João Pedro Alves Fortunato

# ASSINATURA DE DOCUMENTOS EM TERMINAIS MÓVEIS RECORRENDO AO CARTÃO DE CIDADÃO

Relatório de Estágio
Mestrado em Engenharia Informática
orientada por: Prof. Dr. Fernando Barros e Eng. Avelino Martins
apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Janeiro de 2018

· U C ·

Universidade de Coimbra

# Resumo

Com o avanço tecnológico das últimas décadas a escrita é cada vez mais realizada em documentos eletrónicos. Num mundo empresarial cada vez mais competitivo surge a necessidade de assinar documentos digitais de uma forma rápida para agilizar o processo de negócio. As assinaturas digitais foram criadas para garantir a autenticação, integridade e não-repúdio de documentos eletrónicos através de técnicas que ligam o assinante ao documento assinado, tais como o cálculo de um valor criptográfico obtido com o uso de informação privada que só o assinante possui.

Este relatório documenta o trabalho realizado pelo aluno João Pedro Alves Fortunato, no âmbito de estágio em Engenharia Informática relativamente à especialidade de Engenharia de Software do mestrado em Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Este estágio tem como principal objetivo a criação de uma aplicação móvel em Android que permita adicionar uma assinatura digital a documentos PDF , utilizando para isso o Cartão de Cidadão Português.

# Palavras-Chave

*Digital Signature, Detached, External, PDF,* Word*, XML, Cartão de Cidadão, Smart Card,* Android*, Mobile*

# Agradecimentos

À Present Technologies pela oportunidade de estágio pelo excelente ambiente presente, bem como a compreenssividade e flexibilidade proporcionados durante este período de estágio. Quero também agradecer ainda a toda a ajuda disponibilizada e conhecimentos transmitidos.

Ao Engº Avelino Martins por toda a ajuda fornecida na reflexão de conceitos, presença e pela revisão deste documento.

Ao meu orientador do DEI Prof. Fernando Barros pelos seus conselhos ao longo do semestre, bem como a revisão deste documento.

A todos os meus amigos e que me acompanharam nos momentos fáceis e difíceis desta importante etapa.

À minha família por terem acreditado em mim e permitindo que chegasse onde cheguei hoje, tanto como pessoa como a nível académico.

# Índice

# Lista de Figuras

# Lista de Tabelas

# Glossário

**App** *app* é uma aplicação móvel.

**Certificate chain** *Certificate chain* consiste numa corrente de certificados, a contar deste o certificado da *Certificate Authority* até ao certificado da *End Entity*.

**Certificate revocation lists** *Certificate revocation lists* são listas onde nelas estão os certificados que se encontram revogados por alguma razão logica.

**Encoding** converte informações de uma fonte em símbolos para comunicação ou armazenamento.

**Pinpad** é um leitor de pins físico.

**Private key** *Private key* representa uma chave privada que pode ser usada tanto para encriptar como para desencriptar, dependendo do contexto da sua aplicação.

**Public key certificate** *Public key certificate* é um documento eletrônico usado para provar a propriedade de uma chave pública.

**Public key** *Public key* é uma chave publica que pode ser usada tanto para encriptar como para desencriptar, dependendo do contexto da sua aplicação.

**Signature Dictionary** *Signature dictionary* é um objeto necessário num documento PDF assinado.

**Smartcard** é um *token* de segurança que possui um *chip* embutido.

**Timestamps** *Timestamp* é uma sequência de caracteres ou informações codificadas que identificam quando ocorreu um determinado evento, geralmente dando data e hora do dia, às vezes precisas para uma pequena fração de segundo.

**USB token** *USB tokens* são dispositivos físicos usados para obter acesso a um recurso restrito eletronicamente.
**Workflow** consiste em um padrão orquestrado e repetitivo de atividade comercial habilitado pela organização sistemática de recursos em processos que transformam materiais, fornecem serviços ou processam informações.

# Tabela de acronimos

**APDU** Application Protocol Data Unit

**API** Application Programming Interface

**CMS** Cryptographic Message Syntax

**DER** Distinguished Encoding Rules

**ECC** European Citizen Card

**eIDAS** Electronic Identification and Trust Service Regulation

**ESIGN** Electronic Signatures in Global National Commerce

**EU** European Union

**HI** Hardware Interface

**ICAO** International Civil Aviation Organization

**ISO** International Organization for Standardization

**LDAP** Lightweight Directory Access Protocol

**OCSP** Online Certificate Status Protocol

**OPC** Open Packaging Conventions

**OS** Operative System

**PDF** Portable Document Format

**PKC** Public Key Cryptology

**PKCS** Public Key Cryptography Standards

**PKI** Public Key Infrastructure

**SDK** Software development kit

**SHA** Secure Hash Algorithm

**UI** User Interface

**WBS** Work Breakdown Structure

**XAdES** XML Advanced Electronic Signature

**XML** eXtensible Markup Language

# Capítulo 1
# Introdução

Este documento constitui o relatório final de estágio em Engenharia Informática, na especialidade de Engenharia de Software na Faculdade de Ciências e Tecnologia da Universidade de Coimbra. O estágio está a decorrer na empresa Present Technologies, sob a orientação do DEI proveniente do Professor Doutor Fernando Barros e sob a orientação de Avelino Martins da Present Technologies.

## 1.1. Âmbito

Geralmente, para autenticar e verificar a validade de documentos é necessária uma assinatura manuscrita, por exemplo uma convenção entre dois países, um contrato entre duas empresas ou para fazer transferência bancária.

Como resultado da constante expansão do mercado digital, as pessoas têm vindo cada vez mais a optar pelas assinaturas de documentos, contratos ou acordos de forma remota e rápida através da internet, nascendo então a assinatura digital na criptologia, baseada nos rápidos avanços da PKI [1] (Criptologia de chave pública). Com a necessidade de fazer assinaturas digitais de forma remota e rápida surge então o propósito de aplicar este conceito para terminais móveis de forma a que seja possível assinar documentos duma forma tão segura quanto feita num terminal fixo.

## 1.2. Objetivos

Inicialmente o projeto de estágio incluía o desenvolvimento de uma aplicação móvel para iOS e Android para assinatura de documentos PDF e Word usando o Cartão de Cidadão Português. No entanto, devido à extensão do projeto caso fosse necessário implementar também em iOS, o tema do projeto foi limitado à assinatura de documentos PDF e Word usando o Cartão de Cidadão Português em Android.

Esta aplicação permitirá ao utilizador a assinatura de documentos PDF e Word através do seu *smartphone* Android usando para isso os dados privados armazenados no Cartão de Cidadão.

## 1.3. Entidade Acolhedora

A Present Technologies é uma empresa de TI com sede em Coimbra, especializada no desenvolvimento de serviços e aplicações móveis, Web Apps e soluções *enterprise*. Fundada em 2000, por colegas investigadores da Universidade de Coimbra, a Present Technologies conta com clientes em diferentes países de todo o mundo. A empresa encontra-se em atualmente em expansão, contando agora com escritórios também no Porto e foi considerada umas das 100 melhores empresas para trabalhar em 2016 e 2017 pela revista Exame.

## 1.4. Estrutura do documento

Este relatório encontra-se dividido em sete capítulos distintos, nomeadamente a Introdução, Estado da Arte, Análise de Requisitos, Planeamento, Arquitetura e Trabalho Desenvolvido, Validação de Requisitos e Conclusões e Trabalho Futuro.

O primeiro capítulo apresenta a introdução ao projeto e à temática que lhe é adjacente. No segundo capitulo são estudados os aspetos mais relevantes ao tema a desenvolver, bem como uma escolha justificada das várias ferramentas a utilizar e estudo de aplicações semelhantes à do tema proposto. O terceiro capítulo trata da recolha de requisitos feita pelo estagiário e das várias metodologias utilizadas para o fazer. O quarto capítulo trata da arquitetura criada, mais precisamente os atributos de qualidade, restrições, avaliação da arquitetura proposta e arquiteturas alternativas. O quinto capitulo trata do planeamento efetuado pelo estagiário face aos requisitos levantados, bem como uma análise de riscos e respetivos planos de mitigação. O sexto e último capitulo apresenta as conclusões relativas a este semestre.

# Capítulo 2
# Estado da Arte

Neste capítulo são referidos vários aspetos relevantes estudados durante o primeiro semestre de estágio. Inicialmente foi feita uma análise dos vários componentes necessários à criação de assinaturas digitais, documentos PDF e Word e sobre o Cartão de Cidadão Português. Em seguida foi elaborado um estudo comparativo das aplicações em Android existentes no mercado, semelhantes à que se pretende desenvolver neste estágio. Por fim foi feito um estudo comparativo entre várias ferramentas a utilizar neste projeto.

## 2.1. Assinaturas digitais

De acordo com o artigo da Microsoft [2], de forma simples e generalizada, uma assinatura digital pode ser definida como "um selo eletrónico e encriptado de autenticação de informações digitais, tais como mensagens de correio eletrónico, macros ou documentos eletrónicos". A principal funcionalidade de uma assinatura digital pode ser definida como uma forma de garantir que as informações têm origem no signatário e que estas não foram alteradas após terem sido assinadas. Em seguida irão ser referidas algumas das principais características de uma assinatura digital e quais as diferenças existentes entre uma assinatura eletrónica e uma assinatura digital.

Uma assinatura digital possui como principais características:

1. **Integridade:** Garante que o artefacto assinado não foi alterado durante o *workflow* do negócio.
2. **Autenticidade:** Garante que o autor da assinatura é realmente quem pensamos que seja.
3. **Não-repúdio:** Garante que o autor da assinatura não consiga negar que foi ele que assinou.

### 2.1.1. Assinaturas eletrónicas vs assinaturas digitais

Embora seja comum o uso do termo "assinaturas eletrónicas" e "assinatura digital" como se fosse o mesmo, a verdade é que não o são. As assinaturas eletrónicas (*e-signatures*) referem-se a qualquer método eletrónico que indica a concordância de um acordo ou registo [3]. Uma assinatura eletrónica também pode ser vista como uma forma de representar uma assinatura num meio digital, por exemplo uma assinatura feita à mão através de um tablet num documento PDF ou até uma assinatura feita à mão num dispositivo digital quando se recebe uma encomenda.

Uma assinatura digital é muito mais que uma assinatura eletrónica. As assinaturas digitais usam um método específico para assinar documentos eletronicamente, nomeadamente IDs digitais baseados em certificados que permitam garantir a autenticidade da identidade do assinante. Também demonstram a prova de assinatura ligando cada assinatura ao documento através de encriptação. Devido a esta "ligação" entre assinatura e documento, qualquer um que tente alterar o conteúdo do documento depois de este ter sido assinado irá invalidar a assinatura [4].

A Tabela 1 facilita a análise análise comparativa entre estes dois tipos de assinatura.

Tabela 1- Assinaturas digitais VS Assinaturas Eletrónicas adaptada de [4].

| **Assinaturas Digitais** | **Assinaturas Eletrónicas** |
|---|---|
| Se um documento for alterado depois de assinado, a assinatura irá tornar-se inválida. | Uma assinatura eletrónica é facilmente falsificada. |
| A *hash* de um documento é praticamente impossível de reverter e a encriptação por certificados é altamente segura. | Não são baseadas em standards de segurança e tendem a usar métodos proprietários que são menos seguros. |
| Uma assinatura está associada a uma chave privada de um certificado digital, logo é difícil negar que foi essa pessoa que assinou. | Mais difícil de provar que foi realmente o assinante que assinou. |
| Muitas das vezes incluem um *timestamp* associado à assinatura que garante que a assinatura foi feita mesmo num determinado dia a uma determinada hora. | Tem data e hora, mas não está armazenada num local diferente da assinatura, logo está aberta a falsificações. |
| O certificado fornece informações pessoais tais como nome, email e nome da empresa que também são inseridos na assinatura através do certificado. | Se forem necessários dados da pessoa que assinou, estes têm que ser guardados fora da assinatura, estando mais sujeitos a falsificações. |

Analisando a Tabela 1 pode-se facilmente concluir que existem muitas falhas presentes na assinatura eletrónica. Isto deve-se ao facto de as assinaturas eletrónicas não usarem *hashing* nem criptografia de chave pública no seu processo de assinatura, não garantindo integridade, autenticidade e não-repúdio. Em seguida irá ser explicado e exemplificado o funcionamento geral de uma assinatura digital.

### 2.1.2. Funcionamento geral de uma assinatura digital

#### 2.1.2.1. Tecnologias de segurança relacionadas

Antes de entrar nas assinaturas digitais, deve-se ter uma noção das tecnologias de segurança relacionadas com as mesmas, tais como certificados, *Public Key Infrastructure* (PKI) e o algoritmo de *hashing Secure Hash Algorithm* (SHA).

##### 2.1.2.1.1. Secure Hash Algorithm

Em criptografia o SHA é uma função criptográfica de *hashing* concebida pela *National Security Agency* nos Estados Unidos e é uma norma de processamento de informações federal.

Dependendo do tipo de SHA, quando aplicado produz um *hash value* com determinado tamanho chamado de *message digest* [17], sendo este normalmente representado por um valor hexadecimal (40 carateres para SHA-1). O SHA é necessário para uso com o *Digital Signature Algorithm* (DSA) conforme especificado no *Digital Signature Standard* (DSS) e é sempre necessária a aplicação de um algoritmo de *hashing* para aplicações federais. O SHA é usado tanto pelo transmissor como pelo recetor no processo de computação e validação de uma assinatura digital [19].

Na Tabela 2 é possível observar os vários tipos de SHA, o tamanho dos seus *message digest*, estado de segurança e sua data de publicação.

Tabela 2- Tipos de *Secure Hash Algorithms* e suas propriedades adaptada de [17].

| **Tipo (e sua variante)** | **Tamanho de *message digest* (*bits*)** | **Segurança** | **Data de publicação** |
|---|---|---|---|
| SHA-0 | 160 | Baixa | 1993 |
| SHA-1 | 160 | Baixa | 1995 |
| SHA-2 | | Alta | 2001 |
| SHA-224 | 224 | | |
| SHA-256 | 256 | | |
| SHA-384 | 384 | | |
| SHA-512 | 512 | | |
| SHA-512/224 | 224 | | |
| SHA-512/226 | 226 | | |
| SHA-3 | | Alta | 2015 |
| SHA3-224 | 224 | | |
| SHA3-256 | 256 | | |
| SHA3-384 | 384 | | |
| SHA3-512 | 512 | | |

Com a evolução do poder computacional alguns dos algoritmos como o SHA-0 e SHA-1 têm vindo a perder a sua relevância, passando agora o seu testemunho aos algoritmos mais recentes como o SHA-2 e SHA-3. No entanto o SHA-1 ainda é bastante usado e só há pouco tempo empresas como *Google*, *Apple* e *Microsoft* estão a deixar de aceitar certificados SSL que usem SHA-1 nos seus *browsers*.

#### 2.1.2.1.2. Public Key Infrastructure

Uma *Public Key Infrastructure* (PKI) global é um pré-requisito de segurança em grandes redes, sistemas distribuídos e comércio eletrónico. Uma PKI consiste num conjunto heterogéneo de componentes que podem estar envolvidos na emissão, armazenamento e/ou distribuição de certificados. Pode ser vista como uma base de dados distribuída de certificados de chave pública e informações adicionais como *Certificate Revocation Lists* (CRLs) [18].

Quando se assina um documento em papel é necessário um notário ou alguém em que se possa confiar para presenciar o momento. Como o notário é alguém de confiança, é possível confiar na assinatura que este presencia, numa PKI acontece o mesmo [16].

Na Figura 1 podemos observar os principais elementos constituintes de uma PKI, tais como *Certificate Authority* (CA), *Intermediate Certificates*(ICAs) e *End Entity Certificate* (EE). Em seguida serão explicadas quais as funções dos vários elementos presentes numa PKI.

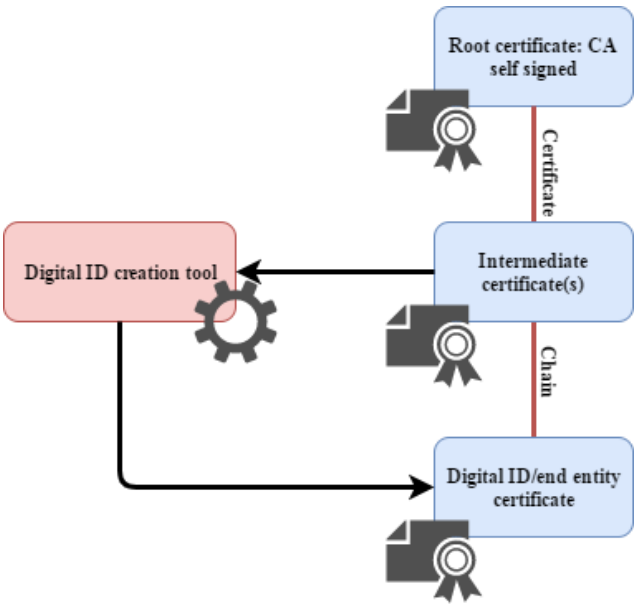



Figura 1- Visão geral de uma *Public Key Infrastructure* adaptada de [16].

***Certificate Authority:*** É a autoridade mais alta que vende ou publica os *Digital IDs*, tal como a *Verisign* ou a *Geotrust*. A CA assina o seu próprio certificado e este certificado é tipicamente a raiz de uma cadeia de certificados, ou *certificate chain*.

***Intermediate Certificates:*** É um tipo de CA intermédio no qual os seus certificados se situam entre a CA e a *End Entity Certificate*. Estes certificados já não são auto assinados e a principal função dos ICAs é fornecer serviços tais como políticas, *timestamps*, *Certificate Revocation Lists* (CRLs) etc.

***End Entity Certificate***: Representa o certificado do assinante e corresponde ao último elemento da uma *certificate chain*.

***Digital ID:*** É a representação eletrónica da informação, associada a uma pessoa ou entidade. Normalmente está armazenada num ficheiro protegido por *pass*Word ou num *token USB*, *smartcard* etc. Um *Digital ID* contém um *public key certificate*, *private key* e mais informação relativa ao proprietário do *Digital ID*.

***Public Key Certificate:*** É o ficheiro que contém a representação numérica da chave pública, bem como os restantes dados que permitem identificar o dono do certificado.

***Private Key:*** Corresponde ao par da *public key* e tem como principal funcionalidade de assinar conteúdo digital, conteúdo este que só pode ser desencriptado pela *public key* correspondente.

***Certificate Chain*:** É a lista de certificados com o inicio no o certificado da *Certificate Authority* ao *End Entity Certificate* inclusive.

***Digital ID creation tool*:** É a ferramenta usada para gerar o *Digital ID* para as *End Entity* presentes na PKI.

#### 2.1.2.1.2. Criação da assinatura

A aplicação da assinatura digital pode ser descrita por dois passos, nomeadamente o processo de *hashing* e a encriptação do *hash* resultante.

O *hashing* consiste na computação da *hash* de algo a ser assinado. Este *hashing* consiste na execução de um algoritmo matemático sobre um valor, por exemplo um conjunto de bytes. Os algoritmos de *hashing* mais comuns nas assinaturas digitais são o SHA-1, SHA-256, SHA-384 e SHA-512. O principal objetivo destes algoritmos é garantir a integridade do ficheiro onde eles são aplicados, por exemplo, o *hash* de um ficheiro dará sempre o mesmo resultado a não ser que algo tenha sido alterado desde que o *hash* foi calculado sobre esse ficheiro. Na Figura 2 é possível observar o funcionamento geral da aplicação de um algoritmo de *hashing* sobre um documento.


Figura 2- Aplicação de um algoritmo de *hash* SHA-1 sobre um ficheiro.

Em seguida à aplicação do *hash* sobre um documento, segue-se a fase de encriptação. Esta fase consiste na encriptação do *hash* calculado no passo anterior utilizando da chave privada do assinante. O conjunto resultante da *hash* encriptada e o certificado publico origina uma assinatura digital. A assinatura digital é então "anexada" ao documento e é de notar que este passo de anexação varia de documento para documento, sendo para PDF de uma maneira e para Word de outra com ligeiras diferenças, no entanto a lógica usada é bastante semelhante. Este tipo de anexação da assinatura ao documento irá ser explicada nas respetivas secções deste capítulo com mais detalhe.

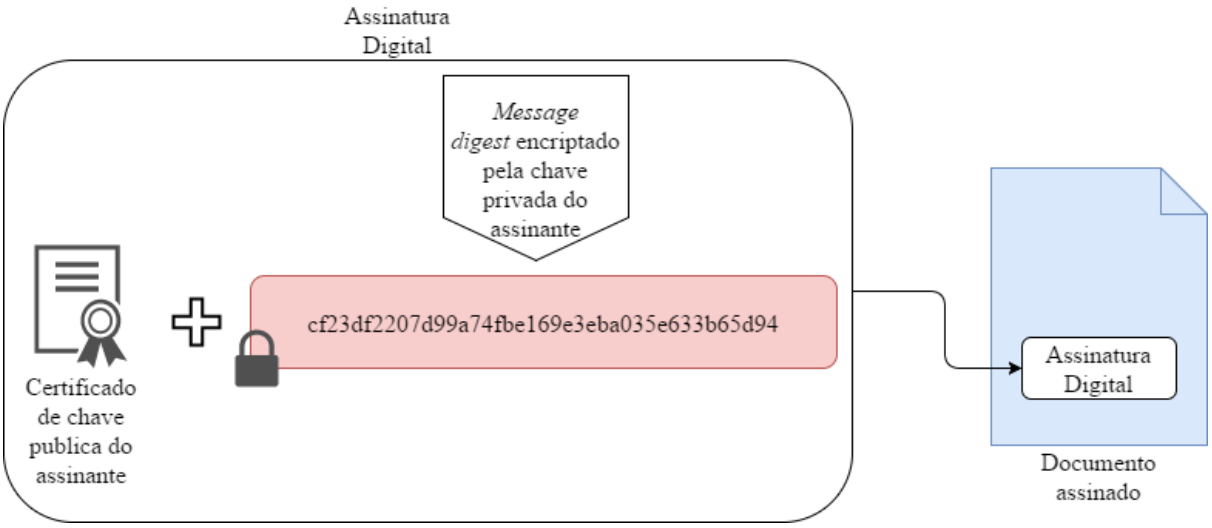



Figura 3- Criação de uma assinatura digital e sua anexação ao documento a assinar.

#### 2.1.2.1.3. Verificação da assinatura

A validação será feita quando um documento assinado é aberto através de um programa de leitura. No entanto, embora seja possível validar assinaturas programaticamente, a validação normalmente é feita pelo *software* de leitura de documentos que suportem assinaturas digitais, por exemplo o Adobe Acrobat Reader ou o Microsoft Office.

Quando o programa abre o documento assinado é utilizada a chave pública presente no documento para desencriptar o *hash* que foi calculado e adicionado ao documento na criação da assinatura. O programa então volta a calcular um outro *hash* do documento e compara-o com o *hash* desencriptado, se forem iguais significa que o documento não foi alterado depois de ter sido assinado.

Na Figura 4 podemos observar a ilustração deste processo de forma mais clara.

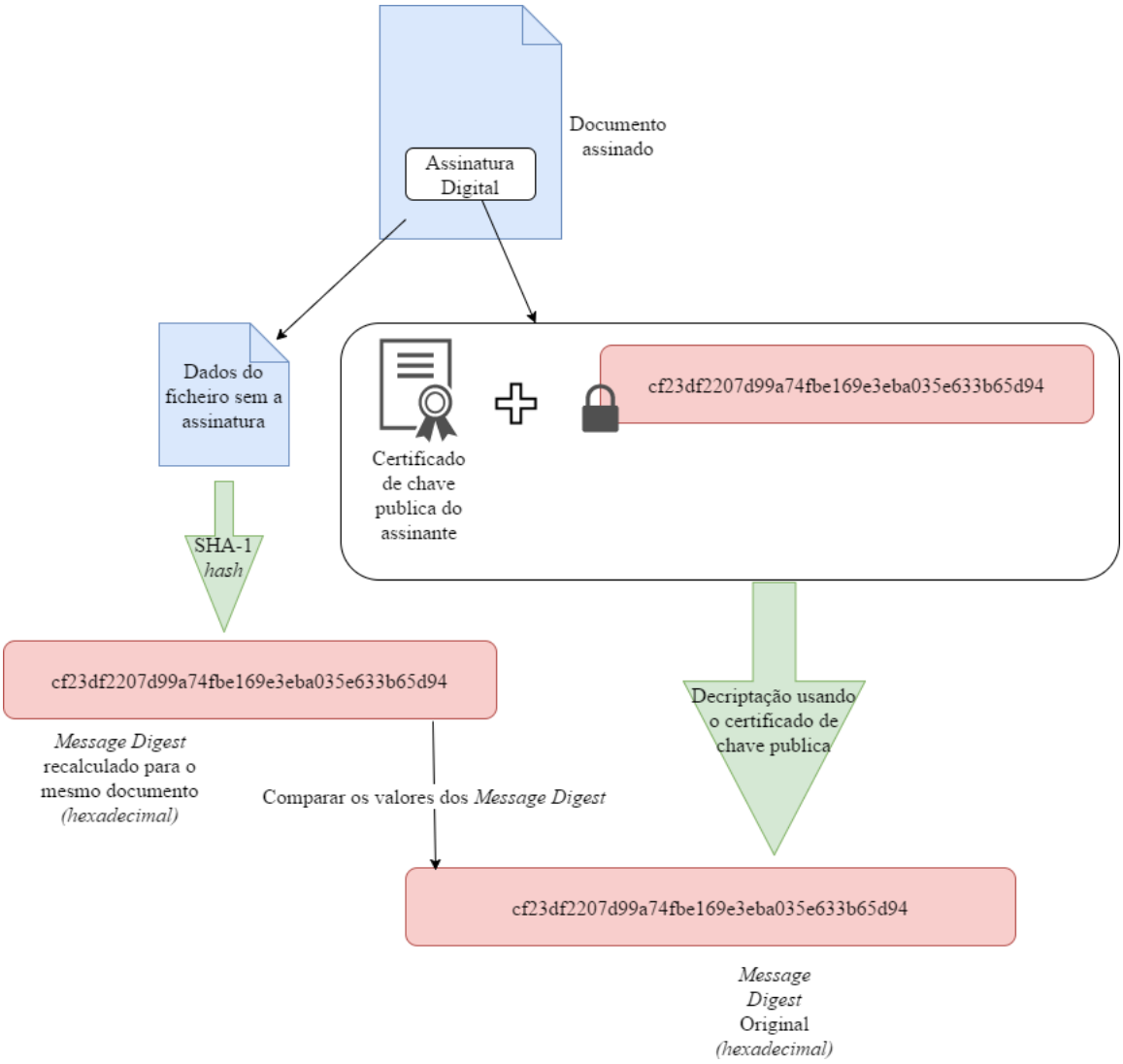



Figura 4- Processo de validação de uma assinatura num documento.

### 2.1.3. Assinaturas digitais perante a lei

As assinaturas digitais são legalmente aceites em quase todas as nações industrializadas. No ano 2000 os Estados Unidos aprovaram o ato *Electronic Signatures in Global National Commerce* (ESIGN), tornando as assinaturas digitais legais para todos os fins virtuais.

No ano 1999 já existia uma diretiva relativamente a assinaturas digitais na União Europeia (EU), diretiva esta publicada pela Comissão Europeia, denominada por 1999/93/EC. Visto que se tratava de uma diretiva e não de uma regulamentação, permitia aos membros da EU interpretar a nova lei e impor-lhe as suas restrições, limitações e exceções [6]. Em 2011 surgiu então a necessidade de criar uma regulamentação em vez de uma diretiva de forma a uniformizar a lei existente relativamente às assinaturas digitais em todos os membros da EU, promovendo assim o desenvolvimento de um mercado digital europeu. Em 2014 esta nova regulamentação foi chamada de *Eletronic Identification and Trust Service Regulation* [8], também conhecida como eIDAS (ou Regulamentação 910/2014/EC). A eIDAS só passou a ter efeito no dia 1 de julho de 2016, substituindo a diretiva 1999/93/EC criada anteriormente, referindo no seu artigo 25 de que todas as "assinaturas eletrónicas" e sistemas de verificação serão admissíveis como prova em procedimentos legais [7], tendo assim o mesmo poder que as assinaturas manuscritas.

## 2.2. Documentos PDF

A *International Organization for Standardization* (ISO) 32000 especifica a forma digital para representar documentos chamada de *Portable Document Format* (PDF). PDF foi desenvolvido pela *Adobe Systems* no início de 1993. Com o crescimento explosivo da internet, o PDF tem vindo a ser cada vez mais usado para a troca de documentos no meio eletrônico. Existem várias aplicações específicas de PDF que têm vindo a evoluir, limitando o uso de alguns recursos do PDF e exigindo o uso de outros, aumentando a utilidade do PDF [11].

### 2.2.1. Normas relevantes

A ISO 32000 é a norma para todo o funcionamento de um PDF, no entanto existem outras como a ISO 15930, ISO 19005 e a ISO 24517 para uso mais especializado.

A ISO 15930 [12] é agora o padrão da indústria para a representação intermediária de material impresso em sistemas de pré-impressão eletrónicos para aplicações de impressão convencionais. A ISO 19005 [13] é agora o padrão da indústria para o arquivamento de documentos digitais. A ISO 24417 [14] fornece um mecanismo para representar documentos e troca de dados de engenharia.

Como grandes empresas, agências governamentais e instituições educacionais simplificam suas operações ao substituir o fluxo de trabalho baseado em papel por troca eletrônica de informações, o impacto e a oportunidade para a aplicação do PDF continuarão a crescer rapidamente.

### 2.2.2. Estrutura interna

Tal como todos os documentos digitais, o PDF possui a sua estrutura característica. Devido à sua extensão e complexidade, será apenas estudada a sua estrutura de uma forma mais abstrata, focando mais nas assinaturas digitais dentro de um documento PDF. Na Figura 5 é possível observar um exemplo da estrutura interna de um documento PDF.

```
%PDF-1.x
%âãÏÓ
1 0 obj
...

2 0 obj
... (Hello World) Tj ...

xref
0 81
0000000000 65535 f
0000000015 00000 n
...
trailer
<< ... >>
startxref
15787
%%EOF
```

Figura 5- Estrutura interna de um documento PDF retirada de [15].

A Figura 5 permite ter uma ideia do quanto um documento PDF é ilegível internamente. No entanto, a estrutura interna dos documentos PDF pode ser alterada de forma fácil, surgindo a necessidade de preservar a sua integridade, para que quando alterado algures no *workflow* do negócio, seja possível detetar que o documento foi modificado. É aqui onde as assinaturas digitais entram.

Um PDF possui vários tipos de objetos no seu interior, os objetos do tipo dicionário são os objetos mais relevantes nas assinaturas digitais em PDF. Na Figura 6 é possível observar um objeto do tipo *Signature Dictionary*.

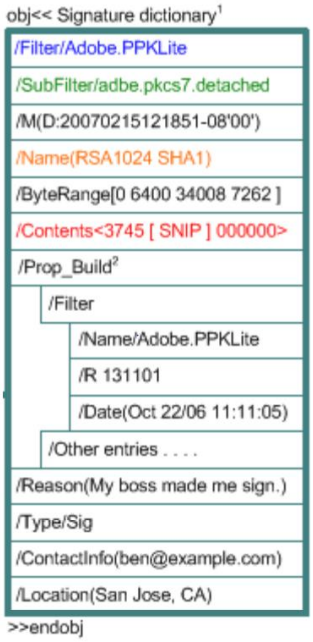



Figura 6- *Signature Dictionary* retirada de [16].

Dentro deste dicionário existem vários elementos, tais como o *Filter, SubFilter, Contents, Name* etc. De acordo com a ISO 32000, os elementos obrigatórios de uma assinatura digital são: *Filter, Contents* e *ByteRange*. Os elementos *Cert* e *SubFilter* podem ser obrigatórios ou não, dependendo do tipo de assinatura a inserir no documento.

Em seguida serão explicados com mais detalhe os elementos de um *Signature Dictionary* e alguns conceitos importantes relacionados com os mesmos, tais como o *PKCS#7*.

- ***Filter:*** Nome do *Signature Handler* a usar para quando a assinatura for válida.
- ***SubFilter:*** Um nome que descreva o *encoding* do valor da assinatura e informação referente às chaves no *Signature Dictionary*. Um leitor de PDF pode usar qualquer *handler* que suporte este formato para validar a assinatura. Os valores *adbe.x509.rsa_sha1, adbe.pkcs7.detached* e *adbe.pkcs7.sha1* devem ser usados.
- ***Contents:*** Representa o valor da assinatura no formato *byte string*. Para assinaturas de chave publica, este elemento deve ser um objeto binário *PKCS#1* no formato *Distinguished Encoding Rules (DER)* ou um objeto *PKCS#7* também no formato *(DER)*. Tanto o PKCS#1 como o PKCS#1 são *Public Key Cryptography Standards*. O #1 fornece definições e recomendações básicas para implementar o algoritmo RSA. O #7, também conhecido como *Cryptographic Message Syntax* (CMS) tem como funcionalidade de descrever uma sintaxe para mensagens, podendo ter criptografia aplicada a ele, tais como assinaturas digitais ou envelopes digitais.

Espaço para o *Contents* deverá ser alocado antes que o *message digest* seja computado.

- ***Cert:*** Um *array* de *byte strings* que representa a *certificate chain* usada ao assinar ou a verificar assinaturas que usam criptografia de chave publica, ou um *byte string* caso a *certificate chain* só tenha um certificado. Caso o *SubFilter* for do tipo *adbe. pkcs7.detached* ou *adbe.pkcs7.sha1*, este elemento não será usado e o *certificate chain* será inserido num envelope *PKCS#7* no elemento *Contents*.
- ***ByteRange:*** É um *array* de quatro números. O primeiro número de cada par é o *offset* no ficheiro do início de uma *byte stream* a ser incluída no *hashing*. O segundo número é o tamanho dessa *stream*. Os dois pares definem duas sequências de *bytes* que definem o que irá ser *hashed*. Para entender melhor o que é o *ByteRange* é melhor observar para a Figura 7.

```
<<
/Type /Catalog
/Pages 4 0 R
/OpenAction [1 0 R /XYZ null null 0]
/Lang (pt-PT)
/AcroForm <<
/Fields [14 0 R]
/SigFlags 3
>>
>>
endobj
14 0 obj
<<
/FT /Sig
/Type /Annot
/Subtype /Widget
/F 132
/T (Signature1)
/V 15 0 R
/P 1 0 R
/Rect [0.0 0.0 0.0 0.0]
>>
endobj
15 0 obj
<<
/Type /Sig
/Filter /Adobe.PPKLite
/SubFilter /adbe.pkcs7.detached
/Name (Example User)
/Location (Los Angeles, CA)
/Reason (Testing)
/M (D:20170504183755+01'00')
/Contents <308006092A239EF170946D84125AF...0000000000000>
/ByteRange [0 8030 26976 316]
>>
endobj
xref
0 1
0000000000 65535 f
12 1
0000007573 00000 n
14 2
0000007719 00000 n
0000007846 00000 n
trailer
<<
/Size 16
/Root 12 0 R
/Info 13 0 R
/ID [<412088BDE1A5407C1F18CC883486C527> <AD651A840C3A6954196C006FF325FB11>]
/Prev 7088
>>
startxref
27035
%%EOF
```

Figura 7- *ByteRange* e restantes elementos de um *Signature Dictionary* dentro de um PDF.

O *ByteRange* funciona como um *snapshot* do documento, que guarda o tamanho do mesmo em *bytes* exceto o valor da assinatura presente no elemento *Contents* (dentro dos < >). A este *snapshot* é que será feito o *hash* usando o algoritmo SHA para posterior encriptação usando a chave privada do utilizador.

### 2.2.3. Assinatura digital

Após analisar a estrutura de um documento PDF e os aspetos mais relevantes relativamente a assinaturas digitais, é possível compreender com mais facilidade o *workflow* da aplicação de uma assinatura digital neste tipo de documento.

O processo de assinatura PDF pode ser descrito da seguinte forma:

1. O documento a assinar é convertido numa *byte stream*.
2. Um *placeholder* da assinatura digital é criado dentro do elemento *Contents* "reservando" o espaço onde a futura assinatura irá ser colocada. Este *placeholder* é constituído apenas por zeros com um *worst case value* do tamanho final da assinatura a ser colocada.
3. Agora que já foi reservado espaço para a assinatura, é calculado um *ByteRange* do documento a assinar, excluindo o *placeholder* da assinatura.
4. Visto que já temos o *ByteRange*, é feito então o *hashing* recorrendo ao algoritmo SHA aplicado no *Byterange*, obtendo agora um *message digest*.
5. Este *message digest* é então encriptado usando a chave privada do assinante, obtendo agora o *encripted message digest*.
6. A este *encripted message digest* é adicionado a *certificate chain* do assinante, originando um envelope *PKCS#7* que constitui a assinatura digital.
7. Por último, agora é necessário inserir este envelope *PKCS#7* no elemento *Contents* do *Signature Dictionary* do documento a assinar, obtendo então o documento PDF assinado.
8. Este processo é quase sempre igual, apenas irá possuir algumas diferenças consoante o nome do elemento *SubFilter* presente no *Signature Dictionary* como foi referido anteriormente.

## 2.3. Documentos Word

### 2.3.1. Estrutura de um documento .docx

O formato .docx foi introduzido no *Microsoft* Word *2007* e é baseado em *OpenXML*, usando compressão ZIP para obter tamanhos reduzidos nos seus ficheiros. A vantagem de ser baseado em *OpenXML* é que permite a outras aplicações ler o seu conteúdo facilmente. Visto que o formato .docx é baseado no *OpenXML*, é possivel analisar a estrutura deste tipo, de forma a compreender a estrutura de um ficheiro .docx.

#### 2.3.1.1. Open Packaging Conventions

*A ECMA-376 OpenXML, 1st Edition, Part 2: Open Packaging Conventions* (OPC) pode ser mais facilmente compreendida através de uma analogia aos sistemas de organização/ armazenamento de ficheiros do mundo real. Quando o armazenamento de ficheiros em formato tradicional é utilizado numa empresa, é necessário que estes ficheiros estejam devidamente organizados para futura consulta. Um sistema deste género é facilmente utilizado por pessoas que que estão habituadas ao mesmo e que sabem como este funciona. No entanto, caso alguém que não conheça o sistema queira utilizar, já não será tão fácil. As aplicações passam por problemas semelhantes no que toca a organizar a sua informação, nomeadamente a informação mais importante e a facilidade com que esta poderá ser acedida por outras aplicações.
A especificação OPC veio a resolver este problema, integrando elementos de tecnologia ZIP, *XML* e *Web*, definindo normas de organização, armazenamento e transporte de dados de forma a tornar o acesso aos mesmos estandardizado e previsível.

Documentos .docx ou OXML são *containers* armazenados como arquivos ZIP. A maioria dos ficheiros dentro deste *container* são do formato XML, à exceção de imagens que têm formato binário. Cada um dos XML presentes num OXML têm funções específicas, por exemplo, um ficheiro XML guarda o conteúdo do .docx, outro guarda estilos, outro poderá guardar metadata etc. No entanto estes ficheiros não possuem uma diretoria fixa.

#### 2.3.1.2. Relações entre *parts*

De acordo com a OPC, é definida uma camada adicional de abstração sobre a diretoria do ZIP, na qual deve ser consultada de forma a determinar as diretorias presentes para os ficheiros [21].

O modelo de pacotes especificado pela OPC descreve *packages, parts* e relações. Os packages contêm *parts*, e as *parts* contêm conteúdo e recursos. As relações estão definidas para ligar *packages* às *parts* e para ligar várias *parts* dentro do *package.* As *parts* estão organizadas como um grafo direcionado, no entanto para ficar mais claro irá ser representado em formato de árvore como presente na Figura 8.

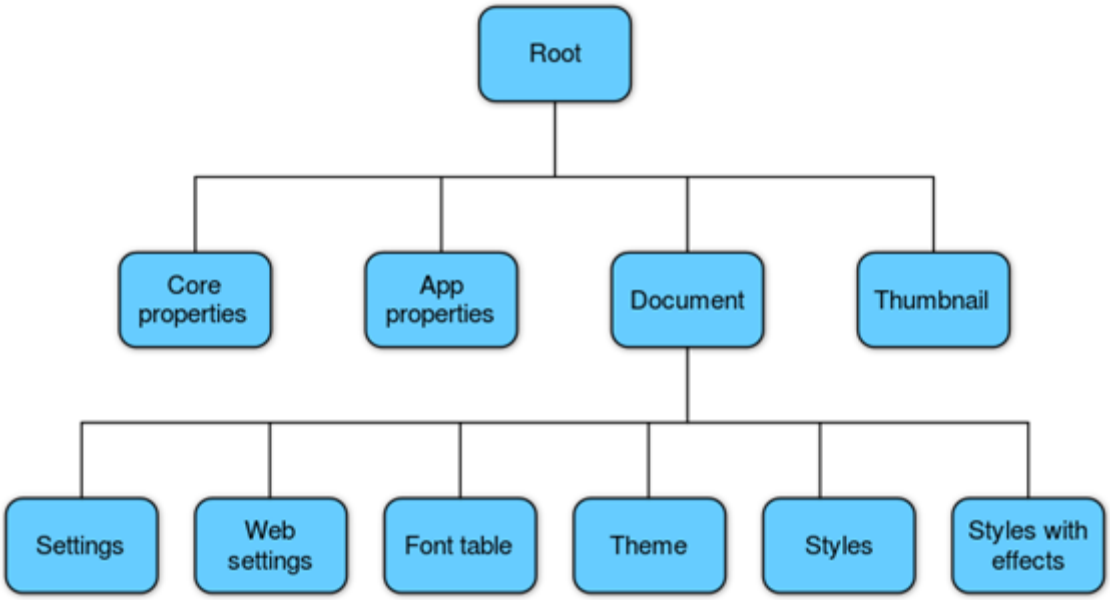



Figura 8- Representação em árvore de diretorias de um .docx retirada de [21].

Cada caixa azul, à exceção da caixa *root* representa uma *part* num OPC *package*. Existe um ficheiro XML que contém as relações relativamente à diretoria onde se encontra, este ficheiro é chamado de *_rels(filename).rels*. Por exemplo para o ficheiro *document.xml* a diretoria do ficheiro de relações seria /Word/*_rels*/*document.xml.rels* e nela estaria a localização dos ficheiros *settings.xml, webSettings.xml, fontTable.xml* etc. Na Figura 9 está representado o conteúdo do ficheiro *document.xml.rels* :

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship
    Id="rId3"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings"
    Target="settings.xml"/>
  <Relationship
    Id="rId4"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings"
    Target="webSettings.xml"/>
  <Relationship
    Id="rId5"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable"
    Target="fontTable.xml"/>
  <Relationship
    Id="rId6"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme"
    Target="theme/theme1.xml"/>
  <Relationship
    Id="rId1"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles"
    Target="styles.xml"/>
  <Relationship
    Id="rId2"
    Type="http://schemas.microsoft.com/office/2007/relationships/stylesWithEffects"
    Target="stylesWithEffects.xml"/>
</Relationships>
```

Figura 9- Ficheiro. rels do *document.xml* retirado de [21].

### 2.3.2. Assinatura digital

Uma assinatura digital num documento .docx é basicamente uma assinatura XML. Esta assinatura é inserida no documento .docx como um package e *part*, tal como qualquer uma *part* normal que constitua o documento. À semelhança de uma assinatura em documentos PDF, aqui também é feito um *hashing*, encriptação do *message digest* e fornecimento dos certificados do assinante e sua respetiva inserção dentro do documento XML. A grande diferença aqui está no facto de que o conteúdo a assinar já não será um *ByteRange* mas sim todas as *parts, packages* e até relações que constituem o .docx. Logo torna-se fundamental analisar o funcionamento de uma assinatura XML e suas variantes para conseguir obter uma assinatura digital num documento .docx.

#### 2.3.2.1. Assinatura digital XML

A XML-DSIG definida pela IETF/*W3C Recommendation for XML Signature Syntax and Processing* já existia desde 2002. Esta XML-DSIG especifica a sintaxe e regras de processamento para criar e representar assinaturas digitais. As assinaturas XML podem ser aplicadas a qualquer conteúdo digital, incluindo XML. Existem três variantes de uma XML-DSIG, nomeadamente *enveloped*, *enveloping* e *detached*. As XML-DSIG *enveloped* e *enveloping* são feitas sobre dados dentro do mesmo ficheiro XML que representa a assinatura, enquanto *detached* são sobre dados externos ao XML da assinatura [22]. Uma assinatura XML-DSIG normal tem a estrutura semelhante à referida na Figura 10.

```
<Signature>
  <SignedInfo>
    <SignatureMethod />
    <CanonicalizationMethod />
    <Reference>
       <Transforms />
       <DigestMethod />
       <DigestValue />
    </Reference>
    <Reference /> etc.
  </SignedInfo>
  <SignatureValue />
  <KeyInfo />
  <Object />
</Signature>
```

Figura 10- Representação de uma XML-DSIG retirada de [24].

#### 2.3.2.1.1. XML Advanced Electronic Signature

A *XML Advanced Electronic Signature* (XAdES) estende especificação XML-DSIG para o domínio do não-repúdio definindo vários formatos XML para *Advanced Electronic Signatures* (AdES) que sejam de acordo com a diretiva 1999/93/EC do Parlamento Europeu [23] já referida anteriormente. Uma assinatura XAdES possui seis extensões, nomeadamente XAdES básica, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L e XAdES-A. Cada extensão adiciona propriedades à extensão anterior, onde a XAdES mais básica adiciona propriedades à XML-DSIG.

Em seguida irão ser explicadas as várias extensões de uma XAdES, indicando a principal característica de cada uma e o que é adicionado ao ficheiro XML respetivo.

**XAdES básico**

Proporciona autenticação e integridade e satisfaz os requisitos legais presentes na diretiva 1999/93/EC para. AdES, no entanto, não proporciona não-repúdio. À assinatura representada na Figura 10 acima devem ser adicionados os seguintes elementos:

QualifyingProperties

SignedProperties

SignedSignatureProperties

SigningTime

SigningCertificate

SignaturePolicyIdentifier

SignatureProductionPlace?

SignerRole?

SignedDataObjectProperties

DataObjectFormat*

CommitmentTypeIndication*

AllDataObjectsTimeStamp*

IndividualDataObjectsTimeStamp*

UnsignedProperties

UnsignedSignatureProperties

CounterSignature*

**XML Advanced Electronic Signature with Time-Stamp**

Tal como o próprio nome indica, a *XAdES-T* inclui *timestamp* e proporciona proteção contra repúdio. Esta adiciona o elemento *SignatureTimeStamp*+ dentro do elemento já presente *UnsignedSignatureProperties.*

**XML Advanced Electronic Signature with Complete validation data**

A *XAdES-C* inclui referências a um conjunto de informação que suporta a validação da assinatura, como por exemplo o estado de revogação do certificado. Isto é útil quando tal informação é armazenada externamente, nomeadamente por um *service provider* de confiança.

Esta extensão adiciona o elemento *CompleteCertificateRefs* e *CompleteRevocationRefs* ao elemento *UnsignedSignatureProperties.*

**XML Advanced Electronic Signature with eXtended validation data**

A *XAdES-X* inclui *time-stamp* nas referências de validação dos dados proporcionadas pela *XAdES-C*, bem como no elemento *Signature* . Isto é feito devido ao facto de que as chaves usadas numa *certificate chain* possam ser comprometidas.

Esta extensão possui duas alternativas, uma adiciona ao *UnsignedSignatureProperties* o elemento *RefsOnlyTimeStamp*. A segunda alternativa adiciona ao *UnsignedSignatureProperties* o elemento *SigAndRefsTimeStamp*.

**XML Advanced Electronic Signature with eXtended validation data Long term**

A *XAdES-X-L* inclui validação da informação para as situações onde a mesma não está armazenada externamente em longo prazo.

Esta extensão adiciona ao elemento *UnsignedSignatureProperties* os elementos *CertificatesValues* e *RevocationValues*.

**XML Advanced Electronic Signature with Archiving validation data**

A XAdES-A proporciona *time-stamps* adicionais para arquivar assinaturas de tal maneira que se encontrem protegidas mesmo que a informação criptografica se torne fraca.

Esta extensão adiciona ao elemento *UnsignedSignatureProperties* o elemento *ArchiveTimestamp*+. Como podemos observar, existem várias extensões de *XAdES* que adicionam novas funcionalidades às assinaturas XML, tornando-as mais e mais credíveis perante a lei consoante a sua extensão. Visto que cada uma das extensões referidas acrescenta funcionalidades à extensão anterior, pode-se concluir que a *XAdES-A* é a mais completa e segura das extensões referidas.

## 2.4. Cartão de Cidadão Português

O Cartão de Cidadão é um documento de cidadania que permite ao cidadão identificar-se de forma segura.

Para além de um documento de identificação físico, o Cartão de Cidadão é um documento eletrónico que possibilita a realização de várias operações sem necessidade de interação presencial.

O documento, em formato de *smartcard*, integra num documento único o Bilhete de Identidade, os cartões de identificação da Segurança Social, de Utente de Saúde e de Contribuinte [26].

O Cartão de Cidadão segue as normas internacionalmente recomendadas para que este seja um documento de identificação e um documento de viagem reconhecido oficialmente. Assim, este encontra-se alinhado pelas as orientações correntes da União Europeia, nomeadamente as do grupo de trabalho para o *European Citizen Card* (ECC), pelas normas definidas pela *International Civil Aviation Organization* (ICAO) para documentos de viagem internacionais (documento 9303) e os *standards* 7501 e 7810 definidos pela *International Organization for Standardization* (ISO)[25].

### 2.4.1. Atributos informáticos

É possível aceder a todos os atributos visíveis no cartão de cidadão através de aplicações específicas, à exceção da assinatura manuscrita. Existem também internamente dois pares de chaves usados para efeitos de encriptação, um par para autenticação e outro par para assinaturas digitais protegidos por *pin*. Dentro do cartão existem ainda os certificados de chave pública, dois deles relativos às chaves publicas de cada par e os restantes para construir as *certificate chains* necessárias na criação de assinaturas digitais. Na Figura 11 é possível observar esses certificados.

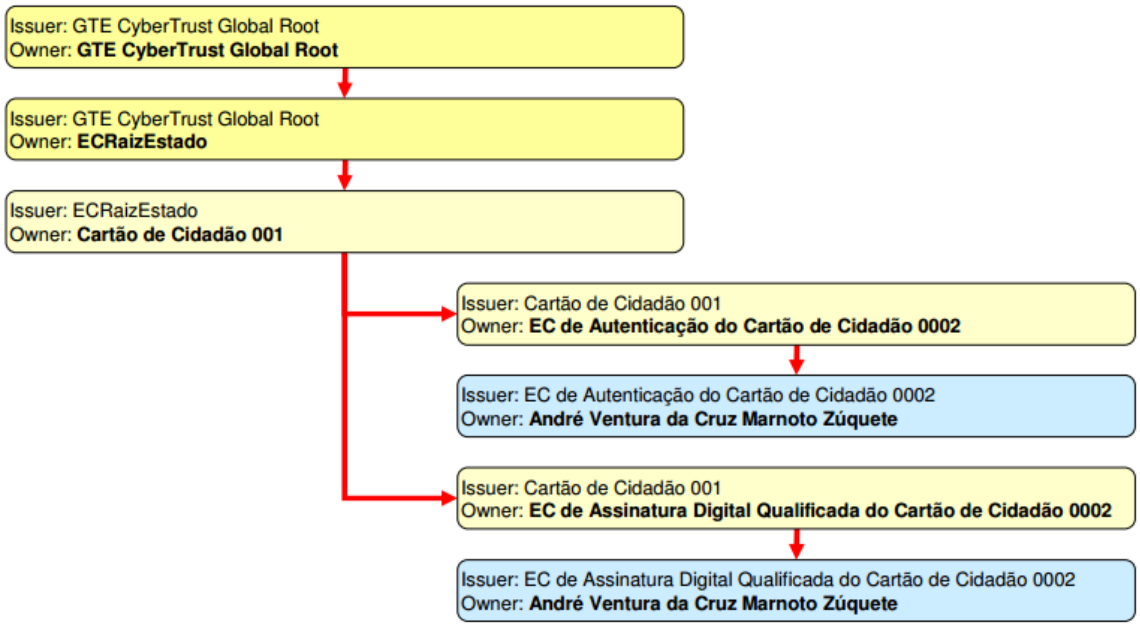



Figura 11- Certificados existentes no cartão de cidadão retirado de: http://sweet.ua.pt/andre.zuquete/Aulas/SIO/14-15/docs/CC.pdf.

Na Figura 11 é exemplificada a PKI já referida na secção 2.1.2. É possível observar parte da PKI existente onde a *Certificate Authority* é a *CyberTrust*, a *Intermediate Certificate* é o Estado Português e a *End Entity* somos nós, os cidadãos portugueses.

O cartão de cidadão ainda fornece serviços importantes para a PKI onde esta está inserido, nomeadamente OCSP, CRLs e LDAP. Estes serviços são importantíssimos porque são usados para validar o estado dos certificados antes de serem usados.

Um bom exemplo seria caso um cidadão perdesse o seu cartão e o fosse cancelar, seria imediatamente notificado ao OCSP e o mesmo alteraria o estado do cartão para revogado. Caso alguém quisesse assinar com o cartão perdido, na fase de verificação do estado do cartão o OCSP retornaria revogado, impedindo assim a assinatura ser feita com este cartão perdido.

**2.4.2.** As características mais importantes presentes no *chip* do Cartão de Cidadão são:

- *Chip Java Card*;
- Suporta o uso de *logical channels;*
- Possui CPU, ROM, EEPROM e RAM;
- Possui memória protegida, bem como gestão de memória e *garbage collection* pela JVM;
- Gestão de espaço de armazenamento;
- Suporta mecanismos de bloqueio em caso de erro na introdução de PIN;
- Possui um motor criptográfico interno que suporta:
    - Assinatura e verificação RSA de 1024 bits;
    - Algoritmos de *hash* MD5, SHA-1 e SHA-256;
    - PKCS#1.
- Está preparado para resistir aos ataques conhecidos do tipo *hardware attack, timing attack, simple power analysis* e *differential power analysis* entre outros;
- É compatível com leitores de cartões da norma EMV-CAP.

### 2.4.3. Acesso aos dados do cartão

Para aceder ao conteúdo do cartão, é necessário usar a interface *PKCS#11*. Em *Java* existe suporte para o uso desta interface, enquanto em aplicações Microsoft implementam a *CryptoAPI* e esta recorre internamente também à PKCS#11. Na Figura 12 ilustra o *middleware* associado ao acesso do cartão:

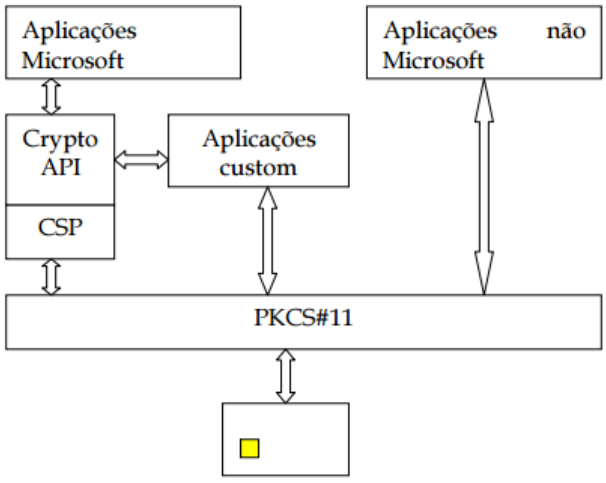



Figura 12- *Middleware* de acesso ao cartão de cidadão retirado de: https://www.autenticacao.gov.pt/documents/10179/11463/Manual+T%C3%A9cnico+do+Middleware+do+Cart%C3%A3o+de+Cidad%C3%A3o/07e69665-9f1a-41c8-b3f5-c6b2e182d697.

O acesso aos dados do cartão também pode ser feito com a utilização do *Application Protocol Data Unit* (APDU), no entanto esta é uma abordagem de muito baixo nível, consequentemente difícil de utilizar por programadores comuns.

### 2.4.4. Assinatura digital

Para assinar algo usando o cartão de cidadão é necessário recorrer aos métodos acima referidos de forma a aceder à chave privada e aos certificados de chave pública presentes, pois sem estes é impossível fazer uma assinatura digital. A chave privada nunca sai do cartão por razões de segurança e é mesmo esta a principal função do cartão. Cada vez que é necessário encriptar algo usando a chave privada do cartão, apenas é enviado um *hash* do conteúdo a encriptar. Este *hash* é feito externamente e o cartão apenas devolve este *hash* assinado usando a chave privada presente no cartão. A Figura 13 ilustra este procedimento e servirá também como diagrama de contexto mais à frente no capítulo 4.

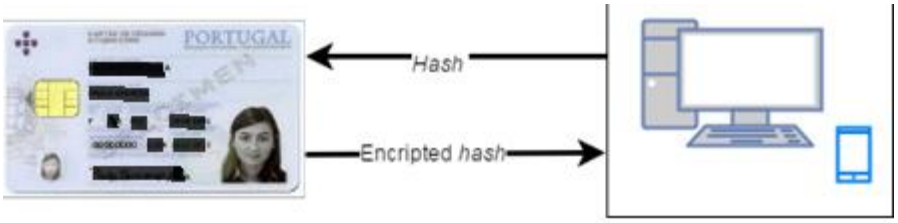



Figura 13- Funcionamento de encriptação de *hash* usando o cartão de cidadão.

## 2.5. Aplicações semelhantes existentes

Nesta secção encontra-se exposto o estudo de mercado feito pelo estagiário referente a aplicações semelhantes à que se pretende fazer neste estágio, nomeadamente aplicações em ambiente Android que permitam fazer assinaturas digitais usando o cartão de cidadão.

As características procuradas foram obviamente o uso do cartão de cidadão português, executar em Android e fazer assinaturas digitais. Na Tabela 3 é possível observar uma análise comparativa entre as *apps* existentes no mercado semelhantes à que se pretende desenvolver neste estágio.

A classificação de "?" indica que não foi possível obter mais informações relativamente a essa *app* nesse requisito. O símbolo "✓" indica que o determinado requisito se verifica, enquanto o símbolo "✗" indica que o mesmo não se verifica.

Tabela 3- Comparação de apps semelhantes presentes no mercado.

| Requisito / App | **Suporta Android** | **Usa CC** | **Faz assinaturas digitais** |
|---|---|---|---|
| DocuSign | ✓ | ✗ | ✓ |
| Digio | ✓ | ✗ | ✓ |
| SignEasy | ✓ | ✗ | ✓ |
| ComsignTrust | ✓ | ? | ✓ |
| Ipegs | ✓ | ✗ | ✓ |
| SigningHub | ? | ? | ✓ |
| Viafirma | ✓ | ? | ✓ |

Como podemos observar na Tabela 3 o requisito do uso do cartão de cidadão português falha em quase todas as *apps* existentes. *Apps* tais como a *ComsignTrust, SigningHub* e *Viafirma* parecem suportar *smartcards,* mas visto que são *apps* integradas dentro de pacotes *enterprise* não foi possível testar e retirar conclusões relativamente ao suporte oficial deste requisito. A *app* a desenvolver irá então atender aos requisitos acima referidos.

## 2.6. Ferramentas de desenvolvimento a utilizar

Após compreender a estrutura dos principais elementos constituintes da *app* a desenvolver surge então a necessidade de encontrar ferramentas que permitam realizar os passos idealizados de forma a obter uma assinatura digital numa *app* em Android usando o cartão de cidadão português.

### 2.6.1. SDK da Present Technologies

O SDK desenvolvido pela Present Technologies permite aceder aos certificados e chaves presentes no cartão de cidadão através dos APDU já referidos tanto em IOS como em Android. Este SDK veio a ser uma ferramenta vital neste projeto de estágio devido ao facto de não existir suporte para PKCS#11 em IOS e Android.

Funcionando como uma *blackbox*, recebe o *hash* do que se pretende assinar e internamente assina o mesmo usando a chave privada presente no cartão, devolvendo o *hash* encriptado para posterior inserção no PDF ou .docx.

### *2.6.2. BouncyCastle*

O *BouncyCastle* (em *desktop*) ou *Spongy Castle* (em Android) é uma API de criptografia em *Java* que é fundamental para vários aspetos deste projeto relacionados com criptografia. Como já foi referido anteriormente, existe a necessidade de fazer *hashing* de conteúdo a assinar para posterior encriptação usando a chave privada do cartão. Em assinaturas digitais PDF este processo é feito à parte em *desktop* ou no Android, criando um envelope PKCS#7 que constitui a assinatura digital para posterior inserção no PDF. O *BouncyCastle* permite criar este envelope, passando-o à *lib* de manipulação de PDF para esta o inserir no documento, originando o PDF assinado. Das *libs* escolhidas em seguida, tanto para manipulação de PDF como para .docx, ambas usam *BouncyCastle* internamente, logo faz todo o sentido a utilização da mesma para procedimentos externos às *libs*.

### 2.6.3. Manipulação de PDF

Visto que a componente de encriptação do *hash* é resolvida pela SDK da *Present Technologies*, surge então a necessidade de uma ferramenta que permita manipular conteúdo interno do documento PDF de forma a conseguir inserir o *hash* encriptado no seu interior. Visto que a programação em Android é feita em *Java*, os primeiros testes a estas ferramentas foram feitos em ambiente *desktop*, visto que em *desktop* estas ferramentas estão mais avançadas do que em Android. Caso estas ferramentas permitissem fazer uma assinatura digital, era então adaptado o código usado, mas para ambiente Android com essa mesma ferramenta, caso existisse.

#### *2.6.3.1. Desktop*

Em *desktop* foram testadas as ferramentas Apache PDFBox e iText. Ambas estas ferramentas permitem manipular o conteúdo interno de um documento PDF, permitindo assim inserir uma assinatura digital. A principal diferença é que o PDFBox não tem custos associados e a iText tem. No entanto ambas as ferramentas foram testadas no ambiente *desktop*, proporcionando também um *hands-on* relativamente à aplicação do conceito de uma assinatura digital considerada "externa", visto que é feita externamente ao dispositivo.

#### 2.6.3.2. Android

Para além das duas ferramentas referidas anteriormente, foram estudadas mais *libs* de assinaturas digitais em PDF, mas em ambiente Android, nomeadamente PDFBlackbox, plugPDF, qPDF Toolkit, PDFTron e *PDF Security SDK*. Os requisitos considerados mais importantes que a *lib* de manipulação de PDF deve ter são os custos, suporte a assinatura externa e uso nativo para Android. Na Tabela 4 a classificação de "❓" indica que não foi possível obter mais informações relativamente a essa *lib* nesse determinado requisito. O símbolo "✓"indica que o determinado requisito se verifica, enquanto o símbolo "❌" indica que o mesmo não se verifica.

Tabela 4- Análise comparativa entre *libs* a utilizar no projeto.

| Requisito / *PDF lib* | **Android nativo** | **Assinatura externa** | **Livre uso** |
|---|---|---|---|
| **PDFBlackbox** | ✗ | ✓ | ✗ |
| **plugPDF** | ✓ | ✗ | ✗ |
| **qPDF Toolkit** | ✓ | ✗ | ✗ |
| **PDFTron** | ✓ | ? | ✗ |
| **PDF Security SDK** | ✓ | ✗ | ✗ |
| **iTextG** | ✓ | ✓ | ✗ |
| **PDFBox port** | ✓ | ? | ✓ |

Após o estudo e reflexão sobre que *lib* a utilizar, foi acordado com a empresa o uso da *lib* IOSpara Android. A maior desvantagem desta *lib* de manipulação de PDF é o facto de que ainda se encontrar em desenvolvimento isto é, embora seja funcional em ambiente *desktop* existem determinadas classes relativas à implementação do requisito de assinatura externa que se encontram em desenvolvimento. Caberá ao estagiário envolver-se neste projeto *Open Source* [27] de forma a "agilizar" este processo de *port* e conseguir terminar as classes necessárias para que o requisito de assinatura externa seja verificado.

### 2.6.4. Manipulação de .docx

Tal como é necessário uma *lib* de manipulação de PDF, também será necessário uma *lib* de manipulação de .docx. Esta lib é encarregue de abrir um ficheiro no formato OXML e inserir a assinatura XML no seu interior, continuando no fim com um ficheiro .docx mas assinado. Aqui foi seguida a mesma analogia que foi utilizada na escolha da *lib PDF*, nomeadamente experimentar em ambiente *desktop,* verificando se atinge os requisitos necessários e só depois ver se a mesma *lib* existe em ambiente Android.

#### 2.6.4.1. Desktop

Relativamente a *libs* no ambiente *desktop* só foram encontradas e comparadas duas, nomeadamente a *Apache POI* e a *doc4j enterprise*. Ambas são escritas em *Java* e permitem a manipulação de documentos no formato OXML tal como o .docx. No entanto nenhuma delas suporta assinaturas externas.

#### 2.6.4.2. Android

Das duas *libs* descritas, somente a doc4j enterprise é que tem suporte em Android, mas trata-se de uma *lib* paga. Infelizmente não existem *libs* de manipulação de .docx que suportem os três requisitos necessários, nomeadamente suporte de assinatura externa, sem custos e em Android.

Caberá ao estagiário envolver-se também no projeto *Open Source* do *Apache POI* [28] primeiro em desktop de forma a conseguir fazer uma assinatura digital externa para futura implementação em ambiente Android. A principal razão que levou à colaboração neste projeto *Open Source* é que o *Apache POI* já inclui *libs* internas que permitem a criação e inserção de uma assinatura XML num .docx, nomeadamente o *BouncyCastle* e *Apache Santuario* ou *Java XML Digital Signature API*.

# Capítulo 3
# Análise de Requisitos

Neste capítulo serão expostos quais os métodos aplicados à análise de requisitos de forma a que seja claro o mais cedo possível as funcionalidades a suportar por esta aplicação. É aqui que devem ser recolhidos todos os requisitos possíveis de forma a responder à questão "O que vou desenvolver?" permitindo que quando a arquitetura vier a ser desenhada seja possível responder à questão de "Como o vou fazer?" da forma mais adequada possível, atendendo aos requisitos especificados.

Visto que a metodologia de desenvolvimento de software usada neste projeto pela empresa inicialmente era a de *waterfall* como descrito na secção 5.2.1 e também devido ao facto de não se tratar de um projeto critico, é aceitável o surgimento requisitos novos pela parte do cliente com o desenrolar do projeto, o que acabou por acontecer no segundo semestre de estágio. Com isto dito foram tidos em maior foco os requisitos inicialmente definidos na proposta de estágio, nomeadamente a inserção de assinaturas digitais em documentos PDF e Word no ambiente Android, usando o cartão de cidadão português.

Com o desenrolar do projeto existiram algumas modificações nos requisitos, nomeadamente requisitos existentes que foram removidos e outros requisitos novos surgiram de forma a preencher/compensar as lacunas deixadas pelos requisitos removidos. Os requisitos presentes na Figura 14 vieram a compensar os requisitos removidos, melhorando a solução de assinaturas em documentos PDF, visto que as assinaturas Word acabaram por ser removidas. A razão pela qual as assinaturas Word foram removidas da lista de requisitos da aplicação deve-se ao facto de não existir a *port* da *lib* Apache POI para Android e era praticamente impossível o estagiário a fazer em tempo útil.

Reduzindo o nível de abstração do sistema e entrando no diagrama de casos de uso específico de cada ator, é possível obter então uma visão menos abstrata de cada funcionalidade a implementar para o respetivo ator. Após saber que funcionalidades cada ator possui no sistema, é reduzido mais uma vez o nível de abstração, entrando em cada caso de uso do diagrama, especificando o que deve ser feito no mesmo.

Após os casos de uso é então desenhada a *User Interface* (UI) e especificada a *Hardware Interface* (HI) de forma a ter uma noção clara de como o utilizador irá interagir com a aplicação a desenvolver. Foram também analisados os requisitos legais em vigor para a aplicação a desenvolver. Por fim, foi então feita uma síntese dos requisitos recolhidos e qual a técnica de engenharia de requisitos que ajudou para que tal fosse possível.

## 3.1. Diagrama de casos de uso

Reduzindo agora o nível de abstração e entrando agora no diagrama de casos de uso referente à "Aplicação de assinaturas móvel", observando a Figura 14 é possível ter uma ideia mais profunda de quais as funcionalidades permitidas ao ator "Utilizador". Comparando a Figura 14 com o Anexo 18, é possível observar quais requisitos foram alterados e quais foram acrescentados.

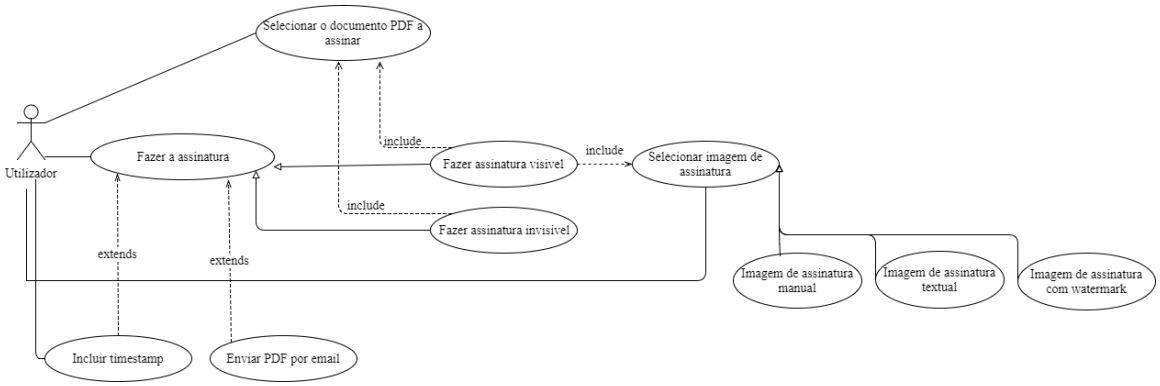



Figura 14- Diagrama de casos de uso do utilizador.

Nesta secção irão ser especificados cada um dos casos de uso do utilizador, visto que é o único ator e é à sua volta que a aplicação é construída. No entanto os casos de uso não são necessariamente um requisito da aplicação, mas sim uma metodologia de engenharia de requisitos muito usada com o principal foco de recolher requisitos funcionais e alguns requisitos de UI, como por exemplo mensagens de *feedback* ao utilizador quando não é possível completar o caso de uso com sucesso. Em seguida serão explicados de forma resumida cada um dos casos de uso referidos na Figura 14. Em cada explicação também estará presente a referência para a analise detalhada do caso de uso respetivo.

### 3.1.1. Resumo dos casos de uso 1 e 1.1- Selecionar documento PDF a assinar

Inicialmente permitindo selecionar documentos do tipo PDF e Word (Anexo 3), este foi um dos requisitos modificados, permitindo agora selecionar somente documentos PDF. Este documento PDF será então utilizado para inserir a assinatura digital. Estes casos de uso podem ser analisados com maior detalhe através dos Anexos 1 e 2.

### 3.1.2. Resumo dos casos de uso 2, 2.1 e 2.2 - Fazer assinatura visivel e invisivel

Estes casos de uso especificam que um utilizador deve conseguir fazer uma assinatura digital, podendo esta ser do tipo visível e invisível. Para fazer uma assinatura visível é necessário selecionar um documento para assinar e também uma imagem para a assinatura. Para fazer uma assinatura invisível, somente será necessário selecionar o documento a assinar. Estes casos de uso podem ser analisados com maior detalhe através dos Anexos 4, 5 e 6.

### 3.1.3. Resumo do caso de uso 3- Incluir *timestamp*

Este caso de uso especifica que o utilizador pode decidir incluir um selo temporal (*timestamp*) na sua assinatura digital. Este caso de uso poderá ser analisado com maior detalhe através do Anexo 7.

### 3.1.4. Resumo do caso de uso 4- Enviar PDF por email

Este caso de uso especifica que o utilizador poderá enviar por email o documento PDF após este ser assinado digitalmente. De qualquer forma, o documento assinado será sempre guardado na pasta de transferências. Este caso de uso poderá ser analisado com maior detalhe através do Anexo 8.

### 3.1.5. Resumo dos casos de uso 5,5.1,5.2,5.3- Selecionar imagem de assinatura

Este caso de uso especifica que um utilizador deve conseguir selecionar uma imagem para inserir numa assinatura digital visível. Esta imagem de assinatura visível pode ser composta por várias combinações das imagens de *watermark*, assinatura manual e textual. Este caso de uso pode ser analisado com maior detalhe através do Anexo 9,10,11,12.

## 3.2. User Interface

Na Figura 15 e 16 está representado o *mockup* da UI da aplicação. Graças aos casos de uso já é possível saber à priori onde e quando aparecerão as *popups* de *feedback* ao utilizador, bem como o texto contido nas mesmas e também nos botões da aplicação.


Figura 15- *mockup* da UI da funcionalidade de assinatura visível e invisível.

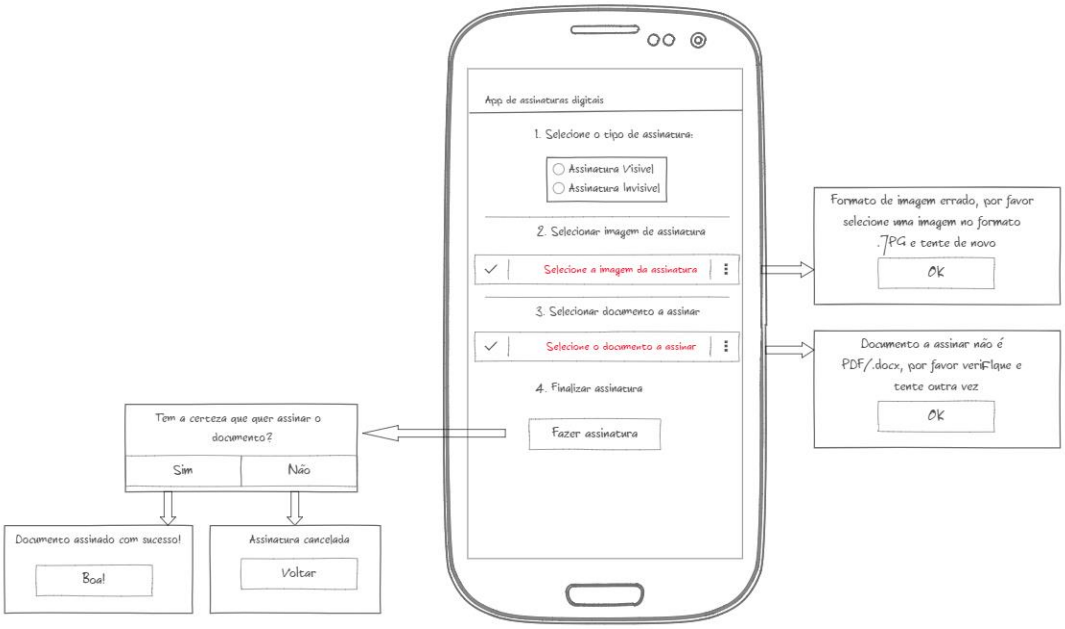



Figura 16- *mockup* das mensagens de erro e notificações.

## 3.3. Hardware Interface

Nesta subsecção são referidos os principais dispositivos físicos necessários para que esta aplicação a desenvolver funcione com sucesso.

Esta aplicação como um todo é composta por quatro componentes de *hardware*, nomeadamente um leitor de *smartcards* compatível com o cartão de cidadão, um *adapter* USB para *smartphones*, um cartão de cidadão e um *smartphone* Android.

### Adaptador OTG (On The Go)

Este adaptador USB tem como principal função de ligar a porta USB do leitor de cartões ao *smartphone*.

### Leitor de *smartcards*

O leitor de *smartcards* tem como principal e óbvia função de ler o conteúdo presente no cartão, neste caso o cartão de cidadão português. O seu aspeto comum pode ser observado pela Figura 17.


Figura 17- Exemplo de leitor de *smartcards* com cartão de cidadão inserido retirado de: http://www.cm-vianadoalentejo.pt/pt/intra/PublishingImages/CC1.png?RenditionID=16&Width=639&Height=362.

Os leitores deverão estar de acordo com as normas:

- PC/SC versão 1.0;
- ISO/IEC 7816-1,2,3,4: IC *Cards with Contacts* [29];
- EMV *Level 1* [30];
- CT-API versão 1.1 [31];
- CCID – *Chip Card Interface Device* 1.0 [32];
- Suportar a norma ISO/IEC 7186 *Class* A, B e C *(smartcards* com voltagens de 5V, 3V,
- 1.8V) [33];
- Suportar leitura e escrita para *smartcards* com microprocessadores alinhados com ISO/IEC 7816-1,2,3,4 protocolos T=0 e T=1;
- Suportar *smartcards* com frequências de relógio até 8Mhz;
- Os leitores deverão suportar *smartcards* de formato ID-1.

Este conjunto de normas aplicadas ao leitor e ao cartão estão fora do âmbito deste projeto. É de alguma importância referir que já foram feitos ataques a *smartcards*, logo é importante que tanto os leitores como os *smartcards* sigam todas as normas acima referidas de forma atualizada para evitar problemas de segurança. Mesmo assim o ideal é usar um leitor de cartões com *pinpad* de forma a que ao introduzir códigos de segurança não o seja feito no dispositivo Android mas sim diretamente no *pinpad* do leitor, evitando que o *pin* tenha que passar pelo *Operative System* (OS) do Android. É importante que o *pin* não passe pelo OS do Android porque pode existir *malware* do tipo *keystroke logger* que esteja à escuta da introdução de *pins* ou outra informação sensível. É aconselhada a implementação de um mecanismo de sessões que termine a ligação entre o leitor e o cartão para quando o utilizador já não estiver a usar o sistema e caso o *pin* seja descoberto sem que o mesmo se aperceba, seja impossível ao *hacker* usar a sessão para más intenções.

Para maior usabilidade o ideal será utilizar um leitor com *pinpad* específico para *smartphones* com porta *microUSB* macho, para remover a necessidade do cabo e do adaptador, permitindo ao utilizador de forma segura e prática de fazer assinaturas digitais usando o cartão de cidadão no seu *smartphone* Android até de pé.

#### Smartphone Android

O *smartphone* é uma peça importante neste sistema, visto que é aqui onde vai estar a correr a *app*.

#### *Smartcards* cartão de cidadão

Relativamente aos cartões de cidadão, os aspetos mais importantes já foram referidos no capítulo 2.

## 3.4. Regulamentação

Para além dos requisitos referidos no Estado da Arte não existem mais requisitos importantes relativamente à regulamentação em vigor. Os que existem são relativamente às entidades certificadores externas presentes na PKI do sistema a desenvolver, nomeadamente a *GeoTrust* e o Estado Português. Vale a pena referir que de acordo com o artigo 2º alínea g) do Decreto-Lei n.º 88/2009 de 9 de abril uma assinatura eletrónica qualificada é descrita como: *"assinatura digital ou outra modalidade de assinatura eletrónica avançada que satisfaça exigências de segurança idênticas às da assinatura digital baseadas num certificado qualificado e criadas através de um dispositivo seguro de criação de assinatura"*.

As características sublinhadas no Decreto-Lei vão de encontro ao que se pretende fazer neste estágio, nomeadamente a aplicação de uma assinatura digital que use certificados qualificados e que seja criada através de um dispositivo seguro como por exemplo o cartão de cidadão.

## 3.5. Síntese dos requisitos levantados

Na Tabela 5 encontra-se uma síntese dos requisitos levantados, bem como o método onde estes foram recolhidos.

Tabela 5- Síntese de requisitos recolhidos.

| **Tipo de requisito** | **Requisito** | **Método onde foi recolhido** |
|---|---|---|
| Funcional | Selecionar documento a assinar *PDF* | Casos de uso |
| Funcional | Fazer assinatura digital visível em *PDF* | Casos de uso |
| Funcional | Fazer assinatura digital invisível em *PDF* | Casos de uso |
| Funcional | Selecionar imagem de assinatura | Casos de uso |
| Funcional | Imagem de assinatura manual | Casos de uso |
| Funcional | Imagem de assinatura textual | Casos de uso |
| Funcional | Imagem de assinatura *watermark* | Casos de uso |
| Funcional | Incluir *timestamp* | Casos de uso |
| Funcional | Enviar PDF por email | Casos de uso |
| Não funcional | **Usabilidade** - Devem ser enviadas mensagens de *feedback* ao utilizador de forma a aumentar a usabilidade da aplicação | Especificação das *interfaces* de utilizador |
| Não funcional | **Segurança** - Deve ser usado um leitor de cartões com *pinpad* embutido | Especificação de *interface* de *hardware* |
| Não funcional | **Usabilidade** - O leitor deve estar diretamente ligado ao smartphone usando uma saída *microUSB* macho | Especificação de *interface* de *hardware* |

Após analisar a Tabela 5, é possível ter ideia das principais características do sistema a desenvolver, nomeadamente requisitos funcionais e não funcionais, tendo em conta também alguns detalhes físicos.

# Capítulo 4
# Planeamento

Neste capítulo será apresentado o planeamento do trabalho a realizar ao longo do ano, bem como análise de riscos associados ao projeto. Serão ainda apresentados os métodos e abordagens usados na realização do planeamento.

## 4.1. Primeiro semestre

Durante o primeiro semestre foi feito um pequeno planeamento de forma a ajudar o estagiário a organizar os seus processos de investigação e escrita das várias fases que foram presentes na entrega intermédia. Na lista  é possível observar quais foram as tarefas:

1. Conceitos intermédios Android
2. Planeamento do primeiro semestre
3. Estado da arte
   3.1. Estudo do tema
   3.2. Estudo de aplicações semelhantes
   3.3. Ferramentas a utilizar
4. Inicio de escrita de relatório intermédio
5. Análise de requisitos
   5.1. Requisitos funcionais
   5.2. Requisitos não-funcionais
   5.3. Restrições
6. Arquitetura
   6.1. Cenários e atributos de qualidade
   6.2. Criação de diagramas
   6.3. Avaliação da arquitetura e iterações
   *6.4. User interface*
7. Planeamento 2º semestre

A Figura 18 apresenta o diagrama de *Gantt* referente ao planeamento feito para o primeiro semestre efetuado no inicio do estágio.

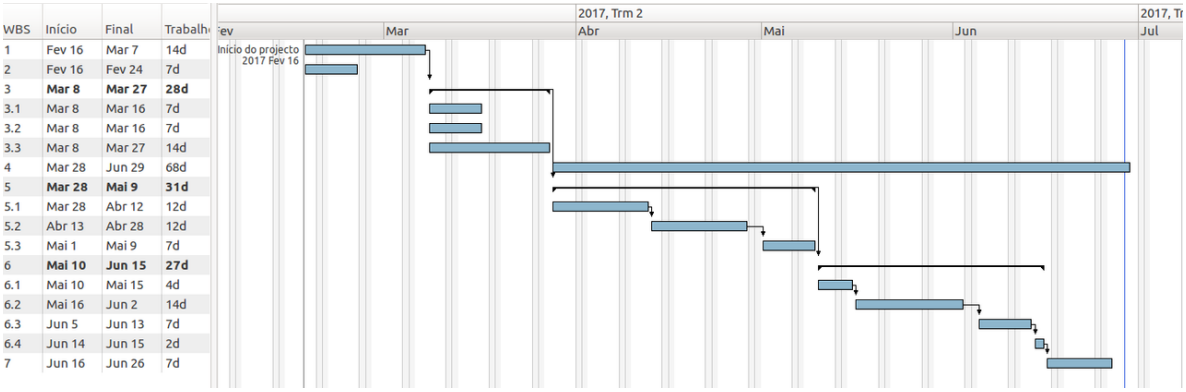



Figura 18- *Gantt* relativo ao primeiro semestre.

## 4.2. Segundo semestre

Nesta secção serão abordados os dois planeamentos feitos para o segundo semestre. O primeiro planeamento do segundo semestre foi feito no fim do primeiro semestre com os requisitos conhecidos a essa data. No entanto alguns dos requisitos foram alterados durante o segundo semestre o que veio a justificar a criação de um segundo planeamento para o segundo semestre. Inicialmente serão referidos quais foram os processos de planeamento usados nestes dois planeamentos e dentro de cada um serão referidos quais foram os seus riscos, os respetivos graus de exposição e quais os seus planos de mitigação.

### 4.2.1. Work Breakdown Structure

O *Work Breakdown Structure* [36] (WBS) é um processo usado em gestão de projetos que consiste na divisão de tarefas grandes em tarefas mais pequenas para que seja possível gerir um projeto com mais facilidade. No entanto existem regras a seguir de forma determinar as tarefas mais pequenas. Uma das regras usadas é a regra 8/80, esta implica que nenhuma tarefa possa demorar menos que 8 horas nem mais de 80 a ser concluída. A apresentação de uma WBS pode ser feita de várias formas, nomeadamente em árvore, em listas e tabelas.

### 4.2.2. Estimação de tarefas

De forma a conseguir estimar o esforço necessário para cada tarefa, foi utilizado o método de estimativa denominado por Estimativa de Três Pontos. Este método consiste em atribuir três valores de esforço para cada tarefa e com base neste calcular o valor esperado de esforço final da mesma.

Os valores de esforço podem ser classificados como:

- **Mais provável:** Caso tudo correr como esperado;
- **Melhor caso:** Caso tudo corra bem;
- **Pior caso:** Caso tudo corra mal .

A fórmula a aplicar após obtidos os três valores é a seguinte:

**( MelhorCaso + (4 x CasoMaisProvável ) + PiorCaso ) / 6**

### 4.2.3. Estimativa de baixo para cima

Para o desenvolvimento do planeamento foi seguida a abordagem Estimativa de Baixo para Cima ou *Bottom-up Estimation* [37] que consiste em criar o WBS, em seguida estimar cada tarefa usando a Estimativa de Três Pontos e no fim somar as estimativas para ser possível obter o esforço final a aplicar no projeto. Esta abordagem é constituída por seis passos, sendo estes:

1. Selecionar o estimador/equipa de estimativas, neste caso é o estagiário
2. Obter os requisitos, arquitetura de alto nível e as várias tecnologias a utilizar
3. Criar o WBS
4. Estimar cada tarefa com a Estimativa de Três Pontos
5. Calcular o esforço estimado de todo o projeto
6. Refinar estimativas se necessário

### 4.2.5. Primeiro planeamento do segundo semestre

As tarefas presentes na lista são referentes ao trabalho a realizar no segundo semestre de estágio e foram baseadas nos requisitos funcionais e restrições de negócio recolhidos anteriormente.

1. Fazer assinatura digital invisível em PDF usando a *lib PDFBox* em Android
   1.1. Familiarização com projetos *Open Source*
   1.2. Fazer o *port* de *desktop* para Android das classes necessárias para fazer assinaturas invisíveis
2. Fazer assinatura digital visível em PDF usando a *lib PDFBox* em Android
   2.1. Fazer o *port* de *desktop* para Android das classes necessárias para fazer assinaturas visíveis
3. Implementação de assinaturas digitais visíveis e invisíveis em documentos PDF e em ambiente Android
   3.1. Importação do SDK de leitura do cartão para o ambiente Android
   3.2. Implementar seleção de documentos PDF
   3.3. Implementar assinatura digital visível em documentos PDF
   3.4. Implementar seleção de imagem de assinatura
   3.5. Implementar assinatura digital invisível em documentos PDF
4. Fazer assinatura digital visível e invisível em documentos Word usando o *Apache POI* em Android
   4.1. Fazer primeiro em ambiente *desktop*
      4.1.1. Criar uma assinatura XML
      4.1.2. Inserir a assinatura XML no documento Word
      4.1.3. Inserir imagem da assinatura no documento Word
   4.2. Implementação da solução 4.1 em ambiente Android
5. Implementação de assinaturas digitais visíveis e invisíveis em documentos Word e em ambiente Android
   5.1. Implementar seleção de documentos Word
   5.2. Implementar assinatura digital visível em documentos Word
   5.3. Implementar assinatura digital invisível em documentos Word
6. Validação dos requisitos não-funcionais
   6.1. Validação da usabilidade
   6.2. Validação da segurança
7. Escrita do relatório final

Em seguida foi feita a estimação de cada tarefa usando o método de Estimação de Três Pontos presente na Tabela 6.

Tabela 6- Estimação de cada tarefa usando o método de Estimação de Três Pontos .

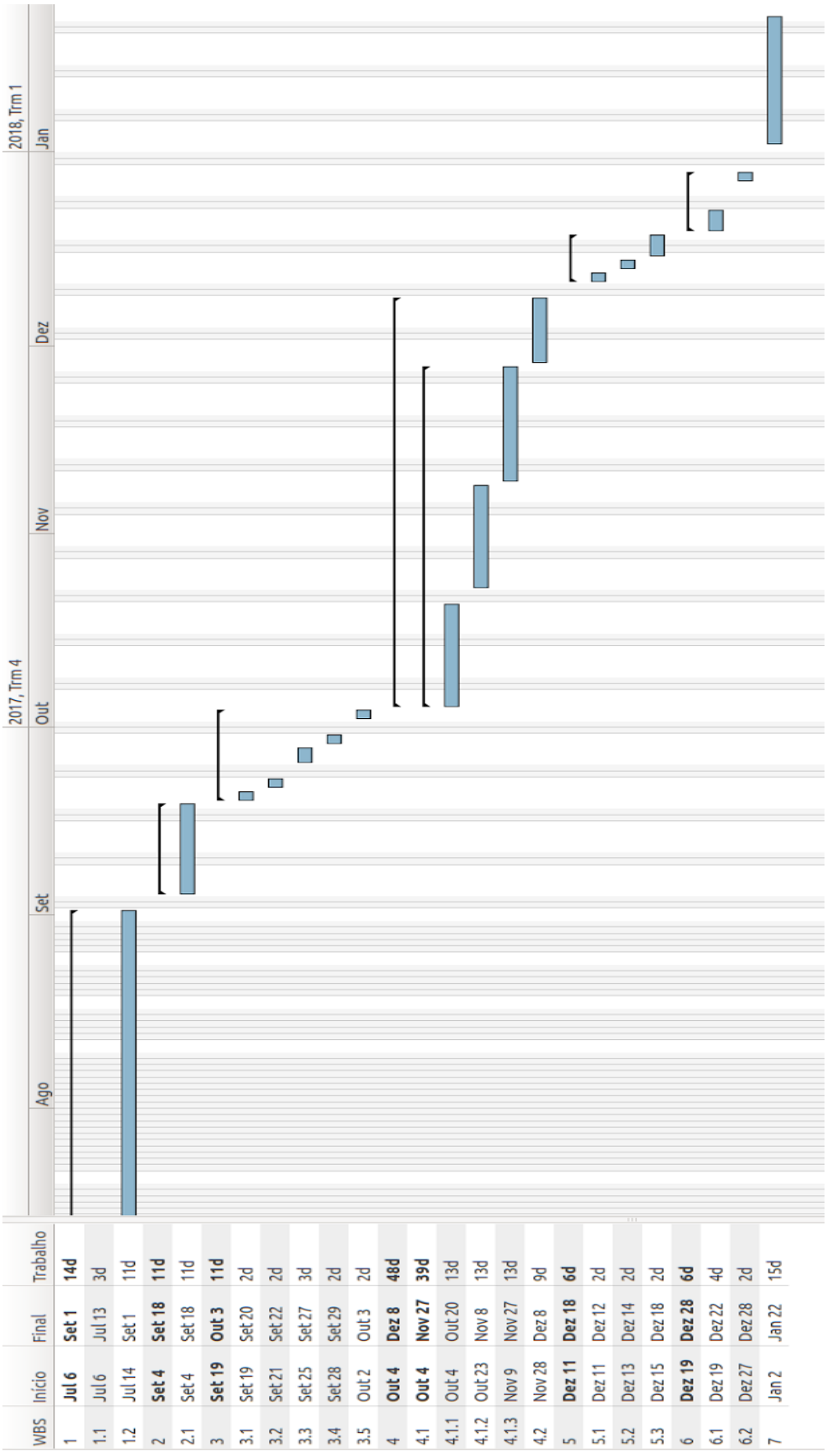
| Tarefas | Melhor Caso | Caso Mais Provável | Pior Caso | Caso Expectado |
|---|---|---|---|---|
| 1.1 | 1 | 3 | 7 | 3,33 |
| 1.2 | 3 | 10 | 21 | 10,67 |
| 2.1 | 3 | 10 | 21 | 10,67 |
| 3.1 | 1 | 2 | 3 | 2 |
| 3.2 | 1 | 2 | 3 | 2 |
| 3.3 | 1 | 3 | 7 | 3,33 |
| 3.4 | 1 | 2 | 3 | 2 |
| 3.5 | 1 | 2 | 3 | 2 |
| 4.1.1 | 3 | 14 | 21 | 13,33 |
| 4.1.2 | 3 | 14 | 21 | 13,33 |
| 4.1.3 | 3 | 14 | 21 | 13,33 |
| 4.2 | 3 | 7 | 21 | 8,67 |
| 5.1 | 1 | 2 | 3 | 2 |
| 5.2 | 1 | 2 | 3 | 2 |
| 5.3 | 1 | 2 | 3 | 2 |
| 6.1 | 2 | 3 | 7 | 3,5 |
| 6.2 | 1 | 2 | 3 | 2 |
| 7 | 10 | 15 | 20 | 15 |
| **SOMA (dias)** | **40** | **109** | **191** | **111,17** |

Na Tabela 7 é apresentado o planeamento do segundo semestre feito pelo estagiário face a cada tarefa, consoante a estimativa de dias obtida na Tabela 6.

Tabela 7- Planeamento do segundo semestre.

| Tarefa | Data de início | Data de fim | Duração |
|---|---|---|---|
| 1.1 | 6-7-2017 | 13-7-2017 | 3 |
| 1.2 | 14-7-2017 | 1-9-2017 | 11 |
| 2.1 | 4-9-2017 | 18-9-2017 | 11 |
| 3.1 | 19-9-2017 | 20-9-2017 | 2 |
| 3.2 | 21-9-2017 | 22-9-2017 | 2 |
| 3.3 | 25-9-2017 | 27-9-2017 | 3 |
| 3.4 | 28-9-2017 | 29-9-2017 | 2 |
| 3.5 | 2-10-2017 | 3-10-2017 | 2 |
| 4.1.1 | 4-10-2017 | 20-10-2017 | 13 |
| 4.1.2 | 23-10-2017 | 8-11-2017 | 13 |
| 4.1.3 | 9-11-2017 | 27-11-2017 | 13 |
| 4.2 | 28-11-2017 | 8-12-2017 | 9 |
| 5.1 | 11-12-2017 | 12-12-2017 | 2 |
| 5.2 | 13-12-2017 | 14-12-2017 | 2 |
| 5.3 | 15-12-2017 | 18-12-2017 | 2 |
| 6.1 | 19-12-2017 | 22-12-2017 | 4 |
| 6.2 | 27-12-2017 | 28-12-2017 | 2 |
| 7 | 2-1-2018 | 22-1-2018 | 15 |

Na Figura 19 é possível observar o *Gantt Chart* respetivo à Tabela 7.




| WBS | Início | Final | Trabalho |
|---|---|---|---|
| 1 | **Jul 6** | **Set 1** | **14d** |
| 1.1 | Jul 6 | Jul 13 | 3d |
| 1.2 | Jul 14 | Set 1 | 11d |
| 2 | **Set 4** | **Set 18** | **11d** |
| 2.1 | Set 4 | Set 18 | 11d |
| 3 | **Set 19** | **Out 3** | **11d** |
| 3.1 | Set 19 | Set 20 | 2d |
| 3.2 | Set 21 | Set 22 | 2d |
| 3.3 | Set 25 | Set 27 | 3d |
| 3.4 | Set 28 | Set 29 | 2d |
| 3.5 | Out 2 | Out 3 | 2d |
| 4 | **Out 4** | **Dez 8** | **48d** |
| 4.1 | **Out 4** | **Nov 27** | **39d** |
| 4.1.1 | Out 4 | Out 20 | 13d |
| 4.1.2 | Out 23 | Nov 8 | 13d |
| 4.1.3 | Nov 9 | Nov 27 | 13d |
| 4.2 | Nov 28 | Dez 8 | 9d |
| 5 | **Dez 11** | **Dez 18** | **6d** |
| 5.1 | Dez 11 | Dez 12 | 2d |
| 5.2 | Dez 13 | Dez 14 | 2d |
| 5.3 | Dez 15 | Dez 18 | 2d |
| 6 | **Dez 19** | **Dez 28** | **6d** |
| 6.1 | Dez 19 | Dez 22 | 4d |
| 6.2 | Dez 27 | Dez 28 | 2d |
| 7 | Jan 2 | Jan 22 | 15d |

Figura 19- *Gantt Chart* para o segundo semestre.

#### 4.2.5.1. Análise de riscos

Como em todo o tipo de projetos, é normal que existam riscos associados ao mesmo, por isso é fundamental fazer uma análise dos riscos existentes e ter sempre um plano de mitigação para que caso os riscos aconteçam, venham a causar o mínimo de impacto negativo possível no produto final.

#### 4.2.5.2. Identificação e classificação dos riscos

Os riscos identificados para este projeto são:

1. Tempo de desenvolvimento associado às *libs Open Source* seja elevado
2. Elevada curva de aprendizagem associada à implementação das *libs*
3. Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas
4. Alteração dos requisitos
5. Alteração da arquitetura

Após serem identificados os principais riscos, surge a necessidade de classificar os mesmos, atribuindo um impacto, probabilidade e janela temporal.

O **impacto** indica o efeito no sucesso do projeto caso o risco venha a ocorrer. Na Tabela 8 é possível observar quais os valores possíveis para o impacto e quais as suas consequências no projeto caso venha o risco venha a acontecer.

Tabela 8- Impacto e consequências.

| **Impacto** | **Consequências no projeto** |
|---|---|
| Alto | O sucesso do projeto fica comprometido |
| Médio | O sucesso do projeto é possível, mas com custos associados mais elevados |
| Baixo | O sucesso é atingido sem grandes dificuldades |

A **probabilidade** é a probabilidade que o risco tem de provocar o impacto definido. Na Tabela 9 é possível observar quais as classificações que podem atribuídas e quais as percentagens associadas.

Tabela 9- Probabilidade de risco.

| **Classificação da probabilidade** | **Percentagem associada** |
|---|---|
| Alta | >70% |
| Média | >40% e <70% (inclusive) |
| Baixa | <40% |

A **janela temporal** é o tempo que vai desde que a identificação do risco até ser necessário lidar com o mesmo. Na Tabela 10 está representado a classificação atribuída à janela temporal e os seus respetivos espaços temporais.

Tabela 10- Janela temporal.

| **Classificação da Janela Temporal** | **Espaço Temporal** |
|---|---|
| Longa | Maior que três meses |
| Média | Entre um e três meses (inclusive) |
| Curta | Menos que um mês |

Nas tabelas 11,12,13,14 e 15 serão representadas as classificações atribuídas a cada um dos riscos

Tabela 11- Tempo de desenvolvimento associado às *libs Open Source* maior que o previsto.

| | |
|---|---|
| **Identificador** | A |
| **Risco** | Tempo de desenvolvimento associado às *libs Open Source* maior que o previsto |
| **Impacto** | Médio |
| **Probabilidade** | Alta |
| **Janela Temporal** | Longo |
| **Consequências** | Atraso ou incumprimento do planeamento |
| **Mitigação** | Aplicação da arquitetura alternativa presente na 47 e 48 |

Tabela 12- Elevada curva de aprendizagem associada à implementação das *libs*.

| | |
|---|---|
| **Identificador** | B |
| **Risco** | Elevada curva de aprendizagem associada à implementação das *libs* |
| **Impacto** | Médio |
| **Probabilidade** | Baixa |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento |
| **Mitigação** | Ajustes no planeamento de forma a obter tempo necessário para a aprendizagem das *libs* |

Tabela 13- Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas.

| | |
|---|---|
| **Identificador** | C |
| **Risco** | Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas |
| **Impacto** | Médio |
| **Probabilidade** | Médio |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos. |

Tabela 14- Alteração dos requisitos.

| | |
|---|---|
| **Identificador** | D |
| **Risco** | Alteração dos requisitos |
| **Impacto** | Alto |
| **Probabilidade** | Média |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento e alterações na arquitetura |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos inicialmente definidos |

Tabela 15- Alteração à arquitetura.

| | |
|---|---|
| **Identificador** | E |
| **Risco** | Alteração à arquitetura |
| **Impacto** | Alto |
| **Probabilidade** | Baixo |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento e alterações no código escrito |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos definidos. |

### 4.2.5.3. Matriz de exposição aos riscos

De forma a expor os riscos, aumentar a visibilidade e permitir uma melhor tomada de decisões foi criada uma matriz (impacto x probabilidade) presenta na Figura 20, acompanhada pela sua respetiva legenda.

| | | **Impacto** | | |
|---|---|---|---|---|
| | | Baixo | Médio | Alto |
| **Probabilidade** | Alta | E | A | |
| | Média | | C | D |
| | Baixa | B | | |

**Legenda:**

| **Cor** | **Exposição** |
|---|---|
| | Baixa |
| | Média |
| | Elevada |
| | Crítica |

Figura 20- Matriz impacto VS probabilidade.

Após a análise da matriz de exposição é possível ordenar os riscos consoante o seu nível de exposição, obtendo então a Tabela 16

Tabela 16- Matriz de exposição de riscos.

| **Identificador** | **Exposição** |
|---|---|
| A | Elevada |
| D | Elevada |
| E | Média |
| C | Média |
| B | Baixa |

### 4.2.6. Segundo planeamento do segundo semestre

Este novo planeamento do segundo semestre deve-se ao facto de que os riscos identificados como A e D na Tabela 16 acabaram eventualmente por acontecer devido ao seu nível de exposição. Como estes riscos foram identificados no inicio do segundo semestre, já estava definido um plano de mitigação dos mesmos de forma a reduzir o impacto negativo ao máximo. Observando as Tabelas dos riscos A e D, 11 e 14 respetivamente e analisando os seus planos de mitigação podemos ver que os mesmos implicam fazer um novo planeamento, dando prioridade aos requisitos definidos inicialmente e só depois aos novos. O plano de mitigação do risco A acabou por não ser aplicado devido à prioridade existente de implementar as assinaturas em PDF em vez de assinaturas em Word. Devido ao facto de que o risco D aconteceu, consequentemente o risco E também aconteceu. Uma das principais causas do risco A ter acontecido foi porque não existia nenhuma *port* de uma das *libs*, nomeadamente do Apache POI, sendo esta uma das principais causas do abandono das assinaturas em Word, juntamente com a baixa prioridade deste requisito na solução final.

As tarefas presentes na lista são referentes ao trabalho a realizar a partir de meio do segundo semestre de estágio e foram baseadas nos novos requisitos funcionais e restrições de negocio que foram surgindo ao longo do segundo semestre. Encontram-se a negrito os requisitos que surgiram a meio do segundo semestre de estágio de forma a acrescentar valor ao projeto, compensando os requisitos que foram removidos, nomeadamente as assinaturas a documentos Word. Encontram-se sublinhados os requisitos que foram removidos de forma a tornar mais claro as alterações que foram feitas às funcionalidades inicialmente definidas.

**1.** Fazer assinatura digital invisível em PDF usando a *lib PDFBox* em Android
   1.1. Familiarização com projetos *Open Source*
   1.2. Fazer o *port* de *desktop* para Android das classes necessárias para fazer assinaturas invisíveis
**2.** Fazer assinatura digital visível em PDF usando a *lib PDFBox* em Android
   2.1. Fazer o *port* de *desktop* para Android das classes necessárias para fazer assinaturas visíveis
**3.** Implementação de assinaturas digitais visíveis e invisíveis em documentos PDF e em ambiente Android
   3.1. Importação do SDK de leitura do cartão para o ambiente Android
   3.2. Implementar seleção de documentos PDF
   3.3. Implementar assinatura digital invisível em documentos PDF
   3.4. Implementar assinatura digital visível em documentos PDF
   3.5. Implementar seleção de imagem de assinatura
      3.5.1. Implementar o modo *preview* da imagem de assinatura
         **3.5.1.1. Imagem de assinatura com *watermark***
         **3.5.1.2. Imagem de assinatura com assinatura manuscrita**
         **3.5.1.3. Imagem de assinatura textual**
**4. Incluir *timestamp* na assinatura**
**5. Enviar PDF por email depois de assinado**

**6.** Fazer assinatura digital visível e invisível em documentos Word usando o *Apache POI* em Android
  6.1. Fazer primeiro em ambiente *desktop*
    6.1.1. Criar uma assinatura XML
    6.1.2. Inserir a assinatura XML no documento Word
    6.1.3. Inserir imagem da assinatura no documento Word
  6.2. Implementação da solução 4.1 em ambiente Android
**7.** Implementação de assinaturas digitais visíveis e invisíveis em documentos Word e em ambiente Android
  7.1. Implementar seleção de documentos Word
  7.2. Implementar assinatura digital visível em documentos Word
  7.3. Implementar assinatura digital invisível em documentos Word
**6.** Validação dos requisitos
  6.1. Funcionais
  6.2. Não funcionais
**7.** Escrita do relatório final

Em seguida foi feita a estimação de cada tarefa usando o método de Estimação de Três Pontos presente na Tabela 17. As estimativas dos requisitos que não foram alterados mantiveram-se, enquanto os requisitos removidos saíram da tabela de estimação e os requisitos novos foram inseridos.

Tabela 17- Nova estimação de cada tarefa usando o método de Estimação de Três Pontos.

| Tarefas | Melhor Caso | Caso Mais Provável | Pior Caso | Caso Expectado |
|---|---|---|---|---|
| 1.1 | 1 | 3 | 7 | 3,33 |
| 1.2 | 3 | 10 | 21 | 10,67 |
| 2.1 | 3 | 10 | 21 | 10,67 |
| 3.1 | 4 | 7 | 10 | 7,00 |
| 3.2 | 1 | 4 | 7 | 4,00 |
| 3.3 | 2 | 5 | 10 | 5,33 |
| 3.4 | 2 | 5 | 10 | 5,33 |
| 3.5 | 1 | 2 | 3 | 2,00 |
| 3.5.1.1 | 3 | 7 | 14 | 7,50 |
| 3.5.1.2 | 3 | 7 | 14 | 7,50 |
| 3.5.1.3 | 3 | 7 | 14 | 7,50 |
| 4 | 2 | 5 | 14 | 6,00 |
| 5 | 2 | 3 | 7 | 3,50 |
| 6.1 | 2 | 3 | 7 | 3,50 |
| 6.2 | 1 | 2 | 3 | 2,00 |
| 7 | 10 | 15 | 20 | 15,00 |
| **SOMA (dias)** | **32** | **78** | **159** | **83,83** |

Na Tabela 18 é apresentado o novo planeamento do segundo semestre feito pelo estagiário face a cada tarefa, consoante a estimativa de dias obtida na Tabela 17.

Tabela 18- Novo planeamento do segundo semestre.

| | Nome | Duração | Ínicio | Fim |
|---|---|---|---|---|
| 1 | ⊟**1** | **41.88dias** | **06/07/2017** | **01/09/2017** |
| 2 | 1.1 | 3dias | 06/07/2017 | 13/07/2017 |
| 3 | 1.2 | 11dias | 14/07/2017 | 01/09/2017 |
| 4 | ⊟**2** | **11dias** | **04/09/2017** | **18/09/2017** |
| 5 | 2.1 | 11dias | 04/09/2017 | 18/09/2017 |
| 6 | ⊟**3** | **34dias** | **19/09/2017** | **03/11/2017** |
| 7 | 3.1 | 2dias | 19/09/2017 | 20/09/2017 |
| 8 | 3.2 | 2dias | 21/09/2017 | 22/09/2017 |
| 9 | 3.3 | 3dias | 25/09/2017 | 27/09/2017 |
| 10 | 3.4 | 3dias | 28/09/2017 | 02/10/2017 |
| 11 | ⊟**3.5** | **24dias** | **03/10/2017** | **03/11/2017** |
| 12 | ⊟**3.5.1** | **24dias** | **03/10/2017** | **03/11/2017** |
| 13 | 3.5.1.1 | 8dias | 03/10/2017 | 12/10/2017 |
| 14 | 3.5.1.2 | 8dias | 13/10/2017 | 24/10/2017 |
| 15 | 3.5.1.3 | 8dias | 25/10/2017 | 03/11/2017 |
| 16 | 4 | 6dias | 06/11/2017 | 13/11/2017 |
| 17 | 5 | 4dias | 14/11/2017 | 17/11/2017 |
| 18 | ⊟**6** | **6dias** | **20/11/2017** | **27/11/2017** |
| 19 | 6.1 | 4dias | 20/11/2017 | 23/11/2017 |
| 20 | 6.2 | 2dias | 24/11/2017 | 27/11/2017 |
| 21 | 7 | 15dias | 28/11/2017 | 18/12/2017 |

É de alguma importância referir que a data final da tarefa 7 não coincide com a data de entrega final da tese, isto permitirá ao estagiário usar o tempo restante para aperfeiçoamentos ou novos requisitos que poderão aparecer. Na Figura 21 é possível observar o *Gantt Chart* respetivo à Tabela 18.

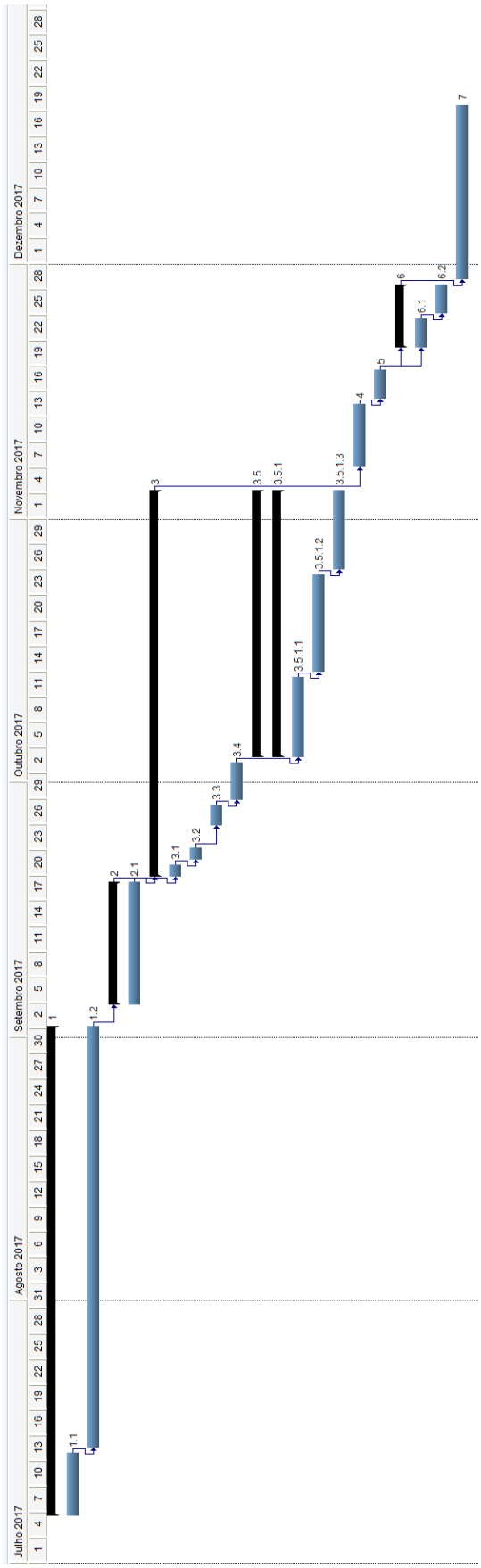



Figura 21- Novo *Gantt Chart* para o segundo semestre.

#### 4.2.6.1. Análise de riscos

Novamente será necessário fazer uma nova análise de riscos para o novo planeamento. No entanto os riscos não serão os mesmos devido ao facto de que nesta altura já era possível fazer uma assinatura digital em documentos PDF tanto visível como invisível, logo este risco detetado no primeiro planeamento do segundo semestre já não se iria aplicar.

#### 4.2.6.2. Identificação e classificação dos riscos

Os riscos identificados para este projeto são:

1. Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas
2. Alteração dos requisitos
3. Alteração da arquitetura

Os riscos serão classificados baseado nos mesmos critérios já definidos na secção 4.2.5.2 tabelas 8,9 e 10.

Nas tabelas 19,20,21 serão representadas as classificações atribuídas a cada um dos riscos

Tabela 19 Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas.

| | |
|---|---|
| **Identificador** | A |
| **Risco** | Estimativas realizadas foram otimistas do tempo necessário à realização das tarefas |
| **Impacto** | Médio |
| **Probabilidade** | Médio |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos. |

Tabela 20- Alteração dos requisitos.

| | |
|---|---|
| **.Identificador** | B |
| **Risco** | Alteração dos requisitos |
| **Impacto** | Alto |
| **Probabilidade** | Média |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento e alterações na arquitetura |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos inicialmente definidos |

Tabela 21- Alteração à arquitetura

| | |
|---|---|
| **Identificador** | C |
| **Risco** | Alteração à arquitetura |
| **Impacto** | Alto |
| **Probabilidade** | Baixo |
| **Janela Temporal** | Média |
| **Consequências** | Atraso ou incumprimento do planeamento e alterações no código escrito |
| **Mitigação** | Ajustes no planeamento, priorizando os requisitos definidos. |

### 4.2.6.3. Matriz de exposição aos riscos

De forma a expor os riscos, aumentar a visibilidade e permitir uma melhor tomada de decisões foi criada uma matriz (impacto x probabilidade) para o novo planeamento presente na Figura 22, acompanhada pela sua respetiva legenda.

| | | **Impacto** | | |
|---|---|---|---|---|
| | | Baixo | Médio | Alto |
| **Probabilidade** | Alta | C | | |
| | Média | | A | B |
| | Baixa | | | |

**Legenda:**

| **Cor** | **Exposição** |
|---|---|
| | Baixa |
| | Média |
| | Elevada |
| | Crítica |

Figura 22- Matriz impacto VS probabilidade do novo planeamento.

Após a análise da matriz de exposição é possível ordenar os riscos consoante o seu nível de exposição, obtendo então a Tabela 22.

Tabela 22- Matriz de exposição de riscos do novo planeamento.

| **Identificador** | **Exposição** |
|---|---|
| B | Elevada |
| C | Média |
| A | Média |

# Capítulo 5
# Arquitetura e Trabalho Desenvolvido

## 5.1. Arquitetura

Tendo em conta que já foram levantados os requisitos da aplicação a desenvolver, é possível responder à questão com um maior grau de certeza de "O que vou fazer?". O principal objetivo deste capítulo será responder à questão de "Como o vou fazer?" atendendo aos requisitos levantados no capítulo 3. É aqui que irão ser desenhadas, analisadas e avaliadas as arquiteturas possíveis num processo iterativo, tendo em conta aos requisitos recolhidos e aos que podem eventualmente aparecer com o desenrolar do desenvolvimento no segundo semestre. No entanto a versão de arquitetura feita neste relatório final tem somente em conta os requisitos conhecidos atualmente, tal como foi especificado no capítulo Análise de Requisitos. Possíveis alterações nos requisitos iniciais implicarão ajustes na arquitetura, como se vai poder constatar mais adiante.

Inicialmente serão abordados os *drivers* arquiteturais da aplicação, mais focados nos seus atributos de qualidade e restrições, visto que os requisitos funcionais terão sido abordados no capítulo anterior. Em seguida foram desenhados três tipos de vistas arquiteturais, nomeadamente a vista componente e conetor, a vista de módulos e a vista de alocação. Em seguida foi feita uma avaliação e validação da arquitetura feita. Por último foram referidas arquiteturas alternativas à arquitetura escolhida.

A Figura 13 já referida permite obter o contexto da aplicação a desenvolver, permitindo perceber que a aplicação consiste fundamentalmente em enviar conteúdo *hashed* para o cartão, onde este devolve este conteúdo encriptado com a chave privada para o dispositivo Android para posterior utilização.

### 5.1.2. Atributos de qualidade

Os atributos de qualidade podem também ser referidos como requisitos não funcionais. Com base em alguns aspetos importantes recolhidos no capítulo de Análise de Requisitos foi possível traçar alguns requisitos não funcionais, nomeadamente a usabilidade e a segurança.

A **usabilidade** pode ser entendida como "*o termo técnico usado para descrever a qualidade de uso de uma interface*" [35]. Podemos considerar dois tipos de utilizador final da aplicação, nomeadamente o utilizador que quer implementar assinaturas digitais na sua empresa e o utilizador que já faz assinaturas digitais, mas em computador. Ambos os utilizadores estão à procura de algo cómodo e prático para fazer assinaturas digitais, logo fará todo o sentido que esta solução a desenvolver tenhas essas mesmas características, sendo fundamental focar na usabilidade. É importante referir de que o uso correto de *popups* fornecendo *feedback* ao utilizador a usar a aplicação irá transmitir uma sensação de segurança ao mesmo, sendo mais uma razão para focar neste requisito sendo uma boa introdução para o próximo.

O outro requisito não funcional é a **segurança**, visto que a aplicação a desenvolver irá usar e trabalhar sobre dados sensíveis externos à aplicação em si, mas internos ao sistema, nomeadamente a utilização da chave privada do utilizador.

Qualquer utilizador terá receio do uso que esta aplicação irá fazer com os dados do seu cartão, logo ter sido considerada a segurança como o segundo requisito não funcional necessário a garantir na aplicação a desenvolver.

**Restrições de negócio**

Uma das restrições de negócio impostas neste estágio é o orçamento. Embora fosse possível pagar por *libs* externas, o estagiário deverá usar as *libs* que não tenham custos associados. Foi também considerada como restrição de negócio a prioridade de desenvolvimento da assinatura PDF sobre a Word, onde mais tarde o requisito das assinaturas Word acabaria por ser removido, dando mais foco às assinaturas PDF.

Outra restrição existente é que somente os dados necessários à aplicação da assinatura poderão ser usados, outros dados como morada, nome NIF etc. não deverão ser usados, estando estes também protegidos por leis de dados pessoais. Estas leis não foram abordadas no capítulo anterior porque esta aplicação foi pensada para não aceder nem armazenar/utilizar de qualquer modo aos dados para além dos fundamentais ao funcionamento da aplicação.

### 5.1.3. Vistas arquiteturais da aplicação

Devido ao facto de ter existido alteração nos requisitos definidos inicialmente, por consequencia foi necessário fazer pequenos ajustes nas vistas já desenhadas no relatorio intermédio. As vistas antigas podem ser consultadas nos Anexos 24 e 25.

Nesta subsecção irão ser expostas as vistas consideradas importantes da aplicação a desenvolver, vistas estas baseadas nos requisitos mais recentes. As vistas representam o *core* de um documento de arquitetura sendo utilizadas para comunicar as decisões de design aos vários *stakeholders*. O nível de detalhe de cada vista deve ser adequado ao *stakeholder* ao qual esta se destina e deve ser suficiente para perceber o modo como a arquitetura satisfaz os drivers arquiteturais. Desta forma, foram criadas três vistas destinadas a diferentes *stakeholders*:

**Vista componente e conetor**: Apresenta o modo como os vários componentes do sistema se relacionam e quais os protocolos que utilizam. Por apresentar um nível de detalhe elevado esta vista destina-se também aos arquitetos e *developers*.

**Vista de módulos**: Apresenta a decomposição da arquitetura em módulos com algum detalhe. Destina-se sobretudo aos arquitetos e *developers*.

**Vista de alocação**: Apresenta o mapeamento entre os componentes do sistema e o seu cenário de implementação. É uma vista de detalhe específica ao *deployment* da aplicação. Por esse motivo, esta destina-se também aos arquitetos e *developers*.

explicam-se resumidamente alguns detalhes necessários à compreensão de cada uma das vistas.

### Vista componente conetor

Na Figura 23 é possível observar a vista componente e conetor. Esta vista tem como principal objetivo dar e entender quais os principais componentes do sistema a desenvolver e como estes se interligam.

Analisando a Figura 23 é possível perceber que existem três componentes principais, nomeadamente a Android *Activity*, o componente de assinatura e o componente de acesso ao cartão de cidadão. O componente Android *Activity* funciona como um elo de ligação entre o componente de assinatura e o componente de acesso ao cartão, enviando/recebendo valores através de uma *Interface Java*. A componente assinatura está encarregada de inserir o valor da assinatura dentro do respetivo documento, enquanto o componente de acesso ao cartão está encarregado da utilização do conteúdo do cartão para gerar a o valor da assinatura.

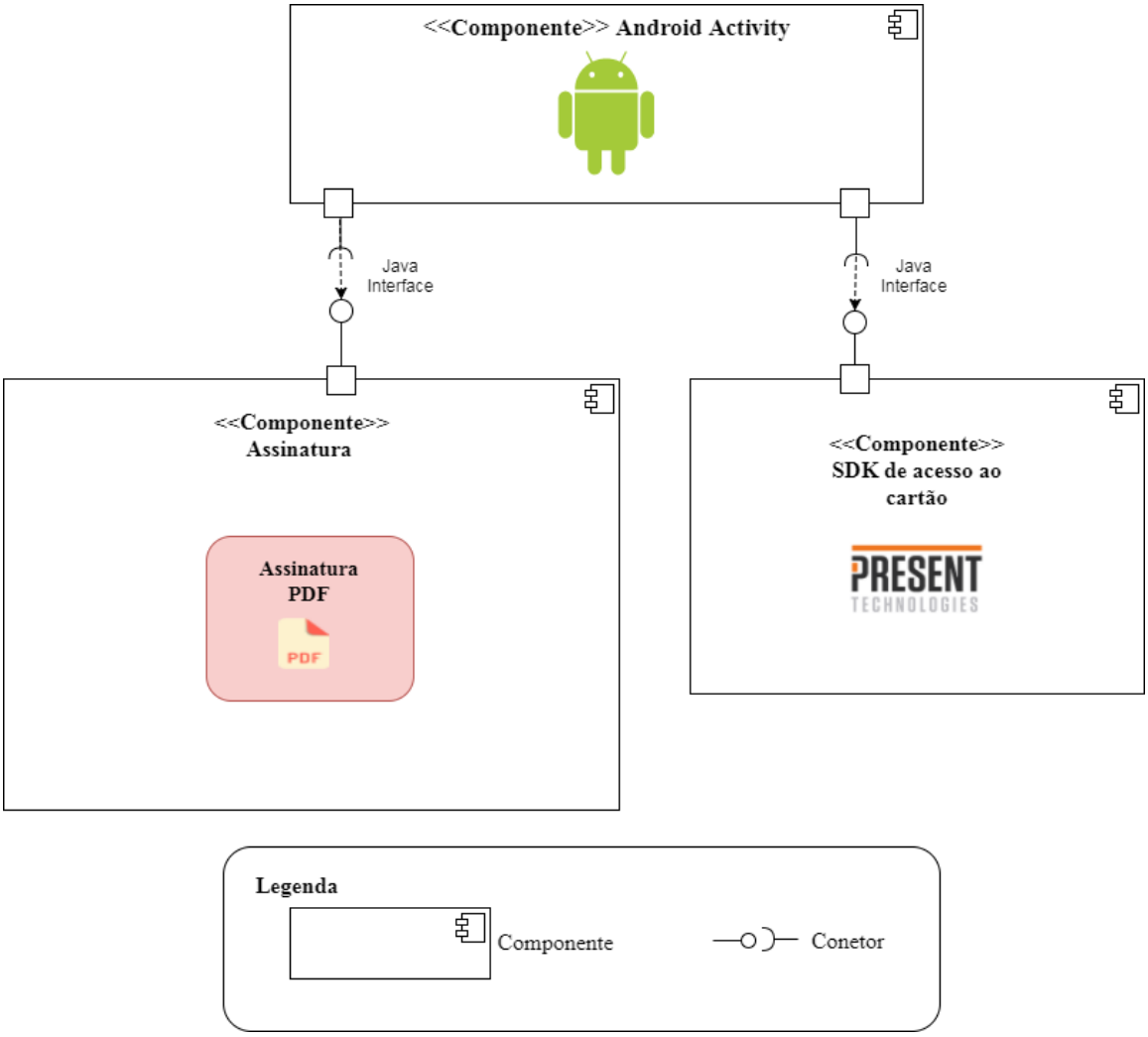



Figura 23- Vista de componente e conetor.

### Vista de módulos

Na Figura 24 é possível observar a vista de módulos. Esta vista tem como principal objetivo dar a entender quais os módulos existentes no sistema e quais as tecnologias que estes usam, baixando o nível de abstração à medida que se analisa cada tecnologia. Ao contrário da vista anterior, aqui é possível analisar as tecnologias usadas dentro do sistema consoante vários níveis de abstração.

No SDK de cartão de cidadão não foi possível analisar quais as tecnologias que usa visto que é um módulo que foi tratado como uma *blackbox* desde o início do estágio e que é da autoria da empresa. À semelhança do diagrama anterior é possível observar que a aplicação Android depende tanto da componente das assinaturas como do SDK de acesso ao cartão. A componente das assinaturas possui agora somente uma vertente, nomeadamente a de assinaturas em PDF.

Na vertente das assinaturas PDF são utilizadas duas *libs* fundamentais, nomeadamente a *Apache* IOSe o *SpongyCastle*. O IOStem como principal função de colocar o valor da assinatura dentro do documento PDF, enquanto o *SpongyCastle* tem como principal função de gerar o PKCS#7/CMS envelope com o valor da assinatura para posteriormente ser inserido dentro do documento.

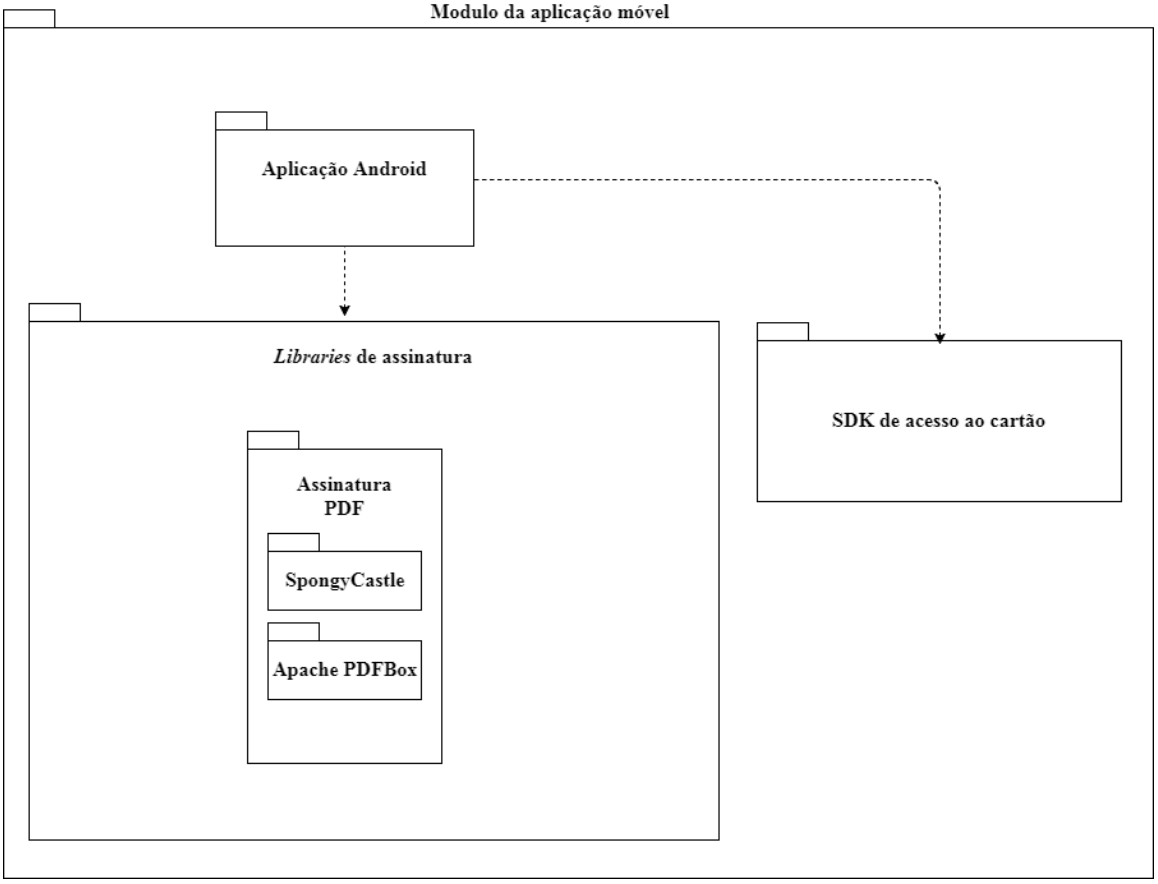



Figura 24- Vista de módulos.

### Vista de alocação

Na Figura 25 é possível observar a vista de alocação do sistema a desenvolver. Esta vista tem como principal foco dar a entender quais são os componentes do sistema e como se ligam, mas de um ponto de vista físico. É possível observar que é constituído por três componentes físicos principais, nomeadamente o *smartphone* Android, o leitor de cartões com *pinpad* e o cartão de cidadão. Tal como já foi referido anteriormente, o dispositivo Android comunica por USB com o cartão e este comunica por APDU com o cartão de cidadão, retornando ao *smartphone* a informação pedida para posterior utilização do valor da assinatura.

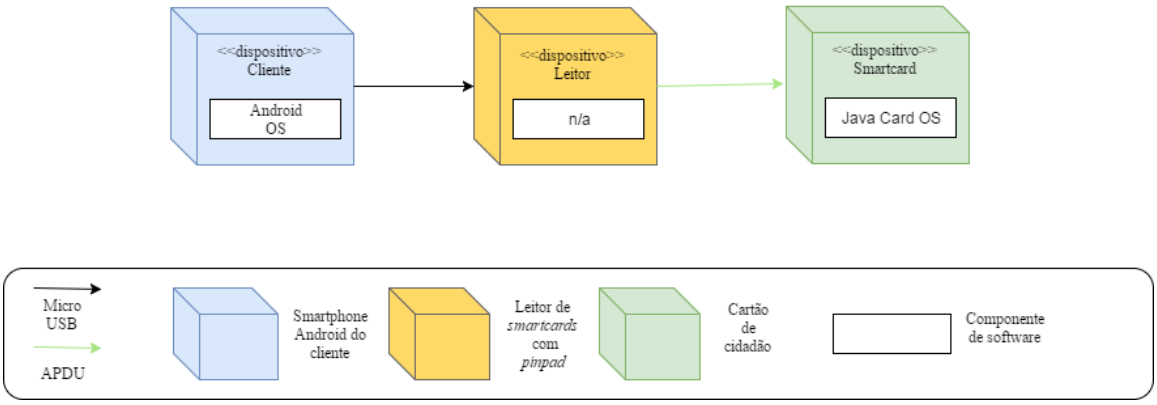



Figura 25- Vista de alocação.

### 5.1.4. Análise e avaliação da arquitetura

Nesta secção são referidas as tecnologias e pacotes de software utilizados e a sua respetiva justificação. Uma das principais restrições de negócio definidas foi o custo, logo faz todo o sentido que as *libs* de assinatura escolhidas sejam *Open Source*, consequentemente reduzindo o custo do sistema a desenvolver. É de notar que as restrições técnicas também foram tidas em conta, sendo as *libs* usadas feitas em *Java*. Relativamente aos atributos de qualidade de usabilidade e segurança estes também foram tidos em conta na elaboração desta arquitetura. A usabilidade é obtida principalmente pela estrutura física do sistema, nomeadamente com a junção do *smartphone* Android com o leitor e o cartão num formato compacto, cômodo e prático de usar como foi possível observar na vista de alocação. A segurança é obtida através da aplicação de assinaturas digitais qualificadas, provenientes do uso seguro do cartão de cidadão e das *libs* escolhidas. É importante referir que a segurança também é obtida com o uso do leitor de cartões com *pinpad* embutido, bem como um mecanismo de controlo de sessões com o objetivo de tornar extremamente difícil o acesso remoto de forma contínua ao cartão por agentes externos. Isto é relevante porque controla um tipo de ataques a *smartcards* conhecidos por *Bleichenbacher attacks*[38], que necessitam de acesso contínuo ao conteúdo do cartão para gerar informações que comprometam a segurança proporcionada pelo mecanismo de chaves. O facto de que todo o sistema está inserido num só módulo, nomeadamente a aplicação móvel, permite que não seja necessária internet, proporcionando não só um ambiente mais seguro como também permitindo ao utilizador assinar documentos em locais sem acesso à internet.

No entanto nem todos os drivers arquiteturais são atendidos com esta arquitetura, nomeadamente algumas das funcionalidades não são suportadas pelas *libs* visto que o conceito de assinaturas digitais móveis usando *smartcards* seja recente. Caberá ao estagiário envolver-se nos projetos *Open Source* das respetivas *libs* e desenvolver as funcionalidades que faltam para que a arquitetura projetada vá em conta a todos os drivers arquiteturais definidos.

### 5.1.5. Alternativas arquiteturais

Devido ao facto de que as *libs* usadas na arquitetura proposta necessitarem de algum trabalho de desenvolvimento por parte do estagiário no segundo semestre foi proposta uma arquitetura alternativa.

Esta arquitetura visa colmatar as falhas presentes na arquitetura anterior, reduzindo os riscos associados à mesma caso não seja possível ao estagiário desenvolver as componentes que faltam nas *libs*. Esta arquitetura surgiu também para satisfazer novas restrição de negócio e técnicas, nomeadamente o tempo de desenvolvimento, assinatura só em PDF e na plataforma IOS. Estas restrições surgiram com o aparecimento de clientes interessados no conceito da aplicação e era necessário fazer de forma rápida um protótipo para mostrar. A restrição da plataforma iOS é satisfeita nesta arquitetura, visto que a componente IOS não será feita pelo estagiário, mas sim por colaboradores qualificados internos à empresa.

A arquitetura alternativa é uma arquitetura cliente servidor e baseia-se no mesmo conceito explorado pelo estagiário, bem como no uso das mesmas *libs*, à exceção de que já não são *libs* em Android, mas sim em ambiente *desktop* situadas na parte do servidor. Visto que estas *libs* são ambientes *desktop*, estas não possuem as mesmas limitações que as mesmas em ambiente Android, logo todas as condições estão reunidas para que seja possível fazer uma assinatura digital móvel em ambiente IOS para mostrar ao cliente. Esta arquitetura tem como cliente o dispositivo IOS que usa somente a *lib* SpongyCastle para construir o envelope PKCS#7/CMS com o valor da assinatura e o SDK de acesso ao cartão para obter o conteúdo a introduzir neste envelope.

O servidor tem como principal função o uso da *lib* PDFBox para inserir a assinatura no documento, assinatura que é enviada pelo cliente ao servidor. Não existem *libs* semelhantes ao PDFBox em iOS, logo existe uma grande vantagem de que sua implementação seja feita na parte do servidor., permitindo assim fazer a assinatura digital em iOS.

No entanto foi pedido ao estagiário para continuar com a arquitetura inicialmente proposta devido à vantagem que esta trazia. A vantagem da arquitetura inicialmente proposta é que o documento PDF nunca sairia do dispositivo, ao contrario da arquitetura alternativa cliente-servidor onde o PDF é enviado para um servidor para assinar, este enviando de volta para o dispositivo o PDF assinado.

## 5.2. Trabalho Desenvolvido

### 5.2.1. Metodo de desenvolvimento

O médoto de desenvolvimento de software usado pela empresa na fase inicial deste projeto era o *waterfall*, método este que por consequencia foi usado pelo estagiário. Este método *waterfall* consiste em definir cada etapa da forma mais completa possivel, e so quando cada etapa estiver concluida até ao fim é que se passa para a proxima, não podendo depois voltar às etapas anteriormente definidas. Este método obriga então a que tudo esteja bem definido inicialmente.Na Figura 26 é possivel ter uma noção mais clara de como funciona esta metodologia de desenvolvimento *waterfall*,

Após apresentação da primeira versão da aplicação a um potencial cliente, chegou-se à conclusão de que seria necessário ajustar aquilo que eram os requisitos iniciais e consequentemente a metodologia de desenvolvimento. Optou-se nessa fase por uma metodologia ágil que conseguisse responder ao dinamismo da interacção com este cliente, esta metodologia é denominada por *Kanban*. Este método tem como principal vantagem de ser mais flexivel a nivel de requisitos, não necessitando de ter todos os requisitos definidos inicialmente, podendo estes vir a surgir gradualmente à medida que os restantes requisitos estão a ser trabalhados. Na Figura 27 é possivel ter uma melhor visão de como este método funciona.

Ter um cliente mais proximo no desenvolvimento do *software* é importantissimo para todo este processo devido ao facto de que se surgirem alterações nos requisitos, mais rápido se saberá e mais rápido se poderão tomar ações corretivas. Neste tipo de projetos não criticos é normal que inicialmente os requisitos não estejam todos completamente definidos, logo faz todo o sentido a implementação deste metodo de desenvolvimento neste projeto. O contacto com o cliente foi feito através de um superior da empresa, e os *feedbacks* do cliente passavam deste superior para o estagiário. Este método veio a provar-se util porque os requisitos vieram a ser alterados e foram tomadas as devidas ações corretivas de forma a reduzir o impacto negativo no projeto.

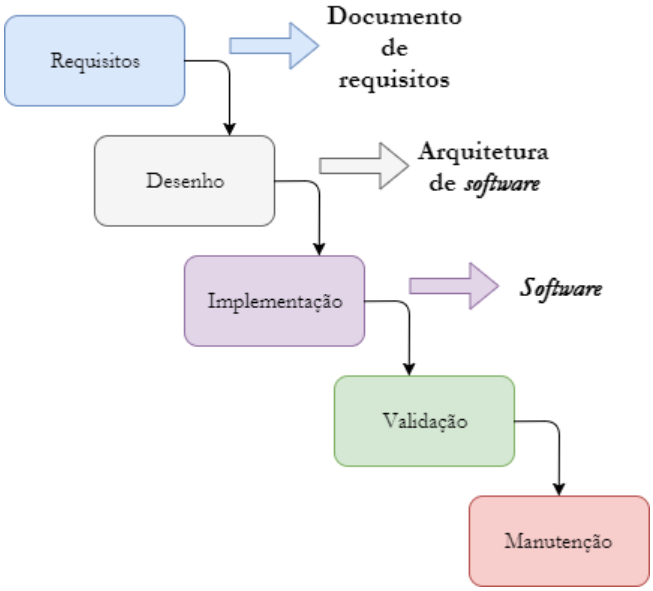



Figura 26- Método de desenvolvimento *waterfall.*

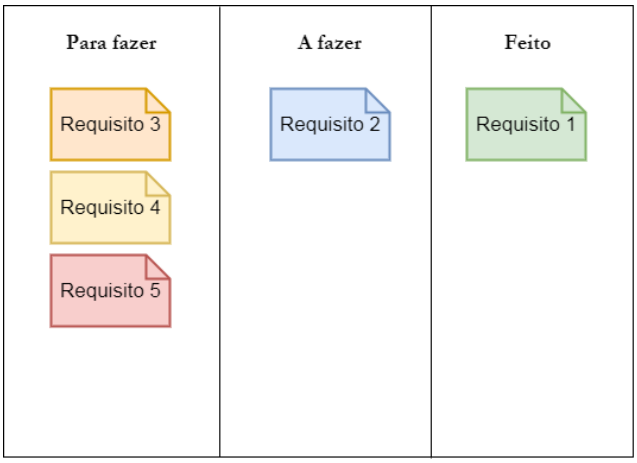



Figura 27- Método de desenvolvimento *Kanban.*

**Entregas efetuadas**

Durante o processo de desenvolvimento existiram duas entregas principais, onde a na primeira entrega foi usado o método *waterfall* e na segunda o *Kanban*. A primeira entrega consistia na prova do conceito de assinatura digital em dispositivos Android para documentos Word e PDF.

Na segunda entrega já estava a ser utilizado o método *Kanban*, logo foi consistia em várias sub entregas. Os requisitos que foram removidos sairam da secção "Para fazer"do quadro *KanBan*, acabando por dar espaço aos novos. Na secção 4.2.6 é possivel analisar com maior detalhe os requisitos novos e quais os foram removidos.

### 5.2.2. Desenvolvimento de solução *desktop*

Conforme planeado, primeiro foi feita uma implementação de assinatura digital em documentos PDF na plataforma *desktop* usando a *lib* PDFBox. A principal vantagem desta abordagem é saber exatamente quais as classes que estão em falta no PDFBox-Android quando tentar correr o mesmo código desta *lib* em Android.

O estudo do processo de assinatura em PDF efetuado no capitulo do Estado da Arte permitiu ao estagiário localizar mais facilmente no PDFBox quais as classes mais importantes para efetuar uma assinatura digital e qual o seu papel na mesma. Em seguida serão mostradas quais as principais classes do PDFBox usadas e qual o seu papel no processo de assinatura em PDF.

**Classes relevantes do PDFBox**

***PDDocument***: Esta classe representa o documentoPDF permite fazer operações sobre o mesmo, tais como abrir e guardar vários tipos de informação, sendo uma delas por exemplo a assinatura digital.

***PDSignature***: Esta classe está responsavel por representar a assinatura a localizar ou a inserir num documento PDF. É nesta classe que são definidos os vários elementos do *Signature Dictionary*,como por exemplo o *Filter,SubFilter,ByteRange* etc (presentes na Figura 6).

***ExternalSigningSupport***: Esta *interface* está responsavel por obter o valor do *ByteRange* em formato *Byte Array* para que futuramente seja feito o *hashing* ao mesmo.

***PDVisibleSignDesigner***: Esta classe está responsavel por definir a localização da assinatura visivel, nomeadamente qual a imagem a inserir,qual o campo de assinatura e em que página este se situa.

### Importância do Bouncy/Spongy Castle

Esta *lib* identificada na arquitetura foi a *lib* responsavel por gerar o pacote *PKCS#7* que contem a assinatura digital a colocar dentro do documento. Embora esta também permitisse o *hashing*, foram utilizadas *libs* nativas do Java, nomeadamente a *lib Security* de forma a reduzir dependências de *libs* externas.

### Utilização do SDK de acesso ao cartão

Um dos passos mais importantes da assinatura digital é feito através de classes privadas da *lib* de acesso ao cartão desenvolvido pela empresa. Esta *lib* permitiu a encriptação do *message digest*, *digest,* este que posteriormente seria inserido dentro do pacote *PKCS#7* a inserir no documento.

Possuindo agora todas as ferramentas para realizar o processo de assinatura digital num PDF e analisando a respetiva lista presente na secção 2.2.3 é agora possível associar cada papel das *libs* utilizadas a cada fase do processo de assinatura. Para demonstrar o processo de assinatura foi construído um diagrama de atividades , relacionando cada ferramenta com o seu papel no processo de assinatura. É importante referir que só foram referidas mais detalhadamente as classes referentes ao PDFBox, as restantes classes das outras ferramentas não serão referidas, visto estas ferramentas foram tratadas como uma caixa preta.

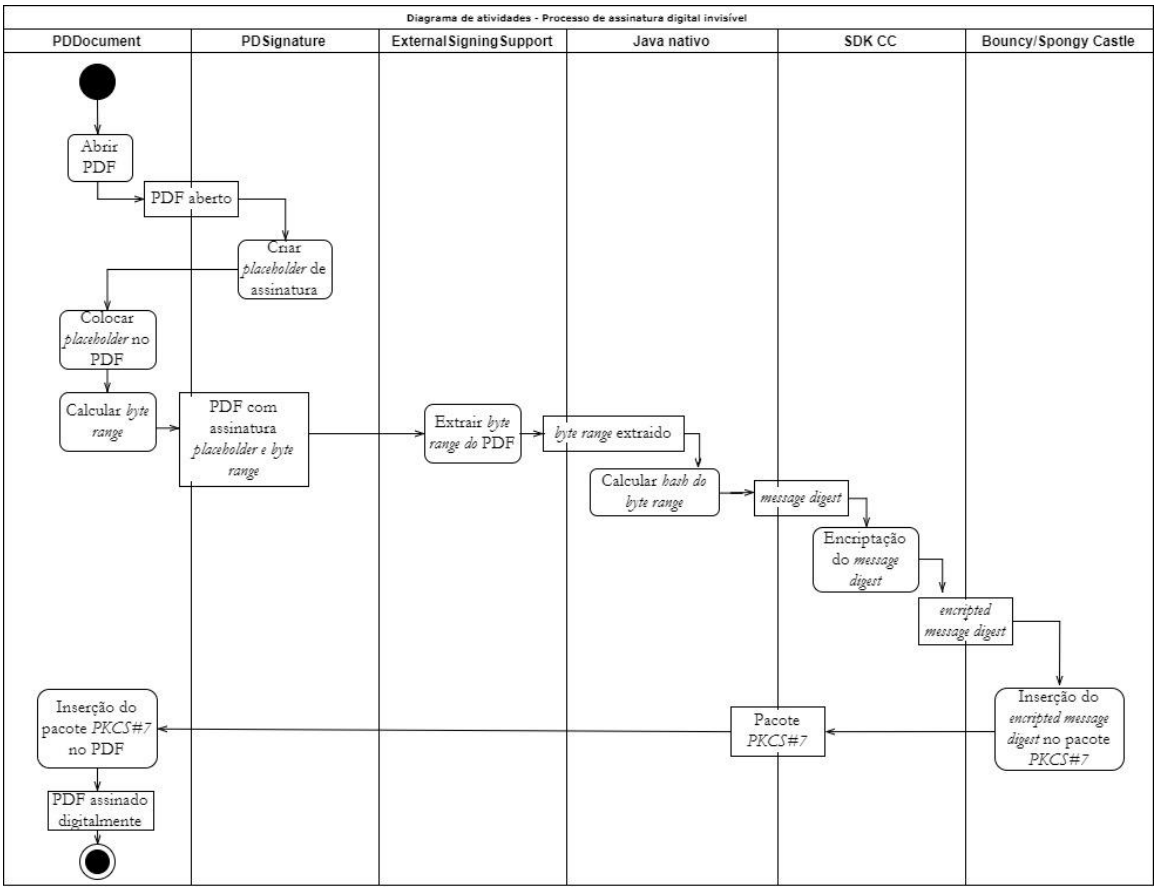



Figura 28- Diagrama de atividades do processo de assinatura digital invisível e ferramentas utilizadas.

Sabendo o processo de assinatura e quais as ferramentas a usar, foi possível então fazer uma assinatura digital invisível num documento PDF como se pode observar na Figura 28.

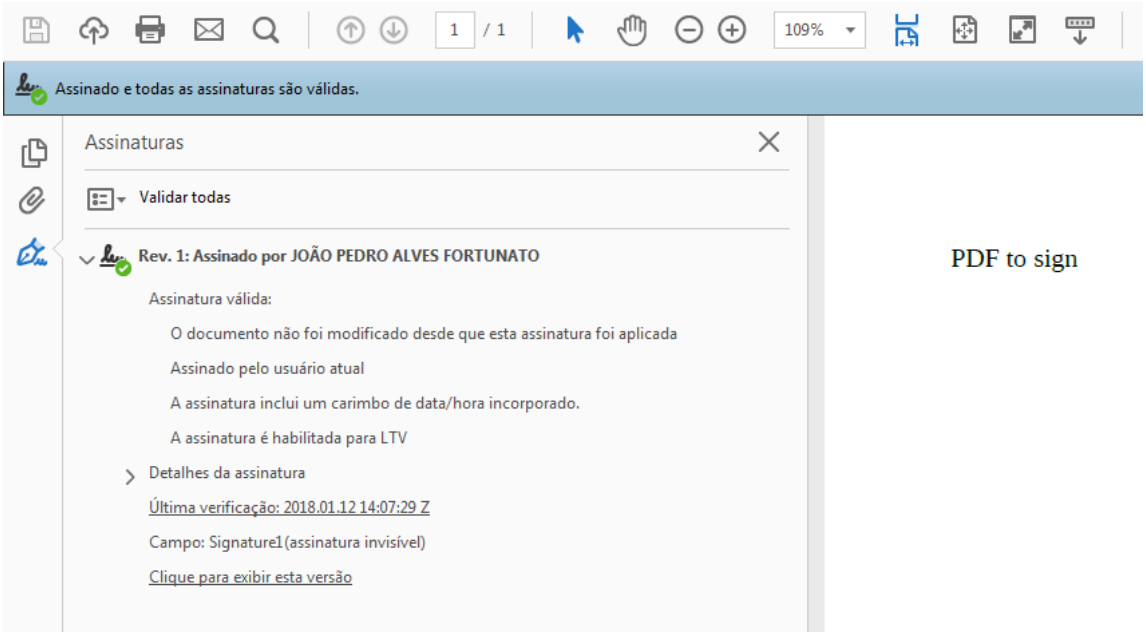



Figura 29- Aspeto de uma assinatura digital válida no Adobe Reader.

### 5.2.3. Desenvolvimento de solução Android

#### Assinatura visível e invisível em PDF antes da contribuição

Tendo a assinatura digital funcional em *desktop* , o próximo passo seria tentar fazer o mesmo, mas agora em ambiente Android. No entanto, o PDFBox-Android como está na versão 1.8.9 e a funcionalidade de assinatura digital só foi implementada a partir da versão 2.0 no PDFBox-Desktop, ainda não era possível fazer assinaturas digitais usando uma chave privada externa. O problema estava na inexistência de dois métodos, um deles na classe *PDDocument* e outro na *interface ExternalSigningSupport*. Identificados os problemas o estagiário teria agora que se envolver e contribuir para o projeto *Open Source* PDFBox-Android.

#### Contribuição para projeto Open Source PDFBox-Android

Uma das dificuldades existentes na contribuição esteve relacionada com o uso do sistema de controlo de versões distribuído *Git* utilizado, devido ao facto de ser uma novidade para o estagiário. Foi também uma grande oportunidade para aprender a usa-lo, até porque o sistema também é utilizado frequentemente em grande parte dos projetos de desenvolvimento de *software* e principalmente na Present Technologies. Para isso foi necessário primeiro fazer uma "copia" do PDFBox-Android chamado de *clone/fork*, isto permitiu ao estagiário ter uma copia local do PDFBox para efetuar as alterações pretendidas.

Inicialmente foi necessário reportar o problema abrindo um *issue*, notificando o responsável do PDFBox-Android chamado de TomRoush. Este *issue* pode ser encontrado em [39].

Os erros ou falhas de implementação foram encontrados através de comparação do código do PDFBox-Desktop e PDFBox-Android, visto que o que se pretendia já estava desenvolvido em *desktop*. Após comparadas as duas *libs* "linha alinha" o estagiário detetou quais as peças que faltavam para que as assinaturas externas em PDFBox-Android passassem a funcionar sem problemas. Após aplicadas as correções localmente, chegou a altura de juntar as mesmas ao PDFBox-Android do Tom através de um chamado de *merge request*. Estas alterações podem ser consultadas em [40] na aba "files changed", sendo que as mesma só poderiam ser aplicadas caso todos os casos de teste desenvolvidos pelo Tom passassem .

No entanto nem todas as alterações foram feitas pelo estagiário, existem alterações que provêm de algum trabalho que já tinha sido feito pelo Tom. As alterações feitas pelo estagiário juntamente com a *patch* enviada pelo Tom possibilitaram a criação funcional das assinaturas digitais externas em Android, isto é, usando a chave privada presente no cartão de cidadão.

No entanto estas alterações embora aceites pelo Tom, só serão publicadas quando a versão do PDFBox-Android chegar à versão 2, de forma a coincidir com a versão em *desktop*.

**Assinatura visível e invisível em PDF depois da contribuição**

Tendo agora as assinaturas invisíveis externas a funcionar ainda existiam alguns problemas com as assinaturas visíveis, sendo estes os seguintes:

1. Dado um campo de assinatura, as coordenadas do mesmo não eram calculadas e quando era inserida a imagem de assinatura, esta era colocada no canto superior esquerdo por defeito;
2. A imagem de assinatura a inserir no campo de assinatura não era redimensionada automaticamente de forma a caber no mesmo;
3. Devido à falta da componente de redimensionamento, a imagem tinha que ser redimensionada pelo estagiário, potencialmente comprometendo a sua qualidade.

Detetados estes problemas, coube ao estagiário os resolver. As soluções desenvolvidas pelo estagiário estão presentes na secção 5.2.3.2. Estes problemas foram eventualmente reportados e o respetivo *issue* pode ser consultado em [41].

#### 5.2.3.1 User Interface

Inicialmente foi feito um esboço de como seria a UI e quais as suas mensagens de *feedback* na secção de 3.2 da Análise de Requisitos. No entanto o aspeto visual e as mensagens de *feedback* acabaram por ser ligeiramente alteradas. A interface gráfica da aplicação pode ser observada na Figura 30 .

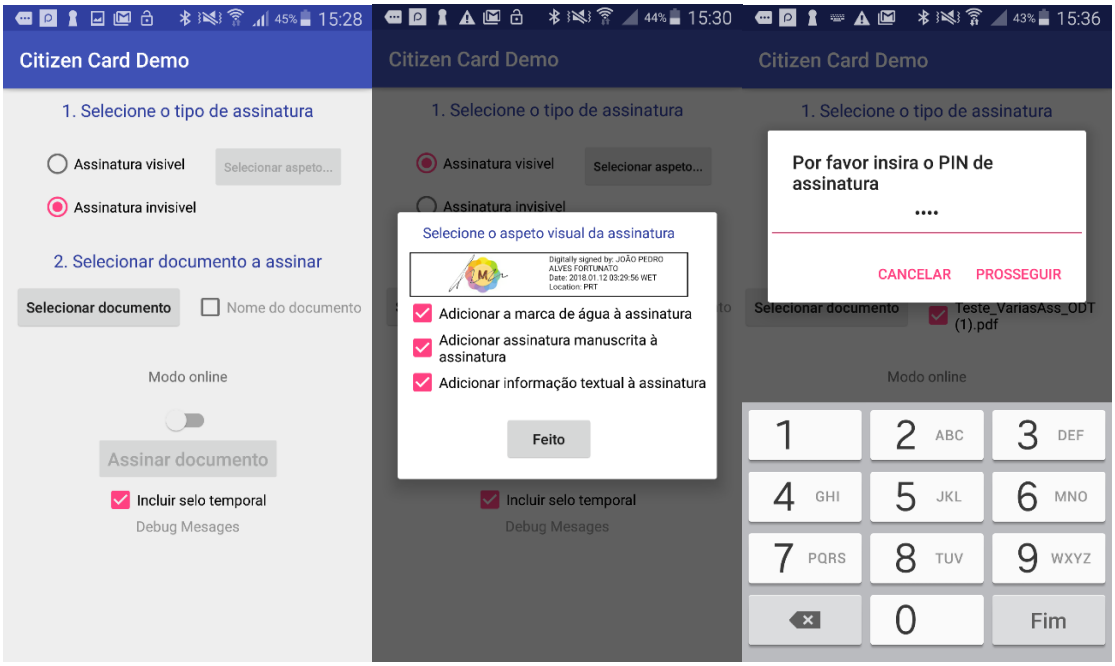



Figura 30- Interface Gráfico da aplicação.

### Mensagens de *feedback*

Durante a execução da aplicação são mostradas várias mensagens de *feedback* ao utilizador de forma a guiar o mesmo pelo *workflow* da assinatura. A Figura 31 ilustra o género de mensagens que poderão aparecer.

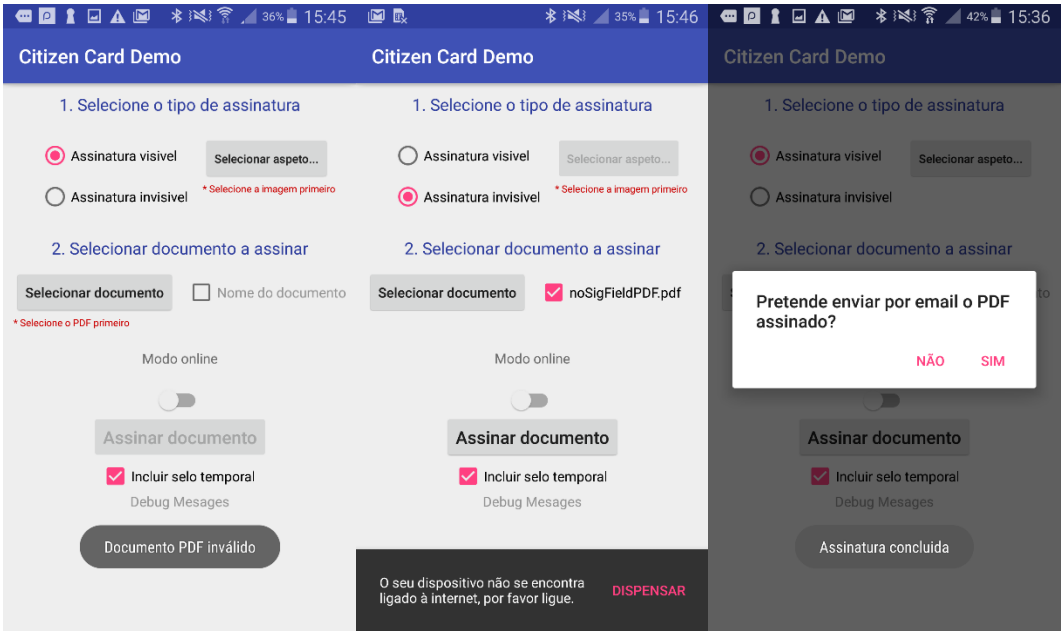



Figura 31- Exemplo de mensagens de *feedback* ao utilizador.

### 5.2.3.2. Desenvolvimento de modo *preview*

O modo *preview* da imagem de assinatura foi um dos requisitos que surgiram por parte do cliente ao longo do segundo semestre. Esta funcionalidade permite ao utilizador pré-visualizar a imagem de assinatura a inserir no documento em tempo real.

É permitido ao utilizador escolher qual a sua imagem *watermark*, imagem de assinatura manuscrita e também geração da assinatura textual com os dados do cartão. Após o utilizador escolher as imagens *watermark*, assinatura manual e textual é lhe permitido experimentar as várias combinações possíveis entre as mesmas, para no final ser gerada a imagem de assinatura final . Na Figura 32 é possível observar todas as combinações possíveis para a imagem de assinatura.


Figura 32- Combinações possíveis da imagem de assinatura.

Para que tudo isto fosse possível foi necessária a criação de uma classe chamada de *ImageSignatureGenerator* por parte do estagiário que com base nos parâmetros de entrada faz o processamento todo da imagem final. Em seguida será explicado o funcionamento deste processamento de imagem que inclui o redimensionamento em falta no PDFBox-Android.

Supondo que o utilizador quer incluir tudo na sua imagem de assinatura, tal como ilustrado no canto superior direito da Figura 31. Esta imagem é composta por três imagens diferentes, do lado direito temos uma imagem textual e do lado esquerdo temos o *watermark* em plano de fundo com a assinatura manual por cima deste. A Figura 32 permite ver com maior clareza esta combinação.

A classe *ImageSignatureGenerator* é composta por um método principal chamado de *generateSignatureImage* e é dentro deste que é gerada a imagem de assinatura consoante os parâmetros de entrada. Em seguida serão referidos quais os métodos privados desta classe e qual o seu papel neste processo de gerar a imagem de assinatura.

***scaleImage***: Este método tem como função de redimensionar a imagem de entrada de forma a que esta não fique "espalmada", mantendo os rácios comprimento/altura do campo onde esta será inserida. Na Figura 33 é possível observar com maior clareza a função deste método.

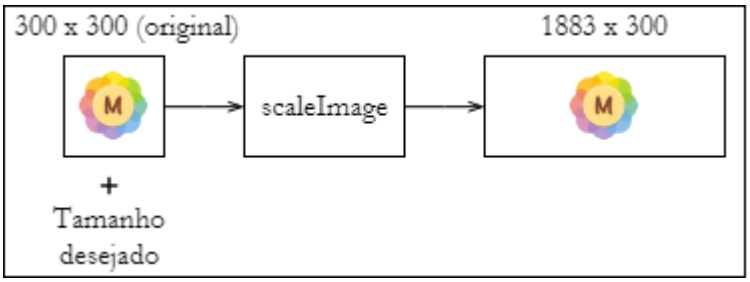



Figura 33- - Exemplificação do comportamento da função *scaleImage*.

***scaleMaintainAspectRatio***: Este método tem como principal função em dado como parâmetro de entrada a imagem com rácio correto e tamanhos desejados, irá calcular o máximo de redimensionamento que poderá ser feito de forma a que o resultado esteja o mais próximo ao tamanho desejado, mas nunca menor que o mesmo (no comprimento e largura). Por exemplo, dado uma imagem 1883x300 e tamanho desejado de 810x129, o maior redimensionamento a ser feito mantendo o rácio da imagem será 2, sendo o resultado aproximadamente 942x150. Na Figura 34 é possível observar este exemplo dado.

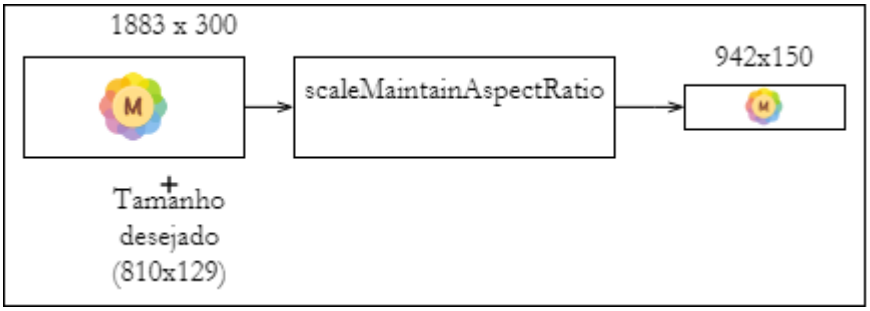



Figura 34- Exemplificação do comportamento da função *scaleMaintainAspectRatio*.

***createScaledBitmap***: Este método tem como principal função de redimensionar uma imagem sem manter os rácios. Apenas foi utilizada para fazer os ajustes finais na imagem proveniente da função *scaleMaintainAspectRatio*. Por exemplo, como foi possível observar na Figura 34, o tamanho desejado de 810x129 não foi alcançado. Por forma a ser é aplicado este método para que a imagem caiba perfeitamente no campo de previsualização e de assinatura. Caso este método fosse aplicado antes de todos os outros, a imagem ficaria com o aspeto "espalmado". Na Figura 35 é possível observar o aspeto final da imagem de assinatura com os tamanhos corretos.

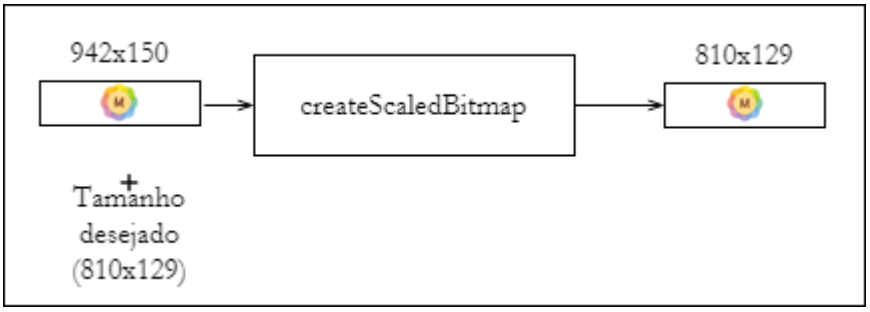



Figura 35- Exemplificação do comportamento da função *createScaledBitmap*.

***turnBackgroundTransparent***: Este método tem como principal função de converter o fundo da imagem para transparente. É particularmente útil quando se quer ter uma imagem sobreposta a outra.

***mergeAndCenter***: Este método tem como principal função de sobrepor uma imagem sobre outra, onde a primeira imagem passada ficará por detrás da segunda.

***combine***: Este método tem como principal função de combinar duas imagens uma ao lado da outra, onde a primeira imagem passada ficará à esquerda da segunda imagem.

***scaleBitmapToFit***: Este método é constituído pelos três métodos já referidos pela ordem apresentada, sendo estes o *scaleImage*, *scaleMaintainAspectRatio* e *createScaledBitmap*. Este método apenas serve pare encapsular os três métodos de *scale*. O seu comportamento pode ser mais facilmente analisando a Figura 36 .

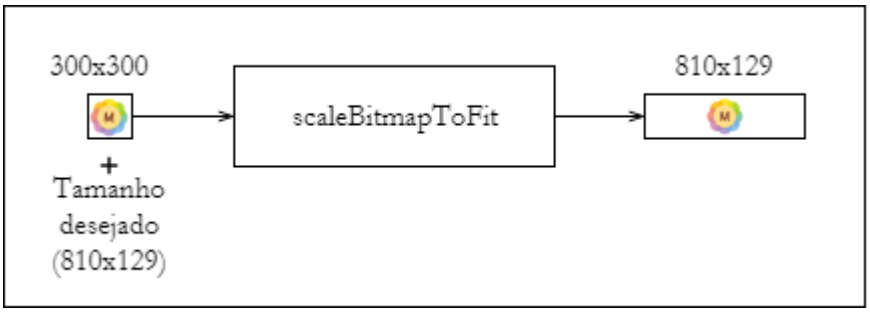



Figura 36- Exemplificação do método *scaleBitmapToFit*.

***makeTextSignature***: Este método consiste na geração da imagem de assinatura textual. Consoante o cartão que estiver inserido e as dimensões desejadas irá buscar o nome do utilizador e país ao cartão e escreverá nos respetivos sítios. O seu comportamento é facilmente compreendido observando a Figura 37 .

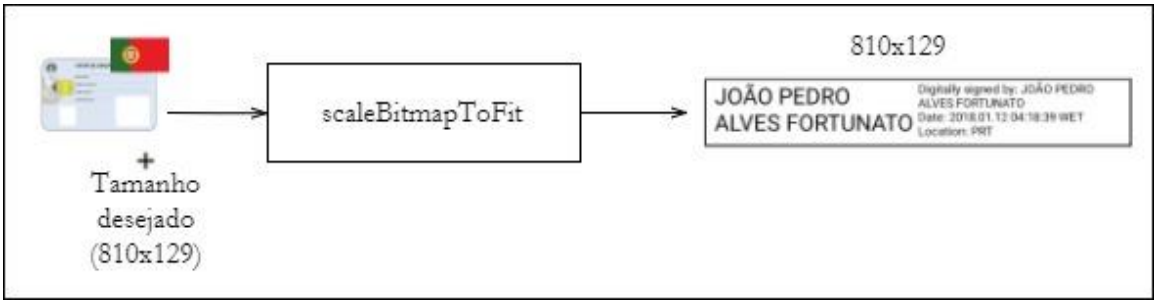



Figura 37- Exemplificação do método *makeTextSignature*.

***generateSignatureImage*** O *generateSignatureImage* é constituído pelo *scaleBitmapToFit* juntamente com os métodos de utilidade *turnBackgroundTransparent* ,*mergeAndCenter* e *combine*. O *generateSignatureImage* consoante os dados de entrada irá então gerar a imagem de assinatura. O método *generateSignatureImage* pode ser mais facilmente compreendido analisando o diagrama de atividades representado.

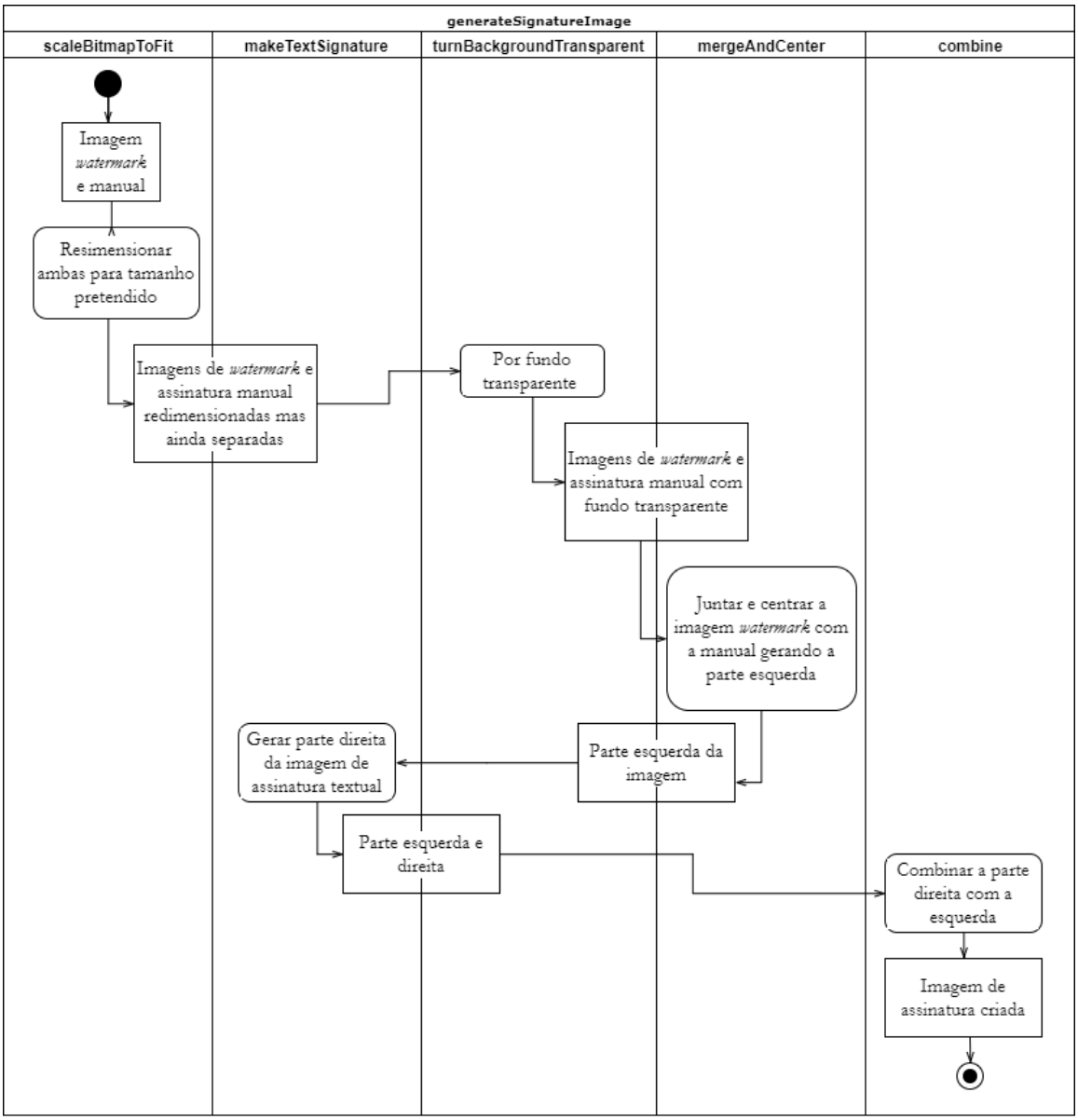



Figura 38- Diagrama de atividades do método *generateSignatureImage*.

O diagrama representado pela Figura 38 representa o *workflow* do processo de geração de imagem de assinatura quando o utilizador fornece os parâmetros: imagem *watermark*, imagem de assinatura manual e quer incluir a assinatura textual. A imagem gerada pode ser observada na Figura 39, onde a vermelho é a parte esquerda e a verde é a parte direita, tal como referido pelo diagrama de atividades.


Figura 39- Exemplo da imagem de assinatura.

### Assinatura visível

Possuindo agora uma maneira de fazer o redimensionamento da imagem da assinatura a inserir de forma a caber exatamente no campo de assinatura a assinar, foi apenas necessário usar o mesmo procedimento utilizado para a assinatura invisível, mas agora com a utilização da classe *PDVisibleSignDesigner* para inserir a imagem de assinatura no PDF. O problema da imagem não ser inserida nas coordenadas do campo de assinatura também foi solucionado pelo estagiário obtendo as coordenadas do campo a assinar, selecionando o canto inferior esquerdo.

Na Figura 40 é possível observar a imagem de assinatura presente no documento PDF.

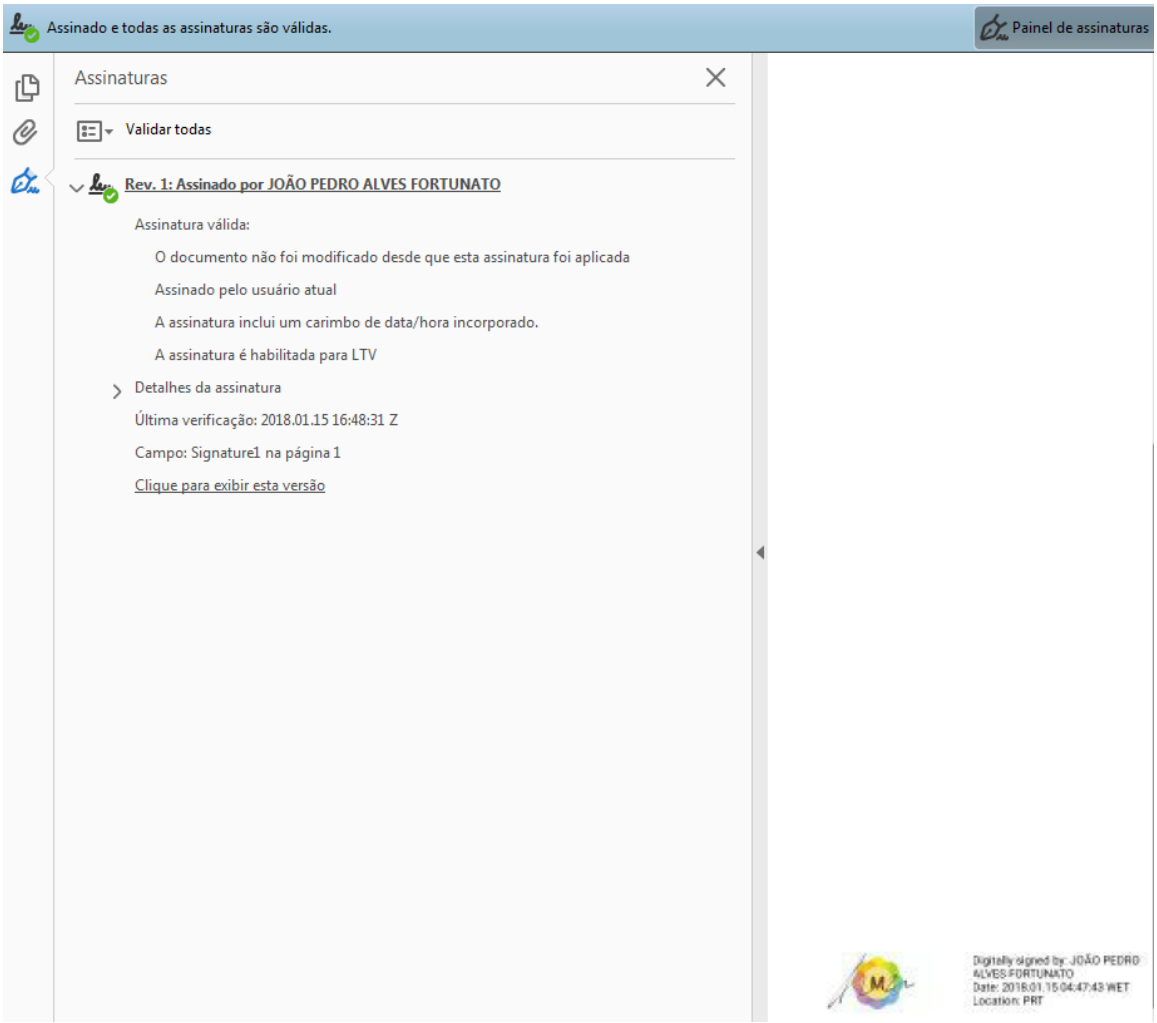



Figura 40- Exemplo de assinatura visível vista no Adobe Reader

#### 5.2.3.3. Desenvolvimento do componente de timestamp

O desenvolvimento da componente de timestamp foi relativamente simples. Usando apenas o código Open Source disponível pela Apache em [42] juntamente com o PDFBox foi possível adicionar o *timestamp* à assinatura digital, usando para isso o mesmo processo de assinatura já referido com pequenas alterações. Na Figura 40 é possível observar que a assinatura visível feita possui *timestamp* devido ao facto de que o Adobe Reader nos dizer "A assinatura inclui um carimbo de data/hora incorporado.

No entanto o servidor de *timestamp* é da posse do estado, logo para além de estar limitado a um máximo de 20 pedidos em cada 20 minutos este também costuma estar indisponível. Para evitar que a aplicação terminasse abruptamente caso o servidor estivesse em baixo, foi implementado um mecanismo pelo estagiário que verifica primeiro o estado do servidor. Caso o servidor esteja em baixo será usada a hora do dispositivo do assinante e é emitida uma mensagem de aviso ao utilizador de que tal foi feito.

Na Figura 41está representada uma assinatura sem *timestamp* efetuada caso o servidor não esteja disponível.

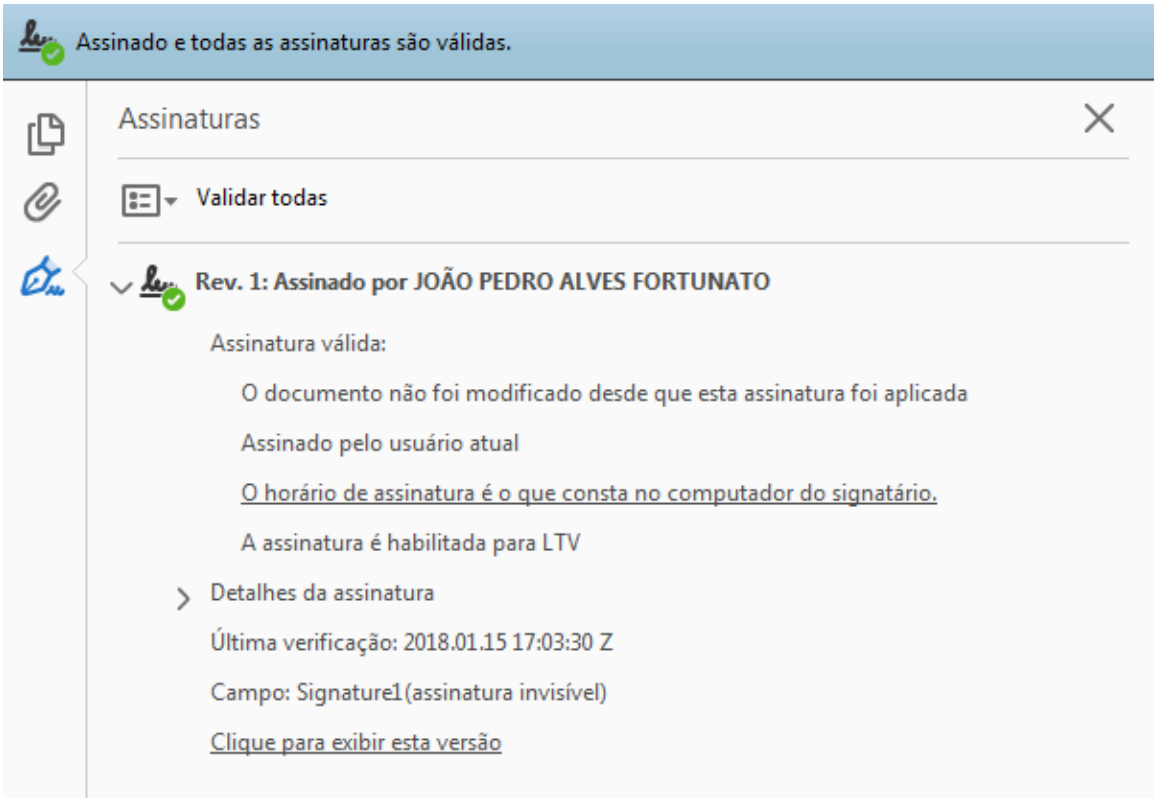



Figura 41-Exemplo de assinatura sem *timestamp*.

#### 5.2.3.4. Desenvolvimento do componente enviar por email

A funcionalidade de enviar o PDF assinado por email também foi bastante simples, apenas foi necessário anexar a *stream* do ficheiro PDF assinado a um *intent* do tipo *email* lançada pela atividade principal. Ao enviar esta *intent* para o sistema, aparecerão as aplicações que consigam responder a este tipo de *intents*, neste caso serão as aplicações de email que o utilizador tiver instaladas no seu dispositivo Android.. Ao selecionar a aplicação do Gmail (por exemplo), automaticamente será adicionado como Anexo o PDF assinado a este e-mail conforme ilustrado na Figura 42 :

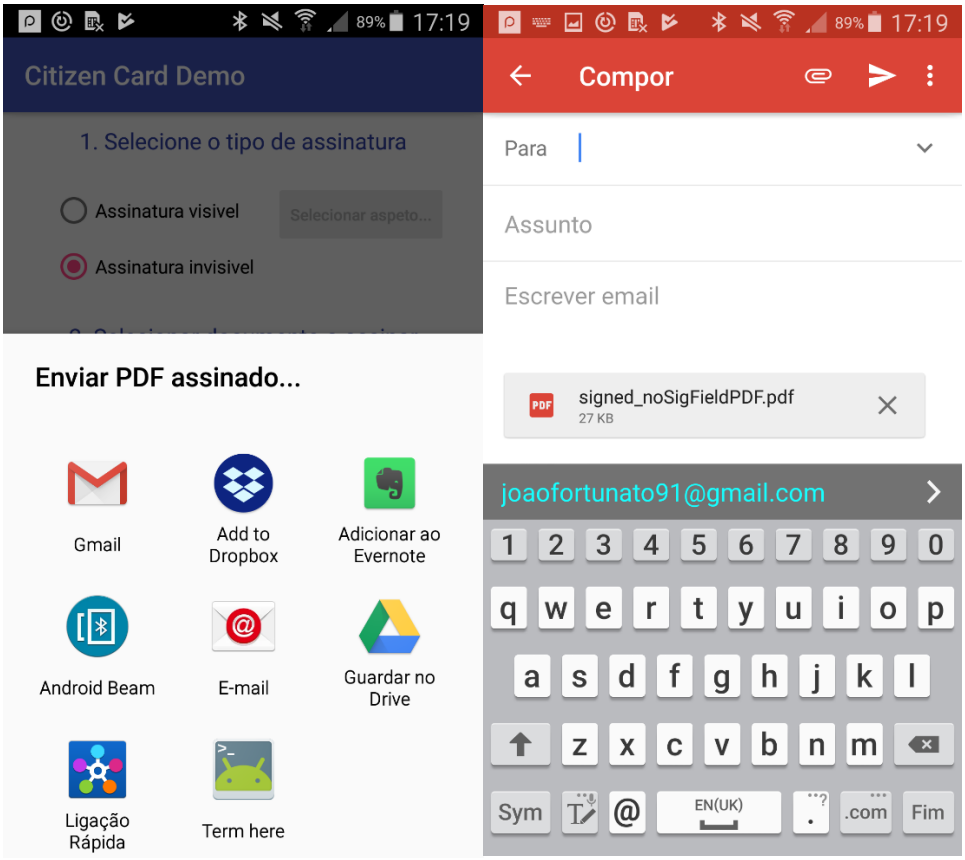



Figura 42- Exemplo do envio do PDF assinado por email.

# Capítulo 6
# Validação de requisitos

Tal como em qualquer projeto de *software*, com o final da fase de desenvolvimento surge a necessidade da realização de testes de forma a verificar se a solução desenvolvida vai de encontro aos requisitos especificados. Neste capitulo serão testados os requisitos funcionais requisitos não funcionais, testes de versão e testes de utilização de recursos.

## 6.1. Metodologia de testes

Considerando o tipo de projeto como não-critico, foi decidido pelo estagiário de que testes do tipo *Blackbox* seriam suficientes de forma a garantir a qualidade do *software* desenvolvido.

Cada caso de uso será testado de forma a garantir a qualidade de todas as funcionalidades implementadas. Antes de executar os testes é necessário criar uma tabela de valores limite (VL) e classes de equivalência (EC) de forma a definir o que vai ser testado. Apos definir o que irá ser testado, uma tabela de casos de teste poderá ser construída e os seus respetivos testes poderão ser executados de forma a testar tudo (ou o que faz sentido) que foi definido na tabela de EC e VL.

### 6.1.1.Critérios de passagem

Os testes realizados só poderão ter dois resultados, podendo estes ser:

**Passou**: Assinalados com um `V´ nas tabelas de casos de teste (Anexos 14,16,18,20, e 22), estão presentes os testes que passaram no respetivo caso de teste;

**Não passou**: Assinalados com um `X´ nas tabelas de resultados dos testes, estão presentes os testes que não passaram no respetivo caso de teste.

Após a execução de todos os testes do respetivo caso de uso, a decisão final referente ao seu resultado é definida consoante os seguintes critérios:

OK: Todos os testes executados para o respetivo caso de uso passaram;

POK: Alguns dos testes executados para o respetivo caso de uso falharam, onde os testes que falharam não impediram o caso de uso de se realizar com sucesso;

NOK: Todos os testes executados para o respetivo caso de uso falharam ou existiu pelo menos um teste que não passou que comprometeu a realização do caso de uso.

### 6.1.2. Ambiente de testes

Para realizar os testes executados foram usados *smartphones* Android e não emuladores devido ao facto de que não era possível usar o leitor de cartões ligado ao computador para testar a aplicação.

#### Versões de SDK Android e recursos utilizados

É de alguma importância referir que os testes referidos na secção seguinte foram executados em várias versões do SDK Android. Isto tem como principal objetivo de atingir a maior quota

de mercado possível de *smartphones* Android, para que não existam quaisquer barreiras caso um utilizador queira ter a aplicação no seu *smartphone*. Para isso foram utilizados quatro dispositivos físicos da empresa onde as versões mais comuns e recentes são abrangidas. As características mais relevantes destes dispositivos de teste poderão ser consultadas na Tabela 23:

Tabela 23- Características dos *smartphones* Android de testes utilizados.

| Marca/modelo | Versão do SDK | CPU | Memória RAM |
|---|---|---|---|
| BQ Aquaris E5 | 19 | 2.0 GHZ *Octa Core* | 2GB |
| Samsung Galaxy Note 3 | 21 | 2.3GHz *Quad Core* | 3GB |
| BQ Aquaris M5 | 23 | 2.0 GHZ *Octa Core* | 3GB |
| Google Pixel 2 | 27 | 2.35Ghz *Octa Core* | 4GB |

Relativamente aos recursos utilizados não foi possível dizer que quantidade de RAM e CPU utilizados ao correr a aplicação porque para isso seria necessário que o cabo USB tivesse ligado diretamente ao computador e não só ao leitor de cartões. No entanto é possível observar que a aplicação corre com fluidez nos dispositivos de teste utilizados.

## 6.2. Validação dos Requisitos Funcionais

Os testes aos requisitos funcionais foram em grande parte desenvolvidos e aplicados à medida que eram desenvolvidas as funcionalidades da aplicação. Estes testes tiveram como principal objetivo de verificar se a funcionalidade em desenvolvimento se encontrava a funcionar de acordo com o especificado. Nos casos em que o desenvolvimento de uma funcionalidade implicava a alteração de outra já testada e aceite, seria necessário voltar a testar esta mesma funcionalidade e todas as suas dependentes de forma a garantir que estas não tinham sido comprometidas. É também de alguma importância referir que as assinaturas feitas pelo modo *remote* não foram testadas, visto que o SDK de cartão usado neste estágio já se encontra desatualizado.

Tal como especificados no capitulo de Análise de Requisitos, os requisitos funcionais identificados e testados foram os seguintes:

- RF selecionar PDF;
- RF Fazer Assinatura Visível;
- RF Fazer Assinatura Invisível;
- RF Incluir *Timestamp*;
- RF Selecionar Imagem de Assinatura;
- RF enviar PDF por Email.

Os testes realizados a estas funcionalidades têm como principal objetivo de testar como a aplicação se comporta face a vários tipos de dados de entrada, devendo sempre evitar que a aplicação termine abruptamente e dar *outputi* correto.

Tal como especificado na metodologia de testes, foram feitas primeiro as tabelas de CE e VL (Anexos 13,15,17,19,21 respetivamente) para os respetivos casos de uso de forma a permitir criar casos de teste mais completos para depois serem executados (Anexos 14, 16, 18, 20, 22).

### 6.2.1. Testes ao Use Case (UC) Selecionar PDF

#### Resultados da primeira ronda de testes

Ao executar a primeira ronda dos casos de teste deste UC é possível observar que existe um deles que falhou, nomeadamente o teste com ID3. Como é possível observar no Anexo 14, a falha deste teste implica a falha do caso de uso, logo o resultado desta primeira ronda de testes deste UC é de NOK. Com este resultado será necessário fazer uma correção no código de forma a suportar a inserção de documentos PDF inválidos (vazios) sem que a aplicação termine abruptamente, em vez disso a aplicação deverá emitir uma mensagem do tipo "Documento PDF inválido".

#### Resultados da segunda ronda de testes

Com a correção do *bug* identificado anteriormente, todos os testes passaram com sucesso, sendo então o resultado final de OK para este caso de uso.

### 6.2.2. Testes ao UC Fazer assinatura Visível

Ao executar a primeira ronda de testes para este UC é possível observar que o resultado final é o de OK, isto devido ao facto de que todos os testes presentes no Anexo 11 passaram com sucesso. Isto foi possível também devido ao facto de que existiu algum cuidado durante o desenvolvimento desta funcionalidade nas interações com a UI, como irá ser referido posteriormente na validação da usabilidade. Como o resultado final deste caso de testes passou com sucesso, já não foi necessário fazer uma segunda ronda, dado que não havia nada para corrigir no código.

### 6.2.3. Testes ao UC Fazer assinatura Invisível

De forma semelhante aos testes anteriores, ao executar a primeira ronda de testes para este UC é possível observar que o resultado final é o de OK, isto devido ao facto de que todos os testes presentes no Anexo 18 passaram com sucesso. Também não foi necessário fazer uma segunda ronda de testes para este UC.

### 6.2.4. Testes ao UC Incluir *Timestamp*

De forma semelhante aos testes anteriores, ao executar a primeira ronda de testes para este UC é possível observar que o resultado final é o de OK, isto devido ao facto de que todos os testes presentes no Anexo 20 passaram com sucesso. Também não foi necessário fazer uma segunda ronda de testes para este UC.

### 6.2.5. Testes ao UC Selecionar Imagem de Assinatura

#### Resultados da primeira ronda de testes

Ao executar a primeira ronda dos casos de teste deste UC é possível observar que existem dois deles que falharam, nomeadamente o teste com ID11 e 12. Como é possível observar no Anexo 22, a falha destes dois testes implica a falha do caso de uso, logo o resultado desta primeira ronda de testes é de NOK. Com este resultado será necessário fazer uma correção no código de forma a suportar a inserção de imagens do tipo PNG inválidas (vazias) sem que a aplicação termine abruptamente, em vez disso a aplicação deverá emitir uma mensagem do tipo "Imagem inválida".

**Resultados da segunda ronda de testes**

Com a correção dos *bugs* identificados anteriormente, todos os testes passaram com sucesso, sendo então o resultado final de OK para este caso de uso. Terminando assim com sucesso a validação de requisitos funcionais.

### 6.2.6. Testes ao UC Enviar PDF Por Email

Relativamente aos testes deste UC apenas foi testado se no fim do processo de assinatura se era possível enviar o PDF assinado através da aplicação de email que o utilizador escolhesse. Como é um teste bastante simples não se justificava a criação de uma tabela de casos de teste com apenas um único teste.

**Resultados da primeira ronda de testes**

Na primeira ronda de teste todas as versões do SDK Android passaram à exceção da versão 27. Isto devido ao facto de que é a versão mais recente, entretanto alguns componentes foram alterados pela Google. Como este erro não permitia que UC fosse executado com sucesso nesta versão, o resultado final foi o de NOK para este caso de uso.

**Resultados da segunda ronda de testes**

Após a correção do *bug* encontrado anteriormente foi possível executar este teste com sucesso para todas as versões, sendo agora o resultado de OK para este caso de uso.

**Sumário de resultados dos testes realizados**

Na Tabela 24 é possível observar um sumário dos resultados obtidos após execução dos testes aos requisitos funcionais da aplicação.Tabela 24- Sumário dos testes aos requisitos funcionais realizados.

| **Nome do teste** | **Data** | **1ª Ronda de testes** | | **2ª Ronda de testes** | |
|---|---|---|---|---|---|
| | | **Nº testes que passaram V** | **Nº testes que não passaram X** | **Nº testes que passaram V** | **Nº testes que não passaram X** |
| Teste do UC Selecionar PDF | De 04-01-2018 a 08-01-2018 | 2 | 1 | 3 | 0 |
| Teste do UC Fazer Assinatura Visível | De 04-01-2018 a 08-01-2018 | 10 | 0 | 10 | 0 |
| Teste do UC Fazer Assinatura Invisível | De 04-01-2018 a 08-01-2018 | 7 | 0 | 7 | 0 |
| Teste do UC Incluir *Timestamp* | De 04-01-2018 a 08-01-2018 | 4 | 0 | 4 | 0 |
| Teste do UC Selecionar Imagem de Assinatura | De 04-01-2018 a 08-01-2018 | 10 | 2 | 12 | 0 |
| Teste do UC Enviar PDF por Email | De 04-01-2018 a 08-01-2018 | 0 | 1 | 1 | 0 |
| **Total** | | **33** | **4** | **37** | **0** |

Podemos concluir que dos *bugs* encontrados todos foram resolvidos e que os casos de uso testados estão a funcionar sem *bugs* que comprometam a execução dos UC, garantindo uma boa qualidade de produto. É de alguma importância salientar de que existiram mais rondas de testes antes da destas duas rondas finais de forma a criar uma versão de demonstração da aplicação com as funcionalidades a funcionar primeiro. Estas várias rondas de teste foram feitas à medida que o código era desenvolvido e por questões de simplicidade não foram contabilizadas.

## 6.3. Validação dos Requisitos Não Funcionais

Conforme especificado na Tabela 5 do capitulo de Análise de Requisitos, existem dois requisitos não funcionais importantes para a aplicação, nomeadamente a usabilidade e a segurança.

### Usabilidade

Como foi referido no Capitulo 3, a usabilidade pode ser alcançada através de mensagens de *feedback* ao utilizador e também com a utilização de um leitor de cartões com *microUSB*.

Grande parte da usabilidade provem das mensagens de *feedback* emitidas por parte da aplicação, de forma a guiar o utilizador pelos vários *workflows* do processo de assinatura digital. Quando algo de errado ocorre, como por exemplo a inserção de um documento PDF ou imagem inválida, a aplicação emite uma mensagem de aviso de forma a que o utilizador saiba o que correu mal.

Analisando as tabelas de casos de teste presentes nos Anexos 14, 16, 18, 20 e 22, é possível verificar que existem testes que garantem que caso algo falhe, deverão ser emitidas mensagens de aviso ao utilizador relativamente ao problema ocorrido. Por exemplo, na primeira ronda de testes, o teste com ID3 deveria não só evitar aceitação do PDF inválido, como também emitir uma mensagem que informasse o utilizador do ocorrido. Analisando também a Tabela 19, podemos concluir que todos os casos de teste passaram, logo quando alguma coisa correr mal, irá sempre ser emitida uma mensagem relativamente a essa ocorrência, garantindo uma boa usabilidade da aplicação.

Não foi possível testar em Android a utilização de leitores de cartão com *microUSB* devido à falta do mesmo na empresa, no entanto na versão para iOS foram utilizados leitores deste tipo sem qualquer problema. É possível afirmar que é muito provável que este tipo de leitores sejam compatíveis com a aplicação Android caso sejam adquiridos futuramente pela empresa.

### Segurança

Como foi referido no capitulo 3, a segurança poderá também pode ser reforçada através da utilização de um leitor de cartões com *pinpad* embutido. Infelizmente o SDK de acesso ao cartão ainda não suporta este tipo de leitores, logo será algo a implementar futuramente. No entanto o facto de que este tipo de leitores não seja suportado não implica que a aplicação não seja segura.

O próprio processo de assinatura em si já é algo muito seguro e impossível de forjar em tempo útil, principalmente com a continua atualização dos algoritmos de *hashing* utilizados neste mesmo processo. A utilização do leitor de cartões com *pinpad* embutido só iria acrescentar um nível de segurança ao PIN de assinatura, no entanto seria necessário possuir também o cartão de cidadão fisicamente para fazer uma assinatura digital pelo nome de outrem.

De forma a aumentar a segurança dos dados do utilizador, para versões do SDK Android superiores à 22, serão pedidas todas as permissões consideradas sensíveis em *runtime* com a respetiva razão para a qual as mesmas irão ser usadas no contexto da aplicação. Olhando para a Figura 43, é possível observar que por exemplo, ao carregar no de seleção de documento num dispositivo Android onde a sua versão é superior à 22, aparecerá a uma mensagem a dizer que necessita de permissão de acesso a ficheiros para executar esta operação. Só depois do utilizador permitir é que a funcionalidade poderá ser executada, podendo o utilizador a qualquer momento retirar a permissão.

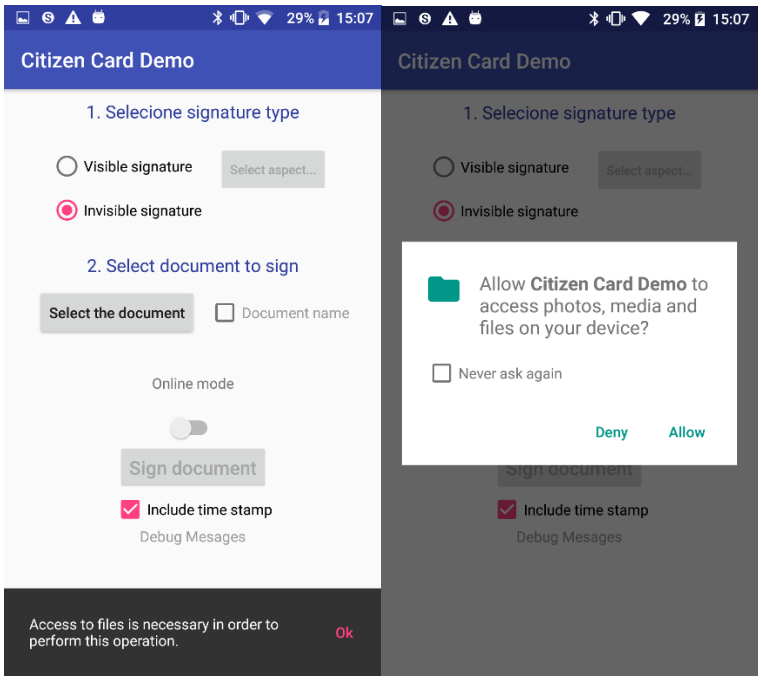



Figura 43 Pedido de permissões em *runtime*.

Relativamente à utilização da internet no processo de assinatura, este só é necessário caso o utilizador queira uma assinatura com *timestamp* e posterior envio por e-mail, reduzindo assim parte do risco de ter o dispositivo "aberto ao mundo".

Podemos concluir também que embora a aplicação ainda não suporte leitores de cartão com *pinpad* embutido, a mesma continua a ser uma aplicação bastante segura devido à tecnologia de segurança que utiliza dentro no processo de assinatura.

# Capítulo 7
# Conclusões e Trabalho Futuro

## 7.1. Trabalho futuro

Com a conclusão deste estágio vale a pena ainda referir funcionalidades surgiram durante o processo de investigação de assinaturas digitais feito inicialmente. São relativos a possíveis melhoramentos a fazer na solução atual de assinatura digital e sendo que a empresa mostrou interesse em perseguir em versões futuras da aplicação, serão explicados brevemente algumas destas funcionalidades.

### Long Term Validation

Conhecido como LTV, este conceito tem como principal objetivo guardar o estado do documento PDF quando foi assinado, incluindo em cada assinatura os dados necessários à validação da mesma, nomeadamente as *certificate revocation lists*(CRLs) e/ou os *online certificate status protocol* (OCSP). Quando um documento é aberto pelo Adobe Reader ou outro suporte para esta especificação, se não existirem estes dados na assinatura, no processo de validação de assinaturas irá ser feita uma verificação dos mesmos *online* de forma a validar se o certificado usado para fazer aquela assinatura se encontra válido ou não. Contrariamente, se estes dados já se encontrarem inseridos juntamente da assinatura, o leitor de PDF irá dizer que a assinatura é válida e não irá fazer a verificação *online*. A principal vantagem de ter uma assinatura LTV é o aumento da validade da assinatura. Enquanto uma assinatura digital normalmente dura consoante o tempo de vida do certificado usado para a criar, uma assinatura LTV tem tempo ilimitado. Isto acontece porque toda a informação que valida a assinatura está presente juntamente com a mesma e se o certificado usado era válido quando foi usado, faz sentido que naquele intervalo de tempo continue a ser.

Este conceito é mais facilmente compreendido utilizando a seguinte analogia, um utilizador fez assinaturas com o seu certificado nos entre janeiro e março, no entanto o seu certificado/cartão foi perdido. Quando o cartão finalmente foi cancelado, invalidando o seu certificado, isto faria com que todas as assinaturas feitas pelo proprietario fossem inválidas porque durante o processo de validação online o Adobe diria que o certificado estava revogado. O mesmo não aconteceria se a informação de validação estivesse presente na assinatura, até porque embora o cartão tenha sido perdido, as assinaturas feitas eram válidas na altura que foram feitas, esta é a grande vantagem de ter assinaturas LTV.

### Seleção dinamica de local de assinatura

A seleção dinâmica do local de assinatura consiste em conseguir apresentar as páginas do PDF e ao mesmo tempo desenhar um retângulo no local pretendido para a assinatura ficar. A Figura 44 ilustra este componente.

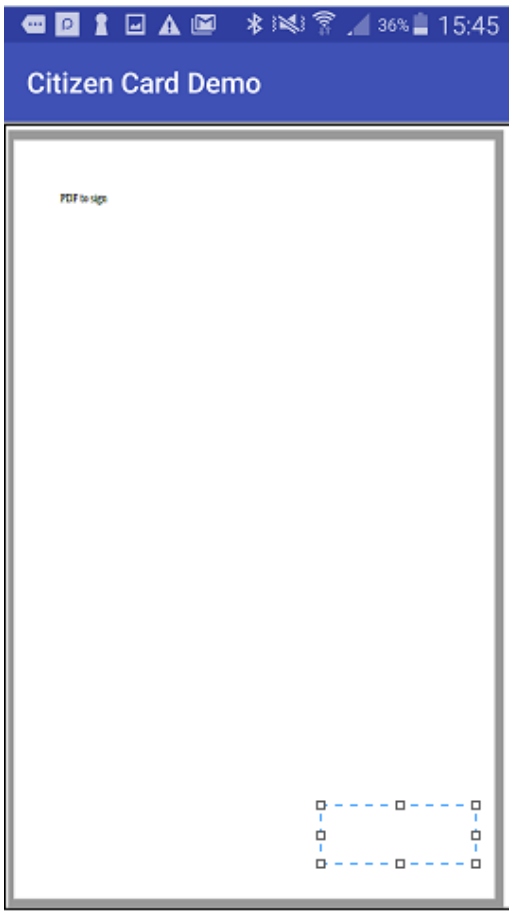



Figura 44- Exemplificação de escolha dinâmica de local de assinatura.

## 7.2. Conclusões

Com a conclusão deste relatório final de estágio espero ter aplicado da melhor forma as várias técnicas de engenharia de software aprendidas ao longo deste mestrado, bem como uma boa análise de todos os aspetos considerados relevantes ao projeto a desenvolver. Espero que com a minha contribuição *Open Source* o conceito de assinatura digital móvel venha a ser cada vez mais utilizado por forma a que todos possam usufruir de todas as vantagens que este poderá trazer ao quotidiano de todos. Espero também ter correspondido positivamente às espectativas de todos os elementos envolvidos neste projeto e de ter conseguido transmitir de forma clara e concisa os conhecimentos obtidos ao longo deste ano.

Por fim, considero toda esta experiência profissional adquirida neste estágio de grande impotância a qualquer aluno desta área e estou eternamente grato a toda a simpatia, compreensão e flexibilidade proporcionados pela empresa acolhedora.

# Referências

[1] Zhang L., Shan L., Wang J. (2012) Summary of Digital Signature. In: Zeng D. (eds) Advances in Control and Communication. Lecture Notes in Electrical Engineering, vol 137. Springer, Berlin, Heidelberg.

[2"Microsoft - Assinaturas Digitais e Certificados" disponível em :<https://support.office.com/pt-pt/article/Assinaturas-digitais-e-certificados-8186cd15-e7ac-4a16-8597-22bd163e8e96> junho de 2017

[3] "Transform business processes with electronic and digital signature solutions Adobe Sign White Paper" disponível em : <https://acrobat.adobe.com/content/dam/doc-cloud/en/pdfs/adobe-transform-business-processes-with-electronic-and-digital-signature-solutions.pdf> junho 2017

[4] Differences between digital and electronic signatures disponível em :<https://www.approveme.com/e-signature/difference-between-digital-signature-and-electronic-signature/> junho 2017

[5] "How do digital signatures work" disponível em :<https://www.globalsign.com/en/blog/how-do-digital-signatures-work/> junho 2017

[6] "Adobe Overview of the electronic signature law in the EU" disponível em : <https://acrobat.adobe.com/content/dam/doc-cloud/en/pdfs/overview-of-electronic-signature-law-in-the-EU.pdf> junho 2017

[7] "Adobe Global guide electronic signature law EU" disponível em: <https://acrobat.adobe.com/content/dam/doc-cloud/en/pdfs/document-cloud-global-guide-electronic-signature-law-ue.pdf> junho 2017

[8] "eIDAS Regulations" disponível em :<https://ec.europa.eu/digital-single-market/en/policies/trust-services-and-eidentification> junho 2017

[9] Lovegrove, William S., and David F. Brailsford. "Document analysis of PDF files: methods, results and implications." Electronic Publishing--Origination, Dissemination and Design 8.3 (1995): 207-220.

[10] Hassan, Tamir. "Object-level document analysis of PDF files." Proceedings of the 9th ACM symposium on Document engineering. ACM, 2009.

[11] Adobe Systems Incorporated, Portable Document Format Reference Manual, AddisonWesley, Reading, Massachusetts, June 1993.

[12] ISO 15930-6:2003, Graphic technology — Prepress digital data exchange using PDF — Part 6: Complete exchange of printing data suitable for colour-managed workflows using PDF 1.4 (PDF/X-3). [13] ISO 19005-1:2005, Document management — Electronic document file format for long-term preservation -- Part 1: Use of PDF 1.4 (PDF/A-1).

[14] ISO 24517-1:2007, Document management — Engineering document format using PDF — Part 1: Use of PDF 1.6 (PDF/E-1).

[15] Bruno Lowagie. "*Digital Signature for PDF Documents - White Paper*" (2012)

[16] "Digital Signatures in PDF" disponível em :<https://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf> junho 2017

[17] "*Secure Hash Algorithm"* disponível em :<https://en.wikipedia.org/wiki/Secure_Hash_Algorithms> junho 2017

[18] Maurer, Ueli. "*Modelling a public-key infrastructure.*" *Computer Security—ESORICS 96*. Springer Berlin/Heidelberg, 1996.

[19] Burrows, James H. *Secure hash standard*. DEPARTMENT OF COMMERCE WASHINGTON DC, 1995.

[20] Microsoft - "*Open Packaging Conventions Fundamentals*" disponível em :<https://msdn.microsoft.com/en-us/library/windows/desktop/dd742818(v=vs.85).aspx#introduction> junho 2017

[21] OpenOffice OPC - "*Open Packaging Conventions*" disponível em :<https://wiki.openoffice.org/wiki/OOXML/Open_Packaging_Conventions_(OPC)> junho 2017

[22] W3C - "*XML-Signature Syntax and Processing*" disponível em :<https://www.w3.org/TR/2002/REC-xmldsig-core-20020212/> junho 2017

[23] W3C - "*XML advanced electronic signature (XaDES)*" disponível em :<https://www.w3.org/TR/XAdES/> junho 2017

[24] Artigo Eldos - "*Introduction to XML Advanced Electronic Signatures (XAdES)*" disponível em :<https://www.eldos.com/security/articles/7014.php?page=1> junho 2017

[25] Características Técnicas do Leitor e do Cartão de Cidadão Português disponível em :<https://www.autenticacao.gov.pt/documents/10179/11463/Caracter%C3%ADsticas+T%C3%A9cnicas+dos+Leitores+Base+%28Desktop%29%20do+Cart%C3%A3o+de+Cidad%C3%A3o/f0ad2328-19f6-453a-ae1e-13ab67912022>

[26] Cartão de Cidadão disponível em :<https://www.autenticacao.gov.pt/o-cartao-de-cidadao> junho 2017

[27] PDFBox-Android disponível em :<https://github.com/TomRoush/PdfBox-Android> junho 2017

[28] Apache POI disponível em :<https://poi.apache.org/> junho 2017

[29] ISO 7816 disponível em :<http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816.aspx> junho 2017

[30] EMV level 1 disponível em :<https://www.emvco.com/approvals.aspx?id=84> junho 2016

[31] CT API disponível em :<https://www.tuvit.de/fileadmin/Content/ TUV_IT/ pdf/ Downloads/CT-API/ct-api-englisch.pdf>

[32] Chip Card Interface Device disponível em :<http://www2.electron.frba.utn.edu.ar/~afurfaro/descargas/Universal%20Serial%20Bus/Especificaciones%20de%20clases/ChipSmart%20Card%20Interface%20Devices%20(CCID)/ccid_classspec_1_00a.pdf> junho 2017

[33] ISO 7186 disponível em :<https://www.iso.org/obp/ui/#iso:std:iso:7186:ed-3:v1:en> junho 2017

[34] Decreto-Lei n.º 88/2009 de 9 de abril disponível em :<http://www.dgpj.mj.pt/sections/leis-da-justica/pdf-ult2/decreto-lei-n-88-2009-de/downloadFile/file/DL%2088.2009.pdf?nocache=1239264486.31> junho 2017

[35] Winckler, Marco, and Marcelo Soares Pimenta. "Avaliação de usabilidade de sites web." *Escola de Informática da SBC SUL (ERI 2002) ed. Porto Alegre: Sociedade Brasileira de Computação (SBC)* 1 (2002): 85-137.

[36] P. D. M. Vieira, "*Module 3: Planning & Tracking – Planning & Monitoring*,"

2015.

[37] P. D. M. Vieira, "*Module 3: Planning & Tracking - Bottom-up Estimation*:

*A Real Process*," 2015.

[39] *Issue* reportado no PDFBox-Android disponível em :https://github.com/TomRoush/PdfBox-Android/issues/108

[40] *Pull request* feito pelo estagiário para a correção do *issue* reportado:< https://github.com/TomRoush/PdfBox-Android/pull/122>

[41] *Issue* reportado sobre problemas com assinatura visível disponível em :< https://github.com/TomRoush/PdfBox-Android/issues/132>

[42] Código de suporte da apache relativamente a *timestamps* disponível em :< https://svn.apache.org/viewvc/pdfbox/trunk/examples/src/main/java/org/apache/pdfbox/examples/signature/>

# Anexos

### Anexo 1- Caso de uso 1 - Selecionar documento

| | |
|---|---|
| **Nome** | Selecionar documento a assinar |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *Summary* |
| **Interessados e interesses** | Utilizador: Selecionar o documento PDF a assinar para futura assinatura |
| ***Trigger*** | O utilizador quer selecionar um documento PDF para assinar |
| **Pré-condição** | A aplicação está a correr normalmente |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O utilizador consegue carregar o ficheiro pretendido para futura assinatura, aparecendo um ✓ ao lado de onde insere o documento |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega no botão "Carregar documento"<br>2. O ator pode selecionar o documento PDF via UC *Selecionar documento PDF*<br>3. O ator carregou o ficheiro a assinar com sucesso |
| **Exceções** | |

## Anexo 2- Caso de uso 1.1 - Selecionar documento PDF

| | |
|---|---|
| **Nome** | Selecionar documento PDF |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Selecionar o documento a assinar do tipo PDF |
| **Trigger** | O utilizador x quer selecionar um documento para assinar do tipo PDF |
| **Pré-condição** | A aplicação está a correr normalmente e o menu de escolha do documento está presente<br>Permissões de acesso aos ficheiros do *smartphone* concedidas |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O utilizador selecionou com sucesso o documento PDF a assinar |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega "Selecionar o documento PDF a assinar"<br>2. O ator seleciona o documento<br>3. O ator selecionou documento a assinar com sucesso |
| **Exceções** | 2.1) Erro de documento inválido<br>2.1.a) É mostrado ao utilizador que ocorreu um erro indicando por *popup* a mensagem do género "Documento a assinar inválido, por favor verifique e tente outra vez" |

## Anexo 3- Caso de uso 1.2 - Selecionar documento Word

| | |
|---|---|
| **Nome** | Selecionar documento Word |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Selecionar o documento a assinar do tipo Word |
| **Trigger** | O utilizador x quer selecionar um documento para assinar do tipo Word |
| **Pré-condição** | A aplicação está a correr normalmente e o menu de escolha do documento está presente |
| **Última revisão** | 22 de junho de 2017 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O utilizador selecionou com sucesso o documento Word a assinar |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega seleciona o documento Word a assinar<br>2. O ator seleciona o documento e carrega "Ok"<br>3. O ator selecionou documento a assinar com sucesso |
| **Exceções** | 2.1) Erro de documento inválido<br>2.1.a) É mostrado ao utilizador que ocorreu um erro indicando por *popup* a mensagem "Documento a assinar não é .docx, por favor verifique e tente outra vez" |

## Anexo 4- Caso de uso 2- Fazer assinatura

| | |
|---|---|
| **Nome** | Fazer assinatura |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *Summary* |
| **Interessados e interesses** | Utilizador: Fazer assinatura digital no documento |
| **Trigger** | O ator quer fazer uma assinatura digital no documento |
| **Pré-condição** | A aplicação está a correr normalmente |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator faz uma assinatura digital com sucesso no documento |
| **Processo: Cenário de sucesso principal** | 1. O ator pode fazer dois tipos de assinatura:<br>  1. Assinatura visível via UC *Fazer assinatura visível*<br>  2. Assinatura invisível via UC *Fazer assinatura invisível*<br>2. O ator faz uma assinatura digital com sucesso, podendo agora enviar o PDF assinado por email via *UC Enviar PDF por email* |
| **Exceções** | |

## Anexo 5- Caso de uso 2.1- Fazer assinatura visivel

| | |
|---|---|
| **Nome** | Fazer assinatura visível |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Fazer assinatura digital visível no documento |
| **Trigger** | O ator quer fazer uma assinatura digital visível no documento |
| **Pré-condição** | • A aplicação está a correr normalmente<br>• Um documento PDF ou Word tem de estar selecionado via UC *Selecionar documento a assinar*<br>• Uma imagem tem de estar selecionada via UC *Selecionar imagem de assinatura*<br>• O cartão de cidadão tem que estar inserido |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator faz uma assinatura digital visível com sucesso no documento |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega no botão "Assinar documento"<br>2. O ator insere o seu PIN de assinatura<br>3. O ator carrega no botão do género "seguinte"<br>4. O ator faz uma assinatura digital visível com sucesso |
| **Exceções** | 4.a) O ator remove o cartão<br>Operação de assinatura digital visível é cancelada, notificando o ator.<br>4.b) O ator inseriu o PIN inválido<br>Operação de assinatura digital visível é cancelada, notificando o ator com uma mensagem do género PIN errado. |

## Anexo 6- Caso de uso 2.2- Fazer assinatura invisivel

| | |
|---|---|
| **Nome** | Fazer assinatura invisível |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Fazer assinatura digital invisível no documento |
| **Trigger** | O ator quer fazer uma assinatura digital invisível no documento |
| **Pré-condição** | • A aplicação está a correr normalmente<br>• Um documento PDF ou Word tem de estar selecionado via UC *Selecionar documento a assinar* |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator faz uma assinatura digital invisível com sucesso no documento |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega no botão "Assinar documento"<br>2. O ator insere o seu PIN de assinatura<br>3. O ator carrega no botão do género "seguinte"<br>4. O ator faz uma assinatura digital invisível com sucesso |
| **Exceções** | 4.a) O ator remove o cartão<br>Operação de assinatura digital invisível é cancelada, notificando o ator.<br>4.b) O ator inseriu o PIN inválido<br>Operação de assinatura digital invisível é cancelada, notificando o ator com uma mensagem do género PIN errado. |

**Anexo 7- Caso de uso 3- Incluir *timestamp***

| | |
|---|---|
| **Nome** | Incluir *timestamp* |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: O utilizador quer fazer uma assinatura digital com *timestamp* |
| **Trigger** | O utilizador quer fazer uma assinatura digital com *timestamp* |
| **Pré-condição** | • Documento selecionado<br>• Imagem de assinatura selecionada<br>• Cartão de cidadão inserido |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator consegue fazer uma assinatura digital com *timestamp* incluído |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega na *checkbox* de *timestamp*<br>2. O ator carrega no botão de assinatura<br>3. O documento é assinado digitalmente com *timestamp* incluído |
| **Exceções** | 2.a) Não existe ligação à internet no *smartphone*<br>É emitida uma mensagem de aviso ao utilizador do género "Imagem escolhida inválida". |

**Anexo 8- Caso de uso 4- Enviar PDF por email**

| | |
|---|---|
| **Nome** | Enviar PDF por email |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: O utilizador quer enviar o PDF assinado por email |
| **Trigger** | O utilizador quer enviar o PDF assinado por email |
| **Pré-condição** | • Documento assinado<br>• Existe ligação à internet |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | É aberto o programa de emails que o utilizador escolher no smartphone e é-lhe anexado o PDF assinado |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega na em "Enviar por email"<br>2. O ator seleciona a aplicação de email desejada<br>3. O documento é anexado ao email dentro da aplicação selecionada<br>4. O ator envia o email usando a aplicação escolhida |
| **Exceções** | 2.a) Não existem aplicações de email no *smartphone* do utilizador<br>É emitida uma mensagem de aviso ao utilizador do género "Não existem aplicações de email, por favor instale uma". |

## Anexo 9- Caso de uso 5- Selecionar imagem de assinatura

| | |
|---|---|
| **Nome** | Selecionar imagem de assinatura |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *Summary* |
| **Interessados e interesses** | Utilizador: Selecionar imagem para a assinatura a usar |
| **Trigger** | O ator quer selecionar a imagem que representará a sua assinatura visível no documento |
| **Pré-condição** | 1. A aplicação está a correr normalmente<br>2. O cartão de cidadão está inserido |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | 1. É informado ao utilizador que ocorreu um problema, informando qual foi<br>2. Apenas a imagem de assinatura com o nome próprio do utilizador aparecerá no ecrã de pré-visualização de imagem de assinatura final |
| **Garantias de sucesso** | O ator consegue selecionar a imagem pretendida para a sua assinatura visível |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega no botão "Selecionar aspeto"<br>2. O ator poderá agora escolher uma imagem do tipo:<br>  1. Manual via *UC Imagem de assinatura manual*<br>  2. Textual via *UC Imagem de assinatura textual*<br>  3. *Watermark* via *UC Imagem de assinatura Watermark*<br>3. O ator pressiona "Feito" quando tiver escolhido a sua imagem de assinatura<br>4. O ator consegue selecionar a imagem pretendida para sua assinatura visível com sucesso |
| **Exceções** | |

**Anexo 10- Caso de uso 5.1- Imagem de assinatura manual**

| | |
|---|---|
| **Nome** | Imagem de assinatura manual |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Escolher qual a sua imagem de assinatura manual a inserir na imagem de assinatura final |
| **Trigger** | O ator quer escolher a sua imagem de assinatura visual a inserir na sua imagem de assinatura final |
| **Pré-condição** | Permissões de acesso aos ficheiros do *smartphone* concedidas |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator consegue selecionar a imagem de assinatura manual a ser inserida na imagem de assinatura final |
| **Processo: Cenário de sucesso principal** | 5. O ator carrega na *checkbox* de assinatura manual<br>6. O ator escolhe a imagem de assinatura manual que pretende<br>7. O ator escolhe com sucesso a sua imagem de assinatura manual aparecendo agora no ecrã de pré-visualização |
| **Exceções** | 2.a) A imagem escolhida pelo utilizador é inválida<br>É emitida uma mensagem de aviso ao utilizador do género "Imagem escolhida inválida". |

**Anexo 11- Caso de uso 5.2- Imagem de assinatura textual**

| | |
|---|---|
| **Nome** | Imagem de assinatura textual |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Escolher se quer inserir a imagem textual na sua imagem de assinatura final |
| **Trigger** | O ator quer inserir a imagem de assinatura textual na sua imagem de assinatura final |
| **Pré-condição** | Permissões de acesso aos ficheiros do *smartphone* concedidas |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | A imagem textual é gerada e inserida na sua imagem de assinatura final |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega na *checkbox* de assinatura textual<br>2. A imagem textual é gerada e inserida no ecrã de pré-visualização da assinatura visual final |
| **Exceções** | |

**Anexo 12- Caso de uso 5.3- Imagem de assinatura com *watermark***

| | |
|---|---|
| **Nome** | Imagem de assinatura *watermark* |
| **Ator primário** | Utilizador |
| **Scope** | *System* |
| **Nível** | *User-goal* |
| **Interessados e interesses** | Utilizador: Escolher qual a sua imagem de assinatura *watermark* a inserir na imagem de assinatura final |
| **Trigger** | O ator quer escolher a sua imagem de assinatura *watermark* a inserir na sua imagem de assinatura final |
| **Pré-condição** | Permissões de acesso aos ficheiros do *smartphone* concedidas |
| **Última revisão** | 10 de janeiro de 2018 |
| **Garantias mínimas** | É informado ao utilizador que ocorreu um problema, informando qual foi |
| **Garantias de sucesso** | O ator consegue selecionar a imagem de assinatura *watermark* a ser inserida na imagem de assinatura final |
| **Processo: Cenário de sucesso principal** | 1. O ator carrega na *checkbox* de assinatura *watermark*<br>2. O ator escolhe a imagem de assinatura *watermark* que pretende<br>3. O ator escolhe com sucesso a sua imagem de assinatura *watermark* aparecendo agora no ecrã de pré-visualização |
| **Exceções** | 2.a) A imagem escolhida pelo utilizador é inválida<br>É emitida uma mensagem de aviso ao utilizador do género "Imagem escolhida inválida". |

## Anexo 13- Classes de equivalência e valores limite do UC Selecionar documento PDF a assinar

| **Classes de Equivalência e Valores Limite de UC Selecionar documento PDF a assinar** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Definição** | **Entrada** | **C.E Válidas** | | **C.E Inválidas** | | **V.L Válidos** | **V.L Inválidos** |
| | | **ID** | **Definição de classe** | **ID** | **Definição de classe** | | |
| A1- Documento a assinar | Formato .pdf de A1 | CEV1 | PDF selecionado | CEI1 | PDF não selecionado | N/a | N/a |
| | | CEV2 | PDF válido | CEI2 | PDF inválido | N/a | N/a |

**Anexo 14- Casos de teste do UC Selecionar documento PDF a assinar (1ª ronda de testes)**

| Casos de teste do UC Selecionar documento PDF a assinar | | | | | | |
|---|---|---|---|---|---|---|
| **ID** | **Entrada** | **Observações** | **Resultados esperados** | **Resultados atuais** | V | X |
| | **A1** | | | | | |
| 1 | documento.pdf | Um documento PDF é selecionado pelo utilizador | Mensagem do tipo "Documento selecionado" | Feedback de PDF selecionado através de uma *checkbox* com um "visto" | V | |
| 2 | (nenhum documento selecionado) | Um documento PDF não é selecionado pelo utilizador | Mensagem do tipo "Documento não selecionado" | Feedback de PDF selecionado através de uma *checkbox* sem um "visto" | V | |
| 3 | (PDF inválido) | O utilizador selecionou um documento .pdf inválido | Mensagem do tipo "Documento PDF é inválido" | Aplicação encerrou abruptamente | | X |

## Anexo 15- Classes de Equivalência e Valores Limite de UC Fazer Assinatura Visível

| **Classes de Equivalência e Valores Limite de UC Fazer Assinatura Visível** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Definição** | **Entrada** | **C.E Válidas** | | **C.E Inválidas** | | **V.L Válidos** | **V.L Inválidos** |
| | | **ID** | **Definição de classe** | **ID** | **Definição de classe** | | |
| B1- Documento a assinar | PDF válido B1 | CEV1 | PDF com campos de assinatura | CEI1 | PDF não selecionado | N/a | N/a |
| | | CEV2 | PDF com sem campos de assinatura | | | | |
| B2- Imagem de assinatura | Imagem válida B2 | CEV3 | Imagem selecionada | CEI2 | Imagem não selecionada | | |
| B3- *Timestamp* | Opção B3 | CEV4 | *Timestamp* selecionado | N/a | N/a | | |
| | | CEV5 | *Timestamp* não selecionado | | | | |
| B4- Servidor de *timestamp* | Estado de B4 | CEV6 | Operacional | | | | |
| | | CEV7 | Em baixo | | | | |
| B5- Cartão de cidadão | Estado de B5 | CEV8 | Inserido | CEI3 | Removido | | |
| B6- PIN do CC | Valor inteiro não negativo de quatro dígitos B6 | CEV9 | Válido | CEI4 | Inválido | | |

## Anexo 16- Casos de teste do UC Fazer Assinatura Visível (1ª ronda de testes)

| Casos de teste do UC Fazer Assinatura Visível | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **ID** | **Entrada** | | | | | | **Observações** | **Resultados esperados** | **Resultados atuais** | V | X |
| | **B1** | **B2** | **B3** | **B4** | **B5** | **B6** | | | | | |
| 1 | (PDF com campos de assinatura selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados de entrada estão escolhidos devidamente (PDF com campos) | Mensagem do tipo "Documento assinado com sucesso" | Verifica-se | V | |
| | | | | | | | | Documento assinado com *timestamp* e imagem inserida no campo de assinatura | Verifica-se | V | |
| 2 | (PDF sem campos de assinatura selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados de entrada estão escolhidos (PDF sem campos) | Mensagem do tipo "Documento assinado com sucesso" | Verifica-se | V | |
| | | | | | | | | Documento assinado com *timestamp* e imagem inserida no canto inferior direito da ultima página | Verifica-se | V | |
| 3 | (PDF não selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados escolhidos exceto o PDF | Mensagem do tipo "Falta selecionar PDF" | Verifica-se | V | |
| 4 | (PDF selecionado) | (Imagem não selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados escolhidos exceto a imagem de assinatura | Mensagem do tipo "Falta selecionar imagem" | Verifica-se | V | |
| 5 | (PDF não selecionado) | (Imagem não selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | PDF e imagem não escolhidos | Mensagem do tipo "Falta selecionar imagem e PDF" | Verifica-se | V | |
| 6 | (PDF selecionado) | (Imagem selecionada) | (*Timestamp* não selecionado) | Operacional | Inserido | Válido | *Timestamp* não selecionado | Mensagem do tipo "Documento assinado com sucesso" | Verifica-se | V | |
| | | | | | | | | Documento assinado sem *timestamp* | Verifica-se | V | |
| 7 | (PDF selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Em baixo | Inserido | Válido | Servidor de *timestamp* esta em baixo e o utilizador queria fazer uma assinatura com *timestamp* | Mensagem do tipo "Servidor de *timestamp* em baixo, vai ser usado o horário do dispositivo" | Verifica-se | V | |
| 8 | (PDF selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Removido | Válido | Todos os dados escolhidos exceto o cartão que se encontra removido | Não permite fazer assinatura | Verifica-se | V | |
| 9 | (PDF com campo já assinado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Campo a assinar do PDF já se encontra assinado | Mensagem de aviso do tipo: "Campo a assinar já se encontra assinado" e não assina o PDF | Verifica-se | V | |
| 10 | (PDF selecionado) | (Imagem selecionada) | (*Timestamp* selecionado) | Operacional | Inserido | Inválido | PIN inválido | Mensagem de aviso do tipo:" PIN invalido e tentativas restantes" | Verifica-se | V | |

## Anexo 17- Classes de Equivalência e Valores Limite de UC Fazer Assinatura Invisível

| **Classes de Equivalência e Valores Limite de UC Fazer Assinatura Invisível** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Definição** | **Entrada** | **C.E Válidas** | | **C.E Inválidas** | | **V.L Válidos** | **V.L Inválidos** |
| | | **ID** | **Definição de classe** | **ID** | **Definição de classe** | | |
| C1- Documento a assinar | PDF válido C1 | CEV1 | PDF com campos de assinatura | CEI1 | PDF não selecionado | N/a | N/a |
| | | CEV2 | PDF com sem campos de assinatura | | | | |
| C2- *Timestamp* | Opção C2 | CEV3 | *Timestamp* selecionado | N/a | N/a | | |
| | | CEV4 | *Timestamp* não selecionado | | | | |
| C3- Servidor de *timestamp* | Estado de C3 | CEV5 | Operacional | | | | |
| | | CEV6 | Em baixo | | | | |
| C4- Cartão de cidadão | Estado de C4 | CEV7 | Inserido | CEI2 | Removido | | |
| C5- PIN do CC | Valor inteiro não negativo de quatro dígitos B6 | CEV8 | Válido | CEI3 | Inválido | | |

## Anexo 18- Casos de teste do UC Assinatura Invisível (1ª ronda de testes)

| Casos de teste do UC Assinatura Invisível | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **ID** | **Entrada** | | | | | **Observações** | **Resultados esperados** | **Resultados atuais** | V | X |
| | **C1** | **C2** | **C3** | **C4** | **C5** | | | | | |
| 1 | (PDF com campos de assinatura selecionado) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados de entrada estão escolhidos (PDF com campos) | Mensagem do tipo "Documento assinado com sucesso" | Verifica-se | V | |
| | | | | | | | Documento assinado com *timestamp* | Verifica-se | V | |
| 2 | (PDF sem campos de assinatura selecionado) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados de entrada estão escolhidos (PDF sem campos) | Mensagem do tipo "Documento assinado com sucesso" | Verifica-se | V | |
| | | | | | | | Documento assinado com *timestamp* | Verifica-se | V | |
| 3 | (PDF não selecionado) | (*Timestamp* selecionado) | Operacional | Inserido | Válido | Todos os dados escolhidos exceto o PDF | Mensagem do tipo "Falta selecionar PDF" | Verifica-se | V | |
| 4 | (PDF selecionado) | (*Timestamp* não selecionado) | Operacional | Inserido | Válido | *Timestamp* não selecionado | Mensagem do tipo "Documento assinado com sucesso" Documento assinado sem *timestamp* | Verifica-se | V | |
| 5 | (PDF selecionado) | (*Timestamp* selecionado) | Em baixo | Inserido | Válido | Servidor de *timestamp* esta em baixo e o utilizador queria fazer uma assinatura com *timestamp* | Mensagem do tipo "Servidor de *timestamp* em baixo, vai ser usado o horário do dispositivo" | Verifica-se | V | |
| 6 | (PDF selecionado) | (*Timestamp* selecionado) | Operacional | Removido | | Todos os dados escolhidos exceto o cartão que se encontra removido | Não permite fazer assinatura | Verifica-se | V | |
| 7 | (PDF selecionado) | (*Timestamp* selecionado) | Operacional | Inserido | Inválido | Inválido | Mensagem de aviso do tipo:" PIN invalido e tentativas restantes" | Verifica-se | V | |

## Anexo 19- Classes de Equivalência e Valores Limite do UC Incluir Timestamp

| **Classes de Equivalência e Valores Limite do UC Incluir *Timestamp*** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Definição** | **Entrada** | **C.E Válidas** | | **C.E Inválidas** | | **V.L Válidos** | **V.L Inválidos** |
| | | **ID** | **Definição de classe** | **ID** | **Definição de classe** | | |
| D1- *Timestamp* | Opção D1 | CEV1 | *Timestamp* selecionado | N/a | N/a | N/a | N/a |
| | | CEV2 | *Timestamp* não selecionado | | | | |
| D2- Acesso à internet | Evento D2 | CEV3 | Existe acesso à internet | CEI1 | Não existe acesso à internet | | |

## Anexo 20- Casos de teste do UC Incluir Timestamp (1ª ronda de testes)

| Casos de teste do UC Incluir *Timestamp* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **ID** | **Entrada** | | **Observações** | **Resultados esperados** | **Resultados atuais** | V | X |
| | **D1** | **D2** | | | | | |
| 1 | (*Timestamp* selecionado) | Device tem acesso à internet | Utilizador quer usar *timestamp* e tem acesso à internet | É permitido assinar | Verifica-se | V | |
| 2 | (*Timestamp* não selecionado) | *Device* tem acesso à internet | Utilizador não quer usar *timestamp* e tem acesso à internet | É permitido assinar | Verifica-se | V | |
| 3 | (*Timestamp* selecionado) | *Device* não tem acesso à internet | Utilizador quer usar *timestamp*, mas não está ligado à internet | Mensagem do tipo "É necessária ligação à internet para usar *timestamp* na assinatura" | Verifica-se | V | |
| 4 | (*Timestamp* selecionado) | *Device* não tem acesso à internet | Utilizador não quer usar *timestamp* e não está ligado à internet | É permitido assinar | Verifica-se | V | |

## Anexo 21- Classes de Equivalência e Valores Limite do UC Selecionar Imagem de Assinatura

| **Classes de Equivalência e Valores Limite de UC Selecionar Imagem de Assinatura** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Definição** | **Entrada** | **C.E Válidas** | | **C.E Inválidas** | | **V.L Válidos** | **V.L Inválidos** |
| | | **ID** | **Definição de classe** | **ID** | **Definição de classe** | | |
| E1- Imagem de *watermark* | E1 no formato .png | CEV1 | Imagem *watermark* selecionada | CEI1 | Imagem *watermark* inválida | N/a | N/a |
| | | CEV2 | Imagem *watermark* não selecionada | N/a | N/a | | |
| E2- Imagem de assinatura manuscrita | E2 no formato .png | CEV3 | Imagem assinatura manuscrita selecionada | CEI2 | Imagem assinatura manuscrita invalida | | |
| | | CEV4 | Imagem assinatura manuscrita não selecionada | N/a | N/a | | |
| E3- Imagem textual | Gerar E3 | CEV5 | Imagem textual gerada | | | | |
| | | CEV6 | Imagem textual não gerada | | | | |
| E4- Cartão de cidadão | Estado de E4 | CEV7 | Inserido | CEI3 | Removido | | |

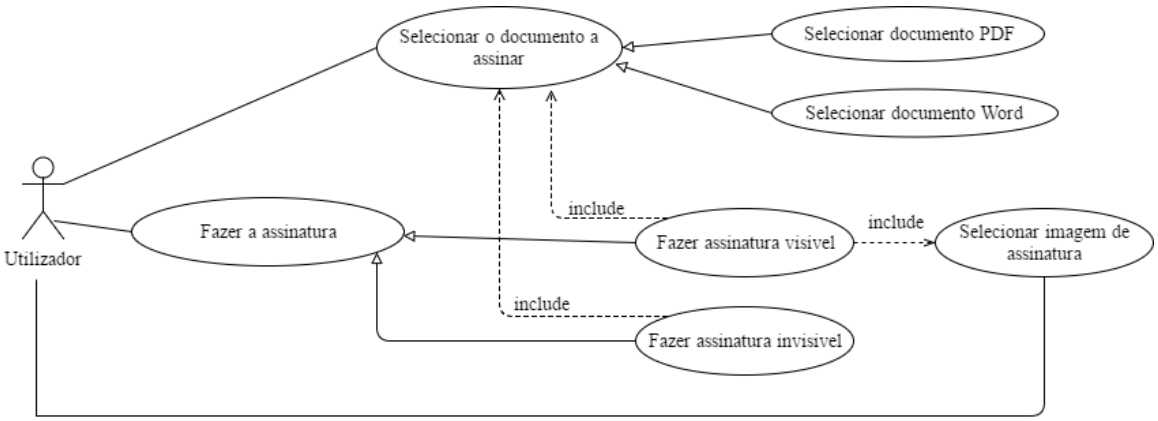
**Anexo 22- Casos de teste do UC Selecionar Imagem de assinatura (1ª ronda de testes)**

| Casos de teste do UC Selecionar Imagem de assinatura | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **ID** | **Entrada** | | | | **Observações** | **Resultados esperados** | **Resultados atuais** | V | X |
| | **E1** | **E2** | **E3** | **E4** | | | | | |
| 1 | (Imagem *watermark* selecionada) | (Imagem de assinatura manuscrita selecionada) | (Imagem textual gerada) | Inserido | Tudo se encontra selecionado e o cartão inserido | É gerada uma imagem de assinatura onde no lado esquerdo encontra-se a *watermark* e a imagem manuscrita, do lado direito encontra-se o lado direito da imagem textual gerada | Verifica-se | V | |
| 2 | (Imagem *watermark* não selecionada) | (Imagem de assinatura manuscrita não selecionada) | (Imagem textual não gerada) | Inserido | Nada esta selecionado/gerado, mas o cartão está inserido | É gerada uma imagem de assinatura *default* com o nome completo do proprietário do cartão inserido | Verifica-se | V | |
| 3 | (Imagem *watermark* não selecionada) | (Imagem de assinatura manuscrita não selecionada) | (Imagem textual não gerada) | Removido | Nada esta selecionado/gerado e o cartão está removido | Mensagem do tipo "Insira o cartão para gerar imagem" irá aparecer | Verifica-se | V | |
| 4 | (Imagem *watermark* selecionada) | (Imagem de assinatura manuscrita selecionada) | (Imagem textual gerada) | Removido | Tudo se encontra selecionado, mas o cartão está removido | É permitido criar a imagem (neste momento já esta tudo em memoria e já não precisa do cartão) | Verifica-se | V | |
| 5 | (Imagem *watermark* selecionada) | (Imagem de assinatura manuscrita selecionada) | (Imagem textual não gerada) | Inserido | *Watermark* e imagem de assinatura manual encontram-se inseridas, mas imagem textual não | É gerada uma imagem de assinatura com a *watermark* e a imagem manuscrita | Verifica-se | V | |
| 6 | (Imagem *watermark* selecionada) | (Imagem de assinatura manuscrita | (Imagem textual não gerada) | Inserido | Só a imagem *watermark* se encontra inserida | É gerada uma imagem de assinatura com a *watermark* e o nome próprio do assinante | Verifica-se | V | |

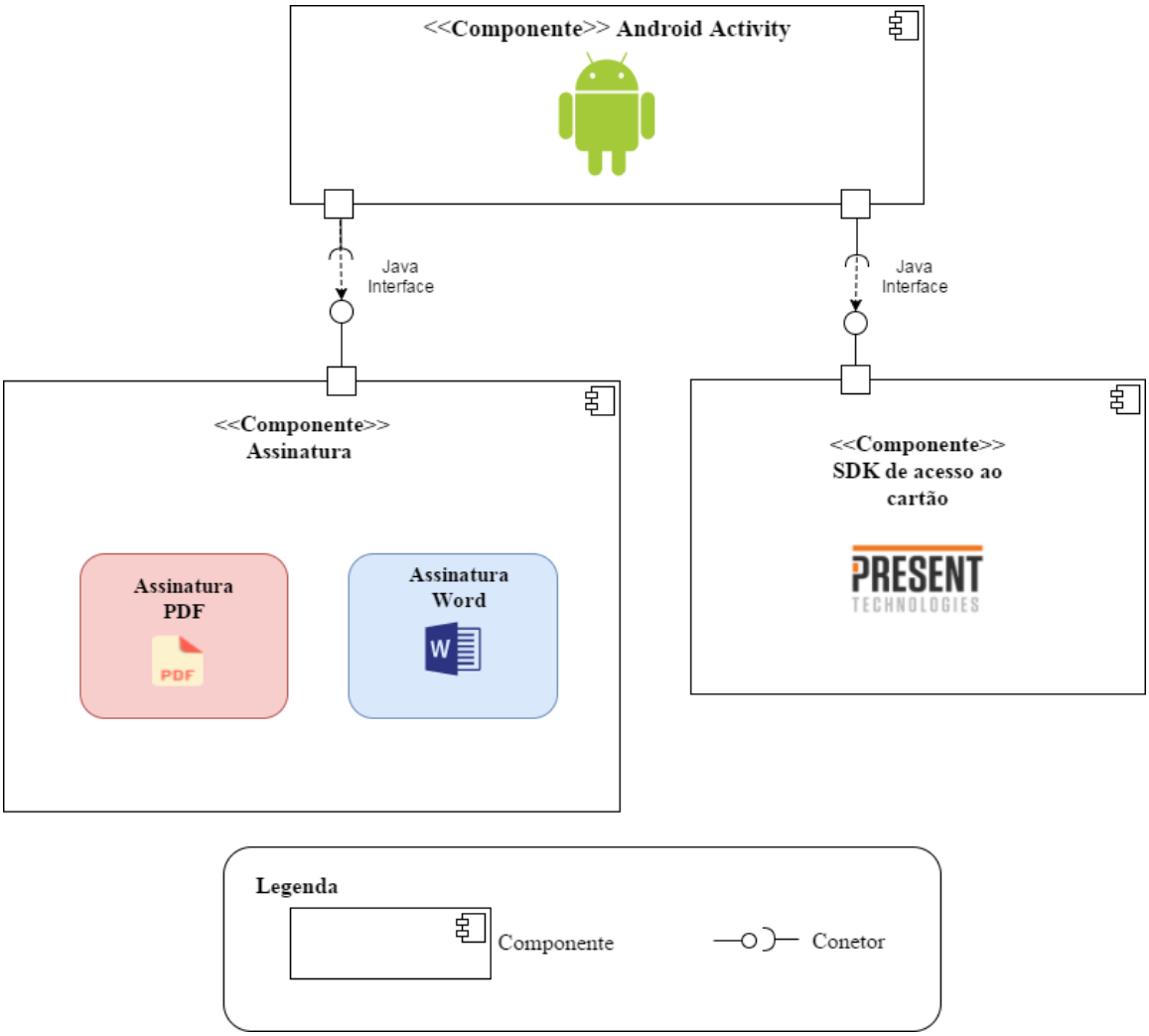
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | não selecionada) | | | | | | | |
| 7 | (Imagem *watermark* selecionada) | (Imagem de assinatura manuscrita não selecionada) | (Imagem textual gerada) | Inserido | Imagem *watermark* inserida e imagem textual gerada | É gerada uma imagem de assinatura com a *watermark* do lado esquerdo juntamente com o primeiro e ultimo nome do assinante. Do lado direito encontra-se o lado direito da imagem textual gerada | Verifica-se | V | |
| 8 | (Imagem *watermark* não selecionada) | (Imagem de assinatura manuscrita selecionada) | (Imagem textual gerada) | Inserido | Imagem textual gerada e a imagem manuscrita inserida | É gerada uma imagem de assinatura onde no lado esquerdo encontra-se a imagem manuscrita e no lado direito encontra-se o lado direito da imagem de assinatura textual | Verifica-se | V | |
| 9 | (Imagem *watermark* não selecionada) | (Imagem de assinatura manuscrita selecionada) | (Imagem textual não gerada) | Inserido | Só a imagem de assinatura manuscrita se encontra inserida | É gerada uma imagem de assinatura só com a imagem manuscrita | Verifica-se | V | |
| 10 | (Imagem *watermark* não selecionada) | (Imagem de assinatura manuscrita não selecionada) | (Imagem textual gerada) | Inserido | Só a imagem textual foi gerada, as restantes não. | É gerada uma imagem somente com a imagem de assinatura textual | Verifica-se | V | |
| 11 | (Imagem de *watermark* invalida) | (Imagem de assinatura manuscrita não selecionada) | (Imagem textual não gerada) | Inserido | Imagem de *watermark* invalida (.png invalido) | Mensagem do tipo "Imagem *watermark* inválida" irá aparecer | A aplicação termina abruptamente | | X |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 12 | (Imagem *watermark* não selecionada) | (Imagem assinatura manuscrita invalida) | (Imagem textual não gerada) | Inserido | Imagem assinatura manuscrita invalida (.png vazio invalido) | Mensagem do tipo "Imagem manuscrita inválida" irá aparecer | A aplicação termina abruptamente | | X |

**Anexo 23– Diagrama de casos de uso inicial (antes da alteração de requisitos)**



**Anexo 24- Vista antiga de componente e conetor**



## Anexo 25- Vista antiga de modulos