



Universidade de Aveiro

Ano 2022/2023

**António Jorge
Ferreira Ramos**

**Validação da Autenticidade de Documentos
Impressos**



Universidade de Aveiro
Ano 2022/2023

**António Jorge
Ferreira Ramos**

Validação da Autenticidade de Documentos Impressos

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor André Zúquete, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e com a colaboração do Engenheiro Luís Amorim, supervisor empresarial

o júri / the jury

presidente / president

Prof. Doutor Paulo Jorge Salvador Serra Ferreira
professor associado da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Carlos Nuno da Cruz Ribeiro
professor associado da Universidade de Lisboa – Instituto Superior Técnico (arguente principal)

Prof. Doutor André Ventura da Cruz Marnoto Zúquete
professor auxiliar da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

Quero agradecer desde já aos meus pais e irmã por me darem apoio e força para nunca desistir.

Não posso deixar de agradecer aos meus Orientadores André Zúquete e Luís Amorim pelo suporte e ideias para desenvolver a dissertação.

palavras-chave

Marca de Água, Códigos de Barras, Documentos Impressos, Documentos Digitais, Documentos Eletrónicos, Segurança, Integridade, Autenticidade, Segmentos de Reta

resumo

A informação é um bem essencial para um bom funcionamento de um país e dentro dela existem matérias classificadas, cuja informação, pode envolver política, saúde, militar e nuclear. Existindo inimigos do país é claro que estes querem de fazer de tudo para aceder a essas matérias. Para evitar que os documentos caem em mãos erradas é necessário criar um método seguro que garanta a sua proteção.

A presente dissertação aborda o problema e possíveis soluções para documentos classificados, que após serem impressos num posto de controlo, podem ser apresentados a terceiros inapropriados ou terceiros credenciados que não têm maneira de validar a autenticidade do documento.

Para resolver o problema do acesso a pessoas terceiras foram estudados vários métodos. Métodos de marca de água visíveis e invisíveis que permitam validar a integridade do documento e outros que permitam a validação de documentos com base nos metadados e informação do documento. Optou-se por aprofundar e desenvolver testes com base na segunda fase de métodos estudados.

O método escolhido foi criar pontos para dar origem a segmentos de retas invisíveis no documento, cujo objetivo é calcular a interseção e guardar o ponto caso esta calhe num conjunto de intervalo de posições de uma letra.

A validação é feita através da adição de letras e pontos ao documento digitalizado para a respetiva comparação, onde a leitura da marca é feita através de um código de barras adicionada no documento.

Através de testes de robustez conseguiu-se determinar a zona de alteração caso estas calhem nas letras guardadas, levando a concluir que se conseguiu proteger os documentos, sem que o hacker ou pessoa terceira saiba onde pode alterar.

keywords

Watermark, Barcodes, Printed documents, Digital Documents, Electronic Documents, Security, Integrity, Authenticity, Line Segments

abstract

Knowledge and information are essential for a country's well functioning and the information can be about politics, health, the military and nuclear. Being that, inevitably, there are enemies to any country logically these will try their best to gain access to the information that's classified. In order to ensure that the high-classified information is kept safe and doesn't fall into the wrong hands, there must be a method put in use to guarantee their safety.

The following thesis speaks on the issue and the potential solutions for classified documents that, after being printed and handed to a third party either credentialed or not allowed to see the document, lose their credibility therefore the authenticity of the document can not be proven.

To solve this third party issue of possible breach various methods were studied. Methods of watermark (visible or not) that are used to insure and validate the credibility of the document and others that allow to verify documents based on the metadata and the information of said document.

The second method mentioned was the one chosen to further analyze and perform tests.

The chosen method creates points in order to create invisible line segments in the document, this is done with the objective of saving the intersection point in case it falls in an section of positions it corresponds to a letter in the document.

The validation is processed by adding letters and points to the digitised document, so the comparison is made and the way its read is through a barcode previously added to the document.

By doing the robust tests, we were able to determine the area of alteration in case these fall into saved letters, leading to believe that the documents are now protected, without the hacker or a third person knows

Índice

ÍNDICE.....	II
LISTA DE TABELAS	III
LISTA DE FIGURAS.....	IV
1 INTRODUÇÃO.....	1
1.1 ESTRUTURA DO DOCUMENTO	1
2 CONTEXTO	3
2.1 CÓDIGO DE BARRAS	3
2.1.1 <i>Análise de Código de Barras.....</i>	<i>3</i>
2.2 PROCESSO DE CRIAÇÃO DA META-INFORMAÇÃO	8
2.2.1 <i>Obtenção das letras e respectivas posições num documento PDF</i>	<i>8</i>
3 TRABALHOS RELACIONADOS	9
3.1 MARCA DE ÁGUA.....	9
3.2 MARCAS DE ÁGUA VISÍVEIS	9
3.2.1 <i>Substituição de Palavras.....</i>	<i>9</i>
3.2.2 <i>Marca de Água sem Mudança de Texto</i>	<i>11</i>
3.2.3 <i>Documentos baseados em Eigenvalues.....</i>	<i>12</i>
3.2.4 <i>Marca de Água com Espaçamento.....</i>	<i>14</i>
4 ALGORITMO DE MARCAÇÃO E ANÁLISE DE DOCUMENTOS IMPRESSOS.....	16
4.1 PROCESSAMENTO DE DOCUMENTOS.....	16
4.2 VERIFICAÇÃO DE DOCUMENTOS IMPRESSOS	17
5 IMPLEMENTAÇÃO	18
5.1 LINGUAGENS DE PROGRAMAÇÃO	18
5.2 ARQUITETURA	18
5.3 DOCUMENTOS.....	19
5.3.1 <i>Conversão para Bitmap</i>	<i>19</i>
5.3.2 <i>Obter Dimensões do PDF e da Imagem.....</i>	<i>20</i>
5.3.3 <i>Criação dos Pontos para o Sistema de Verificação.....</i>	<i>20</i>
5.3.4 <i>Criação do Documento para a Integridade.....</i>	<i>23</i>
5.4 BASE DE DADOS	24
5.5 OBTENÇÃO DAS LETRAS E RESPECTIVAS POSIÇÕES NUM DOCUMENTO PDF	25
5.6 CÓDIGO DE BARRAS.....	27
5.6.1 <i>Criação e Leitura.....</i>	<i>27</i>
5.6.2 <i>Posições</i>	<i>28</i>
5.7 INTERFACE GRÁFICA (GUI).....	31
6 TESTES.....	35
6.1 DOCUMENTOS DIGITAIS COM ROTAÇÕES	35
6.2 MUDANÇA DE ESCALA NO DOCUMENTO	39
6.3 MODIFICAÇÕES NO DOCUMENTO	45
7 CONCLUSÕES.....	51
7.1 TRABALHO FUTURO	51
8 REFERÊNCIAS.....	53
9 ANEXO	54
A) MUDANÇA DE ESCALA DO DOCUMENTO	54

Lista de Tabelas

Tabela 1 - Código de Barras Lineares.....	4
Tabela 2 - Lista de Códigos de Barras Ideias	5
Tabela 3 - Classificação de Erros do QR Code	6
Tabela 4 - Resultados Eigenvalues	14

Lista de Figuras

Figura 1 - Exemplo QR Code.....	6
Figura 2 - Estrutura QR Code.....	6
Figura 3 - Código de Barra 128.....	7
Figura 4 - Dimensões PDF.....	8
Figura 5 - Exemplo da Substituição de Texto	10
Figura 6 - Antes da Marca de Água.....	13
Figura 7 - Depois da Marca de Água.....	13
Figura 8 - Comparação Marca de água	13
Figura 9 - Exemplo Line-Shift	14
Figura 10 - Exemplo Word-Shift.....	15
Figura 11 - Exemplo Character Coding	15
Figura 12 - Diagrama de Fluxo do Processamento de um Documento	16
Figura 13 - Representação dos Valores dos Caracteres.....	16
Figura 14 - Diagrama de Fluxo da Averiguação de um Documento.....	17
Figura 15 - Arquitetura Solução	19
Figura 16 - Código para criar uma Imagem da primeira Página do PDF	19
Figura 17 - Código para obter as dimensões do documento PDF e da Imagem Convertida.....	20
Figura 18 - Código para a criação dos pontos aleatórios da Marca de Água.....	21
Figura 19 - Representação dos Pontos	22
Figura 20 - Espaçamento dos Pontos	22
Figura 21 - Imagem Auxiliar para visualização das Letras e Pontos	23
Figura 22 - Diagrama da Base de Dados.....	24
Figura 23 - Criação do utilizador para a base de dados.....	25
Figura 24 - Código para Abrir o Documento e Extrair Informação	25
Figura 25 - Exemplo de uma Letra e Respetiva posição no PDF com a Utilização da Ferramenta PDF-XChange Editor	26
Figura 26 - Valor Lido pelo Algoritmo Representado na Base de Dados	26
Figura 27 - Criação dos Códigos de Barras 128 e 39.....	27
Figura 28 - Obtenção de Posições do Código de Barras	29
Figura 29 - Pontos obtidos dos Códigos de Barras	30
Figura 30 - Menu.....	32
Figura 31 - Menu Processar	32
Figura 32 - Conclusão do Processamento	33
Figura 33 - Verificação da Autenticidade de um Documento	33
Figura 34 - Visualização da marca	34
Figura 35 - Zoom retângulo	34
Figura 36 - Documento Digitalizado com Rotação.....	36
Figura 37 - Documento digitalizado após correção	37
Figura 38 - Controlo de integridade sobre um documento corrigido após digitalização defeituosa	38
Figura 39 - Apresentação dos elementos de controlo de integridade sobre um documento original.....	39
Figura 40 - Documento Original com 50% de Escala.....	40
Figura 41 - Documento Original	41
Figura 42 - Apresentação dos elementos de controlo de integridade sobre um documento original (2)	42
Figura 43 - Documento Original com 80% de Escala.....	43
Figura 44 - Apresentação dos elementos de controlo de integridade sobre um documento original com 80% de escala	44
Figura 45 - Documento Original (2).....	46
Figura 46 - Apresentação dos elementos de controlo de integridade sobre um documento original (3)	47
Figura 47 - Zoom da palavra "importante".....	47
Figura 48 - Controlo de integridade sobre um documento original com substituição de palavra	48
Figura 49 - Zoom da palavra "necessário"	48
Figura 50 - Controlo de integridade sobre um documento original com eliminação do parágrafo	49
Figura 51 - Zoom sobre o parágrafo removido.....	49
Figura 52 - Controlo de integridade sobre um documento original com eliminação de espaçamento	50
Figura 53 - Zoom da eliminação de espaçamento.....	50
Figura 54 - Alteração da Escala do Documento	54
Figura 55 - Obtenção da Escala do Documento	54
Figura 56 - Adaptar Posições	55

1 Introdução

Cada país tem os seus próprios segredos e informações valiosas tal como militar, político ou saúde, que não querem que sejam divulgadas a terceiros, ou a inimigos. Como diz Francis Bacon “*Conhecimento é poder*” [1], isto propõe que existem pessoas com saciedade para conhecer o desconhecido do inimigo, daí que haja roubo, destruição, acesso a matérias classificadas. Porventura, estas carecem de proteção especial para evitar fugas. Esta proteção pressupõe que a organização responsável por garantir a segurança das matérias classificadas do país, tenha conhecimento da posseção do material classificado, do lugar geográfico do material, em que lugar deveria se encontrar e quem tem ou teve acesso ao mesmo. Como referido anteriormente é necessário prevenir acesso indevido a terceiros, bem como possibilitar a investigação de incidentes ou falhas de segurança. Em Portugal existe a solução GESMAC (Gestão de Matéria Classificada) [2], que cobre o ciclo de vida de uma matéria classificada, desde à sua criação à destruição da mesma, onde o objetivo principal é implementar mecanismos de segurança e controlo capaz de garantir a sua proteção.

A presente dissertação aborda o problema e possíveis soluções para documentos classificados, que após serem impressos num posto de controlo, podem ser apresentados a terceiros inapropriados ou terceiros credenciados, levando ao surgimento de mecanismos capazes de validar a integridade dos documentos.

A presente solução assume que os documentos originais estão em formato PDF, cujo se pode obter as letras e respetivas posições. Os documentos de formato PDF podem conter ficheiros digitalizados, ou seja, ficheiros impressos, que através de um leitor ótico é convertido para imagem e este porventura PDF.

A solução proposta usa os metadados e a metainformação do documento, onde a marca utilizada engloba a utilização de letras obtidas através da interseção de segmentos de retas criados aleatoriamente na primeira página do documento. Esta informação é guardada em tabelas na base de dados, que através de um identificador para a mesma, irá ser colocado no código de barras, onde este está inserido no cabeçalho e rodapé da folha. Isto permite a validação rápida de documentos impressos através da digitalização dos mesmos para depois leitura no programa, onde as letras irão ser colocadas no ponto de interseção (com outra cor), para que apenas os utilizadores de confiança consigam comparar informações e ter acesso à marca original do documento.

Os testes realizados, em documentos de formato eletrónico e digital, mostram que a marca consegue detetar mudanças estruturais no documento e mudança textuais dentro das palavras que contenham a marca. Também se mostrou a adaptação das posições em ficheiros digitais e com mudança de escala do documento.

1.1 Estrutura do documento

- Capítulo 1 - Introdução: Neste capítulo é feita uma breve descrição de matérias classificadas, os objetivos para o projeto, a solução e resultados obtidos;
- Capítulo 2 - Contexto: Neste capítulo aborda-se os tipos de códigos de barras e identificação de letras e posições específicas de um documento PDF;
- Capítulo 3 - Trabalhos Relacionados: Neste capítulo faz-se um estudo de métodos de marca de água visíveis e invisíveis;

- Capítulo 4 - Algoritmo de Marcação e Análise de Documentos Impressos: Neste capítulo explica-se o funcionamento do algoritmo;
- Capítulo 5 - Implementação: Neste capítulo explica-se como é que a solução foi implementada ;
- Capítulo 6 - Testes: Neste capítulo são feitos testes ao algoritmo em documentos exemplares, nomeadamente mudança de texto e documentos digitais com rotações;
- Capítulo 7 - Conclusões e Futuro Trabalho: Neste capítulo são retiradas conclusões do trabalho desenvolvido e apresentam-se sugestões para trabalho futuro;

2 Contexto

Neste capítulo irão abordar-se dois temas fundamentais para o desenvolvimento do sistema, sendo eles divididos em subsecções sendo elas códigos de barras e identificação de letras em posições específicas de um PDF.

2.1 Código de Barras

Uma das formas de esconder informação, é através dos códigos de barras, onde é possível ler informações através das leituras dos mesmos. Usando o código de barras é possível esconder informação através de um identificador para uma base de dados externa que permite um acesso único à visualização da marca aos utilizadores de confiança. Existindo inúmeros e diversos dos códigos de barras é necessário agregar as informações para escolher os ideias, surgindo esta secção.

O código de barras é um modo de representar informação num estado visual, sem ter a necessidade de escrever texto. Classifica-se em duas categorias, linear (1D) e 2D.

Esta informação é apresentada na Tabela 1 onde os critérios escolhidos para as comparações são as seguintes:

- Nome: designação do código de barras;
- Imagens exemplificativas, onde é possível ver como o código de barras é, ou seja, a sua imagem;
- Comprimento: variável ou fixo, essencial para que todos os documentos tenham o mesmo cumprimento do código de barras, sendo que este não é fator predominante;
- Uso: verificar onde se utiliza, para ver se é ideal para colocar num documento;

Em alguns códigos de barras existe um espaço reservado que não se pode alterar titulado *checksum* ou *check digit*, cujo objetivo é verificar se a informação do código de barras foi gerada corretamente.

Existem várias maneiras de descodificar códigos de barras, salientando-se as seguintes:







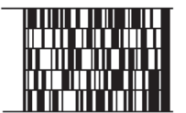








1. Máquina própria (leitor de barras de barras)
2. Smartphone (iOS ou android) com uma aplicação que permita a leitura.

2.1.1 Análise de Código de Barras

Através da análise da Tabela 1, verificou-se que o Post Code Australia, o Intelligent Mail Barcode, o Planet, o PostBar, o PostNet, o RM4SCC/KIX, o RM MailMark C e Code 49 não se enquadravam devido à sua tipologia, por terem tamanho de barras diferentes, o que impede o controlo da altura, o que poderá influenciar na descodificação do código de barras no documento, caso esta calhe em cima de texto. Os restantes enquadram na escolha, só que o barcode 39, 128 e 93 são os mais usados devido à sua capacidade de armazenamento, o que levou à criação da Tabela 2, onde se compara os mesmos, escolhendo o ideal.

Os critérios selecionados foram o tamanho máximo de caracteres que este pode guardar, tipos de dados que consegue codificar, vantagens e desvantagens do mesmo. Como o código de barras 128 permite codificar caracteres especiais, letras e números e a sua capacidade de dados é grande, optou-se por utilizar este, adicionando-o ao rodapé do documento, sendo que o código de barras 39 também seria uma possível escolha devido ao pouco espaço ocupado que iria ocupar no documento.

Tabela 1 - Código de Barras Lineares

Nome	Imagem	Fixo/ Variável	Tamanho das barras	Uso
Post Code Austrália	 12229111A B A 9	Fixo	4	Correios Austrália
CodaBar	 3 1117 01320 6375	Fixo	2	Librarias
Code 25	 0 1 2 3 4 5 6 7 8 9	Variável	2	Librarias
Code 11	 0123452	Fixo	2	Telefones, já não se usa muito devido à sua antiguidade
Code 32	 A012345678	Fixo	2	Farmácia
Code 39	 CODE39	Fixo	2	Vários
Code 49		Variável	Vários	Vários
Code 93	 CODE93	Variável	Vários	Vários
Code 128	 CODE128	Variável	Vários	Vários
EAN 2	 9 770517 847001 05	Variável	Vários	Revistas
EAN 5	 9 781565 924798 54495	Variável	Vários	Livros
EAN-8	 <9 63 8 15 0 7 4 >	Variável	Vários	Retalho
GS1-128	 (01)9901234567890(1180)000123	Variável	Vários	Vários
GS1 Databar	 0010007567816425	Variável	Vários	Vários
Intelligent Mail	 Wikimedia Foundation Inc. PO BOX 78350 SAN FRANCISCO CA 94107-8350	Fixo	4	Correios USA















barcode				
ITF-14		Variável	2	Encomendas
ITF-6		Variável	2	Vários
JAN		Variável	Vários	Usado no Japão
Planet		Variável	Grande/ Pequeno	Correios USA
Plessey		Variável	2	Catálogos, revistas, inventários
PostBar		Fixo	4	Correios Canada
PostNET		Fixo	Grande/ Pequeno	Correios USA
RM4SCC/K IX	 Lloyds TSB Bank plc CREDIT CARD SERVICES BOX 1 BX1 1LT	Fixo	4	Correios
RM Mailmark C		Fixo	4	Correios
Universal Product Code		Variável	Vários	Retalho
Telepen		Variável	2	Librarias da Inglaterra

Tabela 2 - Lista de Códigos de Barras Ideias

Nome	Imagem	Tamanho máximo de dados	Tipos de dados	Vantagens	Desvantagens
Code 39		43	Letras e números	Ocupa menos espaço.	Não permite caracteres especiais tal como ç.
Code 128		48	Letras, números e caracteres especiais	É mais eficiente a codificar texto. Tem <i>checksum</i>	Precisa de espaço reservado no início, fim e <i>check symbols</i> .
Code 93		30	Letras, números e caracteres especiais	Open Source	Não tem checksum

2.1.1.1 QR Code

O QR Code surgiu para localizar os segmentos de reta criados para a criação da marca, já que este é perceptível e contém pontos de referência como por exemplo os quadrados de detecção de posição. O surgimento desta secção adveio da explicação do funcionamento do QR Code para a sua utilização.

O QR Code (Figura 1) surgiu em 1994 pela empresa Denso Wave, originalmente com o propósito de categorizar peças de automóveis. Os QR Code podem ter links para páginas web, texto, um endereço geográfico, uma imagem, um vídeo ou contacto telefónico.

É formado por:

- 3 quadrados de detecção de posição (4.1. Figura 2) que permite a descodificação em várias posições do scanner ou de um smartphone com câmara;
- padrão alinhamento (4.2. Figura 2) que corrige a distorção do QR Code em superfícies curvadas, o seu número varia consoante a informação contida;
- padrões de temporização (4.3. Figura 2) que permite obter o tamanho da matriz de dados;
 - versão, que indica a versão do QR Code que está a ser utilizada (1. Figura 2);
- formato, onde agrega informações sobre a tolerância de erros e o padrão da máscara de dados;
- códigos de dados e erros que podem ser do tipo L, M, Q, H (apresentados na Tabela 3).



Figura 1 - Exemplo QR Code

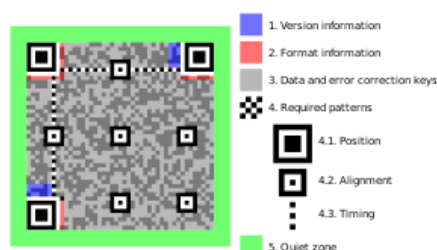


Figura 2 - Estrutura QR Code

Tabela 3 - Classificação de Erros do QR Code

Nível de correção de erros	Percentagem de área danificada (%)
L (Low) – Baixo	7
M (Medium) – Médio	15
Q (Quartile) - Quartil	25
H (High) – Alto	30

2.1.1.2 Código de Barras 128

O código de barras 128 (Figura 3) [3] é um tipo de código de barras linear que pode ser usado para codificar uma grande variedade de dados alfanuméricos, incluindo letras, números e caracteres especiais. É capaz de codificar 128 caracteres da tabela ASCII.

Este é muito utilizado em aplicações de logística, como no controlo de stocks e na identificação de produtos em supermercados e lojas. Isso ocorre porque ele é capaz de codificar informações como o nome do produto, o seu número de série, o código de barras do fabricante e outras informações relevantes num único código.

É constituído por barras largas e compactas que representam cada caracter, com barras de início e fim que indicam onde começa e termina a sequência de caracteres. Este é capaz de codificar uma abundância de informação num espaço relativamente pequeno, o que o torna uma ferramenta eficiente e económica para a gestão de inventário e outras aplicações similares.

É um código de barras universalmente reconhecido e amplamente utilizado em todo o mundo, o que leva a ser uma opção confiável para empresas que necessitam de um sistema de identificação e rastreamento de produtos rápido e preciso.

Como referido inicialmente, o intuito da dissertação é validação de documentos impressos, para isso é necessário guardar informações acerca do documento em algum lado, para a averiguação do mesmo. Como o código de barras consegue agregar informação, como, por exemplo, um identificador único de um documento onde por fora esteja uma base de dados que contenha informação do documento, optou-se por utilizar como medida de análise inserindo-o no rodapé ou cabeçalho do documento.



Figura 3 - Código de Barra 128

2.2 Processo de criação da metainformação

Um dos processos importantes é a criação do sistema de verificação da integridade de um documento que consiste em utilizar um conjunto de letras resultantes da interseção de segmentos de retas aleatórios criados invisivelmente na primeira página do documento original. Este método é interessante, devido a guardar certas letras de várias posições dos documentos, e este método garante que para o mesmo ficheiro exista múltiplas marcas, isto permite assim uma maior proteção ao documento e diminuir as probabilidades de obtenção do funcionamento da criação da marca pois este aleatório. Um objeto aleatório é um escolhido com base num conjunto de intervalo, quanto maior for o intervalo, mais possibilidades de escolha se tem, por exemplo, tem-se os intervalos de 1 a 10, a probabilidade de o utilizador terceiro acertar no número escolhido é $1/10$, e caso o intervalo fosse 1 a 100 era $1/100$ levando a inferir a probabilidade da escolha de $1/n$, onde n é a quantidade de números a escolher. A grande pergunta está como isto funciona?, para explicar vai se abordar a obtenção das coordenadas e letras do documento PDF.

2.2.1 Obtenção das letras e respetivas posições num documento PDF

Num documento PDF com um conjunto de metainformação, consegue-se extrair ou procurar informação nesse documento, levando à possibilidade de extrair letras e posição do mesmo. Sendo que através de leitores PDF não é possível a sua extração, levando à necessidade de criação ou utilização de código para a extração de texto e respetivas posições.

Um documento PDF de dimensão A4 tem as dimensões 595x842 *points* [4], podendo variar em ficheiros digitalizados ou outros, tendo a posição das letras aptada às dimensões do mesmo, sendo que as respetivas letras se encontram nesse intervalo (decimal). As dimensões do ficheiro estão representadas na Figura 4, onde as setas significam a adição dos *points*. As letras ocupam espaço no documento, ou seja, nunca só ocupam um point, o que leva a criação de um intervalo por cada letra dividindo-se em começo e fim da letra, onde estas se subdividem em x e y, perfazendo um intervalo de 4 valores.

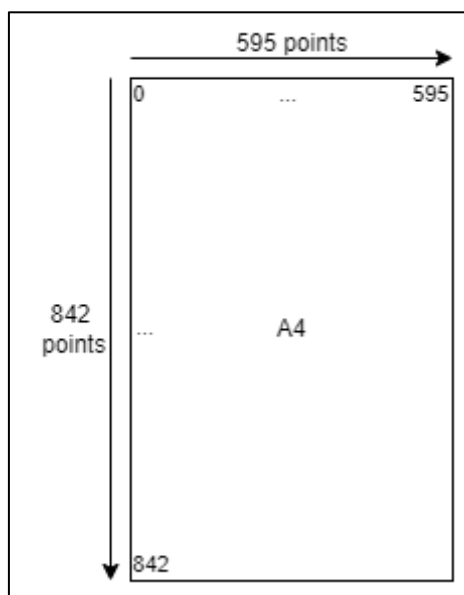


Figura 4 - Dimensões PDF

3 Trabalhos Relacionados

Neste capítulo irão ser abordados conjunto de trabalhos relacionados com a adição da marca de água num texto, sendo dividido em visíveis e invisíveis. Mas antes disso é necessário abordar o que é uma marca de água e porque a sua escolha para estudar métodos de validação em matérias classificadas.

3.1 Marca de Água

A marca de água é um método que permite dificultar a alteração do conteúdo de matérias classificadas criando métodos que garantam a segurança das mesmas, dividindo-se em visíveis e invisíveis. Sendo que as visíveis contêm métodos capazes de verificar a informação das matérias classificado de uma forma implícita. Por outro lado as invisíveis, constituem métodos invisíveis a olho nu que garantem que estas não se alterem.

3.2 Marcas de Água Visíveis

3.2.1 Substituição de Palavras

O documento quando está em circulação em formato papel ou eletrónico pode ser alvo de fuga de informação ou acesso ao documento por pessoas não autorizadas que conseguem alterar o texto (inserir, reordenar, substituir e remover), levando à mudança de conteúdo e da estrutura do texto do mesmo.

A solução proposta pelos autores do artigo [5], envolve focar a marca de água na localização das palavras no texto. Contudo a reordenação do texto é o maior desafio para esta marca de água sobre o texto. A marca de água sobre o texto proposta envolve seleccionar certas palavras no documento e substituir por novas, mantendo o resto da informação intacta no final da criação da mesma. Esta implementação é constituída por dois elementos:

1. Lista de palavras que se pretende substituir no documento.
2. Lista de palavras contidas no documento depois da substituição.

Em suma, escolhe-se um conjunto de palavras no documento que se quer substituir por novas. Quando uma palavra é seleccionada para se substituir todas as ocorrências da mesma são substituídas pela palavra nova. A percentagem do ataque determina a quantidade de modificação do documento. Se a palavra escolhida tiver várias ocorrências no documento, a percentagem vai aumentar.

A percentagem da contagem de palavras no documento é dada pela Equação 1. A percentagem do ataque de substituição é dada pela expressão da Equação 2.

A lista de substituição pode ser escolhida através de uma lista de palavras normal e avançada. A normal consiste em seleccionar palavras aleatórias do documento. A avançada selecciona palavras com o mesmo tamanho que as palavras da lista de substituição.

Em suma, a marcação de água textual proposto por os autores do artigo [5] consiste em três passos, seleccionar palavras do documento para substituição, seleccionar as novas palavras e ocorrência de implementação no texto do documento.

$$WOR(x) = \frac{\text{Number of Occurrence of } x \text{ in Document}}{\text{Total Number of Words in Document}}$$

Equação 1 - Contagem de palavras

$$WSR = \frac{\sum WOR(x)}{\text{Total Number of Words in Document}}$$

Equação 2 - Percentagem de ataque

3.2.1.1 Exemplo

Conforme mencionado no artigo [5] a Figura 5 apresenta um exemplo de categorias de substituição que podem ser usadas para mascarar informações importantes num texto. O exemplo mostra a substituição da palavra "The" (sendo o pronome mais comum em inglês), por "close" na substituição normal e por "job" na substituição avançada.

Este método pode ser útil em várias situações, como na proteção de informações sensíveis em documentos, comunicações confidenciais, entre outros. No entanto, é importante notar que este método não é infalível.

<p>Original Text:</p> <p>The historic art and architecture of Venice is the key to the city's popularity as a tourist destination, but every two years, from May to October, it also becomes a place of pilgrimage for contemporary art lovers.</p>
<p>Normal Replacement Attack:</p> <p>Close historic art and architecture of Venice is close tech to close city's popularity as a building destination, but every two years, from May to October, it also becomes a defence of pilgrimage for contemporary art lovers.</p>
<p>Advanced Replacement Attack:</p> <p>Job historic art and architecture of Venice is job run to job city's popularity as a brokers destination, but every two years, from May to October, it also becomes a there of pilgrimage for contemporary art lovers.</p>

Figura 5 - Exemplo da Substituição de Texto

3.2.1.2 Vantagens

Uma das vantagens do método de substituição descrito é permitir a alteração de palavras num texto por um utilizador fidedigno e, em simultâneo, permitir que o algoritmo saiba em qual zona do documento as alterações foram feitas.

Isso pode ser útil em situações em que uma equipa de revisão precisa de fazer alterações num documento, como num processo de edição de textos. Ao usar esse método, a equipa de revisão pode fazer as alterações necessárias sem perder informações importantes ou prejudicar a integridade do documento. Além disso, o algoritmo pode identificar facilmente as alterações realizadas e rastreá-las para fins de auditoria e averiguação de alterações autorizadas.

3.2.1.3 Desvantagens

Uma das principais desvantagens é que as mudanças no texto podem não fazer sentido para pessoas que não estão familiarizadas com a substituição de palavras usada. Isso pode tornar a leitura do texto difícil e confusa, especialmente se muitas palavras forem substituídas. Se a substituição for realizada de forma excessiva, o texto pode-se tornar ininteligível.

Além disso, o método de substituição pode não ser capaz de detetar métodos de cópia de texto, como copiar e colar o texto num novo documento. Se o novo documento for criado com base no texto original antes da substituição, as informações confidenciais podem ser utilizadas sem que a substituição tenha qualquer efeito.

Outra limitação é que esse método de substituição não garante a segurança completa dos dados, pois é possível que alguém com conhecimento suficiente possa decifrar a substituição e utilizar as informações originais.

3.2.2 Marca de Água sem Mudança de Texto

A necessidade de comparar o método de marca de água com e sem mudança textual, surgiu para escolher o melhor, levando à implementação dos autores do artigo [6] que consiste em usar as características do documento para gerar a marca de água, seguindo o layout “*autor:marcadeágua:data:tempo*”, onde esta, é registada por uma autoridade certificada (CA). O principal objetivo é usar palavras que tenham um comprimento superior a quatro caracteres. Esta escolha adveio da generalidade da mudança textual por parte de pessoas não autorizadas em palavras menores do que quatro caracteres.

Para gerar a marca de água, é necessário ter acesso ao conteúdo do documento, e assim percorrer cada palavra encontrando as palavras superiores a quatro caracteres, guardando o primeiro carácter de cada palavra. No fim, juntam-se todas as marcas de água resultando na final.

Por exemplo, na seguinte frase: “O José gosta muito de ler. A Ana não.” a marca de água final seria JGM (José Gosta Muito).

3.2.2.1 Vantagens

Tem uma entidade certificadora que possui a marca de água original, cuja é a única pessoa que pode comparar os documentos, permitindo assim que o documento esteja só na posse de uma pessoa, diminuindo os ataques ou distribuição do documento. A composição da entidade certificadora é similar ao pretendido no código de barras.

3.2.2.2 Desvantagens

Tem testes realizados com ataques aleatórios, como inserir, apagar, reordenar e alterar, que para textos reais os resultados poderão ser diferenciados, levando com que o comportamento do algoritmo seja diferenciado entre documentos, dificultando a sua implementação para a averiguação da integridade do documento.

3.2.3 Documentos baseados em Eigenvalues

O artigo [7] apresenta um algoritmo que armazena as posições de todas as palavras de um documento numa matriz, juntamente com os seus pesos em ASCII. Um documento é normalmente constituído por palavras, espaçamento, números e pontuações. Os autores consideram cada ocorrência para calcular o peso ASCII e gerar um esquema de marca de água com base numa chave privada, cujo um utilizador fidedigno verifica o documento recebido.

3.2.3.1 Testes

No artigo [7] existem testes realizados a um documento com 47 linhas, não tendo disponibilizado o mesmo, apenas impressões de pequenas secções do mesmo.

A Figura 6 aparece um excerto do texto antes de ter a marca de água, e na Figura 7 surge o texto depois da criação da marca de água. Para testar a eficiência da mesma, foi alterada a palavra “OFF” para “ON” numa zona do texto. Dos resultados binários na Figura 8 (sendo que em primeiro está o documento original e a seguir o alterado), retira-se que os números são diferentes. Uma pequena alteração da palavra levou à mudança de 11 bits.

Os autores realizaram testes na mudança de vogais, consoantes, caracteres especiais, palavras, números, pontuações, e alterações aleatórias no texto, obtendo os resultados da Tabela 4, verifica-se que na coluna “tamper detection” está tudo a 100% que leva a concluir que o algoritmo detetou alterações em todos os casos.

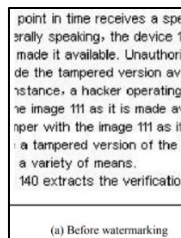


Figura 6 - Antes da Marca de Água

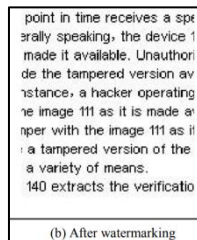


Figura 7 - Depois da Marca de Água

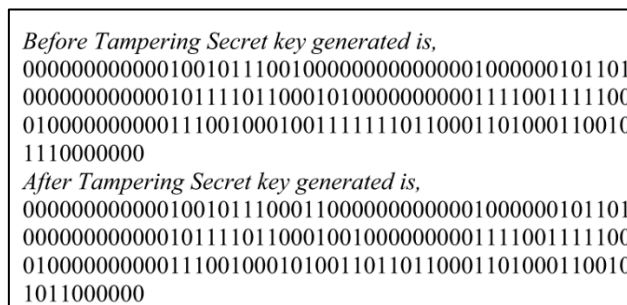


Figura 8 - Comparação Marca de água

Tabela 4 - Resultados Eigenvalues

Type of alteration (a single character)	Average eigen value shift	% bit change in secret key	Tamper Detection
Vowels	68.89	12.29	100%
Consonants	66.83	11.67	100%
Words	95.22	14.38	100%
Numerals	169.28	13.96	100%
Punctuations	53.94	10	100%
Random Alterations	127.95	15.63	100%

3.2.3.2 Vantagens

Guarda a posição das palavras do texto numa matriz. Tem uma identidade certificadora que guarda a marca de água e é a única pessoa que pode confirmar se o documento é original ou falsificado.

3.2.3.3 Desvantagens

Difícil de compreensão, a execução do algoritmo demora bastante tempo.

3.2.4 Marca de Água com Espaçamento

O artigo [8] propõe três métodos de espaçamento para a criação de marca de água, sendo eles o espaçamento de linhas (line-shift), espaçamento de palavra (word-shift) e espaçamento de letras (character). Para a realização da apuração da marca de água é necessário OCR (Optical Character Recognition), já que as modificações são difíceis de visualizar ao olho humano.

3.2.4.1 Line-Shift Coding

Consiste em deslocar as linhas de texto de um documento para cima ou para baixo, enquanto as linhas adjacentes não são movidas. Na Figura 9, tem-se um exemplo. Neste exemplo a linha do meio começada por “Effects...” foi movida para baixo 1/300 inches, que equivale a 0.00846666667 cm.

<p>the Internet aggregates traffic flows from many end systems. Understanding effects of the packet train phenomena on router and IP switch behavior will be essential to optimizing end-to-end efficiency. A range of interesting</p> <p>the Internet aggregates traffic flows from many end systems. Understanding effects of the packet train phenomena on router and IP switch behavior will be essential to optimizing end-to-end efficiency. A range of interesting</p>

Figura 9 - Exemplo Line-Shift

3.2.4.2 Word-Shift Coding

Trata-se em mover as palavras para a esquerda ou para a direita, enquanto as palavras adjacentes não são alteradas. A Figura 10 é um exemplo. A segunda linha, contém quatro palavras movidas com espaçamento de 1/150 polegadas, que equivale a 0.0169333333 cm, enquanto na primeira linha não se altera, a terceira é uma junção das duas anteriores.

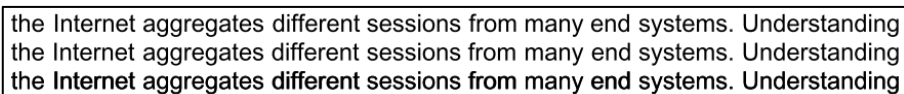


Figura 10 - Exemplo Word-Shift

3.2.4.3 Character Coding

A letra escolhida é movida para cima ou baixo, enquanto as adjacentes não se alteram. A Figura 11 apresenta um exemplo. A primeira letra “e” da palavra “internet” foi movida para baixo 1/600 inches que equivale a 0.0042333333 cm.



Figura 11 - Exemplo Character Coding

3.2.4.4 Vantagens dos Métodos

Tem marca de água invisível, ou seja, não é perceptível para o olho humano.

3.2.4.5 Desvantagens dos Métodos

Uso de OCR para análise de um documento, e ter acesso ao texto para inserir o espaçamento. Erros de impressão, por exemplo, faltar letras no documento, pode levar à má classificação da integridade de um documento.

4 Algoritmo de Marcação e Análise de Documentos Impressos

Esta secção visa a explicação do algoritmo da marcação do documento impresso bem como ocorre a sua análise.

4.1 Processamento de Documentos

O algoritmo de processamento de documentos para a criação de um sistema de verificação da integridade do mesmo segue o diagrama de fluxo presente na Figura 12, descrito abaixo:

1. Escolha do Documento a processar;
2. Obtenção de caracteres e posições do Documento. As posições são constituídas por, começo e o fim da letra em X e Y , seguindo a lógica da Figura 13;
3. Criação de pontos para os segmentos de retas e cálculo dos pontos de interseção. Utilizando as n posições pré-definidas calcula-se as novas posições (r , l , b), com um intervalo de números inteiros, onde cada valor do x e y para cada posição irá ser escolhido aleatoriamente;
4. Criação do código de barras, contendo um identificador do documento onde este está inserido numa base de dados interna ou externa;
5. Inserção do código de barras ao ficheiro duplicado do original, dando origem a um novo ficheiro.

Inicialmente os QR Code, nomeadamente os quadrados de posição, eram utilizados para a criação dos pontos para os segmentos de reta, espalhados pelos n pontos pré-definidos, contudo estes em ficheiros digitais eram difíceis de compreender ou até mesmo desvaneciam a cor, levando à sua remoção e adotando uma nova estratégia.

A nova estratégia envolve utilizar dois códigos de barras distintos, um em cima e outro em baixo do documento, onde permitia calcular a distorção em x e y no ficheiro digitalizado, calculando assim a sua proporção e adaptação das posições ao ficheiro digitalizado. Este cálculo é feito através das posições dos códigos de barras 128 e 39.

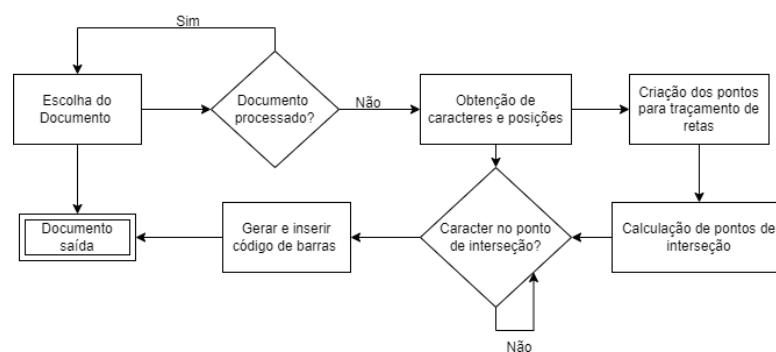


Figura 12 - Diagrama de Fluxo do Processamento de um Documento

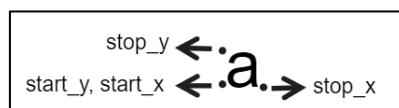


Figura 13 - Representação dos Valores dos Caracteres

4.2 Verificação de Documentos Impressos

O algoritmo de verificação de documentos impressos o diagrama de fluxo presente na Figura 14, descrito abaixo:

1. Escolha do ficheiro a examinar, pode ser formato PDF eletrónico ou digital, nos dois casos o documento é convertido para imagem para a colocação das letras presentes na base de dados;
2. Leitura do código de barras para descodificar a informação contida nele, se o identificador(id) do ficheiro estiver na base de dados e tiver sido aceite ou rejeitado a marca de água. Devolve-se alguns metadados ao utilizador e este faz uma verificação rápida de um conjunto de informações acerca do documento tal como utilizador que imprimiu e posto atual;
3. Verificação de integridade do documento caso a verificação dê errado, ou o utilizador queira processar à mesma. Consiste na comparação, através de um utilizador fidedigno, das letras que estão no documento com as da base de dados, através da conversão do documento PDF para imagem e aquando a colocação das letras e respetivos pontos converte-se de novo para PDF. Nesta abordagem verificou-se que pode existir vários pontos de interseção inseridos na mesma letra, só que com valores diferentes, o que dificultava na visualização do documento, pelo que posteriormente eliminou-se as mesmas.

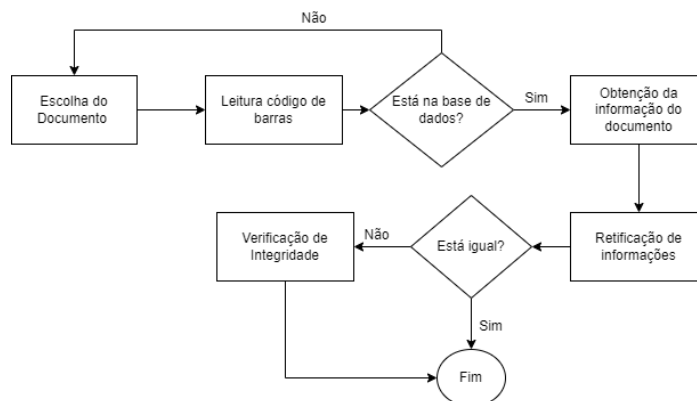


Figura 14 - Diagrama de Fluxo da Averiguação de um Documento

5 Implementação

Neste capítulo irá abordar-se o código feito, as aplicações criadas, a base de dados, a arquitetura e bibliotecas utilizadas.

5.1 Linguagens de Programação

Para o desenvolvimento do programa utilizou-se:

- C#: criação de Windows Forms (aplicações Windows);
- Java: obtenção das letras e posições nos ficheiros PDF;
- SQL: criação da base de dados - ferramenta Microsoft SQL Server Management.

5.2 Arquitetura

A arquitetura criada é baseada em três camadas (Figura 15):

- Servidor: destinado à base de dados que irá conter dados armazenados acerca dos documentos, códigos de barras, pontos dos segmentos de retas, posições dos caracteres no ficheiro de input, a criação da marca de água e dimensões do ficheiro bem como as dimensões do código de barras.
- Programa: onde acontece o desenvolvimento da criação do código de barras, do processamento do ficheiro para obtenção das posições dos caracteres no documento, da verificação do documento, e verificação de integridade, esta também é responsável pelas conexões entre camadas, desencadeadas pelo utilizador.
- Utilizador: responsável por agregar as ações que o utilizador pode fazer, como processar o documento ou verificar.

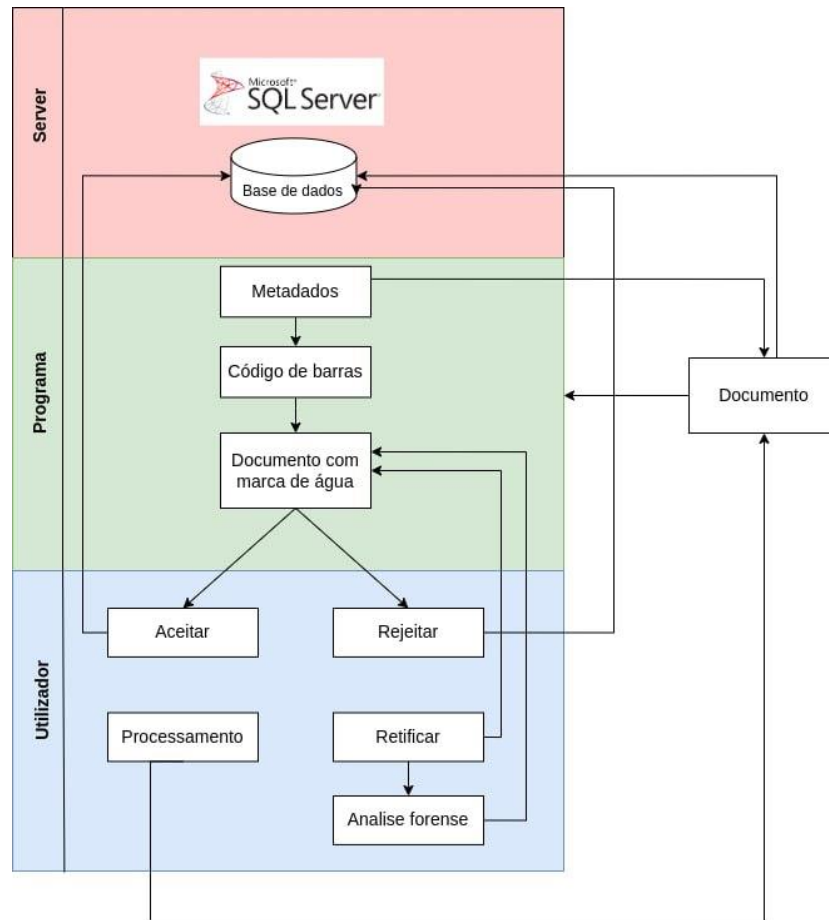


Figura 15 - Arquitetura Solução

5.3 Documentos

O algoritmo lida com três tipos de documentos (formato PDF):

- Documento sem marca de água;
- Documento com marca de água;
- Documento digitalizado com marca de água;

5.3.1 Conversão para Bitmap

Para a conversão do ficheiro PDF para Bitmap, para ser possível a inserção das letras obtidas pela interseção de segmentos de reta, utilizou-se o package FreeWare.Pdf2Png [9]. Código para a conversão é apresentado na Figura 16.

```
public string Convert_pdf_png(string file_name_png)
{
    var dd = File.ReadAllBytes(file_name_png);
    byte[] pngByte = Freeware.Pdf2Png.Convert(dd, 1);
    string[] filename = file_name_png.Split(new[] { ".pdf" }, StringSplitOptions.None);
    if(!File.Exists(filename[0] + ".png"))
        File.WriteAllBytes(filename[0] + ".png", pngByte);
    return filename[0] + ".png";
}
```

Figura 16 - Código para criar uma Imagem da primeira Página do PDF

5.3.2 Obter Dimensões do PDF e da Imagem

A necessidade de obter as dimensões do PDF e da Imagem adveio dos ficheiros digitalizados, já que estes quando criados, dependendo da impressora, tem dimensões diferentes do original o que causou, alguma discrepância nas posições. O package utilizado para obter as dimensões do documento PDF é o `iTextSharp.text.pdf` [10]. O código apresenta-se na Figura 17.

```
public void GetDimensionsImage(string img_file)
{
    Bitmap bmp = new Bitmap(img_file);
    width_bmp = bmp.Width;
    height_bmp = bmp.Height;
    bmp.Dispose();
}

public void GetDimensionsDocument(string file_name)
{
    using (Stream inputPdfStream = new FileStream(file_name, FileMode.Open,
        FileAccess.Read, FileShare.Read))
    {
        var reader = new PdfReader(inputPdfStream);
        PdfReader.unethicalreading = true;
        width = (int)reader.GetPageSize(1).Width;
        height = (int)reader.GetPageSize(1).Height;
        reader.Close();
        reader.Dispose();
    }
}
```

Figura 17 - Código para obter as dimensões do documento PDF e da Imagem Convertida

5.3.3 Criação dos Pontos para o Sistema de Verificação

Esta secção tem como intuito mostrar como são calculadas as posições que irão dar origem aos segmentos de reta.

Inicialmente, utilizava-se os pontos dos quadrados de posição do QR Code como referência para a criação dos segmentos de reta, e como a versão utilizada (1), contém 3 quadrados, criaram-se 3 pontos (r, l, b). Contudo apesar da descartarão do QR Code optou-se por continuar a utilizar os pontos (para reutilizar código), adicionando um fator aleatório aos mesmos nas coordenadas x e y .

A Figura 18 apresenta um pedaço de código que irá escrever as posições dos pontos numa imagem auxiliar (Figura 19), sendo que vai buscar todos os pontos que se predefine, neste caso 9 (*numberPoints*), gerando um valor aleatório de espaçamento (x , y) para criar os pontos aleatórios para dar a origem aos pontos das retas (l - esquerda, r - direita, b - baixo). A Figura 19 exhibe pontos exemplares, sendo que para ponto tem as origens dos pontos das retas aleatórios, podem existir pontos com as mesmas coordenadas, já que estes são aleatórios. A Figura 20 representa a quantidade de espaçamento entre pontos no eixo dos x e y .

Em suma, o algoritmo vai calcular as posições pré-definidas (p - Figura 19), com base nas coordenadas do código de barras 128, adicionando a cada ponto (p), 3 novos pontos (r, l, b) com fator aleatório.

Ou seja, imaginando-se que a coordenada do ponto da margem superior esquerda tem os valores (75,75) de seguida para cada dos 3 pontos restantes (r, l, b) vão ser escolhidos valores e espaçamento aleatório, como por exemplo, r(25,0), l(0,25), b(25,25) de espaçamento, tendo as coordenadas finais: r(75+25,75+0), l(75+0,75+25), b(75+25,75+25).

```

Graphics g = Graphics.FromImage(bmp);
Font drawFont = new Font("Arial", 8);
SolidBrush drawBrush = new SolidBrush(Color.Blue);

Dictionary<string, Point> circle_points = new Dictionary<string,
Point>();
for(int i = 0; i < numberPoint; i++)
{
    string[] pos_circles = positions.Split('|');
    string[] circles = pos_circles[i].Split(',');
    int x_circle = int.Parse(circles[0]) * bmp.Width / w;
    int y_circle = int.Parse(circles[1]) * bmp.Height / h;

    Random random = new Random();
    int randomX = random.Next(min_random, max_random);
    int randomY = random.Next(min_random, max_random);

    Point origin_point = new Point(x_circle, y_circle);
    Point circles_l = new Point(x_circle + randomX, y_circle - randomY);
    Point circles_r = new Point(x_circle - randomX, y_circle - randomY);
    Point circles_b = new Point(x_circle - randomX, y_circle);

    g.DrawString("p", drawFont, drawBrush, origin_point);
    g.DrawString("l", drawFont, drawBrush, circles_l);
    g.DrawString("r", drawFont, drawBrush, circles_r);
    g.DrawString("b", drawFont, drawBrush, circles_b);

    circle_points.Add("point" + (i + 1) + "_l", circles_l);
    circle_points.Add("point" + (i + 1) + "_r", circles_r);
    circle_points.Add("point" + (i + 1) + "_b", circles_b);
}

string path =
Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().
.Location) + @"\\Ficheiros\\spefications.png";
bmp.Save(path);

return circle_points;

```

Figura 18 - Código para a criação dos pontos aleatórios da Marca de Água



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolieiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235236

www.dre.pt



Figura 19 - Representação dos Pontos

Ponto 1	x de 15 e no y	34
Ponto 2	x de 48 e no y	21
Ponto 3	x de 60 e no y	27
Ponto 4	x de 33 e no y	15
Ponto 5	x de 45 e no y	20
Ponto 6	x de 56 e no y	26
Ponto 7	x de 30 e no y	14
Ponto 8	x de 13 e no y	37
Ponto 9	x de 24 e no y	43

Figura 20 - Espaçamento dos Pontos

5.3.4 Criação do Documento para a Integridade

Como referido no ponto 5.3.1 é necessário a criação da imagem do documento PDF, para ser possível obter as suas dimensões, desenhando por cima da imagem, guardando uma nova e só depois converte-se outra vez para PDF para ser possível a visualização ao utilizador na verificação profunda. Neste caso é necessário também calcular a escala do documento para fazer ajustes bem como adaptar os pontos nos ficheiros digitalizados.

O código para o uso da escala e respetiva criação da imagem com as letras na imagem original, com as diferenças e proporções nas coordenadas x e y calculadas, está presente na Figura 21.

```
if (scale_doc >= 1.00f)
    intersection = new Point(res_x, res_y);
else
{
    int n_x = Convert.ToInt16(res_x * (double)w /
bmp.Width);
    int n_y = Convert.ToInt16(res_y * (double)h /
bmp.Height);
    int s_x = Convert.ToInt16(n_x * scale_doc);
    int s_y = Convert.ToInt16(n_y * scale_doc);
    int new_x = Convert.ToInt16(s_x * (double)bmp.Width /
w);
    int new_y = Convert.ToInt16(s_y * (double)bmp.Height /
h);
    intersection = new Point(new_x, new_y); //adjust point
barcode
}
else
{
    int n_x = Convert.ToInt16(res_x * w / bmp.Width);
    int n_y = Convert.ToInt16(res_y * h / bmp.Height);
    int s_x = Convert.ToInt32((n_x - diff_x) * prop_x );
    int s_y = Convert.ToInt32((n_y - diff_y -
(diff_height_doc*2)+1) * prop_y );
    int new_x = Convert.ToInt16(s_x * bmp.Width / w);
    int new_y = Convert.ToInt16(s_y * bmp.Height / h);
    intersection = new Point(new_x, new_y); //adjust point
barcode
}

g.DrawString(ch, drawFont, drawBrush, intersection);
g.DrawLine(red, intersection.X - size_cross, intersection.Y,
intersection.X + size_cross, intersection.Y);
g.DrawLine(red, intersection.X, intersection.Y - size_cross,
intersection.X, intersection.Y + size_cross);
g.DrawArc(yellow, intersection.X, intersection.Y, width_circle,
height_circle, startAngle, sweepAngle);
```

Figura 21 - Imagem Auxiliar para visualização das Letras e Pontos

5.4 Base de Dados

A base de dados SQL foi criada localmente, tendo um utilizador Figura 23 e base de dados, que permite a utilização do mesmo no algoritmo. As informações que se guardam são as características do documento, o código de barras, segmentos de reta traçados entre pontos, posições dos caracteres no documento de entrada (processamento apenas), e a criação da marca de água (aceitação ou rejeição).

Na Figura 22, está presente um diagrama da base de dados que contém as tabelas usadas e respetivas conexões, de relação um para muitos, onde o ícone da chave significa chave principal.

O diagrama é constituído pelas seguintes tabelas:

- “document”: guarda características do documento (metadados);
- “barcode”: guarda informações relativas ao código de barras;
- “watermark”: guarda as confirmações do documento com marca de água, se foi aceite ou não, para efeitos de rastreamento, e as posições dos códigos de barras 128 e 39;
- “forense_analises”: guarda os segmentos de reta traçados entre dois pontos, o ponto de interseção e a letra que aparece no ponto de interseção para efeitos de verificação de integridade do documento;
- “position_char_file”: guarda as posições dos caracteres no documento;
- “dimensions_document”: guarda as dimensões do ficheiro original em formato PDF e bitmap;

Para a utilização da base de dados na aplicação utilizou-se o package “System.Data.SqlClient”, que permite fazer todo o tipo de ações numa base de dados.

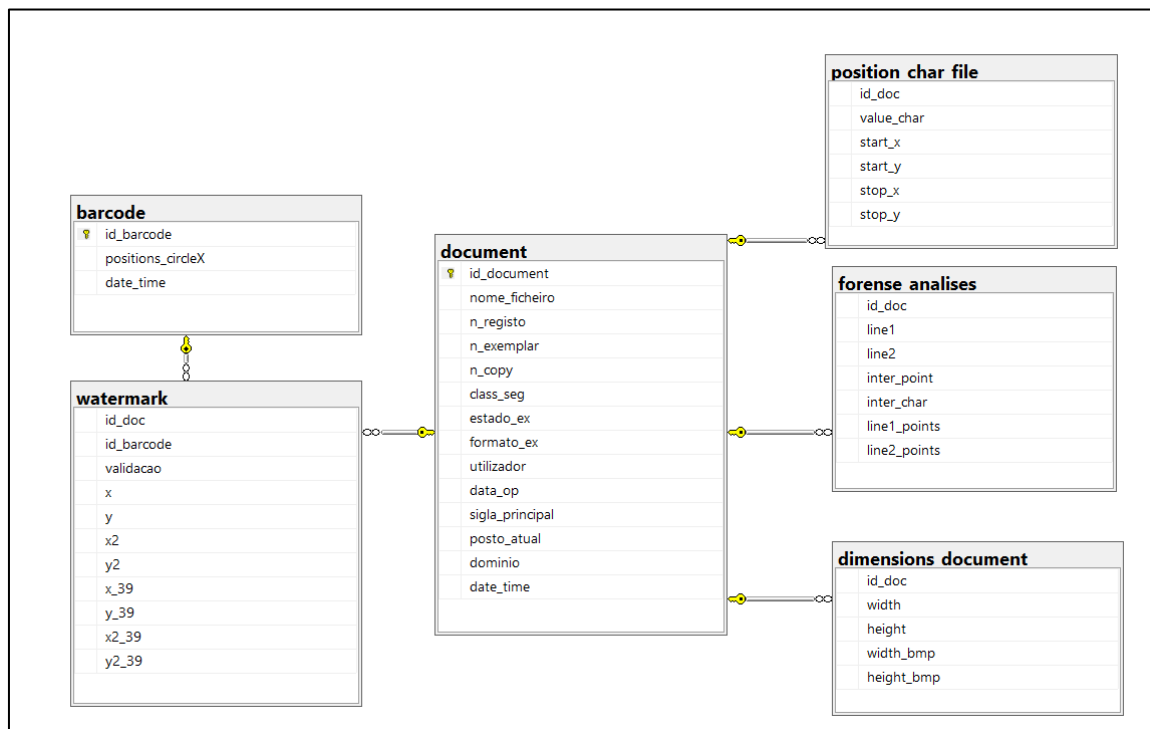


Figura 22 - Diagrama da Base de Dados

```

CREATE LOGIN antonio
    WITH PASSWORD = 'antonio';
GO

-- Creates a database user for the login created above.
CREATE USER antonio FOR LOGIN antonio;
GO

```

Figura 23 - Criação do utilizador para a base de dados

5.5 Obtenção das letras e respetivas posições num documento PDF

Apesar de existirem alguns packages em C# que retirassem as posições dos caracteres bem como os mesmos, não era suficiente, já que se precisava de um intervalo de valores onde começa e acaba a letra para determinar se o ponto de interseção pertence ou não à mesma para ser usado na verificação de integridade, optou-se por utilizar o package Apache PDFBox [11] desenvolvido em java que conseguia dar as posições do começo e fim de cada letra bem como a letra respetiva.

Para diminuir o tempo de processamento e para efeitos de teste apenas se lê a primeira página do documento, sendo possível depois alterar para todas as páginas, contudo o tempo de processamento aumenta também. Para os valores lidos serem acedidos na aplicação desenvolvida C# cria-se um ficheiro temporário que vai guardar as seguintes características “character|start_x,start_y,stop_x,stop_y”.

Contudo é necessário compilar o código desenvolvido em java e criar um ficheiro Jar com os respetivos packages dependentes que permita a execução em C#, para isso abriu-se uma consola e executou-se um comando “-jar jarfile filename”, onde ‘jarfile’ é um ficheiro java compilado e ‘filename’ o nome do ficheiro PDF que se pretende extrair as posições e as respetivas letras.

Para comprovar que o algoritmo obtém as letras precisamente, utilizou-se um leitor PDF denominado “PDF-Xchange Editor” que permite observar as posições do rato no documento como se averigua na Figura 25, a seta aponta para onde está o rato (representado por um círculo), dando os valores (71, 38), na base de dados presente na Figura 26, tem-se que os valores de “start_x” a 71 e o “stop_y” a 37, mostrando que existe um bocado de discrepância de valores devido a arredondamentos.

```

File file = new File(f[0]+"_pos.txt");
if(file.exists())
{
    file.delete();
}
else {
    file.createNewFile();
}

String fileName = args[0];
try {
    document = PDDocument.load( new File(fileName) );
    PDFTextStripper stripper = new PositionCharacter();
    stripper.setSortByPosition( true );
    stripper.setStartPage( 0 );
    stripper.setEndPage( 1 );
    Writer dummy = new OutputStreamWriter(new ByteArrayOutputStream());
    stripper.writeText(document, dummy);
}

```

Figura 24 - Código para Abrir o Documento e Extrair Informação

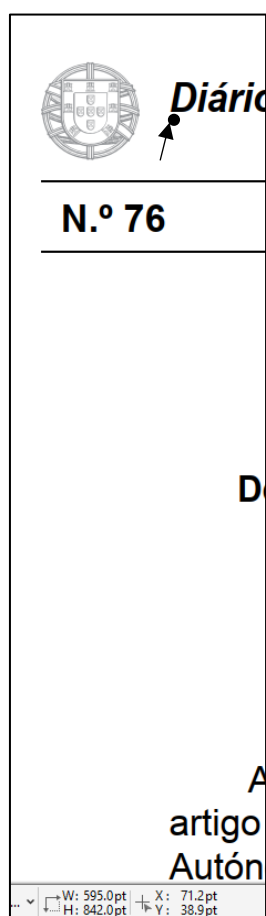


Figura 25 - Exemplo de uma Letra e Respetiva posição no PDF com a Utilização da Ferramenta PDF-XChange Editor

Results		Messages				
	id_doc	value_char	start_x	start_y	stop_x	stop_y
1	308082904	D	71	29	76	37

Figura 26 - Valor Lido pelo Algoritmo Representado na Base de Dados

5.6 Código de Barras

5.6.1 Criação e Leitura

Para a criação do código de barras 128 e 39 foi utilizado o package ZXing.net [12], *open-source* (grátis e mantido por uma comunidade).

Como referido anteriormente o código de barras irá ter um identificador referenciador para o documento e um identificador para o código de barras, ou seja, serve para saber que código de barras corresponde a qual ficheiro, pois este tem ids diferentes e *timestamp* no documento diferentes.

O código de barras 128, podendo o seu tamanho ou tipo ser alterado futuramente, irá ser colocado sempre no cabeçalho da primeira página por ser uma zona livre do documento, e permitir maior parte das vezes a sua leitura, caso falhar o utilizador é informado do insucesso da decodificação. Para gerar os códigos barras utiliza-se o código da Figura 27.

```
public void Generate_barcode(int id_barcode)
{
    string data_barcode = id_doc.ToString() + ";" + id_barcode.ToString();
    var writer = new BarcodeWriter
    {
        Format = BarcodeFormat.CODE_128,
        Options = new EncodingOptions
        {
            Height = resizedBarcode,
            Width = 250,
            Margin = 0
        }
    };
    var barcodeBitmap = writer.Write(data_barcode);
    barcodeBitmap.Save(filename + commom.extension_barcode);
}

public void Generate_barcode_39(int id_barcode)
{
    string data_barcode = id_doc.ToString() + ";" + id_barcode.ToString();
    var writer = new BarcodeWriter
    {
        Format = BarcodeFormat.CODE_39,
        Options = new EncodingOptions
        {
            Height = resizedBarcode,
            Width = 200,
            Margin = 0
        }
    };
    var barcodeBitmap = writer.Write(data_barcode);
    barcodeBitmap.Save(filename + "_code39.png");
}
```

Figura 27 - Criação dos Códigos de Barras 128 e 39

5.6.2 Posições

Com o intuito de retirar as posições dos códigos de barras 128 e 39 (Figura 28), para o cálculo da distorção do ficheiro original para o digital, utilizam-se o package ZXing.Net [12] para o código de barras 128 e o Bytescout [13] para o 39, não se utilizou o ZXing.Net para o 39, devido a erro de leitura nos ficheiros com escala variável, contudo as posições do package Bytescount são necessárias o ajuste para o x_2 e y_2 , que foram calculadas com base no comprimento e altura do código de barras 39, já que este package não fornece a segunda localização.

Os valores dos ajustes das posições foram obtidos através do auxílio da visualização da Figura 29, onde as bolas amarelas (representadas pelas setas pretas) e a cor-de-rosa representam as posições do código de barras 128 e 39 respetivamente.

```

Bitmap bmp = new Bitmap(img_file);
var reader128 = new BarcodeReader
{
    Options = new DecodingOptions
    {
        PossibleFormats = new List<BarcodeFormat> { BarcodeFormat.CODE_128 },
        TryHarder = true
    }
};

var reader39 = new Bytescout.BarCodeReader.Reader();
reader39.BarcodeTypesToFind.Code39 = true;
reader39.MaxNumberOfBarcodesPerPage = 1;
var result2 = reader39.ReadFrom(bmp);

var barcodeResult128 = reader128.Decode(bmp);
var result = barcodeResult128?.ResultPoints;

List<Point> list128 = new List<Point>();
List<Point> list39 = new List<Point>();
Point p1_barcode128 = new Point();
Point p2_barcode128 = new Point();
Point p1_barcode39 = new Point();

if (result != null && result2 != null)
{
    //é necessário percorrer a lista de pontos obtidos para adicionar numa nova lista com
    as conversões necessárias
    foreach (var point in result)
    {
        int x = (int)point.X * width / bmp.Width;
        int y = (int)point.Y * height / bmp.Height;
        list128.Add(new Point(x, y));
    }
    foreach (var point2 in result2)
    {
        int x = point2.Rect.X * width / bmp.Width;
        int y = point2.Rect.Y * height / bmp.Height;
        list39.Add(new Point(x, y));
    }

    for (int i = 0; i < list128.Count; i++)
    {
        p1_barcode128 = list128[0]; //ponto esquerdo do código de barras
        p2_barcode128 = list128[1]; //ponto direito do código de barras
    }

    for (int i = 0; i < list39.Count; i++)
    {
        p1_barcode39 = list39[0];
    }
}
else
{
    trackerServices.WriteFile("erro ao obter as posições do código de barras");
    return "";
}

//ajuste de posição do código de barras 128 e 39 para o pretendido. Neste caso limites a
preto do código de barras.
x_barcode_pos = p1_barcode128.X - 10;
y_barcode_pos = p1_barcode128.Y - 2;
x2_barcode_pos = p2_barcode128.X + 16;
y2_barcode_pos = p2_barcode128.Y - 2 + 15;
x_39 = p1_barcode39.X + 1;
y_39 = p1_barcode39.Y;
x2_39 = p1_barcode39.X + 195;
y2_39 = y_39 + 15;
bmp.Dispose();
return
$"{x_barcode_pos}:{y_barcode_pos}:{x2_barcode_pos}:{y2_barcode_pos}:{x_39}:{y_39}:{x2_39}
:{y2_39}";

```

Figura 28 - Obtenção de Posições do Código de Barras



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolieiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235236

www.dre.pt



Figura 29 - Pontos obtidos dos Códigos de Barras

5.7 Interface Gráfica (GUI)

O objetivo desta secção é mostrar a GUI do programa desenvolvido.

Começando por o menu principal (Figura 30), que é constituído por dois submenus sendo eles “processar” e “retificar”. “Processar” é a opção que cria um sistema de validação de autenticação de documentos, com a obtenção de letras de um conjunto de interseções. “Retificar” é a opção destinada para a verificação da autenticidade e integridade de um documento digital ou eletrónico ou impresso que porventura é digitalizado.

Para adicionar a pré-visualização do documento utilizou-se o Adobe Reader.dll [14], a janela é adaptada às dimensões do ficheiro, fazendo com que a interface gráfica deste *add-on* mude de posição. Também é possível fazer algumas ações do *adobe acrobat*, tal como zoom in/out e imprimir.

“Processar”, escolhe-se o ficheiro para a criação da marca, averiguando os seguintes critérios:

- Extensão “.pdf”;
- Ficheiro processado;

Caso tiver sucesso, é se reencaminhado para o formulário da Figura 31, onde este contém três botões para exercer funções:

- Processar: começa a ação da criação da marca do documento;
- Aceitar: exerce a ação de “aceitar” o documento com marca criado para efeitos de tracking, ou seja, para saber quem foi o utilizador responsável pela sua criação;
- Rejeitar: exerce a ação de “rejeitar” o documento com marca criado para efeitos de tracking, gerando novamente um novo documento;

Por outro lado o menu “Retificar” é apresentado na Figura 33, escolhendo o ficheiro com a marca criada para a sua validação de autenticidade e integridade, averiguando os seguintes critérios:

- Extensão “nome_ficheiro_watermark_dd_mm_yy_hh_m_ss”;
- Leitura do código de barras 128;
- Extensão “.pdf”;
- Documento com marca aceite ou rejeitado na base de dados;

Caso tiver sucesso nestes critérios, é se redirecionado para o formulário presente na Figura 33, onde cabe ao utilizador averiguar os metadados do documento, caso este pretender passar para averiguação da integridade, basta exercer o clique no botão “Verificação de Integridade”, onde é mostrado num ficheiro PDF as letras inseridas e os pontos de interseção dos segmentos de retas presentes na base dados (Figura 34). Na Figura 35 é possível visualizar com zoom a zona do retângulo preto da Figura 34.

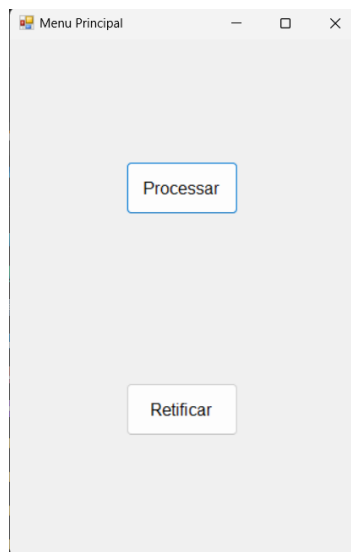


Figura 30 - Menu



Figura 31 - Menu Processar

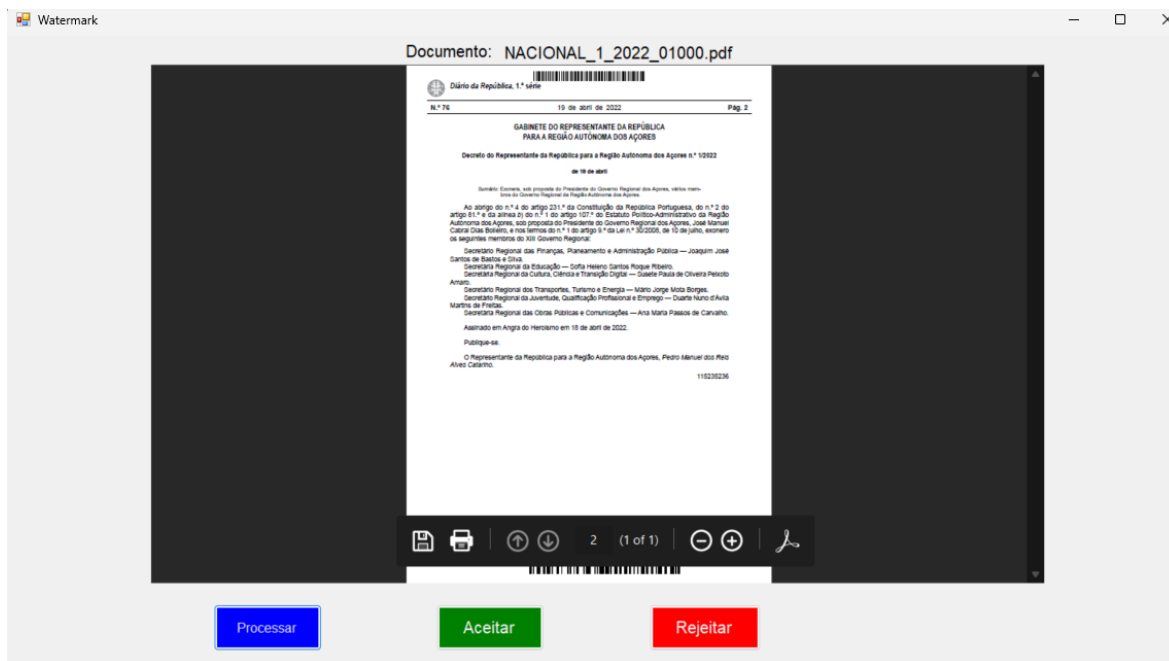


Figura 32 - Conclusão do Processamento

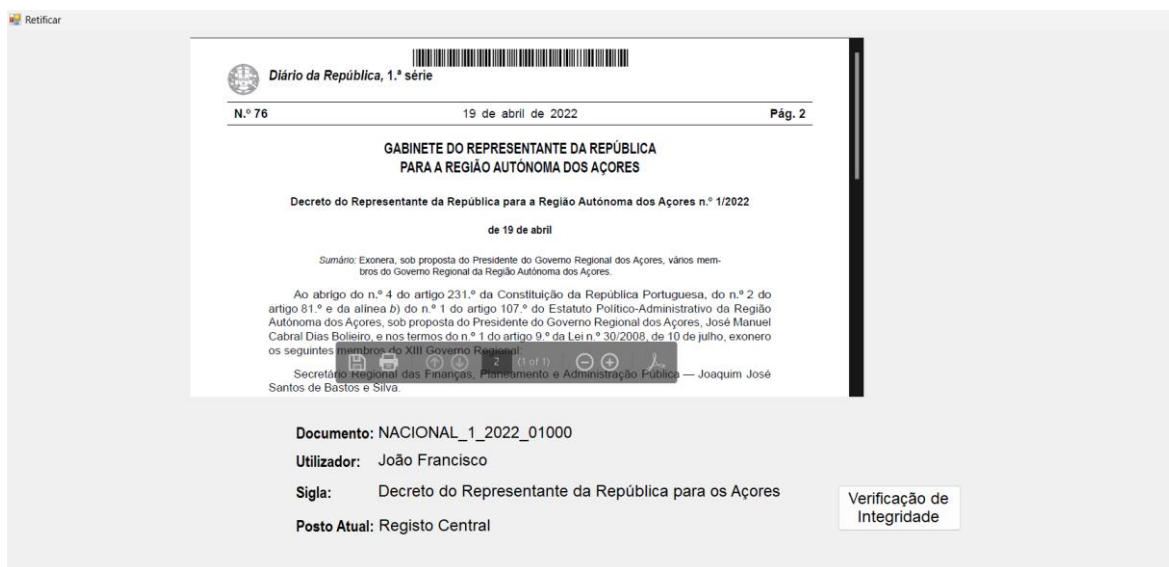


Figura 33 - Verificação da Autenticidade de um Documento

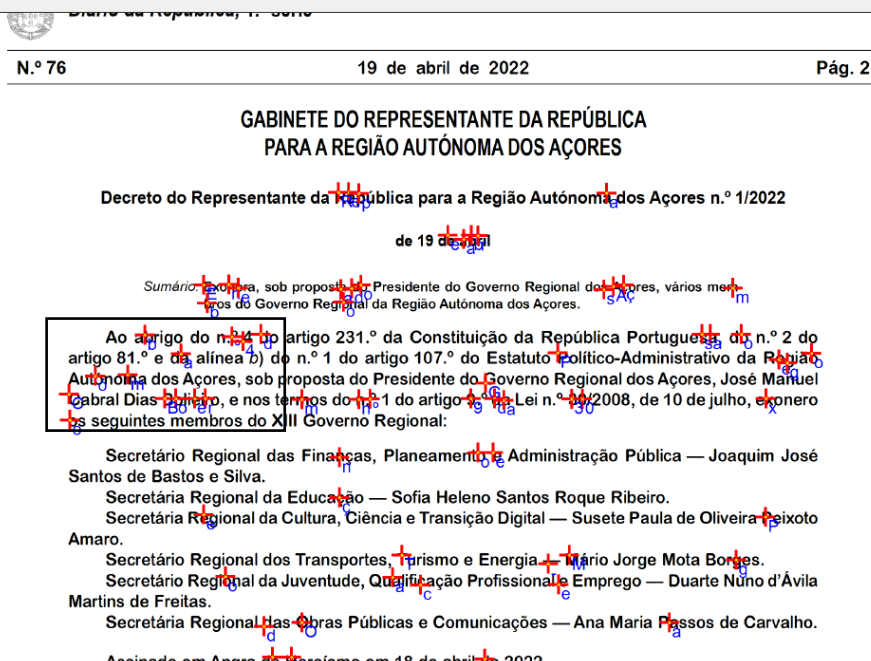


Figura 34 - Visualização da marca

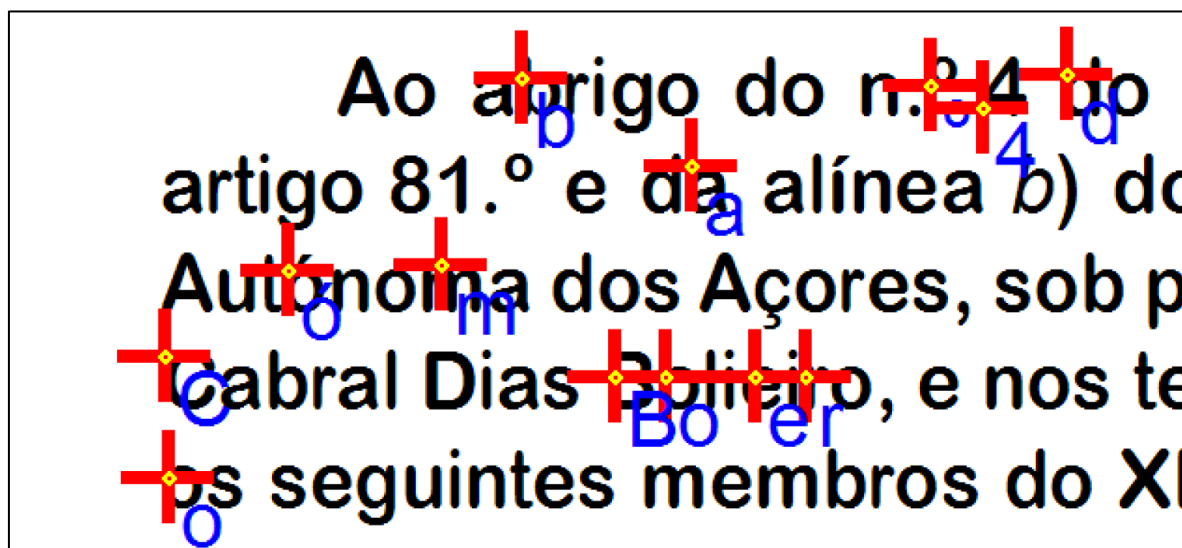


Figura 35 - Zoom retângulo

6 Testes

Para desenvolver um algoritmo é necessário garantir que ele funcione em múltiplos casos, ou seja, que seja robusto. O algoritmo desenvolvido envolve operações em ficheiros a que podem ser alvo de alterações como, por exemplo, mudança de escala do ficheiro para impressão e a folha poder ser colocada torta no digitalizador, sendo que esta é abordada na secção anterior. Nesta secção aborda-se o comportamento do algoritmo em situações de mudança de escala e alteração de letras no ficheiro original.

6.1 Documentos Digitais com Rotações

Numa digitalização é possível que um documento seja digitalizado torto, podendo comprometer a validação de integridade, já que a folha está torta, e as posições podem calhar em zonas brancas, já que a folha rodou, o que leva a que os documentos digitais torto sejam relevantes para os testes de robustez-.

A Figura 36, mostra um documento digitalizado relativamente torto, cujo deu entrada no sistema, depois de algum processamento de imagem o resultado apresenta-se na Figura 37, demonstrando que o algoritmo conseguiu endireitar o documento, contudo podem existir casos em que a folha esteja muito torta e o algoritmo não funciona, caso isto aconteça cabe ao utilizador fazer uma digitalização nova.

Através da comparação da Figura 38, do resultado da verificação de integridade com o original (Figura 39), pode-se afirmar que a discrepância entre pontos é muito pouca sendo a percentagem de eficácia alta, é de salientar que devido à falta de impressoras, só se realizou os testes na impressora Brother MFC7460DN, sendo que os resultados poderão ser diferentes para outras impressoras.



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolleiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235236

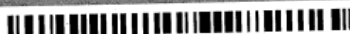


Figura 36 - Documento Digitalizado com Rotação



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolieiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235236



Figura 37 - Documento digitalizado após correção



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Genera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 1 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 107.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Carlos Dias Borges, e nos termos do n.º 1 do artigo 19.º da Lei n.º 307/2008, de 10 de julho, exerce os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, **Rui Manuel dos Reis Alves Catarino**.

115235236



Figura 38 - Controlo de integridade sobre um documento corrigido após digitalização defeituosa

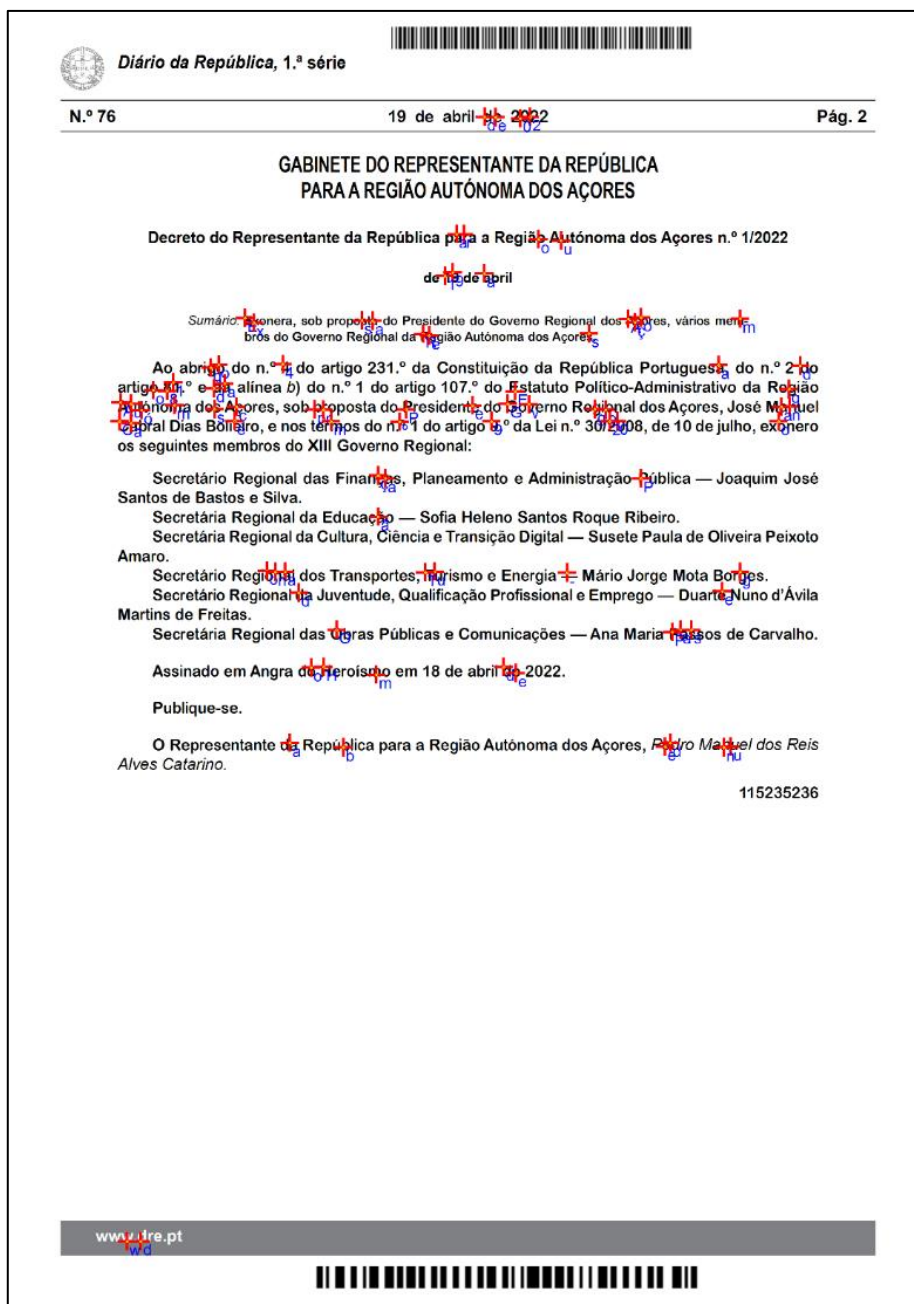


Figura 39 - Apresentação dos elementos de controlo de integridade sobre um documento original

6.2 Mudança de Escala no Documento

A escala ideal para testar a robustez do algoritmo é entre 80-95%, sendo que abaixo de 50%, o documento torna-se ilegível a olho nu (Figura 40), sendo necessário o uso de ferramenta de leitura de PDF para realizar zoom.

O ficheiro utilizado para alteração da escala está apresentado na Figura 41, tendo o resultado da verificação na Figura 42.

Para alterar a escala do ficheiro utilizou-se o package *iTextSharp* do C# (abordado mais detalhadamente em anexo [A](#)).

Na Figura 43 apresenta-se uma captura de ecrã do ficheiro da Figura 41, com uma escala de 80%, sendo que o resultado obtido da verificação se apresenta na Figura 44. Através da comparação de pontos entre a verificação da integridade da Figura 44 e Figura 42, conclui-se que o algoritmo conseguiu adaptar as posições no ficheiro com escala variável.

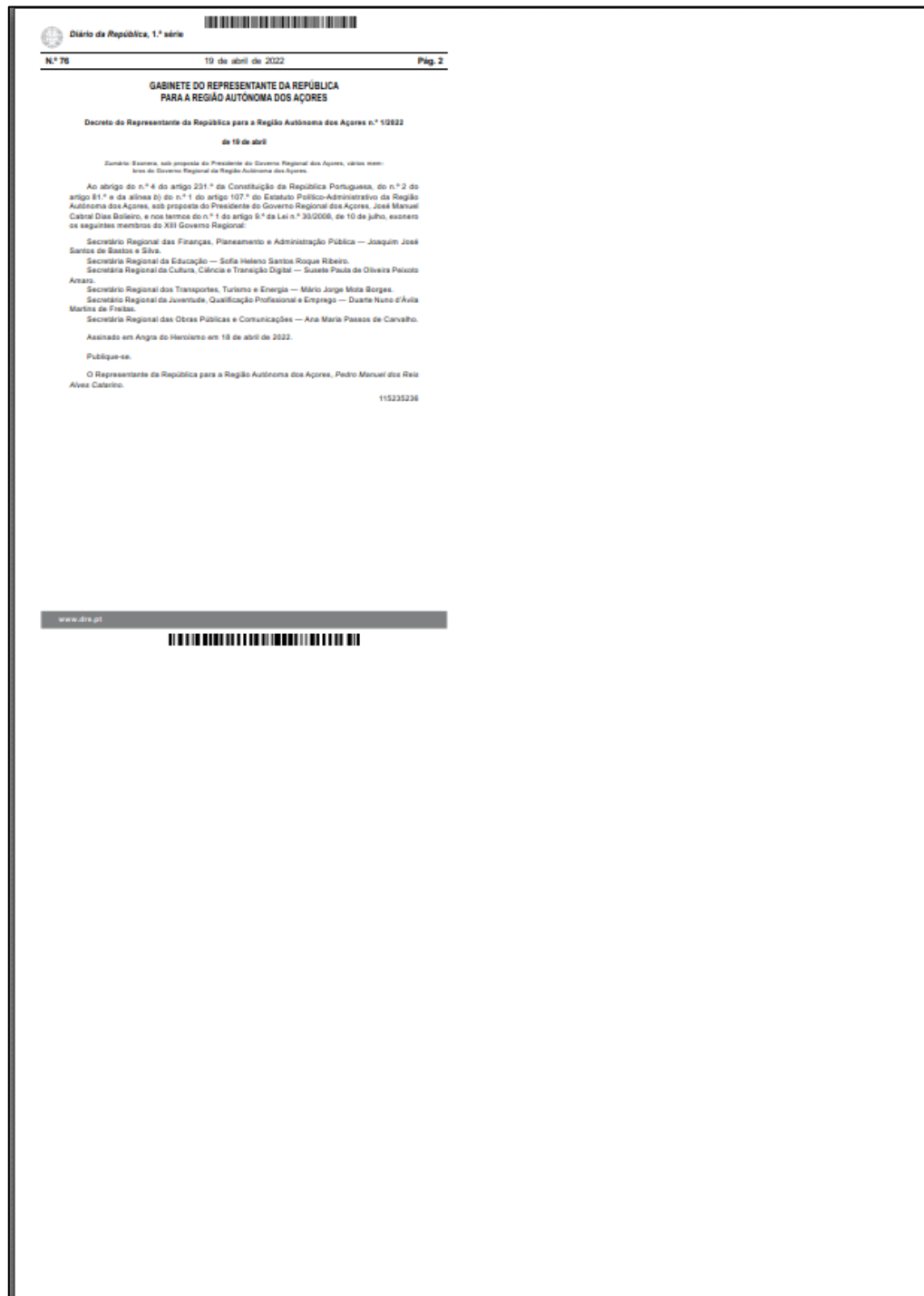


Figura 40 - Documento Original com 50% de Escala



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolieiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235236





GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 18 de abril

Sumário: Aprova, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa do n.º 2 do artigo 107.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Capral Dias Bollero, e nos termos do n.º 1 do artigo 6.º da Lei n.º 30/2008, de 10 de julho, exonerou os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, Pedro Manuel dos Reis Alves Catarino.

115235236



Figura 42 - Apresentação dos elementos de controlo de integridade sobre um documento original (2)



**GABINETE DO REPRESENTANTE DA REPÚBLICA
PARA A REGIÃO AUTÓNOMA DOS AÇORES**

Decreto do Representante da República para a Região Autónoma dos Açores n.º 1/2022

de 19 de abril

Sumário: Exonera, sob proposta do Presidente do Governo Regional dos Açores, vários membros do Governo Regional da Região Autónoma dos Açores.

Ao abrigo do n.º 4 do artigo 231.º da Constituição da República Portuguesa, do n.º 2 do artigo 81.º e da alínea b) do n.º 1 do artigo 107.º do Estatuto Político-Administrativo da Região Autónoma dos Açores, sob proposta do Presidente do Governo Regional dos Açores, José Manuel Cabral Dias Bolieiro, e nos termos do n.º 1 do artigo 9.º da Lei n.º 30/2008, de 10 de julho, exonero os seguintes membros do XIII Governo Regional:

Secretário Regional das Finanças, Planeamento e Administração Pública — Joaquim José Santos de Bastos e Silva.

Secretária Regional da Educação — Sofia Heleno Santos Roque Ribeiro.

Secretária Regional da Cultura, Ciência e Transição Digital — Susete Paula de Oliveira Peixoto Amaro.

Secretário Regional dos Transportes, Turismo e Energia — Mário Jorge Mota Borges.

Secretário Regional da Juventude, Qualificação Profissional e Emprego — Duarte Nuno d'Ávila Martins de Freitas.

Secretária Regional das Obras Públicas e Comunicações — Ana Maria Passos de Carvalho.

Assinado em Angra do Heroísmo em 18 de abril de 2022.

Publique-se.

O Representante da República para a Região Autónoma dos Açores, *Pedro Manuel dos Reis Alves Catarino*.

115235238



Figura 43 - Documento Original com 80% de Escala

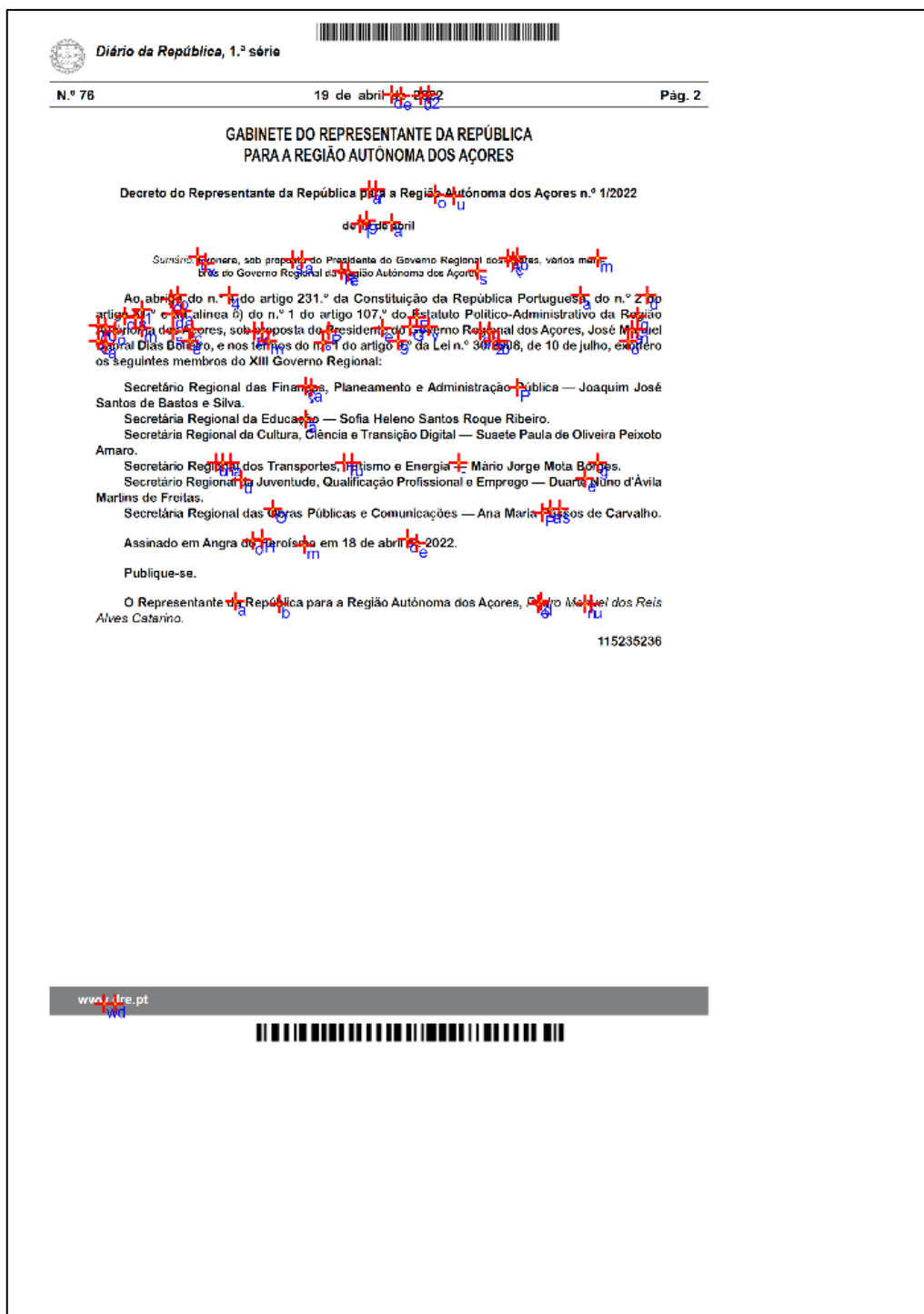


Figura 44 - Apresentação dos elementos de controlo de integridade sobre um documento original com 80% de escala

6.3 Modificações no Documento

A importância de testar o algoritmo para diversas situações é fundamental para que este seja robusto. A funcionalidade principal do algoritmo é a verificação de letras no documento, para isso é necessário visualizar o comportamento do algoritmo em situações de eliminação/adição de informação no ficheiro, alterações de palavras, e eliminação/adição de espaçamento entre linhas.

Para verificar se o algoritmo processa novos ficheiros, criou-se um com informações aleatórias presente na Figura 45. Com a realização do teste do processamento do ficheiro e verificação da integridade do mesmo, retificou-se que a solução é escalável, pelo output da Figura 46.

Criaram-se três ficheiros idênticos ao da Figura 45, com alterações, sendo eles:

1. Substituição de palavra: para a substituição resolveu-se substituir a palavra “importante” por “necessário” presente no retângulo da Figura 48, tendo o zoom representado Figura 47. Pode-se conferir que a letra “n” está por cima da letra “r”, o que leva a inferir que o documento naquela zona foi alterado;
2. Eliminação de palavras: eliminação do último parágrafo do documento da Figura 45 presente no retângulo da Figura 50, tendo o zoom representado Figura 51. Averigua-se através da letra “D” inserida numa zona branca do documento, que houve eliminação de pelo menos uma palavra.
3. Eliminação de espaçamento: removeu-se um espaçamento do documento da Figura 45, representado pelo retângulo na Figura 52, tendo o zoom representado Figura 53. Através da análise do documento, observa-se que existiu uma translação no eixo do y (de cima para baixo) - representada pela seta na Figura 52, inferindo que ocorreram mudanças na estrutura do documento.

Em suma, se o documento que der entrada no sistema tiver os códigos de barras consegue-se inferir as zonas de alteração do documento, caso esta esteja incluída na marca de água. Dando um exemplo, imaginando que a palavra “Nacional” era alterada, o algoritmo não deteta, mas caso existam adições ou eliminações de palavra consegue detetar como é caso do ponto 2 e 3, referidos anteriormente.

A segurança é uma preocupação constante em qualquer empresa, independentemente do ramo de atuação. Para garantir a segurança dos colaboradores e proteger os ativos da empresa, é necessário adotar medidas de prevenção e controle de riscos. Entre as ações mais comuns, destacam-se a instalação de câmeras de segurança, o controle de acesso ao prédio e o treinamento dos funcionários para situações de emergência.

Além disso, é importante que a empresa conte com profissionais especializados em segurança, que possam identificar e mitigar os riscos existentes. Esses profissionais devem estar sempre atualizados em relação às normas e regulamentações de segurança, bem como em relação às novas tecnologias disponíveis para aprimorar a segurança da empresa. Dessa forma, é possível garantir um ambiente de trabalho mais seguro e tranquilo para todos os colaboradores.

Figura 45 - Documento Original (2)



Nacional Segurança

20/04/23

A segurança é uma preocupação constante em qualquer empresa, independentemente do ramo de atuação. Para garantir a segurança dos colaboradores e proteger os ativos da empresa, é necessário adotar medidas de prevenção e controle de riscos. Entre as ações mais comuns, destacam-se a instalação de câmeras de segurança, o controle de acesso ao prédio e o treinamento dos funcionários para situações de emergência.

Além disso, é importante que a empresa conte com profissionais especializados em segurança, que possam identificar e mitigar os riscos existentes. Esses profissionais devem estar sempre atualizados em relação às normas e regulamentações de segurança, bem como em relação às novas tecnologias disponíveis para aprimorar a segurança da empresa. Dessa forma, é possível garantir um ambiente de trabalho mais seguro e tranquilo para todos os colaboradores.



Figura 46 - Apresentação dos elementos de controle de integridade sobre um documento original (3)



Figura 47 - Zoom da palavra "importante"



Nacional Segurança

20/04/23

A segurança é uma preocupação constante em qualquer empresa, independentemente do ramo de atuação. Para garantir a segurança dos colaboradores e proteger os ativos da empresa, é necessário adotar medidas de prevenção e controle de riscos. Entre as ações mais comuns, destacam-se a instalação de câmeras de segurança, o controle de acesso ao prédio e o treinamento dos funcionários para situações de emergência.

Além disso, é necessário que a empresa conte com profissionais especializados em segurança, que possam identificar e mitigar os riscos existentes. Esses profissionais devem estar sempre atualizados em relação às normas e regulamentações de segurança, bem como em relação às novas tecnologias disponíveis para aprimorar a segurança da empresa. Dessa forma, é possível garantir um ambiente de trabalho mais seguro e tranquilo para todos os colaboradores.



Figura 48 - Controle de integridade sobre um documento original com substituição de palavra

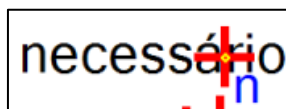


Figura 49 - Zoom da palavra "necessário"



Nacional Segurança

20/04/23

A segurança é uma preocupação constante em qualquer empresa, independentemente do ramo de atuação. Para garantir a segurança dos colaboradores e proteger os ativos da empresa, é necessário adotar medidas de prevenção e controle de riscos. Entre as ações mais comuns, destacam-se a instalação de câmeras de segurança, o controle de acesso ao prédio e o treinamento dos funcionários para situações de emergência.

Além disso, é necessário que a empresa conte com profissionais especializados em segurança, que possam identificar e mitigar os riscos existentes. Esses profissionais devem estar sempre atualizados em relação às normas e regulamentações de segurança, bem como em relação às novas tecnologias disponíveis para aprimorar a segurança da empresa.

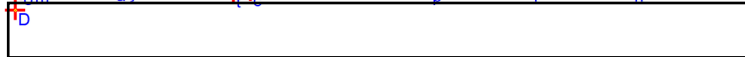


Figura 50 - Controlo de integridade sobre um documento original com eliminação do parágrafo

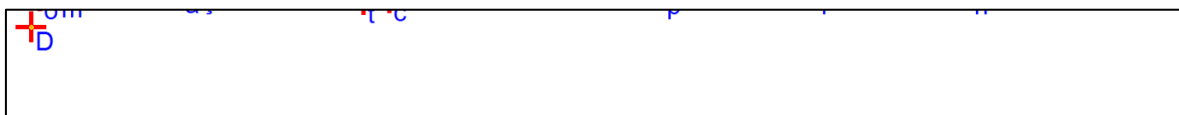


Figura 51 - Zoom sobre o parágrafo removido

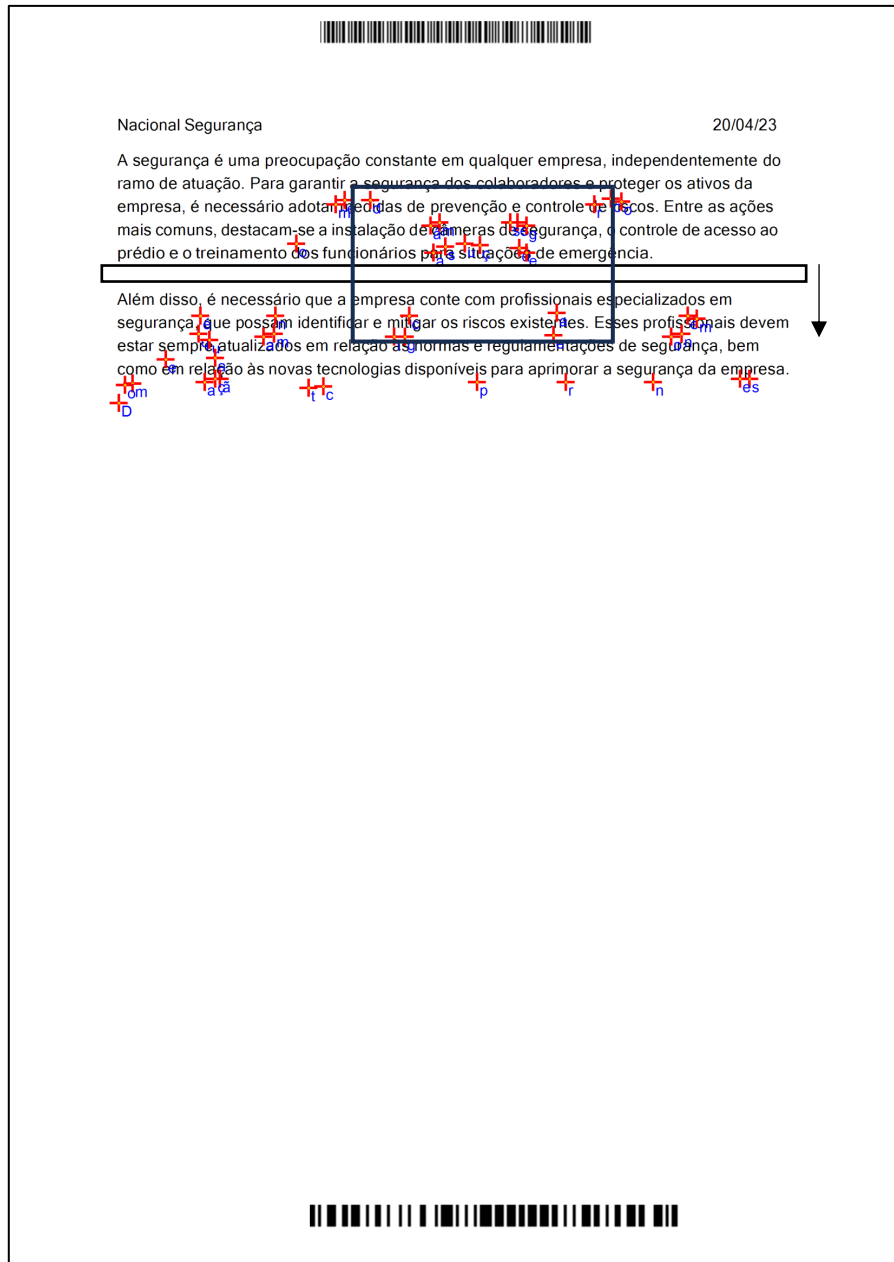


Figura 52 - Controlo de integridade sobre um documento original com eliminação de espaçamento

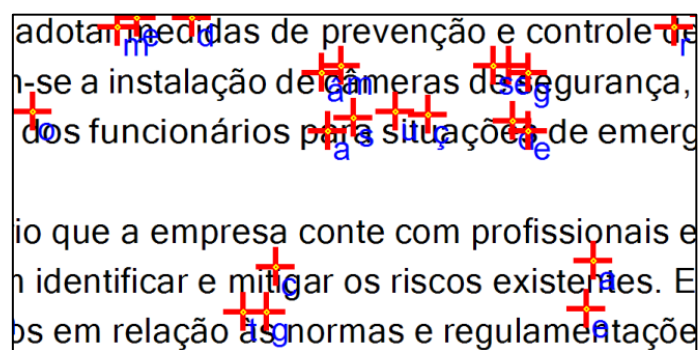


Figura 53 - Zoom da eliminação de espaçamento

7 Conclusões

O intuito da dissertação foi criar um método de verificação da autenticidade e integridade de documentos impressos.

Inicialmente abordou-se a utilização de marcas de água sobre o texto, contudo descartou-se devido a estas alterem texto nos documentos, sendo estes são classificados, o que leva dificultar a análise da integridade do documento. De seguida abordou-se a utilização de códigos de barras para a indexação de metadados referentes ao documento na base de dados e à utilização de QR Codes como método de visualização de segmentos de retas, que após testes em ficheiros digitais se apercebeu que estes perdiam a cor, o que influenciava a visualização do documento, pelo que se retirou os mesmos.

Em suma, o sistema desenvolvido final consiste em criar uma marca com base em letras obtidas através de n interseções de segmentos de retas que estejam inseridos no intervalo da letra da criação, e colocar os metadados indexados ao código de barras que porventura o identificador estará associado a uma base de dados.

Através de testes realizados com documentos disponibilizados e criados com informações aleatórias, conseguiu-se perceber que para as alterações de espaçamento, de letras e de parágrafos, o sistema deteta as alterações, desde que estas estejam contidas na marca. Também foram realizados testes com documentos digitalizados com marca para a deteção da mesma e verificou-se a adaptação das posições em relação à verificação da integridade do documento original.

Concluindo, conseguiu-se atingir o objetivo inicial de criar um método que garanta a autenticidade e integridade de documentos impressos, permitindo assim a sua proteção em matérias classificadas.

7.1 Trabalho Futuro

Como o presente trabalho só garante a verificação de integridade de documentos impressos numa só folha, ou seja, na primeira folha do documento, pode-se alterar o algoritmo para que este consiga garantir a verificação para todas as folhas. Porventura, caso haja essa alteração o tempo de processamento vai aumentar, contudo é garantido a proteção do documento nas marcas de letras registadas na base de dados.

No presente trabalho, os documentos digitais deformados são desconsiderados, especialmente documentos com códigos de barras ilegíveis, pelo que se possa desenvolver um método de computação visual capaz de compor a imagem do código de barras, o que garante assim a taxa de aceitação do documento digitalizado no algoritmo,

Poderá ser considerado realizar um estudo de número ideal de letras a guardar que permitam a eficácia da verificação de integridade do documento impresso. Este estudo pode consistir em mudar o número de pontos para a criação de segmentos de retas e obter as letras do mesmo, em diversos testes, calculando a sua média e porventura escolher o que tem mais letras.

A presente visualização da marca pode ser melhorada com visualização 3D, onde possivelmente existe uma junção de camadas de documentos, o original e as letras da marca, cuja visualização das letras da marca se sobre põe às do documento com uma cor ou tamanho ou tipo de letra diferente.

Outro método possível para melhorar a visualização da marca é criar um mecanismo capaz de salientar e detetar as zonas alteradas do documento. Porventura, pode-se salientar as zonas alteradas que coincidem

pode-se colocar a cor da letra como verde e as que não com o vermelho. Este método aumenta a facilidade da comparação da integridade do documento com as marcas.

8 Referências

- [1] “Conhecimento é Poder - Significado, origem e autores.” <https://conceitosdomundo.pt/conhecimento-e-poder/> (accessed Apr. 31, 2023).
- [2] “IDENTIDADE E BIOMETRIA - Gesmac: Gestão de Matéria Classificada.” <http://www.sinfic.pt/ib/displayconteudo.do2?numero=34595> (accessed Apr 31, 2023).
- [3] “Code 128 - Wikipedia.” https://it.wikipedia.org/wiki/Code_128 (accessed Jan. 19, 2023).
- [4] “A4 size in point. Read here what the A4 size is in po.” <https://www.a4-size.com/a4-size-in-point/> (accessed Mar. 01, 2023).
- [5] M. B. Mohd, S. Mohd, R. Tanzila, and S. A. Rehman, “Replacement Attack: A New Zero Text Watermarking Attack,” *3D Res.*, vol. 8, 2017, doi: 10.1007/s13319-017-0118-y.
- [6] Z. Jalil, A. M. Mirza, and H. Jabeen, “Word length based zero-watermarking algorithm for tamper detection in text documents,” *ICCET 2010 - 2010 Int. Conf. Comput. Eng. Technol. Proc.*, vol. 6, no. May, 2010, doi: 10.1109/ICCET.2010.5486185.
- [7] T. Rethika, I. Prathap, R. Anitha, and S. V. Raghavan, “A novel approach to watermark text documents based on eigen values,” *2009 Int. Conf. Netw. Serv. Secur. N2S 2009*, no. c, pp. 1–5, 2009.
- [8] J. T. Brassil, S. Low, and N. F. Maxemchuk, “Copyright protection for the electronic distribution of text documents,” *Proc. IEEE*, vol. 87, no. 7, pp. 1181–1196, 1999, doi: 10.1109/5.771071.
- [9] “Freeware.Pdf2Png.” <https://www.nuget.org/packages/Freeware.Pdf2Png/> (accessed Jun. 01, 2023).
- [10] “iTextSharp | iText PDF.” <https://itextpdf.com/products/itextsharp> (accessed Feb. 01, 2023).
- [11] “Apache PDFBox | A Java PDF Library.” <https://pdfbox.apache.org/> (accessed Feb. 17, 2023).
- [12] “micjahn/ZXing.Net: .Net port of the original java-based barcode reader and generator library zxing.” <https://github.com/micjahn/ZXing.Net> (accessed May. 01, 2023).
- [13] “Bytescout BarCode Reader is a Desktop App to Read QR Code and Other Barcodes from Image Files (JPG, PNG, TIFF, GIF), PDF Documents and Webcam - ByteScout.” <https://bytescout.com/products/enduser/misc/barcodereader.html> (accessed May. 01, 2023).
- [14] “(2) Making PDF Viewer in C#.net - YouTube.” <https://www.youtube.com/watch?v=pQjskYF9yNY> (accessed May. 01, 2023).

9 Anexo

A) Mudança de Escala do Documento

A escala do documento é modificada através do uso do package *ITextSharp* que permite alterações em ficheiros PDF.

O código apresentado na Figura 54, acede ao ficheiro que se quer alterar permitindo a mudança de escala em ficheiros seguros através da definição *unethicalreading*. A seguir cria um ficheiro novo com a extensão da escolha desejada. O novo documento com a escala criada vai ser colocada na margem esquerda, ou seja, o documento vai ter as dimensões do original, mas não vai preencher a página toda. Na secção 6.1 apresenta-se um documento exemplo com 80% de escala (Figura 43).

O cálculo da escala do ficheiro é calculado com base na posição do ponto y do código de barras 128 sobre a dimensão do ficheiro original que é sempre 842, somando 0.03 para ajustar a escala. Devido aos comprimentos e pontos sofrerem proporção da escala é necessário adaptar os pontos (Figura 55). Para finalizar adaptam-se as posições de interseção com base na percentagem de escala como se apresenta a Figura 56.

```
int s = Convert.ToInt16(scalef * 100);
PdfReader reader = new PdfReader(name);
PdfReader.unethicalreading = true; // aceder a documentos confidenciais
Document doc = new Document();
PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream(name_without_ex + "_scale_"
+ s + ".pdf", FileMode.Create));
doc.Open();
PdfImportedPage page = writer.GetImportedPage(reader, 1); //page #1
PdfDictionary pageDict = reader.GetPageN(1);
pageDict.Put(PdfName.PRINTSCALING, new PdfNumber((float)scalef));

float yPos = reader.GetPageSize(1).Height - (reader.GetPageSize(1).Height *
(float)scalef);
writer.DirectContent.AddTemplate(page, (float)scalef, 0, 0, (float)scalef, 0, yPos);

doc.NewPage();
doc.Close();
reader.Close();
writer.Close();
```

Figura 54 - Alteração da Escala do Documento

```
scale_doc = Math.Round(Convert.ToDecimal(((double)p1_dig.Y / 842)), 2);
scale_doc += 0.03m;

if (scale_doc == 0.99m)
    scale_doc = 1.00m;
if (p1_39_dig.Y == 10)
    p1_39_dig.Y = 11;

if (scale_doc != 1.00m)
{
    int x_scale = Convert.ToInt16(x_diff_or * scale_doc);
    int x_39_scale = Convert.ToInt16(x_39_diff_or * scale_doc);
    int y_scale = Convert.ToInt16(y_diff_or * scale_doc);

    p1_dig.X += 2;
    p2_dig.X = Convert.ToInt16(p1_dig.X + x_scale);
    p2_dig.Y = Convert.ToInt16(p2_dig.Y + y_scale - y_diff_or);
    p2_39_dig.X = Convert.ToInt16(p1_39_dig.X + x_39_scale);
    p2_39_dig.Y = Convert.ToInt16(p2_39_dig.Y + y_scale - y_diff_or + 1);
}
```

Figura 55 - Obtenção da Escala do Documento


```
if (scale_doc >= 1.00f)
    intersection = new Point(res_x, res_y);
else
{
    int n_x = Convert.ToInt16(res_x * (double)w/ bmp.Width);
    int n_y = Convert.ToInt16(res_y * (double)h/ bmp.Height);
    int s_x = Convert.ToInt16(n_x * scale_doc);
    int s_y = Convert.ToInt16(n_y * scale_doc);
    int new_x = Convert.ToInt16(s_x * (double)bmp.Width/w);
    int new_y = Convert.ToInt16(s_y * (double)bmp.Height/h);
}
```

Figura 56 - Adaptar Posições