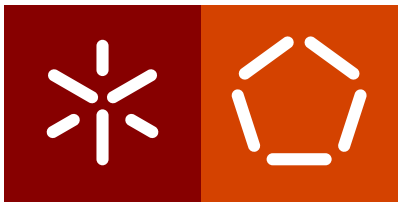
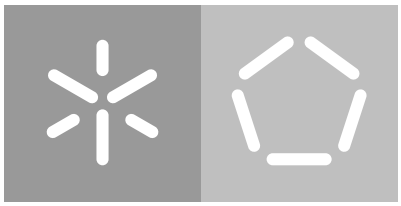


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nuno Cabral Vieira

Conveniência e Segurança com SCMD

Novembro 2019



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nuno Cabral Vieira

Conveniência e Segurança com SCMD

Dissertação de Mestrado
Mestrado em Engenharia Informática

Dissertação supervisionada por
José Bacelar Almeida

Novembro 2019

Despacho RT - 31 /2019 - Anexo 3

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho

Atribuição
CC BY

https://creativecommons.org/licenses/by/4.0/

AGRADECIMENTOS

Esta dissertação de mestrado é o resultado de muitas horas de trabalho e é importante exprimir os meus sinceros agradecimentos às pessoas que me incentivaram, apoiaram e ajudaram em mais uma etapa da minha vida.

Em primeiro lugar, quero agradecer ao professor Doutor José Carlos Bacelar Almeida, pela orientação prestada, pelo seu incentivo, disponibilidade e apoio que sempre demonstrou.

Desejo igualmente agradecer ao José Eduardo Pina Miranda, por não só me ter sugerido este projeto no âmbito de um estágio profissional na empresa Devise Futures, mas também pela sua disponibilidade, apoio e sabedoria que foram um pilar essencial para que este trabalho fosse possível.

Quero agradecer a todos os meus amigos e colegas que de uma forma directa ou indirecta, contribuíram, ou auxiliaram na elaboração da presente dissertação, pela paciência, atenção e força que prestaram em momentos menos fáceis.

Não quero deixar de agradecer à minha família, especialmente aos meus pais, José Vieira e Augusta Vieira, aos meus irmãos, Marta Vieira e Miguel Vieira, à minha avó, Gracinda Cabral, e à minha namorada, Maria Silva, pelo apoio incondicional e motivação que me deram em todo o momento, principalmente nos momentos mais difíceis, e pela paciência e compreensão que demonstraram, pois sem eles não teria sido possível chegar ao fim desta etapa.

Por fim, o meu profundo e sentido agradecimento a todas as pessoas que contribuíram para a concretização desta dissertação, assim terminando esta etapa da minha vida.

Despacho RT - 31 /2019 - Anexo 4

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Em Portugal existe o Serviço Chave Móvel Digital (SCMD), que permite a qualquer cidadão efetuar a assinatura eletrónica qualificada remota de dados. Atualmente é disponibilizada, publicamente a todos os cidadãos, a aplicação Autenticação.gov, que oferece um conjunto de funcionalidades, sendo uma delas a assinatura de documentos PDF utilizando o SCMD. Mas esta é limitada às plataformas Windows, Linux e Mac. Desse modo, esta dissertação de mestrado tem como objectivo desenvolver uma aplicação que assine documentos PDF, com o SCMD, em Android e iOS.

Para desenvolver a aplicação, é utilizada a *framework* de desenvolvimento de aplicações móveis Android e iOS, React Native e a API nativa em conjunto com a ferramenta de geração, programação e manipulação de documentos PDF, IText 7, para desenvolver as operações de assinatura e a comunicação com o SCMD, resultando numa aplicação móvel, que apesar das limitações enfrentadas, assina documentos PDF, com o SCMD.

Com a aplicação desenvolvida, é possível assinar documentos PDF com o SCMD, nos quais as assinaturas são claramente visíveis, válidas e estão em conformidade com a especificação técnica dos *standards* de assinaturas eletrónicas avançadas de PDFs. Tendo isto em consideração, a aplicação desenvolvida oferece assinaturas mais robustas, é mais rápida a concluir todo o processo e produz assinaturas de menor tamanho em comparação com a aplicação existente.

Palavras-chave: Serviço Chave Móvel Digital, Assinatura Eletrónica Qualificada, PDF, Android, iOS.

ABSTRACT

The Serviço Chave Móvel Digital (SCMD), a service available in Portugal, allows for any citizen to sign data remotely with a qualified eletronic signature. Currently, the Autenticação.gov application is publicly available to every citizen, that offers many features, one of these being the signing of a PDF document with the SCMD. But this application is limited to Windows, Linux and Mac. As such, the main objective of this dissertation is to develop a mobile application for Android and iOS, that signs PDF documents using the SCMD.

Both the React Native framework and the Native API are used to develop the application and the PDF Toolkit for PDF generation, programming, handling and manipulation IText 7 is used to develop the signature process and the process of communication with the SCMD, resulting in a mobile application, that despite the limitations faced, signs PDF documents with the SCMD.

With the developed application, it is possible to sign PDF documents with the SCMD, in which the signature are visible, valid and are in accordance with the technical specifications of the PDF Advanced Electronic Signatures standards. Having this in consideration, the developed application offers more robust signatures, signs faster and generates smaller signatures compared to the application currently available.

Keywords: Serviço Chave Móvel Digital, Qualified Eletronic Signature, PDF, Android, iOS.

CONTEÚDO

1 INTRODUÇÃO 1
1.1 Contexto 1
1.2 Objetivos 2
1.3 Estrutura do Documento 2
2 ESTADO DE ARTE 3
2.1 Assinatura Digital Qualificada 3
2.2 OCSP 5
2.2.1 Pedido OCSP 6
2.2.2 Resposta OCSP 6
2.3 CRL 7
2.4 Timestamp 8
2.4.1 Funcionamento 9
2.5 PAdES 9
2.5.1 PAdES Básico 10
2.5.2 PAdES-Baseline-B 11
2.5.3 PAdES-Baseline-T 12
2.5.4 PAdES-Baseline-LT 12
2.5.5 PAdES-Baseline-LTA 13
2.6 SCMD 13
2.7 Trabalhos Relacionados 14
2.7.1 Aplicação Autenticação.gov 14
2.8 Sumário 16
3 PROBLEMA E DESAFIOS 17
3.1 Problema e Desafios 17
3.2 Funcionalidades 18
3.3 Solução 23
3.3.1 Apresentação 23
3.3.2 Lógica de Negócio 24
3.3.3 Dados e Configurações 27
3.3.4 Serviços Externos 27
4 DESENVOLVIMENTO 29
4.1 Tecnologias e Decisões 29
4.1.1 Aplicação Móvel 29

4.1.2 Assinatura e CMD 31
4.2 Implementação 32
4.2.1 Interface 33
4.2.2 Módulos React Native 40
4.2.3 Comunicação com CMD 41
4.2.4 Assinatura 44
4.3 Resultados 50
4.4 Sumário 51
5 TESTES 53
5.1 Setup dos Testes 53
5.1.1 Demonstração 53
5.1.2 Testes 54
5.2 Resultados 55
5.2.1 Demonstração 55
5.2.2 Testes 65
5.3 Discussão 68
5.4 Sumário 70
6 CONCLUSÃO 71
6.1 Trabalhos Futuros 73

LISTA DE FIGURAS

Figura 1	Perfil PAdES-Básico (Fonte: ETSI TS 102 778-1 V1.1.1 [30])	10
Figura 2	Perfil PAdES-Baseline-B e T (Fonte: ETSI TS 102 778-1 V1.1.1 [30])	11
Figura 3	Perfil PAdES-Baseline-LT e LTA (Fonte: ETSI TS 102 778-1 V1.1.1 [30])	12
Figura 4	Pedidos do serviço SCMD segundo a sua especificação[8]	14
Figura 5	Arquitetura da aplicação	23
Figura 6	Componentes da Lógica de Assinatura	25
Figura 7	Interesse ao longo do tempo das *frameworks* de desenvolvimento (fonte: Google Trends)	30
Figura 8	Arquitetura de desenvolvimento utilizando React Native.	32
Figura 9	Ecrã Inicial	34
Figura 10	Menu de escolha do perfil de assinatura	35
Figura 11	Ecrã de Visualização do Documento a Assinar	36
Figura 12	Ecrã de Confirmação da Localização da Assinatura no Documento	37
Figura 13	Ecrã de Autenticação com o SCMD	38
Figura 14	Menu de escolha das Razões de Assinatura	39
Figura 15	Ecrã Final, de conclusão do processo de assinatura	40
Figura 16	Interface do módulo React Native Document Picker	41
Figura 17	Pedido de Certificados do Assinante ao Serviço CMD	42
Figura 18	Pedido de Inicio de Processo de Assinatura ao Serviço CMD	43
Figura 19	Pedido de Validação de OTP ao Serviço CMD	44
Figura 20	Objeto Commitment Type Indication especificado no RFC 5126	45
Figura 21	Objeto ESSCertIDv2 especificado no RFC 5035	46
Figura 22	Inicio do processo de assinatura	46
Figura 23	Assinatura PAdES-B-B ou PAdES-B-T	47
Figura 24	Assinatura PAdES-B-LT	48
Figura 25	Assinatura PAdES-B-LTA	49
Figura 26	Assinatura PAdES-Baseline-B visializada no Adobe Reader, a sua validação e propriedades	55
Figura 27	Painel de Assinaturas do documento assinado com assinatura PAdES-B-B	56
Figura 28	Validação da assinatura PAdE-Baseline-B na *WebApp* do DSSValidação da assinatura PAdES-Baseline-B na *WebApp* do DSS[1]	57

Figura 29 Assinatura PAdES-Baseline-T visializada no Adobe Reader, a sua validação e propriedades 58
Figura 30 Painel de Assinaturas do documento assinado com assinatura PAdES-B-T 59
Figura 31 Validação da assinatura PAdES-Baseline-T na *WebApp* do DSS 59
Figura 32 Assinatura PAdES-Baseline-LT visializada no Adobe Reader, a sua validação e propriedades 60
Figura 33 Painel de Assinaturas do documento assinado com assinatura PAdES-B-LT 61
Figura 34 Validação da assinatura PAdES-Baseline-LT na *WebApp* do DSS 61
Figura 35 Assinatura PAdES-Baseline-LTA visializada no Adobe Reader, a sua validação e propriedades 62
Figura 36 Painel de Assinaturas do documento assinado com assinatura PAdES-B-B 63
Figura 37 Validação da assinatura PAdES-Baseline-LTA na *WebApp* do DSS 64
Figura 38 10 primeiros resultados dos testes da plataforma "ETSI Signature Conformance Checker" 64
Figura 39 Único erro resultante dos testes da plataforma "ETSI Signature Conformance Checker" 65

LISTA DE ACRÓNIMOS

AMA Agência para a Modernização Administrativa, I.P..
API Application Programming Interface.

CA Autoridade de Certificação.
CAdES CMS Advanced Electronic Signatures.
CMD Chave Móvel Digital.
CRL Certificate Revogation List.

DSS Digital Signature Services.

eIDAS electronic IDentification, Authentication and trust Services.
ETSI European Telecommunications Standards Institute.
EUTL European Union Trusted Lists.

HSM Hardware Security Module.
HTTPS Hypertext Transfer Protocol Secure.

OCSP Online Certificate Status Protocol.
OID Identificador de Objeto.
OTP One-time password.

PAdES PDF Advanced Electronic Signatures.
PDF Portable Document Format.
PKCS Public Key Cryptography Standards.

RFC Request For Comments.

SCMD Serviço Chave Móvel Digital.
SMS Short Message Service.
SOAP Simpe Object Access Protocol.

TLS Transport Layer Security.
TSA Time Stamping Authority.

UE União Europeia.
URL Uniform Resource Locator.

WYSIWYS What You See Is What You Sign.

1

INTRODUÇÃO

1.1 CONTEXTO

Há já mais de dez anos que foi emitido o primeiro Cartão de Cidadão, que continha um certificado eletrónico para assinatura digital qualificada, com valor legal de acordo com o Decreto-Lei nº. 290-D/99, de 2 de Agosto [15] (republicado pelo Decreto-Lei nº. 88/2009, de 9 de Abril [18]) e com a Directiva 1999/93/CE do Parlamento Europeu e do Conselho, de 13 de Dezembro de 1999 [16], entretanto substituída pelo Regulamento UE nº. 910/2014 do Parlamento Europeu e do Conselho de 23 de julho de 2014 [20] (regulamento electronic IDentification, Authentication and trust Services (eIDAS)). Esse facto teve um papel fundamental na desmaterialização de documentos, embora tal só tenha sido mais visível nos processos de compras públicas (na sequência do Decreto-Lei n.º 18/2008, de 29 de Janeiro do Código dos Contratos Públicos [17]), já que a assinatura com o Cartão de Cidadão tem o problema da falta de conveniência: obriga à utilização de um leitor de cartões e à instalação de drivers e outro software que não são operações simples para grande parte do público em geral.

Portugal, por intermédio da Agência para a Modernização Administrativa, I.P. (AMA), tem desenvolvido um conjunto de projetos inovadores na área da desmaterialização de documentos e desburocratização de serviços, que se encontram na vanguarda do que é feito a nível europeu e mundial. Um desses projetos, o Serviço Chave Móvel Digital (SCMD), está credenciado de acordo com o regulamento UE 910/2014 [20] (regulamento eIDAS) e de acordo com o Despacho 155/2017 da Entidade Supervisora nacional, de 5 de dezembro [21], permite a assinatura eletrónica qualificada remota, permitindo a qualquer cidadão efetuar a assinatura eletrónica de dados com uma chave privada que se encontra arquivada remotamente e, através de um dispositivo qualificado de assinatura remoto;

O SCMD ao não obrigar à utilização de hardware adicional ou à instalação de drivers, aliam a conveniência à segurança na assinatura eletrónica de documentos.

1.2 OBJETIVOS

Com esta dissertação de tese de mestrado pretende-se:

- Estudar e analisar as operações do SCMD;
- Estudar e analisar a criação de assinaturas eletrónicas qualificadas;
- Estado de arte do desenvolvimento de aplicações móveis Android e iOS;
- Desenvolver uma aplicação móvel (iOS e Android) que assine documentos PDF com o SCMD;
- Idealizar e incluir modelo de adição de anúncios na aplicação móvel referida no ponto anterior.

1.3 ESTRUTURA DO DOCUMENTO

No Capítulo 1, é descrito o contexto e os objetivos desta dissertação e a estrutura deste documento.

No Capítulo 2, são explicadas e descritas as principais partes envolvidas nesta dissertação, desde protocolos, esquemas, serviços e regulamentações.

No Capítulo 3, são apresentados os problemas atuais da assinatura digital qualificada utilizando o SCMD, os desafios enfrentados para tentar resolver os problemas, as funcionalidades necessárias para a aplicação a desenvolver e a solução para os problemas e desafios que implemente as funcionalidades necessárias.

No Capítulo 4, é descrito o processo de decisão durante o desenvolvimento da aplicação, como também a implementação deste e os resultados obtidos.

No Capítulo 5, são demonstradas as operações de assinatura utilizando os perfis PAdES-Baseline-B, T, LT e LTA, como também testes e medições de tempo de finalização destas operações, assim comparando com a aplicação já existente, Autenticação.gov.

No Capítulo 6, é feita uma uma conclusão a esta dissertação, sendo realçado o valor do trabalho realizado. Por fim, é descrito os possíveis trabalhos futuros.

2

ESTADO DE ARTE

Neste capítulo serão explicadas e descritas as partes que tornam possível a existência de serviços como o Serviço Chave Móvel Digital (SCMD), como também esquemas e protocolos que serão utilizados em conjunto com estes serviços no desenvolvimento desta dissertação.

2.1 ASSINATURA DIGITAL QUALIFICADA

O regulamento electronic IDentification, Authentication and trust Services (eIDAS) [20] é a regulamentação europeia relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno europeu. Publicada em 2014, surgiu com o objetivo de facilitar transações eletrónicas seguras na União Europeia, fornecendo um ambiente regulamentar que promova o seu uso. Com isto, os cidadãos e empresas da união europeia podem utilizar esquemas de identificação eletrónica para aceder a serviços públicos, como a Chave Móvel Digital em Portugal. Adicionalmente, esta regulamentação define serviços de confiança qualificados, tais como as assinaturas eletrónicas, selos temporais, selos eletrónicos e certificados qualificados que conferem valor legal às transações eletrónicas.

Segundo o regulamento eIDAS, uma assinatura eletrónica diz-se **avançada** se respeita os seguintes requisitos:

- Estar associada de modo único ao signatário;
- Permitir identificar o signatário;
- Ser criada utilizando dados para a criação de uma assinatura eletrónica que o signatário pode, com um elevado nível de confiança, utilizar sob o seu controlo exclusivo;
- Estar ligada aos dados por ela assinados de tal modo que seja detetável qualquer alteração posterior dos dados.

Uma assinatura eletrónica é qualificada quando uma assinatura eletrónica avançada é criada por um dispositivo qualificado de criação de assinaturas eletrónicas e se baseia num certificado qualificado de assinatura eletrónica.

O certificado qualificado de assinatura eletrónica é um certificado emitido por um prestador de serviços de confiança, que satisfaça os requisitos estabelecidos no anexo I do regulamento eIDAS [20], tendos estes que conter:

1. Uma indicação, pelo menos num formato adequado ao tratamento automático, de que o certificado foi emitido como certificado qualificado de assinatura eletrónica;
2. Um conjunto de dados que representem inequivocamente o prestador qualificado de serviços de confiança que tiver emitido os certificados qualificados, incluindo, pelo menos, o Estado-Membro em que esse prestador se encontre estabelecido e
   - para as pessoas coletivas: a designação e, eventualmente, o número de registo conforme constam dos registos oficiais;
   - para as pessoas singulares: o nome;
3. Pelo menos, o nome do signatário, ou um pseudónimo, caso seja utilizado um pseudónimo, este deve ser claramente indicado;
4. Os dados necessários para a validação da assinatura eletrónica que correspondam aos dados necessários para a criação da assinatura eletrónica;
5. A indicação do início e do termo da validade do certificado;
6. O código de identidade do certificado, que deve estar associado de modo único ao prestador qualificado de serviços de confiança;
7. A assinatura eletrónica avançada ou o selo eletrónico avançado do prestador qualificado de serviços de confiança emitente;
8. O local em que está disponível, a título gratuito, o certificado que sustenta a assinatura eletrónica avançada ou o selo eletrónico avançado a que se refere a alínea 7);
9. A localização dos serviços aos quais se pode recorrer para inquirir da validade do certificado qualificado;
10. Se os dados para a criação da assinatura eletrónica relacionados com os dados para a validação da assinatura eletrónica se encontrarem num dispositivo qualificado de criação de assinatura eletrónica, uma indicação adequada desse facto, pelo menos num formato adequado para tratamento automático.

O Dispositivo Qualificado de Criação de Assinaturas Eletrónicas é um software ou hardware configurado, utilizando para criar assinaturas eletrónicas, que satisfaça os requisitos estabelecidos no anexo II do regulamento eIDAS [20]:

1. Os dispositivos qualificados de criação de assinaturas eletrónicas asseguram, pelos meios técnicos e procedimentais adequados, que pelo menos:
   a) A confidencialidade dos dados necessários para a criação de assinaturas eletrónicas utilizados para criar as assinaturas eletrónicas esteja razoavelmente assegurada;
   b) Os dados necessários para a criação de assinaturas eletrónicas utilizados para criar assinaturas eletrónicas só passam, na prática, ocorrer uma vez;
   c) Os dados necessários para a criação de assinaturas eletrónicas utilizados para criar as assinaturas eletrónicas não possam, com uma segurança razoável, ser deduzidos de outros dados e que as assinaturas estejam protegidas eficazmente contra falsificações produzidas por meio de tecnologias atualmente disponíveis;
   d) Os dados necessários para a criação de assinaturas eletrónicas não podem alterar os dados a assinar nem impedir que esses dados sejam apresentados ao signatário antes da assinatura.
2. Os dispositivos qualificados de criação de assinaturas eletrónicas não podem alterar os dados a assinar nem impedir que esses dados sejam apresentados ao signatário antes da assinatura;
3. A geração ou a gestão, em nome do signatário, dos dados necessários para a criação de assinaturas eletrónicas só podem ser efetuadas por um prestador qualificado de serviços de confiança;
4. Sem prejuízo do ponto 1, alínea d), os prestadores qualificados de serviços de confiança que gerem os dados necessários para a criação de assinaturas eletrónicas em nome do signatário podem duplicar esses dados apenas para fins de cópia de segurança, desde que sejam cumpridos os seguintes requisitos:
   a) A segurança dos conjuntos de dados duplicados estar ao mesmo nível da dos conjuntos de dados originais;
   b) O número de conjuntos de dados duplicados não exceder o mínimo necessário para garantir a continuidade do serviço.

2.2 OCSP

O Online Certificate Status Protocol (OCSP) [33] permite às aplicações verificar o estado atual de um certificado sem recorrer à Certificate Revogation List (CRL). Um cliente OCSP,

presente nas aplicações ou como serviço externo, faz um pedido de estado do certificado ao servidor OCSP, e obtém o estado do certificado como resposta.

2.2.1 *Pedido OCSP*

Um pedido OCSP contém a seguinte informação:

- Versão do protocolo;
- Solicitação do serviço OCSP;
- Identificação do certificado em questão;
- Extensões opcionais que poderão ser processadas pelo servidor OCSP.

No caso do pedido estar mal formulado e/ou o servidor não estiver preparado para responder ao pedido, é devolvida uma mensagem de erro.

2.2.2 *Resposta OCSP*

Uma resposta OCSP consiste no tipo da resposta e nos bytes da resposta. Todos os clientes e servidores OCSP têm que suportar o tipo básico de respostas, que vai ser descrito a seguir.

Uma resposta OCSP é composta por:

- Versão da sintaxe de resposta;
- Nome de quem responde;
- Respostas para todos os certificados presentes no pedido;
- Extensões opcionais;
- OID do algoritmo de assinatura;
- Assinatura da resposta.

As respostas para cada certificado nos pedidos são compostas por:

- Identificador do certificado em questão;
- Estado do certificado (*good*,*revoked* ou *unknown*);
- Intervalo de validade da resposta;
- Extensões opcionais.

O estado *good* representa uma resposta positiva, indicando que o certificado não foi revogado, mas não confirma que o certificado foi realmente emitido ou que a resposta, quando formada, está dentro do prazo de validade do certificado. Para tal, dá-se uso às extensões opcionais para comunicar informações adicionais, tais como validade, confirmação de emissão, etc.

O estado *revoked* indica que o certificado foi revogado, seja temporariamente ou permanentemente.

O estado *unknown* indica que o servidor não reconhece o certificado em questão.

Todas as resposta OCSP são assinadas digitalmente e a chave de assinatura deve pertencer a uma das seguintes entidades:

- Autoridade de Certificação (CA) que emitiu o certificado;
- Um emissor de resposta de confiança, em que o emissor do pedido confia na sua chave pública;
- Um emissor de resposta autorizado pela CA, em que este tem um certificado emitido especificamente para autorizar a responder a pedidos OCSP no lugar da CA.

O Uniform Resource Locator (URL) do servidor OCSP está incluído no certificado, na extensão "*Authority Information Access*".

2.3 CRL

Uma Certificate Revogation List (CRL) [32] é uma lista de certificados revogados pela CA que os emitiu. Esta lista permite que as aplicações, em alternativa ao uso de OCSP, validem se um certificado é valido, mas neste caso só indica se o certificado é válido no momento de emissão da CRL, que geralmente é diferente do momento em que o pedido da CRL é feito. O pedido da CRL é feito a um URL associado à CA do certificado, que responde com a lista de todos os certificados revogados.

Numa CRL, os certificados podem ter um de dois estados:

- Revogado: o certificado está irreversivelmente revogado, geralmente associado à chave privada estar comprometida.
- Suspenso: o certificado está temporariamente inválido, podendo voltar a ser válido. Está geralmente associado a incertezas de comprometimento da chave privada.

Além do estado de revogação do certificado, a CRL também pode conter a razão de revogação.

- *Unspecified*: certificado foi revogado sem razão aparente;

- *KeyCompromise*: a chave privada associada ao certificado foi comprometida. Isto inclui em caso de roubo ou perda do dispositivo que contenha a chave privada;
- *CACompromise*: a chave privada da CA foi comprometida. Nesta situação, todos os certificados da CA são revogados;
- *AffiliationChanged*: o utlizador a quem o certificado pertence, terminou ligações com a organização a que estava associado no certificado;
- *Superseded*: outro certificado foi emitido a substituir o revogado.
- *CessationOfOperation*: a CA cessou o seu funcionamento. Código utilizado na revogação do certificado da CA;
- *CertificateHold*: razão que indica que o certificado está suspenso, podendo ser revogado e assim mudando a razão ou deixar de estar suspenso, voltando a ser válido.
- *RemoveFromCRL*: no caso de um certificado ter sido suspenso e depois ativo, esta razão é usada para indicar que o certificado já esteve suspenso, mas não está revogado.

A utilização das CRL, idealmente, só acontece na situação em que o serviço OCSP não esteja disponível, pois nas CRL a informação de verificação da validade do certificado não é gerada no momento do pedido, enquanto que no serviço OCSP essa informação é gerada no momento do pedido.

2.4 TIMESTAMP

Um serviço de selo temporal (*timestamp*) [37], fornecido por uma Time Stamping Authority (TSA), fornece prova de que uma informação existia e certa altura especifica e que esta não foi alterada desde essa altura.

Sendo uma TSA responsável por criar *timestamps* que validam a existência de um documento num dado momento, estas têm os seguintes obrigações:

- usar uma fonte de tempo de confiança;
- incluir um valor temporal fidedigno em cada timestamp;
- por cada *timestamp* gerado, este ser representado por um inteiro único;
- sempre que possível, produzir um *timestamp* ao receber um pedido valido;
- incluir em cada *timestamp* um identificador da política de segurança utilizada na sua geração;
- só criar *timestamps* da *hash* da informação;

- examinar a função de *hash* identificada pelo OID, verificando se o comprimento da *hash* recebida coincide com o comprimento determinado pela função;
- não examinar a informação recebida, para além do comprimento da *hash*;
- não incluir qualquer identificação de quem requer um *timestamp*;
- assinar cada *timestamp* com uma chave gerada exclusivamente para o propósito de assinar *timestamps* e isto estar indicado no respetivo certificado.
- incluir informação adicional no *timestamp* caso seja pedido, mas só informação suportada pela TSA. Caso tal pedido de informação não seja suportada, deve ser devolvida uma mensagem de erro.

2.4.1 *Funcionamento*

De acordo com o Request For Comments (RFC) 3161 [37]:

1. A entidade que pretende pedir um *timestamp*, faz o pedido à TSA, enumerando o hash e demais informação necessária;
2. TSA gera o *timestamp* e envia-o na resposta à entidade que o requisitou;
3. A entidade ao receber a resposta verifica a mensagem de erro. Caso não exista erro, a entidade deve:
   - verificar se a informação que foi adicionado o selo temporal corresponde à informação enviada;
   - verificar a validade da assinatura digital do *timestamp*;
   - verificar os variados campos da resposta recebida;
   - verificar se o *timestamp* contem o certificado correto da TSA, a *hash* da informação correta e o OID da função de *hash* correto;
   - verificar a prontidão da resposta, verificando o tempo incluído na resposta com uma fonte de tempo fiável;
   - verificar se o certificado da TSA ainda é válido, i.e., se não foi revogado;
   - verificar a política de segurança e avaliar se a política utilizada é aceitável.

2.5 PAdES

Em 2009 surgiu um *standard* de Assinaturas Digitais avançadas, especificado no ETSI TS 102 778 [19], publicado pela European Telecommunications Standards Institute (ETSI) na

Europa com o objetivo facilitar as transações digitais seguras. Este standard definiu uma serie de perfis de PDF Advanced Electronic Signatures (PAdES), que respeitam todos os requisitos exigidos pela Diretiva Europeia 1999/93/EC [16].

2.5.1 *PAdES Básico*

Perfil básico que se diferencia principalmente pelo uso de uma assinatura Public Key Cryptography Standards (PKCS) #7 em vez de uma assinatura CMS Advanced Electronic Signatures (CAdES). [28]

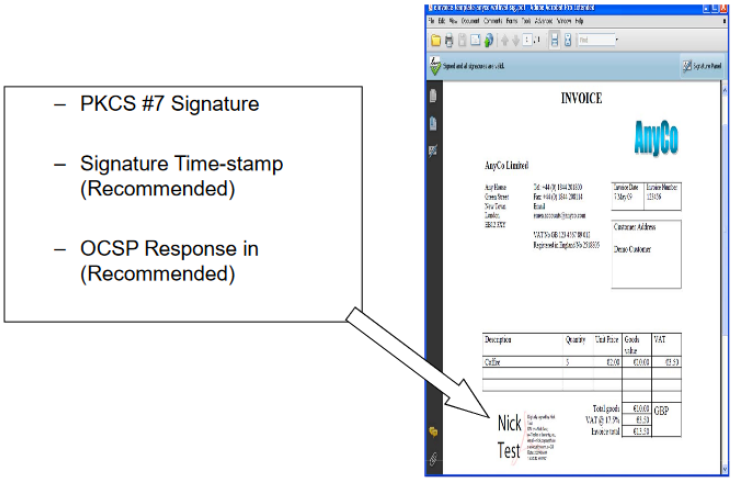


Figura 1: Perfil PAdES-Básico (Fonte: ETSI TS 102 778-1 V1.1.1 [30])

Como é possível verificar na Figura 1 e segundo o ETSI TS 102 778-1 V1.1.1 [30], o perfil PAdES-Básico tem as seguintes características:

- Assinatura PKCS #7;
- Suporta assinaturas em série;
- Recomenda a colocação de *timestamp*;
- Recomenda a inclusão de informação de revogação (CRL ou OCSP);
- A assinatura protege a integridade do documento e autentica o assinante;
- Inclusão opcional das razões de assinatura;
- Inclusão opcional da descrição de localização da assinatura;
- Inclusão opcional do contacto do assinante;
- Inclusão opcional de um "*legal content attestation*".

2.5.2 *PAdES-Baseline-B*

Perfil apropriado para assinaturas elétronicas a curto prazo, que equivale aos perfis da especificação prévia do PAdES, PAdES-BES(sem inclusão da política de assinatura) e PAdES-EPES(com inclusão da politica de assinatura) (ETSI TS 103 172 V2.1.1). [29]

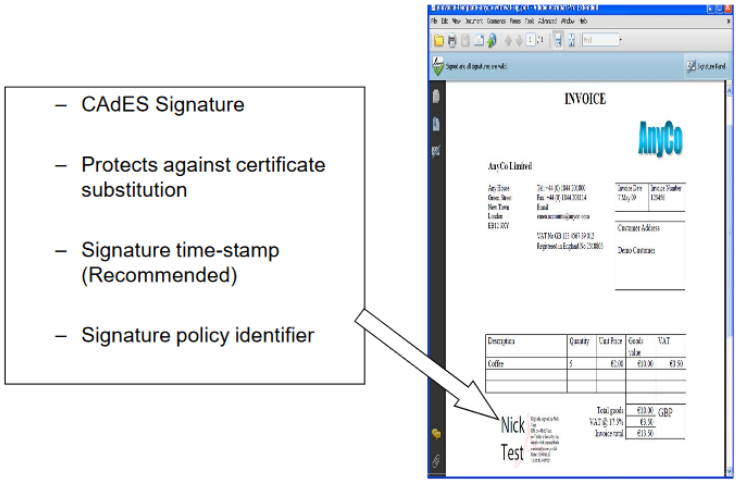


Figura 2: Perfil PAdES-Baseline-B e T (Fonte: ETSI TS 102 778-1 V1.1.1 [30])

Como é possível verificar na Figura 2 e segundo o ETSI TS 102 778-1 V1.1.1 [30], o perfil PAdES-Baseline-B tem as seguintes características:

- Assinatura PDF baseada no CAdES-BES (ETSI TS 101 733 [27]);
- Inclusão obrigatória da referência ao certificado de assinatura;
- Inclusão opcional da política de assinatura (Exemplo: Política CMD de assinatura qualificada POL#16 [4]);
- Inclusão opcional de tipos de compromisso que indicam a razão de assinatura;
- Recomenda a colocação de *timestamp*;
- Suporta assinaturas em serie;
- A assinatura protege a integridade do documento e autentica o assinante;
- Inclusão opcional de atributos CAdES: Data e Hora de assinatura, Atributos do assinante, *timestamp* do conteúdo;
- Inclusão opcional de um "*legal content attestation*".

2.5.3 *PAdES-Baseline-T*

Assinaturas em conformidade com o perfil PAdES-B-T, são assinaturas com conformidade com o perfil PAdES-B-B, com a adição de uma prova temporal, fornecida por um serviço de confiança, comprovando que a assinatura existia numa certa data a uma certa hora, recomendando que seja utilizado um *timestamp* para tal efeito.

2.5.4 *PAdES-Baseline-LT*

Assinaturas em conformidade com o perfil PAdES-LT, são assinaturas com conformidade com o perfil PAdES-T mas que podem ser validades, mesmo depois de um longo período de tempo, quando a autoridade de certificação não está disponível.

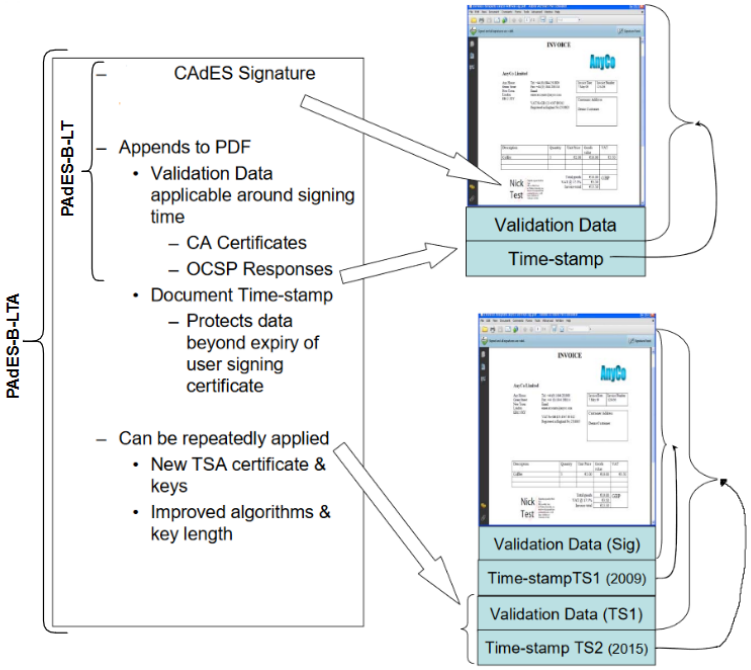


Figura 3: Perfil PAdES-Baseline-LT e LTA (Fonte: ETSI TS 102 778-1 V1.1.1 [30])

As assinaturas PAdES-B-LT e PAdES-B-LTA equivalem ao perfil PAdES-B-LTV da especificação prévia dos perfis PAdES. E como se pode ver pela Figura 3 e segundo o ETSI TS 103 172 [31], uma assinatura em conformidade com o perfil PAdES-B-LT tem que ter as seguintes caraterísticas:

- Inclusão obrigatória da cadeia de certificação usada para validar a assinatura, seus atributos e *timestamps*.

- Inclusão obrigatória de informação de revogação (CRL ou OCSP) dos certificados usados para assinar e validar a assinatura, os seus atributos e *timestamps*.

2.5.5 *PAdES-Baseline-LTA*

Assinaturas em conformidade com o perfil PAdES-LTA, são assinaturas com conformidade com o perfil PAdES-LT em que um ou mais *timestamps* foram adicionados ao documento (diferente de *timestamp* da assinatura) para proteger a assinatura e as informações de validação da mesma presentes. [31]

Como se pode ver pela Figura 3 e segundo o ETSI TS 103 172 [31], uma assinatura em conformidade com o perfil PAdES-B-LTA tem que ter as seguintes caraterísticas:

- Adição de um ou mais *timestamps* ao documento depois de adicionada toda a informação de validação da assinatura.
- Inclusão obrigatória de toda a informação de validação (cadeia de certificados e informação de revogação dos mesmos, CRL ou OCSP) necessárias para validar o certificado de assinatura, certificados dos atributos da assinatura e certificados de criação de *timestamps*, tanto os da assinatura como do documento.

2.6 SCMD

A Chave Móvel Digital (CMD) [22] é um meio de autenticação online simples e seguro para os cidadãos portugueses. Usado em sites públicos e privados suportados, este só necessita do número de telefone e um PIN para obter o segundo fator de autenticação, uma One-time password (OTP) recebida por SMS ou "Push Notification". A CMD surgiu como uma alternativa mais portável e prática ao uso do cartão de cidadão como meio único de autenticação.

Como o Cartão de Cidadão, a CMD também tem um certificado de assinatura digital qualificada, permitindo assinar documentos digitais e a assinatura poder ser verificada por qualquer entidade. Todos os processos de emissão, ativação e revogação do certificado de assinatura digital e o processo de assinatura qualificada *"server-sided"* de documentos são geridos pelo SCMD.

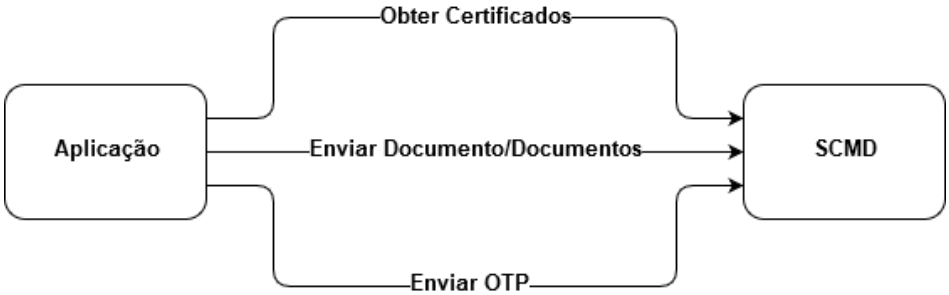


Figura 4: Pedidos do serviço SCMD segundo a sua especificação[8]

Para assinar um documento ou vários documentos com o SCMD, o cidadão tem que, utilizando uma aplicação que comunique com o SCMD, submeter o(s) documento(s) para assinatura e ao mesmo tempo se autenticar com o seu número de telemóvel associado à CMD e com o PIN de assinatura.

Observando a Figura 4, a aplicação começa por, utilizando o número de telemóvel do cidadão, pedir a cadeia de certificados do cidadão, para estes serem colocados com a assinatura no(s) documento(s), para a assinatura poder ser validada. Após o pedido dos certificados, a aplicação prepara e envia a informação a ser assinada na forma de um hash ou vários hashes no caso de serem vários documentos a assinar, em conjunto com o número de telemóvel e pin de assinatura, para se poder autenticar com o SCMD. Como resposta ao pedido, é enviado um identificador do processo.

Por fim, havendo sucesso na autenticação do cidadão, o cidadão recebe um OTP por SMS ou *Push Notification*, para ser submetido pela aplicação ao SCMD em conjunto com o identificador do processo, assim obtendo como resposta a assinatura ou as assinaturas gerada(s) ”server-side’. A aplicação fica responsável por colocar a(s) assinatura(s) no(s) documento(s).

Este processo de assinatura digital qualificada é suportado pelo esquema de assinaturas digitais PAdES descrito na secção 2.5, em que a aplicação que comunica com o SCMD tem que suportar e conseguir colocar a assinatura no documento PDF, como também colocar uma representação visual da mesma.

2.7 Trabalhos Relacionados

2.7.1 *Aplicação Autenticação.gov*

Aplicação desenvolvida para *Windows*, *Linux* e *Mac*, pela Agência para a Modernização Administrativa, que permite ao cidadão português tirar partido das funcionalidades eletrónicas do seu Cartão de Cidadão. Esta pode ser utilizada para visualizar e gerir os dados no Cartão de Cidadão e assinar documentos digitais. [9]

Nesta aplicação o cidadão poderá efetuar as seguintes operações:

- Visualização da informação e foto do cidadão;
- Visualização da morada do cidadão e confirmação da alteração de morada;
- Edição das notas;
- Imprimir os dados do Cartão de Cidadão;
- Assinatura digital de documentos PDF e outros ficheiros;
- Visualização dos certificados do Estado e do cidadão;
- Registo dos certificados do Estado e do cidadão (específico de Microsoft Windows);
- Gestão de PINs;

A operação de assinar digitalmente documentos PDF com a SCMD é a operação com maior interesse para esta dissertação.

Assina utilizando o perfil PAdES-Básico e suporta:

- a assinatura de vários documentos em conjunto;
- a adição de um *timestamp*;
- adição de atributos profissionais;
- adição de motivo de assinatura;
- adição de localidade de assinatura;
- escolha de assinar com assinatura visível ou invisível;
- escolher o local da assinatura, seja a página, seja o local na página;
- visualização do documento prévia à assinatura;.

Tendo esta aplicação operações idênticas às planeadas para a aplicação desta dissertação, esta será objeto de comparação com os resultados finais do desenvolvimento da aplicação móvel de assinatura de documentos PDF com o SCMD.

2.8 SUMÁRIO

Com a criação do regulamento eIDAS na União Europeia, veio a possibilidade de criação de meios de identificação eletrónica legitima e segura e esquemas de assinaturas eletrónicas avançadas e qualificadas.

Em Portugal, aproveitando esta regulamentação, a CMD foi criada. Este serviço dá ao cidadão o acesso a vários serviços sem ser necessário o uso do Cartão de Cidadão, como a autenticação em sites e portais online de serviços públicos e privados, assinatura digital qualificada de documentos verificável e válida em termos legais.

Para tornar a assinatura criadas pelo SCMD mais robusta, dois serviços podem ser utilizados em conjunto com os perfis PAdES de assinatura. Um serviço OCSP ou CRL, que verifica a validade do certificado usado para criar a assinatura digital qualificada SCMD e um serviço *timestamp*, que confirma o tempo em que a assinatura digital qualificada foi criada.

3

PROBLEMA E DESAFIOS

Neste capitulo será apresentado o problema e os desafios à volta da criação de assinaturas digitais utilizando os perfis PDF Advanced Electronic Signatures (PAdES) e de só existir a aplicação Autenticação.gov para assinar documentos PDF com o serviço Chave Móvel Digital. Serão detalhadas as funcionalidades necessárias para a aplicação móvel e por fim as soluções para estes problemas e desafios que consiga implementar todas as funcionalidades necessárias.

3.1 PROBLEMA E DESAFIOS

Com a existência da Chave Móvel Digital (CMD) e da aplicação Autenticação.gov, tornou-se possível a criação da assinatura digital qualificada para cada cidadão sem necessitar de um cartão físico. Esta aplicação, porém, tem as suas limitações.

Estando a aplicação Autenticação.gov apenas disponibilizada para Windows e Mac, limita a assinatura de documentos a dispositivos geralmente associados a um uso mais fixo e menos móvel, não sendo possível assinar documentos num âmbito mais móvel e prático, principalmente em *smartphones* onde é recebido o One-time password (OTP) para validar o pedido de assinatura da CMD.

Além das plataformas que suportam a aplicação Autenticação.gov serem limitadas, esta também é limitada no perfil de assinatura, mais especificamente nos perfis PAdES. Só utiliza PAdES-Básico que, como já referido, não coincide com os perfis PAdES-Baseline, principalmente pelo facto de assinar com Public Key Cryptography Standards (PKCS)#7 e não com CMS Advanced Electronic Signatures (CAdES). Apesar de a aplicação permitir a adição de *timestamp* opcional, tornando a assinatura mais próxima do perfil PAdES-B-T, esta continua a utilizar uma especificação desatualizada que não coincide com PAdES-Baseline mais recente, assim não permitindo, principalmente, a adição de validação a longo prazo.

O desenvolvimento na área das assinaturas digitais qualificadas utilizando os perfis PAdES-Baseline peca pela falta de diversidade das linguagens de programação utiliza-

das. Sendo as melhores implementações em JAVA (as principais sendo IText[1] do iText Group NV e Digital Signature Service[2] da comissão europeia, que por sua vez usa o IText), a sua utilização no desenvolvimento para a plataforma iOS torna-se inviável sem ter recurso a serviços intermédios (na data de escrita deste documento). Porém, existem outras implementações ou futuras implementações em tecnologias e linguagens diferentes, mas estas ou têm custos bastantes elevados, derivado ao facto de pertencerem a pacotes de software mais completos, ou estão numa fase muito inicial de desenvolvimento. o que torna inviável a sua utilização nesta dissertação (i.e. SecureBlackBox[3], PDFix[4], etc).

Para além das implementações pecarem pela falta de linguagens de programação, estas também não estão preparadas para serem utilizadas em conjunto com o Serviço Chave Móvel Digital (SCMD). Como a assinatura é gerada externamente a partir do SCMD, este tem um passo intermédio que requer a inserção de um código único recebido (por SMS ou *Push Notification*), para a assinatura ser enviada.

Assim surgiu o desafio de desenvolver uma aplicação para dispositivos móveis, que, utilizando o SCMD, assine documentos segundo a especificação mais recente dos perfis de assinatura PAdES-Baseline, sem o documento sair do dispositivo eletrónico.

3.2 FUNCIONALIDADES

Tendo em conta os objetivos desta dissertação e o desenvolvimento de uma aplicação iOS e Android, esta terá as seguintes funcionalidades (que já incluem os *guidelides* da Agência para a Modernização Administrativa, I.P. (AMA) para aplicações de assinatura CMD, assim como algumas opções existentes na aplicação Autenticação.Gov da AMA):

1. Utilização das duas últimas versões (1.6 e 1.7) da "Especificação dos serviços de assinatura CMD", estando por omissão configurada para utilizar a versão 1.7, mas podendo o utilizador ir ao menu de configuração e alterar a versão por omissão;

2. Na versão 1.7, é cifrado (no dispositivo) com chave publica disponibilizada pela Agência para a Modernização Administrativa, I.P. (AMA) o número de telefone, PIN e OTP inserido pelo utilizador.

3. Assinatura de um ou mais ficheiros PDF locais e/ou em URL identificados pelo utilizador;

1 https://pdfix.net/
2 https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/DSS
3 https://www.secureblackbox.com/
4 https://pdfix.net/

4. Apresenta o(s) ficheiro(s) a assinar de acordo com a política What You See Is What You Sign (WYSIWYS), diretamente na aplicação ou em aplicação externa (cf. secção 3.2.4 da POL#16 [4]);
5. Guarda do ficheiro assinado na app;
6. "Envio" do ficheiro para outra app, por exemplo para e-mail;
7. "Recepção" de ficheiro PDF de outra app, para assinatura local;
8. Exige a introdução do PIN da CMD por cada assinatura ou *batch* de assinatura efetuada, tendo o utilizador a possibilidade de o guardar o número de telemóvel na app;
9. Não apresenta, visível para o utilizador, os dígitos do PIN da CMD;
10. Não guarda nos logs os dados do PIN;
11. Não transmite para qualquer outra aplicação dados do PIN da CMD;
12. Apresenta sempre o(s) documento(s) a assinar ao utilizador;
13. Deteta a existência de novas versões da aplicação e permite a atualização automática da aplicação;
14. Não partilha as credenciais de acesso (APPLICATION_ID) da aplicação ao serviço CMD assinatura, com qualquer outra entidade ou aplicação;https://www.overleaf.com/project/5c
15. Autentica o utilizador para acesso à aplicação de assinatura com Cartão de Cidadão ou CMD – essa autenticação será feita por PIN de 6 dígitos, impressão digital ou reconhecimento facial;
16. Utiliza canal seguro (Transport Layer Security (TLS)) para transmissão de credenciais do utilizador a serviço CMD assinatura;
17. Apresenta os dados a serem assinados no formato XML (que contém o *hash* do documento a assinar em formato *byte array*) – também designado por representação dos dados a serem assinados (DTBS/R) – ao servidor de assinatura CMD da AMA, garantindo que corresponde aos dados/documento a assinar apresentado pelo/ao assinante (cf. secção 3.1.2 da POL#16[4]);
18. Guarda a assinatura embebida no documento PDF assinado com os dados assinados (cf. secção 3.1.3 da POL#16[4]);

19. A guarda da assinatura é efetuada no formato, perfil e nível de assinatura definida pela aplicação, permitindo a aplicação o seguinte tipo de assinaturas: PAdES_BASELINE_B, PAdES_BASELINE_T, PAdES_BASELINE_LT e PAdES_BASELINE_LTA;

20. O tipo de assinatura por omissão é a PAdES_BASELINE_T, podendo o utilizador ir ao menu de configuração e alterar o tipo de assinatura por omissão;

21. O servidor de *timestamping* a utilizar por omissão é o servidor do Cartão de Cidadão (http://ts.cartaodecidadao.pt/tsa/server - POST), podendo o utilizador ir ao menu de configuração e alterar o tipo de servidor de *timestamping* por omissão;

22. É adicionada a referência à Política CMD de Assinatura Qualificada (indicando o seu Identificador de Objeto (OID) 2.16.620.2.1.2.2), de modo a permitir que partes confiantes e outras pessoas interessadas possam encontrar informação sobre as políticas e práticas seguidas na aposição da assinatura. (cf. secção 3.1.3 da POL#16[4]);

23. Com a assinatura dos dados/documento, a aplicação deve permitir que o assinante associe ao campo Razão/Reason (ou commitment), um (ou vários) dos seguintes tipo de compromissos (e respetivo OID), de modo a contextualizar (e desambiguar) o propósito e significado da assinatura, assim como a natureza da responsabilidade assumida: Prova de origem / *Proof of origin*, Prova de aprovação / *Proof of approval*, Prova de criação / *Proof of creation*, Autenticação de dados / *Data Authentication*, Autenticação de Entidade / *Entity Autentication*, Autoria / *Authorship*, Revisão / *Review*, Cópia / *Copy*, Testemunha de assinatura / *Signature Witness*, Vinculação ao conteúdo assinado / *Bound to data signed*, Aprovação intermédia / *Intermediate approval*. (cf. secção 3.2.2 da POL#16[4]);

24. A aplicação permite que o assinante valide que os dados de identificação do documento, a assinar, recebidos na mensagem SMS/*Push notification*, são os mesmos que lhe são apresentados na interface de utilizador da aplicação. (cf. secção 3.2.4 da POL#16[4]);

25. A aplicação permite identificar e informar sobre os vários passos do processo de assinatura (cf. secção 3.2.4 da POL#16[4]) à medida que os mesmo ocorrem:
    Nota: No processo de assinatura é apresentada uma imagem de publicidade na app, sendo que abaixo (ou acima) da imagem deve passar a seguinte informação (que deveria permitir fazer *scroll* da sessão), à medida que é efetuada a comunicação com o serviço CMD:

    a) Comunicação com serviço CMD da AMA – obtenção de certificado qualificado do assinante – mínimo de 3 segundos

b) Comunicação com serviço CMD da AMA – Obtido certificado qualificado de <Nome indicado no *Common Name* do *Subject* do certificado> – mínimo de 3 segundos

c) Geração do *hash* do documento PDF <nome do documento ou URL identificado pelo assinante> – mínimo de 3 segundos

d) Comunicação com serviço CMD da AMA – Envio do *hash* do documento PDF – mínimo de 3 segundos

e) Deve ter recebido, por SMS ou *push notification*, o código de segurança enviado pelo serviço CMD da AMA, para assinatura do(s) documento(s) PDF <nome do(s) documento(s) enviado(s) na comunicação para a AMA> – mínimo de 3 segundos

f) Introduza o código de segurança recebido: – mínimo de 3 segundos

g) Comunicação com serviço CMD da AMA – Envio do código de segurança – mínimo de 3 segundos

h) Comunicação com serviço CMD da AMA – Obtenção de assinatura do *hash* – mínimo de 3 segundos

i) Aposição da assinatura ao documento PDF – mínimo de 3 segundos

j) Documento PDF assinado gravado com o nome <nome do PDF assinado> – mínimo de 3 segundos

26. Identificar claramente o passo a partir do qual a assinatura é criada, garantindo que o assinante conhece a responsabilidade assumida no ato de assinar e que fica vinculado a essa responsabilidade e ao compromisso assumido (cf. secção 3.2.4 da POL#16[4])
Nota: Na sequência do processo de assinatura anterior a última informação apresentada será a seguinte:

    a) Assinatura eletrónica qualificada aposta ao documento guardado com o nome <nome do PDF assinado>. Este documento tem valor legal de acordo com o Regulamento (UE) nº 910/2014 do Parlamento Europeu e do Conselho de 23 de Julho de 2014 e com a Portaria n.º 77/2018 de 16 de Março – mínimo de 5 segundos, até o utilizador fazer ok.

27. Guiar o assinante na guarda da assinatura embebida no documento PDF assinado (cf. secção 3.2.4 da POL#16[4])
Nota: Já incluído no ponto anterior.

28. A aplicação fornece ao assinante informação e conselho relativo ao processo de criação de assinatura e consequências legais, assim como garante na extensão possível, que o interface do utilizar fornece um ambiente legal válido para assinatura (cf. secção 4.2.1

da POL#16[4]).
Nota: Já incluído nos pontos anteriores.

29. A aplicação permite a assinatura visível e invisível – por omissão, a assinatura é visível, embora o utilizador a possa alterar no menu de configuração;

30. A aplicação permite que o utilizador personalize a assinatura visível (no menu de configuração), incluindo texto e imagem. A parte do texto deve sempre incluir o nome do assinante (obtido do certificado), assim como a hora e data de assinatura;

31. A aplicação permite que o utilizador identifique o local onde pretende colocar a assinatura visível no documento PDF, identificando o local onde a assinatura deverá ficar (um pouco ao exemplo do que é feito na aplicação Autenticaçao.Gov da AMA);

32. A aplicação permite que o utilizador personalize a localidade (e outros dados da morada aceites pelo PAdES) no menu de configuração, assim como selecione se a mesma deve ou não ser incluída no documento (por omissão não é incluída);

33. A aplicação permite que o utilizador configure, no menu de configuração, a utilização de *proxy* de sistema, indicando o endereço e porta do servidor de *proxy*, assim como a autenticação no *proxy* (utilizador e palavra-passe). Por omissão, a utilização de *proxy* de sistema não está selecionada.

34. A aplicação obriga que o utilizador aceite as "Condições Gerais" na primeira vez que a utilizar. Essas "Condições Gerais" estarão depois também disponíveis a partir do menu de configuração, para visualização.

3.3 SOLUÇÃO

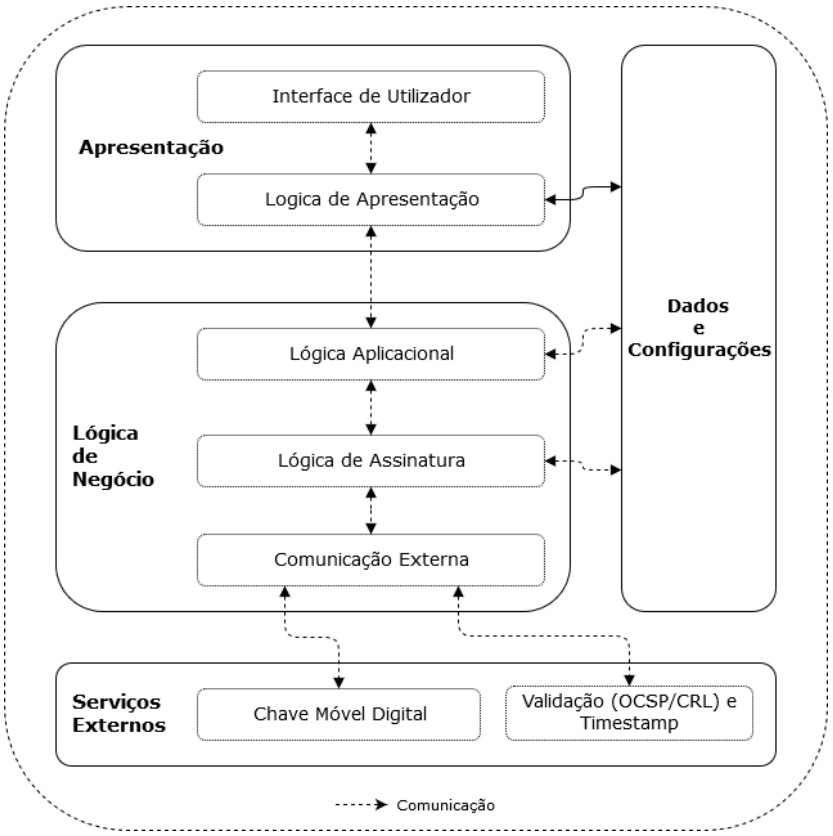


Figura 5: Arquitetura da aplicação

Sendo o objetivo desta dissertação a conveniência e a segurança no uso da Chave Móvel Digital para assinar documentos, a solução começa por uma aplicação móvel Android e iOS que processa toda a lógica no dispositivo, sem comunicar com outros serviços externos, apenas comunicando com o Serviço Chave Móvel Digital e outros relacionados com validação e verificação de validade do certificado de assinatura. Daí, surgiu a arquitetura da Figura 5.

3.3.1 *Apresentação*

Camada responsável pela interação com o utilizador e pelo envio e receção de informação da lógica de negócio. Constituída pelos componentes de interface de utilizador e de lógica de apresentação, estes são responsáveis pela criação de ecrãs, navegação entre estes e comunicação com a camada de lógica de negócio:

- Ecrã principal: Aqui são apresentados ao utilizador todas as informações necessárias, prévias ao uso da aplicação, pedidos de autorização de acesso a componentes do sistema, acesso a ecrã de configurações e inicio de processo de assinatura.
- Ecrã de documentos assinados: Ecrã onde é possível visualizar, partilhar e apagar os documentos já assinados.
- Ecrã de configurações: em um ou mais ecrãs é possível configurar o aspeto de assinatura, o perfile PAdES-Baseline usado para assinar, o servidor de *timestamping* a utilizar e as razões/*commitments* a colocar na assinatura.
- Ecrã de seleção do documento a assinar: o utilizador escolhe o documento que pretende assinar a partir do sistema de ficheiro ou de um URL externo.
- Ecrã de de seleção do local de assinatura: caso o utilizador pretenda assinatura visível, este escolhe a página onde pretende colocar a assinatura e define onde quer que a assinatura seja colocada na página.
- Ecrã de Autenticação com a CMD: aqui o utilizador coloca o seu número de telemóvel e o PIN para proceder à autenticação com a CMD e receber o OTP para poder assinar o documento.
- Ecrã de Inserção do OTP da Chave Móvel Digital: aqui o utilizador coloca o OTP recebido por SMS ou *push notification*, assim recebendo a assinatura.
- Ecrã de informação do processo de assinatura: um ou mais ecrãs que informam o utilizador do passo da assinatura que está a decorrer (ponto 25. das funcionalidades).
- Ecrã de documento assinado: no fim do processo de assinatura, é apresentado ao utilizador um aviso de documento assinado, um botão para poder voltar ao inicio e métodos para abrir e partilhar o documento.

3.3.2 *Lógica de Negócio*

Camada constituída pelos componentes de lógica aplicacional e pelo componente de comunicação externa. Estes são responsáveis pelo processamento de todas as funcionalidades da aplicação.

Lógica de Assinatura

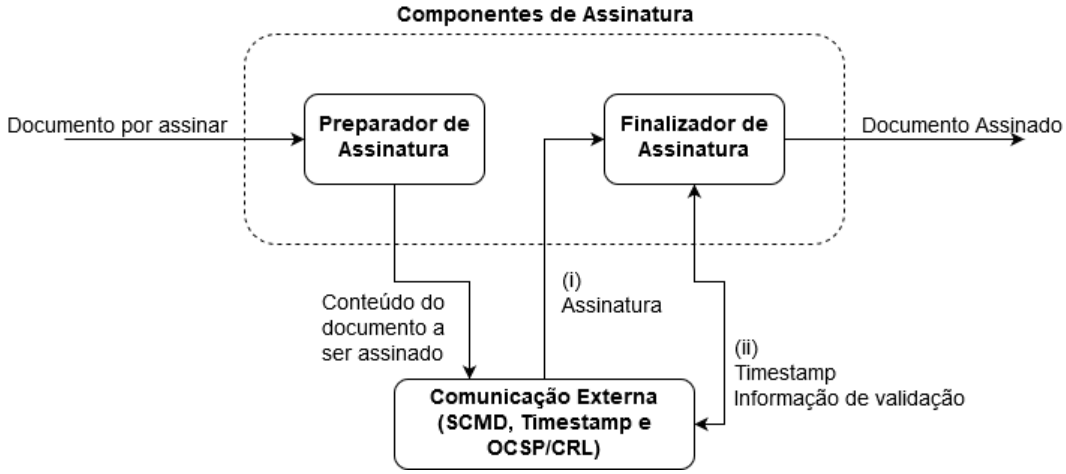


Figura 6: Componentes da Lógica de Assinatura

Sub-camada da Lógica de Negócio, responsável pela criação da assinatura do documento, no perfil PAdES-Baseline escolhido. Como é possível observar pela Figura 6, esta camada está dividida em dois componentes, o Preparador de Assinatura e o Finalizador de Assinatura.

O Preparador de Assinatura recebe o documento escolhido pelo utilizador na camada de Apresentação e extrai o conteúdo a ser assinado. O processo de extração do conteúdo a ser assinado dá-se pelos seguintes passos, conforme a especificação de assinaturas PDF no ISO 32000-1 [11]:

- Extração do conteúdo do documento original;
- Criação de um novo documento com o conteúdo original, com a alocação do espaço para a assinatura e com a cadeia de certificados de assinatura;
- Serialização do documento com o conteúdo e o espaço da assinatura alocado.

Após a extração do conteúdo a ser assinado, este é enviado ao componente de comunicação com o SCMD, para assim proceder à autenticação e ao processo de criação de assinatura com o SCMD.

O Finalizador de Assinatura, após o termino do processo de assinatura com o SCMD, recebe a assinatura enviada pelo componente de comunicação e coloca a assinatura no documento. Dependendo do perfil PAdES-Baseline escolhido, o procedimento a seguir após a colocação da assinatura no documento é diferente:

No caso do perfil PAdES-B-B, o documento é enviado ao utilizador após a colocação da assinatura.

No caso do perfil PAdES-B-T, o Finalizador de Assinatura requisita um *timestamp* para a assinatura ao serviço de *timestamp* predefinido ou definido nas configurações, e este é colocado no documento. Após este processo, o documento é devolvido ao utilizador.

No caso do perfil PAdES-B-LT, o Finalizador de Assinatura requisita um *timestamp* para a assinatura, ao serviço de *timestamp* predefinido ou definido nas configurações, e este é colocado no documento. Requisita informação de validação do certificado de assinatura, seja esta informação a resposta OCSP ou a CRL, e coloca-a no documento. Por fim, devolve o documento assinado ao utilizador.

No caso do perfil PAdES-B-LTA, o Finalizador de Assinatura requisita um *timestamp* para a assinatura, ao serviço de *timestamp* predefinido ou definido nas configurações, e este é colocado no documento. Requisita informação de validação do certificado de assinatura, seja esta informação a resposta OCSP ou a CRL, e coloca-a no documento. Após finalizar a assinatura, um *timestamp* para o documento e toda a informação de validação do certificado do *timestamp* são requisitados, seja a informação de validação uma resposta OCSP ou a CRL, e coloca-os no documento. Por fim, devolve o documento assinado ao utilizador.

Comunicação Externa

Componente responsável pela criação e envio de pedidos a serviços externos e por receber e processar a resposta destes.

Processa pedidos para o SCMD, preparando o conteúdo do documento recebido do Preparador de Assinatura, gerando a sua respetiva *hash*, e envia a *hash* em conjunto com a autenticação do utlizador. Após o utilizador proceder ao envio do OTP recebido por SMS ou *Push Notification*, o pedido de validação do OTP é gerado, enviado e é obtida a assinatura do documento na resposta, que por sua vez é enviada ao Finalizador de Assinatura para ser colocada no documento.

Também processa pedidos a Serviços *Timestamp*, OCSP e CRL. Utilizando o URL do Serviço *Timestamp* definido, gera e envia um pedido de geração de um *timestamp*. Utilizando o URL do Serviço OCSP e CRL presentes no certificado de assinatura, gera e envia um pedido ao serviço OCSP e CRL.

Componente de Visualização de PDF

Componente responsável pela visualização obrigatória, prévia à assinatura, do documento PDF a ser assinado, selecionado pelo utilizador.

Componente de anúncios publicitários

Componente de anúncios, não intrusivos, a serem apresentados ao utilizador durante o processo de assinatura

Componente de partilha

Componente responsável pela partilha externa do documento, no final do processo de assinatura.

3.3.3 *Dados e Configurações*

Camada responsável por manter informação persistente e consultável ao longo do processo de assinatura. É constituída por:

- Configuração de perfil PAdES-B de assinatura;
- Razões/*Commitments* de assinatura;
- Número predefinido de autenticação com a Chave Móvel Digital (CMD);
- Imagem e Texto visível da assinatura personalizado;
- Servidor de *timestamping* a utilizar.

3.3.4 *Serviços Externos*

Serviço Chave Móvel Digital

Serviço responsável pela criação da assinatura do documento. Aceita os seguintes pedidos, conforme o documento Cháve Movel Digital - Especificação dos serviços de Assinatura, versão 1.7 [8]:

- Pedido de certificado público do assinante, este sendo colocado no documento para validar a assinatura. Requer a submissão do número de telemóvel do assinante.
- Pedido de geração de assinatura, que aguarda a submissão do OTP, recebido pelo assinante por SMS ou *Push Notification*. Requer a submissão do número do assinante, PIN de autenticação e *Hash* do documento a ser assinado.
- Pedido de geração de várias assinaturas de documentos diferentes, que aguarda a submissão do OTP, recebido pelo assinante por SMS ou *Push Notification*. Requer a submissão do número do assinante, PIN de autenticação e as *Hash* dos documentos a serem assinados.
- Pedido de retorno de assinatura do documento submetido, enviando o OTP recebido pelo assinante por SMS ou *Push Notification*.
- Pedido de retorno de assinaturas de documentos diferentes submetidos, enviando o OTP recebido pelo assinante por SMS ou *Push Notification*.

Serviço de Timestamp

Predefinido com o serviço de *timestamp* do cartão de cidadão (*ts.cartaodecidadao.pt/tsa/server*), é responsável por devolver uma prova temporal verificável. Este não requer autenticação com nome de utilizador e password e aceita o seguinte pedido, de acordo com o Request For Comments (RFC) 3161 [37]:

- Pedido de prova temporal *Timestamp*, este sendo colocado na assinatura ou no documento, dependendo do perfil PAdES escolhido e a fase de assinatura. Requer a submissão da *hash* do conteúdo a ser *"timestamped"*.

Serviços OCSP e CRL

URL destes serviços estão presentes no certificado de assinatura, preferindo OCSP a CRL, i.e. na situação do URL OCSP não estar presente, CRL é utilizado.

Pedido OCSP devolve uma resposta que indica se o certificado é válido no tempo de assinatura, cf. RFC 6960 [33]. A resposta é colocado no documento com a assinatura.

Pedido CRL devolve uma lista de revogação de certificados emitidos pela Autoridade de Certificação (CA), emitida num certo tempo, indicando se o certificado de assinatura era válido no tempo que a CRL foi emitida, cf. RFC 5280 [23].

4

DESENVOLVIMENTO

Neste capítulo será descrito todo o processo de tomada de decisão das tecnologias as serem utilizadas para o desenvolvimento da aplicação, uma descrição da implementação feita, desde a interface à camada de negócio e por fim os resultados obtidos com a implementação.

4.1 TECNOLOGIAS E DECISÕES

4.1.1 *Aplicação Móvel*

O desenvolvimento desta dissertação começou pela escolha entre desenvolver duas aplicações nativas, uma Android e outra iOS, ou desenvolver uma aplicação multi plataforma, que partilha o mesmo código entre Android e iOS, mas mantendo a possibilidade de utilizar código nativo de cada um.

A decisão partiu de uma perspetiva de reutilizar o máximo de código possível, para o desenvolvimento da dissertação ser mais fácil e rápido. Desenvolver nativo para as duas plataformas requer código duplicado em duas linguagens diferentes (Java/Kotlin e Swift), e isto obriga a uma maior atenção à escrita e manutenção do código da aplicação, tornando o seu desenvolvimento mais lento. Também derivado à pouca experiência pessoal com desenvolvimento em iOS, a opção de desenvolver iOS em nativo não foi tão ponderada.

Disto surge a decisão de optar pelas aplicações multi plataforma. Estas abstraem o desenvolvimento nativo das plataformas Android e iOS, para um desenvolvimento semelhante a Web, mantendo uma ponte que interage com implementações nativas de cada plataforma. Desta forma, parte da aplicação, para ambas as plataformas, é desenvolvida utilizando o mesmo código e sempre que existir a necessidade de uma implementação nativa, esta é possível, tornando o desenvolvimento mais fácil e rápido. [34]

Com o objetivo de desenvolver uma aplicação multi-plataforma, as *frameworks* de desenvolvimento ponderadas foram Cordova/PhoneGap, Xamarin, Flutter e React Native.

Analisando as funcionalidades necessárias para a aplicação a desenvolver nesta dissertação, principalmente a necessidade de criar assinaturas digitais com os perfis PAdES-Baseline, a *framework* de desenvolvimento da aplicação multi plataforma terá que suportar implementações

Java ou C++ já existentes de assinaturas digitais PAdES-Baseline. Como as plataformas Flutter e Xamarin utilizam Dart e C# respetivamente, não suportam implementações existentes de Java e C++ de PAdES, logo foram descartadas.

A framework de desenvolvimento Cordova (também chamada PhoneGap) utiliza tecnologias web para desenvolver a aplicação. Utiliza javascript, HTML e CSS em vez das Application Programming Interface (API) nativas para desenvolver a interface de utilizador e utiliza mecanismos que interagem com a API nativa de Android em Java e C++ e de iOS em Objective-C e C++ para poder desenvolver funcionalidades mais avançadas, como aceder a componentes de hardware (sensores, câmara, etc).

A framework de desenvolvimento React Native combina a utilização de bibliotecas javascript com as Application Programming Interface (API) nativas em Java e C++ de Android, Objective-C e C++ de iOS, para desenvolver a aplicação. Utiliza bibliotecas javascript ou as API nativas para desenvolver funcionalidades e traduz os elementos de interface da biblioteca javascript do React Native em elementos de interface das AAPI nativas de cada plataforma.

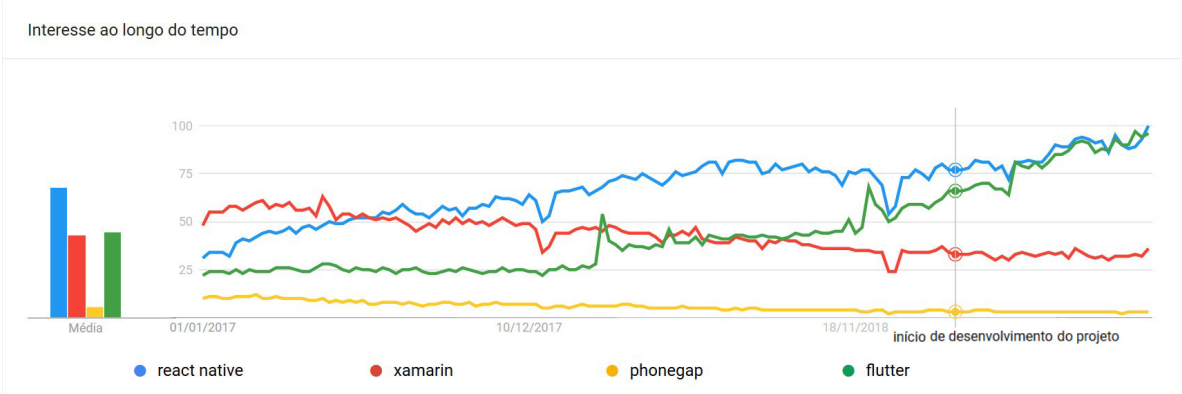


Figura 7: Interesse ao longo do tempo das *frameworks* de desenvolvimento (fonte: Google Trends)

Analisando o esquema da Figura 7, é possível observar que, segundo o Google Trends, no começo do desenvolvimento da dissertação, React Native era a *framework* de desenvolvimento mais popular.

A decisão de optar pela utilização de React Native foi tomada com base na popularidade do React Native, e pelo facto deste utilizar a API nativa de cada plataforma para renderizar a interface, obtendo uma melhor performance que Cordova. Além disto, o desenvolvimento da aplicação não envolve reaproveitamento de código de aplicações web, logo a utilização de Cordova torna-se desnecessária. [36]

4.1.2 *Assinatura e CMD*

Estando a *framework* de desenvolvimento escolhida, foi procedida à escolha da linguagem para desenvolver as funcionalidades de assinatura e de comunicação com o Serviço Chave Móvel Digital, e por sua vez o software a utilizar para assinar digitalmente documentos com os perfis PDF Advanced Electronic Signatures (PAdES).

Dando uso às funcionalidades mais avançadas das API Nativas de cada plataforma e à sua melhor performance comparativamente às bibliotecas de javascript, foi decidido utilizar módulos nativos[12] para o desenvolvimento do processo de assinatura.

Com o objetivo de reutilizar a máxima quantidade de código entre as duas plataformas e como não existem bibliotecas javascript com implementações de assinaturas digitais utilizando o perfil PAdES-Baseline, foi inicialmente decidido, a partir de módulos nativos de cada plataforma [12], utilizar C++ para desenvolver o processo de assinatura e a comunicação com o Serviço Chave Móvel Digital. Este processo seria realizado, criando pontes na API Nativa de Android em Java e de iOS em Swift para a implementação em C++, assim mantendo a mesma fonte de código para as funcionalidades implementadas, só sendo necessário implementar as pontes entre Java e C++ e entre iOS e Swift separadamente.

Sendo a aplicação Autenticação.gov pública e desenvolvida em C++, o plano inicial seria de extrair os métodos de assinatura com o perfil PAdES-Básico e os utilizar no desenvolvimento do processo de assinatura. Mas a estrutura do projeto Autenticação.gov como é muito desorganizada e o código é pouco legível e mal comentado, tornou a extração do código da assinatura PAdES-Básico muito demorada e difícil, logo tornou a reutilização de código da aplicação Autenticação.gov inviável.

Numa tentativa de manter a utilização de C++, foi encontrada uma solução implementada em C++, *The *AdES Collection*, mas esta também não podia ser utilizada nesta dissertação, pois como é dependente de bibliotecas exclusivas ao sistema operativo Microsoft Windows, o uso é limitado ao sistema operativo Microsoft Windows. [1]

Considerando a impossibilidade de utilizar C++ para o desenvolvimento das funcionalidades de assinatura sem ter que desenvolver de raiz o software necessário para assinar com perfis PAdES, foi tomada a decisão de utilizar uma solução em Java em conjunto com os módulos nativos de React Native para implementar as funcionalidade de assinatura, limitando a implementação só a Android.

As soluções de software em Java com implementações dos perfis PAdES-Baseline disponíveis são o Digital Signature Services (DSS)[2] e o *IText*[3], sendo uma das soluções utilizadas no DSS. Como o DSS não anunciava suporte para Android, foi decidido utilizar a ferramenta de geração, programação e manipulação de documentos PDF, IText, pois esta

além de ser utilizado no DSS, anunciava suporte para Android, apesar de ser na versão 5 do IText, enquanto que a atual é a versão 7.

4.2 IMPLEMENTAÇÃO

A implementação da aplicação móvel, utilizando a *framework* React Native, começou por separar o desenvolvimento da interface, do desenvolvimento das funcionalidades necessárias para o processo de assinatura. Para isto, a arquitetura presente na Figura 8 foi feita.

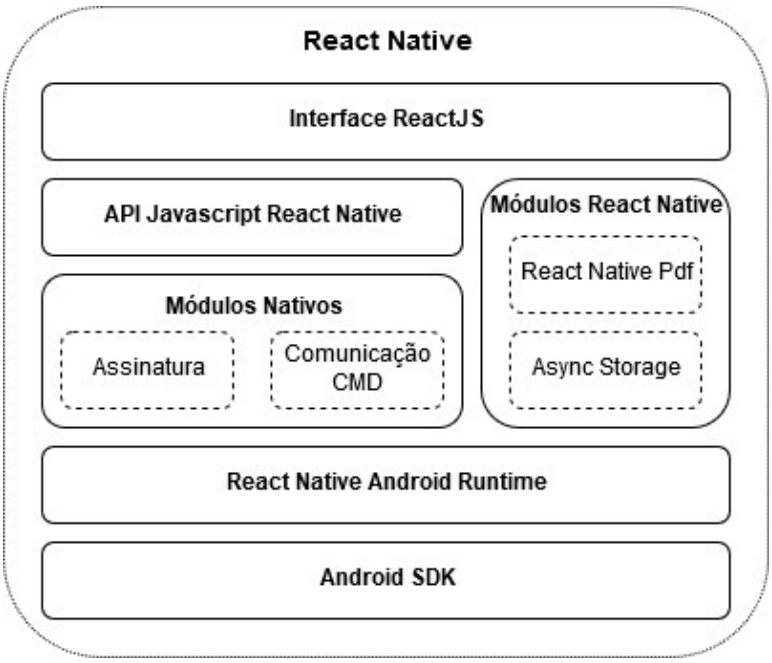


Figura 8: Arquitetura de desenvolvimento utilizando React Native.

A partir do uso das ferramentas fornecidas pelo React Native, foi possível fazer uma interface de utilizador (Interface ReactJS) utilizando as bibliotecas javascript já com componentes de interface implementados e prontos a serem utilizados e personalizados.

Para adicionar funcionalidades à aplicação, existem três opções: utilizar módulos já desenvolvidos da comunidade de React Native (Módulos React Native), sendo possível os adicionar à aplicação e utiliza-los livremente, desenvolver funcionalidades em javascript em conjunto com a interface e desenvolver módulos nativos, que utilizam a API nativa da plataforma Android (Módulos Nativos).

Para os módulos nativos comunicarem com a camada da interface, o React Native tem uma camada intermédia (API Javascript React Native), que estabelece uma ponte entre a Interface desenvolvida em Javascript e os métodos desenvolvidos com as API nativas, sendo possível invocar estes métodos na camada de Interface em javascript. Mas a comunicação

entre a interface onde o método é chamado e a API nativa onde o método vai ser executado é limitada, sendo a única informação enviada os parâmetros do método e a única informação retornada o resultado da operação, não podendo ser introduzida informação durante o processo a correr na API nativa.

Enquanto que a lógica aplicacional desenvolvida em javascript é executada sem haver necessidade de aceder à API nativa, os componentes da interface são traduzidos para componentes Nativos e os módulos desenvolvidos ou importados que dêem uso da API Nativa, são delegados ao *runtime* de Android (React Native Android Runtime). [6]

4.2.1 *Interface*

A interface de utilizar é constituída por vários ecrãs sequenciais, desenvolvidos utilizando a biblioteca *javascript* de *React Native*, que vão avançando ao longo do processo de assinatura. Cada ecrã interage com funcionalidades desenvolvidas ou importadas da comunidade.

Ecrã Inicial

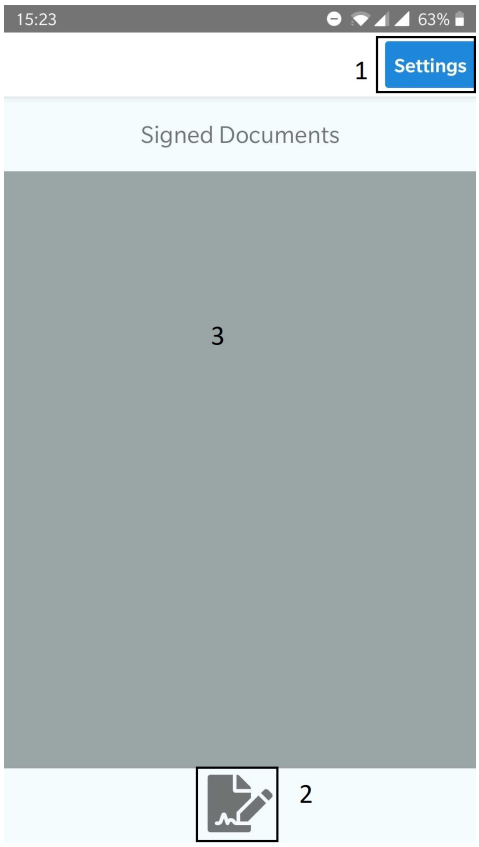


Figura 9: Ecrã Inicial

O ecrã presente na Figura 9 é o ecrã inicial, apresentado ao utilizador quando este abre a aplicação. Neste ecrã é possível abrir o menu de configurações do perfil de assinatura no botão "*Settings*" (1), iniciar o processo de assinatura, avançando para a escolha do documento, no botão com um ícone a representar uma assinatura (2) e visualizar a lista de documentos assinados (3). A lista de documentos assinados não está implementada, só o espaço para esta é que está definido.

Escolha de perfil de Assinatura

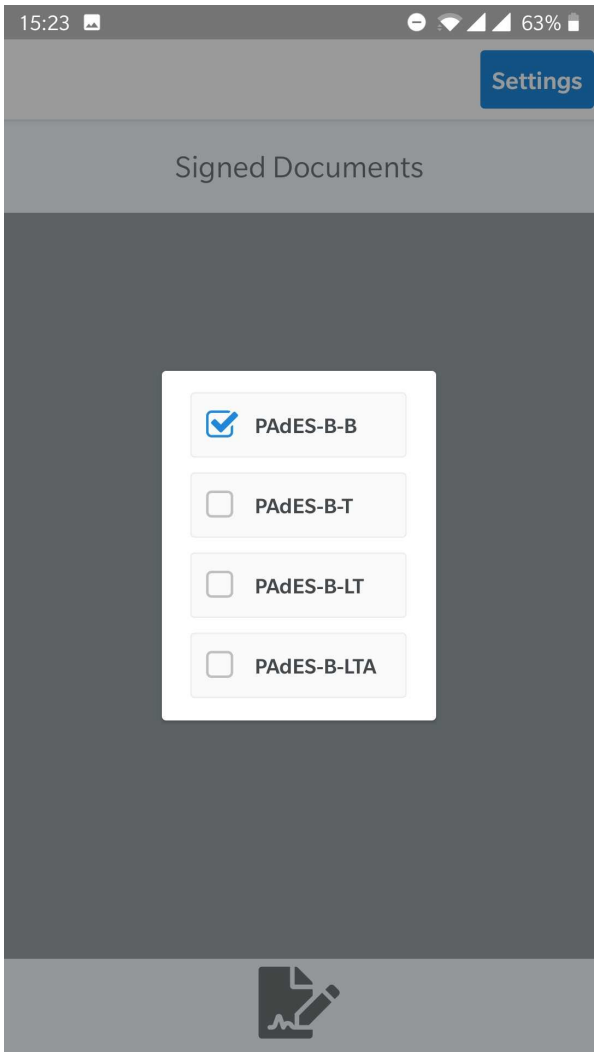


Figura 10: Menu de escolha do perfil de assinatura

O ecrã presente na Figura 10 contém o menu que é aberto quando o botão *Settings* do ecrã descrito anteriormente é clicado. Neste menu é possível selecionar um e só um dos perfis de assinatura PAdES-Baseline, determinando qual o processo de assinatura a seguir para o documento a selecionar. O perfil escolhido fica guardado e só é alterado quando outro for escolhido.

Visualização do Documento

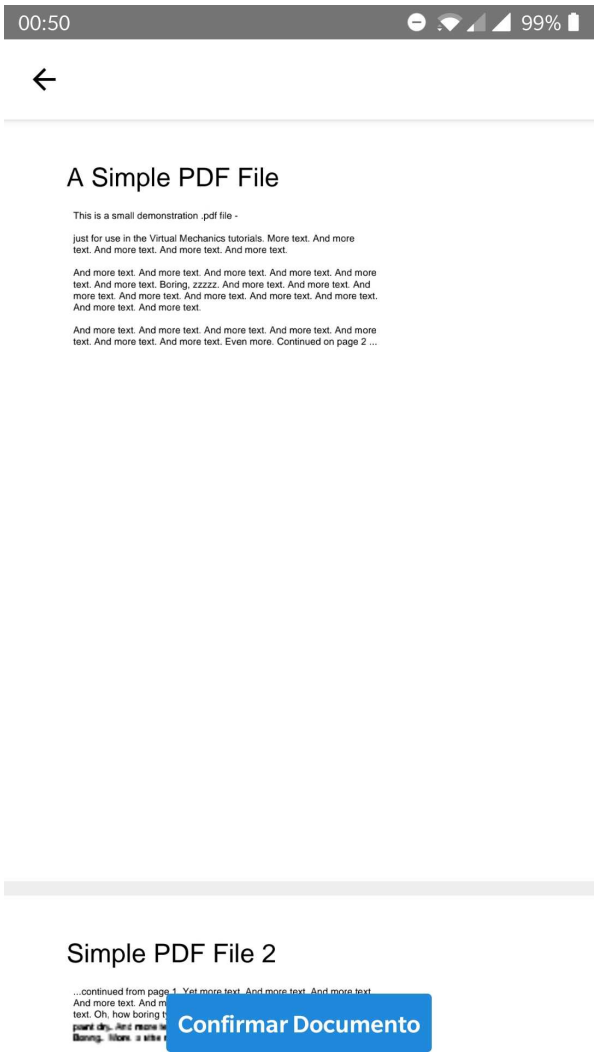


Figura 11: Ecrã de Visualização do Documento a Assinar

O ecrã presente na Figura 11 é apresentado ao utilizador após a escolha do documento a assinar. O utilizador tem que confirmar se o documento visualizado é o certo clicando no botão ”Confirmar Documento”.

Confirmação da Localização da Assinatura

Figura 12: Ecrã de Confirmação da Localização da Assinatura no Documento

O ecrã presente na Figura 12 é o ecrã responsável por indicar onde a assinatura é colocado caso a assinatura seja visível, sendo possível escolher a página e a localização na página onde a assinatura é colocada. Este ecrã não foi implementado, logo a página e a localização da assinatura na página são fixas.

Ecrã de Autenticação

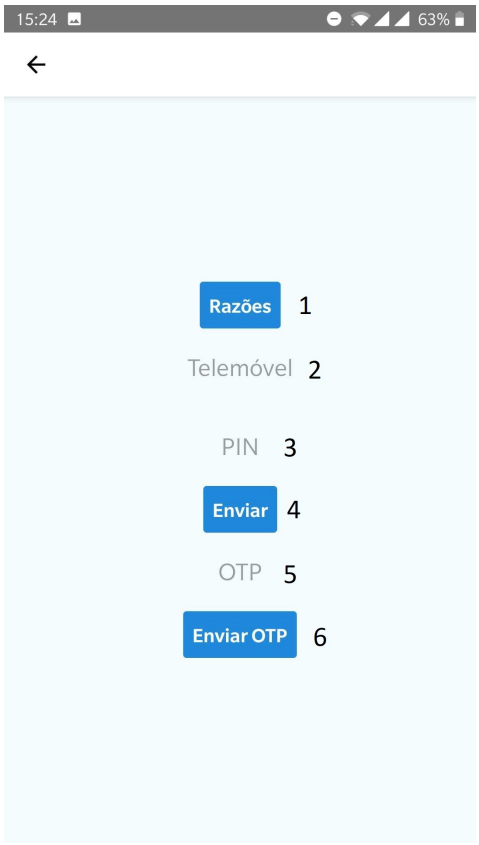


Figura 13: Ecrã de Autenticação com o SCMD

O ecrã presente na Figura 13 é o ecrã de Autenticação com o SCMD e Submissão do OTP para o SCMD. Estas duas operações foram colocadas no mesmo ecrã por razões de teste, mas idealmente seriam em ecrã separados.

Antes de submeter as credenciais de autenticação, é possível abrir o Menu de Razões de assinatura no botão "Razões" (1), para escolher as razões a colocar com a assinatura.

No campo "Telemóvel" (2) é inserido o número de telemóvel de autenticação associado à CMD do utilizador.

No campo "PIN" (3) é inserido o PIN de assinatura associado à CMD do utilizador.

Após a inserção dos dois parâmetros de autenticação, no botão "Enviar" (4) é iniciado o processo de assinatura, utilizando as Razões escolhidas, o número de Telemóvel e PIN inseridos nos campos e o documento escolhido anteriormente.

Havendo sucesso no inicio do processo de assinatura, o utilizador recebe um OTP por SMS ou *Push Notification* para ser inserido no campo "OTP" (5), para este ser enviado, clicando no botão "Enviar OTP" (6), assim concluindo o processo de assinatura.

Razões de Assinatura

Figura 14: Menu de escolha das Razões de Assinatura

O ecrã presente na Figura 14 contém o menu das Razões de Assinatura, especificadas na POL#16 [4]. Este menu pode ser aberto clicando no botão ”Razões” do ecrã descrito anteriormente.

O utilizador pode selecionar as razões que quiser, não sendo obrigatório selecionar qualquer razão de assinatura.

Ecrã Final

Figura 15: Ecrã Final, de conclusão do processo de assinatura

O ecrã presente na Figura 15 é o ecrã Final de conclusão do processo de assinatura. Ao clicar no botão central com uma representação de uma casa, o utilizador é redirecionado para o ecrã inicial, para assim poder voltar a assinar um documento.

4.2.2 *Módulos React Native*

Async Storage

Módulo de *React Native* que permite à aplicação ter armazenamento de dados persistente. É utilizado nas configurações, como guardar o perfil de assinatura *PAdES-Baseline* escolhido e o servidor de *TimeStamp* utilizado. [5]

React Native Document Picker

Módulo de *React Native* responsável por escolher o documento no inicio do processo de assinatura. Este utiliza o explorador de ficheiros nativo do *Android* para selecionar um e só um documento, que obrigatoriamente tem que ser PDF, outra qualquer extensão torna o documento impossível de selecionar. [7]

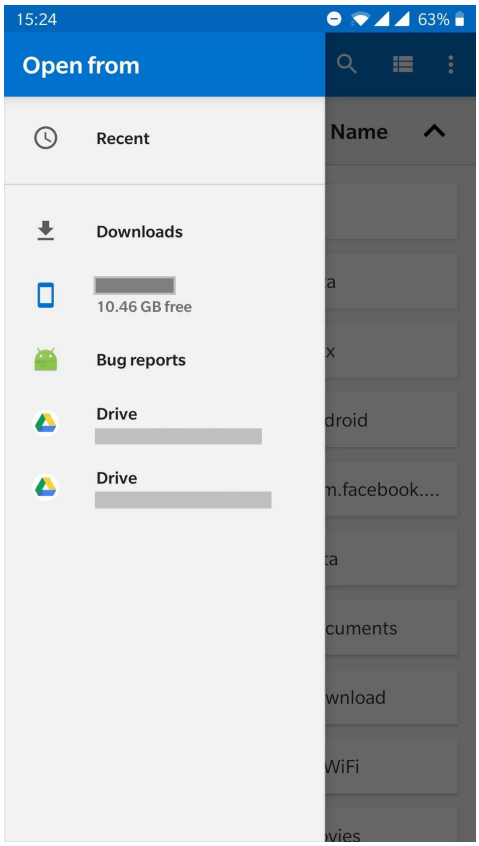


Figura 16: Interface do módulo React Native Document Picker

Ao selecionar o documento no ecrã presente na Figura 16, este módulo devolve todas as informações deste, incluindo o caminho do sistema de ficheiro para o módulo React Native PDF o poder apresentar.

React Native PDF

Módulo de *React Native* responsável por apresentar o PDF ao utilizador no ecrã de Visualização do Documento. Este utiliza o path proveniente da escolha do documento com o módulo React Native Document Picker para ler e apresentar o documento ao utilizador. [13]

4.2.3 *Comunicação com CMD*

Métodos de comunicação com o Serviço Chave Móvel Digital desenvolvidos utilizando os módulos nativos de *React Native*, que utilizam a API nativa de Android. Estes fazem parte do processo de assinatura e estão conforme a especificacção CMD [8], sendo os métodos de Obter Certificados e de Iniciar o Processo de Assinatura parte da Preparação da Assinatura e o método de Validar o OTP parte da Finalização da Assinatura.

Obter Certificados

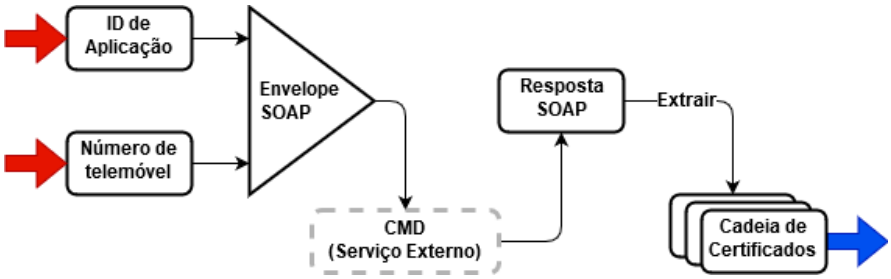


Figura 17: Pedido de Certificados do Assinante ao Serviço CMD

O método de Obter Certificados, representado na Figura 17, tem como parâmetros de entrada o identificador único da aplicação, definido pela Agência para a Modernização Administrativa, I.P. (AMA) e o número de telemóvel do assinante, utilizado na autenticação. Estes parâmetros são colocados num envelope Simpe Object Access Protocol (SOAP)[14] que é enviado por Hypertext Transfer Protocol Secure (HTTPS) para o Serviço Chave Móvel Digital. Como resposta, é obtida a cadeia de certificados do assinante.

Iniciar Processo de Assinatura

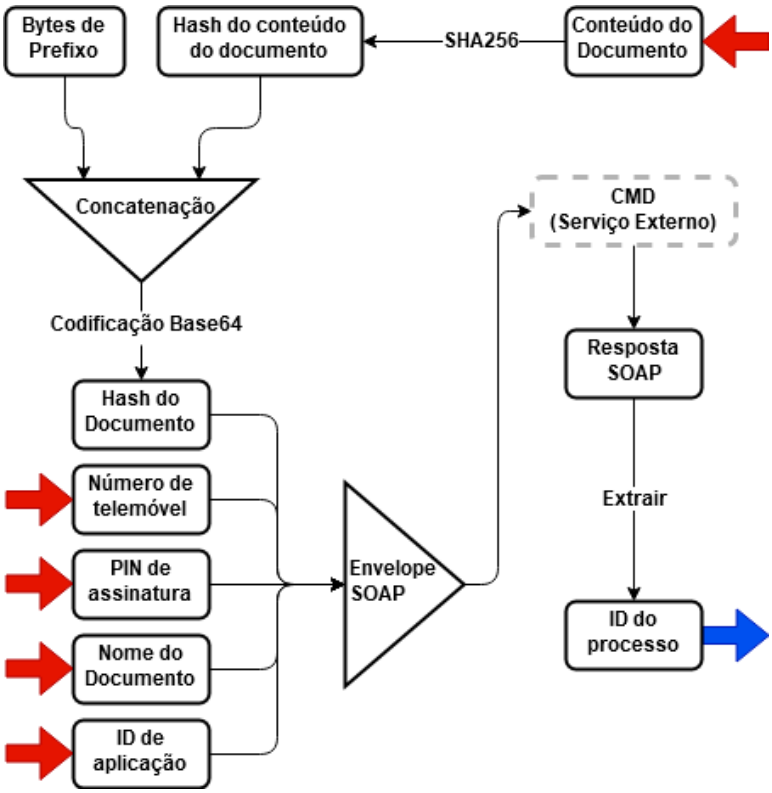


Figura 18: Pedido de Inicio de Processo de Assinatura ao Serviço CMD

O método de iniciação do processo de assinatura, representado na Figura 18, tem como parâmetros de entrada o conteúdo a assinar, o número de telemóvel do assinante, o PIN de assinatura, o nome do documento a assinar e o identificador único da aplicação, definido pela AMA.

Começa por preparar a *hash* do conteúdo a assinar. Para isto, é aplicada uma função de *hash* SHA256 ao conteúdo a assinar recebido, e é concatenado a um prefixo de *bytes* ('3031300d060960864801650304020105000420'). A concatenação deste não estava presente na especificação do serviço, tendo sido necessário contactar os desenvolvedores do SCMD, se não o conteúdo não seria gerado corretamente e por sua vez o processo não concluiria com sucesso. O prefixo de bytes tem como objetivo identificar qual a função *hash* utilizada (SHA256), para no Hardware Security Module (HSM) que vai efetuar a assinatura, permitir identificar que a assinatura a efetuar é *RSA with SHA256*(i.e, a *hash* recebida é SHA256). Por fim, a *hash* é codificada em *base64* para poder ser colocada no envelope SOAP.

Obtendo a *hash* do documento, esta é colocada em conjunto com o número de telemóvel, PIN de assinatura, nome do documento e identificador da aplicação num envelope SOAP,

sendo este enviado por HTTPS para o Serviço Chave Móvel Digital. Como resposta é obtido um identificador do processo, para ser submetido no próximo método Validar OTP.

Validar OTP

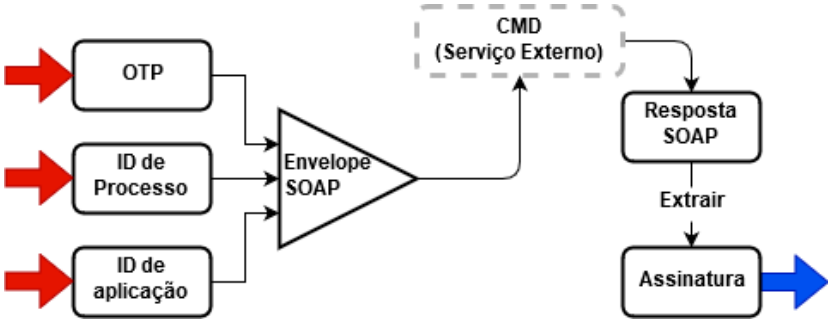


Figura 19: Pedido de Validação de OTP ao Serviço CMD

O método de validação do OTP, representado na Figura 19, tem como parâmetros o OTP recebido pelo utilizador por SMS ou *Push Notification*, o identificador do processo recebido como resposta no método de iniciação do processo de assinatura e o identificador único da aplicação, definido pela AMA.

Os parâmetros são colocados num envelope SOAP, que é enviado por HTTPS para o Serviço Chave Móvel Digital. Na resposta obtida contém a assinatura do documento, que está codificada em base64.

4.2.4 *Assinatura*

Considerando que a aplicação foi desenvolvido utilizando a *framework React Native*, a comunicação da interface com os módulos nativos é unidirecional, isto é, como a funcionalidade de assinatura utiliza os módulos nativos de *React Native*, ao chamar um método nativo na interface *javascript*, o único envio de informação acontece na chamada de função com o envio dos parâmetros e no fim do processo com a informação de retorno do método nativo, não sendo possível comunicação de informação enquanto o método nativo está a decorrer. [12]

Os métodos pertencentes ao processo de assinatura são desenvolvidos utilizando o *software* IText7 [3]. O processo original do IText7 de assinar um documento utilizando uma assinatura externa gerada em servidor não foi feito a pensar na existência de passos intermédios de autenticação, como no Serviço Chave Móvel Digital. Se este processo de assinatura fosse utilizado no desenvolvimento de outro projeto em que o processo pudesse ser interrompido para haver inserção de informação por parte do utilizador, não existiria problema, pois bastaria aguardar pela inserção do OTP recebido e depois continuar o pro-

cesso de assinatura. Mas como em *React Native* a comunicação entre a camada de Interface de utilizador e os módulos nativos só existe com os parâmetros da chamada do método nativo e com o retorno no fim da execução do mesmo, não é possível iniciar o processo de preparação da assinatura e o utilizador inserir o OTP para o processo terminar.

Então o processo de assinatura foi dividido em duas partes, sendo criado uma classe nova que estende a classe original de assinatura [10], que implementa o processo original de assinatura do IText, mas dividido em duas partes:

- A preparação da assinatura, que prepara o documento para poder ser assinado e inicia o processo de assinatura com o Serviço Chave Móvel Digital.
- A finalização da assinatura, que termina o processo de assinatura com o Serviço Chave Móvel Digital e coloca a assinatura no documento. Como existe a possibilidade de assinar com quatro perfis diferentes de *PAdES-Baseline*, existe um método para cada um de finalização de assinatura.

Foi necessário implementar a adição de razões de assinatura presentes na seccção 3.2.2 da POL#16 [4] (denominados por *Commitment Type Indications*, definido no RFC 5126 secção 5.11.1[32]), pois o IText7 não suporta a colocação deste campo na assinatura do documento.

Para inserir as razões de assinatura, foi necessário adicionar ao método, desenvolvido pelo IText, de geração do documento a assinar a colocação dos identificadores das razões de assinatura nos atributos[1] da assinatura a partir do identificador "id-aa-ets-commitmentType", que identifica o objeto, segundo o RFC 5126, presente na Figura 20.

```
CommitmentTypeIndication ::= SEQUENCE {
     commitmentTypeId CommitmentTypeIdentifier,
     commitmentTypeQualifier SEQUENCE SIZE (1..MAX) OF
            CommitmentTypeQualifier OPTIONAL
}
```

Figura 20: Objeto Commitment Type Indication especificado no RFC 5126

Por fim, foi implementada a adição do numero de série do certificado de assinatura aos atributos de assinatura seguindo a especificação do objeto identificado por "id-aa-signingCertificateV2"no RFC 5035 [35].

O IText já continha uma implementação deste objeto, e por sua vez também do sub-objeto que contém o numero de sério do certificado de assinatura, Figura 21, mas só continha a *hash* do certificado e o identificador da função de *hash*, pois o atributo de número de série do certificado de assinatura é opcional. Como a plataforma de validação das assinaturas PDF Advanced Electronic Signatures (PAdES) utilizada na secção 5.2 acusava a falta deste atributo, foi então implementada a adição deste atributo à assinatura.

1 http://javadox.com/org.bouncycastle/bcprov-jdk15on/1.51/org/bouncycastle/asn1/cms/Attribute.html

```
ESSCertIDv2 ::=  SEQUENCE {
    hashAlgorithm           AlgorithmIdentifier
            DEFAULT {algorithm id-sha256},
    certHash                 Hash,
    issuerSerial             IssuerSerial OPTIONAL
}
```

Figura 21: Objeto ESSCertIDv2 especificado no RFC 5035

Preparação da Assinatura

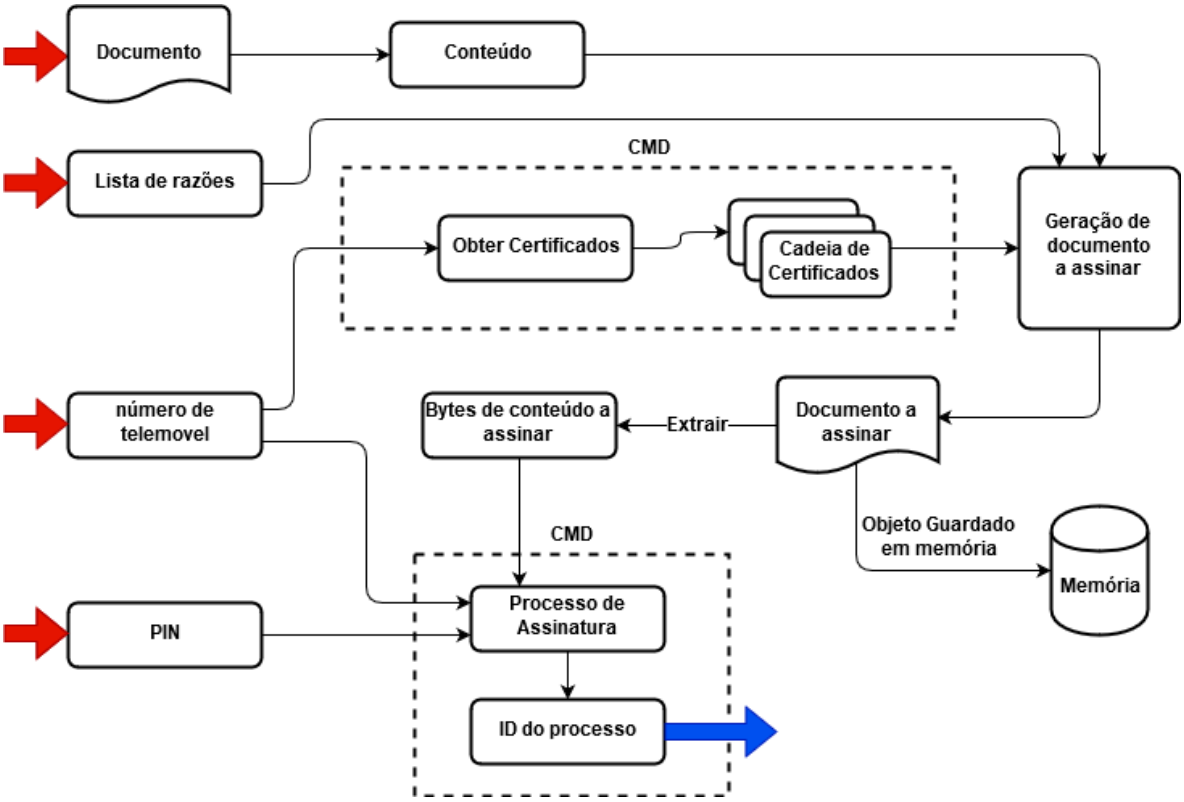


Figura 22: Inicio do processo de assinatura

O método de preparação da assinatura, representado na Figura 22, tem como parâmetros o documento a assinar, a lista de razões da assinatura, o número de telemóvel e PIN do utilizador para a assinatura com o SCMD.

Inicialmente é preciso gerar um documento pronto a colocar a assinatura, para isto é necessário: extrair o conteúdo a assinar, e obter a cadeia de certificados do assinante, a partir do Serviço Chave Móvel Digital. Extraindo o conteúdo a assinar, obtendo a cadeia de certificados e em junção com as lista de razões da assinatura, é gerado um novo documento com o conteúdo extraído, com o espaço para a assinatura alocado, com os certificados do assinante e a lista de razões inseridas. Este documento é colocado em memória, para ser

utilizado no método de Finalização da Assinatura e também é serializado para ser assinado pelo Serviço Chave Móvel Digital.

Tendo o conteúdo a assinar, este em conjunto com o número de telemóvel e o PIN de autenticação, são enviados para o Serviço Chave Móvel Digital para assim iniciar o processo de assinatura. Como resposta à iniciação do processo de assinatura, é devolvido um identificador do processo, que é utilizado para dar continuidade na Finalização da Assinatura.

Finalização da Assinatura PAdES-B-B/T

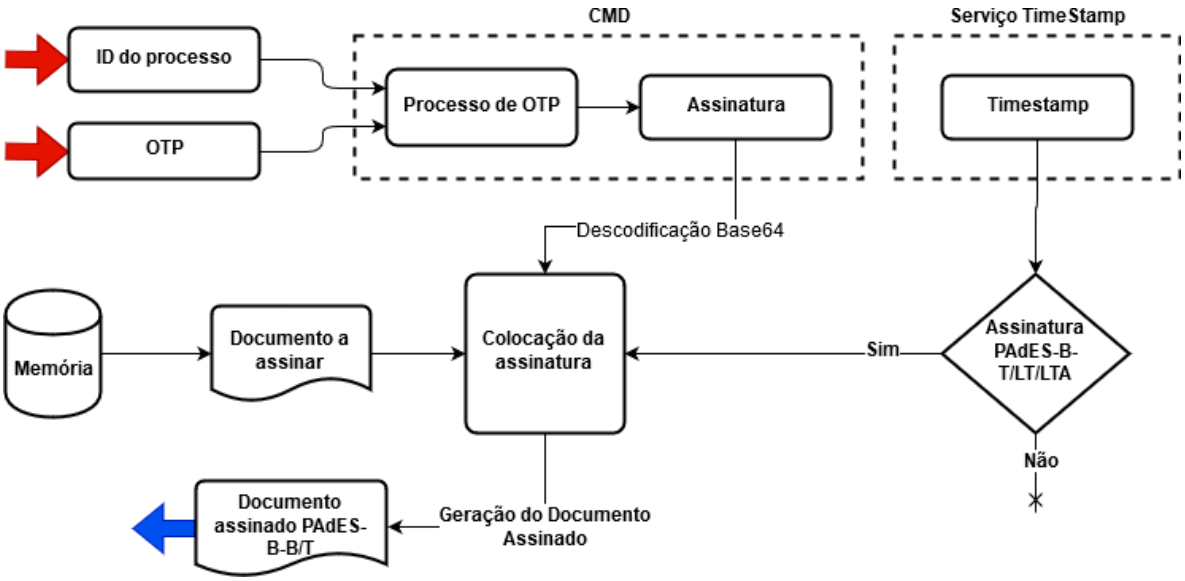


Figura 23: Assinatura PAdES-B-B ou PAdES-B-T

O método de finalização de uma assinatura PAdES-B-B ou T, representado na Figura 23, tem como parametros o identificador do processo, resultado do método de Preparação da Assinatura e o OTP recebido pelo utilizador por SMS ou *Push Notification*. Estes parâmetros são enviados para o SCMD para assim validar o OTP e finalizar o processo de assinatura, sendo devolvida a assinatura a colocar no documento.

Para gerar o documento assinado, utilizando o documento com o espaço alocado para a assinatura gerado no método de Preparação da Assinatura, a assinatura recebida do serviço CMD é descodificada de base64 para bytes e é colocada no espaço alocado do documento, sendo também colocada visível no canto superior direito da primeira página, assim resultando num novo documento com uma assinatura PAdES-B-B, conforme a secção 2.5.2. No caso da assinatura ser PAdES-B-T, um *timestamp* é pedido ao serviço externo definido pelo utilizador ou predefinido do Cartão de Cidadão e este é colocado com a assinatura, conforme a secção 2.5.3.

Finalização da Assinatura PAdES-B-LT

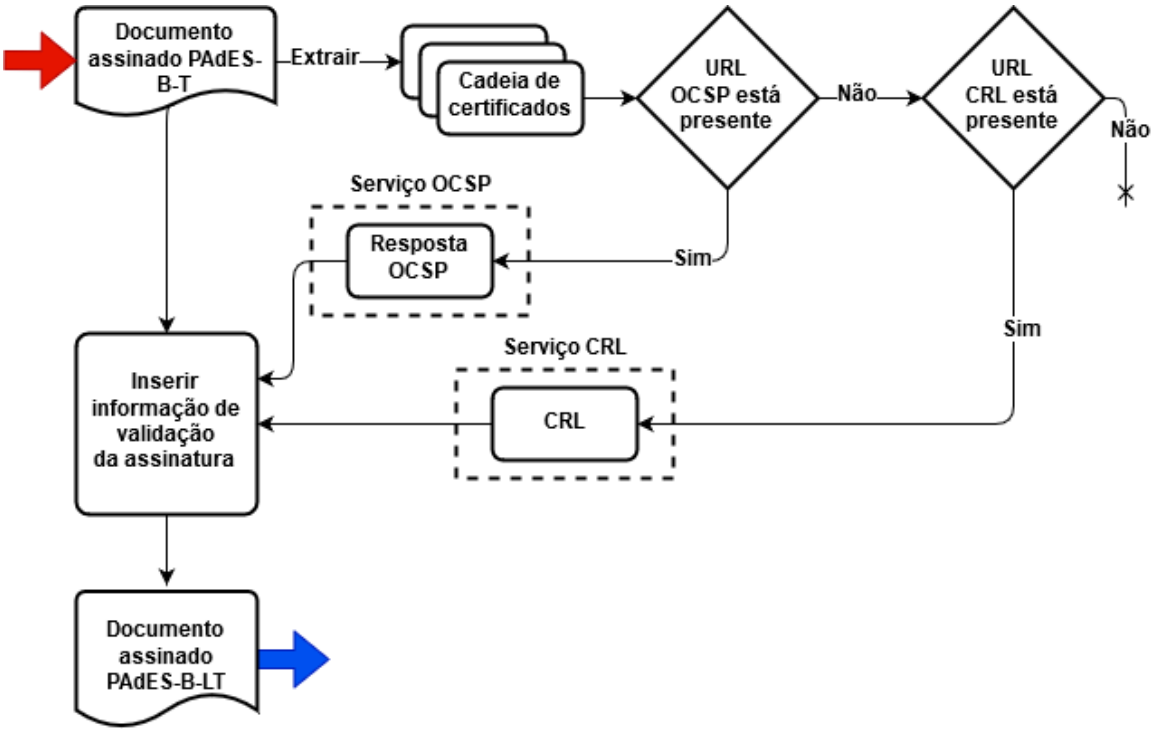


Figura 24: Assinatura PAdES-B-LT

O método de finalização de uma assinatura PAdES-B-LT, representado na Figura 24, é uma extensão do método de finalização de uma assinatura PAdES-B-T, logo o parâmetro de entrada é o documento assinado com PAdES-B-T.

Do documento assinado, é extraída a cadeia de certificados da assinatura, para se poder obter o URL dos serviços de validação dos certificados.

No caso de o URL do serviço Online Certificate Status Protocol (OCSP) estar presente, um pedido é feito ao serviço e a resposta OCSP é colocada no documento, assim confirmando que o certificado é válido no momento em que o pedido ao serviço OCSP é feito. Mas caso o URL do serviço OCSP não esteja presente e o URL do serviço da Certificate Revogation List (CRL) esteja presente, é feito um pedido ao serviço e a CRL é colocada no documento, assim confirmando que o certificado é válido no momento de emissão da CRL.

Assim é colocada a informação de validação do certificado de assinatura, preferindo sempre a resposta OCSP a CRL e o documento assinado com PAdES-B-LT é gerado, conforme a secção 2.5.4.

Finalização da Assinatura PAdES-B-LTA

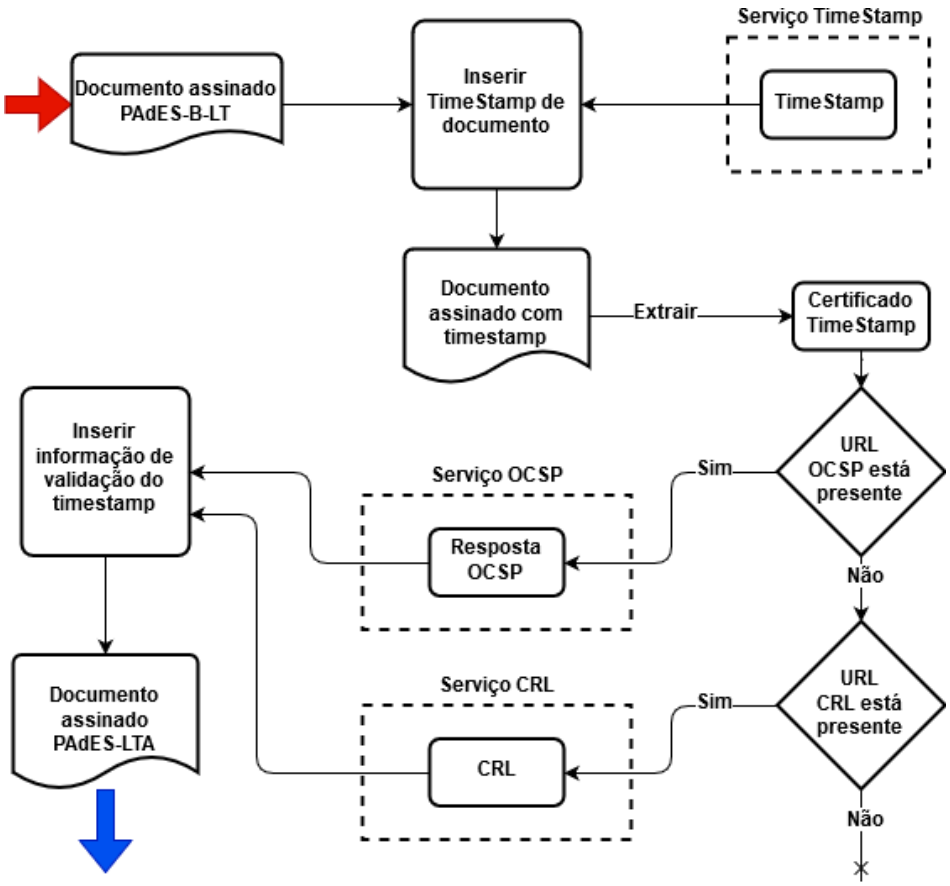


Figura 25: Assinatura PAdES-B-LTA

O método de finalização de uma assinatura PAdES-B-LTA, representado na Figura 25, é uma extensão do método de finalização de uma assinatura PAdES-B-LT, logo o parâmetro de entrada é o documento assinado com PAdES-B-LT.

Inicialmente, um *timestamp* é adicionado ao documento, gerando um documento assinado com *timestamp*. Deste é extraido o certificado de validação do *timestamp* para proceder à requisição da informação de validação do certificado. No caso de o URL do serviço OCSP estar presente, um pedido é feito ao serviço e a resposta OCSP é colocada no documento, assim confirmando que o certificado do *timestamp* é válido no momento em que o pedido ao serviço OCSP é feito. Mas caso o URL do serviço OCSP não esteja presente e o URL do serviço da CRL esteja presente, é feito um pedido ao serviço e a CRL é colocada no documento, assim confirmando que o certificado do *timestamp* é válido no momento de emissão da CRL.

Assim é colocada a informação de validação do certificado do *timestamp*, preferindo sempre a resposta OCSP a CRL e o documento assinado com PAdES-B-LTA é gerado, conforme a secção 2.5.5.

4.3 RESULTADOS

Uma importante parte para tornar possível o desenvolvimento desta dissertação foi a implementação dos módulos nativos de assinatura e comunicação com o serviço Chave Móvel Digital. Estes em conjunto com a interface desenvolvida e os módulos importados, constroem uma aplicação móvel que assina localmente um documento PDF com os perfis PAdES-Baseline. Mas mais importante é o facto de que estes módulos nativos, como estão desenvolvidos em Java, podem ser exportados e utilizados em qualquer software desenvolvido que utilize Java e suporte as bibliotecas utilizadas.

Com isto, as funcionalidades enunciadas na secção 3.2 que foram desenvolvidas, foram as seguintes:

1. Apresenta o(s) ficheiro(s) a assinar de acordo com a política What You See Is What You Sign (WYSIWYS), diretamente na aplicação ou em aplicação externa (cf. secção 3.2.4 da POL#16 [4]);
2. Guarda do ficheiro assinado na app;
3. Exige a introdução do PIN da CMD por cada assinatura ou *batch* de assinatura efetuada, tendo o utilizador a possibilidade de o guardar o número de telemóvel na app;
4. Não apresenta, visível para o utilizador, os dígitos do PIN da CMD;
5. Não transmite para qualquer outra aplicação dados do PIN da CMD;
6. Apresenta sempre o(s) documento(s) a assinar ao utilizador;
7. Não partilha as credenciais de acesso (APPLICATION_ID) da aplicação ao serviço CMD assinatura, com qualquer outra entidade ou aplicação;
8. Utiliza canal seguro (Transport Layer Security (TLS)) para transmissão de credenciais do utilizador a serviço CMD assinatura;
9. Apresenta os dados a serem assinados no formato XML (que contém o *hash* do documento a assinar em formato byte array) – também designado por representação dos dados a serem assinados (DTBS/R) – ao servidor de assinatura CMD da AMA, garantindo que corresponde aos dados/documento a assinar apresentado pelo/ao assinante (cf. secção 3.1.2 da POL#16 [4]);

10. Guarda a assinatura embebida no documento PDF assinado com os dados assinados (cf. secção 3.1.3 da POL#16 [4]);

11. A guarda da assinatura é efetuada no formato, perfil e nível de assinatura definida pela aplicação, permitindo a aplicação o seguinte tipo de assinaturas: PAdES_BASELINE_B, PAdES_BASELINE_T, PAdES_BASELINE_LT e PAdES_BASELINE_LTA;

12. O tipo de assinatura por omissão é a PAdES_BASELINE_T, podendo o utilizador ir ao menu de configuração e alterar o tipo de assinatura por omissão;

13. É adicionada a referência à Política CMD de Assinatura Qualificada (indicando o seu OID 2.16.620.2.1.2.2), de modo a permitir que partes confiantes e outras pessoas interessadas possam encontrar informação sobre as políticas e práticas seguidas na aposição da assinatura. (cf. secção 3.1.3 da POL#16 [4]);

14. Com a assinatura dos dados/documento, a aplicação deve permitir que o assinante associe ao campo Razão/Reason (ou *commitment*), um (ou vários) dos seguintes tipo de compromissos (e respetivo OID), de modo a contextualizar (e desambiguar) o propósito e significado da assinatura, assim como a natureza da responsabilidade assumida: Prova de origem / *Proof of origin*, Prova de aprovação / *Proof of approval*, Prova de criação / *Proof of creation*, Autenticação de dados / *Data Authentication*, Autenticação de Entidade / *Entity Authentication*, Autoria / *Authorship*, Revisão / *Review*, Cópia / *Copy*, Testemunha de assinatura / *Signature Witness*, Vinculação ao conteúdo assinado / *Bound to data signed*, Aprovação intermédia / *Intermediate approval*. (cf. secção 3.2.2 da POL#16 [4]);

15. A aplicação permite que o assinante valide que os dados de identificação do documento, a assinar, recebidos na mensagem SMS/*Push notification*, são os mesmos que lhe são apresentados no user interface da aplicação. (cf. secção 3.2.4 da POL#16 [4]);

4.4 Sumário

Foi optado para esta dissertação, o desenvolvimento de uma aplicação multi-plataforma, tirando a necessidade de desenvolver duas aplicações diferentes para *Android* e iOS, tornando o desenvolvimento da aplicação mais fácil e rápido.

Foram ponderadas várias *frameworks* de desenvolvimento de aplicações multi plataforma, mas *React Native* foi a escolhida, pois além de ser a mais popular, é a mais próxima de um desenvolvimento nativo, mantendo a abstração do desenvolvimento característico das aplicações multi plataforma.

Sabendo a *framework* de desenvolvimento, foi necessário encontrar uma implementação de assinaturas digitais PAdES-Baseline que fosse suportada por ambas as plataformas. Tal

não aconteceu, acabando por ser tomada a decisão de desenvolver só em *Android* utilizando módulos nativos de *React Native* desenvolvidos em *Java* com o *software* IText 7, para implementar o processo de assinatura utilizando os perfis PAdES-Baseline.

Para a implementação das funcionalidades de assinatura utilizando o IText ser possível, foi necessário adaptar o processo original de assinatura utilizando uma assinatura gerada externamente, pois este não estava preparado para ser utilizado com um serviço externo que requer um segundo fator de autenticação e consequentemente interrupção do processo de assinatura, para gerar a assinatura do lado do serviço Chave Móvel Digital e esta ser devolvida. Além disso, mesmo existindo possíveis soluções que o desenvolvimento em *Java* dispõe para informação, como o segundo fator de autenticação, ser introduzida pelo utilizador, estas não são possíveis de utilizar, pois o fluxo de informação entre a interface onde o método nativo é chamado, e o módulo nativo onde o método é executado é unidirecional, tornando impossível a interrupção do processo para ser introduzido o segundo fator de autenticação.

Com isto, foi necessário separar o processo de assinatura em duas partes, uma que prepara o documento para ser assinado pelo serviço Chave Móvel Digital e outra que finaliza o processo de assinatura de acordo com o perfil PAdES-Baseline escolhido, recebendo a assinatura do serviço Chave Móvel Digital e inserindo-a no documento.

Os módulos nativos desenvolvidos, em conjunto com a Interface desenvolvida utilizando a biblioteca *javascript* de *React Native*, formam uma aplicação que assina localmente documentos PDF com os perfis PAdES-Baseline, utilizando o serviço Chave Móvel Digital para gerar a assinatura.

5

TESTES

Neste capítulo serão demonstradas as operações de assinatura com os perfis PAdES-Baseline da aplicação desenvolvida, como feitos testes e medições como meio de comparação com a aplicação Autenticação.gov.

5.1 SETUP DOS TESTES

Terminado o desenvolvimento da aplicação, existe a necessidade de demonstrar e testar as suas funcionalidades. Para isto, será primeiramente feita uma demonstração da assinatura e após a demonstração, um conjunto de testes com os vários perfis de assinatura, comparando-os com resultados da aplicação Auntenticação.gov, pois esta é a única aplicação pública que assine documentos PDF utilizando o Serviço Chave Móvel Digital (SCMD), conforme o website da CMD[1].

Para a demonstração e testes serão utilizados os seguintes dispositivos:

- Aplicação Móvel Android: Smartphone OnePlus 5, 64GB de memória interna, 6GB RAM, Snapdragon 835 Octa-Core 2.45Ghz;
- Aplicação Autenticação.gov: Portátil MSI, 512GB SSD, 8GB RAM, i7-8750H Hexa-core 2.2GHz.

5.1.1 *Demonstração*

Para a demonstração, foi gerado um documento Portable Document Format (PDF) em branco no Microsoft Word. Este documento foi assinado com os quatro perfis PAdES-Baseline individualmente.

De maneira a demonstrar que os documentos estão assinados:

- uma assinatura visual é colocada no documento a comprovar que a assinatura foi colocada;

1 https://www.autenticacao.gov.pt/cmd-assinatura

- cada documento é visualizado no *Adobe Reader*, sendo a assinatura validada também no Adobe Reader;
- cada documento é submetido à Digital Signature Services (DSS) *Demonstration WebApp*[2] da comissão Europeia, que valida a assinatura, indicando várias informações de validação, incluindo o perfil PDF Advanced Electronic Signatures (PAdES) utilizado na assinatura.
- cada documento é submetido ao *ETSI Signature Conformance Checker*[3], ferramenta da European Telecommunications Standards Institute (ETSI) que testa se as assinaturas colocadas num documento PDF estão em conformidade com os ETSI EN 319 142-1[25] e ETSI EN 319 142-2[26].

5.1.2 *Testes*

Para testar as capacidades da aplicação desenvolvida e demonstrar as mais valias em comparação com a aplicação Autenticação.gov, um conjunto de operações (assinar o mesmo documento com os vários perfis PAdES) foram testadas e métricas foram anotadas.

Quatro documentos com 2MB, 10MB, 50MB e 100MB respetivamente, foram assinados com cada perfil PAdES-Baseline utilizando a aplicação desenvolvida e assinados com o perfil PAdES-Básico com a aplicação Autenticação.gov.

Durante as operações de assinatura de aplicação desenvolvida, vários tempos foram registados:

- Tempo de preparação de assinatura;
- Tempo de finalização de assinatura;
- Tempo de preparação de assinatura com inclusão da comunicação com o SCMD;
- Tempo de finalização de assinatura com inclusão da comunicação com o SCMD;

Para obter estes valores, foram registados os valores de começo e fim das operações nos registos de depuração da aplicação desenvolvida.

Por motivos de comparação, os tempos de operação de assinatura da aplicação Autenticação.gov foram registados. Tanto o tempo de preparação de assinatura como o de finalização foram registados, ambos incluindo o pedido ao SCMD. Os tempos foram medidos com o uso de um cronometro, sendo valores menos precisos, devido à impossibilidade de aceder a registos de depuração temporais das operações.

2 https://ec.europa.eu/cefdigital/DSS/webapp-demo/validation
3 https://signatures-conformance-checker.etsi.org/pub/index.shtml

Além dos tempos de operação, também foram registados os tamanhos dos ficheiro assinados, assim comparando com o tamanho original do documento e a diferença de tamanho entre cada perfil de assinatura PAdES gerada pela aplicação desenvolvida e pela Auntenticação.gov.

Para comparar as assinaturas produzidas pela aplicação desenvolvida e a aplicação Auntenticação.gov, a assinatura utilizando o perfil PAdES-Baseline-B foi escolhida como termo de comparação, pois é a mais comparável à assinatura com o perfil PAdES-Básico.

5.2 RESULTADOS

5.2.1 *Demonstração*

Assinatura PAdES-B-B

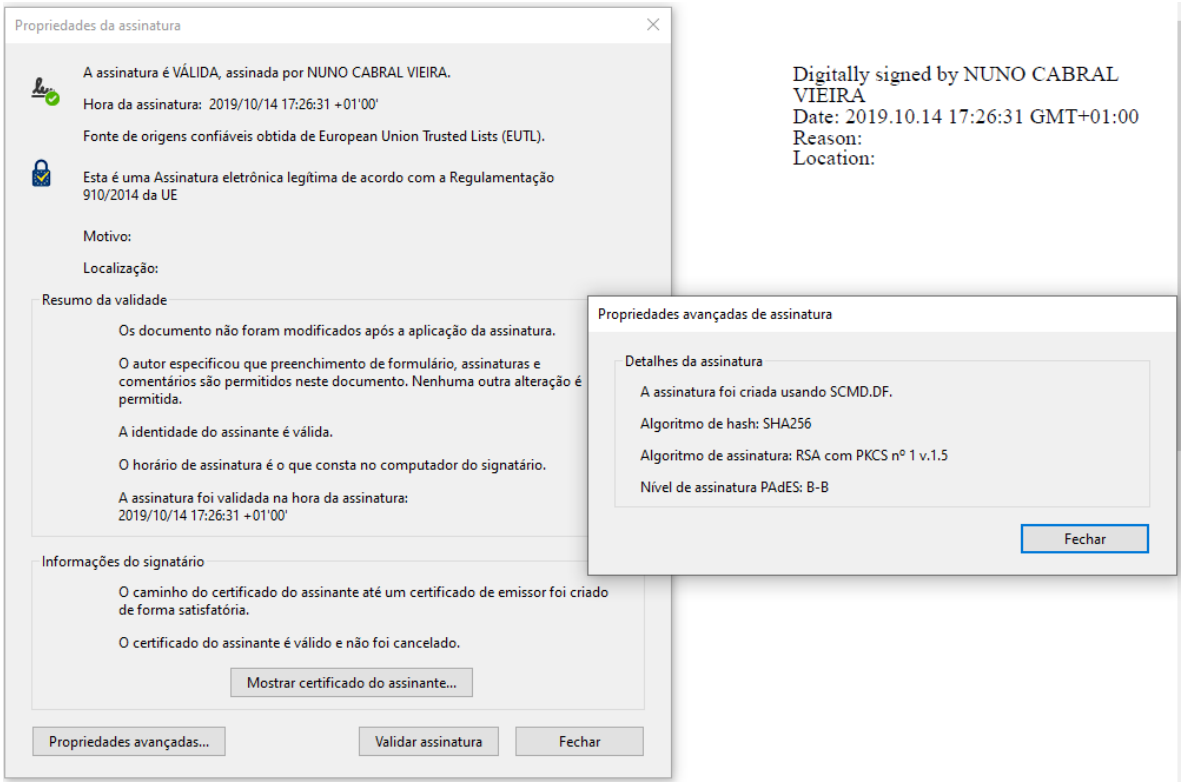


Figura 26: Assinatura PAdES-Baseline-B visializada no Adobe Reader, a sua validação e propriedades

O documento foi aberto no Adobe Reader, onde é possível visualizar a assinatura colocada no documento no canto superior direito, a comprovar que esta é visível, como é possível observar na Figura 26.

Na janela de Propriedades da assinatura (Figura 26) e no menu de Assinaturas (Figura 27) do documento assinado, é possível verificar que a assinatura é válida, o assinante e a hora a que esta foi criada. "Fonte de origens confiáveis obtida de European Union Trusted Lists (EUTL)" indica que o certificado de assinatura foi emitido por uma entidade autorizada pela União Europeia, de acordo com o regulamento eIDAS (910/2014 [20]).

De acordo com o campo "Resumo da validade" (Figura 26) é possível verificar que após a operação de assinatura feita pela aplicação desenvolvida, o documento não é alterado e que a partir do momento que foi assinado só é permitido adicionar formulários, outras assinaturas e comentários.

Na janela de Propriedades avançadas de assinatura (Figura 26), é possível verificar que a assinatura foi criada pela aplicação desenvolvida ("SCMD.DF"), que o algoritmo de geração da *hash* do conteúdo assinado foi SHA256, que o algoritmo de assinatura usado pelo Serviço Chave Móvel Digital foi RSA e que o perfil de assinatura usado foi de facto PAdES-B-B.

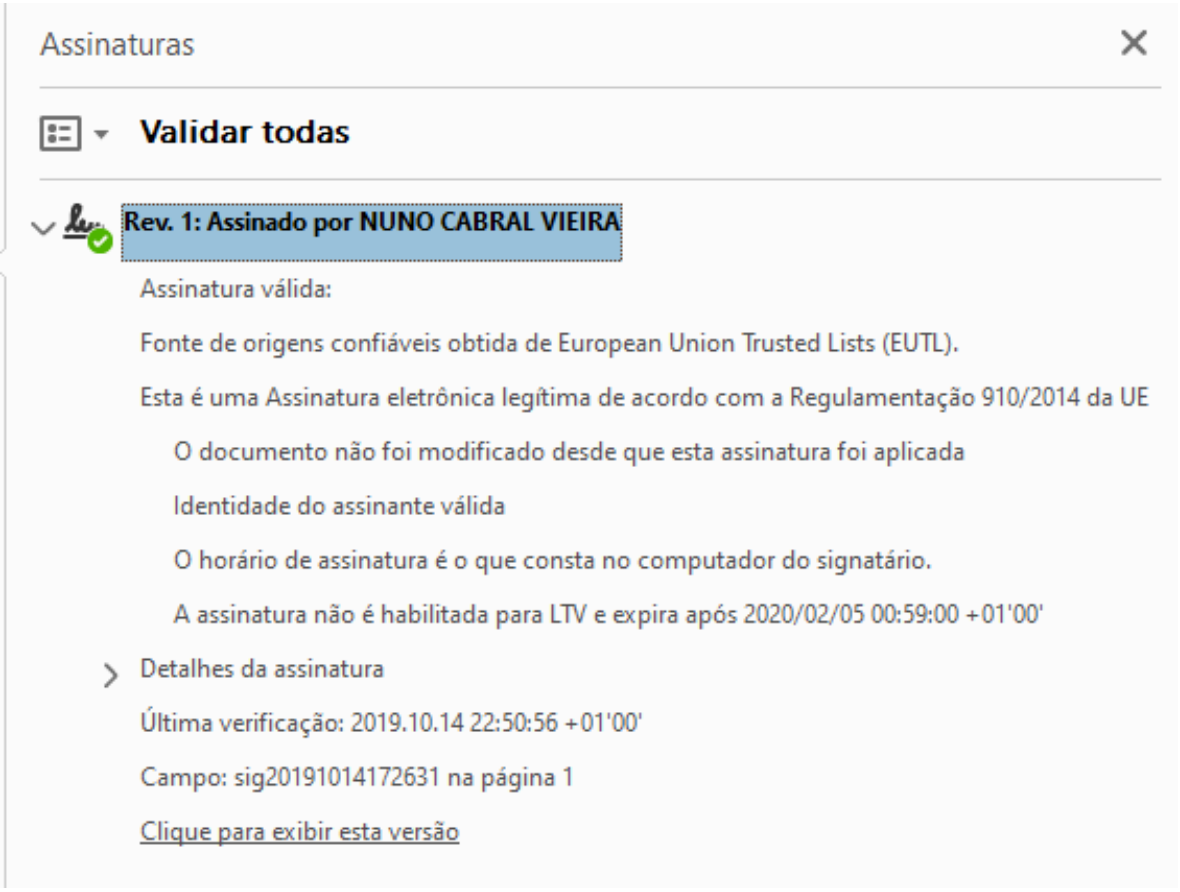


Figura 27: Painel de Assinaturas do documento assinado com assinatura PAdES-B-B

Como a assinatura é PAdES-B-B e é possível verificar na Figura 27, esta não é habilitada a *Long Term Validation*(LTV), i.e., capacidade da assinatura ser validada mesmo após o certi-

ficado de assinatura ter expirado, que é característica dos perfis PAdES-B-LT e LTA. Logo a assinatura expira na data de expiração do certificado.

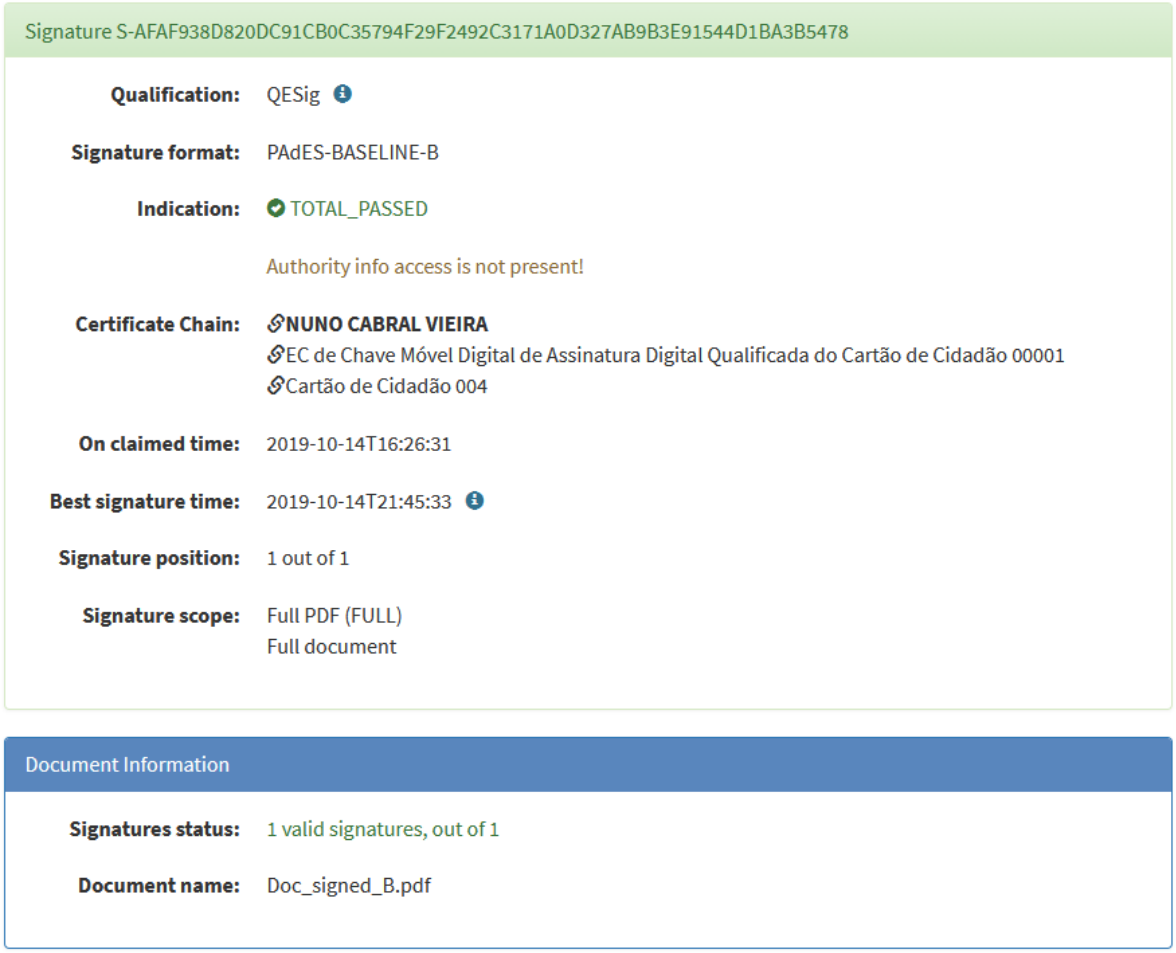


Figura 28: Validação da assinatura PAdE-Baseline-B na *WebApp* do DSSValidação da assinatura PAdES-Baseline-B na *WebApp* do DSS[4]

Como é possível verificar na Figura 28, a assinatura é uma assinatura electrónica qualificada e utiliza o perfil PAdES-Baseline-B.

Nesta plataforma de validação de assinaturas digitais, todas as assinaturas criadas com o uso do serviço Chave Móvel Digital apresentam o aviso "*Authority info access is not present*". Este aviso indica que o certificado de raiz da cadeia de certificados da assinatura, emitido pela entidade presente na EUTL, não contém uma referência para o URL do certificado da entidade de certificação que o emitiu, mas como o certificado está presente na EUTL, sendo confiado à partida, só aparece um aviso (RFC 5280 secção 4.2.2.1 [23]).

4 https://ec.europa.eu/cefdigital/DSS/webapp-demo/validation

Assinatura PAdES-B-T

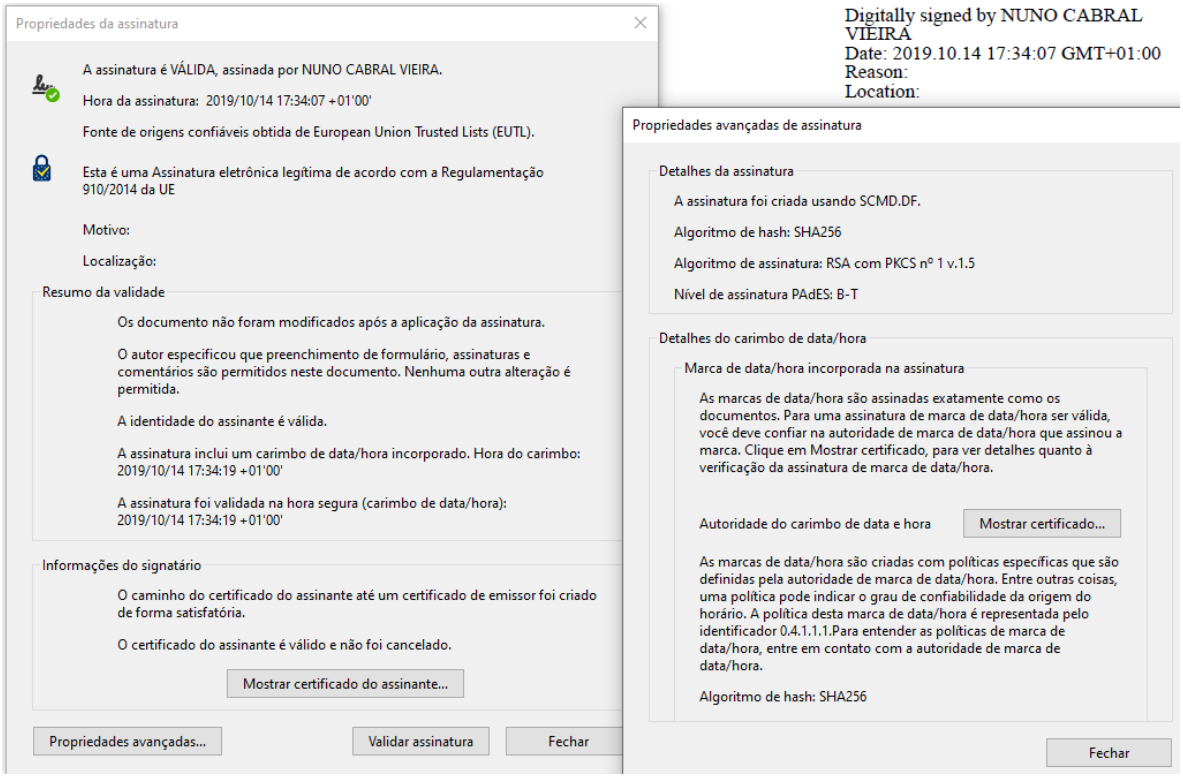


Figura 29: Assinatura PAdES-Baseline-T visializada no Adobe Reader, a sua validação e propriedades

O documento foi aberto no Adobe Reader, onde é possível visualizar a assinatura colocada no documento no canto superior direito, a comprovar que esta é visível, como é possível observar na Figura 29.

Na janela de Propriedades da assinatura (Figura 29) e no menu de Assinaturas (Figura 30) do documento assinado, é possível verificar que a assinatura é válida, o assinante e a hora a que esta foi criada. "Fonte de origens confiáveis obtida de European Union Trusted Lists (EUTL)" indica que o certificado de assinatura foi emitido por uma entidade autorizada pela União Euripeia, de acordo com o regulamento eIDAS (910/2014 [20]).

De acordo com o campo "Resumo da validade" (Figura 29) é possível verificar que após a operação de assinatura feita pela aplicação desenvolvida, o documento não é alterado, que a partir do momento que foi assinado só é permitido adicionar formulários, outras assinaturas e comentários e que de facto foi adicionado à assinatura um *timestamp* (carimbo de data/hora) a comprovar a hora de assinatura.

Na janela de Propriedades avançadas de assinatura (Figura 29), é possível verificar que a assinatura foi criada pela aplicação desenvolvida ("SCMD.DF"), que o algoritmo de geração da *hash* do conteúdo assinado foi SHA256, que o algoritmo de assinatura usado pelo Serviço

Chave Móvel Digital foi RSA, que o perfil de assinatura usado foi de facto PAdES-B-T e os detalhes do *timestamp* incluído na assinatura (carimbo de data/hora).

Figura 30: Painel de Assinaturas do documento assinado com assinatura PAdES-B-T

Como a assinatura é PAdES-B-T e é possível verificar na (Figura 30), esta não é habilitada a *Long Term Validation*, i.e., capacidade da assinatura ser validada mesmo após o certificado de assinatura ter expirado, que é caraterística dos perfis PAdES-B-LT e LTA. Logo a assinatura expira na data de expiração do certificado.

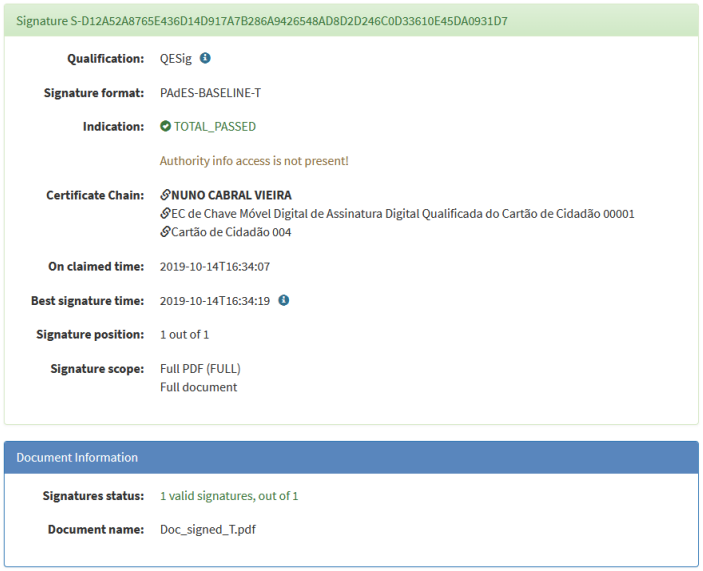


Figura 31: Validação da assinatura PAdES-Baseline-T na *WebApp* do DSS

Como é possível verificar na Figura 31, a assinatura é uma assinatura electrónica qualificada e utiliza o perfil PAdES-Baseline-T.

Assinatura PAdES-B-LT

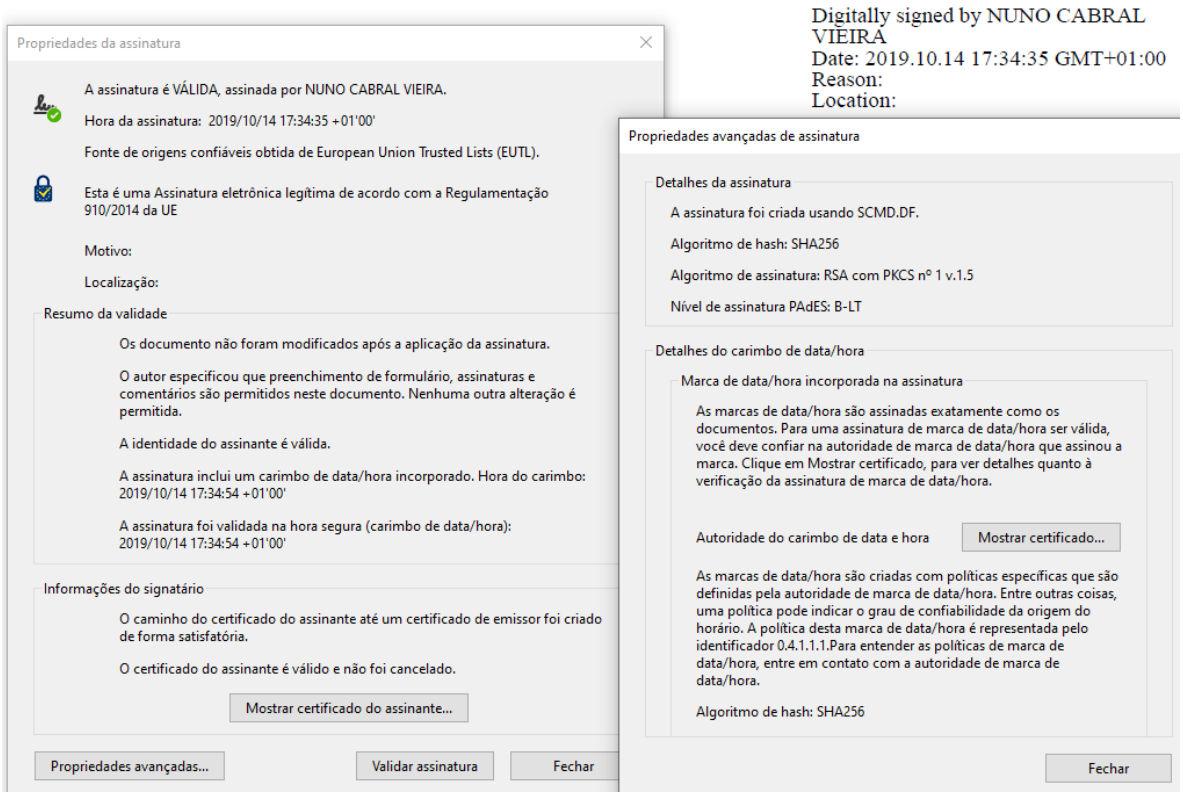


Figura 32: Assinatura PAdES-Baseline-LT visializada no Adobe Reader, a sua validação e propriedades

O documento foi aberto no Adobe Reader, onde é possível visualizar a assinatura colocada no documento no canto superior direito, a comprovar que esta é visível, como é possível observar na Figura 32.

Na janela de Propriedades da assinatura (Figura 32) e no menu de Assinaturas (Figura 33) do documento assinado, é possível verificar que a assinatura é válida, o assinante e a hora a que esta foi criada. "Fonte de origens confiáveis obtida de European Union Trusted Lists (EUTL)" indica que o certificado de assinatura foi emitido por uma entidade autorizada pela União Europeia, de acordo com o regulamento eIDAS (910/2014 [20]).

De acordo com o campo "Resumo da validade" (Figura 32) é possível verificar que após a operação de assinatura feita pela aplicação desenvolvida, o documento não é alterado, que a partir do momento que foi assinado só é permitido adicionar formulários, outras assinaturas e comentários e que de facto foi adicionado à assinatura um *timestamp* (carimbo de data/hora) a comprovar a hora de assinatura.

Na janela de Propriedades avançadas de assinatura (Figura 32), é possível verificar que a assinatura foi criada pela aplicação desenvolvida ("SCMD.DF"), que o algoritmo de geração

da *hash* do conteúdo assinado foi SHA256, que o algoritmo de assinatura usado pelo Serviço Chave Móvel Digital foi RSA, que o perfil de assinatura usado foi de facto PAdES-B-LT e os detalhes do *timestamp* incluído na assinatura (carimbo de data/hora).

Figura 33: Painel de Assinaturas do documento assinado com assinatura PAdES-B-LT

Como a assinatura é PAdES-B-LT e é possível verificar na (Figura 33), esta é habilitada a *Long Term Validation*, i.e., capacidade da assinatura ser validada mesmo após o certificado de assinatura ter expirado, pois contém a informação de validação necessária para indicar que o certificado era valido no momento da assinatura, como respostas Online Certificate Status Protocol (OCSP) e/ou Certificate Revogation List (CRL).

Figura 34: Validação da assinatura PAdES-Baseline-LT na *WebApp* do DSS

Como é possível verificar na Figura 34, a assinatura é uma assinatura electrónica qualificada e utiliza o perfil PAdES-Baseline-LT.

Assinatura PAdES-B-LTA

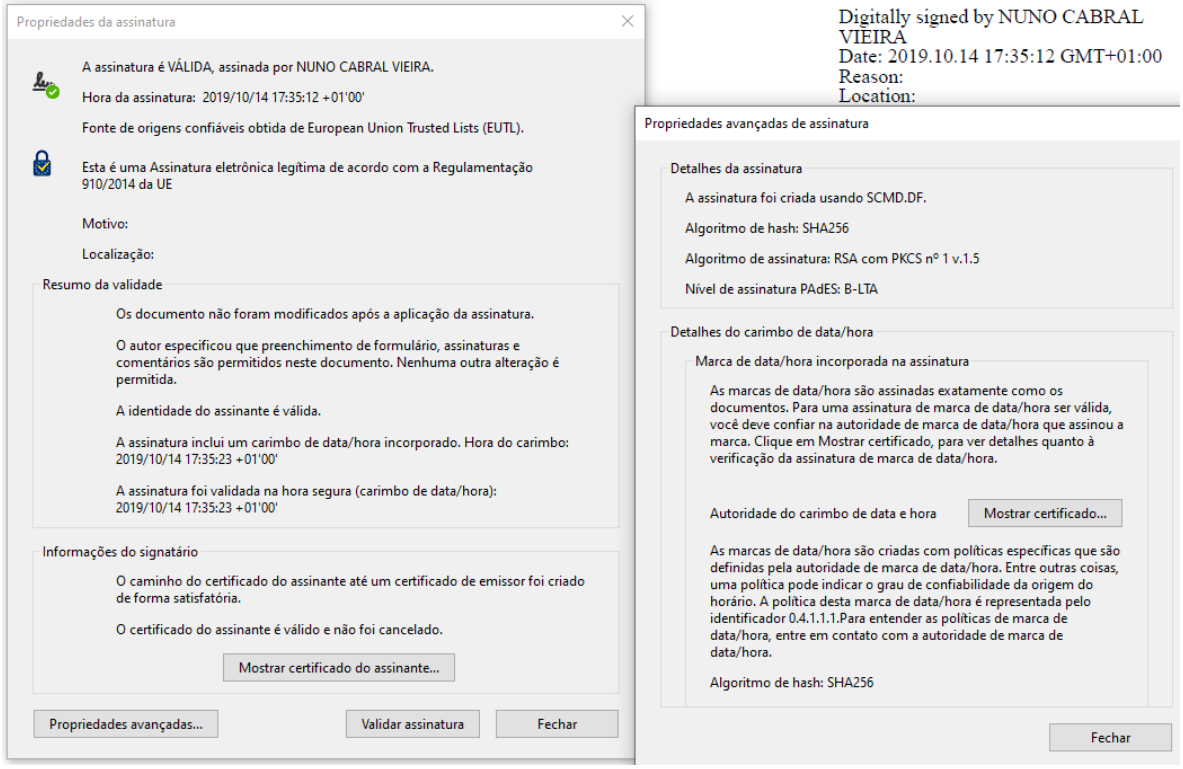


Figura 35: Assinatura PAdES-Baseline-LTA visializada no Adobe Reader, a sua validação e propriedades

O documento foi aberto no Adobe Reader, onde é possível visualizar a assinatura colocada no documento no canto superior direito, a comprovar que esta é visível, como é possível observar na Figura 35.

Na janela de Propriedades da assinatura (Figura 35) e no menu de Assinaturas (Figura 36) do documento assinado, é possível verificar que a assinatura é válida, o assinante e a hora a que esta foi criada. "Fonte de origens confiáveis obtida de European Union Trusted Lists (EUTL)" indica que o certificado de assinatura foi emitido por uma entidade autorizada pela União Euripeia, de acordo com o regulamento eIDAS (910/2014 [20]).

De acordo com o campo "Resumo da validade" (Figura 35) é possível verificar que após a operação de assinatura feita pela aplicação desenvolvida, o documento não é alterado, que a partir do momento que foi assinado só é permitido adicionar formulários, outras assinaturas e comentários e que de facto foi adicionado à assinatura um *timestamp* (carimbo de data/hora) a comprovar a hora de assinatura.

Na janela de Propriedades avançadas de assinatura (Figura 35), é possível verificar que a assinatura foi criada pela aplicação desenvolvida ("SCMD.DF"), que o algoritmo de geração da *hash* do conteúdo assinado foi SHA256, que o algoritmo de assinatura usado pelo Serviço Chave Móvel Digital foi RSA, que o perfil de assinatura usado foi de facto PAdES-B-LTA e os detalhes do *timestamp* incluído na assinatura (carimbo de data/hora).

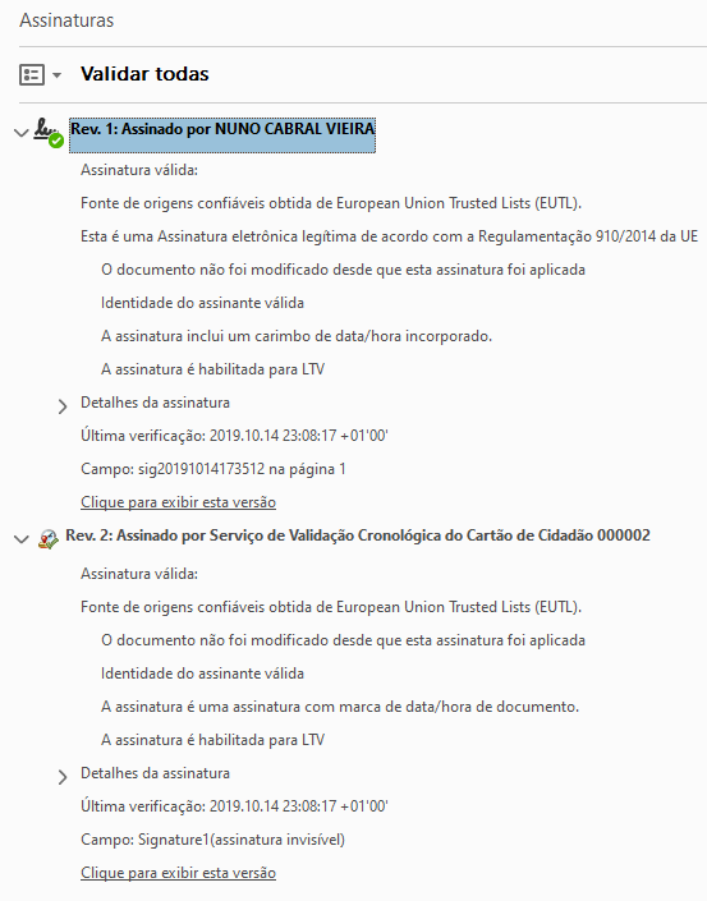


Figura 36: Painel de Assinaturas do documento assinado com assinatura PAdES-B-B

Como a assinatura é PAdES-B-LTA e é possível verificar na (Figura 36), esta é habilitada a *Long Term Validation*, i.e., capacidade da assinatura ser validada mesmo após o certificado de assinatura ter expirado, pois contém a informação de validação necessária para indicar que o certificado era valido no momento da assinatura e contém um *timestamp* de documento para reforçar a validade da assinatura.

É possível verificar no menu de assinaturas (Figura 36) que o timestamp adicionado é válido, que é de fontes confiáveis da EUTL e que também é habilitado a Long Term Validation, sendo possível validar mesmo depois do certificado de criação do *timestamp* expirar. Também é possível adicionar sucessivos *timestamps* para reforçar a validade da assinatura ao longo do tempo.

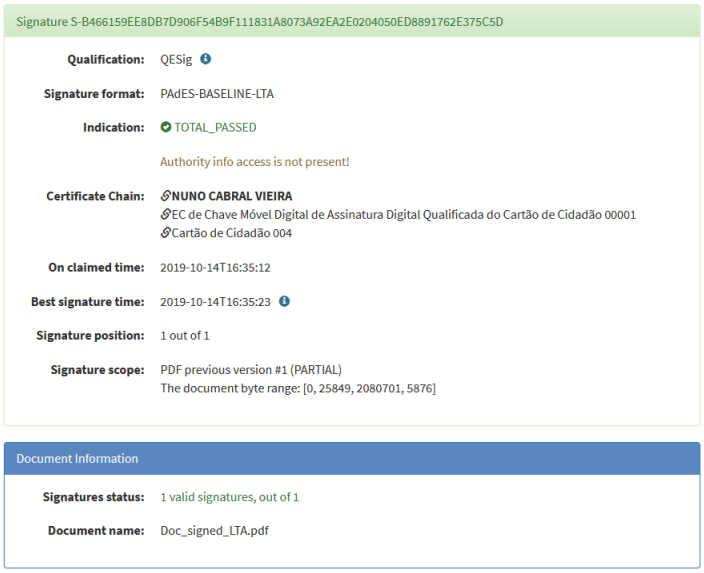


Figura 37: Validação da assinatura PAdES-Baseline-LTA na *WebApp* do DSS

Como é possível verificar na Figura 37, a assinatura é uma assinatura electrónica qualificada e utiliza o perfil PAdES-Baseline-LTA.

Por fim, os documentos foram submetidos à plataforma "ETSI Signature Conformance Checker", sendo validados em 64 elementos de teste, dando todos o mesmo resultado, passando a todos os testes menos a um.

Full Report		
Result	TI/VI	Tested Element and Test Test Result Details
1. Success	Tool	Location-{CodeTest}:Type-{InstancesNumber} Instances specified for child Type: 0..1. Instances found: 1
2. Success	Tool	Location-{CodeTest}:Type-{CheckIfValueIsOneOfDefined} Found value: 'Sig'. Allowed values: Sig
3. Success	Tool	Location-{CodeTest}:Filter-{InstancesNumber} Instances specified for child Filter: 1. Instances found: 1
4. Success	Tool	Location-{CodeTest}:SubFilter-{InstancesNumber} Instances specified for child SubFilter: 1. Instances found: 1
5. Success	Tool	Location-{CodeTest}:SubFilter-{CheckIfValueIsOneOfDefined} Found value: 'ETSI.CAdES.detached'. Allowed values: ETSI.CAdES.detached
6. Success	Tool	Location-{CodeTest}:-{InstancesNumber} Instances specified for child Cert: 0. Instances found: 0
7. Success	Tool	Location-{CodeTest}:ByteRange-{InstancesNumber} Instances specified for child ByteRange: 1. Instances found: 1
8. Success	Tool	Location-{CodeTest}:-{InstancesNumber} Instances specified for child Reference: 0..1. Instances found: 0
9. Success	Tool	Location-{CodeTest}:-{InstancesNumber} Instances specified for child Changes: 0..1. Instances found: 0
10. Success	Tool	Location-{CodeTest}:-{InstancesNumber} Instances specified for child Name: 0..1. Instances found: 0

Figura 38: 10 primeiros resultados dos testes da plataforma "ETSI Signature Conformance Checker"

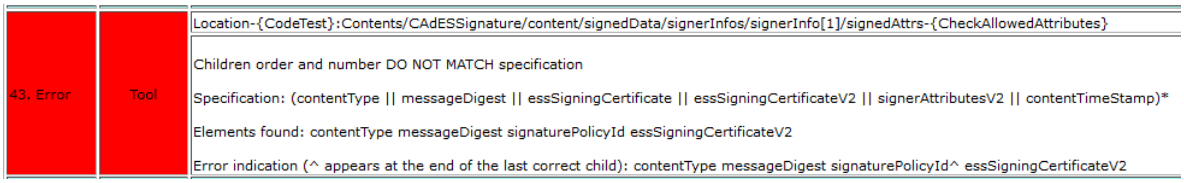


Figura 39: Único erro resultante dos testes da plataforma "ETSI Signature Conformance Checker"

O único erro presente, indica que o atributo de Politica de Assinatura(*signaturePolicyId*) não faz parte do conjunto de atributos *signedAttrs*, mas como é possível verificar no ETSI 319 122-1 na secção 5.2.9.1 [24] que especifica as assinaturas CAdES em que as assinaturas PAdES se baseiam, este atributo faz parte do conjunto de atributos *signedAttrs*. Logo, este erros parece ser um bug do "ETSI Signature COnformance Checker", e estando os outros testes em conformidade, é possível concluir que as assinaturas geradas pela a aplicação desenvolvida estão em conformidade com os ETSI EN 319 142-1[25] e ETSI EN 319 142-2[26].

5.2.2 *Testes*

Os resultados destes testes foram registados uma vez com a utilização de registos de depuração e podem variar tendo em conta o dispositivo utilizado e os processos a correr enquanto a aplicação está a fazer as operações de assinatura.

Tempos do processo de assinatura da aplicação desenvolvida, não incluindo o processo de comunicação com o SCMD:

	B	T	LT	LTA
2MB	1.5	1.5	2.1	2.9
10MB	1	0.8	2.1	2.3
50MB	2.2	2.1	3.4	3.4
100MB	4.1	4.1	5.5	5.4

Tabela 1: Tempos da operação de preparação da assinatura excluindo a operação de comunicação com o SCMD, referidas na secção 4.2, em segundos

	B	T	LT	LTA
2MB	0.35	0.7	4.5	6
10MB	0.15	0.35	4.9	7.7
50MB	0.15	0.4	4.7	8.8
100MB	0.35	0.65	6.9	17.5

Tabela 2: Tempos da operação de finalização da assinatura excluindo a operação de comunicação com o SCMD, referidas na secção 4.2, em segundos

Com os resultados da Tabela 1 e Tabela 2, podemos observar que:

- Para um documento de 2MB, os tempos totais de assinatura são 1.85s, 2.2s, 6.6s, 8.9s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 10MB, os tempos totais de assinatura são 1.15s, 1.15s, 7s, 10s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 50MB, os tempos totais de assinatura são 2.35s, 2.5s, 8.1s, 12.2s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 100MB, os tempos totais de assinatura são 4.45s, 4.75s, 12.4s, 22.9s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.

Tempos do processo de assinatura da aplicação desenvolvida, incluindo o processo de comunicação com o SCMD:

	B	T	LT	LTA
2MB	1.3	1.7	1.7	1.7
10MB	1.3	1.3	2.3	2.4
50MB	2.8	2.8	3.9	4.1
100MB	4.5	4.6	7.3	5.8

Tabela 3: Tempos da operação de Preparação da Assinatura, referida na secção 4.2, em segundos

	B	T	LT	LTA
2MB	2.5	3.2	6.8	8.5
10MB	2.4	2.6	7.7	9.3
50MB	2.4	2.7	7.5	13.7
100MB	2.7	2.9	8.1	18.6

Tabela 4: Tempos da operação de Finalização da Assinatura, referida na secção 4.2, em segundos

Com os resultados da Tabela 3 e Tabela 4, podemos observar que:

- Para um documento de 2MB, os tempos totais de assinatura são 3.8s, 4.9s, 8.5s, 10.2s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 10MB, os tempos totais de assinatura são 3.7s, 3.9s, 10s, 11.7s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 50MB, os tempos totais de assinatura são 5.2s, 5.5s, 11.4s, 17.8s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.
- Para um documento de 100MB, os tempos totais de assinatura são 7.2s, 7.5s, 15.4s, 24.1s, para os perfis PAdES-Baseline-B, T, LT e LTA respetivamente.

Tempo aproximado do processo da assinatura da aplicação Autenticação.gov com o perfil PAdES-Básico, que obrigatoriamente inclui o processo de comunicação com o SCMD:

	Preparação da Assinatura	Finalização da Assinatura
2MB	1s	3s
10MB	3s	4s
50MB	9s	8s
100MB	16s	12s

Tabela 5: Tempos de preparação e finalização da assinatura em segundos

Com os resultados da Tabela 5, podemos observar que:

- Para um documento de 2MB, o tempo total de assinatura é de 4s.
- Para um documento de 10MB, o tempo total de assinatura é de 7s.
- Para um documento de 50MB, o tempo total de assinatura é de 17s.
- Para um documento de 100MB, o tempo total de assinatura é de 28s.

Como referido anteriormente, o assinatura com perfil PAdES-Baseline-B foi usado para comparar com a assinatura com o perfil PAdES-Básico da Aplicação autenticação, pois é o mais comparável.

Tamanho inicial e final dos ficheiros assinados com a aplicação desenvolvida, utilizando cada perfil PAdES-Baseline e dos ficheiros assinados com a aplicação Autenticação.gov, utilizando o perfil PAdES-Básico:

	Aplicação Desenvolvida					Autenticação.gov
	Original	B	T	LT	LTA	Básico
2MB	2.24	2.26	2.27	4.86	6.46	2.49
10MB	10.76	10.85	10.86	13.5	15.23	11.17
50MB	52.95	52.97	52.98	55.56	57.15	53.06
100MB	105.26	105.29	105.3	107.9	109.53	105.5

Tabela 6: Tamanho dos ficheiros assinados antes e depois do processo de assinatura, em MegaBytes

Com os resultados da Tabela 6, podemos observar que:

- Para um documento de 2MB, o tamanho adicionado no fim do processo de assinatura é de 0.02MB, 0.03MB, 2.62MB, 4.22MB e 0.25MB para os perfis PAdES-Baseline-B, T, LT, LTA e Básico da Autenticação.gov respetivamente.
- Para um documento de 10MB, o tamanho adicionado no fim do processo de assinatura é de 0.09MB, 0.1MB, 2.74MB, 4.47MB e 0.41MB para os perfis PAdES-Baseline-B, T, LT, LTA e Básico da Autenticação.gov respetivamente.

- Para um documento de 50MB, o tamanho adicionado no fim do processo de assinatura é de 0.02MB, 0.03MB, 2.61MB, 4.2MB e 0.11MB para os perfis PAdES-Baseline-B, T, LT, LTA e Básico da Autenticação.gov respetivamente.
- Para um documento de 100MB, o tamanho adicionado no fim do processo de assinatura é de 0.03MB, 0.04MB, 2.64MB, 4.27MB e 0.24MB para os perfis PAdES-Baseline-B, T, LT, LTA e Básico da Autenticação.gov respetivamente.

5.3 DISCUSSÃO

Com a demonstração, na qual foram utilizados os meios necessários para validar as assinaturas criadas, podemos concluir que as assinaturas PAdES-Baseline criadas estão a ser geradas, adicionadas corretamente aos documentos e estão de acordo com as especificações técnicas dos standards de PAdES-Baseline, detalhados no ETSI EN 319 142-1 [25] e ETSI EN 319 142-2 [26].

O aviso presente na plataforma de validação de assinaturas da comissão Europeia não se deve ao processo de assinatura e sim ao certificado de raiz da cadeia de certificados da assinatura, logo é da responsabilidade da entidade que emitiu o certificado adicionar ao certificado o campo necessário para evitar este aviso.

Na demonstração da conformidade das assinaturas com o ETSI EN 319 142-1 [25] e o ETSI EN 319 142-2 [26], o erro que surge não coincide com a especificação das propriedades da assinatura presente no ETSI EN 319 122-1 [24], sendo considerado o atributo de politica de assinatura, acusado pela ferramenta como errado, como estando correto. Logo o erro foi considerado como uma falha da ferramenta.

No ambiente de testes, a medição de tempo de assinatura com o inclusão do processo de comunicação com o SCMD tem variações, devido à incertidade do tempo da resposta do serviço. Isto levou a uma medição do processo de assinatura sem a inclusão do processo de comunicação com o SCMD, que por sua vez também tem alguma variação nos tempos do processo de assinatura, devido aos processos que estão a ser executados ao mesmo tempo no dispositivo. Mas mesmo assim, é possível observar um claro aumento do tempo de assinatura com o aumento do tamanho do ficheiro e com a mudança de perfil de asssinatura, sendo o perfil PAdES-B-B o mais rápido por ser o mais básico e o PAdES-B-LTA o mais lento, por ser o mais complexo, como é possível verificar na secção 4.2.4 deste documento. O mesmo se observa na aplicação Autenticação.gov, com o aumento do tamanho do documento, o tempo de assinatura aumenta.

Como a medição dos tempos de assinatura da aplicação Aunteticação.gov tem que ser medidos com a inclusão obrigatória do processo de comunicação com o SCMD e o perfil de assinatura usado é o PAdES-Básico, é correto comparar com os tempos medidos do processo

de assinatura da aplicação desenvolvida, incluindo a cominunicação com o SCMD, do perfil PAdES-B-B.

	Tempo de assinatura em segundos			Tamanho da assinatura em MegaBytes		
	Baseline-B	Básico	Diferença	Baseline-B	Básico	Diferença
2MB	3.8	4	0.2	0.02	0.25	0.23
10MB	3.7	7	3.3	0.09	0.41	0.32
50MB	5.2	17	11.8	0.02	0.11	0.09
100MB	7.2	28	20.8	0.03	0.24	0.21

Tabela 7: Diferenças entre os tempos de assinatura e tamanho da assinatura dos perfis PAdES-Baseline-B e PAdES-Básico

Como podemos observar pela Tabela 7:

- No documento de 2MB, há uma diferença de 0.2 segundos, sendo uma diferença negligente;
- No documento de 10MB, há uma diferença de tempo de assinatura de cerca de 3 segundos;
- No documento de 50MB, há uma diferença de tempo de assinatura de cerca de 12 segundos;
- No documento de 100MB, há uma diferença de tempo de assinatura de cerca de 21 segundos;

Podemos concluir que apesar dos tempos serem muito aproximados na assinatura de um documento de 2MB, a partir do documento de 10MB começa a haver uma disparidade de tempos de assinatura. A diferença entre o tempo do processo de assinatura da aplicação desenvolvida e da aplicação Auntenticação.gov é cada vez maior à medida que o tamanho do documento aumenta, sendo a aplicação desenvolvida mais rápida a concluir o processo de assinatura.

Em relação ao tamanho dos documentos assinados, é possível observar que o tamanho adicionado ao documento no fim do processo de assinatura dos perfis PAdES-Baseline-LT e LTA é consideravelmente maior que o adicionado no processo de assinatura dos perfis PAdES-Baseline-B e T, mas isto deve-se à adição da informação de validação da assinatura ao documento nos perfis PAdES-Baseline-LT e LTA e mais a adição do *timestamp* e da informação de validação do mesmo no perfil PAdES-B-LTA.

Mas a diferença de tamanho dos documentos antes e depois do processo de assinatura também tem alguma incerteza entre cada documento e o mesmo se observa na aplicação Aunteticação.gov. A razão pela qual isto acontece, estará nas decisões de implementação do software IText7 utilizado e da aplicação Autenticação.gov.

Comparando as diferenças de tamanho da assinatura PAdES-B-B, com a assinatuar PAdES-Básico da aplicação Autenticação.gov, presentes na Tabela 7, podemos concluir que as assinaturas produzidas pela aplicação desenvolvida são consideravelmente mais pequenas, sendo o documento de 2MB o mais evidente, em que o tamanho da a assinatura PAdES-B-B e mais de dez vezes inferior que o tamanho da assinatura PAdES-Básico da aplicação Autenticação.gov. Mas, numa situação real, estas diferenças não são muito importantes, pois são diferenças na ordem dos KBytes.

5.4 SUMÁRIO

Após o desenvolvimento da aplicação, foi necessário demonstrar que a operação de assinatura cria uma assinatura e a coloca de maneira correta num documento PDF. Para tal, cada assinatura foi visualizada e validada no *Adobe Reader*, tendo identificado cada perfil PAdES-Baseline corretamente e validado a assinatura. Os documentos assinados, também foram submetidos à plataforma de validação de assinaturas PAdES-Baseline da comissão europeia, sendo comprovado que cada assinatura é uma assinatura eletrónica qualificada e utilizam corretamente os perfis PAdES-Baseline.

Por fim, cada documento foi submetido à plataforma *ETSI Signature Conformance Checker*, que testa se as assinaturas colocadas nos documentos estão em conformidade com o ETSI EN 319 142-1[25] e o ETSI EN 319 142-2[26], estando cada um em conformidade com estes, apenas apresentando um erro nos atributos de assinatura, acusando que o atributo de politica de assinatura não faz parte dos atributos de assinatura. Mas, como segundo o ETSI 319 122-1 [24], o atributo de politica de assinatura, faz parte dos atributos de assinatura, assume-se que tal erro é um *bug* da ferramenta.

Foram realizados testes às operações de assinaturas desenvolvidas, sendo medidos tempos de assinatura, e registadas diferenças de tamanho dos documentos no final da operação de assinatura.

Estes dados registados, foram usados para comparar a operação de assinatura da aplicação Autenticação.gov, que utiliza o perfil PAdES-Básico com a operação de assinatura da aplicação utilizando o perfil PAdES-B-B, pois este é o perfil mais comparável com o PAdES-Básico. Observando os resultados, a aplicação desenvolvida além de ser mais rápida a concluir o processo de assinatura, também produz assinaturas mais pequenas que a aplicação Autenticação.gov.

6

CONCLUSÃO

Considerando que a única oferta pública para assinar documentos PDF com o Serviço Chave Móvel Digital (SCMD), Autenticação.gov, só está disponível para Windows, Mac e Linux, não é possível a assinatura de documentos PDF com o SCMD em dispositivos móveis. A realização desta dissertação teve como objetivo o desenvolvimento de uma aplicação móvel Android e iOS de assinatura de documentos PDF utilizando o SCMD, em que a assinatura é colocada localmente no dispositivo, assim aliando a segurança da geração da assinatura pelo SCMD à conveniência da utilização de um dispositivo móvel para assinar documentos PDF.

Em virtude do objetivo mencionado, foi tomada como tarefa inicial o estudo e análise do serviço (*web services*). Este estudo tinha como propósito analisar a especificação fornecida do serviço, de modo a entender a estrutura dos pedidos e das respostas, assim facilitando a tarefa de implementação da comunicação com este. Durante o estudo da especificação do SCMD, observou-se que esta não era muito precisa no que toca à descrição dos conteúdos dos pedidos e das respostas, o que obrigou à realização extensiva de testes, para assim conseguir definir a correta estrutura e conteúdos dos pedidos. Tendo a correta estrutura e conteúdo dos pedidos, houve sucesso na comunicação com o serviço. Apesar deste sucesso, a assinatura resultante não coincidia com o esperado, sendo necessário contactar os responsáveis do SCMD, que comunicaram a necessidade de adição de um prefixo especifico aos dados a assinar. A adição deste prefixo, possibilitou a obtenção da assinatura esperada.

O conhecimento mais avançado do SCMD, obtido no decorrer deste estudo, vai possiblitar a correta comunicação, sem as dificuldades enfrentadas, com o SCMD na implementação da aplicação e em futuros projetos que pretendam utilizar o SCMD.

Seguidamente foi iniciado o estudo e análise da criação de assinaturas eletrónicas qualificadas. Este estudo tinha como propósito encontrar uma solução já existente de criação de assinaturas, pois não foi planeado para esta dissertação a implementação desta solução. Durante o estudo, foi observado que existe uma escassez de ofertas de soluções, principalmente de soluções para Android e iOS. Acabou por ser utilizada uma ferramenta de geração, programação e manipulação de PDFs em Java, o IText 7, que em versões prévias suportava Android, mas na versão mais recente não indica a existência de suporte para

Android, como também não é suportada em iOS. Apesar de não haver suporte oficial, para Android, foram realizados testes, nos quais se obteve sucesso na criação de assinaturas de documentos PDF, mas a sua utilização em iOS foi impossibilitada, devido a limitações de suporte da linguagem de programação utilizada na implementação da ferramenta. Após o sucesso dos testes, foi desenvolvida uma solução a partir da ferramenta IText 7, que em conjunto com a implementação da comunicação com o SCMD, assina documentos PDF com uma assinatura eletrónica qualificada, visível, válida e estando as assinaturas em conformidade com a especificação técnica dos standards das assinaturas eletrónicas avançadas de PDFs.

Dada a solução de assinatura de documentos PDF, utilizando o SCMD, implementada utilizando a ferramenta IText 7, entrou-se em contacto com os seus responsáveis, de modo a obter uma licença de utilização privada no desenvolvimento de uma aplicação Android. Esta verificou-se impossível de sustentar, pois como o suporte para Android não foi oficializado, o custo da licença proposto foi de um valor exagerado. Em alternativa, a ferramenta IText 7 pode ser utilizada sobre a licença AGPL, sem qualquer custo associado, tendo sido optado por utilizar esta licença, obrigando a disponibilização pública do código-fonte da solução desenvolvida.

Posteriormente, foi feito um estudo ao desenvolvimento de aplicações móveis Android e iOS. Este estudo tinha como propósito encontrar a ferramenta de desenvolvimento que melhor se adequasse ao desenvolvimento para as duas plataformas e que podesse integrar a solução de assinatura de documentos PDF, utilizando o SCMD, implementada utilizando a ferramenta IText 7. Durante o estudo, foi considerado que a utilização de uma ferramenta de desenvolvimento de aplicações multiplataforma seria o ideal, assim poupando no tempo de desenvolvimento separado das plataformas Android e iOS. Foi então optado por utilizar a framework de desenvolvimento de aplicações móveis Android e iOS, React Native, que permite desenvolver uma aplicação para as duas plataformas, utilizando a mesma base de código.

Utilizando a *framework* React Native, foi desenvolvida uma aplicação na qual é possível integrar soluções de assinatura documentos PDF, utilizando o SCMD. Como a solução desenvolvida só é suportada pela plataforma Android, a integração desta com a aplicação, resultou numa aplicação móvel de assinatura de documentos PDF utilizando o SCMD só para Android. Mas na eventualidade da existência de uma solução de assinatura documentos PDF com o SCMD, suportada por iOS, é possível integrar esta mesma com a aplicação desenvolvida em React Native, assim resultando também numa aplicação móvel iOS de assinatura de documentos PDF utilizando o SCMD.

O culminar das tarefas realizadas, resultou no cumprimento de parte do objetivo inicialmente planeado. Resultando numa aplicação Android, que apesar de certas funcionalidades terem ficado por desenvolver e melhorar, conseguiu atingir o objetivo planeado de

assinar documentos PDF usando o SCMD. Tal aplicação, que comparativamente à aplicação Autenticação.gov oferece assinaturas de documentos PDF mais robustas, termina o processo de assinatura mais rapidamente e produz assinaturas de menor tamanho.

6.1 TRABALHOS FUTUROS

Para dar continuidade ao desenvolvimento da aplicação móvel de assinatura de documentos PDF utilizando o SCMD, surge como trabalho futuro, o desenvolvimento das restantes funcionalidades e a consequente idealização e inclusão de um modelo de adição de anúncios à aplicação.

Para além disto, a solução desenvolvida de assinatura de documentos PDF com o SCMD utilizando o ITex7, poderá ser publicada, para assim ser utilizada noutros projetos que pretendam utilizar o SCMD. A solução desenvolvida tendo uma licença associada, só poderá ser utilizada de acordo com a licença AGPL.

Por fim, tendo sido descoberta numa fase avançada da dissertação a implementação em C# da ferramenta IText 7, levantou a hipótese de desenvolver a aplicação móvel Android e iOS de assinatura de documentos PDF com o SCMD, utilizando a framework Xamarin em conjunto com a ferramenta IText 7 em C#. Devido à fase avançado da dissertação, não se justificou recomeçar o desenvolvimento da aplicação já implementada utilizando React Native e a ferramenta IText 7 em Java, logo sendo deixado para trabalhos futuros a possibilidade de transitar a aplicação já desenvolvida para uma aplicação implementada utilizando Xamarin e a ferramenta IText 7 em C#.

BIBLIOGRAFIA

[1] The *ades collection: Cades, xades, pades and asic implementation for windows in c++. URL https://www.codeproject.com/Articles/1256991/The-AdES-Collection-CAdES-XAdES-PAdES-and-ASiC.

[2] Digital signature services. URL https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/DSS+releases.

[3] Itext. URL https://itextpdf.com/en.

[4] Política cmd de assinatura qualificada - pol16. URL https://www.autenticacao.gov.pt/documents/10179/615532/POL%2316.PolAssQual_signed_signed1.pdf.

[5] React native async storage. URL https://www.npmjs.com/package/@react-native-community/async-storage.

[6] Communication between native and react native. URL https://facebook.github.io/react-native/docs/communication-ios.html.

[7] React native document picker. URL https://www.npmjs.com/package/react-native-document-picker.

[8] Cháve movel digital - especificação dos serviços de assiantura, versão 1.7.

[9] *Manual de Utilização doMiddleware do Cartão de Cidadão*. URL https://www.autenticacao.gov.pt/documents/10179/11465/Manual+de+Utiliza%C3%A7%C3%A3o+da+Aplica%C3%A7%C3%A3o+do+Cart%C3%A3o+de+Cidad%C3%A3o+v3/3adb0af9-5852-4fc3-80dc-ef57814e6474.

[10] Java inheritance. URL https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html.

[11] Document management — portable document format — part 1: Pdf 1.7. URL https://www.iso.org/standard/51502.html.

[12] Native modules, react native. URL https://facebook.github.io/react-native/docs/0.60/native-modules-android.

[13] React native pdf. URL https://www.npmjs.com/package/react-native-pdf.

[14] URL https://en.wikipedia.org/wiki/SOAP.

[15] Decreto-lei n.º 290-d/99, 1999. URL https://dre.pt/application/conteudo/445741.

[16] Directiva 1999/93/ce do parlamento europeu e do conselho, de 13 de dezembro de 1999, relativa a um quadro legal comunitário para as assinaturas electrónicas, 1999. URL https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:31999L0093&from=PT.

[17] Decreto-lei n.º 18/2008, 2008. URL https://dre.pt/application/conteudo/248178.

[18] Decreto-lei n.º 88/2009, 2009. URL https://dre.pt/application/conteudo/603950.

[19] Etsi ts 102 778-1 v1.1.1. Technical report, 2009. URL https://www.etsi.org/deliver/etsi_ts/102700_102799/10277801/01.01.01_60/ts_10277801v010101p.pdf.

[20] Regulation (eu) n 910/2014. 2014. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG.

[21] Despacho (extrato) n.º 155/2017, 2017. URL https://dre.pt/application/conteudo/105693278.

[22] Declaração de práticas de operação do scmd. Technical report, 2018. URL https://www.autenticacao.gov.pt/documents/10179/958335/POL_01_DPO_signed_signed.pdf/.

[23] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and Dave Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008. URL https://rfc-editor.org/rfc/rfc5280.txt.

[24] ETSI EN 319 122-1. Part 1: Building blocks and cades baseline signatures. Standard, European Telecommunications Standards Institute, April 2016. URL https://www.etsi.org/deliver/etsi_en/319100_319199/31912201/01.01.01_60/en_31912201v010101p.pdf.

[25] ETSI EN 319 142-1. Part 1: Building blocks and pades baseline signatures. Standard, European Telecommunications Standards Institute, February 2016. URL https://www.etsi.org/deliver/etsi_en/319100_319199/31914201/01.01.00_30/en_31914201v010100v.pdf.

[26] ETSI EN 319 142-2. Part 2: Additional pades signatures profiles. Standard, European Telecommunications Standards Institute, April 2016. URL https://www.etsi.org/deliver/etsi_en/319100_319199/31914202/01.01.01_60/en_31914202v010101p.pdf.

[27] ETSI TS 101 733. Cms advanced electronic signatures (cades). Standard, European Telecommunications Standards Institute, April 2013. URL https://www.etsi.org/deliver/etsi_ts/101700_101799/101733/02.02.01_60/ts_101733v020201p.pdf.

[28] ETSI TS 102 778-2. Part 2: Pades basic - profile based on iso 32000-1. Standard, European Telecommunications Standards Institute, July 2009. URL https://www.etsi.org/deliver/etsi_ts/102700_102799/10277802/01.02.01_60/ts_10277802v010201p.pdf.

[29] ETSI TS 102 778-3. Part 3: Pades enhanced - pades-besand pades-epes profiles. Standard, European Telecommunications Standards Institute, July 2010. URL https://www.etsi.org/deliver/etsi_ts/102700_102799/10277803/01.02.01_60/ts_10277803v010201p.pdf.

[30] ETSI TS 102778-1. Part 1: Pades overview - a framework document for pades. Standard, European Telecommunications Standards Institute, July 2009. URL https://www.etsi.org/deliver/etsi_ts/102700_102799/10277801/01.01.01_60/ts_10277801v010101p.pdf.

[31] ETSI TS 103 172. Electronic signatures and infrastructures (esi); pades baseline profile. Standard, European Telecommunications Standards Institute, April 2013. URL https://www.etsi.org/deliver/etsi_ts/103100_103199/103172/02.02.02_60/ts_103172v020202p.pdf.

[32] Nick Pope, Denis Pinkas, and John Ross. CMS Advanced Electronic Signatures (CAdES). RFC 5126, March 2008. URL https://rfc-editor.org/rfc/rfc5126.txt.

[33] Stefan Santesson, Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Dr. Carlisle Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960, June 2013. URL https://rfc-editor.org/rfc/rfc6960.txt.

[34] Shruthi Sasidaran. Survey on native and hybrid mobile application development tools, 2017. URL http://ijarcet.org/wp-content/uploads/IJARCET-VOL-6-ISSUE-9-1389-1393.pdf.

[35] Jim Schaad. Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility. RFC 5035, August 2007. URL https://rfc-editor.org/rfc/rfc5035.txt.

[36] Ruben Smeets and Kris Aerts. Trends in web based cross platform technologies. 2016. URL https://core.ac.uk/download/pdf/80794774.pdf.

[37] Robert Zuccherato, Patrick Cain, Dr. Carlisle Adams, and Denis Pinkas. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, August 2001. URL https://rfc-editor.org/rfc/rfc3161.txt.