



IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Bárbara Sofia Eiras Ovelheiro

setembro | 2019





Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

RELATÓRIO DE PROJETO DE INFORMÁTICA

SOFTDIGITAL – FLUXO DIGITAL DE
DOCUMENTOS

BÁRBARA SOFIA EIRAS OVELHEIRO

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO
EM ENGENHARIA INFORMÁTICA

Setembro de 2019

Agradecimentos

Em primeiro lugar, um agradecimento interminável aos meus pais e irmão, à minha avó, à minha madrinha e tio, e restante família pelo apoio e por nunca me deixarem desistir. Sem dúvida uma etapa difícil que com determinação e foco foi superada com sucesso.

Um agradecimento especial à minha orientadora, Professora Doutora Maria Clara Silveira, pelo carinho e incentivo, pela atenção e disponibilidade, pela coragem e flexibilidade proporcionada, pelas orientações dadas para que este projeto fosse finalizado com sucesso.

Agradeço também ao Professor Doutor José Carlos Fonseca por tudo o que nos foi transmitido nas aulas de projeto em prol dos alunos e pela preocupação.

A todos os docentes da Licenciatura em Engenharia Informática que contribuíram para o meu sucesso neste percurso académico.

Aos meus amigos, colegas de curso mais próximos por acreditarem que sou capaz e por todo o carinho que me proporcionam, a eles um obrigada do coração.

De coração cheio.

Ficha de identificação

Estudante: Bárbara Sofia Eiras Ovelheiro

Número de estudante: 1012526

Email: bovelheiro@hotmail.com

Curso: Licenciatura em Engenharia Informática

Estabelecimento de ensino: Escola Superior de Tecnologia e Gestão, Instituto Politécnico da Guarda

Ano letivo: 2018/ 2019

Docente Orientador: Maria Clara Silveira

Grau académico: Doutoramento em Engenharia Eletrotécnica e Computadores

Email: mclara@ipg.pt

Resumo

O presente relatório descreve o projeto realizado no âmbito da unidade curricular de Projeto de Informática, integrada na Licenciatura de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

Com o avanço tecnológico das últimas décadas, a escrita é cada vez mais realizada em documentos eletrónicos. As assinaturas digitais foram criadas de modo a garantir a autenticação, integridade e não repúdio de documentos eletrónicos.

O objetivo do projeto desenvolvido consiste numa aplicação móvel em *Android* - SoftDigital, para a gestão de documentos em ambiente hospitalar, permitindo que os funcionários de saúde e utentes ou representante legal do utente integrem a assinatura digital de documentos para aprovações de tratamentos de forma eletrónica. A aplicação permite fluxos de trabalho totalmente digitais, podendo carregar e enviar documentos para que as respetivas pessoas possam assinar. Permite ainda de receber notificações e visualizar o estado do documento a qualquer momento.

Palavras-chave: *Android*, fluxo, digital, documento, *Portable Document Format*.

Abstract

This report describes the project carried out within the scope of the Informatics Project, Integrated Degree in Computer Engineering at the School of Technology and Management of the Guarda Polytechnic Institute.

With the technological advancement of the last decades, the writing is increasingly performed on electronic documents. The digital signatures were created to ensure authentication, integrity and repudiation of electronic documents.

The objective of the project developed is a mobile application in Android - SoftDigital, for document management in a hospital environment, allowing healthcare workers and users or legal representative of the user to integrate a digital signature of documents for approval of procedures electronically. The application allows fully digital workflow, upload and send documents to people sign them. Still allows to receive notifications and view document status at any time.

Keywords: Android, workflow, digital, document, Portable Document Format.

Índice geral

Agradecimentos	ii
Ficha de identificação	iii
Resumo	iv
Abstract.....	v
Índice geral	vi
Índice de figuras	viii
Índice de tabelas	x
Lista de siglas e acrónimos.....	xi
1. Introdução.....	1
1.1. Motivação	2
1.2. Objetivos.....	2
1.3. Estrutura do documento	3
2. Estado da arte	5
2.1. <i>Digital Signature for Contract Signing in Service Commerce</i>	5
2.2. Autenticação.GOV	7
2.3. DocuSign	10
2.4. SignEasy	11
2.5. Análise crítica	12
3. Metodologia	13
3.1. Metodologia de desenvolvimento: <i>Scrum</i>	13
3.2. Síntese das atividades <i>Scrum</i>	14
4. Análise de requisitos	17
4.1. Reunião na ULS da Guarda	17
4.2. Tabela de casos de uso.....	18
4.3. Diagrama de casos de uso	19
4.4. Descrição dos casos de uso	20
4.4.1. Inserir documento para assinatura digital.....	21
4.4.2. Assinar digitalmente o documento	22
4.4.3. Consultar documentos	23
4.5. Diagrama de atividades.....	24
4.6. Diagrama de estados	25
4.7. Diagrama de sequência	26

5.	Implementação	27
5.1.	Tecnologias utilizadas.....	27
5.1.1.	Android Studio	27
5.1.2.	Firebase.....	29
5.1.3.	GitHub	30
5.2.	Arquitetura do sistema	31
5.3.	Estrutura da aplicação	33
5.3.1.	Módulo FileUpload	34
5.3.2.	Módulo AllDocumentsFragment.....	37
5.3.3.	Módulo PdfViewer	38
5.3.4.	Módulo NeedToSignDocumentsFragment.....	40
5.3.5.	Módulo MessageActivity	42
6.	Verificação e validação	45
6.1.	Casos de teste.....	45
6.2.	Sumário de resultados dos testes realizados	48
7.	Conclusão.....	49
	Referências	51
	Anexos.....	53
	Anexo I. Certificado de assinatura pela DocuSign	53

Índice de figuras

Figura 1 - Arquitetura do sistema	6
Figura 2 - Logótipo Autenticação.GOV	7
Figura 3 - Interface da Aplicação do Cartão de Cidadão – Versão Windows.....	8
Figura 4 - Documento assinado digitalmente com assinatura válida	9
Figura 5 - Exemplo de um documento assinado com a Chave Móvel Digital	9
Figura 6 - Logótipo DocuSign.....	10
Figura 7 - DocuSign - Versão Mobile	10
Figura 8 - Logótipo SignEasy.....	11
Figura 9 - SignEasy - Versão mobile.....	11
Figura 10 - Scrum Overview	13
Figura 11 – Planeamento do Sprint Backlog	14
Figura 12 - Planeamento do Sprint Backlog aplicado ao presente projeto.....	15
Figura 13 - Diagrama de casos de uso do sistema SoftDigital	19
Figura 14 - Diagrama de atividades do sistema SoftDigital.....	24
Figura 15 - Diagrama de estados do documento digital do sistema SoftDigital	25
Figura 16 - Diagrama de Sequência "Inserir documento" do sistema SoftDigital	26
Figura 17 - Diagrama de classes do sistema SoftDigital	26
Figura 18 - Logótipo Android Studio, do Android e do Java, respetivamente.....	28
Figura 19 - Evolução do sistema operativo Android	28
Figura 20 - Exemplo de código em Java	29
Figura 21 - Logótipo do Firebase, Firebase Authentication, Realtime Database e Cloud Storage, respetivamente.....	29
Figura 22 - Logótipo GitHub.....	30
Figura 23 - Diagrama de arquitetura do sistema SoftDigital.....	31
Figura 24 - Diagrama de passos para a assinatura digital no sistema SoftDigital.....	32
Figura 25 - Módulos da app Android SoftDigital	33
Figura 26 - Extrato de código do módulo FileUpload: inserir documento para assinatura	34
Figura 27 - Interface Inserir documento na app SoftDigital.....	35
Figura 28 - Extrato do código da classe UploadPdf	35
Figura 29 - Aspeto da tabela Files no Firebase Database.....	36
Figura 30 - Extrato do código da classe DocumentStatus	36
Figura 31 - Aspeto da tabela Document Status no Firebase Database	36
Figura 32 - Extrato de código do módulo AllDocumentsFragment	37
Figura 33 - Interface Todos os Documentos na app SoftDigital	38
Figura 34 - Extrato do código do XML activity_pdf_view.....	38
Figura 35 - Extrato do código da classe PdfViewer	39
Figura 36 - Interface Assinar Documento na app SoftDigital	40
Figura 37 - Extrato de uma parte de código da classe NeedToSignDocumentsFragment	41
Figura 38 - Código para enviar uma notificação	41
Figura 39 - SreenShot de uma notificação recebida no dispositivo móvel	42
Figura 40 - Extrato do código da classe MessageActivity	42

Figura 41 - Interface Enviar para destinatário na app SoftDigital.....	43
Figura 42 - Extrato do código da classe Messages	43
Figura 43 - Aspeto da tabela Messages no Firebase DataBase	43
Figura 44 - Certificado de assinatura pela aplicação DocuSign	53

Índice de tabelas

Tabela 1 - Esquema da criptografia	6
Tabela 2 - Funcionalidades das aplicações similares existentes	12
Tabela 3 - Tabela Casos de Uso do sistema SoftDigital.....	18
Tabela 4 - Descrição de Caso de Uso "Inserir documento para assinatura digital"	21
Tabela 5 - Descrição do caso de uso "Assinar digitalmente o documento" do sistema SoftDigital	22
Tabela 6 - Descrição do caso de uso "Consultar documentos" do sistema SoftDigital..	23
Tabela 7 - Caso de teste “Consultar documentos”	45
Tabela 8 - Caso de teste “Assinar digitalmente o documento”	46
Tabela 9 - Caso de teste “Inserir documento para assinatura digital”	47
Tabela 10 - Sumário de resultados dos testes realizados	48

Lista de siglas e acrónimos

ALERT® PFH - *Alert® Paper Free Hospital*

app – Aplicação móvel

CA – *Certificate Authority*

CC – Cartão de Cidadão

GCM – *Google Cloud Message*

HMAC – *Hash-based Message Authentication Code*

IDE – *Integrated Development Environment*

JSON – *JavaScript Object Notation*

PDF – *Portable Document Format*

PIN – *Personal Identification Number*

PKI – *Public-key Infrastructure*

RA – *Registration Authority*

SDK – *Software Development Kit*

SI – *Signing Interface*

SMS – *Short Message Service*

UI – *User interface*

ULS – Unidade Local de Saúde

UML – *Unified Modeling Language*

XML – *Extensible Markup Language*

1. Introdução

O presente relatório descreve o projeto desenvolvido pela aluna Bárbara Ovelheiro, no âmbito da unidade curricular Projeto de Informática, da Licenciatura em Engenharia Informática no Instituto Politécnico da Guarda sob a orientação da Professora Doutora Maria Clara Silveira.

Atualmente, as pessoas dispõem de muitas formas de acesso à Internet desde computadores, *smartphones*, *tablets* e relógios inteligentes. O *marketing* para dispositivos móveis está constantemente a evoluir. Esta evolução tornou possível a existência de inúmeras aplicações que visam melhorar a qualidade de vida da população, resolvendo problemas do quotidiano.

Com vista a resolver problemas do quotidiano, o projeto desenvolvido consiste numa aplicação móvel em *Android* - SoftDigital, para a gestão de documentos em ambiente hospitalar, permitindo que os funcionários de saúde e utentes ou representante legal integrem a assinatura digital de documentos e aprovações de tratamentos de forma eletrónica, em qualquer dispositivo e fuso horário. A aplicação permite também fluxos de trabalho totalmente digitais, podendo carregar e enviar documentos para que as respetivas pessoas possam assinar. Possibilidade de receber notificações e visualizar o estado do documento a qualquer momento.

É indispensável a necessidade de segurança em todos os tipos de aplicações informáticas porque se exige uma maior fiabilidade e maior confiança em toda a informação, incluindo a sua origem. A segurança é uma questão importante ao projetar uma aplicação com um sistema de assinatura digital. Uma falha na segurança pode colocar em risco um documento confidencial que pode ser assinado usando uma chave privada do utilizador sem ser o proprietário da chave. Sendo a área da saúde uma área que envolve sigilo e confidencialidade quanto à informação e de grande vulnerabilidade deve ser devidamente protegida. A partir do momento em que o médico ou o paciente assinam um documento, sabe-se que foi aquele médico ou aquele paciente a assinar e é essa a garantia que se pretende. Uma assinatura digital sem poder ser verificada é desconsiderada, pois o objetivo é mesmo que se verifique se o documento é íntegro ou não.

Os documentos eletrónicos são documentos que podem ser assinados eletronicamente. A assinatura digital é usada como identificação da autoria dos documentos e têm a mesma autenticidade e valor legal que uma assinatura manuscrita. Para que a assinatura seja considerada válida é necessário conter menções obrigatórias e satisfazer as condições exigidas nas leis em vigor, garantido a veracidade da sua origem e a integridade do conteúdo. As assinaturas digitais foram criadas de modo a garantir a autenticação, integridade e não repúdio de documentos eletrónicos. Para tal, é imprescindível adquirir um certificado disponibilizado pelas entidades certificadoras.

1.1. Motivação

A motivação para desenvolver este projeto surgiu em ambiente hospitalar numa reunião realizada com o médico e enfermeiro responsáveis pelo Serviço de Quimioterapia da Unidade Local de Saúde da Guarda, no âmbito das unidades curriculares de Engenharia de Software II e Programação para a Internet da Licenciatura em Engenharia Informática. Durante a reunião, o médico salientou que o mundo da informática é uma “confusão” e que esta viria inovar, mas que se gasta exatamente o mesmo papel ou até mais. Referiu também que cada exame que o paciente faz tem de ser assinado um documento para o paciente ser informado e cada vez que faz um exame é mais outro documento e assim sucessivamente. Destacou que antigamente um único documento dava para todos os exames que o paciente realizava. Atualmente isso não acontece, é um documento para cada exame ou cirurgia que o doente faça, mesmo que o utente não queira fazer o tratamento tem de assinar o documento em como não aceita.

Posto isto, o motivo para a realização deste projeto foi criar uma solução viável para facilitar a integração com a assinatura qualificada de documentos digitais do hospital, criar um fluxo de trabalho, anular o desperdício de papel e diminuir o tempo de espera.

1.2. Objetivos

Este projeto tem como objetivo geral a implementação de uma aplicação móvel que permitirá ao utilizador o fluxo de documentos para assinatura digital através de um *smartphone Android*.

Os objetivos específicos da aplicação são:

- inserir documentos para assinatura digital – o médico seleciona um documento pretendido e carrega-o para aplicação;
- enviar documentos para o destinatário, que abrange o processo desde a escolha do documento por assinar, à integração com o *software* (Autenticação.GOV ou outro) para a assinatura e posteriormente envio para o destinatário;
- consultar documentos que é necessário assinar, que aguardam por outras assinaturas, documentos que estão assinados e concluídos por todos os intervenientes.

A aplicação a desenvolver terá em conta o fluxo de trabalho baseado em papel por troca eletrónica como consultar estado de assinatura do documento e consultar histórico de atividades do documento e receber notificações com documento por assinar. Deverá também permitir criar utilizadores com diferentes permissões. De salientar que a integração com o *software* Autenticação.GOV está fora do âmbito deste projeto.

1.3. Estrutura do documento

O presente relatório encontra-se dividido em sete capítulos distintos, nomeadamente a Introdução, Estado da arte, Metodologia, Análise de requisitos, Implementação, Verificação e validação e Conclusão.

O primeiro capítulo apresenta a introdução ao projeto, bem como os objetivos e a motivação. No segundo capítulo é realizado um estudo das aplicações existentes e de seguida uma análise crítica de cada uma delas. O terceiro capítulo trata a metodologia utilizada neste projeto. No quarto capítulo é feita uma análise de requisitos. O quinto capítulo trata a implementação, mais precisamente as tecnologias utilizadas e o desenvolvimento da aplicação SoftDigital. No sexto capítulo, apresenta a verificação e validação da presente aplicação. O sétimo e último capítulo são as conclusões relativas a este projeto.

2. Estado da arte

Neste capítulo é abordado o estudo de algumas aplicações existentes no mercado. Estas foram posteriormente analisadas para identificar e procurar contrabalançar as falhas existentes, podendo assim ser benéficas no desenvolvimento da aplicação. As aplicações estudadas têm características inovadoras e semelhanças com os objetivos propostos.

Nos tópicos seguintes são apresentados:

- um resumo do artigo científico, *Digital Signature for Contract Signing in Service Commerce* [1];
- algumas aplicações existentes no mercado para uma análise crítica comparativamente à solução que se pretende desenvolver.

2.1. *Digital Signature for Contract Signing in Service Commerce*

O título desta secção dá o nome ao artigo científico dos autores Emir Husni, Bramanto Leksono e Muhammad Ridho Rosa [1]. O artigo propõe um método de criação de uma assinatura digital igual à assinatura manuscrita de um contrato, que pode ser fornecida com o bilhete de identidade do assinante emitido pelo governo, usando uma aplicação web para poder assinar e verificar os documentos. A assinatura digital usa a criptografia assimétrica: chave privada para gerar uma assinatura de um documento *hash* e uma chave pública para verificar a assinatura. A assinatura digital pode ser verificada usando um certificado digital. No entanto, não há nenhuma garantia que a identidade do certificado digital tenha as mesmas informações que a de um bilhete de identidade.

A identidade do utilizador pode ser usada não apenas para assinatura, mas também no *login* e verificação de identidade. Neste caso, em vez de serem usados os dados, usa-se o valor *hash* da identidade do utilizador armazenada no servidor e no dispositivo do utilizador. Se o HMAC¹ (*Hash-based Message Authentication Code*) do utilizador enviado e o do servidor forem iguais, a ação solicitada pode prosseguir. A segurança é uma questão importante ao projetar um sistema de assinaturas. A falha da segurança pode colocar em risco um contrato. Por isso, a assinatura digital utiliza o algoritmo de criptografia. Os principais fatores a serem considerados na criptografia são: se o algoritmo tem vulnerabilidade conhecida e o comprimento da chave é considerada segura. Na Tabela 1 encontra-se o esquema da criptografia, adaptado de [1].

¹É uma construção específica para calcular o código de autenticação de mensagem (MAC) envolvendo uma função *hash* criptográfica em combinação com uma chave secreta

Tabela 1 - Esquema da criptografia

Purpose	Algorithm
Network Protocol	HTTPS using TLS
Hash	SHA-256
HMAC	SHA-256
Asymmetric Encryption	RSA-2048
Symmetric Encryption	AES-256

O sistema de assinatura digital consiste numa infraestrutura cliente-utilizador. As infraestruturas são prestadoras do serviço de assinatura e o utilizador é a entidade que possui a identidade e o administrador é a entidade que gere o sistema para se comportar como o pretendido. O acesso à arquitetura é feito usando uma API no protocolo HTTP. A Figura 1, mostra a arquitetura do sistema, adaptado de [1].

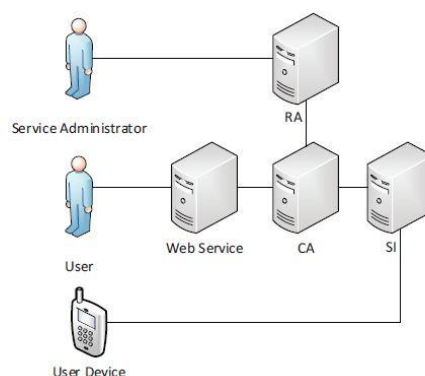


Figura 1 - Arquitetura do sistema

Os sistemas do tipo *Public-key Infrastructure* (PKI) são necessárias para as assinaturas digitais e também para uma gestão de chaves seguras para que seja possível suportar a encriptação e autenticação [1]. É então aqui que diz respeito à *Certificate Authority*.

Certificate Authority (CA) – é a autoridade que emite certificados digitais. Estes certificados são usados para proteger informações, criptografar transações e permitir comunicações seguras [1]. Esta é responsável por uma chave pública e uma chave privada. Para que as entidades sejam certificadas e reconhecidas umas pelas outras, o CA atribui um certificado digital e uma chave privada a cada uma delas.

Registration Authority (RA) – é a autoridade responsável por autenticar utilizadores finais que funciona como intermediário para identificação da identidade dos utilizadores que solicitam certificados [1]. Neste caso, poderá ser usado a Autenticação.GOV ou outro software (*DocuSign*, por exemplo).

Signing Interface (SI) – é a autoridade que fornece segurança à criação, armazenamento e acesso à chave privada do utilizador [1].

Segundo os autores, existem três grandes ideias ao criar a aplicação:

- Registo: garantir que as informações registadas no sistema são idênticas às informações indicadas no bilhete de identidade. Processos seguros no final do registo:
 - 1) Identidade do utilizador, imagem da assinatura, endereço GCM (*Google Cloud Message*) e a chave pública do utilizador armazenada no CA.

- 2) Chave privada do utilizador criptografada usando a chave do servidor e o PIN² do utilizador armazenado no SI.
 - 3) Identidade do utilizador e PIN armazenados no dispositivo do utilizador.
- Função Identidade: consiste numa notificação, *login*, verificação de identidade e assinatura de documentos.
 - Gerir Contratos: é uma função principal para gerir contratos a serem assinados. Três etapas relacionadas com a assinatura do documento:
 - 1) Criar um documento – o criador e o cliente podem criar um documento e quem é a pessoa a assinar;
 - 2) Assinar um documento – o assinante pode visualizar o conteúdo do documento; pode enviar uma solicitação usando o botão assinar e então a web irá continuar o processo de assinatura. O assinante pode responder à mensagem e enviar o PIN usando o dispositivo.
 - 3) Verificar – o criador do documento pode verificar o documento através do *download* do documento assinado. O criador e o cliente têm acesso ao relatório, que contém a identidade, os detalhes, os eventos, assinatura, comentários e atividade de ambas as partes.

Para concluir, o estudo deste artigo mostra a complexidade envolvida no desenvolvimento de uma aplicação com a assinatura digital qualifica e segura.

2.2. Autenticação.GOV

A Autenticação.GOV é um *software* fornecido pela AMA – Agência para a Modernização Administrativa, IP – Presidência do Conselho de Ministros. A Figura 2, apresenta o logótipo da Autenticação.Gov.



Figura 2 - Logótipo Autenticação.GOV

A Autenticação.GOV é uma aplicação dos meios de identificação eletrónica, assinatura digital e autenticação segura do Estado associada ao Cartão de Cidadão português [2]. Este aplicação tem *software* disponível para Windows, MAC(OS) e Linux. Possui também uma aplicação móvel que permite a receção do código de segurança da Chave Móvel Digital³ associado a cada autenticação, através de notificação *push*⁴ para o *smartphone*, sendo uma alternativa ao *sms*, *email* ou mensagem direta no *Twitter*⁵. É igualmente possível gerar novos códigos e controlar o tempo de vida de cada um [3]. A

² Senha com pelo menos quatro caracteres usados em cartões com banda magnética

³ Chave Móvel Digital é um meio simples e seguro de autenticação dos cidadãos em portais e sítios da Administração Pública na Internet, com dois fatores de segurança: uma palavra-chave e um código recebido por SMS.

⁴ Mensagem enviada em forma de notificação para *smartphones* ou *tablets*

⁵ Rede social de notícias de entretenimento, desporto e política

Autenticação.GOV usa a assinatura digital qualificada (assinatura digital admissível com o Cartão de Cidadão ou emitida por uma entidade certificadora) que permite ao titular de um Cartão de Cidadão, por vontade própria, assinar com a chave pessoal existente no seu CC. Qualquer entidade pode verificar a assinatura digital recorrendo ao uso do certificado digital pessoal do cidadão e a meios de verificação da validade deste certificado [2]. A assinatura digital pode ser verificada no painel de assinaturas do Adobe Acrobat Reader DC⁶.

Para assinar um documento eletrónico com o Cartão de Cidadão, é necessário:

- Instalar a aplicação *Desktop* informática;
- Ligar o leitor do Cartão de Cidadão ao computador;
- Arrastar o documento a assinar ou clicar para procurar o ficheiro;
- Clicar em Assinar com o Cartão de Cidadão e guardar documento.

Para assinar um documento eletrónico com a Chave Móvel Digital, é necessário:

- Instalar a aplicação *Desktop* informática;
- Instalar a aplicação móvel Autenticação.GOV no telemóvel;
- Efetuar um registo prévio da Chave Móvel Digital para poder autenticar com o número de telemóvel, PIN pessoal e um código de segurança recebido por *sms*.
- Na aplicação *Desktop*: arrastar o documento a assinar ou clicar para procurar o ficheiro;
- Clicar em Assinar com a Chave Móvel Digital, guardar documento e preencher os campos de autenticação da Chave Móvel Digital.

Na Figura 3, é apresentada a interface da Aplicação do Cartão de Cidadão – Versão *Windows* em que mostra claramente como é apresentada a assinatura através da Chave Móvel Digital.

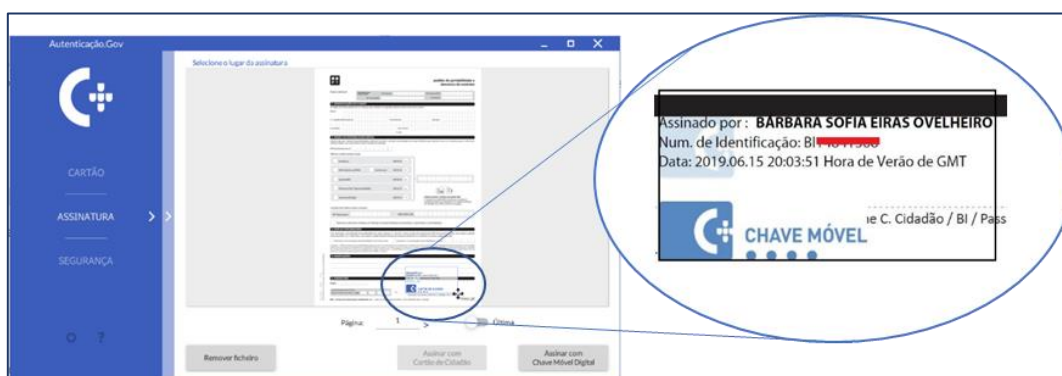


Figura 3 - Interface da Aplicação do Cartão de Cidadão – Versão *Windows*

Na Figura 4, apresenta-se um exemplo de um documento assinado através da versão *Desktop* do Windows, com a Chave Móvel Digital. Esta é constituída pelo nome completo

⁶ Software gratuito que permite visualizar, imprimir e comentar documentos em formato PDF

Figura 4 - Documento assinado digitalmente com assinatura válida

do assinante, pelo número de identificação e pela data do dia e hora de assinatura do documento.

Na Figura 5, ilustra-se um documento assinado digitalmente, com uma assinatura válida e com um documento íntegro, visível através do painel de assinaturas do Adobe Acrobat Reader DC.

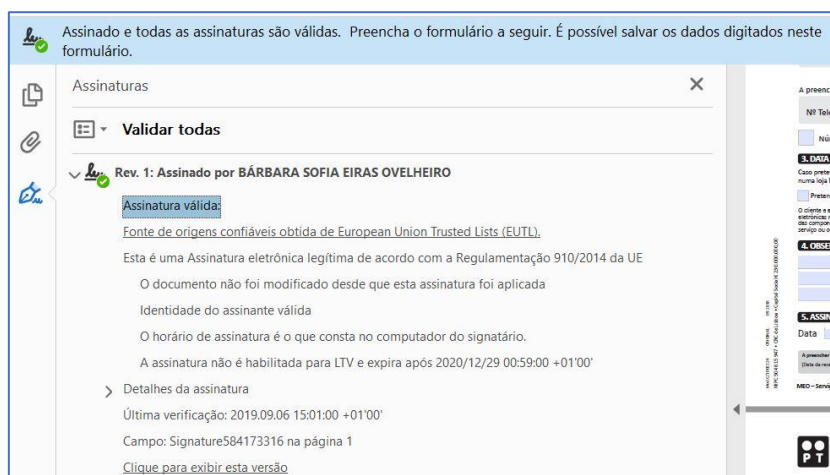


Figura 5 - Exemplo de um documento assinado com a Chave Móvel Digital

Esta aplicação permite a assinatura de documentos em formato PDF (*Portable Document Format*), a assinatura *offline*⁷ e a assinatura com a Chave Móvel Digital.

⁷ Diz-se “fora da linha” quando desconectado de um rede ou sistema de comunicações

2.3. DocuSign

A *DocuSign* é uma empresa americana sediada na Califórnia, que dá nome à aplicação para dispositivos móveis que permite a assinatura de documentos em formato PDF, o envio de documentos assinados, a assinatura sequencial, visualizar o estado do documento, receber notificações e também a assinatura *offline*. A Figura 6, apresenta o logótipo da *DocuSign*.



Figura 6 - Logótipo DocuSign

Esta app permite aos profissionais realizarem o trabalho de uma forma mais rápida, sem que estes necessitem de estar fisicamente no escritório, preparar e enviar documentos para assinatura e realizar coleta de pagamentos, descarta a assinatura presencial e consulta instantânea do estado do documento diretamente do *smartphone* ou *tablet*. É uma plataforma de *System of Agreement*⁸, baseada em *cloud*⁹, que permite a empresas de todos os setores modernizar e digitalizar rapidamente todos os processos que envolvem um contrato - desde a preparação do documento até à assinatura [4].

Na Figura 7, apresenta-se a *interface* da versão móvel da aplicação *DocuSign*. A primeira imagem ilustra a página inicial, a segunda a possibilidade de colocar uma assinatura personalizável e a terceira a escolha do local de assinatura pelo utilizador no documento.

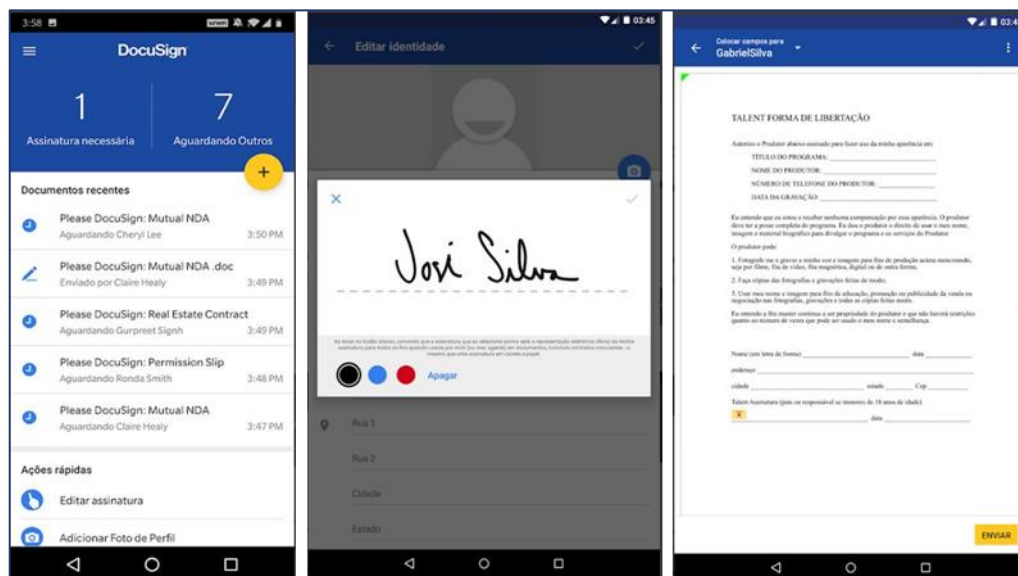


Figura 7 - DocuSign - Versão Mobile

De referir que esta aplicação permite a assinatura sequencial, ou seja, o mesmo documento pode ser assinado por duas ou mais pessoas.

⁸ Sistemas para preparar, assinar, agir e gerir contratos

⁹ Fornecimento de serviços informáticos, incluindo servidores, armazenamento, bases de dados, rede, software, análises e inteligência

No anexo I, encontra-se o certificado de assinatura que é enviado através do Email aquando a assinatura de um documento digital na app *DocuSign* e a respetiva descrição de todos os campos do certificado.

2.4. SignEasy

SignEasy é uma empresa com sede na Califórnia, que dá o nome à aplicação para dispositivos móveis. Na Figura 8, é apresentado o logótipo da *SignEasy*.



Figura 8 - Logótipo *SignEasy*

É uma solução *mobile-first*¹⁰ baseada em *cloud* para assinar digitalmente e preencher documentos em *smartphones*, *tablets* e web. Setores como imóveis, jurídicos, contabilidade, vendas, seguros, recursos humanos, logística usam a *SignEasy* para reduzir o tempo de resposta dos negócios, reduzir custos e eliminar o desperdício de papel. Permite a digitalização e envio de fluxos de documentos [5]. Na Figura 9, apresenta-se a interface da aplicação *SignEasy* – versão *mobile*.

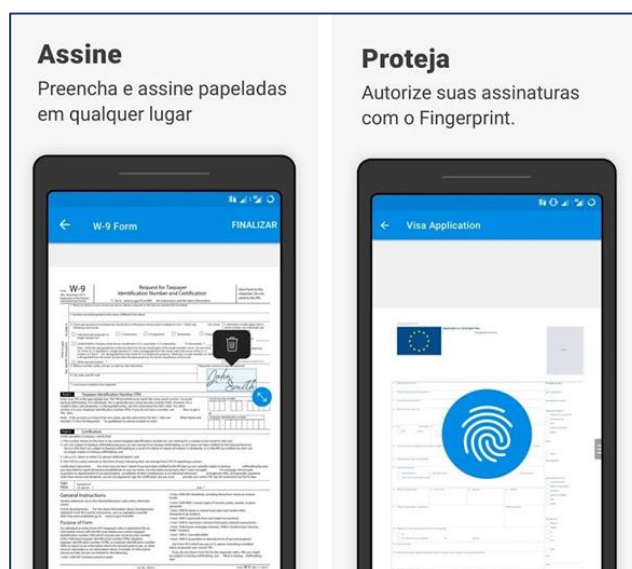


Figura 9 - *SignEasy* - Versão *mobile*

Esta aplicação permite a assinatura de documentos em formato PDF, o envio de documentos assinados, a assinatura sequencial, visualizar o estado do documento, receber notificações, assinatura *offline* e autorização de assinaturas com a impressão digital.

¹⁰ Conceito aplicado a projetos web onde o foco inicial da arquitetura e desenvolvimento é direcionado a dispositivos móveis

2.5. Análise crítica

Neste tópico é abordada uma análise crítica de algumas aplicações existentes. São excelentes soluções que alcançam na sua maioria os objetivos pretendidos para a criação da nova solução. Os autores [1] fazem um enquadramento de como fazer uma aplicação com a assinatura digital segura, que as aplicações estudadas devem incluir.

A aplicação Autenticação.GOV tem como foco a assinatura de documentos digitais através de uma Chave Móvel Digital, no entanto não possibilita ao utilizador fluxo de documentos.

A aplicação *DocuSign* e a *SignEasy* são uma solução bastante sólida sendo as que mais apresentam semelhanças com os objetivos propostos, no entanto, não possuem a assinatura com Chave Móvel Digital.

A Tabela 2, resume as funcionalidades de cada umas das aplicações:

Tabela 2 - Funcionalidades das aplicações similares existentes

Funcionalidades	Autenticação.GOV	DocuSign	SignEasy	SoftDigital
Assinatura com Chave Móvel Digital	Sim	Não	Não	A integrar
Assinatura de Documentos em formato PDF	Sim	Sim	Sim	A integrar
Envio de documentos assinados	Não	Sim	Sim	Sim
Assinatura Sequencial	Não	Sim	Sim	Possível
Visualizar estado do documento	Não	Sim	Sim	Sim
Notificações	Não	Sim	Sim	Sim
Assinatura <i>offline</i>	Sim	Sim	Sim	A integrar
Autorização com impressão digital	Não	Não	Sim	Possível

Depois de analisadas as funcionalidades de cada uma das aplicações, a proposta deste projeto visa melhorá-las e complementá-las. O sistema SoftDigital irá ter as seguintes funcionalidades: envio de documentos assinados; visualizar estado do documento; notificações; assinatura *offline*; autorização com impressão digital. A integração com o *software* Autenticação.GOV está fora do âmbito deste projeto.

3. Metodologia

Entende-se por metodologia a forma de se utilizar um conjunto coerente e coordenado de métodos para atingir um objetivo, de modo que seja evitada a subjetividade na execução do trabalho. Fornecendo um processo dinâmico e interativo para o desenvolvimento estruturado de projetos com vista à qualidade do produto final.

A metodologia adotada neste projeto é a metodologia ágil, em inglês, *Agile development*.

Agile development é uma maneira de desenvolver software onde existe uma grande preocupação em equilibrar as variáveis de custo, tempo e qualidade do produto utilizada em projetos curtos. Utiliza uma abordagem de planeamento incremental e iterativa [6].

No desenvolvimento ágil, assim que a equipa tem disponível todas as informações necessárias para a programação, esta atividade inicia-se. O código é o principal produto resultante do desenvolvimento. A partir de então cada nova funcionalidade implementada é mostrada ao cliente para validação.

3.1. Metodologia de desenvolvimento: *Scrum*

Sendo o *Scrum* umas das *frameworks* da metodologia ágil é, hoje em dia, a mais utilizada principalmente pela sua simplicidade e facilidade de adoção e adaptação. *Scrum* caracteriza-se pelas etapas de desenvolvimento, definidas como *Sprints*.

O funcionamento deste modelo pode ser observado na Figura 10 e na Figura 11. Como pode ser visto na Figura 10, o conjunto de funcionalidades definidas para um produto a ser entregue num projeto são agrupadas no *Product Backlog*.

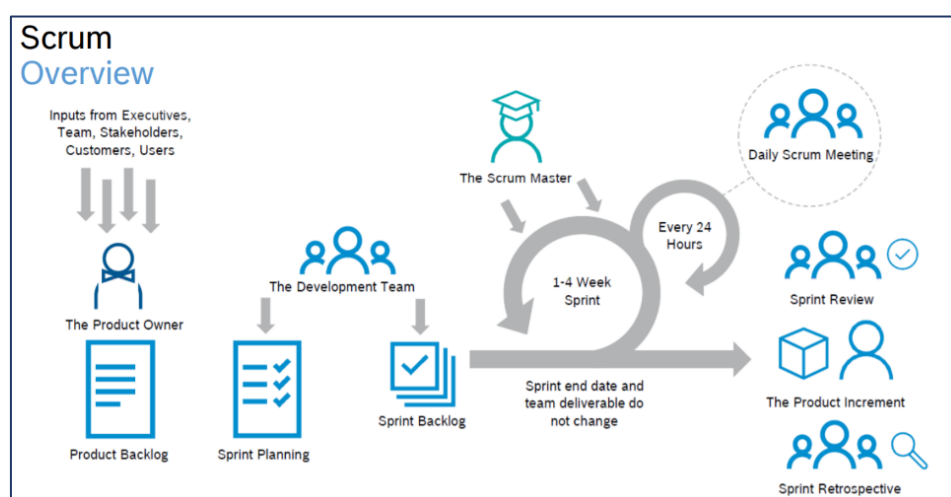


Figura 10 - Scrum Overview

Fonte: [11]

No início de cada *Sprint*, é realizada uma reunião de planeamento em que serão priorizadas o conjunto de funcionalidades do *Product Backlog* que serão implementadas durante o próximo *Sprint*, estas tarefas fazem assim, parte do *Sprint Backlog*. É a equipa de desenvolvimento em conjunto com *Scrum Master* que definem quais as tarefas a ser desenvolvidas e entregues. Encontra-se ilustrado na Figura 11, o quadro com o planeamento do *Sprint Backlog*.

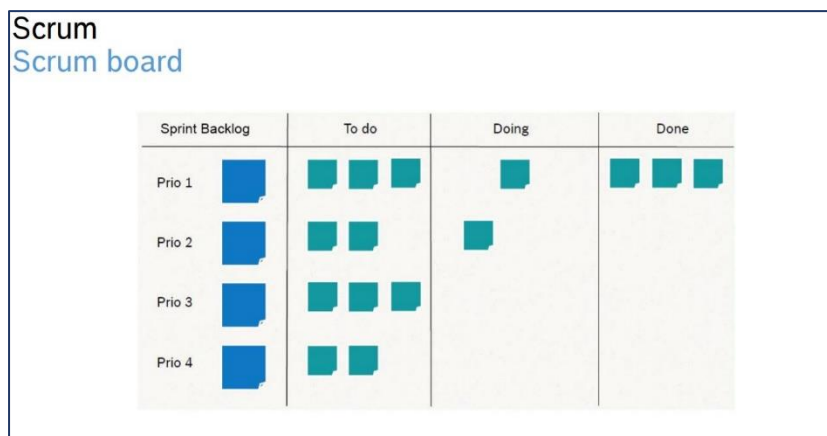


Figura 11 – Planeamento do *Sprint Backlog*

Fonte: [11]

A *Daily Scrum* é uma reunião com a duração máxima de 15 minutos que ocorre no início de cada dia, em que cada membro da equipa informa sobre o que foi feito no dia anterior e o plano para o próprio dia. Nesta reunião também são identificadas as eventuais situações que estejam a bloquear o andamento do processo de desenvolvimento. A *Sprint Review Meeting* é uma reunião realizada no final do ciclo de um *Sprint*, em que a equipa de desenvolvimento apresenta as funcionalidades que foram implementadas nesse ciclo.

Realiza-se também a *Sprint Retrospective* com as lições aprendidas durante o ciclo, os pontos positivos e o que é preciso melhorar nos próximos *Sprints*, é então planeado o próximo *Sprint*.

3.2. Síntese das atividades *Scrum*

O processo de desenvolvimento neste projeto foi uma adaptação personalizada da metodologia *Scrum*. A equipa é representada pela aluna e o *Scrum Master* composto pelo orientador e pelo docente da unidade curricular Projeto de Informática, que em conjunto definem as tarefas a serem desenvolvidas e entregues. O conjunto de tarefas é agrupado num *Product Backlog* que será desenvolvido até à próxima *Sprint*. As reuniões semanais com o orientador simulam uma *Sprint* que consistem em planear as tarefas a desenvolver. Estas tarefas a serem implementadas representam a *Sprint Backlog*. Na Figura 12, apresenta-se as atividades de um *Sprint Backlog* aplicado no presente projeto. O *Sprint Backlog* é composto por 14 semanas, em que cada semana representa uma reunião realizada com o orientador e ou docente da unidade curricular Projeto de Informática.

Sprint Backlog	To Do	Doing	Done
Semana 1	Estado da arte Tabela de casos de uso Apresentação intermédia Estruturar relatório		
Semana 2	Diagrama de arquitetura Diagrama de casos de uso Escrever relatório	Estado da arte Tabela de casos de uso Apresentação intermédia	Estruturar relatório
Semana 3	Diagrama de passos para assinatura digital	Apresentação intermédia Escrever relatório	Estado da arte Tabela de casos de uso Diagrama de arquitetura Diagrama de casos de uso
Semana 4	Implementação Autenticação	Escrever relatório	Apresentação intermédia Diagrama de passos para assinatura digital
Semana 5	Descrição do caso de uso: consultar estado de assinatura Descrição do caso de uso: assinar digitalmente Descrição do caso de uso: gerar certificado	Implementação Autenticação Escrever relatório	
Semana 6	Descrição do caso de uso: inserir documento Diagrama de atividades Protótipo do certificado Diagrama de Sequência "Inserir documento"	Escrever relatório	Implementação Autenticação Descrição do caso de uso: consultar estado de assinatura Descrição do caso de uso: assinar digitalmente Descrição do caso de uso: gerar certificado
Semana 7	Colocar aplicação no GitHub Implementação - Upload do PDF para o firebase Implementação - PDF view	Escrever relatório	Descrição do caso de uso: inserir documento Diagrama de atividades Protótipo do certificado Diagrama de Sequência "Inserir documento"
Semana 8	Diagrama de estados da assinatura digital Implementação - Home page	Implementação - Upload do PDF para o firebase Implementação - PDF view Escrever relatório	Colocar aplicação no GitHub
Semana 9	Implementação Fragmentos para estados dos documentos	Implementação - Home page Escrever relatório	Implementação - Upload do PDF para o firebase Implementação - PDF view Diagrama de estados da assinatura digital
Semana 10	Implementação Notificações documentos pendentes Preparar a primeira revisão do relatório	Implementação Fragmentos para estados dos documentos Escrever relatório	Implementação - Home page
Semana 11	Corrigir erros da 1ª revisão do relatório Implementação - Paint app para assinatura Preparar a segunda revisão do relatório	Implementação Fragmentos para estados dos documentos Escrever relatório	Implementação Notificações documentos pendentes Preparar a primeira revisão do relatório
Semana 12	Estrutura da aplicação Verificação e validação Corrigir erros da 2ª revisão do relatório	Implementação Fragmentos para estados dos documentos Escrever relatório	Corrigir erros da 1ª revisão do relatório Implementação - Paint app para assinatura Preparar a segunda revisão do relatório
Semana 13	Corrigir a versão final	Escrever relatório	Implementação Fragmentos para estados dos documentos Estrutura da aplicação Verificação e validação Corrigir erros da 2ª revisão do relatório
Semana 14	Preparar a apresentação		Escrever relatório Corrigir a versão final

Figura 12 - Atividades do Sprint Backlog aplicado ao presente projeto

O uso desta metodologia *Scrum* facilitou o planeamento das tarefas a realizar durante o desenvolvimento do projeto. O quadro dá uma visão mais clara das tarefas a cumprir. Sendo esta metodologia um caso de sucesso, pretende-se numa eventualidade utilizá-la no futuro em contexto de empresa.

4. Análise de requisitos

Os requisitos são características que o *software* ou o sistema a desenvolver deverá cumprir. Definem claramente aquilo que o *software* terá de realizar para satisfazer as necessidades da organização e dos utilizadores.

Os requisitos podem ter dupla função: base de um contrato aberto às interpretações; base do próprio contrato definido em detalhe. Para capturar os requisitos usam-se casos de uso. Um caso de uso é um modo específico de usar um sistema – alguma funcionalidade é realizada pelo sistema como resposta a um estímulo do ator. Os casos de uso formam um diálogo entre o ator e o sistema. Os casos de uso providenciam um veículo para: capturar os requisitos sobre o sistema; comunicar com os utilizadores finais e os peritos na matéria; testar o sistema.

Ao longo deste capítulo usa-se a linguagem *UML* (*Unified Modeling Language*) para modelar o sistema SoftDigital. A *UML* é uma linguagem de modelação gráfica para especificação, visualização, construção e documentação de sistemas de *software* ou outros [7].

4.1. Reunião na ULS da Guarda

A reunião realizada com o médico e o enfermeiro responsáveis pelo Serviço de Quimioterapia da ULS da Guarda, surgiu no âmbito das unidades curriculares de Engenharia de Software II e Programação para a Internet como já referido anteriormente no capítulo 1. Como a motivação para a criação do presente projeto surgiu neste âmbito, foi proposto solicitar uma opinião sincera ao profissional de saúde que esteve presente nesta reunião.

A citação a seguir, trata-se de um email enviado para o enfermeiro responsável pelo Serviço de Quimioterapia:

“Numa reunião que tivemos em outubro do ano passado, surgiu uma ideia de uma aplicação móvel que permitisse a autorização de operações, exames e cirurgias de forma facilitada. Vinha desta forma solicitar a sua opinião sincera como profissional, uma vez que pretendo realizar o meu projeto final de curso nesse sentido.

O projeto consiste no desenvolvimento de uma aplicação em Android para conectar a respetiva assinatura no telemóvel com o documento pdf no computador da respetiva autorização de operação, exame ou cirurgia do paciente.”

A citação a seguir, apresenta a resposta ao email enviado para o enfermeiro responsável:

“Relativamente á questão que me coloca, sou de opinião que o seu projeto faz sentido e tem aplicabilidade prática. A obrigatoriedade do "consentimento informado" por parte do Utente como prova de autorização para a realização de procedimentos é uma realidade, mas que só ainda é possível no meio hospitalar em suporte de papel. Este tipo de ferramenta viria agilizar e facilitar todo o processo.”

4.2. Tabela de casos de uso

Como já referido no tópico anterior, usam-se casos de uso para o levantamento de requisitos. Um Caso de Uso (*Use Case*) é uma sequência de transações relacionadas que são realizadas por um ator do sistema num processo de diálogo. Mais concretamente, um caso de uso é uma “porção” de funcionalidade [8]. Os casos de uso apresentados correspondem aos objetivos dos atores.

A tabela de casos de uso representada a seguir, Tabela 3, apresenta todos os atores que fazem parte do sistema SoftDigital bem como os objetivos de cada um.

Tabela 3 - Tabela Casos de Uso do sistema SoftDigital

Atores	Objetivos
Médico / Enfermeiro	Inserir documento para assinatura digital; Assinar digitalmente o documento; Consultar estado da assinatura do documento; Consultar histórico de atividade do documento; Criar utilizador; Receber notificação com documento por assinar.
Utente / Representante legal do Utente	Assinar digitalmente o documento; Consultar histórico de atividade do documento; Criar utilizador; Receber notificação com documento por assinar.
Autenticação.GOV ou outra	Assinatura digital qualificada.

O papel dos atores:

- Médico/Enfermeiro - é um utilizador final do sistema que pode: inserir documentos (cirurgias, consentimento informado, ...) para assinatura digital; assinar digitalmente o documento; consultar o estado da assinatura do documento; consultar o histórico de atividade do documento; criar utilizador e receber notificação com documento por assinar;
- Utente/Representante legal do Utente - é um utilizador final do sistema que pode: assinar digitalmente o documento; consultar o histórico de atividade do documento; criar utilizador e receber notificação com documento por assinar;
- Autenticação.GOV ou outra – para integração com o sistema SoftDigital de modo a permitir a assinatura digital qualificada.

4.3. Diagrama de casos de uso

Depois de apresentado no tópico anterior os atores que fazem parte do sistema, bem como os objetivos e o papel de cada um, segue-se o diagrama de casos de uso. Um diagrama de casos de uso mostra os casos de uso, atores e as suas interações. Na Figura 13, encontra-se ilustrado num diagrama os casos de uso e os respetivos atores. Este serve para mostrar todas as funcionalidades que o sistema SoftDigital vai ter e quem tem acesso a essas funcionalidades.

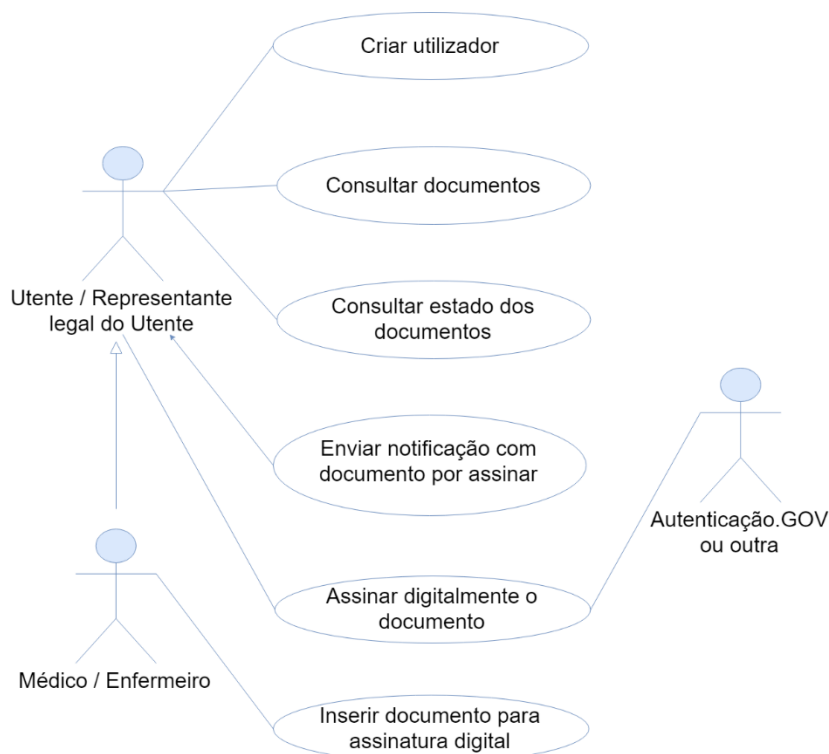


Figura 13 - Diagrama de casos de uso do sistema SoftDigital

Todos os casos de uso incluem o *Login* válido, que por uma questão de fácil leitura não está incluído no diagrama de casos de uso. Mas para proceder a cada operação é necessário o Médico/ Enfermeiro e o Utente/ Representante legal do Utente estarem registados e introduzirem email e password válidos no sistema SoftDigital.

4.4. Descrição dos casos de uso

Depois de identificada a forma como os atores irão interagir com o sistema no tópico anterior, o passo seguinte será documentar os casos de uso. Cada caso de uso precisa de ser documentado com o fluxo de eventos e isto é feito com base no ponto de vista dos atores. Deverá ser detalhado aquilo que o sistema tem de fornecer ao ator quando o caso de uso é executado. Tipicamente, mostrará como o caso de uso começa e termina. Como aspeto de chave para a compreensão deste conceito, convém ter em atenção que um caso de uso não é um módulo de *software* – é antes algo que fornece valor ao ator.

A descrição de um caso de uso mostra a razão da necessidade do Sistema. São muito úteis para auxiliar na análise de requisitos do sistema. Cada caso de uso corresponde a um requisito potencial, sendo por isso essencial para se obter uma visão mais aprofundada de como o utilizador fará a interação com a aplicação.

O que tem de ter a descrição de um caso de uso? Por um lado, o fluxo normal de eventos (*happy path*), em que tudo corre bem. Por outro lado, existe o fluxo de eventos anormal, em que as coisas não correm de forma normal.

Cada descrição de caso de uso é constituída pelos seguintes tópicos (*template*):

- **Nome:** nome do caso de uso que se irá descrever;
- **Descrição:** descrição curta e sucinta do caso de uso em questão. Deverá ser perceptível o que se pretende numa curta frase;
- **Pré-Condição:** condição inicial necessária para que o caso de uso decorra com sucesso;
- **Caminho Principal:** descrição de como o utilizador deve proceder para que tudo corra com sucesso;
- **Caminhos Alternativos:** descrição do que poderá correr mal em determinado passo do caminho principal;
- **Pós-Condição:** condição em que se encontra o sistema após o término deste caso de uso;
- **Suplementos ou adornos:** descrição de testes a realizar, requisitos não funcionais.

As secções seguintes descrevem os principais dos casos de uso com o *template* apresentado.

4.4.1. Inserir documento para assinatura digital

Quando o ator (médico ou enfermeiro) interage com o sistema SoftDigital vai desencadear uma sequência de eventos que permitem desenvolver este caso de uso. A Tabela 4, descreve o processo que o médico/ enfermeiro produz ao inserir um documento para assinatura digital.

Tabela 4 - Descrição de Caso de Uso "Inserir documento para assinatura digital"

Nome	Inserir documento para assinatura digital.
Descrição	Este caso de uso tem como objetivo descrever o processo de inserir um documento para assinatura digital. Os documentos a inserir podem ser para cirurgia, consentimento informado, etc.
Pré-Condição	Login válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Selecione o documento”; 2. O sistema apresenta uma lista; 3. O ator seleciona o documento pretendido, introduz um nome para o documento e seleciona a opção “Inserir documento”; 4. O sistema carrega o documento pretendido e apresenta o documento para assinatura.
Caminhos Alternativos	<ol style="list-style-type: none"> 2. a) Lista de documentos vazia; 4. a) Falha ao carregar documento, o sistema redireciona para a página inicial; b) O sistema alerta o ator que não introduziu um nome para o documento.
Pós-Condição	O sistema envia uma notificação com documento “É necessário assinar”.
Suplementos ou adornos	Testar se o documento foi inserido (clicar em Inserir documento sem ter selecionado nenhum documento; clicar em Inserir documento sem ter introduzido um nome para o documento).

4.4.2. Assinar digitalmente o documento

O médico ao selecionar um documento para assinar, o sistema irá interagir com o *software* Autenticação.GOV.

A Tabela 5, descreve o processo para assinar digitalmente um documento PDF.

Tabela 5 - Descrição do caso de uso "Assinar digitalmente o documento" do sistema SoftDigital

Nome	Assinar digitalmente o documento.
Descrição	Este caso de uso tem como objetivo descrever o processo de assinar digitalmente documentos com a assinatura digital.
Pré-Condição	Login válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Assinar documento”; 2. O sistema apresenta o(s) documento(s) que é necessário assinar; 3. O ator seleciona o documento para assinar; 4. O sistema abre o documento selecionado; 5. O ator seleciona o ícone assinar; 6. O sistema interage com o <i>software</i> (Autenticação.GOV ou outro) de assinatura; 7. O ator assina o documento e submete; 8. O sistema guarda o documento assinado e envia.
Caminhos Alternativos	<ol style="list-style-type: none"> 1. a) O ator seleciona a barra de estados relativo aos documentos é necessário assinar; 2. a) Lista de documentos “é necessário assinar” vazia; 4. a) Falha ao abrir o documento, o sistema redireciona para a página inicial com mensagem “Falha ao abrir documento”; 8. a) Se o ator for o médico/enfermeiro, envia documento para o utente/representante legal; b) Se o ator for o paciente, envia documento para o destinatário de origem.
Pós-Condição	O sistema envia uma notificação de que o documento foi enviado com sucesso.

4.4.3. Consultar documentos

A Tabela 6, mostra o caso de uso consultar documentos e respetivos estados (é necessário assinar, a aguardar por outros e concluído). O estado “é necessário assinar” verifica-se: quando o médico insere um documento na aplicação SoftDigital; e quando o utente recebe o documento para assinar. O estado “a aguardar por outros” verifica-se quando o médico envia o documento para o utente. E o estado “concluído” é quando o médico recebe o documento do utente assinado.

Tabela 6 - Descrição do caso de uso "Consultar documentos" do sistema SoftDigital

Nome	Consultar documentos.
Descrição	Este caso de uso tem como objetivo de consultar os documentos e respetivos estados.
Pré-Condição	Login válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator clica em consultar documentos; 2. O sistema apresenta todos os documentos; 3. O ator seleciona uma das opções disponíveis (é necessário assinar, a aguardar por outros, assinado); 4. O sistema apresenta o documento de acordo com o estado selecionado (Nome do documento e nome do remetente).
Caminhos alternativos	<ol style="list-style-type: none"> 2. a) O sistema apresenta uma lista vazia; 4. a) O sistema apresenta uma lista vazia.
Suplementos ou adornos	<p>Testar se o documento que esteja não assinado se encontre no estado concluído;</p> <p>Testar se o documento já está assinado e ser apresentado na lista de “É preciso assinar”.</p>

4.5. Diagrama de atividades

O diagrama de atividades é usado para representar os vários processos pelo qual uma atividade passa de forma a ser completamente realizada, podendo incluir o fluxo de controlo.

Como ilustrado na Figura 14, podem-se ver as atividades representadas por retângulos arredondados. As atividades são tipicamente estados de ação – estados que transitam automaticamente para o estado seguinte depois da ação estar completa.

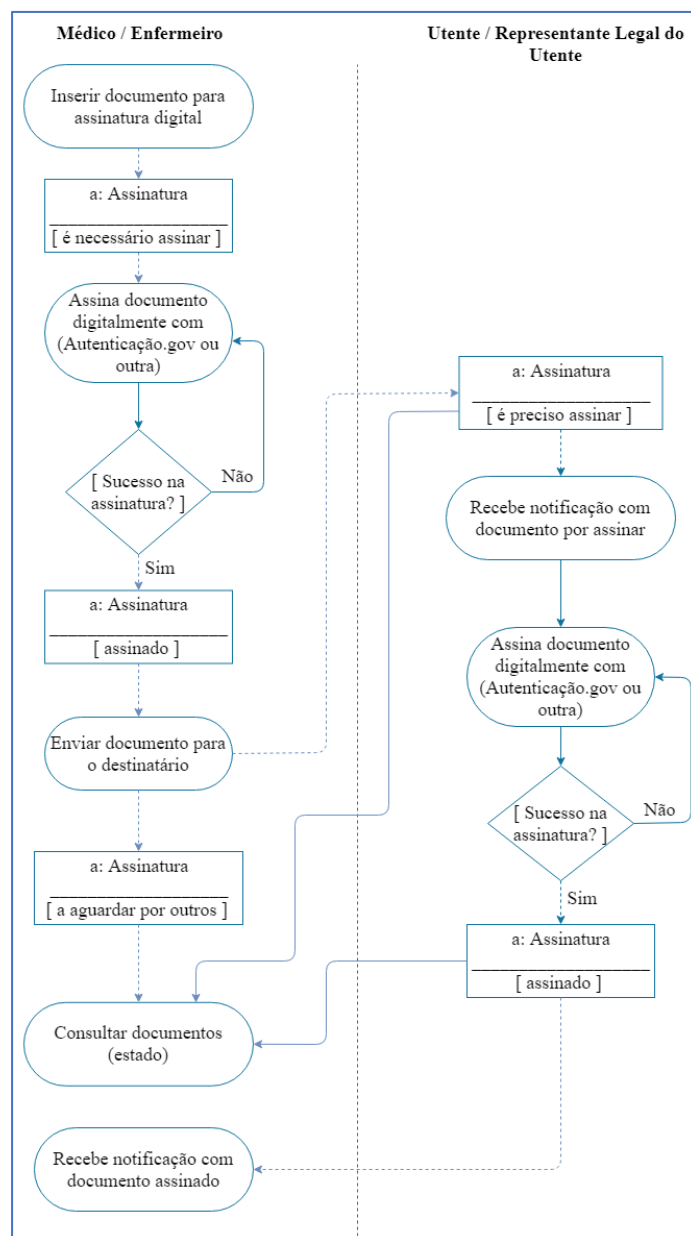


Figura 14 - Diagrama de atividades do sistema SoftDigital

Uma das coisas que se pode mostrar com os diagramas de atividades são as *swimlanes* que representam cada ator na vertical. Uma das melhores formas de utilizar as *swimlanes* e os diagramas de atividade é mostrar quem tem responsabilidade de algo. Desta forma, mostra-se claramente a todos as partes qual o ator que é responsável por cada atividade.

4.6. Diagrama de estados

Um diagrama de estados mostra os eventos que causam transição de um estado para o outro, assim como as ações que resultam de uma alteração de estado. Um evento é uma ocorrência significativa que tem uma localização no tempo e no espaço [8].

Na Figura 15, mostra-se o diagrama de estados do documento digital de acordo com o diagrama de atividades da secção anterior.

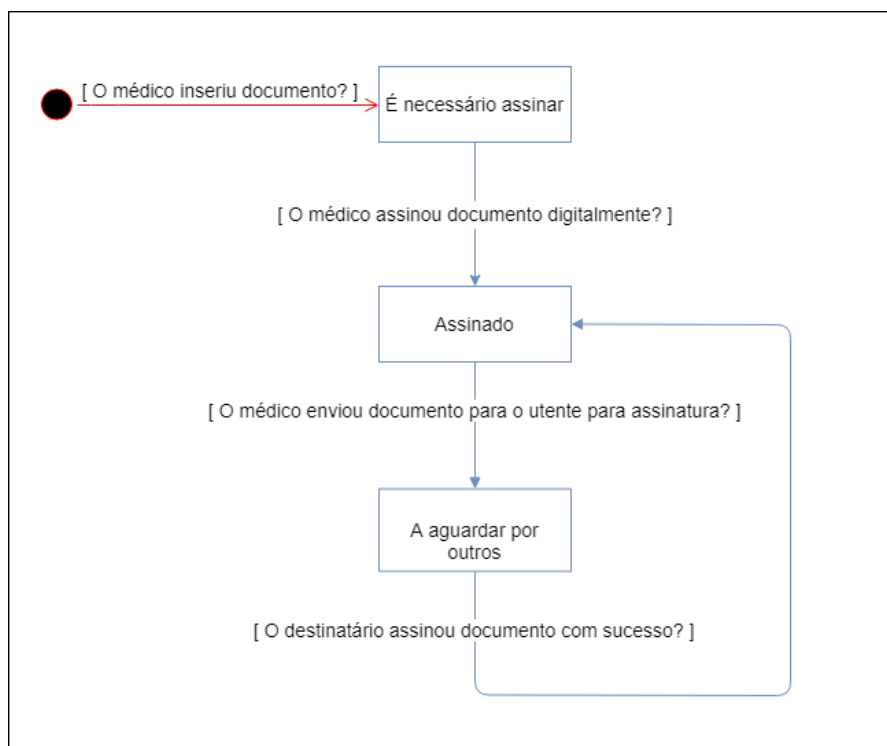


Figura 15 - Diagrama de estados do documento digital do sistema SoftDigital

O médico/ enfermeiro ao inserir um documento na aplicação SoftDigital, este transita para o estado “é necessário assinar”. Quando o médico assina o documento passa para o estado “assinado”, mas se o médico enviar o documento para o utente/ representante legal assinar o estado transita para “a aguardar por outros”. Quando o utente assinar o documento passa para o estado “assinado”.

4.7. Diagrama de sequência

Os diagramas de sequência mostram as interações de objetos organizadas numa sequência temporal. Pode-se utilizar o fluxo de eventos para determinar quais os objetos e interações que é preciso para completar a funcionalidade especificada pelo fluxo de eventos. Os diagramas de sequência são bons para mostrar aquilo que se está a passar, para chegar a requisitos e para trabalhar com os clientes. A Figura 16, mostra como é que o médico/enfermeiro insere um documento no sistema SoftDigital.

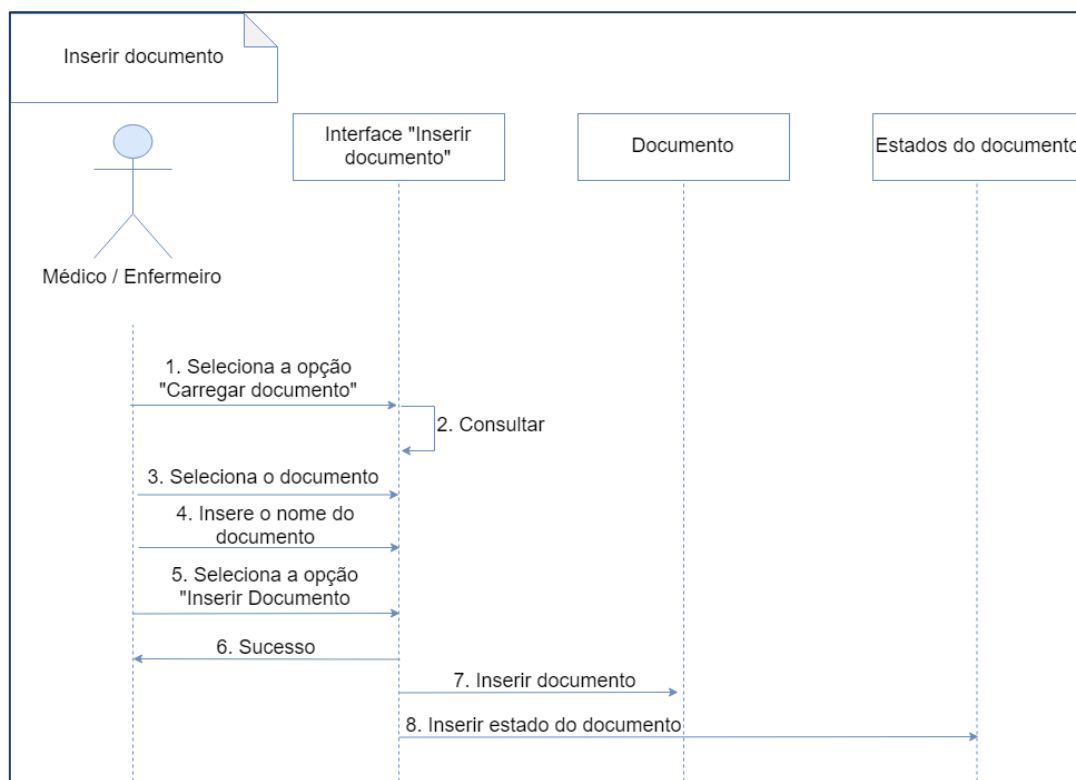


Figura 16 - Diagrama de Sequência "Inserir documento" do sistema SoftDigital

Um documento pode ter vários estados possíveis: é necessário assinar, a aguardar por outros e assinado. Quando o médico insere um documento na aplicação, este documento assume o estado é necessário assinar. Na Figura 17, apresenta-se o diagrama de classes da classe Documento e da classe Estados do documento.

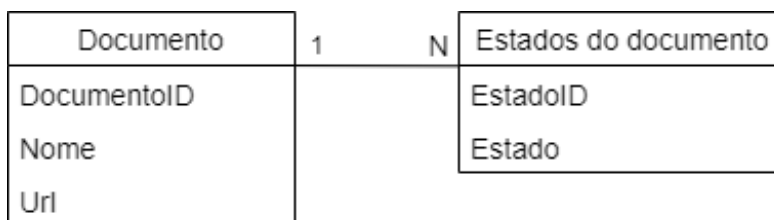


Figura 17 - Diagrama de classes do sistema SoftDigital

Cada classe apresenta os atributos respetivos. A relação 1 para N significa que um documento pode assumir vários estados diferentes.

5. Implementação

Com base no estudo das aplicações existentes no mercado e tendo em conta a análise de requisitos segue-se o desenvolvimento da aplicação móvel SoftDigital.

O objetivo inicial era desenvolver uma aplicação que assinasse documentos PDF através da assinatura digital que se encontra no Cartão de Cidadão português e associar a impressão digital à assinatura digital. Mas devido à não disponibilização de um *middleware*¹¹ para sistemas operativos móveis como o *Android*, encontraram-se dois caminhos alternativos:

- Usar a aplicação móvel da Autenticação.GOV para autenticar e assinar documentos usando a Chave Móvel Digital;
- Assinar os documentos PDF com um *software* como o *DocuSign* ou qualquer outro *software* que permita assinatura digital.

5.1. Tecnologias utilizadas

Neste capítulo será feita a apresentação das tecnologias utilizadas que contribuíram para o desenvolvimento da aplicação:

- Android Studio – plataforma de desenvolvimento para *Android*;
- *Android* – sistema operativo utilizado;
- Java – linguagem de programação utilizada;
- Firebase – armazenar todos os dados relativos à app SoftDigital;
- GitHub – armazenamento de todo o código e registo da evolução da app SoftDigital.

Os tópicos seguintes descrevem uma breve introdução a cada uma delas.

5.1.1. Android Studio

Android Studio é uma plataforma de desenvolvimento integrado (IDE) dedicado à programação em *Android*. Foi lançado em Maio de 2013 na conferência Google I/O¹². Suporta linguagens: Java, Kotlin e C++. Este é disponibilizado gratuitamente sob a Licença *Apache 2.0* – licença de *software* livre permissiva de autoria da *Apache Software Foundation*¹³. A Figura 18, mostra o logótipo do *Android Studio*, do *Android* e do Java, respetivamente.

¹¹ *Software* público e *Open Source* que fornece serviços para outros *softwares*

¹² Conferência de programadores organizada anualmente nos Estados Unidos

¹³ Organização criada para suportar projetos *apache open-source*



Figura 18 - Logótipo Android Studio, do Android e do Java, respetivamente

Por ser uma plataforma dedicada à programação em *Android* e por ter sido lecionada numa unidade curricular da Licenciatura em Engenharia Informática, optou-se por utilizar no presente projeto.

- Android

O *Android* é um sistema operativo móvel baseado no *kernel Linux*. É disponibilizado pela Google sob licença *open-source*, com aplicações com *software* de código fechado. Na Figura 19, apresenta-se a evolução do sistema operativo *Android*, como se pode verificar atualmente encontra-se na versão 9.0 (*Android Pie*). De salientar que as duas primeiras versões: Alpha – *Android* 1.0, lançada em 2007, e a Beta – *Android* 1.1, lançada dois anos depois, não se encontram na Figura 19 pelo facto de a primeira versão a usar oficialmente um nome baseado num doce foi a Cupcake – *Android* 1.5, lançada em 2009.



Figura 19 - Evolução do sistema operativo Android

Fonte: <https://twitter.com/tclapstudios/status/1027003140892225536>

Como a versão que se encontra atualmente é a versão 9.0 (*Android Pie*) e é esta encontra-se no *smartphone* onde a aplicação SoftDigital foi testada, foi assim utilizada neste projeto.

- Linguagem Java

Criada pela *Sun Microsystems*¹⁴ e mais tarde adquirida pela *Oracle Corporation*¹⁵ em 2009, a linguagem Java baseia-se na programação orientada a objetos. A Figura 20,

¹⁴ Fabricante de computadores, semicondutores e *software* localizada nos Estados Unidos da América

¹⁵ Empresa multinacional de tecnologia e informática localizada nos Estados Unidos da América

mostra um exemplo de código em Java aplicado neste projeto. A classe `StatusDocumentsActivity` é a classe que apresenta as abas dos estados do documento. O `TabLayout` é usado para implementar as abas horizontais. `addOnPageChangeListener` (listener) do `ViewPager`, é a responsável pelo conteúdo diferente em cada aba.

```
public class StatusDocumentsActivity extends AppCompatActivity {
    /**
     * Método que executa o programa
     * public = É visto em qualquer lugar da aplicação. É o
    modificador de acesso
     * static = é iniciado automaticamente pela JVM, sem precisar de
    uma instância
     * void = Método sem retorno (retorno vazio)
     * main = Nome do método, que é obrigatório ser este. Recebe
    como parâmetro um array de String.
     * String[] args = Array de argumentos que podem ser repassados
    na chamada do programa.
     */
    TabLayout tabLayout;
    TabItem AllDocuments_TabItem, NecessarySign_TabItem,
    WaitingSignatures_TabItem, Done_TabItem;
    ViewPager viewPager;
    PageAdapter pageAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        (...)
        Done_TabItem = findViewById(R.id.Done_TabItem);
        viewPager = findViewById(R.id.viewPager);
        pageAdapter = new PageAdapter(getSupportFragmentManager(),
        tabLayout.getTabCount());
        viewPager.setAdapter(pageAdapter);
        viewPager.addOnPageChangeListener(new
        TabLayout.TabLayoutOnPageChangeListener(tabLayout));
        (...)
    }
}
```

Figura 20 - Exemplo de código em Java

Optou-se neste projeto utilizar a linguagem Java devido ao facto da aquisição de conhecimentos durante o curso de Engenharia Informática.

5.1.2. Firebase

O *Firebase* é uma plataforma de desenvolvimento de aplicações para dispositivos móveis e para web desenvolvida pela *Firebase*, adquirida pela Google em 2014. Na Figura 21, apresenta-se o logótipo do *Firebase*, *Firebase Authentication*, *Firebase Realtime Database* e *Firebase Cloud Storage*, respetivamente .



Figura 21 - Logótipo do Firebase, Firebase Authentication, Realtime Database e Cloud Storage

A escolha do *Firebase* surgiu pelo facto de se tratar de uma plataforma de aplicações para dispositivos móveis, usar um serviço de autenticação, uma base de dados em tempo real e uma *cloud storage* para armazenar todos dados relativos à aplicação SoftDigital. Segue-se uma breve explicação de alguns serviços utilizados e a aplicabilidade na app SoftDigital.

- *Firebase Authentication*

O *Firebase Authentication* fornece serviços de *back-end*, SDKs fáceis de usar e bibliotecas de UI prontas para autenticar utilizadores na aplicação [7].

Na aplicação móvel SoftDigital, o *Firebase Authentication* é usado para registar e autenticar os médicos/ enfermeiros e utentes/ representante legal.

- *Firebase Realtime Database*

O *Firebase Realtime Database* é uma base de dados alojada numa *Cloud*. Os dados são armazenados como JSON e sincronizados em tempo real [8].

Na aplicação móvel SoftDigital, o *Firebase Realtime Database* é usado para armazenar todos os dados relativos: aos médicos/ enfermeiros e utentes/ representante legal; documentos inseridos na app; estados pelo qual um documento transita; documentos enviados.

- *Cloud Storage*

O *cloud storage* para o *Firebase* é um serviço de armazenamento de objetos. Com os SDKs do *Firebase* para a *cloud storage*, é possível utilizar a segurança do *Google* para fazer *upload* e *download* de ficheiros nos aplicativos do *Firebase* [9].

Na aplicação móvel SoftDigital, o *cloud storage* é usado para armazenar os documentos inseridos na app.

5.1.3. GitHub

O *GitHub* é uma plataforma de controlo de versões grátis e *open source* que usa o *Git*. Na Figura 22, demonstra-se o logótipo do *GitHub*.



Figura 22 - Logótipo GitHub

Todos os ficheiros de um projeto ficam armazenados num repositório no *GitHub*, é possível adicionar alterações ao repositório (*commits*) e por isso criar diferentes versões ao longo do desenvolvimento de um projeto. Entre muitas outras funcionalidades que o *GitHub* disponibiliza, é também possível criar uma cópia do repositório para o desenvolvimento em paralelo e separadamente (*branch*). Depois de finalizado o desenvolvimento na *branch* é possível unir a outra *branch* através de um *merge*, ou unir à master através de um *pull request* (é como informar que irá implementar alterações da *branch* ao repositório *master*). Este tipo de plataformas é uma solução para o

desenvolvimento em equipas de trabalho para controlo e gestão de versões e revisão de projetos.

No presente projeto, o GitHub foi uma solução para armazenar todo código e registo da sua evolução. Consultável em <https://github.com/BarbaraOvelheiro/SoftDigital>.

5.2. Arquitetura do sistema

A arquitetura do sistema SoftDigital consiste em infraestruturas como o Firebase Authentication que autentica os utilizadores, o Firebase Realtime Database e o Firebase Cloud Storage que são as bases de dados que armazenam todos os dados do sistema SoftDigital. A arquitetura do sistema SoftDigital encontra-se ilustrada na Figura 23.

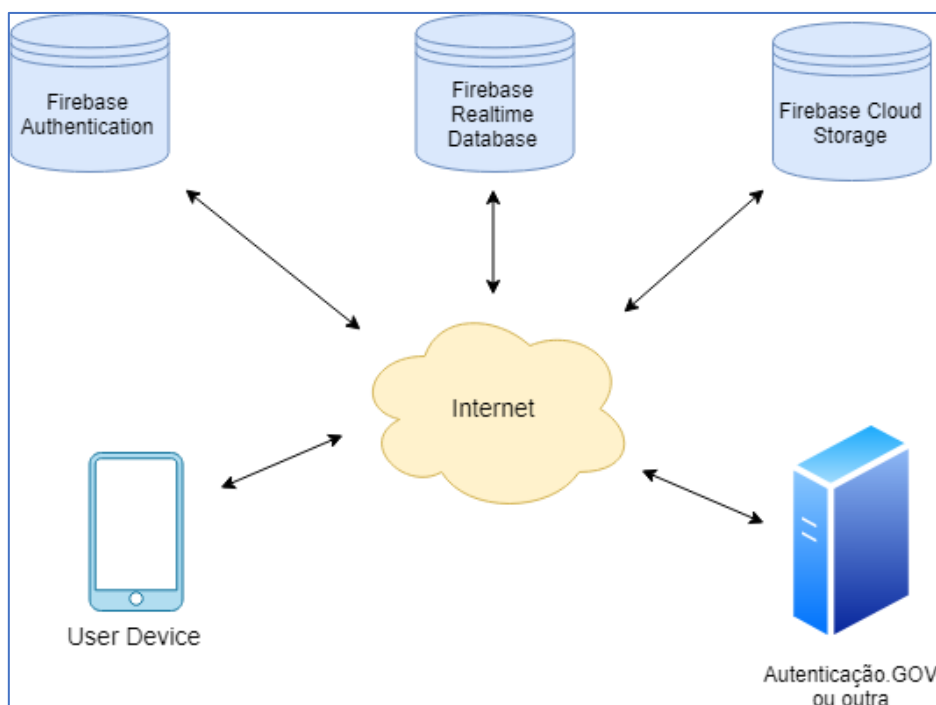


Figura 23 -Diagrama de arquitetura do sistema SoftDigital

O *User Device*, é o ator do sistema SoftDigital que através de um dispositivo móvel comunica com a infraestrutura. A infraestrutura Autenticação.GOV ou outro é responsável pela assinatura digital qualificada.

É possível observar como fazer uma assinatura digital na aplicação SoftDigital. O processo para a realização de uma assinatura digital encontra-se ilustrado na Figura 24, e de seguida a explicação de cada passo. O *Document Creator Device* é o ator do sistema SoftDigital que através de um dispositivo móvel insere um documento para assinatura na aplicação e também assina, que é feito pelo médico ou enfermeiro. O *Signer Device* é o ator do sistema que através de um dispositivo móvel assina o documento, que é feito pelo Paciente ou Representante Legal.

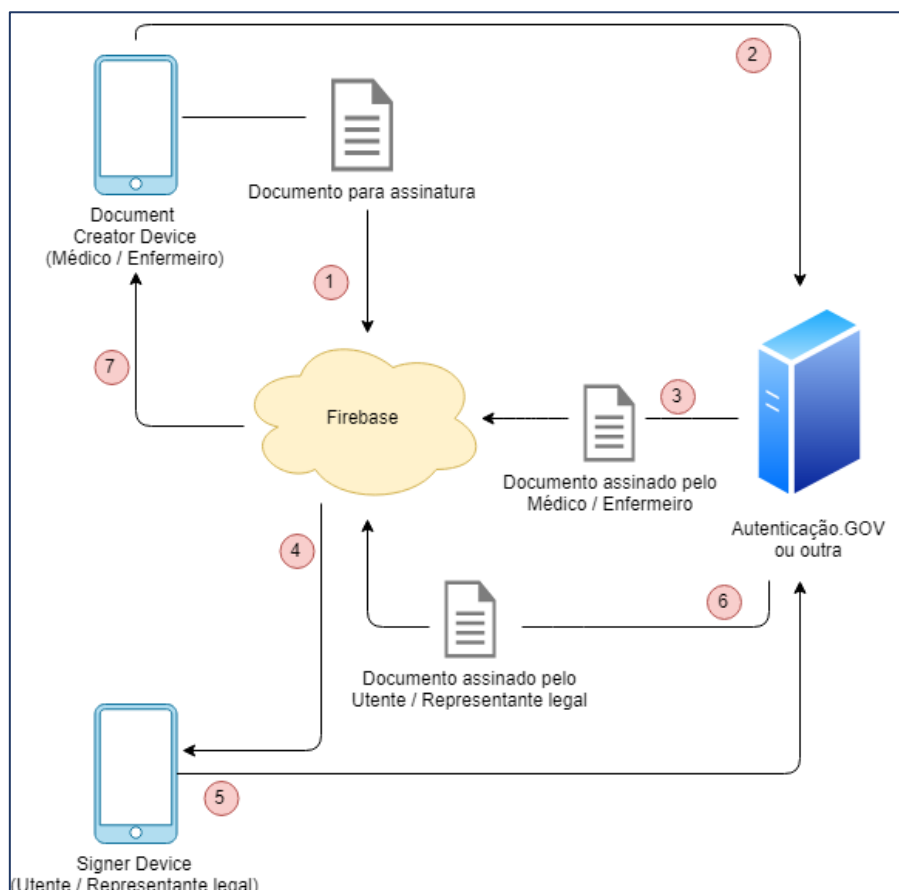


Figura 24 - Diagrama de passos para a assinatura digital no sistema SoftDigital

Passos para a realização de uma assinatura digital. Desde inserir o documento com dispositivo móvel até receber no mesmo dispositivo o documento assinado:

- 1– O Médico insere o documento na aplicação;
- 2– O Médico assina o documento com a Autenticação.GOV ou outro e envia-o assinado para o Utente;
- 3– O *Firebase* armazena o documento assinado;
- 4– O Utente recebe o documento para assinar na aplicação;
- 5– O Utente assina o documento com a Autenticação.GOV ou outro e envia-o para o Médico;
- 6– O *Firebase* armazena o documento assinado;
- 7– O Médico recebe o documento assinado na aplicação.

5.3. Estrutura da aplicação

Os projetos no *Android Studio* contêm tudo o que define o espaço de trabalho para uma app, desde o código de fonte e recursos ao código de teste e configuração de compilação. Cada vez que é criado um novo projeto, o *Android Studio* cria a estrutura necessária para todos os ficheiros envolventes.

Em cada módulo de app *Android*, os ficheiros são agrupados em três pastas principais:

- A pasta **manifest** contém um ficheiro que tem todas as definições e configurações da app;
- A pasta **java** contém os ficheiros de código fonte Java, separados por nome, atividades, fragmentos e classes, incluindo também o código de teste JUnit¹⁶;
- A pasta **res** contém os recursos que não são código, como layouts XML, cadeias de UI e imagens de bitmap divididos em subpastas correspondentes.

Na Figura 25, apresenta-se todos os módulos da app *Android SoftDigital*.

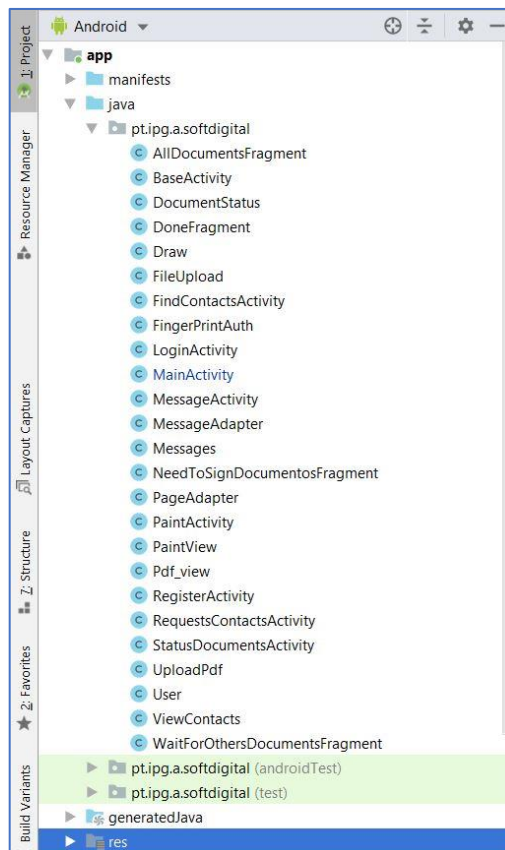


Figura 25 - Módulos da app *Android SoftDigital*

Nos tópicos seguintes, serão apresentados e explicados os módulos mais importantes da app *Android SoftDigital*.

¹⁶ Framework open-source com suporte à criação de testes automatizados na linguagem de programação em Java

5.3.1. Módulo FileUpload

Esta classe é responsável por inserir um documento na aplicação SoftDigital. Quando o utilizador selecciona o documento, é retornado o URI desse documento e é guardado na variável **pdfUri**. O URI vai ser utilizado para enviar o documento para o Firebase. O **currentUserID** é o utilizador que tem sessão iniciada. O **documentID** é o ID do documento e o **statusID** é o ID do estado. O **editPdfName** é a que contém o nome do documento introduzido pelo utilizador. Quando o documento é carregado para o Firebase, o estado do documento passa a é preciso assinar. A variável **messageID** encontra-se a null pelo facto de não existir documentos enviados antes de o inserir.

Na Figura 26, apresenta-se o código para inserir um documento para assinatura digital.

```
(...)
if(requestCode == 1 && resultCode == RESULT_OK && data != null){
    pdfUri = data.getData(); // retorna o uri do ficheiro
    selecionado
    (...)
    private void uploadFile(Uri pdfUri){
        (...)
        StorageReference reference =
        storageReference.child("Files"+System.currentTimeMillis()+".pdf");

        reference.putFile(pdfUri).addOnSuccessListener(new
        OnSuccessListener<UploadTask.TaskSnapshot>() {
            (...)
            Task<Uri> uri = taskSnapshot.getStorage().getDownloadUrl();
            while (!uri.isComplete());
            Uri url = uri.getResult();

            final String messageSenderRef = "Files/" + currentUserID;
            final String documentID = FilesRef.push().getKey();
            final String statusID = documentStatus.push().getKey();

            UploadPdf information = new
            UploadPdf(documentID, editPdfName.getText().toString(), url.toString()
            , currentUserID, statusID);

            FirebaseDatabase.getInstance().getReference().child(messageSenderRef)
            .child(documentID).setValue(information).addOnCompleteListener(new
            OnCompleteListener<Void>() {
                (...)
                String statusIN = "e_necessario_assinar";
                String messageID = null;

                DocumentStatus infoStatus = new DocumentStatus(statusIN, statusID,
                documentID, currentUserID, messageID);

                documentStatus.child(currentUserID).child(statusID).setValue(infoSta
                tus).addOnCompleteListener(new OnCompleteListener<Void>() {
                    (...)
                    if(task.isSuccessful()) {
                        Toast.makeText(FileUpload.this, R.string.upload_file_success,
                        Toast.LENGTH_SHORT).show();
                    }
                    (...)
                }
            }
        }
    }
}
```

Figura 26 - Extrato de código do módulo FileUpload: inserir documento para assinatura

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/FileUpload.java>

A Figura 27, mostra a interface Inserir documento na app SoftDigital.

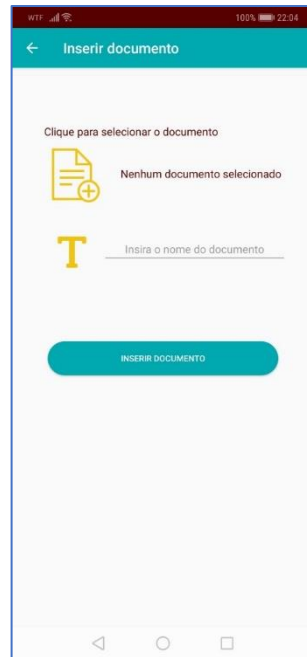


Figura 27 - Interface Inserir documento na app SoftDigital

- Classe UploadPdf

Faz referência à tabela Files que contém a informação de cada documento. Na Figura 28, mostra os atributos da tabela Files.

```
public class UploadPdf {  
  
    public String documentID;  
    public String documentName;  
    public String documentUrl;  
    public String userID;  
    public String statusID;  
  
    public UploadPdf () {  
        (...)  
    }  
}
```

Figura 28 - Extrato do código da classe UploadPdf

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/UploadPdf.java>

A Figura 29, mostra o aspeto da tabela Files no *Firebase Database*.



Figura 29 - Aspeto da tabela Files no *Firebase Database*

Como se pode observar, o nó Files contém um nó filho que é representado pelo ID do utilizador que inseriu o documento, este nó contém um nó filho que é o ID do documento inserido e este nó é também constituído pelos nós filhos que são os atributos da tabela Files.

- Classe DocumentStatus

Faz referência à tabela Document Status que contém a informação de cada estado do documento. Na Figura 30, mostra os atributos da tabela Document Status.

```
public class DocumentStatus {

    private String status;
    private String statusID;
    private String documentID;
    private String userID;
    private String messageID;

    (...)
}
```

Figura 30 - Extrato do código da classe DocumentStatus

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/DocumentStatus.java>

A Figura 31, mostra o aspeto da tabela Document Status no *Firebase Database*.



Figura 31 - Aspeto da tabela Document Status no *Firebase Database*

Como se pode observar, o nó Document Status contém dois nós filhos que são representados: um pelo ID do utilizador que inseriu o documento; outro pelo ID do utilizador que recebeu um documento para assinatura. O primeiro nó filho contém um nó

filho que é o ID do estado e este nó é também constituído pelos nós filhos que são os atributos da tabela Document Status.

5.3.2. Módulo AllDocumentsFragment

Este fragmento é responsável por mostrar todos os documentos, desde a documentos por assinar a documentos a aguardar por outros. Para tal, recorreu-se a uma *Recycler View* para listar todos os documentos. É possível saber quais os documentos que estão para assinatura ou a aguardar por outros através do atributo status da tabela Document Status. A Figura 32, apresenta o código de como distinguir os estados possíveis. A variável **status** guarda o valor do estado que se encontra na tabela Document Status no *Firebase Database* para um determinado documento.

```
(...)  
FirebaseRecyclerOptions options =  
    new FirebaseRecyclerOptions.Builder<DocumentStatus>()  
        .setQuery(documentStatus, DocumentStatus.class)  
        .build();  
FirebaseRecyclerAdapter<DocumentStatus, AllDocumentsViewHolder>  
adapter  
    = new FirebaseRecyclerAdapter<DocumentStatus,  
AllDocumentsViewHolder>(options) {  
    @Override  
    protected void onBindViewHolder(@NonNull final  
AllDocumentsViewHolder holder, final int position, @NonNull final  
DocumentStatus model) {  
    (...)  
    String status = model.getStatus();  
    if (status.equals("e_necessario_assinar")) {  
        holder.fromtextView.setVisibility(View.VISIBLE);  
        holder.send_userName_editText.setVisibility(View.VISIBLE);  
        holder.fromtextView.setText("Estado: ");  
        holder.send_userName_editText.setText("é necessário assinar");  
    } else if (status.equals("a_aguardar_por_outros")) {  
        holder.fromtextView.setVisibility(View.VISIBLE);  
        holder.send_userName_editText.setVisibility(View.VISIBLE);  
        holder.fromtextView.setText("Estado: ");  
        holder.send_userName_editText.setText("a aguardar por outros");  
    }  
    (...)  
}
```

Figura 32 - Extrato de código do módulo AllDocumentsFragment

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/AllDocumentsFragment.java>

A Figura 33, apresenta a interface Todos Os Documentos na APP SoftDigital.



Figura 33 - Interface Todos os Documentos na app SoftDigital

5.3.3. Módulo PdfViewer

Esta classe apresenta o modo como pode ser visualizado um documento PDF numa aplicação *Android* sem necessitar de recorrer a aplicações externas. Para tal, foi usado uma *lib* Android PdfViewer: biblioteca para abrir um documento PDF no *Android*. Licenciado sobre a Licença Apache 2.0.

- XML activity_pdf_view

A Figura 34 mostra como incluir o PDFView no layout.

```
<com.github.barteksc.pdfviewer.PDFView
    android:id="@+id/pdfView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
/>
```

Figura 34 - Extrato do código do XML activity_pdf_view

Este módulo encontra-se disponível no GitHub. Consultável em:

https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/res/layout/activity_view__pdf__files.xml

A Figura 35 apresenta como abrir um documento na aplicação. A variável **pdfURL** contém o URI do documento que se encontra no Firebase Cloud Storage. A subclasse **InputStream** fornece um método que retorna o próximo byte de entrada. A **URLConnection** suporta recursos específicos de HTTP.

```
(...)  
try {  
    URL url = new URL(pdfURL);  
    new RetrievePDF().execute(pdfURL);  
} catch (MalformedURLException e) {  
    e.printStackTrace();  
}  
(...)  
class RetrievePDF extends AsyncTask<String, Void, InputStream> {  
    @Override  
    protected InputStream doInBackground(String... strings) {  
        InputStream = null;  
        try {  
            URL stringURL = new URL(strings[0]);  
            HttpURLConnection urlConnection = (HttpURLConnection)  
stringURL.openConnection();  
            if (urlConnection.getResponseCode() == 200) {  
                inputStream = new  
BufferedInputStream(urlConnection.getInputStream());  
            } catch (IOException e) {  
                return null;  
            }  
            return inputStream;  
        }  
        @Override  
        protected void onPostExecute(InputStream inputStream) {  
            mPDFView.fromStream(inputStream).load(); // mostrar o pdf  
        }  
(...)
```

Figura 35 - Extrato do código da classe PdfViewer

Este módulo encontra-se disponível no GitHub. Consultável em:

https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/Pdf_view.java

A Figura 36 mostra a interface do documento para ser assinado na app SoftDigital.

Figura 36 - Interface “Assinar Documento” na app SoftDigital

5.3.4. Módulo NeedToSignDocumentsFragment

Neste fragmento, foi criada uma lista de documentos PDF que correspondem ao estado: é necessário assinar. Para tal, recorreu-se a uma *Recycler View* para listar os documentos. Na Figura 37, apresenta-se o código de como listar documentos alojados no *Firebase Database* na tabela Document Status, cujo estado corresponde a: é necessário assinar. A variável **status** contém o estado relativo ao documento. Só é possível porque no *Recycler Adapter* é feita a referência à tabela Document Status.

```
(...)
FirebaseRecyclerOptions options = new
FirebaseRecyclerOptions.Builder<DocumentStatus>()
    .setQuery(documentStatus, DocumentStatus.class).build();

FirebaseRecyclerAdapter<DocumentStatus,
NeedToSignDocumentsFragmentViewHolder> adapter
    = new FirebaseRecyclerAdapter<DocumentStatus,
NeedToSignDocumentsFragmentViewHolder>(options) {
    @Override
    protected void onBindViewHolder(@NonNull final
NeedToSignDocumentsFragmentViewHolder holder, final int position,
@NonNull final DocumentStatus model) {

        String messageID = model.getMessageID();
        String status = model.getStatus();
        Log.v(TAG, "status: " +status);
        if(status.equals("e_necessario_assinar")) {

            if (messageID == null) {
                (...)
            }
        }
    }
}
```

Figura 37 - Extrato de uma parte de código da classe *NeedToSignDocumentsFragment*

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/NeedToSignDocumentosFragment.java>

- Notificação com documento para assinatura. Para enviar uma notificação com documento para assinatura, utiliza-se o código que se encontra na Figura 38.

```
private void displayNotification() {
    NotificationCompat.Builder mBuilder = new
NotificationCompat.Builder(getActivity(), CHANNEL_ID)
        .setSmallIcon(R.mipmap.assinatura)
        .setContentTitle("Novo Documento")
        .setContentText("Recebeu um documento para assinatura")
        .setPriority(NotificationCompat.PRIORITY_DEFAULT);

    NotificationManagerCompat =
NotificationManagerCompat.from(getActivity());
    notificationManagerCompat.notify(1, mBuilder.build());
}
```

Figura 38 - Código para enviar uma notificação

A Figura 39 mostra um *ScreenShot* de uma notificação recebida no dispositivo móvel. No caso de ser o médico a enviar um documento para o paciente assinar, o paciente quando iniciar sessão na app SoftDigital irá receber uma notificação com um documento para assinatura.

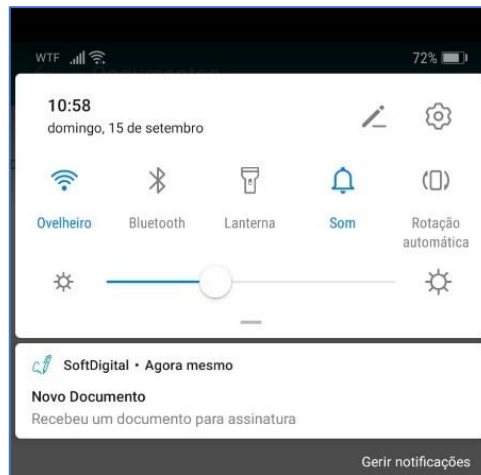


Figura 39 - SreenShot de uma notificação recebida no dispositivo móvel

5.3.5. Módulo MessageActivity

Na Figura 40, apresenta-se o código que insere os atributos na tabela Messages. O atributo **messageID** é o ID da mensagem, o **documentName** é o nome do documento a enviar, **documentoUrl** é onde está alojado o documento a enviar, **currentUserID** é o utilizador com a sessão iniciada, **userReceiverID** é o utilizador que irá receber o documento para assinatura, **receiverStatusID** é o estado do documento a enviar e o **receiverPdfID** é o ID do documento a enviar para o destinatário.

```
(...)
Map messageText = new HashMap();
messageText.put("messageID", messagePushID);
messageText.put("documentName", messageDocumentName);
messageText.put("documentUrl", pdfURL);
messageText.put("userFromID", currentUserID);
messageText.put("userTOID", userReceiverID);
messageText.put("statusID", receiverStatusID);
messageText.put("documentID", receiverPdfID);

Map messageDetails = new HashMap();
messageDetails.put(messageSenderRef + "/" + messagePushID,
messageText);
RootRef.updateChildren(messageDetails).addOnCompleteListener(new
OnCompleteListener() {
(...)
    if(task.isSuccessful()) {
        Toast.makeText(MessageActivity.this,
R.string.sucessful_send_message, Toast.LENGTH_SHORT).show();
    }else{
        Toast.makeText(MessageActivity.this,
R.string.error_send_message, Toast.LENGTH_SHORT).show();
    }
})
}
```

Figura 40 - Extrato do código da classe MessageActivity

Este módulo encontra-se disponível no GitHub. Consultável em:

<https://github.com/BarbaraOvelheiro/SoftDigital/blob/master/app/src/main/java/pt/ipg/a/softdigital/MessageActivity.java>

A Figura 41 mostra a interface Enviar para destinatário na app SoftDigital.

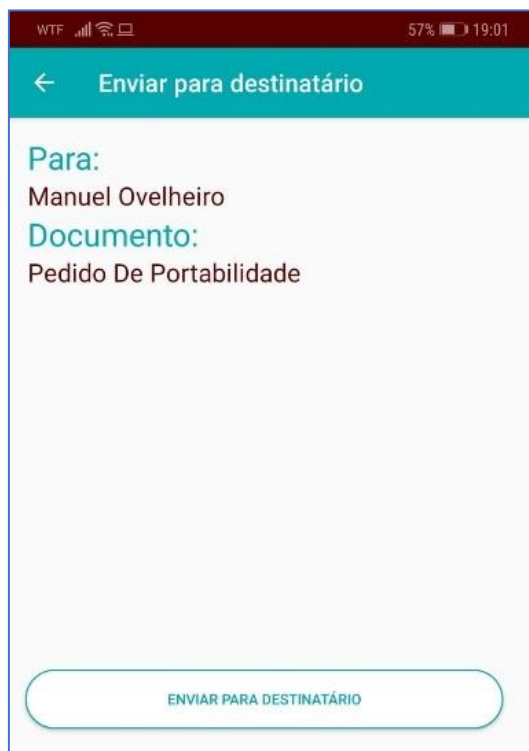


Figura 41 - Interface “Enviar para destinatário” na app SoftDigital

- Classe Messages

Faz referência à tabela Messages que contém a informação de cada documento enviado para assinatura. A Figura 42 mostra os atributos da tabela Messages.

```
public class Messages {
    private String messageID, documentName,
    documentUrl, userFromID, userToID, statusID,
    documentID;
    (...)
}
```

Figura 42 - Extrato do código da classe Messages

A Figura 43 mostra o aspeto da tabela Messages no *Firestore Database*. Como se pode observar, o nó Messages contém um nó filho que é representado pelo ID da mensagem enviada, este nó é constituído pelos nós filhos que são os atributos da tabela Messages.



Figura 43 - Aspeto da tabela Messages no *Firestore Database*

6. Verificação e validação

A verificação e validação são processos independentes que são utilizados em conjunto para verificar se um produto, serviço ou sistema cumpre os requisitos e as especificações e que são adequados aos seus propósitos. A verificação confirma os testes e com provas objetivas que os requisitos especificados foram implementados. A validação confirma por testes e com provas objetivas que o *software* desenvolvido está de acordo com o que o cliente/ utilizador pretende. Para tal, no final da fase de desenvolvimento surge a necessidade de realização de testes de forma a verificar se a solução está de acordo com os requisitos.

6.1. Casos de teste

Cada caso de uso será testado de forma a garantir que todas as funcionalidades foram implementadas. Os casos de uso testados foram os seguintes: inserir documento para assinatura digital; assinar digitalmente o documento; consultar documentos. Os testes realizados só poderão ter dois resultados: **passou** – assinalado com **V** nas tabelas de casos de testes, os testes que passaram no respetivo; **não passou** – assinalado com **X** nas tabelas de casos de testes, os testes que não passaram no respetivo. Para a realização dos testes executados foi usado um *smartphone Android*: versão do SDK – 28; marca – Huawei Mate 20 Pro; CPU – Huawei Kirin 980; Memória RAM – 6.0GB. Na Tabela 7 apresenta-se o caso de teste do Consultar documentos.

Tabela 7 - Caso de teste “Consultar documentos”

Caso de teste do Consultar documentos						
ID	Entrada	Observações	Resultados Esperados	Resultados atuais	V	X
1	Consultar documentos	O utilizador selecionou a opção consultar documentos	Apresentar todos os documentos	Apresenta todos os documentos	V	
2	Estado “a aguardar por outros”	O utilizador selecionou o estado “a aguardar por outros”	Apresentar o documento no estado “a aguardar por outros”	Apresenta o documento no estado “a aguardar por outros”	V	
3	Estado “é preciso assinar”	O utilizador selecionou o estado “é preciso assinar”	Apresentar o documento no estado “é preciso assinar”	Apresenta o documento no estado “é preciso assinar”	V	
4	Todos os documentos	O utilizador selecionou todos os documentos	Apresentar todos os documentos e respetivos estados	Apresenta todos os documentos e respetivos estados	V	

Ao executar o caso de teste Consultar documentos é possível verificar que todos passaram com sucesso.

Na Tabela 8, apresenta-se o caso de teste do Assinar digitalmente o documento.

Tabela 8 - Caso de teste “Assinar digitalmente o documento”

Caso de teste do Assinar digitalmente o documento						
ID	Entrada	Observações	Resultados Esperados	Resultados atuais	V	X
1	Botão Assinar documento	O utilizador selecionou a opção Assinar documento	Apresenta os documentos que é necessário assinar	Apresenta os documentos que é necessário assinar	V	
2	Documento para assinar	O utilizador selecionou o documento para assinar	Abrir o documento selecionado	Abriu documento selecionado	V	
3	Ícone assinar	O utilizador selecionou o ícone assinar	Interação com o <i>software</i> Autenticação.GOV	Nenhuma interação com o <i>software</i>		X
4	Submeter documento	O utilizador selecionou o ícone partilhar	Redirecionar para a interface Contactos	Redireciona para a interface Contactos	V	

Ao executar este caso de teste é possível observar que existe um deles que falhou, nomeadamente o ID3. Devido ao facto de ainda não haver nenhuma integração ao *software* Autenticação.GOV. Os restantes foram testados com sucesso.

Na Tabela 9, apresenta-se o caso de teste do Inserir documento para assinatura digital. Ao executar este caso de teste é possível verificar que todos passaram com sucesso.

Tabela 9 - Caso de teste “Inserir documento para assinatura digital”

Caso de teste do Inserir documento para assinatura digital						
ID	Entrada	Observações	Resultados Esperados	Resultados atuais	V	X
1	Documento.pdf	O documento é selecionado pelo utilizador	Devolve o url do documento.pdf	Devolve o url do documento.pdf	V	
2	Nenhum documento selecionado	O utilizador não selecionou nenhum documento	Mensagem do tipo “Selecione um documento”	Devolve mensagem do tipo “Selecione um documento”	V	
3	Nome do documento	O utilizador introduz um nome ao documento	-	-	V	
4	Nenhum nome do documento introduzido	O utilizador não introduziu nenhum nome ao documento	Mensagem de campo obrigatório	Mensagem de campo obrigatório	V	
5	Nenhum documento selecionado e nenhum nome introduzido	O utilizador não selecionou nenhum documento e não introduziu nenhum nome	Mensagem do tipo “Selecione o documento”	Mensagem do tipo “Selecione o documento”	V	
6	Nenhum documento selecionado e inserido nome do documento	O utilizador não selecionou nenhum documento, mas introduziu um nome	Mensagem do tipo “Selecione o documento”	Mensagem do tipo “Selecione o documento”	V	
7	Documento.pdf e nenhum nome do documento introduzido	O utilizador selecionou um documento e não introduziu nenhum nome	Mensagem de campo obrigatório	Mensagem de campo obrigatório	V	
8	Documento.pdf e nome do documento introduzido	O utilizador selecionou um documento e introduziu um nome ao documento	Mensagem do tipo “Documento carregado com sucesso”	Mensagem do tipo “Documento carregado com sucesso”	V	

6.2. Sumário de resultados dos testes realizados

Na Tabela 10, observa-se um sumário dos resultados obtidos após os testes à aplicação SoftDigital.

Tabela 10 - Sumário de resultados dos testes realizados

Nome do teste	Data	Nº de testes que passaram V	Nº de testes que não passaram X
Teste de Inserir documento para assinatura digital	Agosto	2	0
Teste de Assinar digitalmente documento	Setembro	1	1
Teste de Consultar Documentos	Setembro	2	0
Total		5	1

Como se pode observar na Tabela 10, os testes realizados passaram com sucesso. É de salientar que ao longo do desenvolvimento do código da aplicação SoftDigital foram executados vários testes e por questões de simplicidade não foram apresentadas nem contabilizadas.

7. Conclusão

O grande objetivo do projeto foi a implementação de uma aplicação móvel *Android* SoftDigital que permite ao utilizador o fluxo de documentos para posterior integração com a Autenticação.GOV para assinatura digital qualificada.

A arquitetura do sistema sofreu diversas alterações por restrições operacionais e por falta de tempo. A meta principal seria convergir os dados de assinatura do Cartão de Cidadão com a impressão digital, bem como conceitos de criptografia e segurança devido ao facto de se tratar de dados sensíveis dos funcionários de saúde e pacientes ou representante legal.

Em termos de trabalho futuro, é necessário aceder aos dados do Cartão de Cidadão português em *Android* através da integração do *software* Autenticação.GOV que irá permitir aceder à Chave Móvel Digital para a autenticação de cada utilizador e a assinatura digital qualificada em documentos PDF.

Também para trabalho futuro, de forma a tornar a aplicação mais completa, será a integração da SoftDigital com o *software* Alert® Paper free hospital (ALERT® PFH) para aceder à parte documental do hospital. Este *software* clínico abrange desde a parte documental à integração e revisão de toda a informação relacionada com operações hospitalares, com registo de informação clínica em tempo real [10].

Este projeto possibilitou a aplicação de conhecimentos adquiridos ao longo do curso da Licenciatura de Engenharia Informática, bem como a aquisição de novas experiências e competências que contribuíram para o enriquecimento pessoal e profissional.


Referências

- [1] E. Husni, B. Leksono e M. R. Rosa, “Digital Signature for Contract Signing in Service Commerce,” *International Conference on Technology, Informatics, Management, Engineering & Environment*, p. 6, 7-9 September 2015.
- [2] Agência Para A Modernização Administrativa, IP - Presidência Do Conselho De Ministros, “Autenticação.gov,” AMA, 2019. [Online]. Available: <https://www.autenticacao.gov.pt/cc-assinatura>. [Acedido em 04 Junho 2019].
- [3] Agência Para A Modernização Administrativa, IP - Presidência Do Conselho De Ministros, “app Móvel Autenticação.Gov,” AMA, 2019. [Online]. Available: <https://www.autenticacao.gov.pt/aplicacao-movel-autenticacao.gov>. [Acedido em 04 Junho 2019].
- [4] DocuSign, “Quem Somos,” DocuSign, 2019. [Online]. Available: <https://www.docusign.com.br/quem-somos>. [Acedido em 05 Junho 2019].
- [5] SignEasy, “Company Profile,” SignEasy, 2019. [Online]. Available: <https://signeasy.com/company-profile/>. [Acedido em 17 Junho 2019].
- [6] Robert Bosh GmbH, *Scrum Workshop*, M. Bauerdick e R. Dias, Edits., Guarda, 2019.
- [7] M. C. Silveira, *UML - Unifed Modeling Language*, Guarda, 2019.
- [8] T. Quatrani, “Introduction to the Unified Modeling Language,” *IBM Rational*, 2001.
- [9] Firebase, “Firebase Authentication,” [Online]. Available: <https://firebase.google.com/docs/auth/>. [Acedido em 18 Julho 2019].
- [10] Firebase, “Firebase Realtime Database,” [Online]. Available: <https://firebase.google.com/docs/database?authuser=0>. [Acedido em 18 Julho 2019].
- [11] Firebase, “Cloud Storage,” [Online]. Available: <https://firebase.google.com/docs/storage?authuser=0>. [Acedido em 18 Julho 2019].
- [12] ALERT Life Sciences Computing, S.A., “ALERT® PAPER FREE HOSPITAL (PFH),” ALERT Life Sciences Computing, S.A., 2010-2019. [Online]. Available: <http://www.alert-online.com/pt/pfh>. [Acedido em 4 Junho 2019].

Anexos

Anexo I. Certificado de assinatura pela DocuSign

Neste anexo, apresenta-se a Figura 44 com o certificado de assinatura na aplicação DocuSign e a seguir a descrição de todos os campos do certificado de assinatura .



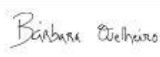
Certificate Of Completion		
Envelope Id: 3C7F695359CD4E218203098B8ED672AF		Status: Completed
Subject: Mod C1002226_New.pdf		
Source Envelope:		
Document Pages: 3	Signatures: 1	Envelope Originator:
Certificate Pages: 1	Initials: 0	Bárbara Ovelheiro
AutoNav: Enabled		bovelheiro@hotmail.com
Envelope Stamping: Disabled		IP Address: 2.80.86.246
Time Zone: (UTC-08:00) Pacific Time (US & Canada)		
Record Tracking		
Status: Original	Holder: Bárbara Ovelheiro	Location: DocuSign
6/15/2019 12:21:42 PM	bovelheiro@hotmail.com	
Signer Events	Signature	Timestamp
Bárbara Ovelheiro		Sent: 6/15/2019 12:21:42 PM
bovelheiro@hotmail.com		Viewed: 6/15/2019 12:21:48 PM
Security Level: Email, Account Authentication (None)	Signature Adoption: Drawn on Device	Signed: 6/15/2019 12:26:04 PM
	Using IP Address: 2.80.86.246	Freeform Signing
	Signed using mobile	
Electronic Record and Signature Disclosure: Not Offered via DocuSign		
In Person Signer Events	Signature	Timestamp
Editor Delivery Events	Status	Timestamp
Agent Delivery Events	Status	Timestamp
Intermediary Delivery Events	Status	Timestamp
Certified Delivery Events	Status	Timestamp
Carbon Copy Events	Status	Timestamp
Witness Events	Signature	Timestamp
Notary Events	Signature	Timestamp
Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	6/15/2019 12:21:42 PM
Certified Delivered	Security Checked	6/15/2019 12:21:48 PM
Signing Complete	Security Checked	6/15/2019 12:26:04 PM
Completed	Security Checked	6/15/2019 12:26:04 PM
Payment Events	Status	Timestamps

Figura 44 - Certificado de assinatura pela aplicação DocuSign

Certificate of Completion – em português, certificado de conclusão:

- *Envelope Id* - é um identificador exclusivo que usa o formato: oito caracteres alfanuméricos seguido de travessão, quatro caracteres alfanuméricos seguido de travessão, quatro caracteres alfanuméricos seguido de travessão e 12 caracteres alfanuméricos (7e5bd754-357b-4f7b-b368-3ec2fc280b56);
- *Subject* – nome do document PDF;
- *Status* – indica o estado o documento (por exemplo, completo);
- *Source Envelope* – indica a origem do envelope;
- *Document Pages* – indica o número de páginas do documento;
- *Certificate Pages* – indica o número de páginas do certificado digital;
- *Signatures* – indica quantas assinaturas têm o documento;
- *Initials* – indica quantas rubricas têm o documento;
- *AutoNav* – controla a maneira como o assinante interage com os documentos, usando o rato, as setas do teclado, deslizar a página para cima;
- *Envelope Originator* – indica quem criou o envelope através do nome e email do utilizador de origem e o IP Address;
- *IP Address* – endereço de IP atribuído a cada dispositivo ligado à rede;
- *Time Zone* – indica o fuso horário.

Record Tracking – registo do acompanhamento da assinatura:

- *Status* – indica o estado do documento quando enviado para o destinatário;
- *Holder* – indica o titular do documento através do nome e email do titular;
- *Location* – indica a aplicação onde o documento foi assinado.

Signer Events – indica o nome e email do assinante e o nível de segurança utilizado.

Signature – apresenta a assinatura e onde capturada, IP Address e em que dispositivo foi utilizado.

Timestamp – é uma sequência de caracteres ou informações codificadas que identificam a data e hora do dia em que um determinado evento ocorreu.

Envelope Summary Events – indica informações de entrega do envelope, entrega do certificado digital, se a assinatura está completa; complementando com o estado que indica se foi encryptado e também indica a data e hora de cada ocorrência.