UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CIÊNCIA DA COMPUTAÇÃO

Luciano José Dos Santos Rosa

Uma Proposta de Infraestrutura para Disponibilização de Assinatura Digital

São Luís - MA
2021

Luciano José Dos Santos Rosa

Uma Proposta de Infraestrutura para Disponibilização de Assinatura Digital

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação...

Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Mário Antônio Meireles Teixeira

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Rosa, Luciano José Dos Santos.
Uma Proposta de Infraestrutura para Disponibilização de Assinatura Digital / Luciano José Dos Santos Rosa. - 2021.
57 f.

Orientador(a): Mário Antonio Meireles Teixeira.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís - MA, 2021.

1. Aplicação web - Assinatura Digital. 2. Assinatura Digital. 3. Criptografia. 4. Hash. I. Teixeira, Mário Antonio Meireles. II. Título.

Luciano José Dos Santos Rosa

Uma Proposta de Infraestrutura para Disponibilização de Assinatura Digital

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação..

BANCA EXAMINADORA

Prof. Dr. Mário Antonio Meireles Teixeira
Orientador
Universidade Federal do Maranhão

Prof. Me. Carlos Eduardo Portela Serra de Castro
Examinador Interno
Universidade Federal do Maranhão

Prof. Dr. Samyr Béliche Vale
Examinador Interno
Universidade Federal do Maranhão

São Luís - MA
2021

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por essa oportunidade, pois só ele sabe a luta que foi chegar até aqui e sem ele nada seria possível.

Gostaria de agradecer a minha família, em especial às minhas duas mães por sempre me apoiarem, me ajudarem e serem minha principal fonte de inspiração e força para continuar os estudos.

Agradeço também ao meu pai pelo apoio financeiro e incentivo contínuo aos estudos desde a minha infância.

Agradeço grandemente também à minha namorada Beatriz Castro e ao amigo Anderson Silva por suas imensas colaborações e ajudas com este trabalho.

Agradeço também ao meu orientador Professor Mário Meireles, por ter aceitado esse desafio, me orientando de maneira assertiva sempre.

Por último e não menos importante, gostaria de agradecer aos amigos de turma, Paulo Renato, Pedro e Ylderlan por suas ajudas indispensáveis durante todo o curso. Aos amigos Erik e Robert pelas caronas que sempre salvavam a ida e a volta à Universidade e a toda galera do grupo codebuilders que tornaram o dia a dia da graduação mais agradável com suas companhias, ajudas e brincadeiras.

"Ninguém é tão grande que não possa aprender, nem tão pequeno que não possa ensinar. "

(Esopo)

RESUMO

A evolução tecnológica cada vez mais forte e necessária, tem trazido uma alta demanda para automatizar processos completos de trabalho, e a Assinatura Digital tem trazido de forma eficiente e segura essa necessidade por diminuir distâncias, economizar tempo e agilizar processos de forma simples do ponto de vista dos usuários comuns. Dessa forma buscou-se estudar e criar uma arquitetura de Assinatura Digital que pudesse ser utilizada da forma mais simples possível de modo a facilitar o processo de autenticação de documentos no dia a dia. O objetivo deste trabalho é discorrer sobre Assinatura Digital, bem como seus conceitos e processos necessários. Para isso foi pesquisado ferramentas e estratégias para trazer uma proposta de Arquitetura de Assinatura Digital, além disso criou-se uma aplicação para demonstrar de forma prática o funcionamento de todo o processo necessário.

Palavras-chave: Assinatura Digital, Criptografia, Hash, Aplicação web - Assinatura Digital.

ABSTRACT

The increasingly strong and necessary technological evolution has brought a high demandto automate complete work processes, and the Digital Signature has brought this need efficiently and safely by reducing distances, saving time and streamlining processes in a simple way from the point of view of ordinary users. Thus, we sought to study and create a Digital Signature architecture that could be used in the simplest way possible in order to facilitate the process of authenticating documents on a daily basis. The objective of thiswork is to talk about Digital Signature, as well as its concepts and necessary processes.For this, tools and strategies were researched to bring a proposal for a Digital Signature Architecture, in addition to that, an application was created to demonstrate in a practical way the functioning of the entire necessary process.

Keywords: Digital Signature, Encryption, Hash, Web Application - Digital Signature.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIAÇÕES

AR	Autoridade Registro
CWI	Centrum Voor Wiskunde En Informatica
NCSA	Centro Nacional De Aplicações De Supercomputação
DES	Data Encryption Standart
DH	Diffie Hellman
ECC	Elliptic Curves Cryptography
ITI	Instituto Nacional De Tecnologia E Informação
IDEA	International Data Encryption Algorithm
IDEA	International Data Encryption Algorithm
ICP	Infraestrutura De Chaves Públicas
JS	Java script
LCR	Lista De Certificados Revogados
MD	Message Digest
MD5	Message Digest Algorithm 5

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

Com os avanços tecnológicos das redes de computadores e a massificação do uso da internet, o número de usuários cresce diariamente, sendo justificado pela eficiência do serviço, comodidade e redução dos custos. (CRUZ; STEINMANN, 2007)

O uso de documentos eletrônicos torna-se essencial para atender as necessidades cotidianas dos usuários. Neste contexto surge a demanda de garantir a segurança das informações que trafegam por essas grandes redes. (FRIEDRIH; MEDINA, 2007)

Dessa forma começou-se a estudar conceitos, estratégias e ferramentas necessárias para trazer para os usuários comuns, o conhecimento e entendimento das assinaturas digitais bem como melhorar e facilitar o seu uso.

1.2 Justificativa

Com o advento da tecnologia, inúmeras tarefas que anteriormente requeriam o engajamento físico e presencial, deixaram de existir economizando tempo e trabalho às pessoas. Como por exemplo, operações bancárias, que tempos atrás demandavam a presença física das pessoas a uma agência bancária e atualmente podem facilmente serem resolvidas pelo celular.

A Assinatura Digital surgiu como uma excelente solução para os métodos tradicionais de assinatura e autenticação de documentos que anteriormente exigiam papel e caneta, além, obviamente, de se ter o trabalho da presença física das pessoas na hora de assinar um documento quando mais de uma pessoa precisava assinar o mesmo documento por exemplo.

Desta forma a Assinatura Digital traz as já conhecidas vantagens do mundo digital de assinar um documento de maneira rápida, fácil e acessível, tornando o processo mais seguro e eficiente.

1.3 Objetivo Geral

Este trabalho tem por objetivo implantar um sistema de Assinatura Digital, simples, prático, de fácil acesso e utilização do ponto de vista da experiência do usuário, visando facilitar o processo e garantir a segurança e eficiência na assinatura de documentos digitais.

1.3.1 Objetivos Específicos

- Desenvolver os conceitos de Assinatura Digital;

- Levantar os requisitos para o desenvolvimento de uma aplicação Web;
- Modelar e planejar os aspectos para a construção de uma arquitetura;

1.4 Organização do Trabalho

Esta introdução compõe o Capítulo 1, onde se discorre sobre a contextualização, justificativa, objetivos e organização do trabalho.

No Capítulo 2, é feito o referencial teórico do trabalho onde são abordados os principais conceitos de criptografia, como a sua história, os tipos de criptografia, algoritmos e o seu funcionamento na Assinatura Digital.

O capítulo 3 aborda a Assinatura Digital, bem como seu funcionamento, sua composição e os Certificados Digitais.

No capítulo 4, são abordadas as principais tecnologias utilizadas para o desenvolvimento da aplicação web de Assinatura Digital como ferramentas de programação, linguagens e algoritmos.

No capítulo 5, é feita toda a demonstração da arquitetura proposta para a Assinatura Digital, seus processos, funcionamento, aplicação e engenharia de software.

No capítulo 6, é feita a conclusão onde finaliza-se este trabalho, juntamente com as considerações finais e recomendações para trabalhos futuros.

2 CRIPTOGRAFIA

A Criptografia surgiu a partir da necessidade de proteger canais de comunicação e o conteúdo da mensagem. De acordo com Ordonez, Pereira e Chiaramonte (2005) a criptografia está inserida nas civilizações há milênios de anos, sendo identificadas nas escritas antigas dos egípcios, os hieróglifos.

Etimologicamente o termo criptografia originou-se do Grego: *Kryptós* que significa oculto, secreto e *graphos* de graphein que significa escrita, dessa forma a Criptografia é o método utilizado para codificar mensagens. O termo Criptografia foi criado em 1920, mas a técnica certamente surgiu simultânea ao da escrita, sendo utilizada pelo Espartanos, em 400 a.C. Um dos métodos mais clássicos foi criado em 850, por um matemático AI-Kindus, publicou um manuscrito com decifrações de mensagens codificadas. (WAZLAWICK, 2016)

Há muitos anos surgiu a necessidade de manter informações em sigilo. Reis, rainhas e generais, tentavam proteger seus governos, tropas e exército, neste contexto surge a Criptografia, maneiras de codificar mensagens a fim de que apenas o destinatário pudesse ter acesso às informações contidas na mensagem. (SINGH, 1999)

Acredita-se que na Roma antiga os planos de guerra eram criptografados, assim só os aliados conseguiam entender o conteúdo da mensagem, esse método foi implementado como forma de assegurar que caso as tropas inimigas interceptassem a mensagem o conteúdo estaria protegido, por esta ilegível. (SINGH, 2012)

Neste período, simultâneo ao surgimento da criptografia, surgia também a criptoanálise, que consistia em descobrir os segredos contidos na mensagem. Os criptoanalistas eram conhecidos também como "Quebrar-Códigos" eram pessoas que tinham como função descriptografar as mensagens, ou quando interceptadas pelos inimigos, os criptoanalistas mudavam as informações contidas na mensagem antes que o destinatário pudesse recebê-las. (COSTA, 2010)

Outra forma de ocultar mensagem era através da Esteganografia, diferente da Criptografia, essa técnica consistia em esconder o conteúdo da mensagem através da camuflagem. A esteganografia foi mencionada por Heródoto (pai da história), para que a mensagem fosse protegida era necessário raspar a cabeça do mensageiro, tatuar a mensagem que esperar que o cabelo crescesse novamente, camuflando o conteúdo transmitido. Historicamente essa técnica mostrava um grau de segurança, porém caso fosse descoberta colocaria em risco a mensagem. (SINGH, 1999)

2.1 Cifras de César

Com o passar dos anos as técnicas para codificar mensagens foram se tornando cada vez mais eficientes. O general Júlio César utilizou o primeiro exemplo de código secreto que se tem notícias hoje, para enviar instruções aos seus generais. (COUTINHO, 2016)

A cifra de César, ou cifra de troca, e até mesmo a troca de César é uma das técnicas mais conhecidas de realizar uma mensagem criptografada. Foi desenvolvida pelo general romano Júlio César, 50 a. C, o princípio da técnica consistia em substituir a letra do alfabeto por outra que estivesse a três posições a frente (Figura 1). Essa técnica possui este nome pois foi uma das formas de comunicação realizada entre o Imperador Júlio César e os seus generais nas Guerras de Gália. (SINGH, 2011)

Figura 1- Cifra de César

Alfabeto:	A	B	C	D	E	F	G	H	I	J	K	L	M
Cifra:	D	E	F	G	H	I	J	K	L	M	N	O	P
Alfabeto:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cifra:	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

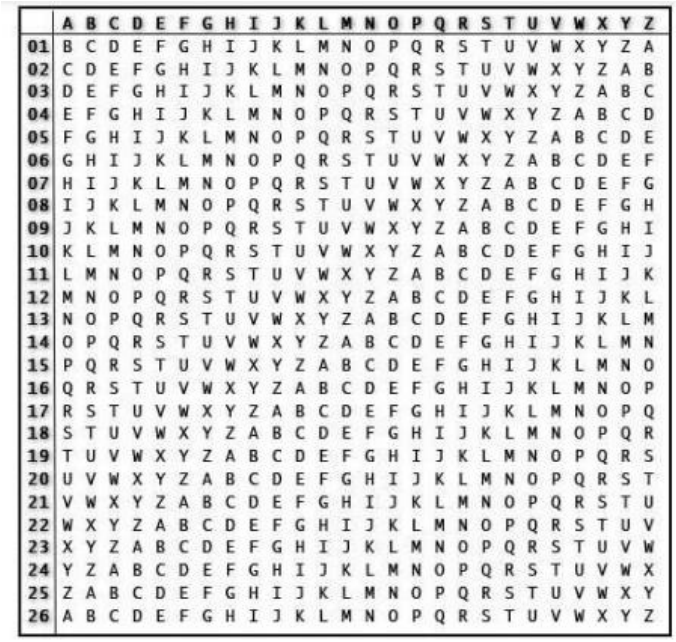
Fonte: TEIXEIRA, Marco Antonio Fávaro. Números inteiros e criptografia RSA. 2020.

A criptografia por substituição era muito utilizada o que tornou a decifração do conteúdo da mensagem fácil, com isso surgiu outras formas de tornar a mensagem ilegível como a técnica do quadrado de Vigenère. (SINGH, 2002)

2.2 Quadro De Vigenère

A técnica foi criada em 1563 por Blaise de Vigenère, a criptografia necessitava do uso de uma tabela para criptografar e descriptografar o conteúdo da mensagem. O princípio da cifra de Vigenère consistia em construir uma tabela onde a primeira linha estivesse disposto o alfabeto original, a segunda linha estaria a cifra de César e a partir da terceira linha seguiria o deslocamento de cada letra a partir da comparação da linha anterior. (Figura 2) (FREIRE; CASTILHO 2005)

Figura 2- Quadro de Vegenère



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
01	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
02	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
03	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
04	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
05	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
06	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
07	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
08	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
09	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
26	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Fonte: PEREIRA, Nádia Marques Ikeda. Criptografia: uma nova proposta de ensino de matemática no ciclo básico. 2015.

O quadro de Vigenère foi produzido para quebrar a descrição através da análise da frequência. Atualmente considera-se o quadro de Vigenère facilmente decifrável, porém foi considerado altamente eficaz na época sendo quebrado somente após 300 anos de utilização. (GROENWALD; FRANKE; OLGIN, 2003)

A cifra de Vigenère é um aprimoramento da utilizada por Cesar, tornando-se diferente apenas na substituição que não é a mesma para todas as letras, no qual cada letra possui uma chave diferente, ou seja uma substituição diferente. Dessa forma afirma-se que a cifra de Vigenère é apenas uma sequência de cifras de César. (MARQUÊS, 2013)

Segundo Nogueira (2017) A cifra de Vigenère também pode ser traduzida através de algoritmos citado abaixo.

$Ci = Pi + ai \pmod{26}$, (30)

em que Ci representa o valor de cada letra do texto cifrado, Pi representa o valor da letra do texto original e a quantidade de posições que a letra será deslocada. A decodificação pode ser vista como:

$Pi = Ci - ai \pmod{26}$

2.3 Máquina Enigmas

Durante a 2° Guerra mundial a criptografia era amplamente utilizada entre as tropas, porém consistia em técnicas simples, as mensagens eram descritas com papel e lápis, logos erros eram comumente identificados. Foi então que em 1915, dois britânicos inventaram uma máquina capaz de produzir códigos tidos como indecifráveis, através de um sistema eletromecânico. (Figura 3) (ELIS, 2005)

A máquina enigma foi muito utilizada pelas forças armadas alemãs a fim de codificar sua mensagem, a partir de 1924, eles a consideravam de extrema segurança, imbatível. Dessa forma as tropas britânicas iniciaram a busca pela descoberta do enigma, foram construídas instalações militares secretas, onde matemáticos, jogadores profissionais de palavras cruzadas e xadrez, cientistas e até mesmo astrólogos foram convocados para decodificar as mensagens. Foi então que em 1940 o enigma foi descoberto por Alan Turing, matemático da universidade de Cambridge, que estabeleceu os princípios da computação moderna. (ELIS, 2005)

Figura 3- Máquina Enigma

Fonte: PEREIRA, Nádia Marques Ikeda. Criptografia: uma nova proposta de ensino de matemática no ciclo básico. 2015.

O matemático britânico Alan Turing, conhecido como um dos fundadores da Ciências da computação, desempenhou um papel fundamental na história da criptografia, sendo o principal responsável por decifrar mensagens do exército alemão na 2° Guerra mundial. (COUTINHO, 2016)

Ao longo da história, os avanços da Criptografia ultrapassaram a imbatível máquina enigma. Os algoritmos clássicos foram substituídos por algoritmos computacionais. (TEIXEIRA, 2020)

> Os algoritmos tradicionais do período clássico da criptografia cederam lugar aos algoritmos computacionais para criptografar dados. A tecnologia de processamento de dados oferecida pelos computadores acelerou a forma de encriptar e decriptar informações, tornando essa prática significativamente complexa. Segundo Weber (1995, p. 6)

A utilização de códigos ficou cada vez mais importantes, o que fez com que diversos estudiosos de diferentes áreas como matemáticos, engenheiro, cientistas da computação, se dedicassem ao desenvolvimento de técnicas mais eficaz para codificar e decodificar mensagens. (TEIXEIRA, 2020)

Com os avanços tecnológicos, principalmente a partir da propagação de computadores e o surgimento da internet, houve uma necessidade ainda maior, a segurança de dados e informações. (TEIXEIRA, 2020)

2.4 Sistemas De Criptografia

A criptografia pode ser classificada através das cifras, como a de César, a de Vigenère, mas também podem ser classificadas de acordo com a quantidade de chaves utilizadas. (JASPER,2009)

De acordo com ICP Brasil, (2009) as chaves criptográficas são os valores numéricos ou códigos utilizados para transformar, validar, autenticar e decifrar dados.

Com relação a classificação através de chaves existem dois tipos de criptografia a simétrica e a assimétrica, que se diferem a partir da quantidade de chaves utilizadas em seu algoritmo. (JASPER, 2009)

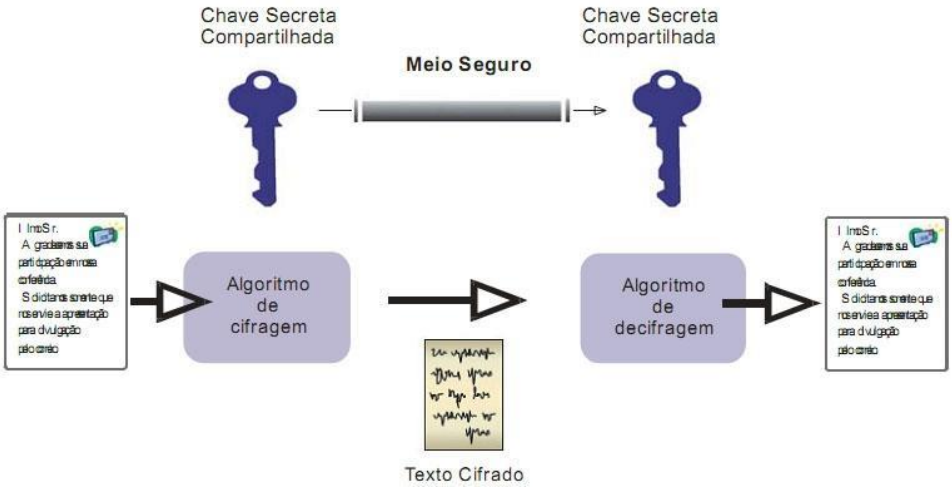
2.4.1 Criptografia Simétrica ou Chave Privada

Também conhecida como criptografia convencional, ou criptografia de chave privada, foi o primeiro tipo de criptografia a ser criada. É um modelo que utiliza uma chave única para realizar a encriptação e a decriptação. (PETRI, 2004)

Cavalcante (2005) afirma que a chave Simétrica funciona transformando a mensagem em um texto cifrado utilizando uma chave secreta, o destinatário ao receber o texto cifrado utilizará o mesmo código para transformar novamente em texto.

Exemplificando a definição de criptografia Simétrica, quando a origem (JOÃO) envia uma mensagem ele utiliza uma chave para transformar o texto em uma mensagem cifrada, dessa forma quando o destino (Daniel) ao receber a mensagem cifrada utiliza a mesma chave utilizada pela origem (JOÃO) para transformar o texto cifrado novamente em mensagem. (OLIVEIRA, 2012) (Figura 4)

Figura 4- Criptografia Simétrica



Fonte: Adaptado da internet, disponível em: https://www.gta.ufrj.br/grad/07_2/delio/Criptografiasimtrica.html (25/03/2021)

A utilização de chaves únicas entre a origem e o destino foi desenvolvida para ampliar a segurança das informações contidas nas mensagens, anteriormente caso houvesse interceptação da mensagem se um intruso conhecesse o algoritmo de ciframento seria facilmente descoberta o conteúdo da mensagem. Dessa forma apenas os interessados terão

acesso ao conteúdo da mensagem, visto que caso um intruso soubesse o algoritmo, ele ainda não conseguiria decifrar a mensagem por não possuir a chave de segurança. (OLIVEIRA, 2012)

JASPER (2009) afirma que o maior problema desse modelo de criptografia está na distribuição das chaves. Uma única chave é capaz de realizar dois processos criptografar uma mensagem, bem como descriptografá-la uma vez descoberta uma das chaves torta todo processo inseguro.

A utilização de criptografia Simétricas são utilizadas quando os dados contidos na mensagem não necessitam de um grande nível de segurança, para o autor uma das desvantagens desse método é que não apenas o transmissor tem conhecimento sobre a chave utilizada, mas também o receptor, assim como o volume de dados transmitidos é limitado. (CAVALCANTE, 2005)

Existem diversos algoritmos criptográficos que fazem uso da chave Simétrica.

2.4.2 Data Encryption Standart (DES)

O Data Encryption Standart foi o primeiro modelo de algoritmo de Criptografia Simétrica moderna, desenvolvido na década de 70 pela IBM, a princípio foi protegido pela agência nacional de segurança Americana, mas tornou-se padrão por ser o principal e mais difundido Algoritmo de chave privada do mundo. Em 1997 o DES foi quebrado após tentativas de todas as combinações possíveis. (Teixeira, 2006)

Foi desenvolvida em 1997, usa a Criptografia de 64 bits, é um codificador que corresponde a cerca de 72 quadrilhões de chaves diferentes. O DES cifra blocos de 64 bits, e utiliza chaves compostas por 56 bits, com 8 bits de paridade, e utiliza o algoritmo de Feistel. (JASPER, 2009)

Uma das aplicações do DES foi em um protocolo de segurança pela internet. O objetivo era garantir uma comunicação segura entre o servidor e o browser do cliente. (SANTOS, 2005)

Segundo Teixeira (2006) o DES é utilizado na cifragem de senhas do sistema UNIX, bem como em sistemas de segurança de e-mail, e em outros protocolos de segurança como S-HTTP.

O DES é muito eficiente quando se trata de aplicações simples, mas ele perde qualidade quando se trata de atuações em processos ultrassecretos pois há possibilidade de ser quebrado em pouco tempo. (BARBOSA et. al., 2005)

2.4.3 International Data Encryption Algorithm (IDEA)

O IDEA surgiu com objetivo de substituir o DES devido a maior desvantagem desse padrão de ciframento de dados que é a falta de segurança. O INTERNATIONAL DATA ENCRYPTION ALGORITHM (Idea) foi criado em 1990, por James Massey e Xuejia Lai, como um novo padrão de ciframento em blocos. (MORAES, 2004)

Foi criado em 1991 por Massey e Xuejia Lai, utiliza 128 bits, a ideia consistia em misturar as operações de grupos algébricos diferentes, para misturar os caracteres iniciais de forma a ficarem incompreensíveis. (LEONG, 2000)

Uma outra vantagem superior ao padrão DES é que na maioria dos microprocessadores, o software do IDEA é mais rápido do que a implementação do DES. (MAIA; PUGLIESE, 2000)

O IDEA, é utilizado no mercado financeiro e também no PGP, que é o programa para criptografia de email pessoal mais utilizado do mundo. (TEIXEIRA, 2006)

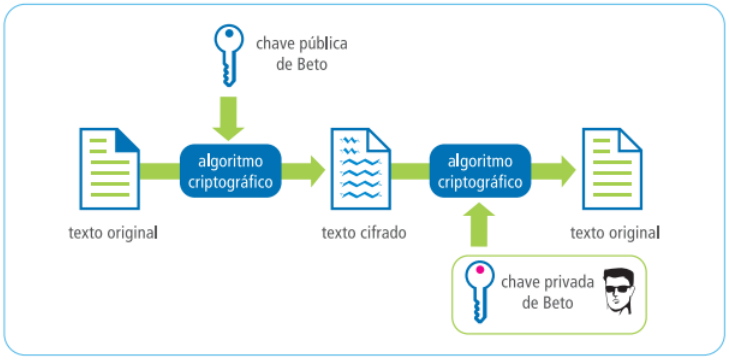
Apesar dos algoritmos simétricos serem mais rápidos do que os assimétricos, a utilização de chave única apresenta erros graves, como a impossibilidade de ser utilizada como fim de autenticidade, já que a chave privada pode ser compartilhada.

2.5 Criptografia Assimétrica ou Chave Pública

Este modelo de criptografia foi criado em 1970 pelo matemático Clifford Cocks, que trabalhava no serviço secreto inglês. (OLIVEIRA, 2012)

A criptografia Assimétrica também conhecida como chave pública utiliza duas chaves, uma pública e outra privada, a primeira é utilizada para cifrar a mensagem, a segunda possui a função de decifrar. (Figura 5) (CAVALCANTE, 2004)

Figura 5- Criptografia Assimétrica



Fonte: DE LA ROCHA LADEIRA, Ricardo; RAUGUST, Anderson Schwede. Uma análise da complexidade do algoritmo RSA implementado com o teste probabilístico de Miller-Rabin. **Revista de Empreendedorismo, Inovação e Tecnologia**, v. 4, n. 1, p. 24-33, 2017.

O método de criptografia chave pública, é de conhecimento público, a chave privada é utilizada apenas pela pessoa que deseja decifrar as informações contidas na mensagem. Por utilizar duas chaves a Criptografia assimétrica tende a ser mais segura do que a Simétrica. (SILVA; OLIVEIRA, 2017)

Um dos problemas do sistema de criptografia Simétrico estava na distribuição de chaves, a solução para essa desvantagem foi uma das maiores revoluções da criptografia. (SINGH, 2008)

A finalidade inicial da chave pública se semelha a chave privada, ou seja, promover o sigilo nas trocas de informações, porém as chaves Assimétricas possuem propriedades adicionais, que permitem ampliar a aplicabilidade desse modelo de Criptografia, que são sigilo, confiabilidade, autenticidade e não repúdio. (SCHNEIER, 1996)

A chave pública foi desenvolvida para garantir uma segurança ainda maior durante o compartilhamento de dados. A origem (DANIEL) deseja enviar um texto, porém deseja que esse texto tenha uma segurança maior, dessa forma, a origem (Daniel) opta por usar uma chave assimétrica. O destino (JOÃO) possui um par de chaves, uma pública utilizada para codificar texto, que ele pode compartilhar, e outra privada, que é pessoal, utilizada para decodificar. O destino (João) compartilha a chave pública com a origem (Daniel) que irá codificar e enviar. Caso um invasor (Pedro) consiga ter acesso a mensagem, ele não conseguirá descobrir o conteúdo da mensagem, pois, para decodificar é necessário utilizar a chave privada que é de domínio apenas do destinatário (João). (BRAGA; DAHAB, 2018)

Uma das maiores vantagens da criptografia Assimétrica é a segurança, e a possibilidade de qualquer pessoa possa enviar uma mensagem criptografada utilizando apenas a chave

pública do destinatário. Por ser uma chave pública não há necessidade do compartilhamento de chaves como é feito no modo Simétrico, logo, enquanto a chave privada estiver em sigilo do destinatário, suas informações também estarão em segredo. (OLIVEIRA, 2012)

Já as desvantagens mencionadas por Oliveira (2012) estão na complexidade dos algoritmos utilizada na criptografia Assimétrica, o tempo de processamento das mensagens é maior do que na Criptografia Simétrica, o que limita algumas ações, necessitando de um poder de processamento computacional grande.

2.5.1 Sigilo da encriptação assimétrica

A Criptografia de chaves públicas foi desenvolvida para garantir o sigilo durante as trocas de informações. Esse sistema permite que qualquer um envie criptogramas para o proprietário da chave privada. Mesmo que a troca de mensagem seja realizada por um canal inseguro as informações estarão em sigilo, pois apenas o destinatário conseguirá obter o texto claro, mesmo se houver interceptação da mensagem, o conteúdo estará protegido, apenas o proprietário do par de chaves pública e privada conseguirá decodificar a mensagem. (SILVA, 2004)

A confidencialidade permite que apenas entidades autorizadas tenham acesso a informações contidas nos documentos. (BRAGA; DAHAB, 2018)

2.5.2 Autenticidade da encriptação assimétrica

A Criptografia de chaves públicas permite que a troca de informações seja autêntica e íntegra. O procedimento para garantir a autenticação da mensagem é o oposto do sigilo. A assinatura digital é realizada pelo proprietário do par de chaves (pública e privada), no qual todos aqueles que têm acesso a chave pública podem verificar a autenticidade da assinatura, visto que somente quem tem acesso a chave privada poderá gerar a assinatura digital. (BRAGA; DAHAB, 2018)

2.5.3 Não- repúdio da encriptação assimétrica

Também conhecida como irretratabilidade ou irrefutabilidade, é um princípio que garante que o emissor não negue sua autoria. (FREITAS, 2018)

A assinatura digital é realizada a partir de uma chave privada, então qualquer um que tenha acesso a chave pública correspondente consegue verificar a autenticidade da assinatura, logo a mensagem não pode ser enviada por mais ninguém além do proprietário do par de chaves. Previne que alguém negue o envio ou o recebimento da mensagem. (BRAGA; DAHAB, 2018)

2.5.4 Algoritmo RSA

O algoritmo RSA foi desenvolvido no Massachusetts Institute of Technology (MIT) em 1978, cujo o nome foi batizado a partir das iniciais dos seus autores Ron Rivest, Adi Shamir e Leonard Adleman. Atualmente é o algoritmo criptográfico assimétrico mais conhecido e utilizado. (SANTOS, 2000)

O algoritmo RSA utiliza a técnica de fatoração de dois números primos grandes como base de segurança, ou seja, para que o algoritmo seja seguro é necessário utilizar chaves na ordem de 1776 bits. (SILVA et, al., 2013)

A ideia inicial por trás do RSA é a facilidade em multiplicar dois números primos, a fim de obter um terceiro número, mas a recuperação dos números primos iniciais a partir do resultado torna-se difícil. O algoritmo RSA consiste em utilizar números primos grandes, estima-se que o tempo necessário para fatorar um número de 308 dígitos é de 100 mil anos, tornando o computacionalmente inquebrável. (OLIVEIRA, 2021)

Para gerar chaves criptográficas através do algoritmo RSA primeiro tem que possuir dois números primos inteiros e diferentes denominados p e q, deles obtém-se n= p.q. em seguida utiliza-se a função de Euler para gerar uma condição que satisfaça a números naturais inferiores ou iguais a n e que são primos com n. O processo pode ser simples mas torna-se difícil de achar números inteiros que satisfaçam essas condições. (CAMPELO; LEAL, 2007)

Moraes (2004) afirma que o RSA é o Algoritmo mais utilizado no comércio eletrônico com chaves de 512 bits, ele tem como aplicabilidade a distribuição de chaves, mas também é muito eficaz para criptografar e descriptografar mensagens. Outra função do Algoritmo RSA são as seguranças de certificados digitais e assinaturas digitais.

Criação de chave

```
##Determinação das Chaves Pública e Privada:
##
## -> p,q dois números primos.
## <- (e,n) e (d,n), as chaves públicas e privadas. function chaves(p, q)
n = p*q;
fi = (p-1)*(q-1);
```

```
e = 2;
k = 0;
do
e = e+1;
until (gcd(fi,e) == 1)
achou = false;
while (!achou)
d = (1 + (k * fi))/e;
if ( d == round(d))
achou = true;
else k = k+1;
endif
endwhile
 printf("\n A chave pública ´e (%d, %d). \n", e, n);
printf("\n A chave privada ´e (%d, %d). \n \n", d, n);
endfunction
```

Para os valores de p = 11 e q = 23 ter-se-ia:

```
octave> chaves(11,23)
 A chave pública ´e (3, 253).
```

2.5.5 Diffie Hellman

O grande problema do sistema de Criptografia simétrica é a dificuldade de estabelecer um compartilhamento de chaves seguro. Uma das possíveis soluções é a utilização do Algoritmo criado por Whitfield Diffier e Martin Hellman. (PIRES; COSTA, 2007)

O protocolo de Diffier e Hellman foi publicado pela primeira vez em 1976, esse protocolo permite que dois usuários possam compartilhar uma chave secreta por meio de um canal inseguro. (FIGUEIREDO, 2010)

O acordo de chave exponencial consiste na possibilidade de dois usuários gerarem o mesmo segredo, desde que eles possuam suas próprias chaves privadas e tenham conhecimento da chave pública de sua contraparte, que poderá ser utilizado como chave Simétrica para compartilhamento de dados. (BURNETT, STEVEN; PAINE 2002)

A segurança deste protocolo está na dificuldade de calcular logaritmos discretos em um campo infinito, por outro lado com a facilidade de realizar exponenciais em um mesmo campo. (TKOTZ, 2005)

De acordo com Pires e Costa (2010), o DH funciona da seguinte maneira:

Sejam q um número primo e α uma raiz primitiva de q .

1 Alice gera um valor privado Xa e Bob gera um valor privado Xb ;

2 Alice envia Ya = α Xa mod q para Bob e Bob envia Yb = α Xb mod q para Alice;

3 A chave secreta K = (Yb) (Xa) mod q de Alice e a chave secreta K = (Ya) (Xb) mod q de Bob são iguais.

Xa e Xb são ditas as chaves privadas de Alice e de Bob, respectivamente. Enquanto, Ya e Yb são ditas chaves públicas do acordo.

O Diffier- Hellman é um protocolo que oferece apenas trocas de chaves secretas por meio inseguro como a internet. Esse sistema não permite que sejam realizadas criptografias de mensagens nem assinatura digital. (BARBOSA ET Al., 2003)

Outra forma de aplicação do DH está no compartilhamento de senhas. Se houver a necessidade de uma senha para acessar um volume criptografado, todos os participantes podem utilizar o Algoritmo DH para gerá-la. Basta que sejam obedecidos os requisitos contidos no acordo de chaves exponenciais. (MELO; PIRES, 2010)

2.5.6 Curvas elípticas

O modelo Elliptic Curves Cryptography (ECC) foi criado em 1985 por Neal Koblitz e Victor Miller, os autores não inventaram um novo Algoritmo, mas houve a implementação de um Algoritmo de chaves públicas já existentes utilizando a matemática de curvas elípticas. (MAIA; PUGLIESE, 2000)

O sistema ECC consiste em modificações de Algoritmos já existentes que possam trabalhar no domínio de curvas elípticas ao invés de trabalharem em corpos finitos. Atualmente é um sistema de criptografia mais seguro que trabalha com chaves pequenas. (OLIVEIRA, 2012)

Utilizando o sistema de curvas elípticas resolve um dos maiores problemas dos Algoritmos de chaves públicas que é o grande tamanho das suas chaves.

Em contrapartida esse Algoritmo mesmo tendo potencial para serem mais rápidos que os já existentes acabam sendo mais demorados devido sua matemática ser mais complicada. (MAIA; PUGLIESE, 2000)

O método da curva elíptica é um dos mais modernos utilizados atualmente, possibilita tanto a cifragem e decifragem de mensagem, quanto a trocas de chaves e assinatura digital. (FIGUEIREDO; GADIS, 2004)

3 ASSINATURA DIGITAL

A assinatura digital é uma das aplicações da Criptografia de chave pública, ela garante a autoria de uma determinada informação. A assinatura digital é um código que está associado a uma mensagem, no qual gera a autenticidade da informação ou que afirma que o autor concorda com o documento. (OLIVEIRA, 2012)

Tatara (pág. 22, 2003) conceitua a assinatura digital como "uma sequência de bits que identifica o autor de um documento, garantindo também a integridade das informações contidas nele".

A assinatura digital é um sistema de Criptografia assimétrica, com função reversa. Sabe-se que duas pessoas podem trocar informações sigilosas utilizando um par de chaves, onde a origem (Alice) envia um texto criptografado utilizando a chave pública do destino (Beto), ou seja, mesmo usando um canal inseguro, apenas a origem (Beto) conseguiria decifrar a mensagem utilizando a chave privada. (WERLANC; MARTINS, 2010)

Na assinatura digital a autenticação é feita pela origem (Beto) que utiliza a chave privada para gerar um código garantindo a autenticidade da mensagem, dessa forma apenas quem possui a chave pública correspondente conseguirá identificar o código gerado, verificando a validade da assinatura. (CARVALHO, 2014)

Como somente com a chave privada do emissor é possível gerar o código da assinatura digital esse processo garante autenticidade, integridade e não repúdio da mensagem. (OLIVEIRA, 2012)

Autenticidade: Ao receber uma mensagem cifrada Beto utiliza a chave pública de Alice para decifrar a mensagem, somente Alice poderia ter assinado o documento pois somente ela é detentora da chave privada. (MENKE, 2003)

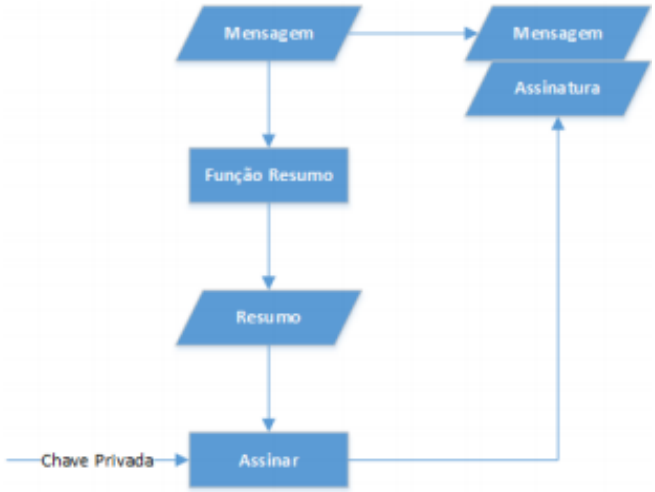
Não repúdio: Alice não poderá negar o envio do documento assinado, pois somente quem possui a chave privada conseguirá cifrar o código da assinatura digital, que foi decifrada por Beto utilizando a chave pública correspondente. (MENKE, 2003)

A assinatura digital que vai junto com mensagem que será codificada pode ser relativamente muito grande, dessa forma utiliza-se uma função que irá obter um resumo de tamanho pequeno, isso só é possível a partir da chave privada do emissor. Assim que o destinatário recebe a mensagem ele irá fazer validação da informação, o destino obtém um resumo da mensagem que será comparada com a assinatura decifrada utilizando a chave pública correspondente, se ambas forem iguais a assinatura é válida. (BECKER, 2013)

Em 1994, o NIST publicou o documento Digital Signature Standard, esse documento garante que os sistemas trabalhem em conjunto e é possível identificar os algoritmos de assinatura digital realmente seguros. Esse documento já recebeu revisões, onde foram adicionadas novas padronizações e incluídos novo algoritmo o ECDSA, baseado em curvas elípticas, antes apenas DSA e RSA estavam inclusos. (WERLANG; MARTINS 2010)

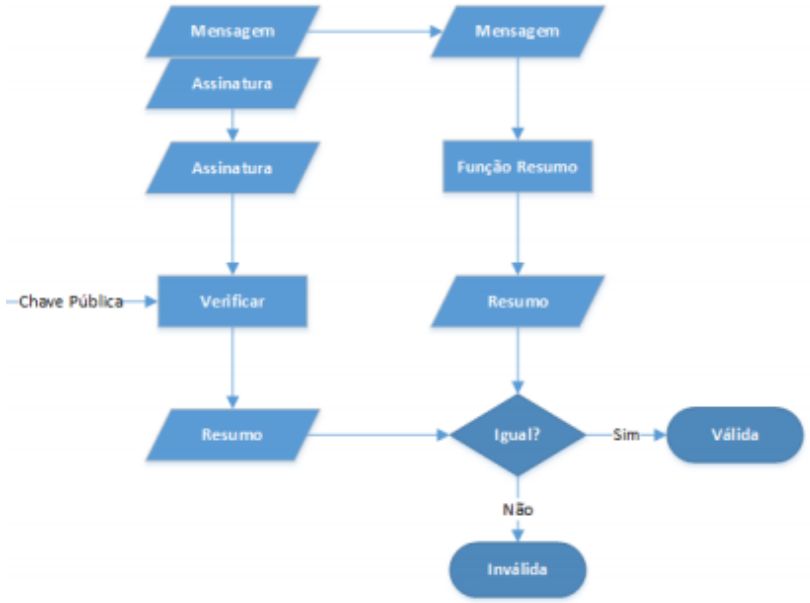
Adans (2003) descreve a assinatura digital da seguinte maneira, o assinante gera um resumo com código de tamanho fixo, essa operação é realizada a partir da chave privada. Já o processo de validação é feito quando um verificador gera um resumo da mensagem utilizado um valor de tamanho fixo, o verificador vai analisar a assinatura utilizando a chave pública correspondente ao par de chaves privadas que gerou a assinatura, assim a assinatura é validada, caso contrário há falha na verificação. (Figura 6 e 7)

Figura 6: Geração de Assinatura Digital



Fonte: BECKER, Gabriel Gaspar. HAWA-SISTEMA GERENCIADOR DE CERTIFICADOS DIGITAIS ONLINE. 2012.

Figura 7: Verificação da Assinatura digital



Fonte: BECKER, Gabriel Gaspar. HAWA-SISTEMA GERENCIADOR DE CERTIFICADOS DIGITAIS ONLINE. 2012.

O uso da criptografia assimétrica possui um custo computacional muito alto, por isso a realização da assinatura digital poderia levar muito tempo. Para solucionar esse problema e ao mesmo tempo garantir a integridade dos documentos passou-se a utilizar a função resumo. Em vez de assinar um documento os usuários assinam um hash, que é considerada equivalente a assinatura do documento correspondente. (MARTINS; WERLANG, 2010)

3.1 Função Resumo ou Hash

A assinatura digital é uma aplicação da criptografia assimétrica, na prática ela não pode ser empregada de forma isolada, é necessário estabelecer um mecanismo adequado para o emprego da assinatura. Esse mecanismo é a função hash. (OLIVEIRA,2012)

Costa (2006) define a função hash como um resultado de algoritmos que fazem um mapeamento de uma sequência de bits de tamanho fixo definido podendo ser de 128 bits, 160 bits e até 260 bits.

Os algoritmos hash são muito versáteis e podem ser utilizados em diversos algoritmos criptográficos e em protocolos de segurança como assinatura digital, verificação de integridade, autenticação de mensagens e até mesmo em senhas. (COSTA, 2006)

As funções hash são utilizadas em diversas áreas da computação, elas têm como objetivo garantir a integridade de arquivos e também são capazes de acelerar acesso a informações. (OLIVEIRA, 2012)

Além de acelerar processos computacionais, a função hash também pode ser utilizada em senhas, correios eletrônicos, criação de chaves criptografadas e até mesmo em transferência de arquivos. (LABORATORIES, 1996)

A verificação de integridade de arquivos será exemplificada por Alice que deseja armazenar um arquivo em seu servidor, e que se posteriormente desejar ter acesso a esse arquivo ela precisa que ele esteja íntegro. Alice então deverá calcular o hash do arquivo, envia para o servidor e guarda o hash em um local seguro. Mais tarde, Alice deseja acessar novamente o arquivo, para ter certeza que ele continua íntegro e que não sofreu nenhuma alteração ela precisará calcular novamente o hash, caso esse novo valor seja igual ao primeiro, significa que o arquivo não foi alterado, isso garante a integridade do arquivo. (SERAFIM, 2012)

Quando se trata da aplicação da função hash no armazenamento de senhas é necessário que o usuário realize o cadastro da senha em um sistema, o sistema então irá calcular o hash da senha e armazenará no banco de dados. Mais tarde, caso o usuário queira entrar novamente no sistema precisará usar a senha cadastrada, o sistema então, irá calcular o hash da senha inicial e comparar com o hash da senha utilizada, isso evita que intrusos queiram invadir o sistema. Esse método fornece uma boa segurança visto que a senha nunca fica armazenada no banco de dados e sim o hash da senha e este por sua vez é irreversível. (SERAFIM, 2012)

Uma das propriedades do hash é chamado de efeito avalanche. Essa propriedade foi descrita por Horst Feistel e consiste em que pequenas alterações na entrada causam grandes mudanças na saída, isso garante a integridade da informação, ou seja, se apenas um bit da mensagem original for alterado, o valor final do hash seria completamente diferente do valor original. (MOLDOVYAN; MOLDOVYAN, 2007)

Outra característica foi estabelecida por Serafin (2012), ele afirma que a função hash é unidirecional, ou seja é possível apenas calcular o hash, mas não é possível reverter o cálculo. Ou seja, uma vez calculado o hash é impossível, a partir dele, obter novamente a mensagem.

O mesmo autor demonstra outro aspecto interessante da função hash, que foi considerada uma das soluções para o grande problema da Criptografia assimétrica. Independentemente do tamanho ou formato da mensagem, o valor do hash sempre será de tamanho fixo que geralmente são de 128, 160 ou 256, bits. De forma prática, não importa se a mensagem gerar 360KB ou se chegar a 64GB, os valores do hash terão um valor fixo de, por exemplo, 128 bits. (SERAFIM, 2012)

Outra característica mencionada por Serafim (2012) é a resistência de colisões. É dito como colisão, quando o hash de duas mensagens tiver o mesmo resultado. Uma criptografia de hash deve ser resistente a esse tipo de ocorrência. Logo a resistência é definida não como a não ocorrência de colisões, mas a impossibilidade de encontrá-las.

Ferguson, Schneier (2003) atesta que quanto maior o valor da função hash mais difícil será a possibilidade de colisão, dessa forma é mais fácil encontrar colisões em hash de 128 bits do que em de 256 bits.

Principais funções hash

- MD4

O algoritmo hash MD4 foi desenvolvido em 1990 pelo matemático Ronald R. Rivest, a sogra MD significa Message digest, a tradução livre é mensagem resumida. Esse algoritmo foi divulgado oficialmente no artigo RFC 1186 e especificado no artigo RFC 1320. (MAZZETTO,2014)

- MD5

Em 1991 Ronald R. Rivest desenvolveu o Messenger Digest Algorithm 5, o matemático gerou uma função que é quase impossível de reverter, ou seja não é possível recriar o parâmetro de entrada utilizando somente o valor gerado pelo hash. (ALVES; AMÉRICO; JESUS, 2018)

O MD5 é um algoritmo unidirecional que produz hash de 128 bits para uma mensagem de entrada de tamanho arbitrário. Ele foi desenvolvido a partir de ataques feitos por criptoanalistas ao algoritmo MD4. (MAZZETTO, 2014)

O MD5 foi projetado para ser mais rápido e seguro do que o anterior. Algumas fraquezas foram encontradas, mas nada que comprometa a segurança global desse algoritmo, entretanto o fato dele produzir valores de apenas 128 bits é preocupante, pois, como vimos anteriormente, quanto maior o valor menor a chance de haver colisões de hash. (OLIVEIRA, 2012)

3.2 Certificado Digital

Alecrim (2005) define o certificado digital como um documento que garante a integridade e veracidade das informações contidas nele. Para isso, é necessário que contenha nome do usuário e sua chave pública, data de emissão e período de validade, além disso é necessário que a autoridade que emitiu o certificado assine o documento ligando oficialmente um usuário a sua chave pública. (Figura 8)

Figura 8: Conteúdo de um Certificado Digital

FONTE: MENEZES, Vinicius Corrêa et al. Gestão de dossiê pertencente ao titular de um Certificado Digital com garantia de integridade em uma Autoridade de Registro. 2018.

Becker (2013) descreve que o certificado digital tem como função verificar a autenticidade de um documento assinado, ou seja através do certificado digital é possível relacionar uma entidade a uma chave privada.

O uso do certificado digital tornou-se importante para garantir a segurança das informações que percorrem por grandes redes, como por exemplo transações comerciais eletrônicas ou quando o documento contém informações sigilosas como senhas. (FRIEDRICH E MEDINA, 2007)

O certificado digital foi criado por uma medida provisória 2.200-2 em agosto de 2001, é um documento eletrônico que permite assinar digitalmente, garantindo transações online com confiança, integridade e com validação jurídica. Pode ser requisitada por pessoas físicas e jurídicas que visam a troca de informações seguras através de canais inseguros como a internet. (RESENDE, 2009)

Atualmente há necessidade de realizar atividades na internet como transações bancárias, envios de e-mail, compras online, por isso a preocupação dos usuários em executar essas tarefas com segurança está crescendo, de modo a incentivar a prática dos certificados digitais. (RESENDE, 2009)

Uma das principais vantagens da utilização de certificados digitais é a redução de custos burocráticos, além de gastos com impressão e cartório, otimizando processos de assinatura sem perder as características de um documento tradicional. (MENEZES, 2018)

3.3 Lista de Certificados Revogados

A lista de certificados revogados (LCR) é uma lista contendo número de serial dos certificados que não são mais confiáveis. O motivo da inserção de um certificado a essa lista é vasto, a chave privada do dono do certificado pode ter sido comprometida, ou o emissor do certificado pode colocá-lo na LCR. (HOUSLEY, 2002)

Um certificado possui uma data de validade, porém em alguns casos os certificados perdem a validade antes dessa data pré definida, quando isso ocorre é necessário notificar outras entidades, além disso a Autoridade certificadora que emitiu o certificado publique o comprometimento do mesmo, os chamados mecanismos de revogação. (COOPER et al. 2008).

Cooper et al. (2008) considera a LCR a forma mais simples de revogar esses certificados, pois é um padrão bem estabelecido e disseminado.

Em 1999 Cooper criou dois mecanismos baseados nessas LCRs, com objetivo de reduzir a largura de banda necessário para atualizar a lista, e manter mais de uma LCR válida ao mesmo tempo, que é o mecanismo de LCRs sobrepostas, já a LCRs Segmentadas tem por objetivo reduzir o tamanho da lista dividindo ela em segmentos, porém neste método a dificuldade está em saber em qual segmento determinado certificado estará. (COOPER, 1999)

Goyal (2007) propôs a organização de uma LCR em conjuntos de certificados com número de séries consecutivos. Para revogar uma partição é necessário apenas incluir os primeiros números de série do segmento.

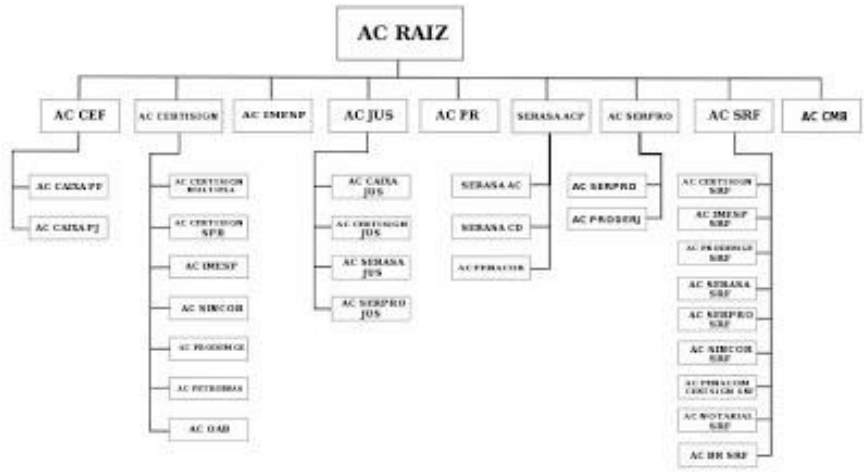
3.4 Autoridade Certificadora

A autoridade certificadora é a base de uma ICP, possui a função de emitir certificados e assinar os certificados, é responsável pela publicação das LCR, pública os certificados não

expirados, mantém os arquivos de informação sobre os certificados expirados ou revogados emitidos por ela. (SILVÉRIO, 2011)

As atribuições realizadas por uma AC são extensas e para garantir que todas as funções sejam cumpridas, houve a criação de uma hierarquia, ou seja, cada AC é responsável por um grupo de requerentes de certificados. A figura 9 evidencia, na prática, o funcionamento da hierarquia das ACs. A AC raiz é responsável por emitir os certificados apenas às autoridades certificadoras finais ou intermediárias e esta por sua vez, emitirá os certificados para os funcionários do setor que é responsável. (SILVÉRIO, 2011)

Figura 9: Hierarquia ICP-Brasil



Fonte: adaptado internet, disponível em: http://cryptoid.com.br/wp-content/uploads/2011/02/t15.jpg , acesso dia 07/04/2021

No Brasil, o governo federal estabeleceu a própria política de emissão de certificados e assinaturas digitais, criando a Infraestrutura de Chaves Públicas e Privadas, a chamada ICP-Brasil. Dessa forma somente os certificados emitidos por autoridades registradoras credenciadas pela ICP- Brasil têm validade jurídica reconhecida. (PECK, 2008)

Uma infraestrutura de chaves públicas é conceituada como um sistema que tem por finalidade atribuir certificados digitais para os usuários, podendo ser pessoas, entidades ou órgãos. A existência de uma ICP garante a comunicação entre os entes envolvidos, congregando um número maior de pessoas. (MENKE, 2011)

A estrutura hierárquica é uma das principais características da ICP- Brasil, a autoridade certificadora raiz da ICP-Brasil é o ITI (Instituto Nacional de Tecnologia e Informação) que tem como função credenciar outras autoridades certificadoras, e supervisioná-las. (RESENDE, 2009)

Em 2009 a ICP-Brasil coordenava oito ACs de primeiro nível que são, Presidência da República, Serasa, caixa econômica federal, Secretaria de Receita Federal, Serpro, AC Jus, Imprensa Oficial de São Paulo e Certisgn, 20 AC de segundo nível e mais de 800 autoridades registradoras. (RESENDE, 2009)

3.5 Autoridade Registro (AR)

"Essa entidade não tem função emitir certificados. Ela presta um serviço para a Autoridade Certificadora (AC), realizando todo o processo burocrático na emissão de um certificado digital. " (BECKER, pág.26, 2013)

Uma AR é uma entidade que verifica os dados do solicitante de um certificado digital, além disso é a autoridade registro que verifica pessoalmente a identidade do usuário e a conferência de seus dados. (MENEZES, 2018)

A autoridade de registro tem a função de verificar as informações de usuários que desejam obter um certificado, ou que já tenha um certificado e deseja revogá-lo. (SILVÉRIO, 2011)

Para que essas funções sejam cumpridas existe a atuação de Agentes Registros, que irão realizar a identificação do usuário, conferem os dados do mesmo e encaminham para as AC. Após a emissão do certificado os Agentes Registros entregarão pessoalmente aos solicitantes, a partir de então os solicitantes já poderão assinar documentos eletrônicos. (MENEZES, 2018)

4 ARQUITETURA PARA ASSINATURA DIGITAL

4.1 Introdução

Para facilitar o acesso dos usuários a uma assinatura digital, foi proposta uma arquitetura via aplicação web, onde as únicas ferramentas necessárias para seu uso, seriam apenas um computador com acesso à internet e um navegador.

A aplicação web foi desenvolvida utilizando o método de autenticação da conta google dos usuários, para reconhecimento e autenticação de quem vai utilizar o sistema. Após o login, o usuário terá a sua disposição as opções de assinar um documento ou verificar a autenticidade de um documento já assinado, necessitando apenas fazer o upload do documento na opção desejada e deixando o resto por encargo do sistema que já coleta os dados dos usuários através da conta google e gera um hash criptografado para cada documento, ao final disponibilizando o documento já assinado via download no caso da escolha da opção de assinar documento ou disponibilizando o Hash do documento já assinado no caso da verificação de autenticidade.

4.2 Tecnologias Utilizadas

4.2.1 Python

A principal e mais utilizada ferramenta para o desenvolvimento deste trabalho foi a linguagem python, que trata-se de uma linguagem de programação de alto nível, interpretada e interativa orientada a objetos.

A linguagem python foi criada por Guido Van Rossum, um matemático e programador de computadores holandês, que trabalhou no projeto no final dos anos 1980 no Instituto Nacional de Pesquisa de Matemática e Ciência da Computação na Holanda, ou Centrum voor Wiskunde en Informatica (CWI) como é conhecido em holandês. O trabalho do programador no CWI, se resumia em implementar uma linguagem de programação chamada ABC que buscava substituir a linguagem B (basic), uma das mais utilizadas da época. Guido Van Rossum estava buscando uma linguagem de script que fosse semelhante a linguagem ABC mas que ao mesmo tempo pudesse as chamadas de sistema de um Sistema Operacional chamado Amoeba (Andrew S. Tanenbaum), como não conseguiu encontrar, ele decidiu criar sua própria linguagem de script simples para atender suas necessidades.

Ao contrário do que muitos pensam, o nome da linguagem não deriva do gênero de cobras, apesar do logotipo da linguagem estampar uma cobra. O nome “python” é derivado de um programa de tv do canal BBC chamado “ Fly Circus de Monty Python”, o qual o Guido Van Rossum era muito fã.

Atualmente a linguagem Python está sendo continuamente desenvolvida de forma comunitária, aberta e gerenciada pela organização sem fins lucrativos Python Software Foundation.

4.2.2 Biblioteca PyPDF2

A PyPDF2 é uma biblioteca da linguagem python que foi utilizada para fazer as modificações nos arquivos PDF disponibilizados pelos usuários para assinatura.

A biblioteca PyPDF2 trata-se de uma ferramenta poderosa para manipulação de arquivos PDF, podendo extrair dados de arquivos PDF ou manipular (adicionar ou remover textos e páginas) PDFs existentes para produzir um novo arquivo.

Atualmente a biblioteca é mantida pela Phaseit, Inc e estão aceitando doações para manter o projeto.

4.2.3 Reportlab

A biblioteca Reportlab foi utilizada como mecanismo para adicionar estilo ao texto do pdf, como posicionamento do texto, espaçamento, caixas de texto e etc.

A Reportlab é uma biblioteca da linguagem python de código aberto e gratuita. Uma das bibliotecas mais utilizadas no mundo com mais de 50000 downloads por mês, utilizada no site wikipedia por exemplo.

As principais camadas do toolkit do reportlab utilizadas nesse trabalho, foram a Canvas e a PLATYPUS.

4.2.4 Biblioteca Crypto

A biblioteca pycrypto foi a ferramenta utilizada para gerar os algoritmos de criptografia do Hash gerado. Apesar da biblioteca pycrpto também possuir uma coleção de funções hash seguras, ela não foi utilizada para esse propósito. Foi utilizado o algoritmo de RSA para criptografar a função hash anteriormente gerada em função da mensagem. O objetivo central da biblioteca é oferecer uma interface simples e consistente onde todos os objetos de cifra de bloco têm os mesmos métodos e valores de retorno e suportam os mesmos modos de feedback.

4.2.5 Biblioteca Hashlib

Para gerar a função Hash da mensagem, foi utilizado o módulo Hashlib do python. Esse módulo utiliza o comprimento variável de bytes e os converte em uma sequência de comprimento fixo. Esta é uma função unilateral. Isso significa que é feito o hash de uma

mensagem e em seguida se obtém uma sequência de comprimento fixo. Mas não é possível obter a mensagem original dessas sequências de comprimento fixo, trazendo maior confiabilidade e robustez, e por esse motivo a biblioteca Hashlib foi escolhida para gerar a função hash neste trabalho.

4.2.6 Flask

Como a aplicação deveria funcionar via web para facilitar o seu uso, foi necessário utilizar uma ferramenta que viabilizasse esse processo. A ferramenta escolhida foi o framework Flask. Apesar de ser considerado um ‘micro’ framework por não necessitar de ferramentas ou bibliotecas próprias e justamente pelo fato de sua simplicidade a ferramenta ter sido escolhida, o Flask se mostrou poderoso durante o desenvolvimento do trabalho. O flask foi lançado no dia 1 de Abril de 2010 pelo 32ustríaco Armin Ronacher e reúne características simples que facilitam o seu uso, como um núcleo minimalista e expansível, permitindo que um projeto possua apenas os recursos necessários para sua execução, podendo ser adicionados novos pacotes posteriormente conforme necessário.

4.2.7 JavaScript

Outra linguagem de programação utilizada nesse trabalho foi a linguagem JavaScript. Muitas vezes abreviada de ‘JS’, é uma linguagem de script de alto nível estruturada e interpretada. Quase sempre utilizada com o HTML e o CSS, o JavaScript é uma das principais e mais utilizadas tecnologias para construção de aplicações web. O JavaScript é uma linguagem multiparadigma suportando estilos de programação orientados a eventos, funcionais e imperativos (incluindo orientado a objetos e prototype-based), apresentando recursos como fechamentos (closures) e funções de alta ordem comumente indisponíveis em linguagens populares como Java e C++. A linguagem foi lançada em 1995 pela Universidade de Illinois, no Centro Nacional de Aplicações de Supercomputação (NCSA).

4.2.8 Mysql

Como uma forma a mais de garantir a segurança das assinaturas, foi construído um Banco de Dados para guardar informações dos usuários que utilizaram o sistema de assinaturas e para tal foi construído um sistema de Banco de Dados utilizando o Mysql.

O Mysql é um dos sistemas de gerenciamento de banco de dados mais conhecidos e utilizados. Lançado em 23 de maio de 1995, o Mysql foi criado na Suécia por David Axmark, Allan Larsson e Michael "Monty" Widenius. Após ser comprada pela Sun Microsystems em

2008 pela cifra de US $1 bilhão (Valores recordes para época), a Oracle comprou a Sun Microsystems juntamente com todos seus produtos, sendo então a detentora do Mysql.

4.2.9 HTML

Como bloco de construção mais básico de aplicações web, a linguagem de Marcação de Hipertexto, mais conhecida como HTML, é fruto da junção entre os padrões HyTime e SGML.

Os Hipertextos podem ser conhecidos como os links que conectam as páginas web, gerando conexões externas ou internamente das páginas web. O HTML foi criado pelo físico britânico e cientista da computação Timothy John Berners-Lee em 1991 como uma ferramenta para resolver um problema de Tim, que era a comunicação e a disseminação das pesquisas desenvolvidas por ele e seu grupo de colegas.

4.3 Funcionalidades da Aplicação

A aplicação web oferece opções ao usuário de forma simples e intuitiva, para que funcione de forma clara e objetiva, facilitando e agilizando a vida do usuário na hora em que ele precisar assinar ou verificar a autenticidade de um documento.

É listado abaixo as funções disponíveis na aplicação:

- **Autenticação:** Ao acessar a aplicação web, as opções de assinatura só aparecem após o login do usuário no sistema. O login é feito através da conta google do usuário, que ao acessar o site, aparecerá a tela de autenticação google onde o usuário preencherá com seu email e senha e somente após o login a página é redirecionada para a tela principal da aplicação.

- **Assinar Documento:** Na tela principal da aplicação, o usuário terá a sua disposição a opção de “Assinar Documento” onde o mesmo poderá fazer o upload do documento desejado e o sistema assinará o documento e em seguida disponibilizará o documento fazendo o download automático já assinado com os dados do usuário retirados da sua conta google através do login, como nome, email, data, hora e o hash criptografado exclusivo daquele documento. Também aparecerá na tela, o hash não criptografado do documento para que o usuário possa confirmar a autenticidade daquele documento junto ao seu destinatário posteriormente.

- **Verificar Autenticidade:** Para garantir que um certo documento foi realmente assinado por um remetente, o usuário terá a opção de verificar a autenticidade daquele documento, onde na tela principal da aplicação, o usuário deverá apenas fazer o upload do documento e em seguida o sistema verificará o hash daquele documento. Ao final do processo,

aparecerá na tela em forma de texto o hash daquele documento para que o usuário possa verificar junto ao seu remetente se aquele hash é o mesmo do documento assinado anteriormente pelo remetente.

4.4 Modelagem Da Aplicação

Para modelagem da aplicação web, primeiramente foram levantados os requisitos do sistema e em seguida, foi utilizado a linguagem UML para o desenvolvimento de todos os diagramas.

A UML, é definida como Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software (Rumbaugh, Jacobson, Booch, Addison- Wesley, 2005).

A seguir temos a lista de requisitos levantados para construção do sistema:

- **Não funcionais:**

Multiplataforma: O sistema deverá ser executado em diferentes sistemas operacionais, como Windows e Linux e diferentes navegadores web. (Prioridade: essencial, Esforço: médio)

Robustez: O sistema deverá ter confiabilidade, funcionando de maneira estável, sem travamentos e paradas inesperadas. (Prioridade: essencial, Esforço: Alto)

Internet: Para atender a facilidade de acesso dos usuários, o sistema será uma aplicação web, sendo assim sendo obrigado o uso de internet para acessar o sistema. (Prioridade: Desejável, Esforço: Médio)

- **Funcionais:**

Fazer Login: O sistema deverá ter um método seguro para identificar seus usuários. (Prioridade: Essencial, Esforço: Médio)

Assinar documento: O usuário poderá escolher qualquer documento pdf do seu dispositivo. (Prioridade: Essencial, Esforço: Baixo)

Verificar Autenticidade: Deverá ter a opção para que usuário possa verificar a autenticidade de um documento já assinado. (Prioridade: Essencial, Esforço: Médio)

Segurança: O sistema deverá oferecer segurança quanto aos dados dos usuários e quanto ao documento, a garantir a inviolação do conteúdo do documento. (Prioridade: Essencial, Esforço: Alto)

Facilidade de acesso e usabilidade: Como objetivo principal, o sistema deverá ser de fácil acesso para os usuários, não necessitando de cadastros ou autenticações trabalhosas e deverá ser de fácil manuseio do ponto de vista dos usuários sendo intuitivo para que o usuário aprenda rapidamente a obter resultados com o sistema. (Prioridade: Desejável, Esforço: Médio)

Claro e Objetivo: O funcionamento do sistema deverá ser de tal forma que os usuários compreendam claramente os processos e os resultados do sistema. (Prioridade: Desejável, Esforço: Baixo)

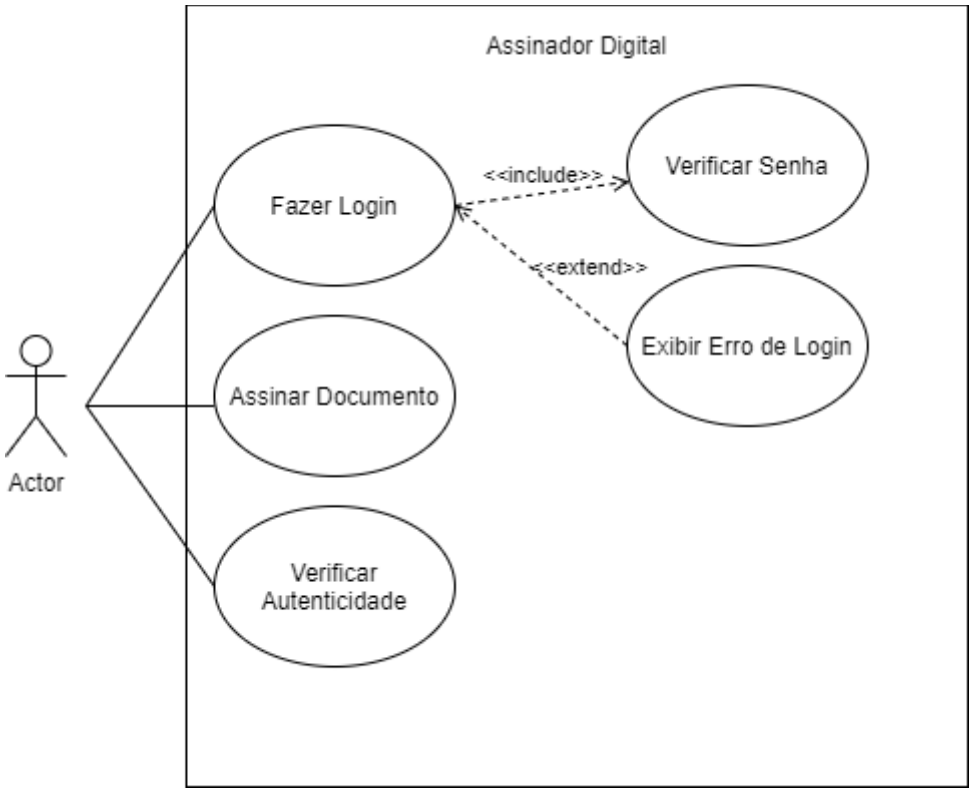
4.4.1 Diagramas UML

UC01 - Fazer Login: Este caso de uso permite que o usuário autentique sua identidade para acessar o ambiente web de assinatura caso seu e-mail e senha estejam corretos, caso contrário é apresentado uma mensagem de erro para que o usuário tente novamente efetuar sua autenticação;

UC02 - Assinar Documento: Este caso de uso permite que o usuário escolha seu documento e faça o upload para que o mesmo possa ser assinado digitalmente e disponibilizado para o usuário novamente, já assinado;

UC03 - Verificar Autenticidade: Este caso de uso permite que o usuário escolha um documento desejado e faça upload do mesmo no sistema para que possa ser verificada a autenticidade do documento, onde será mostrada na tela uma mensagem com o hash não criptografado daquele documento específico (Figura 10);

Figura 10-Diagrama Casos De Uso



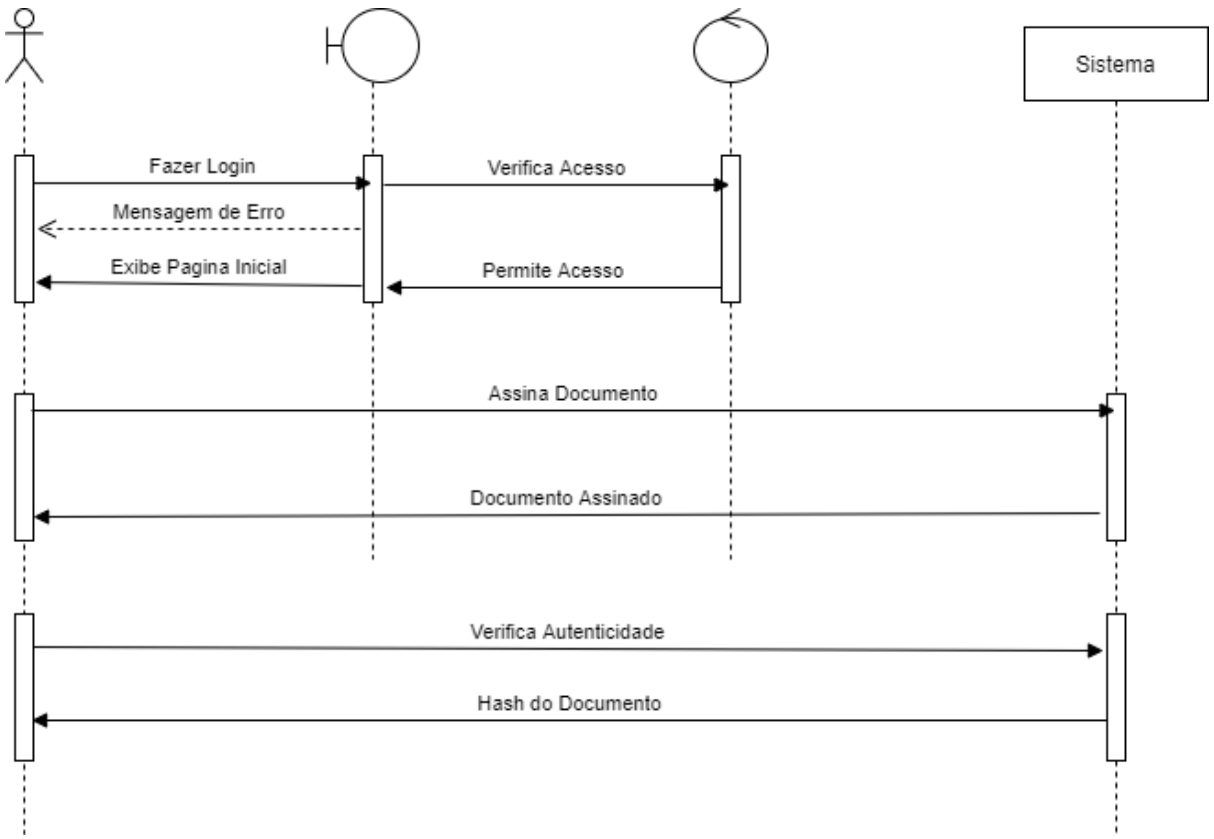
Fonte: Autor

4.4.2 Diagrama de Sequência:

O Diagrama de Sequência pode ser definido como um diagrama de interação cuja ênfase está na ordenação temporal das mensagens (Rumbaugh; Jacobson; Booch, Addison-Wesley, 2005); (figura 11)

No Diagrama é mostrado a sequência de eventos para assinar e para verificar a autenticidade de um documento. Primeiramente deve ser feito o login no sistema, onde o usuário só tem acesso às funcionalidades após a autenticação da sua identidade. Após a autenticação de identidade, o usuário tem acesso a opção de assinar um documento, onde primeiramente ele envia o documento escolhido para o sistema e posteriormente o sistema retorna o mesmo ao usuário. Da mesma forma sequencial, funciona a opção de verificar a autenticidade do documento.

Figura 11- Diagrama de Sequência

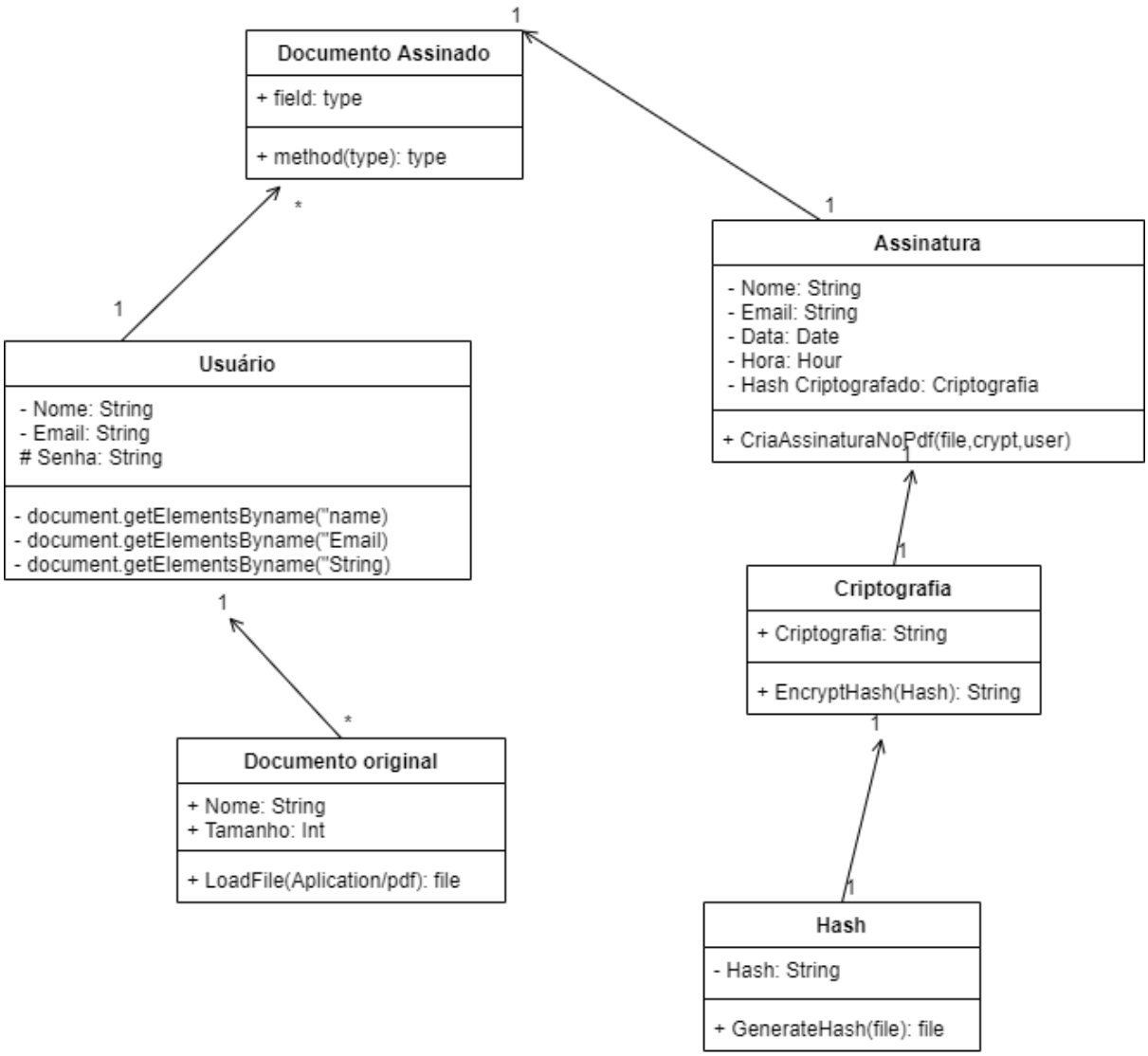


Fonte: Autor

4.4.3 Diagrama de Classes:

O Diagrama de Classes foi modelado de modo a orientar a implementação do sistema, organizando as classes e as partes do sistema, bem como seus componentes e suas relações, tornando a implementação bem direcionada e objetiva. (Figura 12)

Figura 12- Diagrama de Classes



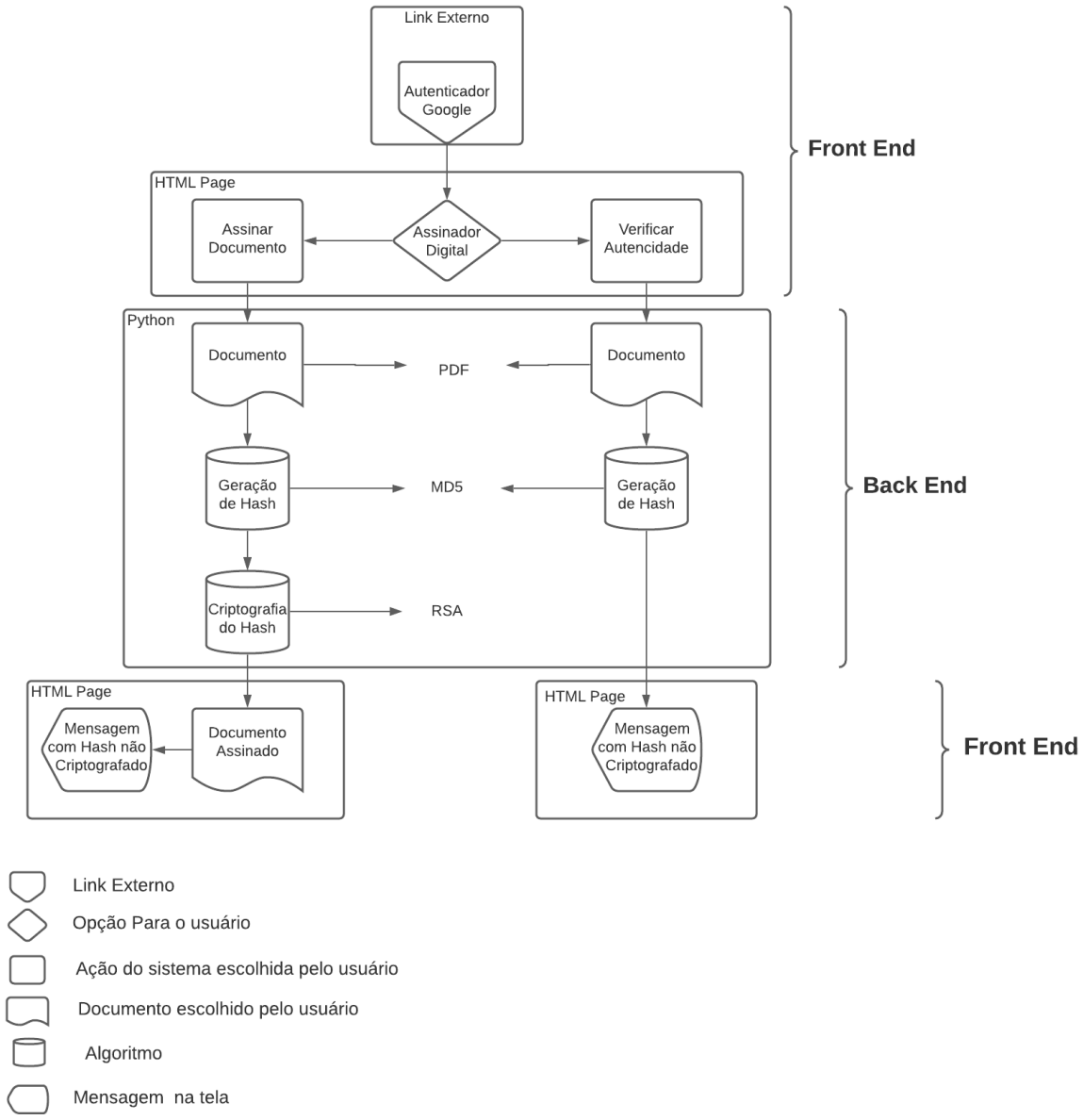
Fonte: Autor

4.5 Arquitetura Para Assinatura Digital

Como um dos papéis principais deste trabalho, foi desenvolvido uma proposta de arquitetura para um sistema de assinatura digital, utilizando as tecnologias citadas anteriormente.

A seguir temos um diagrama mostrando a estrutura dessa arquitetura (figura 13)

Figura 13-Diagrama de arquitetura para um sistema de assinatura digital



Fonte: Autor

O primeiro contato do usuário com o sistema se dá por meio de uma página *HTML*, onde é feita a autenticação de identidade através da sua conta google. É nessa página que o

sistema armazena os dados do usuário, como Nome e E-mail. A seguir temos o trecho de código de um script onde essas informações são coletadas:

```
<script>
  function onSignIn(response) {
        // Conseguindo as informações do seu usuário:
        var perfil = response.getBasicProfile();

        // Conseguindo o ID do Usuário
        var userID = perfil.getId();

        // Conseguindo o Nome do Usuário
        var userName = perfil.getName();

        // Conseguindo o E-mail do Usuário
        var userEmail = perfil.getEmail();

        // Conseguindo a URL da Foto do Perfil
        var userPicture = perfil.getImageUrl();

        document.getElementById('user-photo').src = userPicture;
        document.getElementById('user-name').innerText = userName;
        document.getElementById('user-email').innerText = userEmail;

        // Recebendo o TOKEN que você usará nas demais requisições à API:
        var LoR = response.getAuthResponse().id_token;
        console.log("~ le Tolkien: " + LoR);
        document.getElementsByName("id")[0].value = userID
        document.getElementsByName("name")[0].value = userName
        document.getElementsByName("email")[0].value = userEmail
        document.getElementsByName("image")[0].value = userPicture
        document.getElementById("redirect").submit()
    };
</script>
```

Script HTML da Página de Autenticação google

Ao efetuar a autenticação de identidade, o usuário tem acesso a tela principal do sistema, onde ele escolhe se deseja Assinar ou Verificar a Autenticidade de um documento fazendo o upload do documento. O upload do documento é feito por meio de dois formulários diferentes, correspondentes a cada opção:

```
<center> <font color="fff">
        <h1> <font face="Arial"> <font color="#fff">Assinador Digital</font> </font> <br /> </h1>

        <h2><font face="Arial"> <font color="#fff">Olá</font> </font> <br />
                <div><font face="Arial"><font color="#fff">{{name}}, assine seu documento aqui:</font> </font> <br /></div>
        </h2>
        <form enctype="multipart/form-data" method="POST" action="/uploadpdf" id="redirect">
                <input class="button formbackgound" id="signfile" type="file" name="file" accept="application/pdf">
                <input type="hidden" name="name" value="{{name}}">
                <input type="hidden" name="email" value="{{email}}">
        </form>

        <h2><font face="Arial"><font color="#fff">Verifique a autenticidade do seu documento aqui:</font> </font> <br /></h2>
        <form enctype="multipart/form-data" method="POST" action="/verifypdf" id="redirect2">
                <input class="button formbackgound" id="fileverify" type="file" name="file" accept="application/pdf">
        </form>
        </font>
</center>
```

Script HTML para Upload de Documento

Ao fazer o upload do documento, entra em funcionamento a parte do *Back End* do sistema escrito em python, onde será gerado o *hash* deste documento e logo em seguida, é feita a criptografia desse *hash* para ser anexado ao documento.

Para gerar o *hash* do documento, foi utilizado o algoritmo *MD5*, que na linguagem *python* pode ser utilizado através da biblioteca *hashlib*:

```
#Gera o Hash para um arquivo
def HashData(data):
    Hash=md5(str(data).encode('utf-8')).hexdigest()
    return Hash

#gera o Hash resultante do PDF
def HashPdf(file,IgnoreLastPage = 0):
    existing_pdf = PdfFileReader(open(file, "rb"))
    #Criando o PDF resultante
    output = PdfFileWriter()
    #Copia as pags do pdf original para o pdf resultante
    for i in range(0,existing_pdf.getNumPages()-IgnoreLastPage):
        output.addPage(existing_pdf.getPage(i))
    outputStream = open(UPLOAD_FOLDER + "StreamCript.pdf" , "wb")
    output.write(outputStream)
    outputStream.close()
    f=open(UPLOAD_FOLDER + "StreamCript.pdf" ,"rb")
    data=f.read()
    f.close()
    return HashData(data)
```

Função de Geração do *Hash*

Com o *hash* gerado, o sistema passa para o módulo de criptografia, onde é feito a criptografia do *hash* gerado através do algoritmo *RSA*. No *python*, isso pode ser feito através da biblioteca *Crypto*:

```
    #Função de encriptação
def encrypt_blob(blob, public_key):
    #Importa a Chave Publica e usa para encriptação usando PKCS1_OAEP
    rsa_key = RSA.importKey(public_key)
    rsa_key = PKCS1_OAEP.new(rsa_key)

    #comprime o dado primeiro
    blob = zlib.compress(blob)
    #Ao determinar o tamanho do bloco, determine o comprimento da chave privada usada em bytes
    #e subtrai 42 bytes (ao usar PKCS1_OAEP). Os dados serão criptografados
    #em blocos
    chunk_size = 470
    offset = 0
    end_loop = False
    encrypted = bytearray()

    while not end_loop:
        #O bloco
        chunk = blob[offset:offset + chunk_size]

        #Se o bloco de dados for menor que o tamanho do bloco, então precisamos adicionar
        #um preenchimento com "". Isso indica que chegamos ao final do arquivo
        #então terminamos o loop aqui
        if len(chunk) % chunk_size != 0:
            end_loop = True
            #chunk += b" " * (chunk_size - len(chunk))
            chunk += bytes(chunk_size - len(chunk))
        #Anexa o fragmento criptografado ao arquivo criptografado geral
        encrypted += rsa_key.encrypt(chunk)

        #Aumenta o deslocamento pelo tamanho do bloco
        offset += chunk_size

    #Codifica o arquivo criptografado em Base 64
    return base64.b64encode(encrypted)

     #Criptografa o Hash salvando cada chave para cada usuário
     def encrypt(hash, UserId):

          private_key = Path("UsersKeys/" + str(UserId) + '_private.pem')
          public_key = Path("UsersKeys/" + str(UserId) + '_public.pem')

          encrypted_msg = encrypt_blob(str.encode(hash), public_key.read_bytes())
          decrypted_msg = decrypt_blob(encrypted_msg, private_key.read_bytes())
          print (decrypted_msg)

          return encrypted_msg
```

Função de Criptografia do *Hash*

Ao final de todo o processo, é gerada a assinatura no documento, onde são reunidas todas as informações do usuário, juntamente com o *hash* Criptografado para compor a página de assinatura utilizando a biblioteca *Reportlab* para fazer as alterações no *PDF* e disponibilizado o documento resultante para o usuário por meio de download automático:

```
def CriarAssinatura(filepath,name, email,Hash):
    #Cria um PDF com Assinatura
    packet = io.BytesIO()
    # cria um novo PDF usando a bib Reportlab
    can = canvas.Canvas(packet, pagesize=A4)
    can.drawString(200, 800, "Assinado Digitalmente por:")
    can.drawString(200, 780, name)
    can.drawString(200, 770, email)
    can.drawString(200, 755, time.strftime('%d/%m/%Y às %H:%M:%S', time.localtime()))
```

Função de Criação de Assinatura no Documento

Na opção de Verificar autenticidade, após o upload da documentação, o Back End do Sistema faz a descriptografia do hash e apresenta como mensagem na tela para o usuário. A função responsável pelo processo de decriptografia é demonstrada abaixo:

```
#Função de Decriptação
def decrypt_blob(encrypted_blob, private_key):

    #Importa a Chave Publica e usa para decriptação usando PKCS1_OAEP
    rsakey = RSA.importKey(private_key)
    rsakey = PKCS1_OAEP.new(rsakey)

    #Decodifica da Base 64
    encrypted_blob = base64.b64decode(encrypted_blob)

    #Ao determinar o tamanho do bloco, determina-se o comprimento da chave privada usada, em bytes.
    #Os dados serão descriptografados em blocos
    chunk_size = 512
    offset = 0
    decrypted = bytearray()
```

Função de Decriptografia

E ao final, é mostrado em uma página HTML para o usuário em forma de mensagem na tela, o hash daquele documento:

```
<div class="box1">
        <center>
        <h1><font face="Arial"> <font color="#fff">Assinador Digital</font> </font> <br/></h1>
        <h2><font color="#fff">Olá, Esse é código de verificação do seu arquivo. Verifique junto ao seu remetente para garantir a autenticidade do arquivo:
                <div><font face="Arial"> <font color="#fff">{{hash}}</font> </font> <br/></div>
        </h2>
        </center>
```

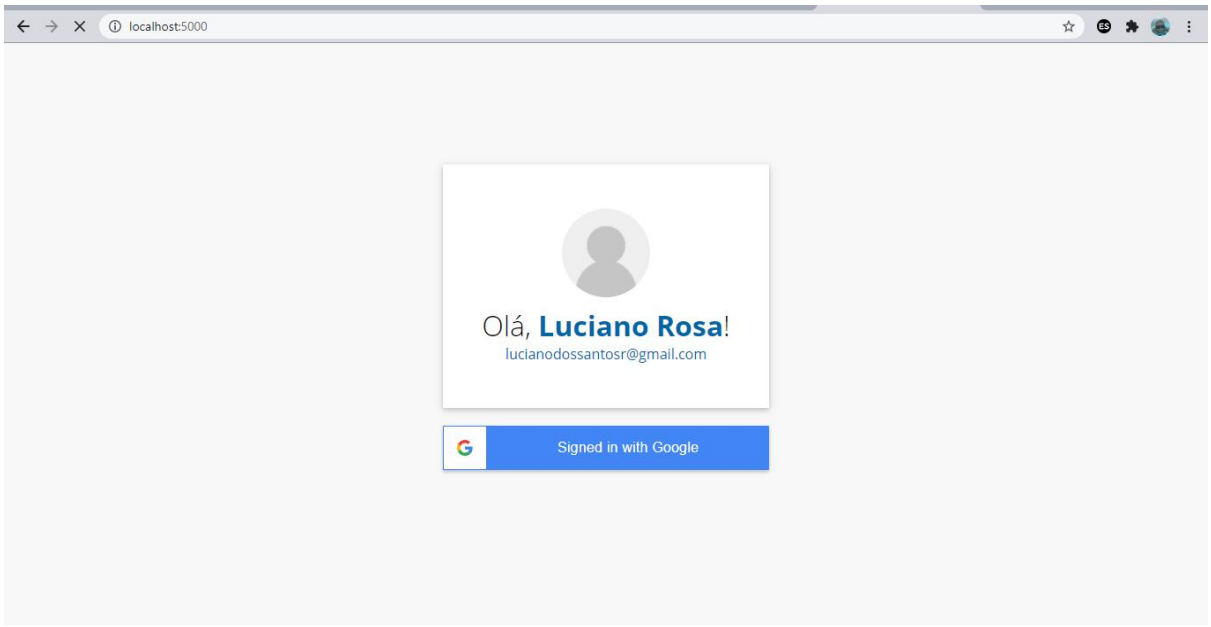
Script da Página de Verificar Autenticidade

4.6 Interfaces Externas

Nesta seção, serão mostradas as interfaces do sistema onde o usuário fará toda a interação com a aplicação web.

A primeira tela a ser mostrada ao acessar a aplicação, é a tela de login google, onde o usuário entra com sua conta para autenticar sua identidade e ter acesso à tela principal. (Figura 14)

Figura 14- Tela de Autenticação de Identidade



Fonte: Autor

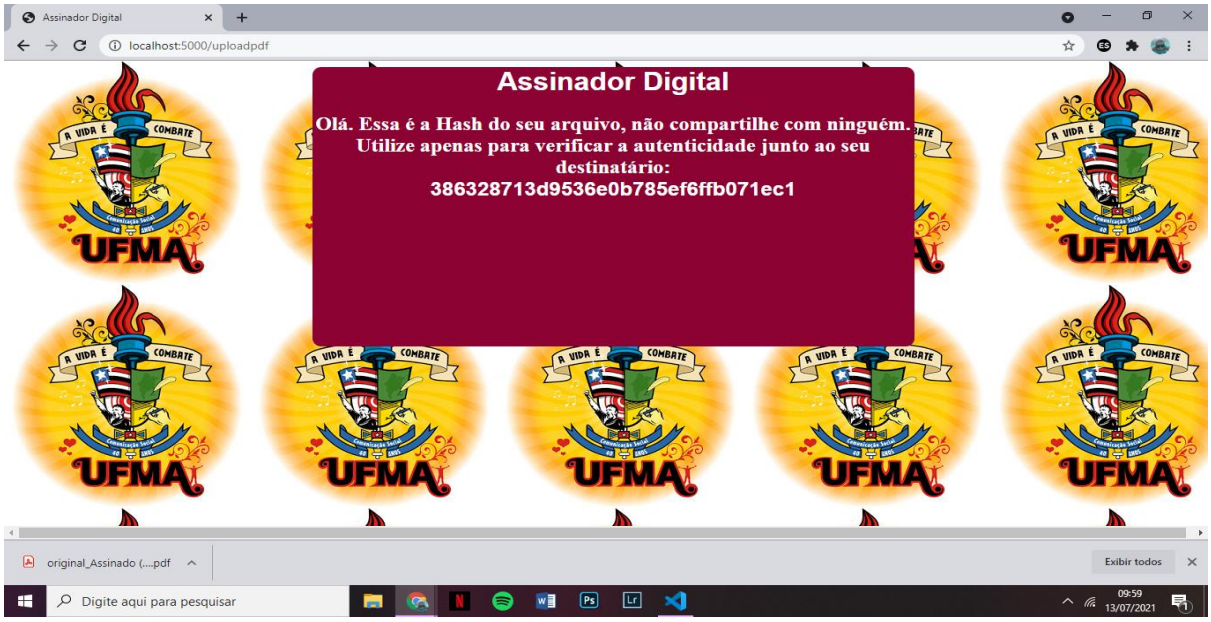
Em seguida, o usuário terá acesso a tela principal da aplicação, onde são apresentadas as opções que o sistema oferece. (Figura 15)

Figura 15- tela principal da aplicação

Fonte: Autor

Para escolher a opção de Assinar Documentação, o usuário deverá apenas fazer o upload do documento desejado no local indicado e ao final do processo, o sistema irá fazer o download automático do documento já assinado com os dados do usuário juntamente com a mensagem na tela indicando qual o hash daquele arquivo. (Figura 16)

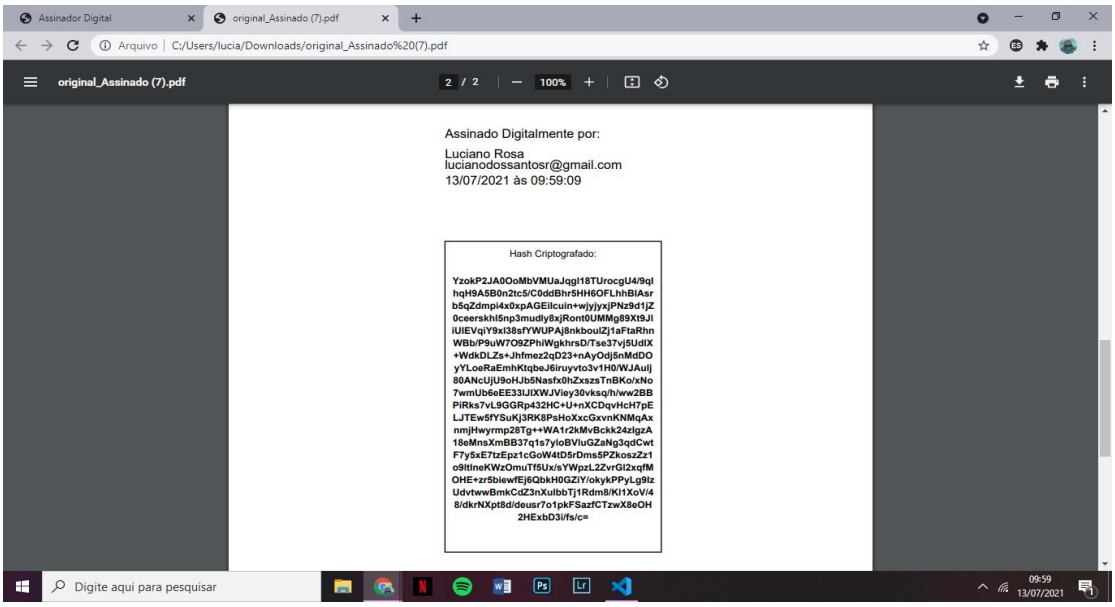
Figura 16- Tela da Opção de Assinar Documento



Fonte: Autor

Ao executar o documento gerado, na última página do documento, irá constar o Nome, e-mail, data, hora da assinatura e o hash Criptografado. (Figura 17)

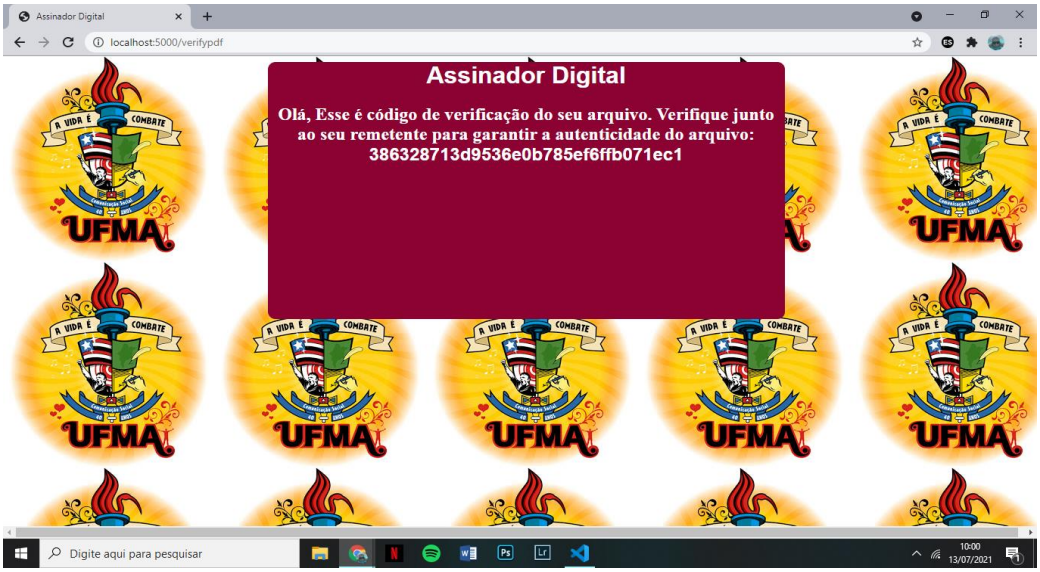
Figura 17- Documento Assinado Digitalmente



Fonte: Autor

Já ao escolher a opção de Verificar Autenticidade e fazer o upload do documento no local indicado, a aplicação redireciona para outra página web onde será mostrado na tela o hash daquele arquivo escolhido (Figura 18)

Figura 18- Tela da Verificação de Autenticidade



Fonte: Autor

A aplicação demonstrou excelente performance, funcionando muito bem, estável e sem travamentos. Cumprindo com todos os requisitos especificados na modelagem e projeto. Garantindo todos os aspectos de segurança e robustez.

5 CONCLUSÃO

Este trabalho abordou os conceitos de criptografia e como esse conceito foi importante para garantir a segurança das informações durante a história da humanidade e como ele pode ser utilizado nos dias atuais para garantir os princípios da segurança digital.

A assinatura digital apesar de suas enormes vantagens e comodidade, ainda é pouco conhecida e utilizada pela maioria dos usuários dos meios digitais. Pouco conhecimento esse que muitas vezes gera um sentimento de desconfiança por meio dos usuários comuns, que acabam não utilizando e não confiando na segurança das assinaturas digitais, deixando de usufruir da comodidade, rapidez e facilidade na hora de autenticar documentações.

Este trabalho tem por objetivo implantar um sistema de assinatura digital de modo simples, prático, rápido e de fácil acesso e utilização do ponto de vista da experiência dos usuários, visando facilitar o processo e garantir a segurança e eficiência na assinatura de documentos digitais bem como difundir os conceitos da Assinatura Digital de modo a quebrar o paradigma e a desconfiança que ainda se tem sobre a Assinatura Digital. Destaca-se que tal aplicação não foi testada com usuários e por consequência, não foi possível apresentar resultados práticos de métricas de avaliação.

O objetivo do trabalho foi alcançado, resultando em uma aplicação simples, prática, ágil e de fácil acesso, que demonstrou funcionamento estável, alcançando todos os objetivos propostos pela modelagem e planejamento do mesmo. Foi um trabalho complicado de ser desenvolvido mas ao mesmo tempo muito prazeroso por se tratar de uma ferramenta extremamente útil.

Como trabalhos futuros, podemos citar a intenção de disponibilizar a aplicação em um servidor para acesso livre dos usuários e implementar outras formas de autenticação para reforçar a segurança de todo o processo.

REFERÊNCIAS

ADAMS, C.; LLOYD, S. Understanding PKI: Concepts, Standards, and Deployment Considerations. [S.l.]: Addison-Wesley, 2003.
Alecrim, Emerson. **Assinatura digital e certificação digital**. InfoWester, 2005. Disponível em: http://www.infowester.com/assincertdigital.php

ALVES, Gabriel Rodolfo Teodoro; AMERICO, Patrick Araujo; Jesus, Wagner Santos. **Autenticação de Certificados Emitidos em Eventos Usando Algoritmo Message-Digest algorithm 5**

BARBOSA, Luis Alberto De Moraes; BRAGUETTO, Luis Fernando B.; BRISQUI, Marcelo Lotierso; SILVA, Sirlei Cristina Da. RSA – **Criptografia Assimétrica e Assinatura Digital**, UNICAMP, julho, 2003.

BRAGA, Alexandre; DAHAB, Ricardo. **Criptografia Assimétrica para Programadores–Evi-tando Outros Maus Usos da Criptografia em Siste-mas de Software**. Caderno de minicursos do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais—SBSeg, v. 2018, p. 1-50, 2018.

BECKER, Gabriel Gaspar. HAWA-**SISTEMA GERENCIADOR DE CERTIFICADOS DIGITAIS ONLINE**. 2012.

BURNETT, STEVEN; PAINE, Stephen**. Criptografia e segurança: o guia oficial RSA. Gulf Professional** Publishing, 2002.
CAMPELLO, A. C.; LEAL, I. **Teoria Aritmética dos Números e Criptografia RSA**. Monografia, IME-UNICAMP, 2007.

CARVALHO, Sérgio Luiz de. **Autoridades certificadoras e segurança na assinatura digital**. 2014.

CAVALCANTE, André LB. Teoria dos números e criptografia. **Revista Virtual**, 2005.

COSTA, C. **Introdução à criptografia**. Rio de Janeiro: Centro de Estudos de Pessoal, 2010. V. 1.

Cooper, D. (1999). **A model of certificate revocation**. In Computer Security Applications Conference (ACSAC ’99) Proceedings. 15th Annual pages 256 –264, Phoenix, USA

Cooper, D. (2000). **A more efficient use of delta-crls**. InProceedings of the 2000 IEEE Symposium on Security and Privacy (SP’00), pages 190 –202, Washington, DC, USA

COUTINHO, S. C. **Número inteiros e criptografia RSA**. 2. Ed. Rio de Janeiro: IMPA, 2014.

COUTINHO, S. C. **Criptografia**. Rio de Janeiro: IMPA, 2016.

DE COMPUTAÇÃO, Curso de Engenharia. **Criptografia**. 2006. Tese de Doutorado. Universidade do Vale do Paraíba. (Teixeira,2006).

DE LA ROCHA LADEIRA, Ricardo; RAUGUST, Anderson Schwede. Uma análise da complexidade do algoritmo RSA implementado com o teste probabilístico de Miller-Rabin. **Revista de Empreendedorismo**, Inovação e Tecnologia, v. 4, n. 1, p. 24-33, 2017

ELLIS, Claire. **Exploring the Enigma**. Plus magazine: 2005. Disponível em: http://plus.maths.org/content/os/issue34/features/ellis/index

FIGUEIREDO, Luiz Manoel. **Introdução à Criptografia**. Fundação CECIERJ. Rio de Janeiro: UFF/CEP-EB, v. 2, 2010.

FIGUEIREDO, Rafael C.; GADIS RIBEIRO, Vinícius. **Uma proposta de emprego de smart cards em infra-estrutura de chave pública**. In: X Congreso Argentino de Ciencias de la Computación. 2004.

FREIRE, Paloma Barbosa; CASTILHO, José Eduardo. **A matemática dos códigos criptográficos**. Universidade Católica de Brasília, 2005.

FREITAS NETO, Natanael de. **Esquemas de assinaturas digitais: o uso de criptografia assimétrica como um método técnico para autenticidade e não-repúdio em emissão de laudos médicos remotos para PACS**. 2018. Dissertação de Mestrado. Brasil.

FRIEDRICH, Diego Mostardeiro; MEDINA, Roseclea Duarte. **Certificação Digital Acadêmica: Implantação do Sistema de Gerenciamento de Certificados Digitais ICPEDU na UFSM. RENOTE**, v. 5, n. 2, 2007.

FERGUSON, SCHENEIER, Pratical Criptograph. 1 eedição, Wiley, 2003

Goyal, V. (2007). **Certificate revocation using fine grained certificate space partitioning**. In Financial Cryptography and Data Security (FC'07), pages 247–259. Springer Berlin Heidelberg, Scarborough, Trinidad and Tobago.

GROENWALD, Claudia Lisete Oliveira; FRANKE, Rosvita Fuelber; OLGIN, Clarissa de Assis. **Códigos e senhas no Ensino Básico**. Educação Matemática em Revista–RS, p. 41-50, 2009.

HOUSLEY, R. et al. **Certificate and Certificate Revocation List (CRL) Profile**. 2002.

JASPER, Nicholas Aron. História, Técnica e classificação dos algoritmos esteganográficos, 2009

LEONG, Monk-Ping et al. **Uma implementação bit-serial do algoritmo de criptografia de dados internacional IDEA**. In: Proceedings 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No. PR00871) . IEEE, 2000. P. 122-131.

MENKE, Fabiano. Assinaturas Digitais, certificados digitais, infra-estrutura de chaves públicas brasileira e a ICP alemã. **Revista de Direito do Consumidor**, v. 12, n. 48, p. 17, 2003.

MAIA, Luiz Paulo; PAGLIUSE, Paulo Sergio. **Criptografia e Certificação Digital**, 2000.

MARQUES, Thiago Valentim. **Criptografia: abordagem histórica, protocolo Diffie-Hellman e aplicações em sala de aula**. 2013

MAZZETTO, Muriel. **FUNÇÃO HASH CRIPTOGRAFADA (MD5, E A FAMÍLIA SHA**). Universidade Tecnológica Federal do Paraná. ICCEEg, 2014.

MENEZES, Vinicius Corrêa et al. **Gestão de dossiê pertencente ao titular de um Certificado Digital com garantia de integridade em uma Autoridade de Registro**. 2018.

MELO PIRES DE, Rayner. **Aplicaçao do Algoritmo Diffie-Hellman no Compartilhamento de Volumes Criptografados do TrueCrypt**. (2010)

MORAES, ROSANE FRANÇA DE. **Construção de um ambiente web com ferramentas para estudo de algoritm os de criptografia através do Matlab**, Escola de Engenharia, Departamento de Eletrônica, Universidade Federal do Rio de Janeiro, junho, 2004.

MORAIS[1], F. M. F.; NORONHA, I. C. P. **UMA ABORDAGEM HISTÓRICA, EVOLUTIVA E APLICACIONAL DA CRIPTOGRAFIA.**

NOGUEIRA, Ana Flávia Cesário Machado Alckmin**. Proposta de atividades usando Criptografia nas aulas de matemática**. 2017.

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

OLIVEIRA, Vinícius Sousa de. **Estudo e aplicação de algoritmos criptográficos para redes de sensores sem fio em um ambiente de rádio definido por software**. 2021. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte.

ORDONEZ, E.; PEREIRA, F.; CHIARAMONTE, R. **Criptografia em Software e Hardware**. 1st edition. Ed. São Paulo: Novatec, 2005. ISBN 85-7522-069-1.

PECK, Patricia. **Direito Digital**. 2. Ed, Editora Saraiva, São Paulo, 2008

PEREIRA, Nádia Marques Ikeda. **Criptografia: uma nova proposta de ensino de matemática no ciclo básico**. 2015.
PETRI, Marcelo. Esteganografia, Joinville, dez. 2004.

PIRES, Rayner M.; COSTA, Vaston G. **Uso do algoritmo Diffie-Hellman na geração de keyfile para o TrueCrypt**. (2010)

PROF. RIBEIRO, VINICIUS GADIS. **Esquemas de chave pública**. Porto Alegre, maio, 2002

QUARESMA, Pedro; LOPES, Elsa. 2 **O Surgimento da Criptografia**.

RESENDE, Dilma A. Certificação digital. **Revista Jurídica UNIGRAN**, v. 11, n. 22, p. 111, 2009

SANTOS, EDUARDO. **Criptografia – Parte 2**, setembro, 2005. Disponível em: http://www.imasters.com.br/artigo.php?cn=3577&cc=210 .

DA SILVEIRA SERAFIM, Vinícius**. Introdução à Criptografia: Algoritmos de Chaves Assimétricas**.2012

SERAFIM, Vinícius. **Introdução à Criptografia: Função criptográficas de hash** .2012

DA SILVA, Bruno et al. **CRIPFTOGRAFIA ASSIMÉTRICA DE IMAGENS UTILIZANDO ALGORITMO RSA**. 2013.

SILVA, Mariana Godoy; OLIVEIRA, Cintia Carvalho. **O USO DA CRIPTOGRAFIA EM ÁUDIO**. Anais do Seminário de Pesquisa e Inovação Tecnológica-SEPIT, v. 1, n. 1, 2017.

SILVA, Bruno de Melo. **Uma abordagem de infra-estrutura de chaves públicas para ambientes corporativos**. 2004

Silverio, A., Kohler, J., and Cust ´ odio, R. (2011). **Analise e implementacão de um protocolo de gerenciamento de certificados**. In WTICG – SBSeg 2011.

Schneier, B. "**Applied Cryptography: Protocols, Algorithms, and Source Code in C"**, John Wiley & Sons Inc., 1996.

SINGH, S. **The code book: the secret history of codes and code-breaking**. United Kingdom: Fourth Estate, 1999

SINGH, Simon. **O Livro dos Códigos**. Rio de Janeiro: Record, 2002.

SINGH, S. **O Livro dos Codigos**. São Paulo, SP: Editora Record, 2011. 446 p.

TATARA, Carlos Francisco. **S2Card**. Trabalho de Conclusão de Curso de Graduação apresentado á Universidade Federal de Santa Catarina. 2003

TEIXEIRA, Marco Antonio Fávaro. **Números inteiros e criptografia RSA**. 2020.

WAZLAWICK, R. S. **História da Computação**. 1º edição. Ed. Rio de Janeiro: Elsevier, 2016. 584 p. ISBN 9788535285451.

WEBER, R. F. **Criptografia Contemporânea**. Congresso Brasileiro de Computação. Canela: 1995.

WERLANG, Felipe Carlos; MARTINS, Lucas Gonçalves. **Sistema de Gerenciamento de Certificados Offline**. 2010.