UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TÉCNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Arthur Moreira Rodrigues Alves

SERVIÇO DE EMISSÃO DE CERTIFICADOS DIGITAIS COM BASE EM DADOS DE CNHs DIGITAIS

Florianópolis
2022

Arthur Moreira Rodrigues Alves

SERVIÇO DE EMISSÃO DE CERTIFICADOS DIGITAIS COM BASE EM DADOS DE CNHs DIGITAIS

Trabalho de Conclusão de Curso do Curso de Graduação em Ciências da Computação do Centro Técnológico da Universidade Federal de Santa Catarina para a obtenção do título de bacharel em Ciências da Computação.
Orientador: Prof. Jean Everson Martina, Dr.
Coorientador: Pablo Rinco Montezano

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Alves, Arthur Moreira Rodrigues
SERVIÇO DE EMISSÃO DE CERTIFICADOS DIGITAIS COM BASE EM DADOS DE CNHs DIGITAIS / Arthur Moreira Rodrigues Alves ; orientador, Jean Everson Martina, coorientador, Pablo Rinco Montezano, 2022.
70 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2022.

Inclui referências.

1. Ciências da Computação. 2. Certificados Digitais. 3. Certificados Digitais Avançados. 4. CNHs Digitais. I. Martina, Jean Everson. II. Montezano, Pablo Rinco. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Arthur Moreira Rodrigues Alves

SERVIÇO DE EMISSÃO DE CERTIFICADOS DIGITAIS COM BASE EM DADOS DE CNHs DIGITAIS

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "bacharel em Ciências da Computação" e aprovado em sua forma final pelo Curso de Graduação em Ciências da Computação.

Florianópolis, 19 de Dezembro de 2022.

Prof. Jean Everson Martina, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Jean Everson Martina, Dr.
Orientador
Universidade Federal de Santa Catarina

Pablo Rinco Montezano
Coorientador
Universidade Federal de Santa Catarina

Lucas Mayr de Athayde
Avaliador
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus amigos, família e mentores que me ajudaram a seguir em frente quando eu não acreditava ser possível.

AGRADECIMENTOS

Agradeço aos meus amigos por serem meus companheiros nas aventuras que a vida adulta e universitária proporcionam. Aos meus mentores, agradeço pelos ensinamentos que me passaram. A minha mãe e meu irmão eu agradeço por todo o amor e suporte, sem eles eu jamais seria sido capaz de começar essa caminhada.

"Blackbird singing in the dead of night
Take these broken wings and learn to fly
All your life
You were only waiting for this moment to arise"
(The Beatles, 1968)

RESUMO

Certificados digitais são documentos eletrônicos que podem ser utilizados para identificar uma entidade e prover outras funcionalidades como assinaturas digitais. No Brasil, a principal infraestrutura de chaves públicas é a ICP-Brasil, instaurada pela Medida Provisória 2.200 no ano de 2001. A autoridade raiz desta ICP é o Instituto Nacional de Tecnologia e Informação (ITI), responsável por definir as políticas de certificação e requisitos de operações que as entidades integrantes da ICP-Brasil devem aplicar e seguir. Os certificados digitais emitidos pela ICP-Brasil são considerados certificados qualificados, e possuem validade equivalente a assinaturas físicas. Por conta dos custos de operação, as entidades integrantes da ICP-Brasil aplicam taxas aos certificados emitidos por elas. Para permitir que a população possa utilizar serviços associados a certificados digitais, o governo fornece certificados digitais gratuitos, que podem ser emitidos utilizando o serviço Gov.br. Alguém que deseje utilizar este serviço deve se autenticar em plataformas consideradas confiáveis pelo Gov.Br. Este processo pode ser difícil mesmo para usuários familiarizados com serviços digitais. Como alternativa para este processo, este trabalho descreve a implementação de um serviço capaz de emitir certificados digitais utilizando CNHs digitais. O sistema desenvolvido segue usa o framework Spring da linguagem Java, utiliza o Tesseract para extrair os dados dos usuários dos documentos de CNH e utiliza o Hawa CA para emitir os certificados desejados. A estrutura do sistema desenvolvido e suas capacidades também são descritas no texto.

Palavras-chave: Certificados Digitais. Certificados Digitais Avançados. CNHs Digitais.

ABSTRACT

Digital certificates are electronic documents that can be used to identify an entity and also provide other applications such as digital signatures. In Brazil, the main public key infrastructure is ICP-Brasil, and it's root authority is the Instituto Nacional de Tecnologia e Informação (ITI), responsible to define the certificate policies and operational requirements that all the authorities which integrate ICP-Brasil should apply and follow. The digital certificates issued by entities of ICP-Brasil are considered qualified certificates, and have validity equivalent to physical signatures. Because of the costs related to the operation, entities of ICP-Brasil charge the certificates issued by them. To allow the population to use services related to digital certificates the government provides free digital certificates to use that can be issued using the Gov.br services. To use these services a person needs to authenticate itself in the platforms trusted by the Gov.br. This process can be difficult even for users familiarized with digital services. As an alternative to this process, this work describes the implementation of a service capable of issuing digital certificates using digital CNHs. The developed system uses the Spring framework of Java language, utilizes Tesseract to extract the user data from the CNH documents and utilizes Hawa CA to issue the desired certificates. The structure of the developed system and its capabilities are also described in the text.

Keywords: Digital Certificates. Advanced Digital Certificates. CNHs Digitais.

LISTA DE FIGURAS

Figura 1 – Modelo de arquivo 1. 28
Figura 2 – Modelo de arquivo 2. 29
Figura 3 – Modelo de arquivo 3. 30
Figura 4 – Imagem de CNH no formato novo. 32
Figura 5 – Imagem de CNH no formato antigo. 32
Figura 6 – Extração de campo específico da imagem de CNH no formato novo. . . 33
Figura 7 – Extração de campo específico da imagem de CNH no formato antigo. . 33
Figura 8 – Fluxo de criação do Dossiê. 36
Figura 9 – Visão geral do sistema. 39
Figura 10 – Uso direto da figura 3 com Tesseract. 56
Figura 11 – Uso direto da figura 1 com Tesseract. 57
Figura 12 – Tentativa extração de texto diretamente da imagem de CNH no formato novo. 58
Figura 13 – Tentativa extração de texto diretamente da imagem de CNH no formato antigo. 59

LISTA DE QUADROS

Quadro 1 – Valores para Page Segmentation Mode. 36
Quadro 2 – Resultados dos testes para o *endpoint /issue* 36
Quadro 3 – Resultados dos testes para o *endpoint /revoke* 37
Quadro 4 – Resultados dos testes para o *endpoint /get-cert* 37
Quadro 5 – Services implementados. 38
Quadro 6 – Estrutura do corpo da requisição do *endpoint /issue*. 40
Quadro 7 – Respostas do *endpoint /issue*. 40
Quadro 8 – Parâmetros de requisição do *endpoint /revoke*. 40
Quadro 9 – Respostas do *endpoint /revoke*. 40
Quadro 10 – Respostas para o restante dos endpoints. 41
Quadro 11 – Variáveis de ambiente gerais. 42
Quadro 12 – Variáveis de ambiente relacionadas a Java Keystore (JKS). 43
Quadro 13 – Variáveis de ambiente relacionadas ao Tesseract. 43

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
API	Application Programming Interface
AR	Autoridade Registradora
CNH	Carteira Nacional de Habilitação
CPF	Cadastro de Pessoa Física
DETRAN	Departamento Estadual de Trânsito
HSM	Hardware Secure Module
ICP	Infraestrutura de Chaves Públicas
ITI	Instituto Nacional de Tecnologia da Informação
JKS	Java Keystore
OCR	Optical Character Recognition
PDF	Portable Document Format
SENATRAN	Secretaria Nacional de Trânsito

SUMÁRIO

1 INTRODUÇÃO 14
1.1 DESCRIÇÃO DO PROBLEMA 15
1.2 MOTIVAÇÃO E JUSTIFICATIVA 15
1.3 OBJETIVOS 16
1.3.1 Objetivo Geral 16
1.3.2 Objetivos Específicos 16
1.3.3 Escopo 16
1.4 ORGANIZAÇÃO DOS CAPÍTULOS 16
2 FUNDAMENTAÇÃO TEÓRICA 18
2.1 ASSINATURAS ELETRÔNICAS 18
2.2 CRIPTOGRAFIA 18
2.3 OBJETIVOS DA CRIPTOGRAFIA 19
2.4 CRIPTOGRAFIA DE CHAVES SIMÉTRICAS 20
2.5 CRIPTOGRAFIA DE CHAVES ASSIMÉTRICAS 20
2.6 FUNÇÕES DE *HASH* 21
2.7 ASSINATURA DIGITAL 21
2.8 INFRAESTRUTURA DE CHAVES PÚBLICAS 22
2.9 CERTIFICADOS DIGITAIS 23
2.9.1 Certificados Digitais Avançados 23
2.9.2 Certificados Digitais Qualificados 24
3 DESENVOLVIMENTO E METODOLOGIA 25
3.1 TECNOLOGIAS 25
3.1.1 Linguagem de programação 25
3.1.2 Spring Boot 25
3.1.3 Base de dados 25
3.1.4 Tesseract 25
3.1.5 Portable Document Format (PDF)Box 26
3.1.6 APIs 26
3.1.6.1 Verificador de Conformidade 26
3.1.6.2 Hawa CA 26
3.1.7 Docker 26
3.2 DESENVOLVIMENTO 26
3.2.1 Verificação da assinatura do arquivo de Carteira Nacional de Habilitação (CNH) Digital 27
3.2.2 Formatos diferentes de arquivo de CNH 27
3.2.3 Extrair dados das CNHs 31
3.2.4 Chaves necessárias 35

3.2.5 Persistir as informações dos dados usados para emitir os certificados 35
3.2.6 Ajustes nos parâmetros do Tesseract 35
4 RESULTADOS 38
4.1 VISÃO GERAL DO SISTEMA 38
4.2 FUNCIONALIDADES IMPLEMENTADAS 39
4.2.1 Emissão 39
4.2.2 Revogação 40
4.2.3 Obter certificados emitidos e documentos associados 41
4.3 CONFIGURAÇÃO DO SISTEMA 42
4.3.1 Variáveis de ambiente 42
4.3.2 Execução 44
4.4 LIMITAÇÕES 44
5 CONCLUSÕES 46
5.1 TRABALHOS FUTUROS 46
REFERÊNCIAS 47
APÊNDICE A – ESTRUTURA DO DOSSIÊ ASSINADO 50
APÊNDICE B – ESTRUTURA DE CNHSERVICEISSUERESPONSE 51
APÊNDICE C – ESTRUTURA DE EXTRACTEDCNHINFO . . . 52
APÊNDICE D – ESTRUTURA DE SIMPLEMESSAGERESPONSE 53
APÊNDICE E – ESTRUTURA DE CNHSERVICEREVOKERESPONSE 54
APÊNDICE F – CÓDIGO FONTE 55
APÊNDICE G – USO DIRETO DO TESSERACT COM ARQUIVO DE CNH NO FORMATO NOVO 56
APÊNDICE H – USO DIRETO DO TESSERACT COM ARQUIVO DE CNH NO FORMATO ANTIGO . . 57
APÊNDICE I – USO DIRETO DO TESSERACT COM IMAGEM DE CNH NO FORMATO NOVO 58
APÊNDICE J – USO DIRETO DO TESSERACT COM IMAGEM DE CNH NO FORMATO ANTIGO 59
APÊNDICE K – ARTIGO NO FORMATO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO 60
ANEXO A – ESTRUTURA DA RESPOSTA DO SERVIÇO VERIFICADOR ITI 70

1 INTRODUÇÃO

Assinaturas eletrônicas são meios que permitem criar uma relação de autoria entre uma entidade e um conjunto de dados eletrônicos. O simples ato de renomear um arquivo ou adicionar uma assinatura digitalizada pode categorizar uma assinatura eletrônica.

Do ponto de vista prático se espera que as assinaturas eletrônicas sejam o equivalente a assinaturas físicas, o que auxilia a implementação de serviços digitais baseados em características como autenticação e integridade. Entretanto, nem todos os mecanismos de assinatura eletrônica são capazes de garantir tais características.

As assinaturas digitais são um meio de implementação de assinaturas eletrônicas que podem garantir as características mencionadas anteriormente de maneira relativamente simples e eficiente. O seu uso requer a emissão de certificados digitais, documentos eletrônicos que relacionam uma chave pública, elemento necessário para execução desse tipo de assinatura, a uma entidade.

Os certificados digitais garantem a todos que decidem confiar neles que determinada chave pública identifica diretamente uma entidade no meio digital. Essa afirmação garante que qualquer transação, que pode ser entendida como algum tipo de comunicação, realizada entre partes que possuam certificados digitais as propriedades de confidencialidade, integridade, autenticidade e não repúdio.

Geralmente uma Autoridade Certificadora (AC) é a entidade responsável por emitir um certificado digital. Para garantir a confiabilidade de um certificado as ACs requisitam e validam um conjunto de informações das entidades que desejam obtê-los. Apenas as entidades que atendem aos critérios de validação das ACs têm seus certificados emitidos.

As ACs geralmente integram algo chamado Infraestrutura de Chaves Públicas (ICP), uma estrutura de entidades organizadas de maneira hierárquica que realizam ações relacionadas a emissão e manutenção de certificados digitais seguindo determinados padrões de operação.

No Brasil a principal ICP em operação é a ICP-Brasil. As assinaturas digitais realizadas com os certificados emitidos por ela possuem validade e aplicabilidade, inclusive no âmbito jurídico, idênticas a de uma assinatura física.

No fim das contas, os gastos associados às diversas tarefas e infraestrutura que uma entidade integrante de um ICP deve executar e manter, especialmente aquelas que integram a ICP-Brasil, são cobertos com taxas cobradas para emissão de certificados. Entretanto, essas taxas associadas à emissão de determinados certificados, principalmente os que possuem vasta aplicabilidade, acabam sendo um impeditivo para o uso deste tipo de assinatura.

Para contornar este problema o governo disponibiliza serviços como o Gov.br, que permite realizar a emissão gratuita de certificados digitais. Em contrapartida os certificados emitidos por esse serviço acabam tendo uma aplicabilidade limitada; além

disso, para emiti-los também são exigidas diversas etapas de verificação de identidade através de informações provenientes de outras instituições, como bancos e órgãos públicos.

Entretanto, esse processo de verificação de identidade através de outros serviços é algo que também pode acabar se tornando um fator impeditivo para o uso deste serviço. Isto ocorre devido a própria premissa do processo de verificação, que necessita que o usuário utilize os serviços suportados. Além disso, o próprio processo de autorização de uso dos dados armazenados nas bases de dados destes serviços pode ser complicado e extensivo.

Com base na complexidade envolvida no processo de verificação de identidade para emissão de certificados avançados em serviços como o Gov.br, este trabalho desenvolve um sistema em formato de Application Programming Interface (API) capaz de realizar a emissão de certificados digitais avançados. Para validar a identidade dos usuários, o sistema utiliza a versão digital de Carteiras Nacionais de Trânsito (CNH), documentos gerados e assinados digitalmente pelos departamentos estaduais de trânsito.

1.1 DESCRIÇÃO DO PROBLEMA

Os passos de autenticação necessários para a emissão de certificados digitais avançados em serviços como o Gov.br[1] podem ser complexos demais, uma fator que pode acabar impedindo pessoas leigas de emitir certificados digitais.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Serviços que fornecem certificados de maneira gratuita como o Gov.br geralmente demandam um processo de construção de identidade. Nesse processo dados de diversas fontes como órgãos públicos e bancos são utilizados para criar um perfil e efetivamente provar que alguém realmente existe; nesse modelo quanto mais fontes de dados, mais confiável é o fato da existência de uma pessoa.

Sendo assim esse processo, apesar de confiável, muitas vezes acaba se tornando complexo e demorado, principalmente para pessoas com pouca familiaridade com tecnologia que podem acabar deixando de utilizar tal tipo de serviço. Com base nesse problema, a elaboração de uma aplicação capaz de emitir certificados digitais utilizando apenas documentos pré-validados é uma maneira de diminuir a dificuldade atualmente presente nos serviços de emissão disponíveis, consequentemente aumentando a acessibilidade a esse tipo de serviço.

[1] https://www.gov.br/pt-br

1.3 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste TCC.

1.3.1 Objetivo Geral

Desenvolvimento de uma API que realize a emissão de certificados digitais avançados com o sistema Hawa Ca usando os dados contidos em CNHs digitais. A API poderá então ser utilizada como base para a elaboração de sistemas mais robustos que facilitem a obtenção certificados digitais por parte de pessoas leigas.

1.3.2 Objetivos Específicos

Os objetivos específicos deste trabalho de conclusão de curso são:

a) Apresentar conceitos relacionados a assinaturas eletrônicas voltados para assinaturas digitais.

b) Desenvolver uma aplicação capaz de validar CNHs digitais e extrair os dados contidos nelas;

c) Desenvolver uma aplicação deve ser capaz de realizar requisições para o sistema Hawa a fim de emitir e revogar certificados;

d) Desenvolver uma aplicação capaz de retornar um certificado previamente emitido;

e) Apresentar detalhes referentes às escolhas e tecnologias adotadas no processo de desenvolvimento da aplicação;

f) Apresentar dos resultados obtidos com a aplicação desenvolvida;

1.3.3 Escopo

O desenvolvimento da solução se limita ao desenvolvimento de uma API que permita emitir e revogar certificados, além de obter dados referentes a certificados já emitidos. O mecanismo de verificação utilizado pelo sistema se baseia em documentos de CNH digital. A solução desenvolvida não segue nenhum tipo de regulamentação referente a emissão de certificados. O trabalho não contempla o uso de mecanismos mais sofisticados para o gerenciamento das chaves criptográficas necessárias para algumas operações.

1.4 ORGANIZAÇÃO DOS CAPÍTULOS

No primeiro capítulo deste trabalho é apresentada a organização dos capítulos usada, além da motivação e objetivos do trabalho. Em seguida é apresentada a fundamentação

teórica, e no terceiro capítulo é descrito o processo de desenvolvimento da aplicação. No quarto capítulo são discutidos os resultados obtidos com o desenvolvimento da aplicação e as limitações encontradas, e logo após são apresentadas as conclusões e sugestões para trabalhos futuros. Por fim são apresentadas as referências utilizadas.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão definidos conceitos associados à área de segurança da computação pertinentes a execução do trabalho.

2.1 ASSINATURAS ELETRÔNICAS

Uma assinatura eletrônica é um conjunto de dados que pode ser associado a outro, ambos em formato digital, de forma que uma relação de autoria seja criada sobre os dados do segundo conjunto (RESOLUÇÃO. . . , 2021). Dados como o texto digitado em um arquivo ou o nome de um documento digital podem compor o conjunto que garante a autoria; esses dados são a assinatura eletrônica que identifica alguém ou algo. O conjunto de dados que é relacionado a alguém é chamado dado assinado, e pode ser qualquer informação que esteja em formato digital. O processo de assinatura, que pode ser o simples ato de digitar um nome ou digitalizar uma assinatura física, é o que atribui uma assinatura eletrônica a um conjunto de dados, o tornando um dado assinado.

Como uma assinatura pode ser realizada de diversas formas, existem termos que caracterizam tipos específicos de assinaturas eletrônicas. Atualmente apenas um conjunto específico de assinaturas eletrônicas podem ser usadas em contextos como o jurídico e empresarial; isso ocorre pois nem todo tipo de assinatura eletrônica consegue manter uma lista de características que as mantenham confiáveis mediante diversos casos de uso. Este conjunto de características será apresentado e definido na Seção 2.7 e se relaciona com o tipo de assinatura eletrônica que será usada para o desenvolvimento da aplicação deste trabalho, a assinatura digital. Para melhor compreensão do tópico serão apresentados os conceitos de criptografia e chaves simétrica e assimétrica.

2.2 CRIPTOGRAFIA

Devido ao grande volume de dados usados e compartilhados por aplicações atualmente muitas vezes se faz necessário o uso de algum tipo de ferramenta que possa garantir a autenticidade e segurança das informações contidadas neles (YASSEIN *et al.*, 2017). E ferramentas de criptografia são amplamente usadas para esta finalidade no contexto de computação.

Criptografia é a ciência de manter informações em segredo (DELFS; KNEBL, 2002). Isso é feito ao se converter a mensagem que se deseja transmitir, chamada de texto puro, em uma mensagem que é incompreensível, chamada de texto cifrado. O texto puro é transformado em um texto cifrado através da cifragem, processo no qual a informação original é transformada em algo que pode ser compreendido apenas por partes que saibam realizar o processo necessário para decifrá-la, no qual é aplicado o algoritmo reverso ao algoritmo de cifragem usado (AGRAWAL; MISHRA, 2012).

Para que os processos de cifrar e decifrar possam ser realizados, além do uso de algum algoritmo também é necessário o uso de uma ou mais chaves, um texto numérico ou alfanumérico que é utilizado pelo algoritmo de cifragem escolhido (AGRAWAL; MISHRA, 2012). O uso de uma chave e um texto pleno únicos por um algoritmo de criptografia gera um texto cifrado específico e incompreensível que apenas pode ser revertido para sua forma original alimentando o algoritmo de decifração apropriado com o texto cifrado gerado e a mesma chave utilizada pelo algoritmo de cifragem.

Desta maneira, duas partes que tenham interesse em trocar informações de maneira segura e confidencial podem acordar o uso de algum tipo de algoritmo de cifragem, e por consequência do algoritmo de decifração apropriado, e uma chave; o remetente da informação cifra a informação original e a transmite para o destinatário, que poderá revertê-la para seu estado original com o uso do algoritmo de decifragem e chave apropriados.

Para que esse exemplo tenha sucesso, o destinatário precisa conseguir acessar a chave apropriada. Dependendo do algoritmo usado, a chave necessária para revertê-lo pode ser a mesma ou alguma muito relacionada a usada no processo de cifragem. Em criptografia o grau de semelhança das chaves usadas no processo de cifragem e decifragem é o que divide o conjunto de algoritmos de cifragem em simétricos e assimétricos, subconjuntos que serão definidos posteriormente.

2.3 OBJETIVOS DA CRIPTOGRAFIA

Prover confidencialidade não é o único objetivo da criptografia (DELFS; KNEBL, 2002). Além dela o uso de criptografia também tem como objetivo garantir integridade de dados, autenticação e não repúdio (DELFS; KNEBL, 2002), além de controle de acesso (YASSEIN *et al.*, 2017). As definições para esses conceitos e como os mesmos podem ser traduzidos para problemas comuns de computação serão apresentados a seguir.

A confidencialidade é atingida quando uma informação que deve ser transmitida é acessada apenas pelas partes a qual é destinada (SURYA; C.DIVIYA, 2011). A integridade de dados diz respeito à capacidade que o destinatário de uma mensagem de averiguar se a mesma foi alterada de alguma maneira (DELFS; KNEBL, 2002). Autenticidade define a capacidade do destinatário de uma mensagem de verificar a origem do que foi recebido; desta forma não deve ser possível que uma parte se passe por outra durante o processo de troca de mensagens, e os envolvidos na comunicação devem conseguir identificar uns aos outros (DELFS; KNEBL, 2002). O não repúdio trata de meios que impossibilitam o remetente de posteriormente negar o envio de uma mensagem (DELFS; KNEBL, 2002). Por fim, o controle de acesso trata dos meios pelos quais pode-se garantir que uma informação seja acessada apenas por quem é autorizado para tal (SURYA; C.DIVIYA, 2011).

No contexto de computação, a aplicação desses objetivos pode ser observada no processo de transmissão de informações sigilosas em formato eletrônico. Independente do meio pelo qual isso é feito, na grande maioria dos casos se espera que os dados sendo

transmitidos entre computadores não possam ser compreendidos por ninguém que os intercepte, ou seja, tenham sua confidencialidade preservada; além disso esses dados não podem ser alterados durante o processo, e que caso isso ocorra, o destinatário deve ser capaz de identificar esta mudança, constatando sua integridade. Também é esperado que os computadores envolvidos no processo sejam capazes de se identificar e constatar a origem das mensagens que recebem, atestando a autenticidade das informações compartilhadas e partes envolvidas.

Finalmente, sobre os dados que trocados é esperado o acesso exclusivo apenas as partes autorizadas, além da incapacidade de uma máquina de negar ter enviado alguma mensagem que ela transmitiu, duas ideias que se traduzem respectivamente em controle de acesso e não repúdio.

Os mais diversos algoritmos de cifragem atendem a todos, ou alguns, desses objetivos. Por consequência a lista de objetivos garantidos por um algoritmo acaba por especificar as finalidades para as quais ele pode ser utilizado.

2.4 CRIPTOGRAFIA DE CHAVES SIMÉTRICAS

Criptografia de chaves simétricas diz respeito a técnicas de criptografia nas quais uma mesma chave é usada tanto para cifrar e decifrar uma mensagem (SURYA; C.DIVIYA, 2011). Técnicas desse tipo garantem segredo no processo de comunicação entre duas partes, de forma que qualquer parte interessada em interceptar as mensagens trocadas não obtém nenhum tipo de informação útil (DELFS; KNEBL, 2002).

Algoritmos de criptografia simétrica são bastante eficientes em processar grandes quantidades de dados (SURYA; C.DIVIYA, 2011), podendo ser quase 1000 vezes mais rápidos que algoritmos baseados em chaves assimétricas por necessitar de menos poder computacional para serem executados (YASSEIN *et al.*, 2017). Tais tipos de algoritmos podem garantir integridade e confidencialidade com seu uso (DELFS; KNEBL, 2002).

Entretanto, um problema comum para algoritmos desse tipo é a maneira como a chave utilizada no processo é transmitida de uma parte a outra. Soluções para essa tarefa podem envolver o uso de outros algoritmos de criptografia para transmissão inicial da chave (DELFS; KNEBL, 2002), porém não eliminam esta característica intrínseca dos algoritmos pertencentes a esse grupo.

2.5 CRIPTOGRAFIA DE CHAVES ASSIMÉTRICAS

Criptografia de chaves assimétricas diz respeito a um conjunto de técnicas de cifragem na qual são utilizadas duas chaves distintas, porém relacionadas, para o processo de cifrar e decifrar uma mensagem. Neste tipo de sistema criptográfico as mensagens são cifradas com uma chave e decifradas com outra. Uma destas chaves é chamada de chave

pública, e geralmente é compartilhada. A outra chave é chamada de chave privada, e é utilizada apenas pelo proprietário.

Para comunicações com esse tipo de algoritmo, o remetente usa a chave pública do destinatário para cifrar sua mensagem; fazendo isso o conteúdo da mensagem pode ser decifrado apenas pelo destinatário, que contém a chave secreta associada à chave pública. Enquanto cifrada a mensagem se torna incompreensível, garantindo a confidencialidade da informação. Estas características fazem os algoritmos de chave pública fortes candidatos como meio de transmissão de chaves secretas para algoritmos de criptografia simétrica (DELFS; KNEBL, 2002).

Complementando seu leque de casos de uso, funcionalidade desse tipo de algoritmo como a assinatura digital são necessários para garantir aspectos como autenticidade e não repúdio (DELFS; KNEBL, 2002) no meio digital; essas características podem ser observadas em cenários como a assinaturas de contratos ou transações bancárias, e devem ser garantidas em sistemas computacionais que as implementem.

2.6 FUNÇÕES DE *HASH*

Funções de Hash recebem como entrada uma mensagem em formato binário de tamanho qualquer e produzem uma sequência de bits de tamanho fixo (DELFS; KNEBL, 2002). A principal característica de funções hash é o fato de serem *one-way functions*; funções deste tipo permitem calcular facilmente uma saída para uma determinada entrada, enquanto o processo de cálculo da entrada com base em uma saída é complexo e inviável computacionalmente. Isso significa que para uma sequência de bits qualquer é computacionalmente inviável encontrar a mensagem para qual a aplicação da função de hash a geraria (DELFS; KNEBL, 2002).

As funções de hash não necessariamente mapeiam cada mensagem para uma sequência de bits única. Uma boa função de hash tem seu valor associado ao fato de que para um conjunto de saídas de tamanho arbitrariamente grande se torna inviável encontrar outra mensagem que gere o mesmo resultado.

Graças a estas propriedades, as funções de hash criptográficas são comumente usadas para armazenamento de senhas em sistemas computacionais e também na implementação de assinaturas digitais.

2.7 ASSINATURA DIGITAL

Assinaturas digitais são mecanismos de autenticação que possibilitam ao remetente de uma mensagem um meio adicionar um código que pode ser usado como uma assinatura (KAUR; KAUR, 2012). Tal mecanismo fornece um meio de se provar a autoria de uma mensagem, e geralmente são implementados com algoritmos de criptografia assimétrica.

O remetente usa sua chave secreta para assinar o hash de sua mensagem; o resultado é a sua assinatura digital. A assinatura digital é enviada em um pacote que também contém a mensagem original e a chave pública associada à chave secreta usada. O recebedor do pacote realiza seu desempacotamento e obtém a mensagem, assinatura e chave pública.

Para comprovar que a integridade e autenticidade da assinatura recebida primeiramente aplica-se o mesmo algoritmo de *hash* usado pelo remetente ao texto recebido; em seguida o destinatário usa a chave pública para decifrar a assinatura, ambas contidas no pacote recebido. Por fim, o remetente compara o hash obtido pela assinatura decifrada com o hash gerado localmente; caso sejam iguais o dado recebido é considerado íntegro, e a sua autenticidade é comprovada em relação a chave pública recebida; por garantir estas características os mecanismos de assinatura digital são considerados meios de implementação de assinaturas eletrônicas.

Apesar das garantias que o uso de assinaturas digitais proporcionam, ainda existe o seguinte problema: um impostor pode se passar por outra entidade, utilizando um par de chaves próprias. Os mecanismos de assinatura digital não são capazes de identificar a quem uma chave pública pertence, e esta limitação inspirou o desenvolvimento de um mecanismo que tratasse desse problema: certificados digitais (KOHNFELDER, 1978).

2.8 INFRAESTRUTURA DE CHAVES PÚBLICAS

Para distribuir e verificar certificados digitais é necessária a existência de ICPs. Estas infraestruturas provêm o gerenciamento de chaves públicas usadas para suportar serviços de autenticação, cifragem, integridade e não repúdio (RECOMMENDATION..., 2019). O seu principal objetivo é prover e gerenciar certificados digitais que ligam uma chave pública a uma entidade de maneiras que terceiros sejam capazes de validar esse vínculo (SCHUKAT; CORTIJO, 2015).

Por consequência as ICPs acabam sendo bases sobre as quais aplicações e sistemas de segurança são construídas (WEISE, 2001). Alguns exemplos de sistemas dependentes do uso de mecanismos de ICPs são serviços de email, aplicações que usam cartões com chips, sistemas bancários e de comércio eletrônico (WEISE, 2001).

No Brasil a principal infraestrutura de chaves públicas é a ICP-Brasil, instituída pela Medida Provisória 2.200-2 de 2001 (MEDIDA..., 2001).

Duas entidades importantes que integram uma ICP são autoridades certificadoras e registradoras. Uma AC é uma entidade na qual uma ou mais entidades confiam para criar e assinar certificados digitais, e opcionalmente emitir as chaves públicas contidas nesses certificados (RECOMMENDATION..., 2019).

Uma Autoridade Registradora (AR) é uma entidade cujo a função é identificar e autenticar os dados de uma entidade para qual uma AC emitirá um certificado (RECOMMENDATION..., 2019). As responsabilidades atribuídas a uma AR podem ou não ser atribuídas a uma autoridade certificadora (RECOMMENDATION..., 2019).

2.9 CERTIFICADOS DIGITAIS

Um certificado digital é um documento eletrônico que provê informações para provar a identidade de uma entidade associando-a com uma chave pública (AFSHAR, 2015). Desta maneira qualquer um que deseje se comunicar com o suposto dono de uma chave pública que possui uma certificado associado pode validar o relacionamento dos dados. Os certificados contém diversas informações pertencentes ao detentor de uma chave pública, além da assinatura e algoritmo usado pela entidade emissora para obtenção da assinatura. Como as assinaturas contidas em certificados podem ser verificadas de maneira independente, os mesmos podem ser distribuídos por conexões inseguras e servidores, além de serem guardados em sistemas de armazenamento inseguro sem maiores problemas (COOPER *et al.*, 2022).

Como exemplo de uso pode-se citar o processo de comunicação entre um navegador e um servidor qualquer. O navegador inicialmente tenta se comunicar com o servidor, que para provar sua identidade disponibiliza um certificado digital. Inicialmente o navegador verifica se confia no emissor de certificado, e caso isso seja verdade ele de alguma forma terá acesso a chave pública necessária para prosseguir com a validação.

Após a etapa inicial o navegador verifica os algoritmos usados para gerar a assinatura, e com o algoritmo de *hash* especificado gera um novo resumo criptográfico usando como base os dados pertencentes ao detentor da chave contidos no certificado. Com base no algoritmo de cifragem especificado e na chave pública associada ao emissor do certificado, o navegador tenta decifrar a assinatura e obter o hash gerado pela entidade emissora do certificado. Por fim o navegador compara o hash calculado localmente com o hash contido na assinatura, e se ambos forem iguais ele utiliza a chave pública contida no certificado para se comunicar com o servidor de maneira segura.

Atualmente a grande maioria dos certificados segue o padrão X.509 (COOPER *et al.*, 2022), que já se encontra em sua terceira versão (AFSHAR, 2015).

2.9.1 Certificados Digitais Avançados

Certificados digitais avançados são documentos eletrônicos que possibilitam a implementação de assinaturas eletrônicas avançadas. Para ser considerada uma assinatura eletrônica avançada uma assinatura eletrônica deve atender aos seguintes requisitos (REGULATION. . . , 2014):

a) ser unicamente ligada ao assinante;

b) ser capaz de identificar seu assinante;

c) ser criada com dados de criação de assinatura eletrônica que o assinante pode, com alto nível de confiança, usar com controle exclusivo;

d) ser ligada ao dado assinado de maneira que qualquer mudança realizada nos dados assinados possa ser identificada;

Por atenderem a esses requisitos, certificados digitais podem ser utilizados como meio de implementação de assinaturas eletrônicas avançadas. Os certificados que atendem a todos os requisitos apresentados anteriormente são considerados certificados digitais avançados.

No Brasil um certificado é considerado avançado quando a sua emissão é feita por entidades que não fazem parte da ICP-Brasil. O uso e validade desse tipo de certificado fica limitado a situações acordadas entre o proprietário do certificado e outras partes.

2.9.2 Certificados Digitais Qualificados

Certificados digitais qualificados são um tipo de certificado digital avançado que é emitido por entidades que seguem modelos definidos legalmente e tem como objetivo identificar entidades com um alto nível de confiança (SANTESSON; NYSTROM; POLK, 2004).

Na prática as características específicas que definem quando um certificado pode ser considerado qualificado variam de acordo com o país. No Brasil, os certificados são considerados qualificados quando emitidos por entidades integrantes da ICP-Brasil e possuem validade legal reconhecida (MEDIDA..., 2001).

3 DESENVOLVIMENTO E METODOLOGIA

Nessa seção serão apresentados diferentes aspectos referentes à elaboração da solução como tecnologias usadas, problemas e soluções encontradas no processo de desenvolvimento.

3.1 TECNOLOGIAS

A seguir será apresentada a linguagem de programação na qual a aplicação foi desenvolvida. Além disso também serão apresentadas bibliotecas que foram escolhidas para solucionar problemas identificados no processo de desenvolvimento, que será detalhado em seções posteriores.

3.1.1 Linguagem de programação

A linguagem de programação escolhida para o desenvolvimento da aplicação foi a Java da Oracle, em sua versão 11.0.16. A escolha se deu por três motivos, sendo o primeiro a afinidade e preferência pessoal do autor com a linguagem; o segundo motivo é o *framework* Spring Boot, escolhido como base para o desenvolvimento da aplicação. Por fim, o último motivo foi a presença de bibliotecas para integração da linguagem com outros componentes necessários para implementação da solução.

3.1.2 Spring Boot

Spring Boot é um framework usado para o desenvolvimento de aplicações web, especialmente APIs; o framework provê acesso a diversos componentes que facilitam a interação com APIs externas, base de dados e configuração de servidores; devido a familiaridade do autor e a necessidade de utilização de todos estes componentes o framework foi escolhido como base para o desenvolvimento da solução.

3.1.3 Base de dados

O sistema para gerenciamento da base de dados escolhido foi o MySQL; a escolha foi feita com base na familiaridade do autor e na presença de ferramentas no framework Spring Boot que facilitam a integração da solução com bases de dados deste tipo.

3.1.4 Tesseract

O Tesseract é um software de Optical Character Recognition (OCR) que permite o reconhecimento de caracteres contidos em imagens; ele pode ser executado em diversos sistemas operacionais e possui bibliotecas de integração com diversas linguagens de programação. O Tesseract foi escolhido por ser uma opção *open source*, confiável e por

possuir bibliotecas, também de código aberto, para integração com a linguagem na qual o trabalho foi desenvolvido. A biblioteca utilizada para integração com a solução foi o Tess4J.

3.1.5 PDFBox

O PDFBox é uma biblioteca para linguagem Java que permite a manipulação de PDFs; com ela é possível criar documentos PDF, além de manipular documentos já existentes. O fato de ser uma biblioteca de código aberto foi o motivo para sua escolha.

3.1.6 APIs

As APIs abaixo foram utilizadas para prover algumas funcionalidades necessárias para a implementação da solução.

3.1.6.1 Verificador de Conformidade

O Verificador de Conformidade é um serviço que pode ser consumido como uma aplicação web ou API; ele é mantido pelo Instituto Nacional de Tecnologia da Informação (ITI) e permite a verificação da conformidade de assinaturas digitais qualificadas e avançadas em relação às regulamentações estabelecidas pela ICP-Brasil.

3.1.6.2 Hawa CA

O Hawa é um conjunto de softwares de gerenciamento e operação de CAs e RAs; nessa família o Hawa CA é o software utilizado para operação de autoridades certificadoras, sendo capaz de emitir, atualizar e revogar certificados digitais avançados e qualificados.

3.1.7 Docker

Docker é uma plataforma de aplicações que permite virtualizar aplicações. Esta virtualização acontece com a criação de imagens docker, peças de software imutáveis que contém todas as especificações e arquivos necessários para executar uma aplicação. Com base nas imagens docker é possível então instanciar containers, que são a representação da aplicação especificada na imagem em execução. O uso de imagens e containers docker torna o processo de replicar a execução de aplicações mais simples, e foi utilizado neste trabalho para execução da base de dados, do Tesseract e da própria solução.

3.2 DESENVOLVIMENTO

Os principais requisitos definidos para o sistema foram a implementação das funcionalidades de emissão e revogação de certificados, além da obtenção de um certificado previamente emitido.

Com base nesses requisitos o sistema foi sendo construído em três etapas consecutivas, uma para cada funcionalidade. Notavelmente a primeira etapa, referente à implementação da funcionalidade de emissão, foi a mais demorada e trabalhosa. A segunda e terceira etapas foram respectivamente a de implementação da revogação e obtenção de certificados já emitidos.

Adicionalmente, o processo acabou levantando a necessidade de implementação de uma quarta funcionalidade, a geração de dossiês com os dados utilizados para as emissões. No fim das quatro etapas de desenvolvimento o sistema atingiu sua forma atual, sendo capaz de realizar as quatro funcionalidades especificadas.

As próximas subseções detalham as escolhas de projeto realizadas durante o desenvolvimento da solução.

3.2.1 Verificação da assinatura do arquivo de CNH Digital

Os dados necessários para emitir os certificados digitais devem ser retirados dos arquivos de CNH digital exportados pelo aplicativo "Carteira Digital de Trânsito"[1], que possuem assinaturas realizadas pelas unidades estaduais do Departamento Estadual de Trânsito (DETRAN) associadas ao estado de emissão do documento. Para que a solução seja confiável é necessário que as assinaturas dos documentos enviados sejam verificadas antes de efetivamente emitir um certificado.

Para validar o arquivo de CNH digital exportado o aplicativo propõe o uso do Assinador Serpro[2]. Esta alternativa foi descartada devido a complexidade do processo de instalação do software e as limitações do processo de validação que a mesma disponibiliza. Para que um documento seja validado é necessário a execução de um processo manual, tornando inviável que um software externo a utilize facilmente.

Outra alternativa, que foi a escolhida, é a proposta pelo Secretaria Nacional de Trânsito (SENATRAN): o uso do Verificador ITI[3], uma solução que pode ser usada diretamente via navegadores ou chamadas de API. Testes foram executados e foi possível verificar as assinaturas dos documentos facilmente e com sucesso utilizando o software Postman. Em seguida o processo de verificação também foi realizado programaticamente com sucesso através da aplicação, sendo apenas necessária a adição dos certificados digitais dos serviços à Truststore do Java para que as requisições realizadas fossem concluídas com sucesso.

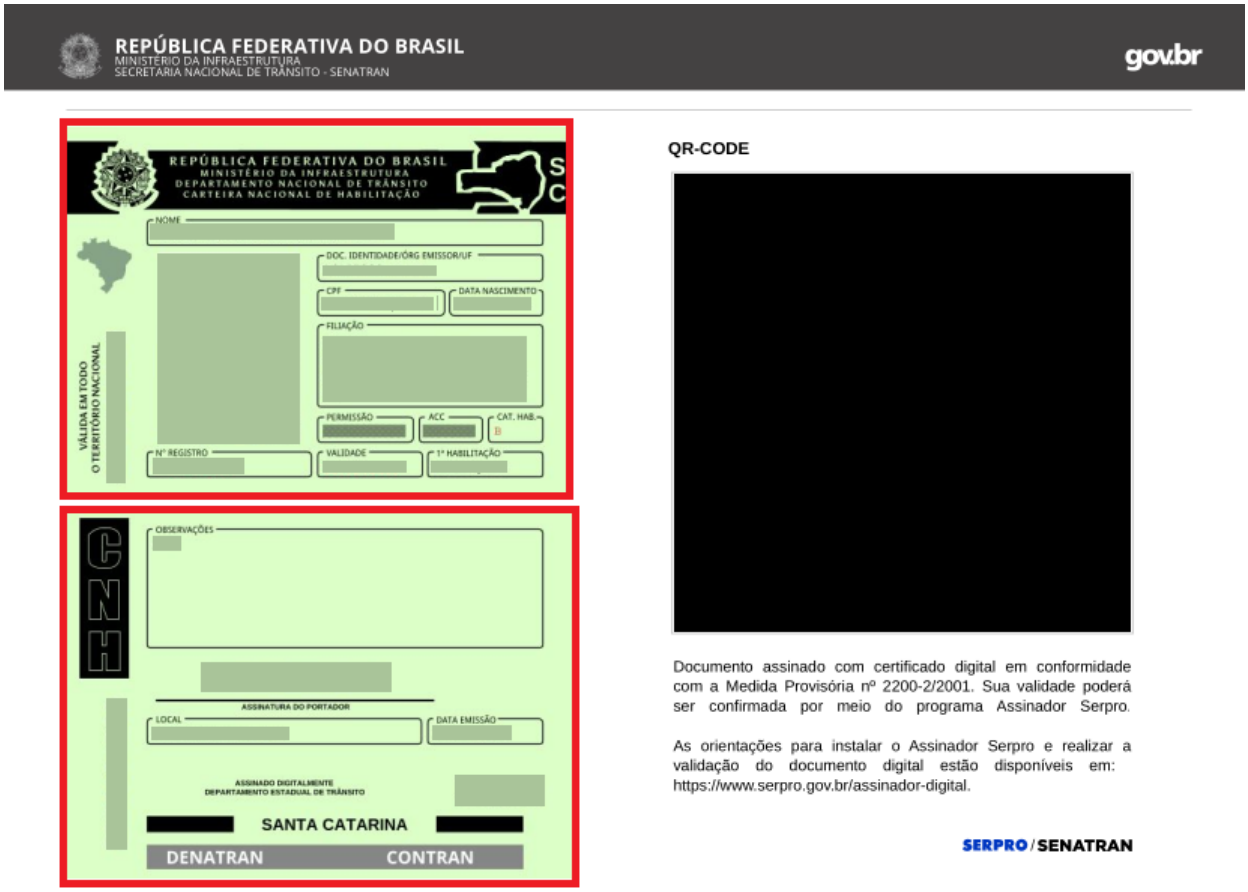
3.2.2 Formatos diferentes de arquivo de CNH

Durante o processo de desenvolvimento da lógica de extração dos dados foram identificados três modelos válidos de documento de CNH digital, visíveis nas páginas

[1] https://www.gov.br/pt-br/servicos/obter-carteira-digital-de-transito
[2] https://www.serpro.gov.br/links-fixos-superiores/assinador-digital/assinador- serpro
[3] https://verificador.iti.gov.br/verifier-2.10/

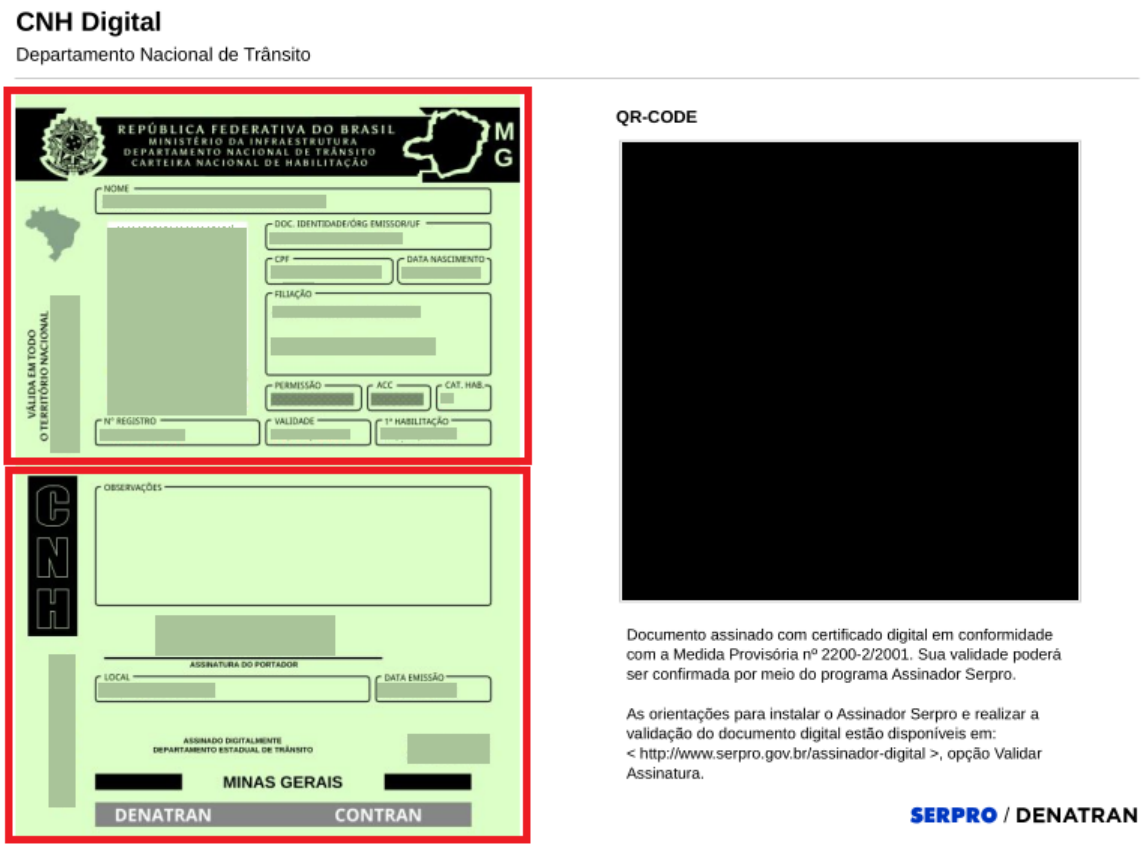
seguintes.

Figura 1 – Modelo de arquivo 1.



Fonte: Elaboração própria.

Figura 2 – Modelo de arquivo 2.



Fonte: Elaboração própria.

Figura 3 – Modelo de arquivo 3.

Fonte: Elaboração própria.

As diferenças entre os modelos da Figura 1 e Figura 2 se restringiram a formatação dos textos e quantidade de imagens; entretanto, as imagens consideradas úteis para a aplicação, marcadas em vermelho, possuem exatamente as mesmas dimensões e estrutura interna, permitindo que as coordenadas usadas para extrair os dados desses dois modelos fossem as mesmas.

Entretanto, para o modelo da Figura 3 foi necessário utilizar coordenadas diferentes para realizar a extração. No fim, apesar das diferenças de estrutura dos PDFs e formato dos documento contidos nas imagens o algoritmo para extração das informações foi o mesmo para todos os modelos, demandando apenas ajustes nos parâmetros passados ao Tesseract.

3.2.3 Extrair dados das CNHs

Após a implementação do processo de verificação dos documentos recebidos pela aplicação, a próxima etapa foi a implementação da lógica pertinente à extração dos dados do portador da CNH contida no documento.

Ao analisar o corpo dos documentos recebidos pela aplicação pode-se verificar a presença de dois tipos de componentes: textos padronizados e imagens.

Sabendo que todas as informações úteis dos documentos se encontravam nas imagens, o próximo passo foi a busca de um meio de extrair tais informações utilizando Java. Após pesquisas, a ferramenta escolhida foi o Tesseract[4], um mecanismo de OCR de código aberto que possui bibliotecas de integração com diversas linguagens. Para integração do Tesseract com o Java foi utilizada a biblioteca Tess4j[5]).

Logo após a escolha um problema facilmente encontrado foram os formatos dos documentos de entrada esperados pelo Tesseract, que durante o processo de desenvolvimento suportava apenas imagens. Testes de extração foram realizados utilizando imagens geradas a partir da conversão completa dos PDFs para imagens, e os resultados obtidos para dois dos três modelos podem ser observados no Apêndice G e Apêndice H. Para todos os modelos de CNH digital o Tesseract foi incapaz de extrair as informações contidas nas imagens dos PDFs.

Para a realizar esta tarefa foi escolhido o Apache PDFBox[6], uma biblioteca de código aberto para Java de manipulação de documentos PDF. Utilizando-a foi possível extrair com sucesso todas as imagens contidas nos documentos, que puderam então ser usados como input para o Tesseract. A Figura 4 e Figura 5 mostram imagens de CNH extraídas dos documentos utilizando o PDFBox.

Com as imagens obtidas, novas tentativas de extração de dados com o Tesseract foram realizadas, as quais não geraram resultados satisfatórios. O principal fator que levou

[4] https://github.com/tesseract-ocr
[5] https://tess4j.sourceforge.net/
[6] https://pdfbox.apache.org/

Figura 4 – Imagem de CNH no formato novo.

Fonte: Elaboração própria.

Figura 5 – Imagem de CNH no formato antigo.

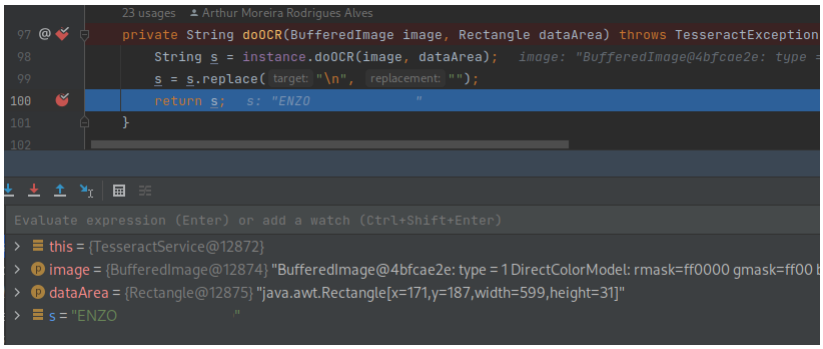
Fonte: Elaboração própria.

a esses resultados foi a maneira como as informações são apresentadas no documento; a presença de fontes diferentes, caixas de texto e informações escritas de maneira vertical acabaram sendo fatores que influenciaram negativamente a maneira como o Tesseract identificou e extraiu os dados. O Apêndice I e Apêndice J mostram os resultados obtidos utilizando as imagens extraídas diretamente com o Tesseract.

Para as tentativas seguintes a estratégia de extração foi alterada de maneira que cada uma das informações fosse obtida individualmente. Isso foi possível utilizando a biblioteca Tess4J; com ela foram fornecidas coordenadas, especificadas em pixel, que

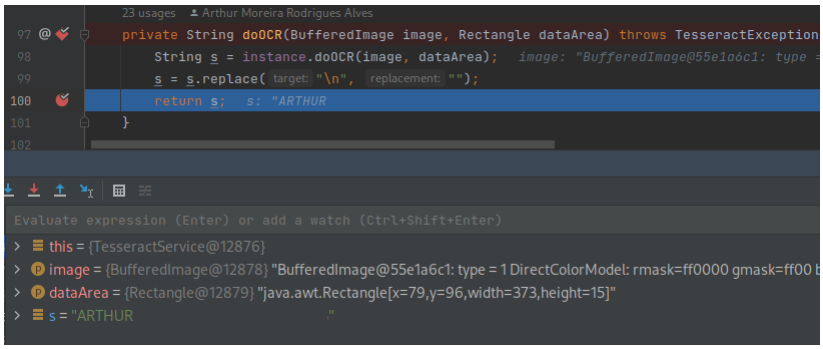
permitiram delimitar a área da imagem contendo o texto a ser extraído. Ao seguir esta estratégia foi possível extrair com sucesso as informações contidas no documento de maneira satisfatória. A Figura 6 e Figura 7 mostram a extração do nome utilizando esta abordagem.

Figura 6 – Extração de campo específico da imagem de CNH no formato novo.



Fonte: Elaboração própria.

Figura 7 – Extração de campo específico da imagem de CNH no formato antigo.



Fonte: Elaboração própria.

É possível notar que esta abordagem representou melhorias nos resultados, principalmente para os dados do modelo de CNH novo. Entretanto esta abordagem acabou acarretando em aumento no tempo de execução do Tesseract, uma vez que múltiplas rodadas de reconhecimento passaram a ser necessárias para extrair as informações.

Por fim, na etapa de pós-processamento das informações extraídas apenas foi necessário remover os caracteres *new line* adicionados pelo próprio Tesseract no processo de extração. Além disso, posteriormente foram implementados métodos para validar o Cadastro de Pessoa Física (CPF) e data de nascimento extraídos, informações utilizadas para emitir os certificados. Sendo assim, caso estas informações estejam fora do formato esperado a emissão do certificado não é realizada.

3.2.4 Chaves necessárias

Durante o processo de implementação do sistema foi necessário a criação de algum mecanismo para criação de chaves criptográficas necessárias para emissão dos certificados digitais em conjunto com o sistema Hawa CA. Para atender esta necessidade foi implementado um mecanismo simples de criação de chave do tipo RSA em Java utilizando a biblioteca bouncycastle; as chaves geradas tem a suas chaves públicas extraídas para uso no processo de emissão, enquanto as chaves privadas são cifradas e posteriormente armazenadas na base de dados da aplicação.

3.2.5 Persistir as informações dos dados usados para emitir os certificados

Após a implementação do fluxo de assinatura foi identificada a necessidade de algum mecanismo interno que permitisse averiguar quais dados foram verificados e usados na emissão de algum certificado.

A solução adotada foi a criação de um dossiê interno da emissão de um certificado. O dossiê em questão é um documento XML assinado que deve conter o *hash* de cada uma das informações usadas e obtidas no processo de emissão: a CNH Digital, o relatório do serviço Verificador ITI e a as informações extraídas pelo Tesseract.

Para a criação do dossiê primeiramente foi necessário definir entidades no sistema para mapear as informações do relatório gerado pelo Verificador ITI e as informações extraídas pelo Tesseract; após criadas essas entidades são então serializadas para o formato json, e assim como o PDF da CNH, tem seus valores de *hash* calculados e adicionados ao dossiê, que é então assinado usando uma chave previamente criada e armazenada em uma *trustore*. O dossiê gerado é então armazenado na base de dados MySQL junto com os documentos. Este fluxo pode ser observado na Figura 8

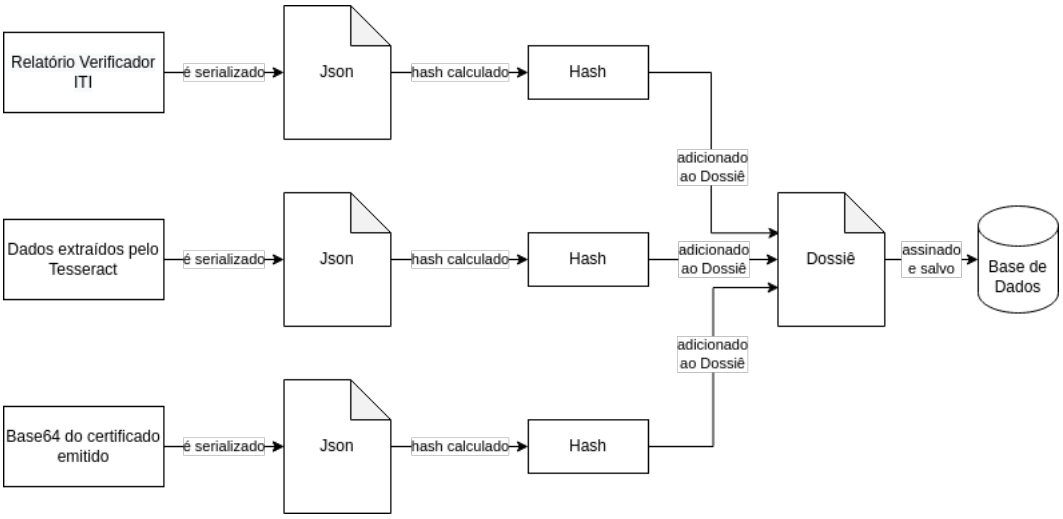
Por fim, para permitir que essas informações sejam consultadas foram criados endpoints que permitem a obtenção das mesmas através do serial number do certificado associado.

3.2.6 Ajustes nos parâmetros do Tesseract

Uma vez que o sistema teve todas as suas funcionalidades básicas implementadas buscou-se melhorar a performance da aplicação. Para tal foram realizados testes com diferentes valores para o parâmetro *Page Segmentation Mode*, que permite instruir o *Tesseract* a interpretar as imagens passadas como parâmetro de maneiras específicas. Os modos de operação testados e suas descrições são apresentadas no Quadro 1.

Foram realizadas rodadas de teste para cada um dos modos apresentados no Figura 1 com o intuito de medir o tempo médio de execução para cada um dos endpoints principais da aplicação: */issue*, */revoke*, */get-cert*. As rodadas de teste foram compostas por 100

Figura 8 – Fluxo de criação do Dossiê.



Fonte: Elaboração própria.

Quadro 1 – Valores para Page Segmentation Mode.

Page Segmentation Mode	**Descrição**
1	Modo padrão do Tesseract; neste modo o Tesseract assume que a imagem recebida como entrada representa uma página de texto.
4	Neste modo de operação o Tesseract assume que a imagem de entrada possui textos organizados em colunas com tamanhos variáveis.
6	Neste modo de operação o Tesseract assume que a imagem contém um bloco de texto uniforme.
7	Neste modo de operação o Tesseract assume que a imagem contém apenas uma linha de texto.
13	Neste modo de operação o Tesseract assume que a imagem contém apenas uma linha de texto, porém ignora algumas otimizações internas.

Fonte: Elaboração própria.

requisições realizadas consecutivamente para cada um dos endpoints. As informações obtidas para cada um dos testes são apresentadas no Quadro 2, Quadro 3 e Quadro 4.

Quadro 2 – Resultados dos testes para o *endpoint /issue*

Page Segmentation Mode	**Tempo médio de execução(em milissegundos)**
1	2597
4	2632
6	2566
7	2629
13	2810

Fonte: Elaboração própria.

Em conjunto destes testes também foi checada a taxa de consistência dos resultados obtidos pelo Tesseract ao extrair os dados contidos nas CNHs. Isso foi comparando com

Quadro 3 – Resultados dos testes para o *endpoint /revoke*

Page Segmentation Mode	**Tempo médio de execução(em milissegundos)**
1	293
4	400
6	695
7	513
13	666

Fonte: Elaboração própria.

Quadro 4 – Resultados dos testes para o *endpoint /get-cert*

Page Segmentation Mode	**Tempo médio de execução(em milissegundos)**
1	14
4	13
6	14
7	14
13	13

Fonte: Elaboração própria.

base nos resultados retornados pelo *endpoint /get-extracted=cnh-info*, responsável por retornar os dados extraídos de uma CNH digital associada a um certificado emitido anteriormente; os dados retornados por este *enpoint* são retornados em formato JSON e sua estrutura pode ser checada no APÊNDICE C.

Para calcular a consistência inicialmente foram emitidos certificados com os arquivos de teste; após emitidos os dados extraídos destes arquivos de teste foram obtidos utilizando o *endpoint /get-extracted-cnh-info* em formato JSON. Com estes dados armazenados as rodadas de testes foram realizadas, e para cada certificado emitido com sucesso utilizando um arquivo de teste os dados extraídos pela aplicação foram comparados com o valor de referência obtido anteriormente para este arquivo; casos de igualdade foram considerados sucesso, enquanto o contrário foram considerados fracassos. Para todas as opções de *Page Segmentation Mode* o sucesso da aplicação foi de 100

Com base nestas duas informações optou-se por definir a opção 6 como padrão para o *Page Segmentation Mode*, dado que este foi o modo que forneceu melhor performance para a operação de emissão, que engloba o processo de extração dos dados.

4 RESULTADOS

Nesta seção serão apresentadas as funcionalidades desenvolvidas para a API, resultados de uso e as limitações identificadas.

4.1 VISÃO GERAL DO SISTEMA

A API obtida com o desenvolvimento é capaz de emitir e revogar certificados, além de retornar certificados emitidos anteriormente e os documentos utilizados e gerados nesse processo: o dossiê assinado, relatório do serviço Verificador ITI e dados extraídos pelo Tesseract. Todas essas funcionalidades são acessadas por meio de chamadas de API.

Cada uma dessas funcionalidades demanda o uso de recursos específicos do sistema, provenientes de componentes de software chamados *services*, responsáveis por isolar a lógica referente ao funcionamento de partes específicas do sistema. Em seu estado atual a aplicação conta com oito *services*, apresentados e descritos no Quadro 5.

Quadro 5 – Services implementados.

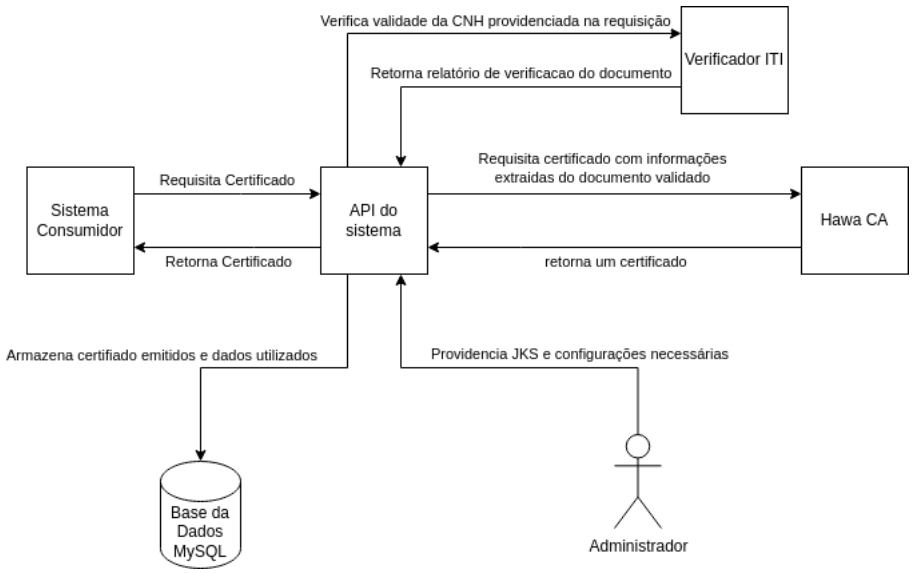
CNHService	Service que integra todos os demais, acessando suas funcionalidades e realizando a persistência de novas entidades de sistema criadas, como certificados e dossiês.
EncodeAndDecodeService	Provê a lógica necessária para armazenar e carregar documentos, codificados em Base64, da base de dados do sistema.
DossierSignerService	Service que gera e assina dossiês para cada certificado emitido no sistema. Este service utiliza uma chave RSA armazenada em uma JKS para realizar a assinatura do dossiê gerado.
HawaCaService	Service responsável por realizar a comunicação com o Hawa CA para emissão e revogação de certificados.
KeyService	Service que realiza a criação de chaves RSA de 2048 para emissão dos certificados. As chaves privadas geradas são cifradas utilizando , e posteriormente armazenadas na base de dados.
PDFBoxService	Service responsável por utilizar a biblioteca PDFBox para extrair as imagens contidas nos documentos PDF recebidos.
TesseractService	Service responsável por utilizar o Tesseract para extrair as informações contidas nas imagens obtidas pelo service PDFBoxService.
VerifierService	Service responsável por utilizar a API Verificador ITI para verificar a validade da assinatura dos documentos PDF recebidos. Este service verifica a conformidade das assinaturas e validade dos certificados da cadeia de certificação.

Fonte: Elaboração própria.

Além dos *services* a aplicação utiliza quatro componentes externos: o Verificador ITI, o Hawa CA, uma JKS e a base de dados MySQL. Para a interação com esses componentes a aplicação deve ser configurada através de variáveis de ambiente.

Por fim, a Figura 9 fornece uma visão geral do sistema, e como ocorre a interação entre os diferentes componentes.

Figura 9 – Visão geral do sistema.



Fonte: Elaboração própria.

4.2 FUNCIONALIDADES IMPLEMENTADAS

As funcionalidades da API estão disponíveis a partir do endereço base /cnh. Cada uma das funcionalidades será descrita nas subseções a seguir. As estruturas dos objetos retornados pelos *endpoints* estão descritas nos apêndices e anexos do documento.

4.2.1 Emissão

Permite a emissão de certificados digitais. O *endpoint* de emissão pode ser acessado em */issue* e aceita apenas requisições do tipo POST. A estrutura do corpo das requisições para esse *endpoint* é descrita no Quadro 6, enquanto os possíveis valores de resposta são apresentados no Quadro 7.

Quadro 6 – Estrutura do corpo da requisição do *endpoint /issue*.

Tipo do corpo da requisição	**Valores do corpo da requisição**	**Descrição**
multipart/form-data	file	Arquivo de CNH digital.
	key-password	Senha para cifragem da chave privada gerada.

Fonte: Elaboração própria.

Quadro 7 – Respostas do *endpoint /issue*.

Status da resposta	**Corpo da resposta**	**Estrutura do corpo da resposta**	**Descrição**
200	*appication/json*	CNHServiceIssueResponse	Json contendo informações do certificado emitido.
400		SimpleMessageResponse	Indica a presença de erros relacionados a ausência ou validade do arquivo de PDF enviado.
500			Indica a ocorrência de erros relacionados aos componentes internos da API, como o serviço de criação de chaves, acesso à base de dados ou utilização do Tesseract.

Fonte: Elaboração própria.

4.2.2 Revogação

Permite revogar um certificado digital avançado emitido anteriormente. Este *endpoint* pode ser acessado em */revoke* e aceita apenas requisições do tipo POST. Os parâmetros da requisição são definidos no Quadro 8, e as possíveis repostas no Quadro 9.

Quadro 8 – Parâmetros de requisição do *endpoint /revoke*.

Nome	**Descrição**
serial-number	Serial number de um certificado emitido.

Fonte: Elaboração própria.

Quadro 9 – Respostas do *endpoint /revoke*.

Status da resposta	**Corpo da resposta**	**Estrutura do corpo da resposta**	**Descrição**
200	*appication/json*	CNHServiceRevokeResponse	Json contendo informações referentes ao certificado revogado.
400		SimpleMessageResponse	Indica que não foi possível encontrar um certificado com o serial number informado.
500			Indica a ocorrência de erros relacionados aos componentes internos da API relacionados ao acesso a base de dados.

Fonte: Elaboração própria.

4.2.3 Obter certificados emitidos e documentos associados

Estas funcionalidades permitem que um certificado emitido previamente, assim como o dossiê e informações associadas ao processo sejam obtidas. Os parâmetros de requisição para estes endpoints já foram apresentados na Quadro 8. As respostas para cada um destes *enpoints* estão definidas no Quadro 10. Estas funcionalidades podem ser acessadas nos endpoints */get-cert*, */get-cnh*, */get-verifier-response*, */get-extracted-cnh-info* e */get-dossier*.

Quadro 10 – Respostas para o restante dos endpoints.

Status da resposta	**Corpo da resposta**	**Estrutura do corpo da resposta**	**Descrição**
200	*application/pdf*	CNH Digital	CNH digital utilizada para emissão do certificado associado ao serial number informado na requisição. Esta resposta é gerada apenas pelo *endpoint /get-cnh*.
	application/xml	Dossiê assinado	Dossiê gerado e assinado após a emissão do certificado associado ao serial number informado na requisição. Esta resposta é gerada apenas pelo *endpoint /get-dossier*.
	application/json	CNHServiceIssueResponse	Json contendo informações do certificado associado ao serial number informado na requisição. Esta resposta é gerada apenas pelo *endpoint /get-cert*.
		Extracted CNH Info	Json contendo as informações extraídas de um arquivo de CNH digital no momento de emissão do certificado associado ao serial number informado na requisição. Esta resposta é gerada apenas pelo *endpoint /get-extracted-cnh-info*.
		Relatório do verificador de documentos	Json contendo o relatório retornado pelo Verificado ITI referente a verificação da CNH digital utilizada para emitir o certificado associado ao serial number informado. Esta resposta é gerada apenas pelo *endpoint /get-verifier-response*.
400		SimpleMessageResponse	Indica a ausência de recursos associados ao serial number informado na requisição. Esta resposta é gerada por todos os endpoints.
500			Indica a ocorrência de erros relacionados aos componentes internos da API como o acesso a base de dados. Esta resposta é gerada por todos os endpoints.

Fonte: Elaboração própria.

4.3 CONFIGURAÇÃO DO SISTEMA

O sistema desenvolvido utiliza diversos componentes, internos e externos, que necessitam de configurações que serão apresentadas nessa seção.

4.3.1 Variáveis de ambiente

Para execução do sistema é necessário que diversas variáveis de ambiente sejam configuradas. O primeiro conjunto de variáveis, apresentado n Quadro 11, diz respeito a informações que serão adicionados aos certificados emitidos, tamanho das chaves geradas e os endereços dos serviços Verificador ITI, Hawa CA e base dados utilizados.

Quadro 11 – Variáveis de ambiente gerais.

Variável de ambiente	**Valores padrão**	**Descrição**
SERVER_PORT	8080	Porta na qual a API ficará exposta.
CITY	Florianópolis	Cidade de emissão do certificado.
STATE	SC	Estado de emissão do certificado.
RSA_KEY_SIZE	2048	Tamanho das chaves RSA geradas no processo de emissão de certificados
VERIFIER_ADDRESS	https://pbad.labsec.ufsc.br/verifier-hom/report	Endereço do serviço Verificador ITI
HAWA_ADDRESS	Não há	Endereço do Hawa CA utilizado para emitir os certificados.
DB_ADDRESS		O endereço da base de dados que será utilizada.
DB_PORT		A porta da base de dados.
DB_SCHEMA_NAME		O nome da base de dados.
DB_USERNAME		Usuário da base de dados.
DB_PASSWORD		Senha do usuário da base de dados.

Fonte: Elaboração própria.

O segundo conjunto de variáveis de ambiente possui as configurações referentes ao JKS utilizado pelo sistema para assinar os dossiês de emissão de certificado. Os valores padrão para essas variáveis são apresentados no Quadro 12.

Quadro 12 – Variáveis de ambiente relacionadas a JKS.

Variável de ambiente	**Valores padrão**	**Descrição**
KEY_STORE_PATH	key-store.jks	Endereço da JKS que contém as chaves utilizadas para assinar os dossiês dos certificados gerados.
KEY_ENTRY	dossier-key	Nome da entrada na JKS que contém a chave que será utilizada para assinar os certificados.
KEY_ENTRY_PASSWORD	dossier-key-1234	Senha da key entry.
KEY_STORE_PASSWORD	key-store-1234	Senha da JKS.

Fonte: Elaboração própria.

O último conjunto de variáveis de ambiente está relacionado ao diretório contendo os dados de treinamento utilizado pelo Tesseract para a realização do OCR, além dos parâmetros de configuração referentes à engine e modo de segmentação utilizados. Estas informações podem ser observadas no Quadro 13.

Quadro 13 – Variáveis de ambiente relacionadas ao Tesseract.

Variável de ambiente	**Valores padrão**	**Descrição**
TESS_DATAPATH	/usr/share/tesseract/tessdata	Caminho para o diretório com as informações de treinamento utilizadas pelo tesseract para realização do OCR.
TESS_OCR_ENGINE_MODE	1	Engine utilizada pelo Tesseract para realizar o OCR.
TESS_PAGE_SEG_MODE	6	Meio pelo qual o Tesseract busca interpretar as informações contidas em uma imagem.

Fonte: Elaboração própria.

4.3.2 Execução

Devido ao grande número de requisitos para inicialização do sistema a abordagem de execução escolhida foi através do uso de imagens docker. O uso de imagens docker permite simplificar etapas referentes a configuração do Tesseract, além de permitir uma plataforma homogênea para a execução da aplicação em qualquer computador.

O primeiro passo para a execução da aplicação é gerar o arquivo .jar da aplicação. Para esta etapa deve-se executar o comando *"mvn install"* no diretório raiz do projeto; após o fim da execução do comando o .jar da aplicação será gerado.

Em seguida, ainda no diretório raiz do projeto, deve-se executar o comando *"docker build -t <NOME_IMAGEM> ."*. Durante o processo de construção da imagem diversas etapas de configuração necessárias para a execução da aplicação serão executadas, como a adição dos certificados dos serviços Verificador ITI suportados para verificação das CNHs digitais e instalação do Tesseract e seus pacotes de suporte ao idioma português.

Após o fim da execução do comando anterior já é possível executar a aplicação. Recomenda-se que a execução seja realizado com o comando *"docker-compaso -f <NOME_ARQUIVO_COMPOSE>.yml up"*; este comando demanda o uso de um arquivo *docker-compose* que pode ser elaborado conforme o modelo disponibilizado no diretório raiz do projeto.

4.4 LIMITAÇÕES

Apesar de ser capaz de executar suas funcionalidades mínimas, o sistema atualmente possui limitações que precisam ser mitigadas para permitir o seu uso em ambientes de produção.

A primeira dessas limitações diz respeito à criação e persistência dos pares de chaves utilizados para emissão de certificados; atualmente estes processos são realizados pelo próprio sistema, e podem ser externalizados através do uso de componentes como Hardware Secure Module (HSM)s que são capazes de realizar tais atribuições de maneira mais eficiente e segura.

Outra funcionalidade que também pode se beneficiar do uso de HSMs é a assinatura de dossiês. No estado atual a assinatura é realizada com uma chave armazenada em uma keystore que deve ser criada e providenciada pelo administrador do sistema; o uso de HSMs para armazenar e assinar os dossiês pode auxiliar tanto na diminuição do tempo de emissão quanto na segurança do processo.

Ainda sobre a funcionalidade de assinatura de dossiês, deve-se ressaltar que os XMLs gerados não estão de acordo com nenhum tipo de política de assinatura definido para esse tipo de documento atualmente no Brasil; em um cenário de uso real da API a estrutura do documento gerado deverá ser analisada e ajustada para atender aos requisitos estabelecidos pelos órgão regulamentadores responsáveis.

Outra limitação do software é o formato dos arquivos de CNH digital aceitos. Como foram utilizadas poucas CNHs durante o processo de desenvolvimento não é possível garantir que a aplicação será capaz de processar todos os modelos eventualmente enviados. Essa limitação pode ser corrigida posteriormente adaptando a lógica já existente para englobar novos modelos de CNH.

Como atualmente a aplicação apenas valida o CPF e data de nascimento extraídos internamente a implementação do uso de fontes externas para verificar os dados extraídos com o Tesseract é uma alternativa que deve ser considerada a fim de garantir mais confiabilidade aos dados extraídos pela aplicação

Por fim, um detalhe identificado foi a falta de uso para algumas informações extraídas pela aplicação ao realizar a emissão de um certificado digital utilizando o Hawa CA, algo que pode ser alterado no futuro com a implementação de políticas de certificação que incluam essas informações nos certificados emitidos.

5 CONCLUSÕES

Inicialmente este trabalho apresentou conceitos pertinentes e necessários relacionados à área de assinatura digital. Com esses conceitos apresentados foi desenvolvida uma aplicação capaz de verificar e extrair os dados contidos em uma CNH digital com sucesso e utilizá-los para emitir e revogar certificados digitais com as funcionalidades do sistema Hawa CA; além disto, a aplicação também é capaz de retornar certificados emitidos previamente.

Também foram apresentados os motivos que influenciaram na escolha de ferramentas utilizadas no desenvolvimento da aplicação, como o Tesseract e PDFBox, e em seguida apresentados os resultados obtidos. O código fonte da aplicação, além de informações sobre como executá-la, podem ser acessados nos repositórios listados nos apêndices.

Por fim, pode-se considerar que os objetivos propostos neste trabalho foram alcançados.

5.1 TRABALHOS FUTUROS

Sistemas que decidam utilizar as funcionalidades desenvolvidas para esta API devem levar em consideração as limitações apresentadas na Seção 4.4. A implementação de outras alternativas para o gerenciamento das chaves criptográficas e validação dos dados extraídos pelo Tesseract devem ser considerados objetivos prioritários.

Além disso, a definição de políticas de certificação que utilizem todos os dados extraídos é algo que também deve ser considerado, assim como o desenvolvimento de ferramentas para o gerenciamento dos usuários e certificados armazenados na base de dados no sistema.

Finalmente, as funcionalidades atuais do sistema também podem ser utilizadas como base para implementar fluxos de emissão que utilizem outros documentos digitais validados.

REFERÊNCIAS

AFSHAR, Reshma. Digital Certificates, 2015.

AGRAWAL, Monika; MISHRA, Pradeep. A Comparative Survey on Symmetric Key Encryption Techniques. **International Journal on Computer Science and Engineerin**, v. 4, n. 5, p. 877–882, 2012.

APACHE SOFTWARE FOUNDATION. **Apache PDFBox® - A Java PDF Library**. [*S.l.*]. Disponível em: https://web.archive.org/web/20221223034208/https://pdfbox.apache.org/index.html. Acesso em: 23 dez. 2022.

COOPER, D *et al.* **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**. [*S.l.*], 2022. Disponível em: https://web.archive.org/web/20221219133757/https://www.rfc-editor.org/rfc/rfc5280. Acesso em: 23 dez. 2022.

DELFS, Hans; KNEBL, Helmut. **Introduction to Cryptography**. 1. ed. Nurnberg, Germany: Springer-Verlag Berlin Heidelberg, 2002. Disponível em: https://books.google.com.br/books?id=DOWpCAAAQBAJ&pg=PR8&hl=pt-BR&source=gbs_selected_pages&cad=2#v=onepage&q&f=false.

INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **RESOLUÇÃO CG ICP-BRASIL Nº 182, DE 18 DE FEVEREIRO DE 2021**. [*S.l.*], 2021. Disponível em: https://web.archive.org/web/20221012072545/https://www.gov.br/iti/pt-br/assuntos/legislacao/resolucoes/resolucoes-old/Resoluo_CGICPBrasil_182Dec10139Etapa3DOC15_assinada.pdf. Acesso em: 23 dez. 2022.

INSTITUTO NACIONAL DE TECNOLOGIA E INFORMAÇÃO. **Verificador de Conformidade**. [*S.l.*]. Disponível em: https://web.archive.org/web/20221223034141/https://verificador.iti.gov.br/verifier-2.10/. Acesso em: 23 dez. 2022.

ITU. **Recommendation ITU-T X.509**. [*S.l.*], 2019. Disponível em: https://web.archive.org/web/20221223033458/https://www.itu.int/rec/T-REC-X.509-201910-I/en. Acesso em: 23 dez. 2022.

KAUR, Ravneet; KAUR, Amandeep. Digital Signature. **International Conference on Computing Sciences**, n. 4, p. 295–301, 2012.

KOHNFELDER, Loren M. Towards a Practical Public-key Cryptosystem. **Massachusetts Institute of Technology**, 1978.

MAQSOOD, Faiqa *et al.* Cryptography: A Comparative Analysis for Modern Techniques. **International Journal of Advanced Computer Science and Applications**, v. 8, n. 6, p. 442–448, 2017.

MINISTÉRIO DA ECONOMIA. **Assinatura Eletrônica do GOV.BR**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223033733/https://www.gov.br/governodigital/pt-br/assinatura-eletronica`. Acesso em: 23 dez. 2022.

_____. **Obter a Carteira Digital de Trânsito (CDT)**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223033840/https://www.gov.br/pt-br/servicos/obter-carteira-digital-de-transito`. Acesso em: 23 dez. 2022.

PRESIDÊNCIA DA REPÚBLICA. **DECRETO Nº 10.543, DE 13 DE NOVEMBRO DE 2020**. [*S.l.*], 2020. Disponível em: `https://web.archive.org/web/20221223034226/http://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2020/Decreto/D10543.htm`. Acesso em: 23 dez. 2022.

_____. **MEDIDA PROVISÓRIA No 2.200-2, DE 24 DE AGOSTO DE 2001**. [*S.l.*], 2001. Disponível em: `https://web.archive.org/web/20221223033559/http://www.planalto.gov.br/ccivil_03/MPV/Antigas_2001/2200-2.htm`. Acesso em: 23 dez. 2022.

QUAN NGUYEN. **Tess4J**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223034311/https://tess4j.sourceforge.net/`. Acesso em: 23 dez. 2022.

SANTESSON, S; NYSTROM, M; POLK, T. **Internet X.509 Public Key Infrastructure: Qualified Certificates Profile**. [*S.l.*], 2004. Disponível em: `https://web.archive.org/web/20221223033112/https://www.rfc-editor.org/rfc/rfc3739`. Acesso em: 23 dez. 2022.

SCHUKAT, Michael; CORTIJO, Pablo. Public Key Infrastructures and Digital Certificates for the Internet of Things. **Irish Signals and Systems Conference (ISSC)**, v. 26, 2015.

SERPRO. **Assinador Serpro**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223034130/https://www.serpro.gov.br/links-fixos-superiores/assinador-digital/assinador-serpro`. Acesso em: 23 dez. 2022.

SURYA, E.; C.DIVIYA. A Survey on Symmetric Key Encryption Algorithms. **International Journal of Advanced Computer Science and Applications**, v. 2, n. 4, p. 475–477, 2011.

TESSERACT OCR. **Tesseract GitHub Repository**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223034041/https://github.com/tesseract-ocr`. Acesso em: 23 dez. 2022.

_____. **Tesseract User Manual**. [*S.l.*]. Disponível em: `https://web.archive.org/web/20221223033941/https://tesseract-ocr.github.io/tessdoc/`. Acesso em: 23 dez. 2022.

THE EUROPEAN PARLIAMENT e THE COUNCIL OF THE EUROPEAN UNION. **REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC**. [*S.l.*], 2014. Disponível em: `https://web.archive.org/web/20221223033638/https://ec.europa.eu/futurium/en/system/files/ged/eidas_regulation.pdf`. Acesso em: 23 dez. 2022.

WEISE, Joel. Public Key Infrastructure Overview. **Sun BluePrints™ OnLine**, v. 2, 2001.

YASSEIN, Muneer Bani *et al.* **Introduction to Cryptography**. [*S.l.*]: IEEE, 2017. Disponível em: `https://web.archive.org/web/20221223032806/https://ieeexplore.ieee.org/document/8308215`. Acesso em: 23 dez. 2022.

—

APÊNDICE A – ESTRUTURA DO DOSSIÊ ASSINADO

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DigestDocumentos>
    <CNH>
        Base64 do hash documento de CNH digital
    </CNH>
    <InformacoesExtraidasCNH>
        Base64 do hash das informações extraídas
    </InformacoesExtraidasCNH>
    <CertificadoEmitido>
        Base64 do certificado emitido
    </CertificadoEmitido>
    <VerificadorDeDocumentos>
        <URL>
            endereço do Verificador ITI utilizado
        </URL>
        <Report>
            Base64 do hash do relatório gerado pelo verificador
        </Report>
    </VerificadorDeDocumentos>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        Informações da assinatura realizada
    </Signature>
</DigestDocumentos>
```

APÊNDICE B – ESTRUTURA DE CNHSERVICEISSUERESPONSE

```
{
    "certB64": "Base64 do certificado emitido",
    "certSerialNumber": "Número serial do certificado emitido"
}
```

APÊNDICE C – ESTRUTURA DE EXTRACTEDCNHINFO

```
{
    "name": "Nome do dono do documento",
    "docInfo": "Informacoes do documento de identidade",
    "cpf": "Número do cpf",
    "birthDate": "dd/mm/yyyy",
    "birthData": "dd/mm/yyyy, cidade natal, estado natal",
    "fatherName": "Nome do pai",
    "motherName": "Nome da mãe",
    "cnh": "Número da cnh",
    "validity": "dd/mm/yyyy",
    "firstCNHDate": "dd/mm/yyyy",
    "issuePlace": "Local de emissão do documento",
    "issueDate": "dd/mm/yyyy",
    "nationality": "Nacionalidade"
}
```

APÊNDICE D – ESTRUTURA DE SIMPLEMESSAGERESPONSE

```
{
    "message": "Mensagem informando algum tipo de erro"
}
```

APÊNDICE E – ESTRUTURA DE CNHSERVICEREVOKERESPONSE

```
{
    "revoked": true OU false,
    "serialNumber": "Número serial do certificado revogado",
    "certB64": "Base64 do certificado revogado",
    "revocationDate": "Data de revogação do certificado"
}
```

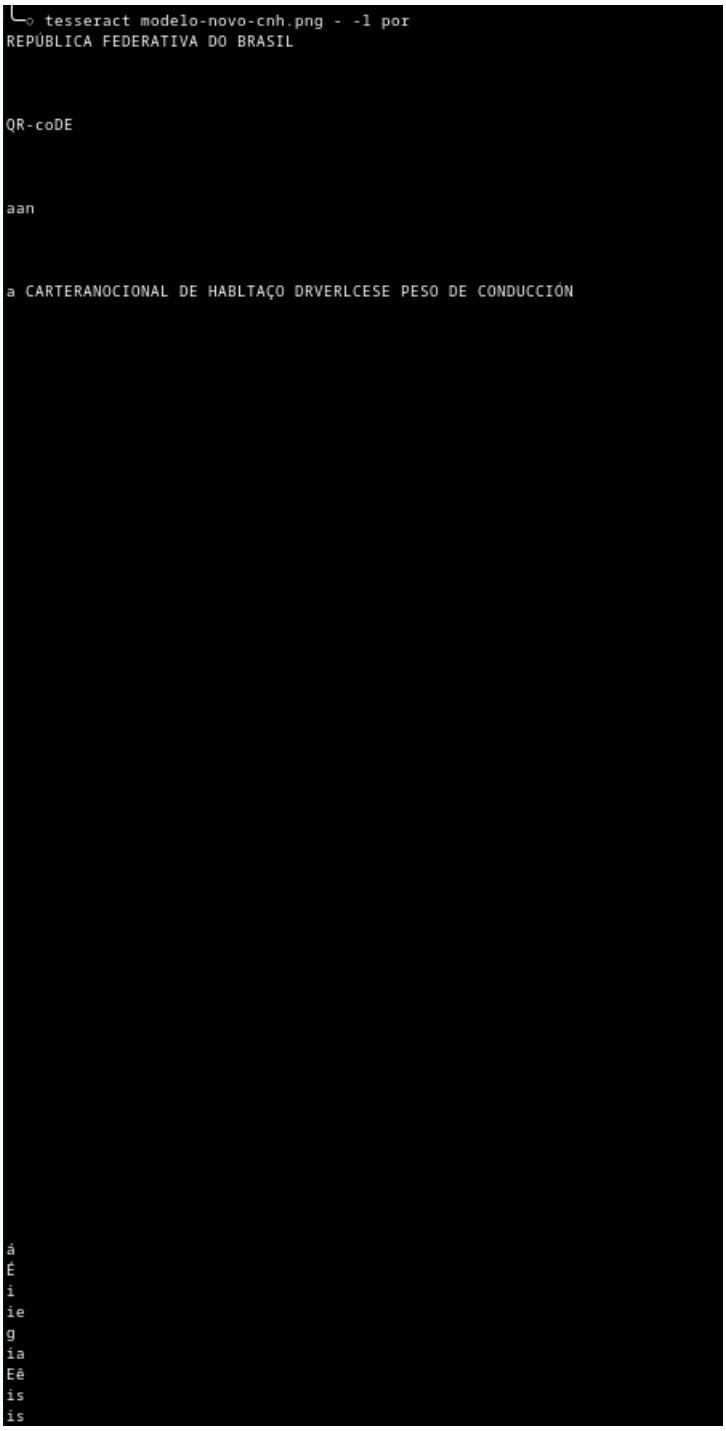
APÊNDICE F – CÓDIGO FONTE

O código fonte da aplicação desenvolvida pode ser acessado nos repositórios listados a seguir:

- `https://github.com/artmra/advanced-certificate`
- `https://web.archive.org/web/20221223032429/https://github.com/artmra/advanced-certificate/tree/1.0.0`

APÊNDICE G – USO DIRETO DO TESSERACT COM ARQUIVO DE CNH NO FORMATO NOVO

Figura 10 – Uso direto da figura 3 com Tesseract.



Fonte: Elaboração própria.

APÊNDICE H – USO DIRETO DO TESSERACT COM ARQUIVO DE CNH NO FORMATO ANTIGO

Figura 11 – Uso direto da figura 1 com Tesseract.

Fonte: Elaboração própria.

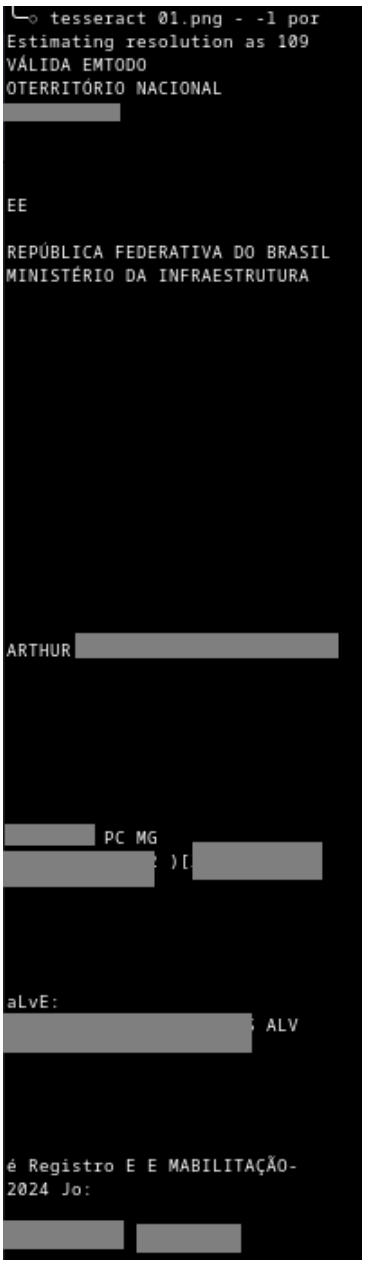
APÊNDICE I – USO DIRETO DO TESSERACT COM IMAGEM DE CNH NO FORMATO NOVO

Figura 12 – Tentativa extração de texto diretamente da imagem de CNH no formato novo.

Fonte: Elaboração própria.

APÊNDICE J – USO DIRETO DO TESSERACT COM IMAGEM DE CNH NO FORMATO ANTIGO

Figura 13 – Tentativa extração de texto diretamente da imagem de CNH no formato antigo.



Fonte: Elaboração própria.

APÊNDICE K – ARTIGO NO FORMATO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

Serviço de Emissão de Certificados Digitais com Base em Dados de CNHs Digitais

Arthur Moreira Rodrigues Alves

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
CEP: 88040-370 – Florianópolis – SC – Brasil

artmr.alves@gmail.com

Abstract. *Digital certificates, also called certificates, are digital documents that link an entity to a key pair. Entities of ICP-Brasil, Brazil's main public key infrastructure, issue certificates that are highly reliable. However, issuing this kind of certificate is a paid service. As an alternative the government provides free certificates with reduced applicability. To use this service a person needs to authenticate itself in the platforms trusted by the Gov.br, the issuer of these certificates. However, this process can be tricky and become an impeding factor to use the service. As a simpler alternative, this work describes the implementation of a service that issues certificates using documents validated by trusted entities.*

Resumo. *Certificados digitais, também chamados de certificados, são documentos digitais que ligam uma entidade a um par de chaves. Entidades da Infraestrutura de Chaves Públicas Brasileira emitem certificados altamente confiáveis. Entretanto, este tipo de certificado é um serviço pago. Como alternativa o governo provê certificados gratuitos com aplicabilidade reduzida. Para obtê-los é preciso se autenticar em plataformas consideradas confiáveis pelo Gov.br, emissor destes certificados. Entretanto, este processo pode ser complicado e tornar-se um impeditivo para o uso do serviço. Como uma alternativa mais simples este trabalho descreve a implementação de um serviço que emite certificados utilizando documentos validados por entidades confiáveis.*

1. Introdução

Assinaturas eletrônicas são meios que permitem criar uma relação de autoria entre uma entidade e um conjunto de dados eletrônicos. O simples ato de renomear um arquivo ou adicionar uma assinatura digitalizada pode categorizar uma assinatura eletrônica.

Do ponto de vista prático se espera que as assinaturas eletrônicas sejam o equivalente a assinaturas físicas, o que auxilia a implementação de serviços digitais baseados em características como autenticação e integridade. Entretanto, nem todos os mecanismos de assinatura eletrônica são capazes de garantir tais características.

As assinaturas digitais são um meio de implementação de assinaturas eletrônicas que podem garantir as características mencionadas anteriormente de maneira relativamente simples e eficiente. O seu uso requer a emissão de certificados digitais, documentos eletrônicos que relacionam uma chave pública, elemento necessário para execução desse tipo de assinatura, a uma entidade.

Os certificados digitais garantem a todos que decidem confiar neles que determinada chave pública identifica diretamente uma entidade no meio digital. Essa afirmação garante que qualquer transação, que pode ser entendida como algum tipo de comunicação, realizada entre partes que possuam certificados digitais as propriedades de confidencialidade, integridade, autenticidade e não repúdio.

Geralmente uma Autoridade Certificadora (AC) é a entidade responsável por emitir um certificado digital. Para garantir a confiabilidade de um certificado as ACs requisitam e validam um conjunto de informações das entidades que desejam obtê-los. Apenas as entidades que atendem aos critérios de validação das ACs têm seus certificados emitidos.

As ACs geralmente integram algo chamado infraestrutura de chaves públicas (ICP), uma estrutura de entidades organizadas de maneira hierárquica que realizam ações relacionadas a emissão e manutenção de certificados digitais seguindo determinados padrões de operação.

No Brasil a principal ICP em operação é a ICP-Brasil. As assinaturas digitais realizadas com os certificados emitidos por ela possuem validade e aplicabilidade, inclusive no âmbito jurídico, idênticas a de uma assinatura física.

No fim das contas, os gastos associados às diversas tarefas e infraestrutura que uma entidade integrante de um ICP deve executar e manter, especialmente aquelas que integram a ICP-Brasil, são cobertos com taxas cobradas para emissão de certificados. Entretanto, essas taxas associadas à emissão de determinados certificados, principalmente os que possuem vasta aplicabilidade, acabam sendo um impeditivo para o uso deste tipo de assinatura.

Para contornar este problema o governo disponibiliza serviços como o Gov.br[1], que permite realizar a emissão gratuita de certificados digitais. Em contrapartida os certificados emitidos por esse serviço acabam tendo uma aplicabilidade limitada; além disso, para emiti-los também são exigidas diversas etapas de verificação de identidade através de informações provenientes de outras instituições, como bancos e órgãos públicos.

Entretanto, esse processo de verificação de identidade através de outros serviços é algo que também pode acabar se tornando um fator impeditivo para o uso deste serviço. Isto ocorre devido a própria premissa do processo de verificação, que necessita que o usuário utilize os serviços suportados. Além disso, o próprio processo de autorização de uso dos dados armazenados nas bases de dados destes serviços pode ser complicado e extensivo.

Com base na complexidade envolvida no processo de verificação de identidade para emissão de certificados avançados em serviços como o Gov.br, este trabalho desenvolve um sistema em formato de Application Programming Interface (API) capaz de realizar a emissão de certificados digitais avançados. Para validar a identidade dos usuários, o sistema utiliza a versão digital de Carteiras Nacionais de Trânsito (CNH), documentos gerados e assinados digitalmente pelos departamentos estaduais de trânsito.

[1] https://www.gov.br/pt-br

1.1. Objetivos

Este trabalho tem como objetivo o desenvolvimento de um serviço que realize a emissão de certificados digitais avançados com o sistema Hawa CA usando os dados contidos em CNHs digitais. A solução poderá então ser utilizada como base para a elaboração de sistemas mais robustos que facilitem a obtenção certificados digitais por parte de pessoas leigas.

A solução desenvolvida deve ser capaz de verificar os documentos de CNH recebidos e extrair os dados contidos nos mesmos para emissão dos certificados. A solução também deve permitir revogar e obter certificados emitidos anteriormente. Os dados referentes ao processo de emissão dos certificados, além do próprio certificado, devem ser persistidos e recuperáveis.

1.2. Escopo

O desenvolvimento da solução se limita ao desenvolvimento de uma API que permita emitir e revogar certificados, além de obter dados referentes a certificados já emitidos. O mecanismo de verificação utilizado pelo sistema se baseia em documentos de CNH digital. A solução desenvolvida não segue nenhum tipo de regulamentação referente a emissão de certificados. O trabalho não contempla o uso de mecanismos mais sofisticados para o gerenciamento das chaves criptográficas necessárias para algumas operações.

2. Tecnologias

A seguir será apresentada a linguagem de programação na qual a aplicação foi desenvolvida. Além disso também serão apresentadas bibliotecas e APIs que foram escolhidas para solucionar problemas identificados no processo de desenvolvimento.

2.1. Linguagem de programação

A linguagem de programação escolhida para o desenvolvimento da aplicação foi a Java da Oracle, em sua versão 11.0.16. A escolha se deu por três motivos, sendo o primeiro a afinidade e preferência pessoal do autor com a linguagem; o segundo motivo é o *framework* Spring Boot, escolhido como base para o desenvolvimento da aplicação. Por fim, o último motivo foi a presença de bibliotecas para integração da linguagem com outros componentes necessários para implementação da solução.

2.2. Base de dados

O sistema para gerenciamento da base de dados escolhido foi o MySQL; a escolha foi feita com base na familiaridade do autor e na presença de ferramentas no framework Spring Boot que facilitam a integração da solução com bases de dados deste tipo.

2.3. Tesseract e PDFBox

O Tesseract é um software de *Optical Character Recognition* (OCR) que permite o reconhecimento de caracteres contidos em imagens; ele pode ser executado em diversos sistemas operacionais e possui bibliotecas de integração com diversas linguagens de programação. O Tesseract foi escolhido por ser uma opção *open source*, confiável e por possuir bibliotecas, também de código aberto, para integração com a linguagem na qual o trabalho foi desenvolvido. A biblioteca utilizada para integração com a solução foi o

Tess4J. O PDFBox é uma biblioteca para linguagem Java que permite a manipulação de PDFs; com ela é possível criar documentos em Portable Document Format (PDF), além de manipular documentos já existentes. O fato de ser uma biblioteca de código aberto foi o motivo para sua escolha.

2.3.1. Verificador de Conformidade

O Verificador de Conformidade é um serviço que pode ser consumido como uma aplicação web ou API; ele é mantido pelo Instituto Nacional de Tecnologia da Informação (ITI) e permite a verificação da conformidade de assinaturas digitais qualificadas e avançadas em relação às regulamentações estabelecidas pela ICP-Brasil.

2.3.2. Hawa CA

O Hawa é um conjunto de softwares de gerenciamento e operação de autoridades certificadoras e autoridades registradoras; nesta família o Hawa CA é o software utilizado para operação de autoridades certificadoras, sendo capaz de emitir, atualizar e revogar certificados digitais avançados e qualificados.

2.4. Docker

Docker é uma plataforma de aplicações que permite virtualizar aplicações. Esta virtualização acontece com a criação de imagens docker, peças de software imutáveis que contém todas as especificações e arquivos necessários para executar uma aplicação. Com base nas imagens docker é possível então instanciar containers, que são a representação da aplicação especificada na imagem em execução. O uso de imagens e containers docker torna o processo de replicar a execução de aplicações mais simples, e foi utilizado neste trabalho para execução da base de dados, do Tesseract e da própria solução.

3. Análise do problema

As funcionalidades básicas estabelecidas para o sistema foram a emissão e revogação de certificados, além da possibilidade de recuperar certificados emitidos anteriormente. A implementação do processo de emissão foi responsável pelas principais escolhas de projeto realizadas. Estas escolhas serão descritas nas próximas sub-seções.

3.1. Verificação da assinatura do arquivo de CNH Digital

Os dados necessários para emitir os certificados digitais são retirados dos arquivos de CNH digital exportados pelo aplicativo ”Carteira Digital de Trânsito”[2], que possuem assinaturas realizadas pelas unidades estaduais do Departamento Estadual de Trânsito (DETRAN) associadas ao estado de emissão do documento.

A aplicação sugerida para validar os documentos é o Assinador Serpro[3]. Esta alternativa foi descartada devido à complexidade do processo de instalação do software e a ausência de processos de verificação automáticos.

[2]https://www.gov.br/pt-br/servicos/obter-carteira-digital-de-transito

[3]https://www.serpro.gov.br/links-fixos-superiores/assinador-digital/assinador- serpro

Outra alternativa, que foi a escolhida, é a proposta pela Secretaria Nacional de Trânsito (SENATRAN): o uso do Verificador ITI [4], uma solução que pode ser usada diretamente via navegadores ou chamadas de API.

3.2. Extrair dados das CNHs

Durante o processo de desenvolvimento da lógica de extração dos dados foram identificados três modelos válidos de documento de CNH digital. Ao analisar o corpo dos documentos recebidos pela aplicação pode-se verificar a presença de dois tipos de componentes: textos padronizados e imagens.

Sabendo que todas as informações úteis dos documentos se encontravam nas imagens, o próximo passo foi a busca de um meio de extrair tais informações utilizando Java. Após pesquisas, a ferramenta escolhida foi o Tesseract[5], um mecanismo de OCR de código aberto que possui bibliotecas de integração com diversas linguagens. Para integração do Tesseract com o Java foi utilizada a biblioteca Tess4j[6]).

Testes de extração foram realizados utilizando imagens geradas a partir da conversão completa dos PDFs para imagens. Para todos os modelos de CNH digital o Tesseract foi incapaz de extrair as informações contidas nas imagens dos PDFs.

Para a realizar esta tarefa foi escolhido o Apache PDFBox[7], uma biblioteca de código aberto para Java de manipulação de documentos PDF. Utilizando-a foi possível extrair com sucesso todas as imagens contidas nos documentos, que puderam então ser usados como input para o Tesseract.

Com as imagens obtidas, novas tentativas de extração de dados com o Tesseract foram realizadas, as quais não geraram resultados satisfatórios. A presença de fontes diferentes, caixas de texto e informações escritas de maneira vertical acabaram sendo fatores que influenciaram negativamente a maneira como o Tesseract identificou e extraiu os dados.

Para as tentativas seguintes a estratégia de extração foi alterada de maneira que cada uma das informações fosse obtida individualmente. Utilizando a biblioteca Tess4J foram fornecidas coordenadas, especificadas em pixel, que permitiram delimitar a área da imagem contendo o texto a ser extraído. Ao seguir esta estratégia foi possível extrair com sucesso as informações contidas no documento de maneira satisfatória.

Entretanto esta abordagem acabou acarretando em aumento no tempo de execução do Tesseract, uma vez que múltiplas rodadas de reconhecimento passaram a ser necessárias para extrair as informações.

Por fim, na etapa de pós-processamento das informações foi necessário remover os caracteres *new line* adicionados pelo próprio Tesseract no processo de extração. Além disso, foram implementados métodos para validar o número de Cadastro de Pessoa Física (CPF) e data de nascimento extraídos, informações utilizadas para emitir os certificados. Sendo assim, caso estas informações estejam fora do formato esperado a emissão do certificado não é realizada.

[4]https://verificador.iti.gov.br/verifier-2.10/
[5]https://github.com/tesseract-ocr
[6]https://tess4j.sourceforge.net/
[7]https://pdfbox.apache.org/

3.3. Chaves necessárias

Durante o processo de implementação do sistema foi necessário a criação de algum mecanismo para criação de chaves criptográficas necessárias para emissão dos certificados digitais em conjunto com o sistema Hawa CA. Para atender esta necessidade foi implementado um mecanismo simples de criação de chave do tipo RSA em Java utilizando a biblioteca bouncycastle; as chaves geradas tem a suas chaves públicas extraídas para uso no processo de emissão, enquanto as chaves privadas são cifradas e posteriormente armazenadas na base de dados da aplicação.

3.4. Persistir as informações dos dados usados para emitir os certificados

Após a implementação do fluxo de assinatura foi identificada a necessidade de algum mecanismo interno que permitisse averiguar quais dados foram verificados e usados na emissão de algum certificado.

A solução adotada foi a criação de um dossiê interno da emissão de um certificado. O dossiê em questão é um documento XML assinado que contém o *hash* de cada uma das informações usadas e obtidas no processo de emissão: a CNH Digital, o relatório do serviço Verificador ITI e a as informações extraídas pelo Tesseract. Este documento é gerado após a emissão bem sucedida de um certificado. A Figura 1 apresenta o fluxo de criação deste documento.

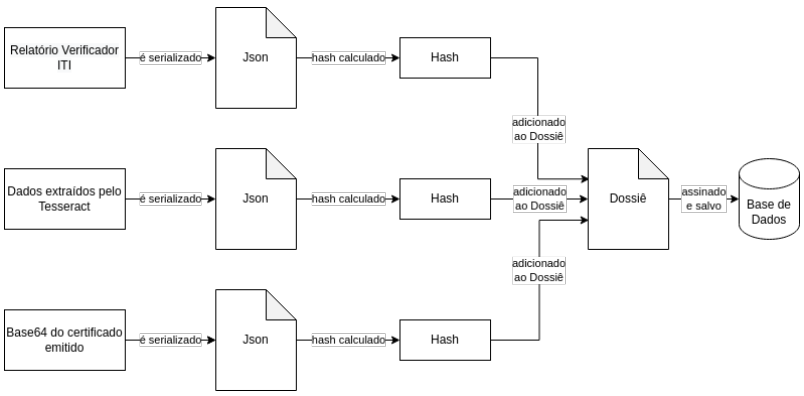


Figura 1. Fluxo de criação do Dossiê

4. Desenvolvimento do Sistema

Os principais requisitos definidos para o sistema foram a implementação das funcionalidades de emissão e revogação de certificados, além da obtenção de um certificado previamente emitido.

Com base nesses requisitos o sistema foi sendo construído em três etapas consecutivas, uma para cada funcionalidade. Notavelmente a primeira etapa, referente à implementação da funcionalidade de emissão, foi a mais demorada e trabalhosa. A segunda e terceira etapas foram respectivamente a de implementação da revogação e obtenção de certificados já emitidos.

5. Resultados

A API obtida com o desenvolvimento é capaz de emitir e revogar certificados, além de retornar certificados emitidos anteriormente e os documentos utilizados e gerados nesse

processo: o dossiê assinado, relatório do serviço Verificador ITI e dados extraídos pelo Tesseract. Todas essas funcionalidades são acessadas por meio de chamadas de API. O código fonte pode ser acessado no repositório do GitHub da aplicação[8].

Cada uma dessas funcionalidades demanda o uso de recursos específicos do sistema, provenientes de componentes de software chamados *services*, responsáveis por isolar a lógica referente ao funcionamento de partes específicas do sistema. Em seu estado atual a aplicação conta com oito *services*, apresentados e descritos na Tabela 1.

Tabela 1. Services implementados

CNHService	Service que integra todos os demais, acessando suas funcionalidades e realizando a persistência de novas entidades de sistema criadas, como certificados e dossiês.
EncodeAndDecodeService	Provê a lógica necessária para armazenar e carregar documentos, codificados em Base64, da base de dados do sistema.
DossierSignerService	Service que gera e assina dossiês para cada certificado emitido no sistema. Este service utiliza uma chave RSA armazenada em uma JKS para realizar a assinatura do dossiê gerado.
HawaCaService	Service responsável por realizar a comunicação com o Hawa CA para emissão e revogação de certificados.
KeyService	Service que realiza a criação de chaves RSA de 2048 para emissão dos certificados. As chaves privadas geradas são cifradas utilizando , e posteriormente armazenadas na base de dados.
PDFBoxService	Service responsável por utilizar a biblioteca PDFBox para extrair as imagens contidas nos documentos PDF recebidos.
TesseractService	Service responsável por utilizar o Tesseract para extrair as informações contidas nas imagens obtidas pelo service PDFBoxService.
VerifierService	Service responsável por utilizar a API Verificador ITI para verificar a validade da assinatura dos documentos PDF recebidos. Este service verifica a conformidade das assinaturas e validade dos certificados da cadeia de certificação.

Além dos *services* a aplicação utiliza quatro componentes externos: o Verificador ITI, o Hawa CA, uma JKS e a base de dados MySQL. Para a interação com esses componentes a aplicação deve ser configurada através de variáveis de ambiente descritas nos arquivos do repositório da solução.

Por fim, a Figura 2 fornece uma visão geral do sistema, e como ocorre a interação entre os diferentes componentes.

5.1. Funcionalidades implementadas

As funcionalidades da API estão disponíveis a partir do endereço base /cnh. A Tabela 2 descreve os endpoints e as funcionalidades associadas. A estrutura do corpo das requisições para cada endpoint e as respostas geradas estão listadas no arquivo README do repositório da solução.

6. Conclusões

Inicialmente este trabalho apresentou conceitos pertinentes e necessários relacionados à área de assinatura digital. Com esses conceitos apresentados foi desenvolvida uma aplicação capaz de verificar e extrair os dados contidos em uma CNH digital com sucesso e utilizá-los para emitir e revogar certificados digitais com as funcionalidades do sistema Hawa CA; além disso, a aplicação também é capaz de retornar certificados emitidos previamente.

[8]https://github.com/artmra/advanced-certificate

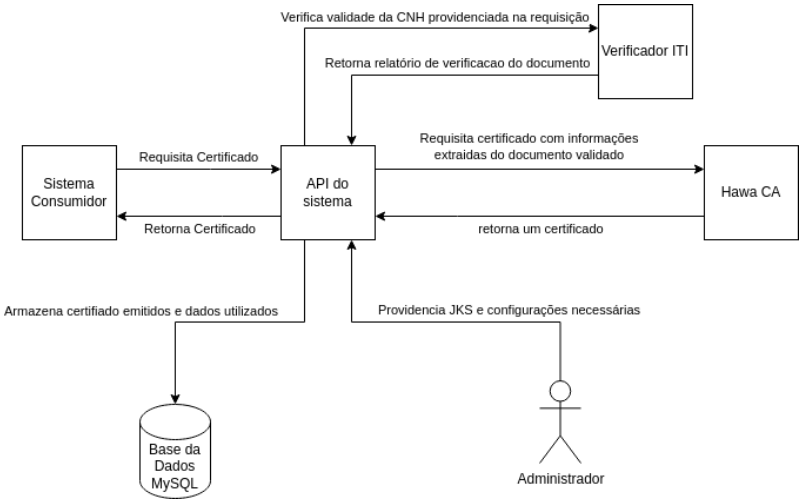


Figura 2. Visão geral do sistema

Tabela 2. Funcionalidades implementadas

Endpoint	**Descrição**
/issue	Permite emitir um certificado
/revoke	Permite revogar um certificado emitido anteriormente
/get-cert	Permite obter um certificado emitido anteriormente
/get-dossier	Permite obter o dossie associado a um certificado emitido anteriormente
/get-extracted-cnh-info	Permite obter os dados extraídos pelo Tesseract no processo de emissão de um certificado
/get-cnh	Permite obter a cnh associada ao processo de emissão de um certificado
/get-verifier-response	Permite obter o relatório gerado pelo serviço Verificador ITI associado ao processo de emissão de um certificado

Também foram apresentados os motivos que influenciaram na escolha de ferramentas utilizadas no desenvolvimento da aplicação, como o Tesseract e PDFBox, e em seguida apresentados os resultados obtidos.

Sistemas que decidam utilizar as funcionalidades desenvolvidas para esta API devem levar em consideração as limitações apresentadas. A implementação de outras alternativas para o gerenciamento das chaves criptográficas e validação dos dados extraídos pelo Tesseract devem ser considerados objetivos prioritários.

Além disso, a definição de políticas de certificação que utilizem todos os dados extraídos é algo que também deve ser considerado, assim como o desenvolvimento de ferramentas para o gerenciamento dos usuários e certificados armazenados na base de dados no sistema.

Finalmente, as funcionalidades atuais do sistema também podem ser utilizadas como base para implementar fluxos de emissão que utilizem outros documentos digitais validados.

Referências

(2001). *MEDIDA PROVISÓRIA No 2.200-2, DE 24 DE AGOSTO DE 2001*. Presidência da República.

(2014). *REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC*. THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION.

(2019). *Recommendation ITU-T X.509*. ITU.

(2020). *DECRETO Nº 10.543, DE 13 DE NOVEMBRO DE 2020*. Presidência da República.

(2021). *RESOLUÇÃO CG ICP-BRASIL Nº 182, DE 18 DE FEVEREIRO DE 2021*. Instituto Nacional de Tecnologia da Informação.

Afshar, R. (2015). Digital certificates.

Agrawal, M. and Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. *International journal on Computer Science and Engineerin*, 4(5):877–882.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2022). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF.

Delfs, H. and Knebl, H. (2002). *Introduction to Cryptography*. Springer-Verlag Berlin Heidelberg, Nurnberg, Germany, 1 edition.

Kaur, R. and Kaur, A. (2012). Digital signature. *International Conference on Computing Sciences*, (4):295–301.

Kohnfelder, L. M. (1978). Towards a practical public-key cryptosystem. *Massachusetts Institute of Technology*.

Maqsood, F., Ali, M. M., Ahmed, M., and Shah, M. A. (2017). Cryptography: A comparative analysis for modern techniques. *International journal of Advanced Computer Science and Applications*, 8(6):442–448.

Santesson, S., Nystrom, M., and Polk, T. (2004). *Internet X.509 Public Key Infrastructure: Qualified Certificates Profile*. IETF.

Schukat, M. and Cortijo, P. (2015). Public key infrastructures and digital certificates for the internet of things. *Irish Signals and Systems Conference (ISSC)*, 26.

Surya, E. and C.Diviya (2011). A survey on symmetric key encryption algorithms. *International journal of Advanced Computer Science and Applications*, 2(4):475–477.

Weise, J. (2001). Public key infrastructure overview. *Sun BluePrints™ OnLine*, 2.

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., and Khamayseh, Y. (2017). *Introduction to Cryptography*. IEEE.

ANEXO A – ESTRUTURA DA RESPOSTA DO SERVIÇO VERIFICADOR ITI

A estrutura da resposta da API do Verificador ITI utilizado no desenvolvimento está disponível nos seguintes endereços:

- `https://pbad.labsec.ufsc.br/codigos-de-referencia/docs/verifier-api/`
- `https://web.archive.org/web/20221223035327/https://pbad.labsec.ufsc.br/codigos-de-referencia/docs/verifier-api/`