE DOCUMENTOS – GED. ESTUDO DE CASO.** 2003. 62 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, FlorianÓpolis, 2003. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/85790/191647.pdf?sequence=1&isAllowed=y>. Acesso em: 13 jun. 2018.

MICROSOFT. **Certificados de Chave Pública X.509.** 2018. Disponível em: <https://msdn.microsoft.com/pt-br/library/windows/desktop/bb540819(v=vs.85).aspx>. Acesso em: 13 jun. 2018.

MORAES, Rafael Luis. **O uso da assinatura digital em contratos de câmbio: estudo de caso no sicredi.** 2009. 55 f. Monografia (Especialização) - Curso de Pós Graduação em Latu Sensu Gestão em Arquivos, Universidade Federal de Santa Maria, Santa Maria, 2009. Cap. 4. Disponível em: <http://repositorio.ufsm.br/bitstream/handle/1/2990/Moraes_Rafael_Luis.pdf?sequence=1>. Acesso em: 19 maio 2018.

MUZZI, Fernando Augusto Garcia; TAMAE, Rodrigo Yoshio. PADRÃO DE CRIPTOGRAFIA BASEADA NO PKCS. **Revista CientÍficaEletrÔnica de Sistemas de InformaÇÃo**, Garça, v. 1, n. 1, p.1-4, set. 2004. Disponível em: <http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/OgMCnWz3AsnO9BS_2013-5-24-11-16-41.pdf>. Acesso em: 13 jun. 2018.

NÚÑEZ, José Manuel Ramírez. **IBE Aplicado à Criptografia Temporal.** 2014. 111 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2014.

OLIVEIRA, Ronielton Rezende. **Criptografia simétrica e assimétrica: os principais algoritmos de cifragem.** 2002. Disponível em: <http://www.ronielton.eti.br/publicacoes/artigorevistasegurancadigital2012.pdf>. Acesso em: 08 jun. 2018.

OTWELL, Taylor. **Laravel.** 2018. Disponível em: <https://laravel.com/>. Acesso em: 23 maio 2018.

PIRES, Cleyton André; DIAS, Marcos Aurélio. **Auditoria de uma Protocolizadora Digital de Documentos Eletrônicos.** 2013. 87 f. Tese (Doutorado) - Curso de CiÊncias da ComputaÇÃo, Universidade Federal de Santa Catarina, Florianópolis, 2013. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_358/TCC_Cleyton_e_Marcos.pdf>. Acesso em: 23 maio 2018.

POFFO, Recígio. **SERVIÇO DE PROTOCOLAÇÃO DIGITAL DE DOCUMENTOS ELETRÔNICOS.** 2010. 66 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade Regional de Blumenau, Blumenau, 2010. Disponível em: <http://campeche.inf.furb.br/tccs/2010-II/TCC2010-2-24-VF-RecigioPoffo.pdf>. Acesso em: 23 maio 2018.

SAGE. **Entenda as diferenças entre o certificado digital A1 e A3.** 2018. Disponível em: <https://blog.sage.com.br/certificado-digital-a1-e-a3/>. Acesso em: 09 jun. 2018.

SERASA. **Afinal, qual a diferença entre o Certificado Digital A1 e o A3?** 2016. Disponível em: <https://serasa.certificadodigital.com.br/novidades/afinal-qual-a-diferenca-entre-o-certificado-digital-a1-e-o-a3/>. Acesso em: 10 jun. 2018.

SETTI, Rodrigo JosÉ. **ESTUDO DO GED – GERENCIAMENTO ELETRÔNICO DE DOCUMENTOS NO ÂMBITO ORGANIZACIONAL.** 2008. 62 f. Monografia (Especialização) - Curso de Gestão da Informação, Universidade Federal do ParanÁ, Curitiba,

2008. Disponível em: <https://acervodigital.ufpr.br/bitstream/handle/1884/48210/TCC - Rodrigo Jose Setti.pdf?sequence=1&isAllowed=y>. Acesso em: 13 jun. 2018.

STALLINGS, William. **CRIPTOGRAFIA E SEGURANÇA DE REDE:** PRINCÍPIOS E PRÁTICAS. 6. ed. São Paulo: Pearson, 2015. 558 p.

TECMUNDO. **O que é hash.** 2009. Disponível em: <https://www.tecmundo.com.br/o-que-e/1663-o-que-e-hash-.htm>. Acesso em: 13 jun. 2018.

VUEJS. **Introdução:** O que é Vue.js?. 2018. Disponível em: <https://br.vuejs.org/v2/guide/index.html>. Acesso em: 23 maio 2018.