



Universidade do Minho  
Escola de Engenharia

Paulo Miguel Barbosa Soares

EXTENSÃO DE FUNCIONALIDADES DO  
ERP CHAIN PARA DISPOSITIVOS MOBILE

EXTENSÃO DE FUNCIONALIDADES DO  
ERP CHAIN PARA DISPOSITIVOS MOBILE

Paulo Miguel Barbosa Soares

UMinho | 2014

outubro de 2014





Universidade do Minho  
Escola de Engenharia

Paulo Miguel Barbosa Soares

EXTENSÃO DE FUNCIONALIDADES DO  
ERP CHAIN PARA DISPOSITIVOS MOBILE

Dissertação de Mestrado  
Ciclo de Estudos Integrados Conducentes ao Grau de  
Mestre em Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação do  
Professor Doutor Jorge Oliveira e Sá

Orientador na Empresa:  
Eng. Filipe Carlos Martins Pereira

## DECLARAÇÃO

Nome: **Paulo Miguel Barbosa Soares**

Endereço electrónico: [a51147@alunos.uminho.pt](mailto:a51147@alunos.uminho.pt)

Telefone: 919686102

Número do Bilhete de Identidade: 13395369

Título dissertação

**EXTENSÃO DE FUNCIONALIDADES DO ERP CHAIN PARA DISPOSITIVOS MOBILE**

Orientadores:

Doutor Jorge Oliveira e Sá

Eng. Filipe Carlos Martins Pereira

Ano de conclusão: 2014

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia e Gestão de Sistemas de Informação

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, 16/12/2014

Assinatura: \_\_\_\_\_

## Agradecimentos

Este Projeto de Mestrado apresenta-se como a reta final do meu percurso académico no Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação.

Desta forma, quero agradecer a todos aqueles que fizeram parte, direta ou indiretamente deste percurso e que, de alguma forma, contribuíram para o meu desenvolvimento pessoal e profissional.

Para começar, à minha família, que em todos os momentos me apoiou e me deu formas de chegar ao que sou hoje.

Ao Prof. Doutor Jorge Oliveira e Sá pelos ensinamentos e apoio prestado durante o meu percurso académico e principalmente pela sua orientação na realização deste projeto.

De igual modo, quero agradecer ao Eng. Filipe Carlos Martins Pereira pela confiança depositada ao aceitar a realização deste projeto na sua empresa, Doctron - Sistemas de Informação, Lda., e por toda a orientação e apoio prestado durante a sua realização.

A toda a equipa docente que me acompanhou e facultou conhecimentos ao longo da minha formação, primeiro na Licenciatura em Tecnologias e Sistemas de Informação e agora no Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação.

A todos os meus colegas com quem tive oportunidade de trabalhar e me apoiaram nesta jornada.

A todos quero agradecer e dizer-lhes um humilde e sincero obrigado.

## Resumo

O aparecimento de dispositivos *mobile* veio abrir novos horizontes ao mundo dos negócios. Sendo estes dispositivos cada vez mais potentes, com acesso à internet, a preços cada vez mais baixos e com ecrãs de maior dimensão, permitiram aos seus utilizadores realizar tarefas e obter informação em tempo real e em qualquer lugar. Estas novas possibilidades, proporcionaram uma tomada de decisão mais informada e ponderada.

As empresas, apercebendo-se das potencialidades que um dispositivo simples e comum lhes podia oferecer, focaram-se no desenvolvimento de soluções *mobile* de forma a complementar os seus produtos e serviços e assim obter vantagem competitiva relativamente aos seus concorrentes.

Este trabalho analisa a forma como os concorrentes da Doctron encararam o aparecimento destes dispositivos *mobile* e como iniciaram o desenvolvimento de aplicações *mobile* capazes de fornecer análises de informação e de criar documentos, integrando-os, em tempo real, no seu próprio ERP (*Enterprise Resource Planning*).

Após a análise de diversos fatores, foi desenvolvida uma solução *mobile* em função dos requisitos da empresa. Essa solução é capaz de comunicar com o servidor do CHAIN e de suportar algumas das suas funcionalidades, utilizando dispositivos *mobile*, que se complementam com as funcionalidades oferecidas nativamente por este tipo de dispositivos.

Ao longo deste trabalho foram identificados e compreendidos os conceitos básicos necessários ao desenvolvimento da solução, tendo resultado uma aplicação *mobile*, capaz de estender funcionalidades do ERP CHAIN para dispositivos *mobile*, conferindo a esses dispositivos capacidades idênticas às de um computador com a vantagem de serem móveis.

**Palavras-Chave:** Desenvolvimento; Extensibilidade; Sistema Integrado de Gestão Empresarial; Mobilidade; Soluções Móveis;

## Abstract

New horizons were open for the business world, with the appearing of increasingly powerful devices, with internet access, with ever lower prices, larger screens and allowing users to perform tasks and get real-time information. This allows them a more informed decision making and weighted decisions.

Companies, realizing the potential that a simple device could offer to them, such common device nowadays, led them to focus on mobile developing to complement its product and service solutions in order to gain competitive advantage over their competitors.

This project will discuss how Doctron's competitors faced the emergence of these mobile devices and how they began developing mobile applications capable of providing analysis from information and create documents, integrating them in real time, on their own ERP.

After analyzing various factors, a mobile solution was developed, according to the requirements of the company. The solution is able to communicate with CHAIN's server and support some of its features, by using mobile devices, complementing them with functionalities offered natively by this type of devices.

Throughout this work, were identified and understood the basics needed for the development of the solution, resulting in a mobile application, able to extend some of the ERP CHAIN functionalities to mobile devices, conferring to such devices, identical capabilities of computer with the advantage of being mobile.

**Keywords:** Development; Extensions; Enterprise Resource Planning; Mobility; Mobile Solutions;

# Índice

Agradecimentos .....	iii
Resumo.....	iv
Abstract .....	v
Índice de Figuras .....	viii
Índice de Tabelas.....	ix
1 – Introdução .....	1
1.1 - Descrição do Problema .....	1
1.2 - Objetivos .....	3
1.3 - Abordagem Metodológica .....	5
1.4 - Estrutura do Projeto.....	5
2 – Plano do Projeto .....	8
2.1 - Fases do Projeto .....	9
2.2 - Plano de Atividades.....	10
2.3 - Análise de Riscos .....	14
2.4 – Resumo do Capítulo .....	17
3 – Fundamentação Teórica.....	18
3.1 Mapa de Conhecimento.....	18
3.2 Abordagem de Desenvolvimento.....	19
3.3 Plataformas .....	25
3.3.1 Android.....	25
3.3.2 IOS .....	26
3.3.3 Windows Phone .....	26
3.3.4 Comparação das plataformas .....	27
3.4 Ferramentas .....	30
3.5 Linguagem .....	32
3.6 Web Services .....	34
3.6.1 SOAP .....	36
3.6.2 RESTful.....	39
3.6.3 RESTful VS SOAP .....	40
3.7 – <i>Unified Modeling Language (UML)</i> .....	42
3.8 – Metodologia Scrum .....	43
3.8.1 <i>A Framework</i> .....	44
3.9 – Resumo do Capítulo .....	49
4 – Contexto da Empresa e Enquadramento do Produto .....	51

4.1 - Descrição da Empresa .....	51
4.2 - Descrição do Produto .....	51
4.2.1 – Análise da Concorrência .....	56
4.2.2 – Posicionamento do Produto.....	60
4.3 – Resumo do Capítulo .....	63
5 – Desenvolvimento do Produto .....	65
5.1 Conceção do Produto.....	65
5.1.1 Modelação UML .....	65
5.1.2 Definição da Interface Gráfica do Utilizador (GUI).....	76
5.1.3 Documentação .....	78
5.2 Programação do Produto .....	80
5.2.1 Aplicação da <i>Framework</i> Scrum .....	80
5.3 Resumo do Capítulo .....	84
6 – Resultados .....	85
6.1 Impacto do Produto .....	85
6.1.1 Na organização .....	85
6.1.2 No cliente .....	85
6.2 Resumo do Capítulo .....	86
7 – Conclusão .....	87
7.1 Revisitar os Objetivos.....	88
7.2 Oportunidades .....	89
7.3 Trabalhos Futuros.....	91
Referências Bibliográficas .....	92
Anexos	
Anexo A - Modelação UML	
Anexo B - Interface Gráfico do Utilizador e Manual Técnico	
Anexo C - Complemento Scrum	

## Índice de Figuras

Figura 1 - Fases do Projeto .....	9
Figura 2 - Plano de Atividades.....	10
Figura 3 - Mapa de Conhecimento.....	19
Figura 4 - Mapa de Conhecimento - Abordagens de Desenvolvimento .....	19
Figura 5 - Mapa de Conhecimento - Plataformas Mobile .....	25
Figura 6 - Média do preço dos dispositivos por plataforma a nível global, retirado de (IDC, 2014) .....	28
Figura 7 - Mapa de conhecimento - Ferramentas de desenvolvimento.....	30
Figura 8 - Mapa de conhecimento - Linguagens de desenvolvimento .....	32
Figura 9 - Modelo conceptual da Linguagem JAVA retirado de (ORACLE, 2014).....	33
Figura 10 – Top 30 popularidade das linguagens de programação retirado de (IEEE Spectrum, 2014) .....	34
Figura 11 - Mapa de conhecimento - Web Services.....	35
Figura 12 - Estrutura da mensagem SOAP .....	37
Figura 13 - Arquitetura SOAP adaptada de (Mumbaikar & Padiya, 2013).....	38
Figura 14 - Mapa de conhecimento - UML.....	42
Figura 15 - Mapa de conhecimento - Scrum .....	43
Figura 16 - Processo Scrum retirada de (Scrum Alliance, 2014) .....	44
Figura 17 - Exemplo de um documento (CHAIN) .....	52
Figura 18 - Controlo de Custos (CHAIN) .....	53
Figura 19 - Interface dos Artigos (CHAIN) .....	54
Figura 20 - Interface de Seleção de Família (CHAIN).....	55
Figura 21 - Manutenção de Tabelas (CHAIN) .....	55
Figura 22 - Processo de Marketing adaptado de (Kotler & Armstrong , 2013).....	62
Figura 23 - Sistema CHAIN Mobile (UML-UC) .....	65
Figura 24 - Subsistema de Manuseamento de Documentos (UML-UC) .....	66
Figura 25 - Visualizar Documento (UML-DA) .....	67
Figura 26 - O utilizador consegue visualizar os documentos com sucesso (UML-DS) .....	68
Figura 27 - O utilizador não consegue visualiza os documentos por falta de acesso à internet (UML-DS) .....	68
Figura 28 - Enviar Documento (UML-DA).....	69
Figura 29 - O utilizador anexa um documento com sucesso (UML-DS).....	70
Figura 30 - O utilizador não consegue anexar um documento por falta de internet (UML-DS). 70	
Figura 31 - Criar Documento (UML-DA) .....	72
Figura 32 - Criar documento com um artigo associado (UML-DS).....	73
Figura 33 - Criar Documento com dois artigos (UML-DS) .....	74
Figura 34 - Remover um artigo de um potencial documento (UML-DS) .....	75
Figura 35 - Sem acesso à internet para aceder a documentos (UML-DS).....	75
Figura 36 - Falha a associar um artigo a um documento (UML-DS).....	76
Figura 37 - Menu login (Android-GUI).....	77
Figura 38 - Menu login (IOS-GUI) .....	77
Figura 39 - Menu login (Windows Phone-GUI) .....	78
Figura 40 - Menu principal (Android-GUI).....	78
Figura 41 - Menu Terceiros (Android-GUI).....	79
Figura 42 - Criar Terceiro (GUI) .....	80

## Índice de Tabelas

Tabela 1 - Análise de Riscos .....	14
Tabela 2- Fatores de Decisão Abordagens de Desenvolvimento adaptado de (Dalmasso, Datta, Bonnet, & Nikaein, 2013) .....	24
Tabela 3 - Estudo Sistemas Operativos Mobile 2012-2013 retirado de (IDC, 2014).....	27
Tabela 4 - Comparação Técnica das Plataformas mobile.....	29
Tabela 5 - Ferramentas de desenvolvimento multiplataforma e funcionalidades nativas suportadas adaptada de (Palmieri, Singh, & Cicchetti, 2012).....	31
Tabela 6 - Comparação SOAP RESTful adaptada de (AlShahwan, Moessner, & Carrez, 2010)...	41
Tabela 7 - Soluções MOBILE por Empresa.....	59
Tabela 8 - Perspetivas vendedor e consumidor adaptado de (Kotler & Armstrong , 2013) .....	62
Tabela 9 - Product Backlog (Scrum) .....	81
Tabela 10 - Primeiro Sprint (Scrum) .....	83

# 1 – Introdução

## 1.1 - Descrição do Problema

O mundo dos negócios tem vindo a sofrer ao longo do tempo grandes mudanças, adaptando-se dinamicamente às crescentes exigências com que se depara ao nível da rapidez e qualidade de execução de tarefas (Galvin, 2004; Fischer & Smolnik, 2013; IDC, 2014). No entanto, a garantia de melhor qualidade continua a ser um requisito obrigatório. Neste contexto, os sistemas *mobile* permitem acesso fácil e eficaz à informação, em tempo real e em qualquer lugar, oferecendo um contributo importante para se atingirem os objetivos de competitividade necessários ao sucesso de qualquer empresa.

Os dispositivos *mobile* têm vindo a ser crescentemente utilizados para as mais diversas funções. Cada vez mais, equipamentos como os PDAs (*Personal Digital Assistants*), *tablets* e *smartphones* têm vindo a substituir os simples telemóveis, que agora são considerados dispensáveis e pouco práticos face às novas tecnologias e sistemas desenvolvidos (Oliveira & Medina, 2007). Esta crescente adoção deve-se ao facto destes dispositivos somarem à capacidade de comunicação de texto e voz o acesso à internet, com todo um universo de possibilidades de interação com outros sistemas de informação, permitindo mesmo a análise de informação em tempo real, auxiliando o utilizador na tomada de decisões e na execução de tarefas (Hsieh, 2007). Além disso, o valor de aquisição destes equipamentos tem vindo a descer, tornando acessível a sua aquisição com um *ratio* de custo/benefício cada vez mais apelativo.

Convém lembrar que o problema central dos dispositivos móveis prendia-se com o *hardware* e respetivo *firmware*, uma vez que cada equipamento detinha exclusivamente uma finalidade específica, limitando bastante as suas funções de utilização fora do contexto para o qual tinha sido criado. Além disso, impossibilitavam uma resposta mais eficaz a programas complexos e personalizados (Hall & Anderson, 2009).

O despoletar da internet e a sua rápida divulgação como um fenómeno global, gerou necessidades cada vez mais complexas por parte dos utilizadores, obrigando à criação de soluções exigentes. Por conseguinte, houve a necessidade de contornar limitações associadas aos recursos presentes nos dispositivos, nomeadamente no que concerne à capacidade das redes móveis associadas, à limitada capacidade de memória, ao poder de processamento do dispositivo, ao tamanho do ecrã e à incompatibilidade da integração de sistemas (Sanaei, Abolfazli, Gani, & Buyya, 2014). Criaram-se e desenvolveram-se novos ambientes de programação especificamente preparados e dedicados a estes novos equipamentos. Surgem,

em consequência, as mais diversas aplicações para inúmeras finalidades, totalmente independentes do tipo de dispositivo onde são utilizadas e do fabricante que as criou (Oliveira & Medina, 2007).

Esta nova realidade, com a internet em constante desenvolvimento e atualização, onde um equipamento não se encontra limitado a uma única função, estando apto a realizar um vasto número de tarefas, gerou uma nova forma de comunicação e informação que caracterizam a atualidade. A internet surge, assim, como elemento essencial para a realização das várias atividades sociais e profissionais do cotidiano das pessoas. Esta situação provoca a criação de serviços e ferramentas, cuja utilização garantem flexibilidade a quem as utiliza.

Os dispositivos móveis atuais assemelham-se, em termos de abrangência de funcionalidades, aos computadores pessoais. Permitem aceder a uma grande variedade de serviços e acompanhar em tempo real as atividades quotidianas (Gil, 2011). As aplicações *mobile* já conseguem competir a nível de complexidade, riqueza visual e interação em tempo real com as aplicações desenvolvidas para os *desktops* (Mengshoel, Iannucci, & Ishihara, 2013).

Estes dispositivos já têm previamente incorporado e instalado nos seus sistemas algumas tecnologias como Infrared e Bluetooth, para além de já incluírem Java Virtual Machine (JVM), compatíveis com a tecnologia Java Mobile Edition (J2ME).

Com a utilização massiva de dispositivos *mobile*, desenvolveu-se um elevado número de sistemas operativos, criando um novo mercado aplicacional que suporta milhares de aplicações destinadas para as mais diversas finalidades (Galvin, 2004; Lane, et al., 2010). A partir destas tecnologias foi possível a execução de muitas aplicações que, anteriormente, só poderiam ser executadas em computadores (Machado, et al., 2008). Surgiram também diversas *frameworks*, normalmente associadas a sistemas operativos e desenvolvidas para uma determinada plataforma (Gil, 2011).

Entre os vários sistemas operativos *mobile* desenvolvidos, destacaram-se o Android, iOS e Windows Phone, que poderão ser alvo da aplicação a criar no âmbito do presente projeto. Ainda que todos estes sistemas tenham sido desenvolvidos para dar resposta às necessidades comuns dos sistemas *mobile*, eles apresentam diferenças entre si que devem ser tidas em consideração quando se pretende desenvolver uma aplicação a ser incorporada em qualquer um desses sistemas.

Este novo paradigma de comunicações levou a que as empresas sentissem que a sua competitividade dependia também da utilização destas ferramentas (Heitkötter, Hanschke, &

Majchrzak, 2012). Do ponto de vista de desenvolvimento de tecnologias e de sistemas de informação, já não é possível pensar em novas soluções que não sejam compatíveis com dispositivos *mobile* (Galvin, 2004; Xu & Yang, 2012).

A empresa que acolhe o presente projeto sentiu a necessidade de se manter competitiva e de se adaptar a um mundo mais atual, decidindo investir no desenvolvimento de um sistema que permitisse estender as funcionalidades do seu ERP CHAIN ao ambiente *mobile*. Esta situação flexibilizará a utilização do sistema informático de gestão, permitindo-lhe aceder ou executar determinadas funcionalidades através de um qualquer dispositivo *mobile* que contenha a nova aplicação desenvolvida.

Uma pesquisa de mercado direcionada para o mercado português permitiu observar que não existem disponíveis soluções com as características pretendidas. Sendo a aplicação *ERP* focada no sector têxtil, um dos mais importantes sectores industriais nacionais, apenas se encontraram soluções *mobile* que permitem visualizar e analisar informação sem possibilidade de alteração e adição de novos dados. Este facto, permite acreditar no carácter inovador da aplicação a desenvolver e no seu potencial comercial.

## 1.2 - Objetivos

Os objetivos são um fim identificado que algo ou alguém se propõe atingir. Um projeto apenas termina quando esse fim é atingido, quando deixa de ser necessário ou quando deixa de ser possível atingir em tempo útil os objetivos propostos (PMI, 2013; Porto Editora, 2014; Oxford Dictionaries, 2014). Assim, para terminar o projeto com sucesso, é necessário avaliar uma série de variáveis, nomeadamente se o projeto é factível num determinado período de tempo e se existem os recursos necessários, para nesse período atingir determinadas necessidades identificadas. Neste projeto, o seu sucesso depende de se atingir os objetivos propostos para cada etapa, o que implica a concretização dos objetivos definidos para o plano do projeto, contextualização da empresa e enquadramento do produto e desenvolvimento da aplicação informática. Para cada etapa foram definidos os seguintes objetivos:

- Plano do Projeto
  - Definir as fases do projeto  
Identificar o trabalho necessário ao desenvolvimento do projeto, decompondo-o em tarefas de pequena dimensão e agrupadas de forma a constituir as diferentes fases do projeto.
  - Elaborar o plano de atividades  
Divisão do trabalho identificado ao longo do tempo do projeto.

- Analisar riscos  
Identificar os potenciais riscos que podem condicionar o projeto. Elaborar estratégias de mitigação de forma a procurar absorver os riscos com impacto positivo e limitar os riscos com impacto negativo no projeto.
- Contextualização da empresa e enquadramento do produto
  - Descrever a Empresa  
Breve caracterização da empresa abrangendo a sua atividade, produto e sector de atividade.
  - Traçar o Produto  
Identificar as funcionalidades do ERP (*Enterprise Resource Planning*) CHAIN.
  - Analisar Concorrência  
Fazer o levantamento da concorrência relativamente à existência de soluções *mobile* capazes de estender os seus ERPs. Identificar as funcionalidades existentes nas soluções encontradas.
  - Posicionar o Produto  
Importância do produto na estratégia de *Marketing*.
- Desenvolvimento da aplicação informática
  - Conceção da Solução  
Elaborar a modelação UML (*Unified Modelling Language*) para todo o produto.  
Definir o interface gráfico do utilizador para a plataforma Android.
  - Programar Produto  
Aplicar a metodologia ágil Scrum ao desenvolvimento do produto.

Para além dos objetivos do projeto já enunciados, existem também objetivos pessoais inerentes ao seu desenvolvimento, nomeadamente:

- Integrar o Mercado de Trabalho  
Conhecer novas realidades, enfrentar as dificuldades associadas ao mercado trabalho, aplicar a minha formação académica num contexto real, obtendo experiência através do desenvolvimento de um projeto requisitado por uma empresa, trabalhando para a empresa e com a empresa, a fim de atingir um resultado.
- Complementar a formação Académica  
Sendo esta uma área de futuro e em constante desenvolvimento, é uma oportunidade de investigar e desenvolver competências numa área pouco explorada na formação em Engenharia e Gestão de Sistemas de Informação.

- Melhorar e Desenvolver as *Soft Skills*

Desenvolver atitudes e comportamentos que facilitam a relação interpessoal, a fim de melhorar o desempenho profissional e social.

### 1.3 - Abordagem Metodológica

Ao longo da minha formação académica foram adquiridas e constantemente aplicadas, como Engenheiro e Gestor de Tecnologias e Sistemas de Informação, as melhores práticas, nomeadamente:

- Planeamento de Projetos

Neste projeto, o seu plano será auxiliado pelo *Project Management Body Of Knowledge* (PMBOK) que é definido pelo PMI (2013) como um guia de diretrizes para a gestão de projetos individuais e onde estão definidos os conceitos e boas práticas relacionadas com gestão de projetos.

- Caracterização e Análise de Organizações, Produtos e Ambiente

Neste projeto, não será descrita detalhadamente a organização, apenas será dado foco ao produto que esta comercializa. Foi detalhado o CHAIN, identificados e caracterizados os potenciais concorrentes da solução a desenvolver e efetuada uma análise do reposicionamento do produto face à nova solução desenvolvida.

- Conceção e Desenvolvimento de Aplicações informáticas

Na conceção da aplicação informática, serão modelados através de UML, os requisitos da solução e desenhado o interface gráfico do utilizador. O desenvolvimento do produto será auxiliado pela utilização de uma *framework* iterativa e incremental vocacionada para o desenvolvimento ágil de *software* denominada Scrum (Scrum Alliance, 2014).

### 1.4 - Estrutura do Projeto

A investigação assume um papel fulcral para o desenvolvimento de saberes próprios da disciplina, nomeadamente ao nível das tecnologias e sistemas de informação, servindo de base científica para a produção e desenvolvimento de tecnologias inovadoras. Assim, existe uma necessidade crescente, como engenheiro e gestor de sistemas de informação, de participar ativamente e contribuir para a investigação de forma a adquirir novos conhecimentos e ajudar a desenvolver o futuro das novas tecnologias, especificamente na área do desenvolvimento *mobile*, sobre a qual versa o presente projeto.

Inicialmente, anterior à contextualização do tema em estudo e à enunciação dos seus objetivos neste primeiro capítulo, o projeto que se segue foi alvo de uma revisão sistemática da literatura

teórica e técnica, permitindo efetuar uma pesquisa aprofundada relativamente ao tema em questão. Desta forma, esta revisão permitiu, para além da definição dos objetivos e identificação do problema, definir a estrutura deste projeto bem como a abordagem metodológica a seguir, tal como apresentado neste primeiro capítulo.

O segundo capítulo irá versar sobre o plano do projeto em si, onde serão abordados aspetos relativos ao mesmo. Assim, o objetivo deste capítulo passa por elucidar o leitor, inicialmente, sobre a forma como este projeto foi estruturado, descrevendo as várias fases através das quais este se foi desenvolvendo. De seguida, irá ser apresentado o plano das atividades delineadas e implementadas no intuito de se alcançarem os objetivos propostos dentro do tempo previsto. O capítulo termina com uma análise de riscos, onde se pretende definir estratégias no intuito de dar respostas a possíveis riscos com impacto negativo ou positivo a que o projeto possa estar sujeito. O propósito é reduzir o impacto danoso dessa incerteza e definir ações que potencializem a ocorrência de riscos com impacto proveitoso ao projeto.

O capítulo três iniciará com um mapa de conhecimento, onde todos os conceitos necessários ao desenvolvimento do projeto se encontram representados de forma simples, para uma melhor perceção da organização dos conceitos necessários ao desenvolvimento da solução e como se relacionam. Segue-se uma revisão exaustiva da literatura no intuito de explicar os conceitos implicados no projeto. Inicialmente, serão esclarecidas as abordagens de desenvolvimento existentes atualmente e suas características distintas. Dá seguimento uma especificação das plataformas Android, IOS e Windows Phone, sendo retratadas as suas diferenças. Prossegue com um levantamento das ferramentas existentes e a caracterização da linguagem JAVA. O sexto ponto deste capítulo é relativo a *Web Services*, esclarecendo o seu funcionamento, elucidando o tipo que será utilizado na comunicação da solução. Por fim, mas não menos importante, será explicada a metodologia de modelação aplicada e os elementos que dela serão utilizados para desenhar os requisitos e interações presentes na solução. Termina com a definição da metodologia ágil Scrum, *framework* que dará suporte ao desenvolvimento do produto.

No quarto capítulo consta uma contextualização da empresa e enquadramento do produto. Este capítulo pretende fornecer uma breve descrição da empresa e enquadrar o produto no contexto da organização e do mercado. Será efetuada uma análise da concorrência, identificando os produtos *mobile* oferecidos, procedendo ao posicionamento do produto no mercado. É realizada uma apresentação da oferta disponível atualmente, analisando-se as diversas soluções existentes e caracterizando cada uma delas.

No capítulo cinco, será retratado o conceito pensar em contraste com o conceito fazer. Iniciará com a conceção do produto através de modelos obtidos utilizando a linguagem UML e com a definição da interface gráfica do utilizador que correspondem ao conceito de pensar o produto. Termina com o conceito fazer, onde será adaptada e aplicada a metodologia ágil Scrum, na programação do produto.

O sexto capítulo é referente aos resultados do desenvolvimento do produto, avaliando inicialmente o impacto do mesmo na organização e prevendo os benefícios inerentes ao cliente. Contemplará também um manual técnico com uma linguagem simples e perceptível ao utilizador final.

Por fim, o projeto termina com a conclusão, onde numa primeira instância são revisitados os objetivos inicialmente definidos, a fim de avaliar se estes foram atingidos. Englobará também oportunidades que o seu desenvolvimento proporcionou e sugestões para trabalhos futuros.

## 2 – Plano do Projeto

Para se perceber a importância deste ponto, é importante definir o que é um projeto. O PMI (2013), define um projeto como um esforço temporário, com um início e um fim bem definido, empreendido para criar um produto, serviço ou resultado exclusivo. Isto é, algo criado em função de determinadas especificações e restrições, que exige recursos, durante um determinado período de tempo do qual resulta um produto, serviço ou resultado único.

Através desta definição, é possível concluir que um bom plano terá de garantir que o esforço é compatível com o tempo proposto; que o projeto tem um início e um fim bem definido; que as ferramentas escolhidas são as mais indicadas; que a solução apresentada é exclusiva e satisfaz as necessidades do cliente, isto é, se cumpre à risca as especificações exigidas, o que implica um bom levantamento de requisitos; se os recursos existentes são suficientes para o cumprimento dos prazos estabelecidos; antecipar de modo a preparar uma resposta a algum desvio do plano que possa surgir ao longo do tempo, que implique um atraso em alguma fase do projeto. Portanto, fazer um levantamento de riscos a que o projeto está sujeito, ao longo das várias etapas definidas e definir possíveis soluções para os mesmos, é crítico para o seu resultado. Todos estes são fatores de sucesso de um projeto, o que revela a importância de um bom plano.

Nesta segunda secção serão identificadas e detalhadas as fases do projeto e as tarefas que as constituem. Segue-se o plano de atividades onde essas tarefas são distribuídas ao longo do tempo e termina com o levantamento de possíveis riscos e medidas que poderão ser aplicadas ao longo do desenvolvimento do projeto, para que este decorra como planeado e da melhor maneira possível.

## 2.1 - Fases do Projeto

De acordo com o PMI (2013) a decomposição do trabalho a realizar em componentes mais pequenos torna-o mais fácil de gerir em função das suas entregas. Na Figura 1, encontram-se as fases decompostas em tarefas a realizar durante o projeto.

<b>Projeto de Extensão de Funcionalidades do ERP CHAIN para Dispositivos MOBILE</b>	<b>Primeira fase - Planear o Projeto</b>	Especificar Fases
		Definir Plano de Atividades
		Elaborar Análise de Riscos
	<b>Segunda Fase - Contexto da Empresa e Equadramento do Produto</b>	Descrição da Empresa
		Detalhar Produto
	Analisar Concorrência	
	Posicionar Produto	
	<b>Terceira Fase - Fundamentação Teórica</b>	Elaborar Mapa de Conhecimento
		Determinar Abordagem de Desenvolvimento
		Escolher Plataforma
		Eleger Ferramentas
		Identificar Linguagem
		Definir <i>Web Services</i>
		Retratar Linguagem de Modelação
		Explicar Metodologia Ágil de Desenvolvimento
	<b>Quarta Fase - Desenvolver Produto</b>	Conceber Produto
		Programar Produto
	<b>Quinta Fase - Elaborar Documentos Técnicos</b>	

Figura 1 - Fases do Projeto

## 2.2 - Plano de Atividades

A alocação das tarefas ao longo do tempo permite que, a qualquer momento, seja possível verificar se o projeto está a ser desenvolvido no tempo previsto, de modo a garantir as entregas nos respetivos prazos. Na Figura 2, é possível verificar a distribuição do esforço pela duração deste projeto.

Fases	Tarefas	2013			2014										
		out	nov	dez	jan	fev	mar	abr	mai	jun	jul	ago	set	out	
Primeira Fase	Especificar Fases	■													
	Definir Plano de Atividades	■													
	Elaborar Análise de Riscos	■	■												
Segunda Fase	Descrever Empresa		■												
	Detalhar Produto			■	■										
	Analisar Concorrência				■	■									
	Posicionar Produto						■								
Terceira Fase	Elaborar Mapa de Conhecimento						■								
	Determinar Abordagem de Desenvolvimento						■	■							
	Escolher Plataforma							■	■						
	Eleger Ferramentas							■	■						
	Identificar linguagem								■	■					
	Definir <i>Web Services</i>								■	■					
	Retratar Linguagem de Modelação									■	■				
	Explicar Metodologia Ágil de Desenvolvimento									■	■	■			
Quarta Fase	Conceber Produto										■	■			
	Programar Produto											■	■	■	
Quinta Fase	Elaborar Documentos Técnicos													■	

Figura 2 - Plano de Atividades

Como se pode verificar na Figura 2, o trabalho implicará a execução das seguintes tarefas:

- Primeira Fase – Planear o Projeto

Esta é a mais importante de todas as tarefas servindo como guia e fornecendo diretrizes ao trabalho. Está subdividida em três tarefas:

- Especificar Fases

Consiste na decomposição do produto em várias fases e tarefas a desenvolver, permitindo uma melhor análise e estimativa do trabalho fazível para proceder à sua distribuição pelo tempo do projeto.

- Definir Plano de Atividades

Compreende a distribuição do trabalho ao longo do tempo útil do projeto. Permite uma visão em tempo real do estado do projeto, através da identificação das tarefas realizadas e por realizar em função do tempo já empregado e do ainda disponível.

- Elaborar Análise de Riscos

Compreende a identificação dos riscos potenciais ao projeto, analisando o seu impacto, definindo estratégias que minimizem o seu impacto negativo e beneficiem dos riscos com impacto positivo. Pode ter um papel fundamental para o sucesso do projeto.

- Segunda Fase – Contexto da Empresa e Enquadramento do Produto

Esta segunda fase tem como foco o produto CHAIN. Decompõem-se em quatro pontos:

- Descrever Empresa

Uma breve descrição da empresa e identificação do seu setor de atividade.

- Detalhar Produto

Descrição das funcionalidades do CHAIN e identificação das suas características.

- Analisar Concorrência

Estudo dos concorrentes da Doctron com intuito de verificar os que já detêm produtos *mobile* que podem ser potenciais competidores. Identificar as funcionalidades desses produtos.

- Posicionar Produto

Retratar o posicionamento do produto com o do desenvolvimento desta extensão.

- Terceira Fase – Fundamentação Teórica

São oito as tarefas que constituem esta terceira fase. Compreende todos os documentos que dão apoio à tomada de decisão da empresa. Envolve uma revisão sistemática da literatura teórica e técnica sobre os conceitos necessários ao desenvolvimento da solução.

- Elaborar Mapa de Conhecimento  
Desenhar e relacionar de forma simples os conceitos necessários ao desenvolvimento do projeto.
- Determinar Abordagem de Desenvolvimento  
Identificar, analisar e descrever as abordagens existentes para o desenvolvimento *mobile* bem como suas respectivas vantagens e desvantagens.
- Escolher Plataforma  
Descrever e caracterizar as plataformas Android, IOS e Windows Phone. Aludir a plataforma alvo do desenvolvimento conforme a escolha da empresa.
- Eleger Ferramentas  
Abordar as ferramentas existentes, identificando as escolhidas para o desenvolvimento do produto, em função da plataforma escolhida anteriormente.
- Identificar Linguagem  
Breve referência à história e características da linguagem a utilizar.
- Definir *Web Services*  
Definir e explicar o funcionamento dos *Web Services*. Esclarecer como irá comunicar o servidor do CHAIN com a solução *mobile*.
- Retratar Linguagem de Modelação  
Introduzir a linguagem UML e identificar os diagramas que vão ser utilizados neste projeto.
- Explicar Metodologia Ágil de Desenvolvimento  
Apresentar as características da metodologia ágil Scrum e elucidar do seu funcionamento.

- Quarta Fase – Desenvolver Produto

Nesta quarta fase existem dois conceitos que originam a divisão das tarefas em dois grupos:

- Conceber Produto  
Este grupo de tarefas é referente ao conceito de pensar o produto que compreende as seguintes tarefas:

- Modelar o Produto usando UML  
Aqui será desenhado, através de *Unified Modeling Language* (UML), todas as funcionalidades da solução, isto é, será possível visualizar todos os módulos, interações, funcionalidades do produto e seus utilizadores, de modo diagramático, com o intuito de auxiliar o programador no cumprimento rígido das funcionalidades e requisitos da solução.
- Definir a Interface Gráfica para o Utilizador (GUI)  
Nesta tarefa, serão definidos os menus com os quais os clientes terão contacto direto com a aplicação. Estes devem ser intuitivos e fáceis de utilizar.
- Programar Produto  
Este grupo de tarefas é referente ao conceito fazer. Através do que foi elaborado no grupo anterior, proceder-se-á à programação fiel dos requisitos da solução a entregar ao cliente.
- Quinta Fase – Elaborar Documentos Técnicos  
Por fim, serão elaborados os documentos técnicos da solução, nomeadamente o manual técnico, com intuito de dar suporte ao cliente durante a utilização da aplicação.

## 2.3 - Análise de Riscos

De modo a maximizar o sucesso de um projeto, é importante levantar possíveis riscos e definir estratégias de mitigação que minimizem o seu impacto negativo e ações que procurem o seu impacto positivo. Assim, é possível ter algum controlo sobre a incerteza inerente ao desenvolvimento das várias fases do projeto. O PMI (2013, p. 310), define o risco do projeto como “Um evento ou condição incerta que, se ocorrer tem um efeito positivo ou negativo num ou mais objetivos do projeto, nomeadamente no escopo, cronograma, custo e qualidade”. A Tabela 1 aborda a probabilidade, o nível de impacto previsto, os indicadores, as consequências e quais as estratégias que podem limitar o impacto negativo ou perseguir o impacto positivo para os riscos identificados.

Tabela 1 - Análise de Riscos

Riscos	Probabilidade [1...5]	Impacto [1...5]	Indicadores	Consequências	Estratégia
<b>Má definição dos objetivos e/ou Mau levantamento de requisitos</b>	2	5	A solução não corresponde ao acordado com o cliente;	Atraso na entrega do projeto; Insatisfação do cliente; Perda de recursos;	Aumentar o número de reuniões com o cliente; Melhorar estratégias de comunicação;
<b>Atraso numa fase do projeto</b>	3	5	Dificuldade em cumprir os prazos inicialmente definidos.	Atraso na conclusão das tarefas; Necessidade de renegociação dos prazos estabelecidos.	Aumento do número de horas de trabalho; Possível realocação dos recursos disponíveis.
<b>Perda de dados</b>	1	5	Dificuldade em cumprir as tarefas Não cumprimento dos prazos definidos.	Atraso na conclusão; Necessidade de refazer o trabalho perdido.	Fazer <i>backups</i> diários do trabalho em duas plataformas partilhadas (Dropbox e Google Drive); Manter um <i>backup</i> num dispositivo físico; Aplicar controlo de versões;

Riscos	Probabilidade [1...5]	Impacto [1...5]	Indicadores	Consequências	Estratégia
<b>Falta de experiência e <i>know-how</i></b>	2	5	Dificuldade de interpretação do material adquirido; Dificuldade no cumprimento das tarefas; Falta de qualidade na realização das tarefas; atraso no planeamento.	Má utilização das ferramentas. Resistência à mudança, lacunas na formação.	Procurar cursos de formação; procurar ajuda com os colegas, empresa e docentes; Aquisição de material de formação.
<b>Más práticas de trabalho</b>	2	3	As tarefas não são desempenhadas com a qualidade pretendida ou são realizadas fora de horas.	Atraso para futuras atribuições, Necessidade de rever essas tarefas.	Rever os artefactos. Realizar as tarefas cumprindo o prazo estipulado das mesmas.
<b>Avaria de equipamento</b>	2	2	Dificuldade em cumprir os prazos definidos; Perda de dados; Atraso na entrega das tarefas.	Atraso na conclusão das tarefas.	Compra, aluguer ou empréstimo de novo equipamento.
<b>Alteração de requisitos e funcionalidades do sistema</b>	5	5	Dificuldade em cumprir os prazos estipulados; Atraso na entrega das tarefas; Constante redefinição dos requisitos e funcionalidades.	Atraso na conclusão das tarefas, Comprometimento dos objetivos; Diminuição do tempo útil do projeto.	Aumentar reuniões com cliente; Realizar levantamento de riscos cuidada e de acordo com a pretensão do cliente; Melhorar comunicação;
<b>Estabilidade do servidor</b>	1	5	Dificuldade em estabelecer ligações ao servidor; Longos períodos sem acesso; Não receção de dados.	Atraso nas tarefas; Falha de testes; Diminuição do tempo útil; Comprometimento dos objetivos.	Garantir com o cliente a qualidade do servidor; Sensibilizar o cliente para a importância de um sistema estável.
<b>Falta de Suporte e/ou incompatibilidade do <i>software</i></b>	3	4	Dificuldade na interoperabilidade do <i>software</i> ; <i>Falta de soluções nativas</i> ;	Atraso nas tarefas; Diminuição do tempo útil, Comprometimento dos objetivos.	Sensibilizar o cliente para a escolha do <i>software</i> ; <i>Encontrar soluções alternativas</i> .
<b>Falta de disponibilidade do cliente</b>	3	5	Dificuldade em reunir com o cliente; Reuniões menos produtivas.	Atraso nas tarefas; Comprometimento dos objetivos; Diminuição do tempo útil do projeto.	Sensibilizar o cliente para a importância das reuniões.
<b>Familiaridade com alguma ferramenta,</b>	5	5	Facilidade de compreensão.	Aumento do tempo útil do projeto;	Sensibilizar o cliente para essa ferramenta, <i>framework</i> ou linguagem.

Riscos	Probabilidade [1...5]	Impacto [1...5]	Indicadores	Consequências	Estratégia
<i>framework</i> ou linguagem					
Utilização de <i>software Open Source</i> já existente	4	4	Necessidade de programar todas as funcionalidades inexistente.	Aumento do tempo útil.	Realizar pesquisas de <i>software</i> com regularidade; Frequentar fóruns dedicados ao desenvolvimento.
Ocorrência de formações e/ou workshops relacionados	3	3	Aparecimento de formação ou <i>workshop</i> .	Aumento do <i>know-how</i> ; Aumento da qualidade das tarefas; Diminuição do tempo útil.	Registrar contacto na <i>Mailing List</i> de empresas de formação especializadas.
Boa relação com o Cliente	5	5	Satisfação do cliente. Confiança depositada.	Maior influência sobre a tomada de decisão do cliente.	Praticar uma comunicação formal e adequada; Ter uma abordagem transparente com o cliente.
Entrega antecipada de etapas	3	3	Entrega das tarefas antes do tempo previsto.	Aprovação ou Rejeição com maior antecedência; Melhor gestão de recursos.	Boa comunicação; Efetuar um bom planeamento e levantamento de requisitos.

Como se pode verificar na Tabela 1, são vários os potenciais riscos que podem ocorrer ao longo do projeto, com impacto positivo ou negativo. Para o sucesso do projeto interessa procurar a ocorrência dos riscos com impacto positivo e evitar os riscos com impacto negativo. Assim sendo, foram utilizados os seguintes parâmetros para cada risco:

- Probabilidade:  
Refere-se à probabilidade da ocorrência do risco. Emprega um intervalo de 1 a 5 onde o 1 corresponde a uma probabilidade muito baixa e 5 a uma probabilidade extremamente alta.
- Impacto:  
Refere-se ao nível de impacto que o risco pode ter no projeto. Esse impacto pode ser positivo ou negativo. É medido através de uma escala de 1 a 5, onde 1 significa que o impacto é muito baixo e 5 que o impacto é extremamente elevado.
- Indicadores:  
Esta coluna engloba todos os sinais que podem indicar o aparecimento do risco.

- Consequências:

Esta coluna representa o resultado de cada risco. Esse resultado pode ser positivo resultando numa melhoria para o projeto ou negativo levando a um agravamento.

- Ações:

Identifica as ações que podem ser tomadas para evitar a ocorrência de riscos com impacto negativo e as ações que podem ser realizadas para procurar a ocorrência dos riscos com impacto positivo.

## 2.4 – Resumo do Capítulo

Este é um capítulo muito importante que retrata o plano do projeto. Através das fases do projeto e da descrição das tarefas é possível garantir que nada será esquecido e que os requisitos do projeto serão cumpridos e estão planejados. O plano de atividades permite verificar atrasos na realização das tarefas e em qualquer altura é possível verificar se o projeto está a decorrer conforme o planeado ou se existem atrasos. O capítulo termina com a análise de riscos, que nos oferece um guia de ações e estratégias para lidar com a incerteza. Este guia contém uma previsão dos riscos potenciais do projeto, contendo indicadores, consequências e estratégias para lidar com os diferentes riscos, independentemente do seu impacto no projeto ser positivo ou negativo.

## 3 – Fundamentação Teórica

Nesta secção será retratada toda a pesquisa bibliográfica efetuada sobre os conceitos teóricos que serão aplicados no projeto, servindo de preparação e de apoio à decisão no planeamento e desenvolvimento da solução. A secção começa com o mapa de conhecimento, cujo objetivo é de forma simples representar todos os conceitos e informação relevante ao projeto. Segue-se uma análise às diferentes abordagens de desenvolvimento disponíveis, para as distintas plataformas de suporte de aplicações, para dispositivos *mobile*. A secção prossegue com uma identificação das diferentes plataformas, com maior foco para a plataforma Android, que será alvo de maior reflexão neste projeto. No ponto seguinte, será abordado a linguagem de programação utilizada, assim como as ferramentas empregadas. Sendo este um projeto de extensão de funcionalidades do ERP CHAIN, um tópico sobre a comunicação entre a aplicação *mobile* e o servidor do CHAIN deve ser bastante refletido e estudado, sendo este o ponto que se segue. Por fim, os dois últimos tópicos descrevem os conceitos básicos aplicados, referentes à metodologia de modelação seguida e à *framework* ágil Scrum, que auxiliou o desenvolvimento da solução.

### 3.1 Mapa de Conhecimento

Quando se inicia um projeto é sempre bom ter um mapa que nos indique o caminho e as decisões que temos de tomar para que o percurso seja empreendido e percorrido com sucesso. O mapa de conhecimento representado na Figura 3, tem esse mesmo objetivo. Oferece ao leitor uma vista geral e externa do caminho a percorrer e das opções que devem ser realizadas para que o caminho seja bem-sucedido e compreendido. O rumo tem como meta o desenvolvimento do produto CHAIN Mobile, mas para ser atingida essa meta é necessário fazer uma análise cuidada e estudar as funcionalidades que a versão final do CHAIN Mobile deve englobar, que abordagens de desenvolvimento existem, que módulos vão ser focados e desenvolvidos, que plataformas *mobile* existem e atribuir-lhes uma prioridade de desenvolvimento, analisando também as suas linguagens, como será realizada a comunicação entre o servidor e a solução e que ferramentas existem para o seu desenvolvimento. Representa também a linguagem de modelação utilizada, servindo como mapa para o percurso a seguir e a *framework* Scrum, que orientará ao longo do percurso.

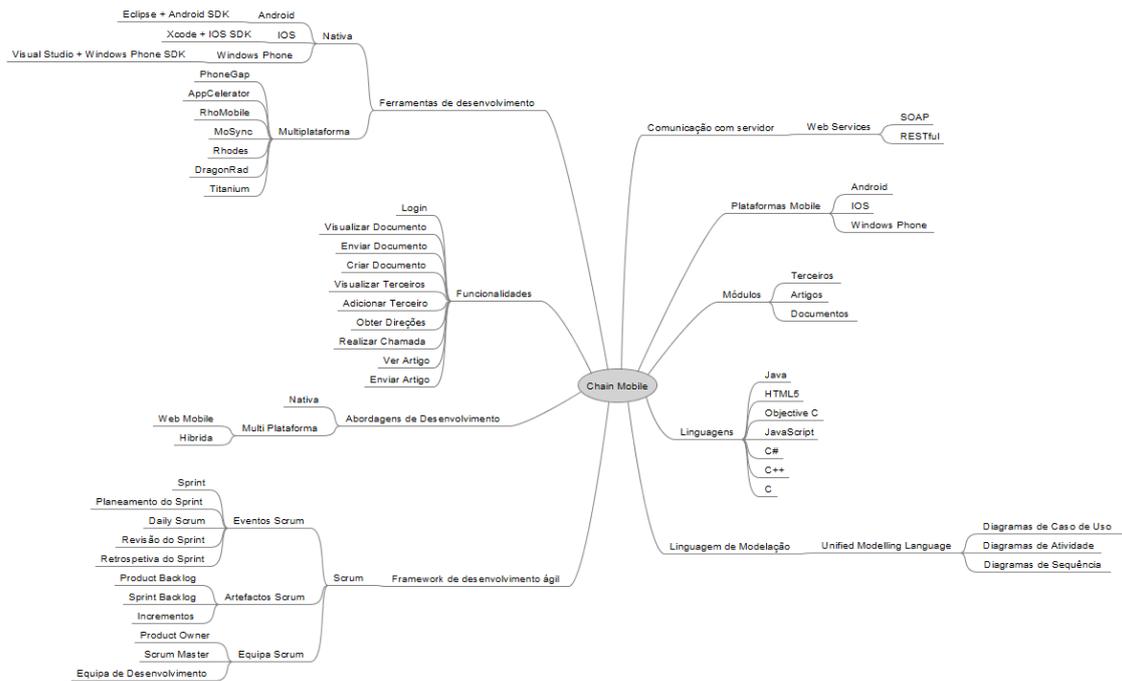


Figura 3 - Mapa de Conhecimento

### 3.2 Abordagem de Desenvolvimento

Neste subcapítulo será abordada a secção do mapa de conhecimento relativa às abordagens de desenvolvimento, que se encontra representada na Figura 4.

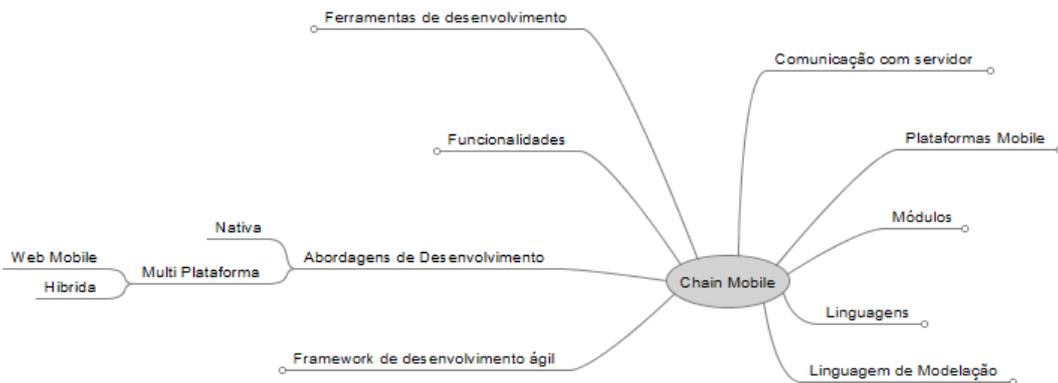


Figura 4 - Mapa de Conhecimento - Abordagens de Desenvolvimento

No desenvolvimento de aplicações *mobile*, existem duas abordagens diferentes que devem ser consideradas: nativa e multiplataforma. A abordagem multiplataforma, pode ser dividida em dois tipos: a abordagem *Web Mobile* e a *Hybrid* (Híbrida), sendo que a híbrida tenta misturar a abordagem nativa e a *Web Mobile* (Dehlinger & Dixon, 2011; JOŠT, HUBER, & HERIČKO, 2013). Cada uma destas abordagens tem as suas características, vantagens e desvantagens para o

programador, utilizador e cliente. Por razões diversas, quer de mercado ou organizacionais, os programadores ou equipas de desenvolvimento, devem criar aplicações suportadas em diferentes plataformas (Charland & Leroux, 2011).

Na abordagem nativa, não é possível desenvolver uma aplicação compatível com todas as plataformas. Esta impossibilidade prende-se com as diferenças existentes na linguagem de programação e do Interface de Programação de Aplicações (API) de cada plataforma, obrigando ao desenvolvimento de uma nova aplicação para uma nova plataforma, segundo a sua linguagem de programação e API. (Heitkötter, Hanschke, & Majchrzak, 2012; White, 2013).

Devido a esta peculiaridade, o desenvolvimento nativo para diferentes plataformas obriga que a equipa encarregada pela criação das aplicações nativas, tenha experiência e um *know-how* específico com vasto conhecimento nas diferentes linguagens, *Software Development Kits* (SDK) e APIs (Heitkötter, Hanschke, & Majchrzak, 2012; Charland & Leroux, 2011; Dalmaso, Datta, Bonnet, & Nikaein, 2013; Sharif, Gull, & Nazir, 2013). Por outro lado, no desenvolvimento multiplataforma, a equipa irá utilizar linguagens *web*, nomeadamente o HTML, CSS e JavaScript, e um SDK sem necessitar de conhecimentos sobre as linguagens específicas de cada plataforma e API's.

Para além das diferenças já enumeradas, existe ainda um elevado número de dispositivos com capacidades distintas, fabricados especificamente para as várias plataformas. O que os torna apetecíveis para a abordagem *Web Mobile* é o facto de, independentemente da plataforma suportada, existir um *Web Browser* que pode ser utilizado e alvo de programação (Ronkainen, Eskeli, Urhema, & Koskela-Huotari, 2013). A solução híbrida irá funcionar em qualquer plataforma através de uma *Web View* permitindo aceder a algumas funcionalidades nativas, utilizando JavaScript (Charland & Leroux, 2011). A *Web View*, pode apresentar-se em *fullscreen*, com intuito de imitar uma aplicação nativa.

No conceito multiplataforma, as aplicações são geralmente chamadas de *Web Apps*, *Browser Apps* ou aplicações não nativas (White, 2013). Quanto às aplicações *Web*, as suas nomenclaturas advêm do facto de estas tirarem partido do *Web Browser* existente nas plataformas *mobile* e serem desenvolvidas utilizando tecnologias *Web* (Sharif, Gull, & Nazir, 2013). Mais concretamente, são uma página *Web* otimizada para dispositivos *mobile* (Heitkötter, Hanschke, & Majchrzak, 2012; Juntunen, Jalonen, & Luukkainen, 2013). Esta otimização pode consistir nomeadamente, na adaptação da página *Web* ao tamanho do ecrã de cada dispositivo e da usabilidade (Heitkötter, Hanschke, & Majchrzak, 2012). Através desta abordagem, o código

gerado é bastante inferior ao produzido no desenvolvimento nativo, tal como acontece com o tempo necessário ao desenvolvimento da aplicação para as várias plataformas.

O Interface Gráfico do Utilizador também difere nas aplicações resultantes das duas abordagens. As aplicações nativas, são desenvolvidas de acordo com um conjunto de especificações fornecidas pelo fabricante do sistema operativo e acompanhadas de bibliotecas completas com código, que pode ser usado pelos programadores garantindo a consistência das aplicações. Os elementos utilizados no interface gráfico do utilizador não variam de aplicação para aplicação, ostentam uma aparência familiar e específica daquela plataforma, caracterizando cada aplicação. Em contraste, o Interface Gráfico do Utilizador de uma *Web App* difere do existente na nativa, podendo duas *Web Apps* ter interfaces completamente diferentes e não familiares à plataforma. A mesma *Web App* pode apresentar ligeiras diferenças em *Web Browsers* distintos, o que pode ser um problema e fazer toda a diferença (Charland & Leroux, 2011; Juntunen, Jalonen, & Luukkainen, 2013; Ronkainen, Eskeli, Urhema, & Koskela-Huotari, 2013). Mesmo tentando imitar o interface de uma aplicação nativa uma *Web App* nunca passará de uma aproximação (White, 2013).

A nível de desempenho também encontramos diferenças no resultado das duas abordagens. As aplicações nativas apresentam um desempenho superior às não nativas. Nas *Web Apps*, a internet é um requisito para o seu funcionamento. Todo o conteúdo é carregado através da internet e descarregado do servidor, tendo impacto negativo no tempo de carregamento dos menus e de execução de operações, necessitando de um acesso contínuo à internet para aceder ao seu conteúdo (White, 2013; Juntunen, Jalonen, & Luukkainen, 2013). As operações de otimização demoram tempo a ser processadas, ao contrário das nativas que geralmente já se encontram otimizadas à plataforma e ao dispositivo. As aplicações nativas são mais rápidas a executar jogos 3D e aplicações de processamento de imagem, mas a diferença de desempenho numa aplicação orientada ao negócio, bem desenvolvida e utilizando as tecnologias *Web* é insignificante ou impercetível (Charland & Leroux, 2011; Ronkainen, Eskeli, Urhema, & Koskela-Huotari, 2013). No caso das aplicações híbridas, estas dificilmente se encontram otimizadas e podem apresentar défices de estabilidade (Dalmaso, Datta, Bonnet, & Nikaiein, 2013).

Existem diferenças também ao nível das funcionalidades inerentes ao dispositivo, às quais as aplicações resultantes das duas abordagens conseguem aceder. Uma aplicação para um dispositivo *mobile* ganha valor acrescentado por cada funcionalidade do dispositivo que consegue incorporar e tirar partido. Devido ao facto das aplicações nativas serem desenvolvidas segundo as especificações e utilizando o SDK do fabricante da plataforma, é possível integrar de

forma simples, todas as funcionalidades do dispositivo na aplicação, tornando-as totalmente utilizáveis como serviços de valor acrescentado à aplicação final, trazendo vantagens ao cliente. O mesmo não acontece com as aplicações resultantes das abordagens de multiplataforma, onde nem todas as funcionalidades do dispositivo são compatíveis ou acessíveis, o que impossibilita a sua utilização (Charland & Leroux, 2011; Heitkötter, Hanschke, & Majchrzak, 2012; Sharif, Gull, & Nazir, 2013). As funcionalidades que conseguem ser acedidas e são compatíveis têm uma integração mais complexa (White, 2013; Juntunen, Jalonen, & Luukkainen, 2013).

O uso da internet também difere nas duas abordagens. Enquanto nas aplicações baseadas em *Web Mobile*, todas as suas funcionalidades dependem totalmente da internet, as nativas conseguem fornecer funcionalidades que não dependem da internet. Quando é feito o *download* de uma aplicação, é possível definir um limite na utilização dos dados móveis e até mesmo, definir se o seu *download* só é possível através de WI-FI, consoante o tamanho da aplicação. Nas *Web Apps* isso não acontece. A simples utilização da aplicação consome tráfego que em excesso pode trazer um custo adjacente para o utilizador (White, 2013). No caso das aplicações híbridas, também são disponibilizadas algumas funcionalidades *offline*. No entanto, estas aplicações podem apresentar défices de estabilidade (Dalmasso, Datta, Bonnet, & Nikaiein, 2013).

O suporte dado às aplicações através de atualizações difere a nível de procedimentos. Como em qualquer aplicação, existe a necessidade constante de corrigir *bugs*, acrescentar funcionalidades, melhorar a usabilidade, melhorar o interface gráfico e otimizar o *software*. Quando é lançado um novo *update* para uma aplicação nativa tem que ser aprovado pela loja/mercado da plataforma, tal como acontece nas aplicações resultantes de uma abordagem híbrida, enquanto as *Web Apps* são mais flexíveis e podem receber um *update* em qualquer altura, sem que este seja monitorizado ou tenha a necessidade de ser aprovado pela loja/mercado. A aprovação por parte da loja/mercado pode, em alguns casos, ser demorada e só é dado após uma análise ao conteúdo e à qualidade do *software* (White, 2013; Dalmasso, Datta, Bonnet, & Nikaiein, 2013).

A nível de segurança, uma aplicação nativa é desenvolvida segundo as especificações do fabricante da plataforma, através da utilização do SDK e das bibliotecas disponibilizadas, o que confere um nível elevado de segurança. Na abordagem *Web Mobile*, a segurança depende exclusivamente do *Web Browser* que utiliza (Wang, Xing, Wang, & Chen, 2013). Lamentavelmente, as tecnologias *Web* têm uma funcionalidade bastante perigosa: permitem que os dados e o código sejam misturados. Esta funcionalidade permite que o JavaScript seja

incorporado dentro das páginas HTML, mas também consente ataques generalizados de injeção de código denominados por *Cross-Site Scripting* (XSS) (Jin, Hu, Ying, Du, & Yin, 2014). Enquanto nas *Web Apps* essa injeção só é possível através de um canal do Servidor *Web*, as aplicações *mobile* baseadas em HTML5, tem muitos canais de injeção de código possíveis. As aplicações *mobile* interagem com outras entidades através de vários canais de dados, nomeadamente, código de barras, SMS, Sistema de ficheiros, contactos, Wi-Fi, NFC, entre outros, e todos estes canais podem ser alvo deste tipo de ataques. A simples pesquisa de um *Access Point* Wi-Fi pode originar um destes ataques. Esta falha de segurança está presente na maior parte das plataformas *mobile* incluindo o Android, IOS e Windows Phone denunciando a falta de segurança alargada nas aplicações híbridas, mas também presente em menor escala nas *Web Apps* (Jin, Hu, Ying, Du, & Yin, 2014).

Na perspetiva do cliente, o desenvolvimento nativo para as diferentes plataformas acarreta um maior custo e requer um período de tempo maior, dependendo do número de plataformas para as quais é necessário reescrever a aplicação, que por sua vez, tem de respeitar as especificações da plataforma alvo (Charland & Leroux, 2011; Heitkötter, Hanschke, & Majchrzak, 2012; Juntunen, Jalonen, & Luukkainen, 2013; Zahra, Khalid, & Javed, 2013; JOŠT, HUBER, & HERIČKO, 2013; Ciman, Gaggi, & Gonzo, 2014). Desenvolver uma aplicação que corra em todas as plataformas acarreta menores custos do que replicar várias vezes e desenvolver nativamente sempre que surja uma plataforma ou uma nova versão, (Wasserman, 2010).

Em suma, tudo o que pode ser feito através de uma abordagem multiplataforma pode ser feito nativamente (Heitkötter, Hanschke, & Majchrzak, 2012). Quando é desenvolvida uma aplicação nativa, o programador tem como alvo uma plataforma específica, utilizando o seu SDK e respetivas *frameworks*. A aplicação está limitada a essa plataforma, mas permite tirar partido e aceder a todas as funcionalidades que a plataforma pode oferecer. São aplicações com maior qualidade, melhor desempenho e mais seguras que oferecem uma experiência do utilizador notável, relativamente às aplicações não nativas. No caso de ser necessário reescrever a aplicação nativa para outras plataformas, a equipa de desenvolvimento deve ser mais experiente e ter o *know-how* necessário, implicando um maior investimento monetário e de tempo por parte do cliente. A *Mobile Web* baseia-se no conceito das *Web Apps*, aplicações desenvolvidas especificamente para serem executadas num *Web Browser*. Utilizando linguagens *Web* e através da otimização para os diferentes dispositivos, proporcionam uma experiência ao utilizador bastante boa através de um custo baixo. O código produzido é bastante inferior e tem um potencial de utilizadores máximo. A segurança depende do *Browser* que estiver a utilizar e é muito flexível para receber atualizações. O seu funcionamento só é possível se houver uma

ligação à internet, estando o acesso ao seu conteúdo e funcionalidades única e exclusivamente *online*, podendo acarretar custos acrescidos ao utilizador. Por fim, as aplicações híbridas, apesar de se encontrarem em constante evolução ao ponto de acederem a mais funcionalidades nativas, ainda não o fazem totalmente. Utilizam linguagens *Web* e *Web Views*, com o intuito de imitar as aplicações nativas. Todavia as funcionalidades, desempenho, interface nativo e comportamento é mais limitado. Conseguem, tal como as aplicações nativas, garantir algumas funcionalidades *offline*. Porém, a segurança e a estabilidade das aplicações podem ser um problema.

Na Tabela 2, encontram-se de forma reduzida os critérios de decisão e a avaliação dos mesmos para cada abordagem.

*Tabela 2- Fatores de Decisão Abordagens de Desenvolvimento adaptado de (Dalmasso, Datta, Bonnet, & Nikaein, 2013)*

<b>Critério de Decisão</b>	<b>Abordagem Nativa</b>	<b>Abordagem Multiplataforma Web Mobile</b>	<b>Abordagem Multiplataforma Híbrida</b>
<b>Qualidade da Experiência do Utilizador</b>	Excelente	Muito Boa	Não tão boa como as aplicações nativas
<b>Qualidade das Aplicações</b>	Alta	Media	Media baixa
<b>Potencial de Utilizadores</b>	Limitada a uma plataforma em particular	Máxima	Alargada, abrange diferentes plataformas
<b>Custo de Desenvolvimento</b>	Alto	Baixo	De Medio a Baixo
<b>Segurança da Aplicação</b>	Excelente	Depende da segurança do Browser	Fraca
<b>Suporte</b>	Complexo	Simple	De Medio a Complexo
<b>Facilidade de Realizar Updates</b>	Complexo	Simple	De Medio a Complexo
<b>Apta a Extensões</b>	Sim	Sim	Sim

Inicialmente, este projeto tinha como objetivo o desenvolvimento de uma aplicação que fosse suportada pelas plataformas Android, IOS e Windows Phone. Para esse desenvolvimento foi sugerido a utilização de uma abordagem multiplataforma híbrida. No entanto, com o levantamento do estado da arte das plataformas e abordagens de desenvolvimento, a empresa optou nativamente e o desenvolvimento foi feito à imagem desta opção. Esta mudança, obrigou

a uma reformulação do tempo e dos objetivos do projeto. A empresa definiu também a ordem de preferência do desenvolvimento, onde a plataforma Android surge em primeiro lugar e será alvo da aplicação resultante deste projeto.

### 3.3 Plataformas

Como podemos verificar na Figura 5, esta secção é referente ao ramo das plataformas *mobile*, representado no mapa de conhecimento anteriormente desenvolvido e retratará as três plataformas que a empresa considerou mais interessantes.

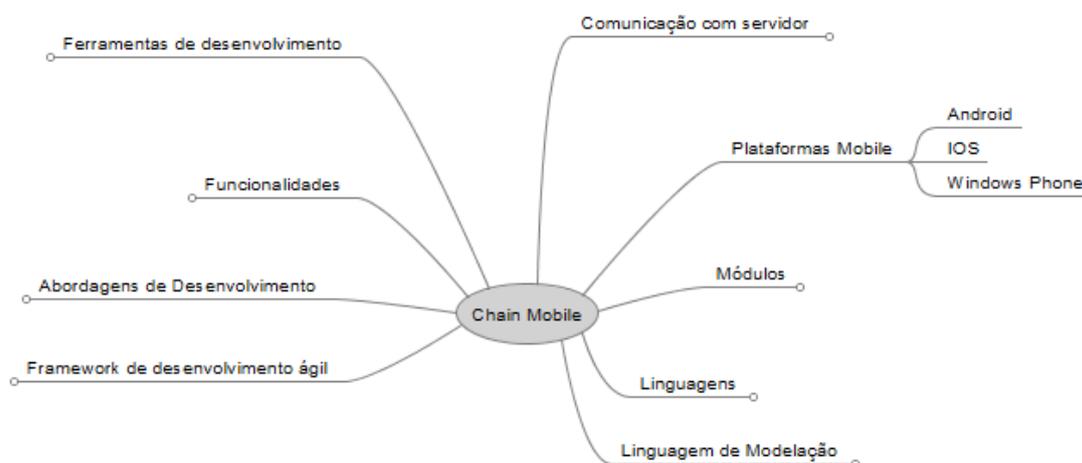


Figura 5 - Mapa de Conhecimento - Plataformas Mobile

Inicialmente, o projeto incidia no desenvolvimento de uma aplicação que fosse suportada pelas plataformas Android, IOS e Windows Phone. Com a opção, por parte da empresa, por uma aplicação nativa, houve a necessidade de esta escolher uma plataforma para se iniciar o desenvolvimento. Neste ponto, encontra-se uma breve apresentação das três plataformas e segue-se uma comparação entre elas, englobando toda a informação necessária à empresa para a tomada de decisão. Essa informação tem grande foco na análise de mercado e de vendas das três marcas e em algumas características técnicas que diferem entre elas.

#### 3.3.1 Android

Fundada em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White a Android Inc foi em 2005 adquirida pela Google (Bloomberg, 2014; FRED VOGELSTEIN, 2014). Após esta aquisição, a Google posicionou-se no mercado *mobile* e em 2008, foi oficialmente estabelecido um consórcio de empresas tecnológicas denominado de Open Handset Alliance, onde a Google estava incluída e através da plataforma Android detinha um papel de relevo (Snadberg & Rollins, 2013; Open Handset Alliance, 2014). Aquando dessa mesma oficialização foi revelada a primeira versão da

plataforma Android que desde então não parou de evoluir, contando já com diversas versões sendo a mais recente o Android L (Lollipop).

O Android é uma plataforma *mobile* desenvolvida pela Google (2014), que a define na sua página *Web*, como sendo a plataforma *mobile* mais popular, totalmente personalizável e fácil de usar. Foi projetada inicialmente apenas para *smartphones*, porém, hoje em dia já é suportada por diferentes dispositivos, nomeadamente *tablets*, relógios, TVs e por sistemas integrados em automóveis (Gandhewar & Sheikh, 2010; Google, 2014). É uma plataforma *open source* desenvolvida a partir da versão 2.6.25 do *kernel* do Linux, que já se encontra em mais de um bilhão de dispositivos. A cada nova versão que surge é-lhe atribuído o nome de uma sobremesa, pois segundo a Google (2014), estes dispositivos tornam a nossa vida mais doce.

### 3.3.2 IOS

O IOS é um sistema operativo *mobile* criado pela Apple que teve a sua origem em 2007 quando Steve Jobs apresentou o primeiro iPhone ao mundo. Inicialmente não tinha um nome especificamente atribuído, tendo este sido conferido aquando do aparecimento do SDK, tornando-se o iPhone OS (Rhiannon Williams, 2014). Baseado no MAC OS X, o IOS tornou-se ao longo do tempo num dos sistemas operativos mais utilizados e populares entre os utilizadores dos dispositivos *mobile*.

Até à sua versão atual (IOS8), o sistema operativo *mobile* da Apple evoluiu e sofreu grandes alterações, passando a suportar o iPad e o Ipod Touch (Apple, 2014). Para o IOS 8, a Apple (2014) criou uma linguagem de programação que caracteriza como “mais poderosa e intuitiva” para facilitar o desenvolvimento de aplicações para o IOS 8. Nos IOS, anteriores a linguagem que era utilizada para o desenvolvimento de aplicações era o Objective-C.

### 3.3.3 Windows Phone

A Microsoft (2014), define o Windows Phone como “tão exclusivo como cada utilizador”. É um sistema operativo *mobile* criado pela Microsoft, que surgiu da reformulação do seu predecessor Windows Mobile. A sua primeira versão surgiu em 2010 com o nome Windows Phone 7 e a versão atual é a Windows Phone 8.1 com o nome de código Blue. Baseado no Windows CE 7, é o mais novo dos três sistemas operativos *mobile*. Recentemente, com a aquisição da Nokia, a Microsoft reforçou o seu investimento no mundo *mobile*, tendo garantido com essa aquisição patentes registadas e um maior espaço de progressão.

### 3.3.4 Comparação das plataformas

Quando uma empresa decide desenvolver uma aplicação para uma determinada plataforma *mobile*, é importante estar informada da popularidade dessa plataforma, do número de utilizadores que a utilizam ou que potencialmente a irão utilizar, da quota de mercado que esta ocupa, da evolução ocorrida ao longo dos anos e da segurança que a marca lhes apresenta para o investimento. Desta forma, segue-se uma análise do mercado aos últimos anos com o intuito de responder às dúvidas da Doctron quanto à plataforma indicada para investir.

Segundo uma notícia da IDC – International Data Corporation (2014), o mercado global dos *smartphones* atingiu a marca histórica de 1 bilhão de remessas em 2013, onde o Android e o IOS foram os grandes contribuidores atingindo 93,8% das remessas de todo o ano. Em detalhe, podemos consultar na Tabela 3 os valores do estudo referente ao ano de 2012 e de 2013, apresentando-se o total das remessas em milhões e a diferença entre os anos, que reflete o crescimento de um ano para o outro:

Tabela 3 - Estudo Sistemas Operativos Mobile 2012-2013 retirado de (IDC, 2014)

Sistema Operativo	Remessas 2013	Quota de Mercado 2013	Remessas 2012	Quota de mercado 2012	Diferença entre anos
Android	793.6	78.6%	500.1	69.0%	58.7%
iOS	153.4	15.2%	135.9	18.7%	12.9%
Windows Phone	33.4	3.3%	17.5	2.4%	90.9%
BlackBerry	19.2	1.9%	32.5	4.5%	-40.9%
Outros	10.0	1.0%	39.3	5.4%	-74.6%
<b>Total</b>	<b>1009.6</b>	<b>100.0%</b>	<b>725.3</b>	<b>100.0%</b>	<b>39.2%</b>

Através da análise da Tabela 3 e dos três sistemas operativos que a empresa se focou, destaca-se largamente o Android, que só no último ano vendeu mais dispositivos que todos os outros juntos. Segue-se o IOS e por fim a Microsoft que ainda só é detentora de uma pequena parcela de mercado com o seu Windows Phone. Já no estudo de Yoon (2012) e no estudo de Snadberg e Rollins (2013), eles definiam o Android e o IOS como os principais sistemas operativos no mundo *mobile*. Com a recente aquisição da Nokia, a Microsoft poderá aumentar a sua quota de mercado (Microsoft, 2014; IDC, 2014). Importante referir, que das três plataformas alvo, a IOS foi a que apresentou o menor crescimento de um ano para o outro, enquanto o Windows Phone teve um crescimento avultado de 90,9%, seguido pelo Android com um crescimento de 58,7%.

Para além do estudo das remessas, da evolução e da quota de mercado, o IDC (2014) calculou a média de preços, em dólares, associados aos *smartphones* de acordo com a plataforma que utilizam. Encontramos essa análise na Figura 6.

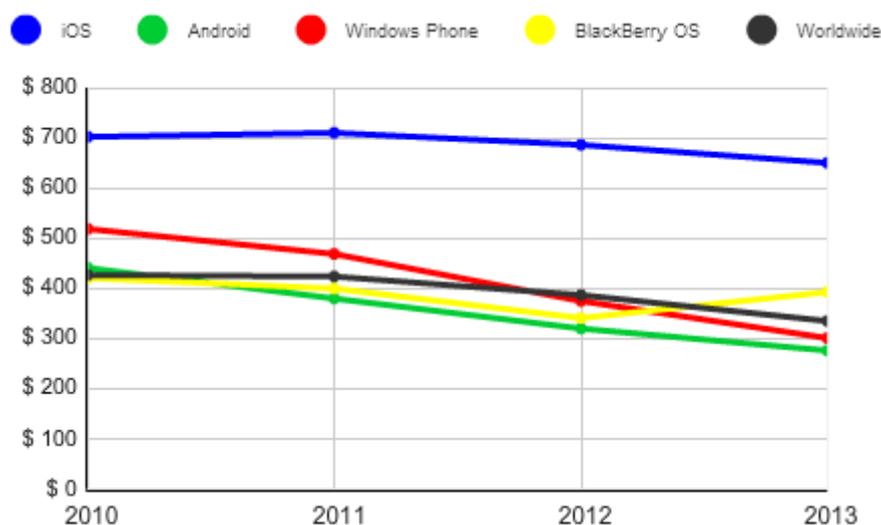


Figura 6 - Média do preço dos dispositivos por plataforma a nível global, retirado de (IDC, 2014)

Como podemos constatar, analisando a Figura 6, os dispositivos com a plataforma IOS lideram a tabela de preços com uma média bastante superior aos restantes dispositivos ao longo dos anos. É possível verificar também que, os dispositivos com a plataforma Android contrastam com os dispositivos IOS, com a média de preços mais baixa entre as diferentes plataformas. Os dispositivos com Windows Phone, garantiram em 2013 a segunda média de preços mais baixa. De destacar também a redução da média de preços ao longo dos anos e em todos os equipamentos, com exceção dos apetrechados com o Blackberry OS.

No âmbito do mercado nacional e segundo a revista Exame Informática (2014) que cita os dados mais recentes da GfK referentes ao período entre fevereiro e setembro de 2014, a Samsung foi o fabricante de *smartphones* que mais vendeu equipamentos livres em Portugal. Também segundo esse estudo, a Apple apenas atingiu 6,5% da cota de mercado, refletindo uma subida de 2,5% e a Nokia, após ter sido adquirida pela Microsoft, representou uma queda de 3%, comparativamente a 2013, ocupando 3,2% da cota de mercado neste período. A seguir à Samsung, com 48,3% atribui o segundo lugar à Wiko, que com o seu primeiro ano de presença em Portugal, atingiu os espantosos 14,2%, com os seus dispositivos Android. A LG com os seus dispositivos Android e Windows Mobile, obteve uma cota igual à da Apple. São ainda referidas a BQ e a Huawei, atingindo 4% e 3,9% respetivamente, com os seus dispositivos Android. É possível concluir através deste estudo que em Portugal, de fevereiro a setembro de 2014, a plataforma Android se encontrava presente na maioria dos dispositivos vendidos.

A APDC - Associação Portuguesa para o Desenvolvimento das Comunicações (2014) refere um estudo anual levado a cabo pela Millward Brown, que indica que a Google ultrapassou a Apple como marca mais valiosa do mundo, contribuindo para isso a desvalorização em 20% do valor

da Apple, que se encontra em segundo lugar, estando em terceiro lugar a IBM e seguindo-se a Microsoft.

Através da análise do estudo acima referido, é possível concluir que a plataforma Android é a que abrange um maior número de utilizadores a nível mundial e a que possui equipamentos cuja média de preço é mais reduzida, o que poderá representar um maior número de potenciais clientes. Contudo, existem outros fatores de natureza técnica, que devem ser analisados para as três plataformas focando o processo de desenvolvimento de aplicações, que podem, a par das conclusões supramencionadas, ter algum impacto referente à tomada de decisão por parte da Doctron. Na Tabela 4, compara-se lado a lado algumas das características técnicas das três plataformas (Apple, 2014; Google, 2014; Microsoft, 2014; JOŠT, HUBER, & HERIČKO, 2013; Goadrich & Rogers, 2011).

*Tabela 4 - Comparação Técnica das Plataformas mobile*

	<b>Android</b>	<b>IOS</b>	<b>Windows Phone</b>
<b>Fabricante</b>	Google	Apple	Microsoft
<b>Código</b>	<i>Open Source</i>	<i>Closed Source</i>	<i>Closed Source</i>
<b>Linguagem de Desenvolvimento Nativa</b>	Java	Objetive-C (até ao IOS7 inclusive)	C# ou VB
		Swift (IOS8)	
<b>Versões</b>	Android 1.0 (2008)	iPhone IOS1 (2007)	Windows Phone 7 (2010)
	Android 1.5 Cupcake(2009)	iPhone IOS2 (2008)	Windows Phone 7.5 Mango (2011)
	Android 1.6 Donut(2009)	iPhone IOS3 (2009)	Windows Phone 7.8 Mojo (2013)
	Android 2.0 Éclair(2009)	IOS4 (2010)	Windows Phone 8 Apollo (2012)
	Android 2.2 Froyo(2010)	IOS5 (2011)	Windows Phone 8 Apollo (2012)
	Android 2.3 Gingerbread (2010)	IOS6 (2012)	Windows Phone 8.1 Blue (2014)
	Android 3 Honeycomb (2011)	IOS7 (2013)	
	Android 4 Ice Cream Sandwich (2011)	IOS8 (2014)	
	Android 4.1 Jelly Bean (2012)		
	Android 4.4 Kit Kat (2013)		
	Android 5 Lollipop(2014)		
<b>Kernel</b>	Linux	OS X	Windows CE 7
<b>Idade</b>	6 anos	7 anos	4 anos
<b>Cloud</b>	Google Sync	iCloud	Skydrive
<b>Personalização</b>	Totalmente	Limitada	Totalmente
<b>Hardware Suportado</b>	iPhone, iPad, e iPod Touch	Larga variedade	Variedade Média
<b>Mercado</b>	Google Play Store	App Store	Windows Marketplace

	Android	IOS	Windows Phone
<b>Preço de Licença de Desenvolvimento</b>	Baixo 25 Dólares Taxa Única	Alto 99 Dólares/ano	Médio-Alto 19 Dólares/ano (individual) 99 Dólares/ano (empresarial)
<b>Custo das Ferramentas de Desenvolvimento</b>	Grátis	Grátis	Grátis
<b>Web Services Suportados Nativamente</b>	RESTful	RESTful	RESTful e SOAP
<b>Hardware de Desenvolvimento</b>	Qualquer equipamento com Windows, Linux ou Mac OS	Macintosh	Qualquer equipamento com Windows, Linux ou Mac OS

Após proceder a uma análise cuidada de toda esta informação, a empresa (Doctron) priorizou o desenvolvimento para a plataforma Android, colocando as plataformas IOS e Windows Phone em segundo e terceiro lugar respetivamente.

### 3.4 Ferramentas

Este subcapítulo aborda o ramo do mapa de conhecimento relacionado com as ferramentas de desenvolvimento. Esse ramo está representado na Figura 7, onde estão identificadas algumas das ferramentas existentes para abordagens multiplataforma e aquelas que, através da informação recolhida junto de empresas, foram identificadas como indicadas para o desenvolvimento nativo.

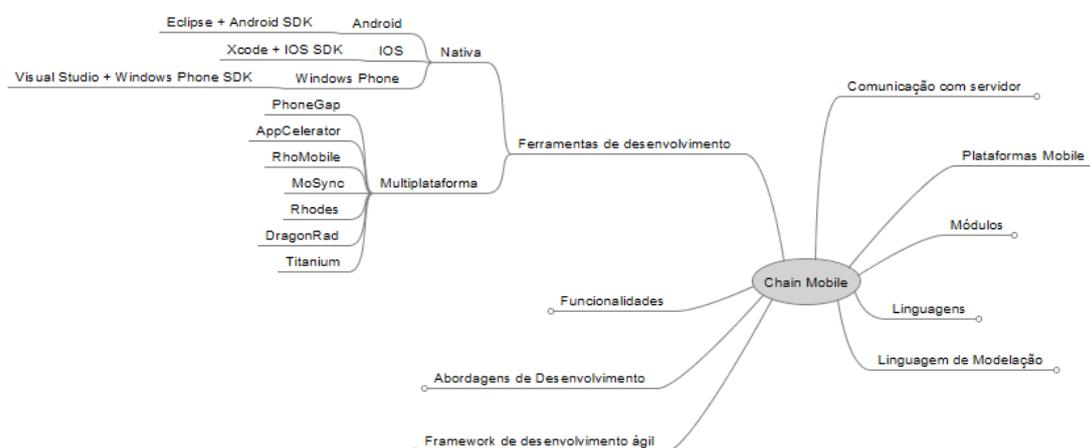


Figura 7 - Mapa de conhecimento - Ferramentas de desenvolvimento

Inicialmente, o objetivo deste projeto era desenvolver uma aplicação suportada nas plataformas Android, IOS e Windows Phone. No intuito do desenvolvimento multiplataforma, são inúmeras as ferramentas de desenvolvimento existentes atualmente. Num estudo elaborado por (Palmieri, Singh, & Cicchetti, 2012), encontra-se uma análise que associa o suporte de determinadas ferramentas de desenvolvimento multiplataforma às funcionalidades nativas das plataformas Android, IOS e Windows Phone (Tabela 5).

Tabela 5 - Ferramentas de desenvolvimento multiplataforma e funcionalidades nativas suportadas adaptada de (Palmieri, Singh, & Cicchetti, 2012)

API	Rhodes (JavaScript)	PhoneGap (JavaScript)	MoSync (JavaScript)	MoSync (C, C++)	DragonRad
Acelerómetro	Não	Sim	Sim	Não	Não
Código de Barras	Sim	Sim	Não	Não	Sim
Bluetooth	Sim	Sim	Não	Sim	Não
Calendário	Sim	Sim	Sim	Sim	Sim
Câmara	Sim	Sim	Não	Sim	Não
Captura de imagens	Não	Sim	Sim	Sim	Sim
Compasso	Não	Sim	Sim	Não	Não
Conexão	Não	Sim	Sim	Sim	Não
Contactos	Sim	Sim	Não	Não	Sim
Dispositivo	Sim	Sim	Sim	Sim	Sim
Ficheiros	Sim	Sim	Sim	Sim	Não
Geolocalização	Sim	Sim	Sim	Sim	Sim
Menu	Sim	Não	Não	Não	Sim
NFC	Sim	Sim	Sim	Sim	Sim
Notificações	Sim	Sim	Sim	Sim	Não
Rotação do Ecrã	Sim	Sim	Não	Sim	Não
Armazenamento	Sim	Sim	Sim	Sim	Sim

Como podemos comprovar na Tabela 5, nenhuma das ferramentas suporta todas as funcionalidades nativas enumeradas, suporte esse, que é possível através da utilização de ferramentas de desenvolvimento nativo.

Mais recentemente, Pastore (2014) também menciona ferramentas multiplataforma de desenvolvimento *Web mobile* abordando as ferramentas JQuery Mobile e Zepto e no conceito de híbridasmenciona o Phonegap, Titanium, AppCelerator, RhoMobile e Rhodes como ferramentas com suporte para as plataformas Android, IOS e Windows Phone.

Estas são apenas algumas das muitas ferramentas encontradas através de pesquisas realizadas na *web*. A empresa com base no estudo anterior, considerou suficiente o referido estudo e deixa de fazer sentido incluir uma análise alargada a este tipo de ferramentas.

Tendo a empresa optado pelo desenvolvimento nativo e tendo como alvo inicial, e deste trabalho, a plataforma Android, houve a necessidade de definir as ferramentas a utilizar para o desenvolvimento da solução *mobile*. Com esse intuito, foi consultado o site da Google (2014) e efetuado um aconselhamento perante algumas empresas na área do desenvolvimento Android a fim de obter suporte à tomada de decisão.

A informação adquirida através das empresas foi unânime, convergindo com a utilização IDE (*Integrated Development Environment*) Eclipse com o ADT (*Android Developer Tools*) em conjunto com o Android SDK (*Software Development Kit*) disponibilizado pela Google. Em alguns casos foi mencionado o Android Studio Beta, mas nenhuma das empresas contactadas ainda o tinha testado. Através destes pareceres e também pelo facto do Android Studio ainda se encontrar numa versão beta, não oferecendo suporte ao NDK (*Native development Kit*) em contraste com algumas vantagens, nomeadamente na escrita do código, decidiu-se excluir o Android Studio Beta. Neste projeto será utilizado o IDE Eclipse com o ADT, em conjunto com o Android SDK disponibilizado pela Google. Para testes será utilizado o emulador oferecido com o SDK e um dispositivo *mobile* com a plataforma Android 4.2 Jelly Bean.

### 3.5 Linguagem

No subcapítulo anterior, verificamos que as linguagens utilizadas pelas ferramentas de desenvolvimento podem ser diferentes, variando de acordo com a ferramenta escolhida. No caso do desenvolvimento desta solução será de acordo com a ferramenta escolhida e do tipo de abordagem que a empresa optou, utilizando a linguagem JAVA. Este tópico aborda essa linguagem de programação empregada no desenvolvimento desta solução e representada na Figura 8 do mapa de conhecimento, no ramo das linguagens.

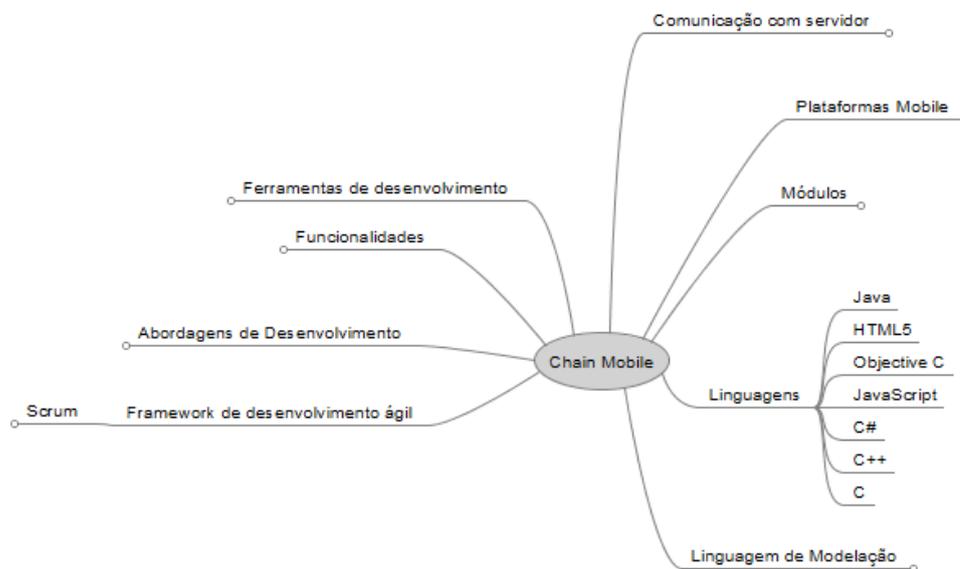


Figura 8 - Mapa de conhecimento - Linguagens de desenvolvimento

O JAVA é uma linguagem de programação que começou a ser desenvolvida em 1991 por uma equipa de engenheiros da Sun Microsystems apelidada de "Green Team". Essa equipa era

liderada por James Gosling, que acreditava que o futuro convergia para a interoperabilidade entre os dispositivos digitais utilizados pelos consumidores e os computadores. Em 1995 foi incorporada pela primeira vez a tecnologia JAVA num *browser* chamado NetScape. Em 2010, após um longo período de cooperação entre a ORACLE e a Sun Microsystems, esta foi adquirida pela ORACLE (ORACLE, 2014).

É uma linguagem orientada a objetos baseada no conceito WORA (*Write Once Run Anywhere*), que significa escreve uma vez corre em qualquer lado, o que lhe concede a característica de portabilidade. O seu modelo conceptual está representando na Figura 9.

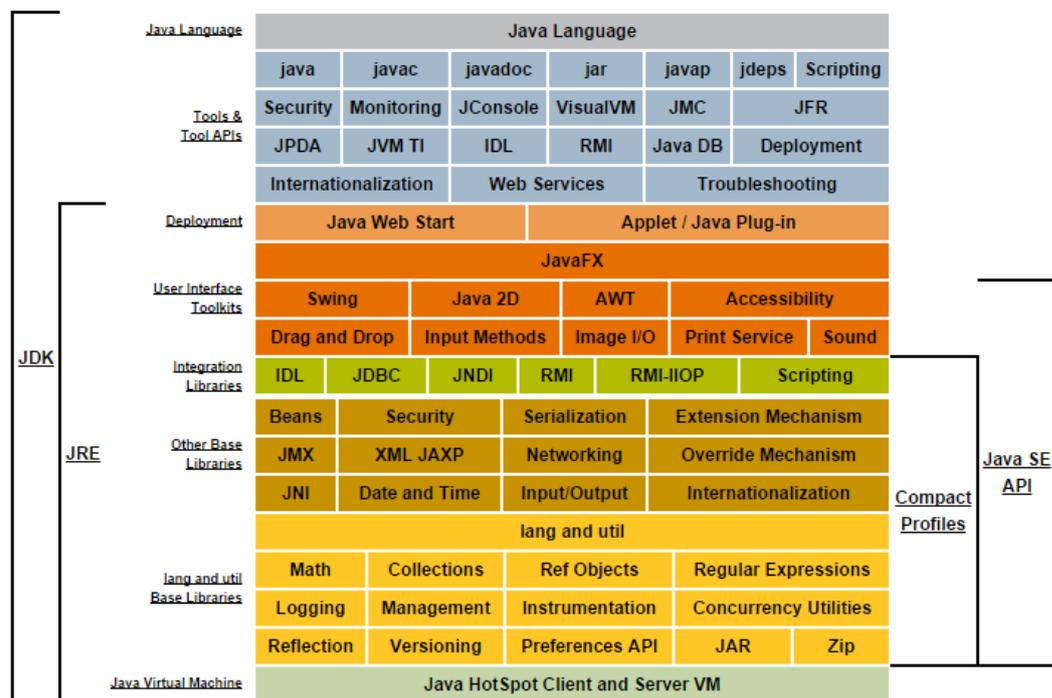


Figura 9 - Modelo conceptual da Linguagem JAVA retirado de (ORACLE, 2014)

Segundo Liang (2014), o JAVA é uma linguagem completa e versátil, utilizada no desenvolvimento de aplicações robustas, não só para a Web, mas também para *desktops*, servidores e dispositivos *mobile*. É especialmente utilizada na plataforma Android, apesar das respetivas API's denotarem algumas diferenças (Darwin, 2014).

Hoje em dia e segundo a ORACLE (2014), a linguagem de programação JAVA é o padrão global para o desenvolvimento e distribuição de aplicações *mobile*, conteúdo baseado na *web* e *softwares* corporativos, sendo a base para praticamente todas as aplicações em rede qualquer que seja o seu tipo. De acordo com a página *web* IEEE Spectrum (2014), o JAVA é a linguagem com maior popularidade. A Figura 10, representa o Top 30 normalizado de linguagens de programação, elaborado após análises à popularidade e utilização das diversas linguagens de

programação em diferentes fontes, identificando a linguagem JAVA como líder absoluto no contexto da Web, mobile e empresarial.

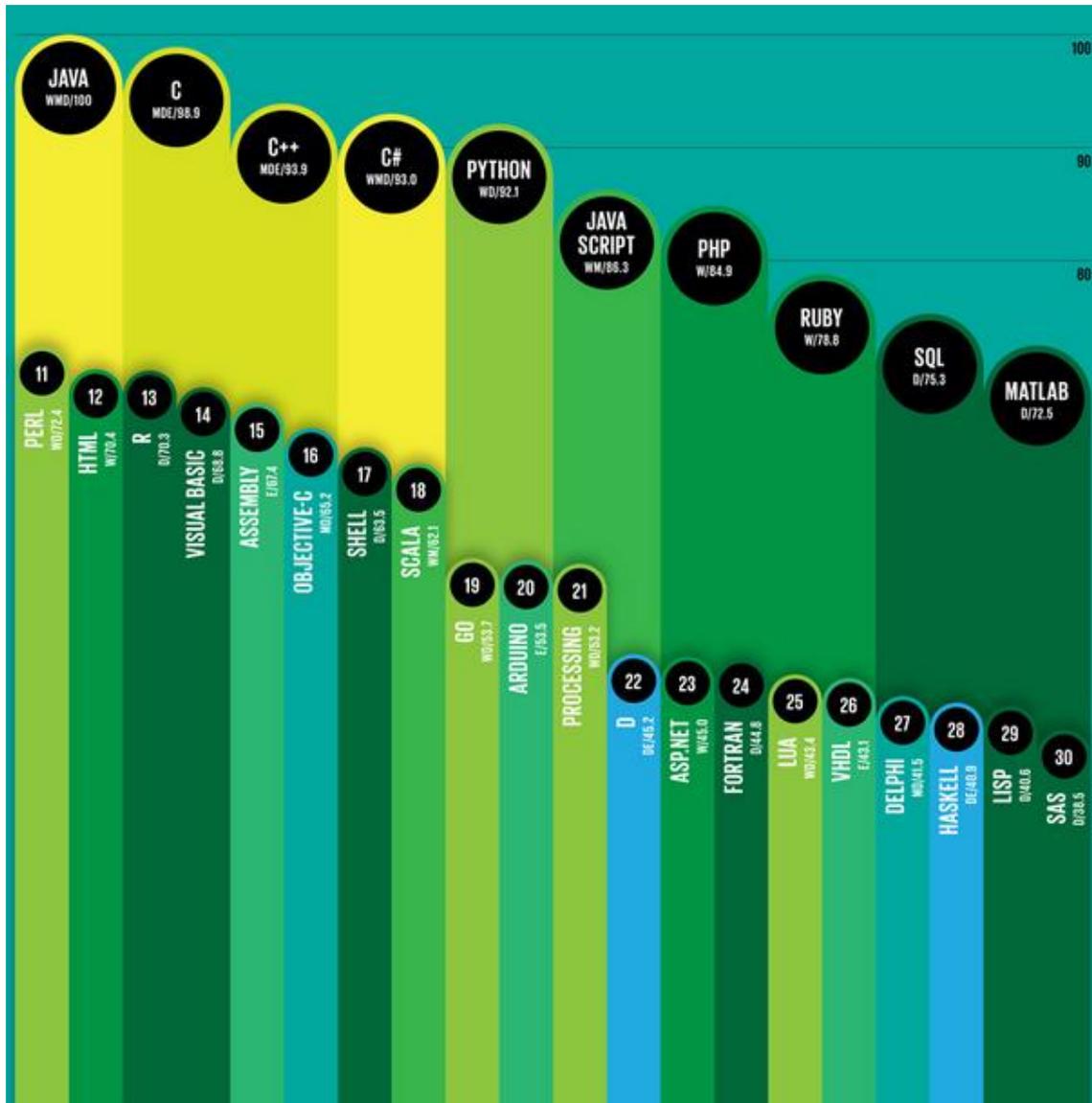


Figura 10 – Top 30 popularidade das linguagens de programação retirado de (IEEE Spectrum, 2014)

### 3.6 Web Services

Na Figura 11, referente ao ramo da comunicação com o servidor no mapa de conhecimento, verificamos que existem dois tipos de *Web Services*, que podem ser utilizados para o desenvolvimento da solução.

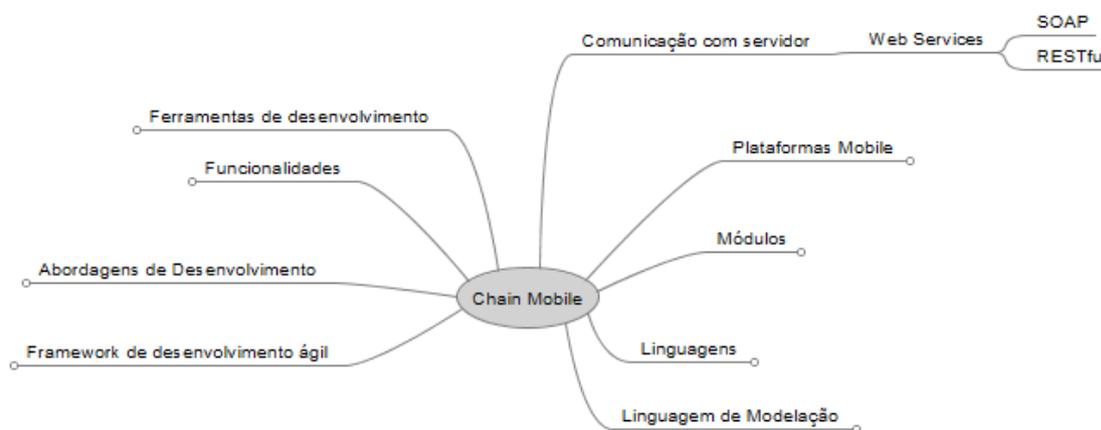


Figura 11 - Mapa de conhecimento - Web Services

Como o próprio nome indica, um *Web Service* é um serviço disponível na *web*. Funciona como uma *framework* de comunicação entre dois dispositivos através da internet (Tidwell, Snell, & Kulchenko, 2001). Constitui uma técnica de *software* distribuído, cujos componentes podem ser implementados e executados em aparelhos fisicamente distintos independentemente das suas linguagens de programação, bibliotecas de suporte, sistemas operativos e *hardware* (Kalin, 2013). Esta heterogeneidade só é possível dada a existência de uma camada intermediária, que lida com as diferenças nos meta dados e linguagens utilizados pelo cliente e pelo servidor, sendo as linguagens do cliente e do servidor traduzidas para uma linguagem universal no formato XML, que desempenha esse papel (Kalin, 2013; Sharif, Gull, & Nazir, 2013).

No desenvolvimento de uma aplicação, o programador pode escrever em qualquer linguagem que tenha preferência e com as bibliotecas de apoio adequadas. Na verdade, uma das características dos *Web Services* é a transparência de linguagem, isto é, o servidor e o cliente podem comunicar independentemente da linguagem que utilizem. A transparência de linguagem é um dos principais contribuintes para a interoperabilidade dos *Web Services*, que se resume à capacidade dos *Web Services* e dos seus consumidores interagirem perfeitamente entre si, apesar das diferenças de linguagens de programação, bibliotecas de apoio, sistemas operativos e plataformas de *hardware* (Kalin, 2013; W3Schools, 2014).

Atualmente, os *Web Services*, dividem-se em duas categorias: os baseados em SOAP e os do estilo RESTful (Kalin, 2013; Mumbaikar & Padiya, 2013; Jendrock, et al., 2014). Ambos tem um papel importante na arquitetura do *software* da atualidade, utilizando para comunicação protocolos baseados em padrões na *web* (Leite, Lago, Gerosa, & Kon, 2013). Estes dois tipos,

apesar das suas diferenças, tornam possível utilizar o *software* já existente, disponibilizando os seus recursos para qualquer aplicação cliente, que invocando os seus métodos, possa aceder e utilizar esses recursos, sem a necessidade de estes serem fornecidos através de uma nova programação. Através do consumo de um *Web Service*, é possível aceder à informação que se encontra noutra aplicação e publicá-la para o resto do mundo (W3Schools, 2014).

A nível empresarial, os *Web Services*, permitem a interoperabilidade de todos os sistemas da empresa, removendo todo e qualquer tipo de incompatibilidades entre eles, possibilitando a comunicação eficiente e sem problemas, entre todas as aplicações utilizadas pela empresa (Panetto & Cecil, 2013). Conferem também um incremento à segurança através de uma comunicação dinâmica sem intervenção humana. Através dessa integração, é possível oferecer ao cliente os mesmos serviços, noutras plataformas e sistemas, reutilizando os já desenvolvidos sem ter que recriá-los, poupando recursos e alargando a oferta, criando valor acrescentado.

Em suma, os *Web Services*, podem ser publicados e consumidos em várias plataformas de *hardware* e em diferentes sistemas operativos, constituindo uma excelente forma de integrar diversos sistemas de *software*, utilizando protocolos baseados em padrões da *web*, que permitem a troca de informações entre diferentes aplicações e plataformas, permitindo ao programador comunicar na linguagem que desejar (Kalin, 2013). É um serviço que está disponível na *web* e o seu propósito é ser consumido por uma aplicação, fornecendo informação sem problemas de interoperabilidade. Os *Web Services* têm como objetivo permitir interações aplicação-aplicação através da *web* e integrar a infraestrutura de rede existente com a *web* (Sharif, Gull, & Nazir, 2013). Através desta tecnologia, uma aplicação pode invocar outra para realizar tarefas simples ou complexas.

### 3.6.1 SOAP

Como já vimos anteriormente, um *Web Service* é um serviço que está disponível na *Web* e que permite a comunicação entre um cliente e um servidor fisicamente separados. Esse serviço pode ser do tipo SOAP (*Simple Object Access Protocol*) ou RESTful (*REpresentational State Transfer*). O SOAP é um protocolo baseado em XML que emprega o protocolo HTTP, o protocolo mais utilizado da internet. No entanto, pode também empregar HTTPS, SMTP, FTP ou qualquer outro protocolo de transporte, para efetuar a comunicação cliente-servidor na troca de informação estruturada. (Chandio, Zhu, & Sodhro, 2012; Cerami, 2002; Lee, Lee, & Wang, 2014; Kalin, 2013). A informação é trocada sob a forma de mensagens, onde cada mensagem (*Envelope*) é constituída por um cabeçalho (SOAP *Header*) opcional e por um corpo de mensagem (SOAP *Body*) obrigatório. A Figura 12 representa a estrutura de uma mensagem (SOAP *Envelope*).

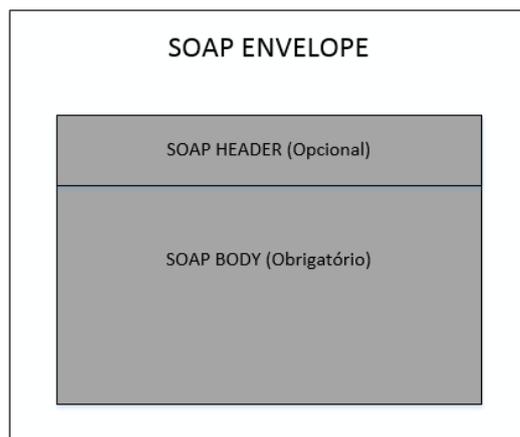


Figura 12 - Estrutura da mensagem SOAP

Um *Web Service* do tipo SOAP rege-se pelas boas práticas de interoperabilidade dos *Web Services*, definidas pela WS-I, que ao longo dos anos desenvolveu especificações que regem as tecnologias que lhe estão associadas, nomeadamente para o WSDL, SOAP e UDDI (WS-I, 2014; OASIS, 2014). Para uma melhor compreensão, serão brevemente explicadas as tecnologias que os *Web Services* do tipo SOAP tiram partido:

**SOAP (Simple Object Access Protocol):** como já foi referido anteriormente, é um protocolo baseado em XML utilizado na comunicação entre o cliente e o servidor. Permite invocar serviços através de qualquer protocolo de transporte, sendo o mais utilizado o HTTP. O SOAP, ao contrário do RESTful, suporta chamadas síncronas e assíncronas (Sharif, Gull, & Nazir, 2013).

**WSDL (Web Services Description Language):** Ou em português, Linguagem de Descrição dos *Web Services*, é uma linguagem baseada no XML, contendo a descrição do *Web Service* e caracteriza particularmente os *Web Services* do tipo SOAP (W3schools, 2014). É através deste que são descritas as informações do serviço, como invocá-lo e quais métodos e operações podem ser requisitadas (Chandio, Zhu, & Sodhro, 2012). Através da análise do WSDL é possível identificar o *SOAP-ACTION*, *Namespace*, *URL* e os nomes dos métodos necessários à sua utilização. É importante também referir que os *Web Services* do estilo RESTful não utilizam esta tecnologia de descrição.

**XML (Extensible Markup Language):** em português significa linguagem de marcação extensível e é a linguagem dedicada à troca de informações com o *Web Service*. Para existir essa troca é necessário que os pedidos sejam enviados através de mensagens de XML. Serve de camada intermediária, pois suporta as diferenças de linguagens e meta dados das diferentes plataformas para a troca e processamento de documentos estruturados. (kalin, 2013).

**XSD (XML Schema definition):** Traduzido para o português como definição do esquema XML. É responsável pela validação dos parâmetros recebidos nos métodos do *Web Service*. Através do XSD é possível verificar se a informação que está a chegar é a pretendida. Contem todas as informações que o *Web Service* espera, funciona como um *layout*, englobando os elementos, atributos, hierarquia dos elementos, tipos de dados e valores que os elementos podem tomar (W3C, 2014).

**UDDI (Universal Description, Discovery and Integration):** É um protocolo que permite a organização e o registo de *Web Services*. É um diretório onde as empresas podem registar e invocar *Web Services* (Sharif, Gull, & Nazir, 2013). Contém três tipos de informação, nomeadamente: informações gerais de cada organização, informações gerais e técnicas de serviços dispostos por área de negócio.

Na Figura 13, encontra-se arquitetura de um *Web Service* do tipo SOAP, que demonstra como um servidor e um cliente interagem, utilizando este tipo de *Web Services*. Verificamos que o servidor é o prestador de serviços e que este regista os seus serviços num diretório, onde o cliente, que é o requerente do serviço os vai procurar. Quando o cliente encontra o serviço pretendido, irá interagir com o servidor através de mensagens SOAP, invocando os métodos disponíveis no serviço e obtendo a informação do servidor.

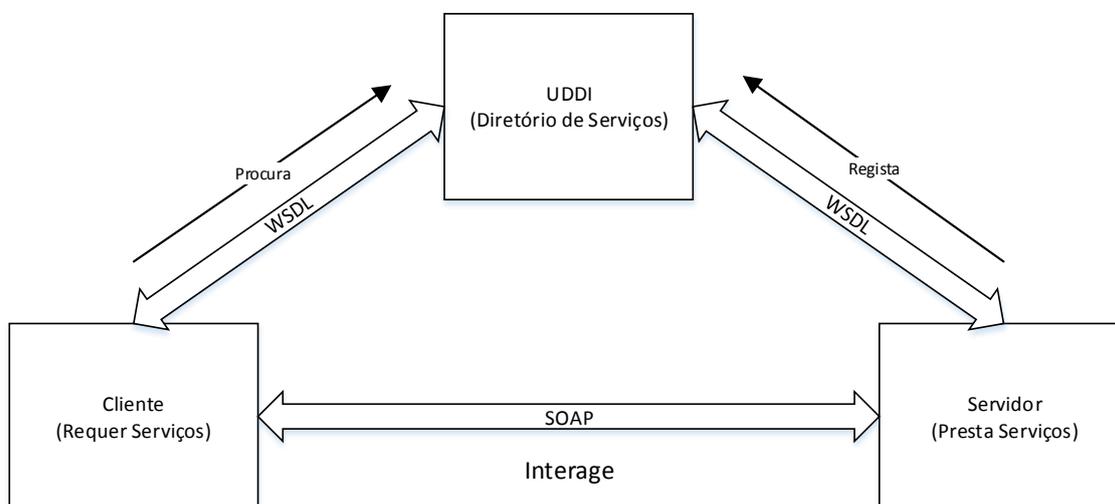


Figura 13 - Arquitetura SOAP adaptada de (Mumbaikar & Padiya, 2013)

### 3.5.1.1 KSOAP2

Na plataforma Android, existe suporte nativo para a utilização de *Web Services* do estilo RESTful o que não acontece para *Web Services* do tipo SOAP (Oliveira & Castro, 2013; Murphy, 2014). Visto que a empresa definiu que a comunicação seria efetuada através de *Web Services* do tipo

SOAP, surgiu a necessidade de encontrar uma alternativa desenvolvida por alguém, que pudesse ser utilizada para lidar com os *Web Services* SOAP. Segundo (Liu, Sung, & Wenli, 2013) e (Xu, Yin, & Rong, 2013), existe uma biblioteca desenvolvida por terceiros chamada ksoap2 que pode ser utilizada para lidar com este tipo de *Web Services*.

O ksoap2, é uma biblioteca de cliente SOAP leve e eficiente para Android, que utiliza o seu conjunto de classes para criar o SOAP *Envelope* e as propriedades que lhe estão associadas (Ravi, Mala, Srinivasan, & Sarukesi, 2013). É um projeto que desde o seu começo tem sido constantemente melhorado e expandido, de forma a incluir cada vez mais funcionalidades (ksoap2, 2014). Será também esta a biblioteca que será utilizada no desenvolvimento da solução de forma a lidar com os *Web Services* do tipo SOAP, permitindo a comunicação entre o CHAIN Mobile e o servidor.

Porém, esta biblioteca não suporta funcionalidades de segurança dos *Web Services* SOAP (Kleiner & Schneider, 2011). Apesar de ser possível desenvolver a solução segundo os requisitos do cliente utilizando esta biblioteca, é importante num futuro desenvolvimento criar uma extensão para conferir critérios de segurança mais exigentes, particularmente, conferir não-repudição por via de assinatura digital (por chave privada) à segurança dos *Web Services* ou converter os *Web Services* para o estilo RESTful que é suportado pelo SDK disponibilizado pela Google para o Android.

### 3.6.2 RESTful

O RESTful (*Representational State Transfer*) é um estilo de *Web Services* que surgiu depois do SOAP como uma alternativa mais simples. Contrastando com os *Web Services* do tipo SOAP não contêm WSDL com a descrição da sua existência, nem XML. Eles não precisam de nenhum *middleware* para efetivar a comunicação desde que haja suporte para HTTP/HTTPs. O resultado dos *Web Services* RESTful pode ser oferecido em XML, JASON, Plain Text ou JavaScript (Upadhyaya, Zou, Xiao, Ng, & Lau, 2011). É muito mais leve que o SOAP e pode ser implementado usando quase qualquer ferramenta. Dada a sua simplicidade, proporcionam uma aprendizagem mais rápida. No entanto, os clientes precisam de saber o que enviar e o que vão receber. Como os RESTful não facultam qualquer informação sobre como os componentes se vão interligar ou sobre os seus elementos, isto obriga a que o cliente saiba especificamente o que o *Web Service* faz e as suas necessidades.

Ao contrário do que acontece no SOAP, que pode utilizar qualquer protocolo de transporte genérico, o RESTful cinge-se ao HTTP/HTTPs e tem como principais características (Verborgh, et

al., 2013; Upadhyaya, Zou, Xiao, Ng, & Lau, 2011; AlShahwan, Moessner, & Carrez, 2010; Mumbaikar & Padiya, 2013):

**Sem Estado:** Os pedidos do cliente não englobam um estado. Cada pedido contém toda a informação necessária ao servidor para ser processado.

**Permitem guardar dados no Cache (Cacheable):** Através da utilização da Cache, estes *Web Services* permitem armazenar dados que ficam acessíveis para o cliente, sem a necessidade de realizar um novo pedido ao servidor, isto é, dados requisitados anteriormente encontram-se armazenados na cache e não existe a necessidade de serem reenviados. O tempo de armazenamento desses dados também pode ser definido. Desta forma, são diminuídas as iterações necessárias, traduzindo-se numa melhoria na performance e numa redução da largura de banda necessária.

**Utilizam um sistema por camadas:** A utilização de camadas permite melhorar a escalabilidade do processo e balancear a carga dos serviços. Cada pedido é independente dos outros podendo ser tratado por diferentes intervenientes.

**URI (Uniform Resource Identifier):** Os recursos são facultados ao cliente e identificados através de um caminho único.

**Uniform Interface:** todos os recursos interagem através de um interface uniforme, que oferece ações baseadas em métodos HTTP nomeadamente, PUT, GET, POST e DELETE. Suporta *code on demand*.

### 3.6.3 RESTful VS SOAP

Para além da descrição dos diferentes tipos de *Web Services* disponíveis, é importante detalhar as suas diferenças para o processo de tomada de decisão. A Tabela 6 apresenta diferenças entre os *Web Services* do tipo SOAP e os do estilo RESTful (Mumbaikar & Padiya, 2013; AlShahwan, Moessner, & Carrez, 2010; Cerami, 2002; Lee, Lee, & Wang, 2014; Sivamani, Kwak, & Cho, 2014; Mazo, Otón, de-Marcos, García, & García, 2012; Sharif, Gull, & Nazir, 2013; JOŠT, HUBER, & HERIČKO, 2013).

Tabela 6 - Comparação SOAP RESTful adaptada de (AlShahwan, Moessner, & Carrez, 2010)

<b>Critério</b>	<b>SOAP</b>	<b>RESTful</b>
<b>Cliente/Servidor</b>	Ligação direta (contrato)	Ligação passa por vários intervenientes
<b>URI</b>	Um para o <i>Service Endpoint</i>	Um por recurso
<b>Protocolos de Transporte Suportados</b>	Todos	HTTP
<b>Armazenamento em <i>Cashe</i></b>	Não Suporta	Suporta
<b>Interface</b>	Não Uniforme (WSDL)	Uniforme
<b>Perceção do Conteúdo</b>	O cliente tem perceção do conteúdo do <i>Web Service</i>	Implícito no comportamento do <i>Web Service</i>
<b>Mensagens Assíncronas</b>	Suporta	Não suporta
<b>Tipo de dados</b>	Binário, necessita <i>parsing</i>	Suporta todos os tipos de dados diretamente
<b>Formato de Dados</b>	XML e anexos	Múltiplos
<b>Informação dos Métodos</b>	No corpo ( <i>body</i> ) do HTTP	Método HTTP
<b>Informação dos Dados</b>	No corpo ( <i>body</i> ) do HTTP	URI HTTP
<b>Descrição do Web Service</b>	WSDL	WADL
<b>Expansível</b>	Não, não permite <i>hyperlinks</i>	Sim e sem a necessidade de um novo <i>Web Service</i> (usando <i>xlink</i> )
<b>Padrões</b>	Específicos SOAP (WSDL, UDDI, WS-*)	Web Standards ( métodos HTTP, XML, MIME Types)
<b>Segurança</b>	WE-Security	HTTP Security
<b>Transações</b>	WE- AtomicTransaction	Implementação Complexa
<b>Largura de Banda necessária</b>	Maior	Menor
<b>Performance</b>	Menor	Maior
<b>Suporte Nativo em Plataformas Mobile</b>	Windows Phone	Android, IOS, Windows Phone

Analisando a Tabela 6, a grande vantagem dos *Web Services* do tipo SOAP é o facto de tirar partido de todas as boas práticas e especificações que lhe são proporcionadas pelo WS-\*, abrindo um leque de novas possibilidades, nomeadamente na segurança com o WE-Security e nas transações com o WE- AtomicTransaction, entre outros padrões que o suportam e garantem maior interoperabilidade no mundo empresarial não garantidas pelos RESTful. Funcionam em qualquer protocolo de transporte ao contrário dos RESTful. Por outro lado, os *Web Services* do estilo RESTful, são suportados nativamente por todas as plataformas alvo da empresa, garantindo um desempenho superior e uma utilização de recursos, presentes no dispositivo, mais reduzida como a largura de banda necessária ou a memória.

Apesar dos *Web Services* do tipo RESTful terem uma performance superior aos SOAP quer na comunicação com fios quer na comunicação sem fios, serem mais leves, fáceis e auto descritivos, com maior flexibilidade e menor sobrecarga, serem suportados nativamente no Android, IOS e Windows Phone, exigirem menor largura de banda e menor memória, terem uma performance superior e perceptível pelo utilizador, no desenvolvimento desta solução e por opção da empresa, a comunicação que irá existir entre a solução *mobile* e o servidor do CHAIN, será realizada através de *Web Services* do tipo SOAP (Mumbaikar & Padiya, 2013). Esta opção prendeu-se na sua maioria com o fato de a empresa já utilizar este tipo de *Web Services*.

### 3.7 – Unified Modeling Language (UML)

Na Figura 14, o ramo do mapa de conhecimento referente à linguagem de modelação, onde se encontra o UML, decompõe-se em três modelos diagramáticos distintos que serão utilizados na conceção da solução.

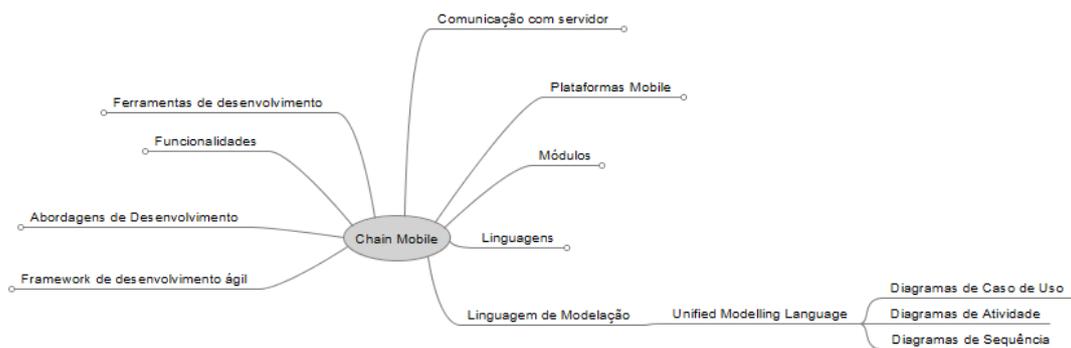


Figura 14 - Mapa de conhecimento - UML

No desenvolvimento de *software*, a análise e o *design* fornecem as diretrizes para o código. Essas diretrizes podem encontrar-se embutidas em modelos concebidos através da utilização da *Unified Modelling Language*, com base nos requisitos colocados pelo cliente (Mallya & Bhagat, 2013). O UML é uma linguagem de modelação que permite especificar e desenhar através de modelos, os requisitos de um cliente para um sistema de *software*. Estes modelos permitem uma visão do sistema e da comunicação entre os objetos (OMG, 2014). Serão modelados para este projeto e de acordo com os requisitos do cliente, os seguintes diagramas:

- Casos de uso

Oferecem uma visão externa do sistema com um nível alto de abstração. São elaborados em função dos requisitos do cliente e identificam as funcionalidades de um sistema, os autores e todos os relacionamentos existentes.

- Atividade

Os diagramas de atividade são definidos pela Microsoft (2014), como um fluxo de trabalho resultante de uma sequência de ações que constituem um processo de negócios ou um processo de *software*. Ações essas que podem ser executadas por pessoas, componentes de *software* ou computadores.

- Sequência

A Microsoft (2014), define um diagrama de sequência como um modelo que explicita a sequência de mensagens trocadas entre instâncias de classes, atores, subsistemas ou componentes ao longo do tempo. São aplicados para modelar a interação dos objetos em cenários específicos de um caso de uso, tendo como foco a troca de mensagens (OMG, 2014).

### 3.8 – Metodologia Scrum

Neste subcapítulo, será abordado o ramo do mapa de conhecimento referente à *framework* de desenvolvimento ágil Scrum, que se encontra representado na Figura 15. De Seguida, serão explicados os conceitos associados à metodologia e à sua aplicação.

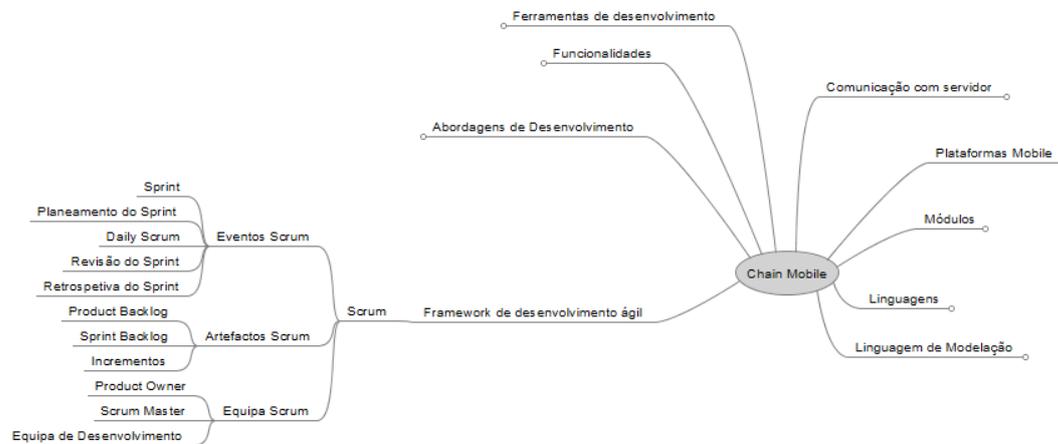


Figura 15 - Mapa de conhecimento - Scrum

O Scrum é uma *framework* ágil vocacionada ao desenvolvimento e manutenção de produtos complexos com o propósito de aumentar o seu valor. Na sua guia Schwaber e Sutherland (2011, p. 3) definem o Scrum como “uma estrutura processual na qual as pessoas podem resolver complexos problemas de adaptação, enquanto que, de forma produtiva e criativa, oferecem produtos de maior valor”. Com uma abordagem iterativa e incremental, visa facilitar todo o processo de planeamento, desenvolvimento e manutenção do produto através de um conjunto de valores, princípios e práticas, para otimizar a previsibilidade e controlo do risco em função

dos objetivos definidos pela organização. Tal como qualquer *framework* de processos empíricos, os seus pilares são (Schwaber & Sutherland, 2013):

- Transparência  
Todos os aspetos relevantes do processo devem ser visíveis pelos responsáveis dos seus resultados. Deve ser adotado um padrão que permita a todos os intervenientes um entendimento comum sobre os aspetos observados.
- Inspeção  
Devem ser analisados com frequência os artefactos Scrum elaborados e o progresso do projeto de forma a detetar variações indesejáveis. Estas vistorias devem ser conduzidas por inspetores qualificados no trabalho a verificar e nunca representar um entrave à execução das tarefas.
- Adaptação  
Quando é detetado um ou mais desvios que comprometem para além do razoável o produto final, levando à sua não-aceitação, o processo ou o objeto de processamento deve ser ajustado. Estes ajustes, a fim de evitar outros desvios, devem ser efetuados com a maior brevidade possível.

### 3.8.1 A Framework

Como em qualquer *framework*, existe uma sequência de processos a realizar desde que um projeto inicia até o seu fim. Na Figura 16, podemos encontrar os conjuntos de atividades que de forma cíclica definem os processos desta *framework*, denominados *Sprints*.



Figura 16 - Processo Scrum retirada de (Scrum Alliance, 2014)

Para uma melhor compreensão e perspetiva da *framework*, as atividades presentes na Figura 16 serão de seguida detalhadas e explicadas (Schwaber & Sutherland, 2011):

### **O Sprint**

O *Sprint* é visto como o recipiente para todos os eventos do *Scrum*, é o coração e a essência do *Scrum*, com um tempo variável de quatro semanas ou menos do qual resulta um “Done”. Um *Sprint* inicia após o término de um outro de forma cíclica. Este, alberga e decompõe-se na Reunião de Planeamento do *Sprint*, em *Daily Scrums*, no trabalho relacionado com o desenvolvimento per si, na Revisão do *Sprint*, e na Retrospetiva de *Sprint*.

No decorrer do *Sprint* a sua meta não pode ser alterada. Isto é, quaisquer que sejam as alterações a serem efetuadas, estas não podem alterar a meta do *Sprint*. Para além da meta, não pode haver substituições, aquisições ou dispensas na equipa de desenvolvimento durante esse período. Tal como os anteriores, os objetivos de qualidade mantêm-se podendo o âmbito do trabalho ser clarificado e renegociado entre o *Product Owner* e a equipa de desenvolvimento ao longo do processo de aprendizagem sobre o conteúdo.

Apenas o *Product Owner* tem o poder de cancelar o *Sprint*. Este evento pode ser cancelado antes do seu término se a sua meta deixar de fazer sentido ou se tornar obsoleto. Diversas razões podem levar ao cancelamento do *Sprint*, nomeadamente a alteração de objetivos, devido a mudanças na empresa, no mercado ou tecnológicas.

O cancelamento de um *Sprint* obriga a uma revisão dos itens que estejam concluídos e considerados *Done*. No caso do trabalho realizado durante o *Sprint* ser potencialmente comercializável, este pode ser aceite pelo *Product Owner*. Se estes se encontrarem incompletos terão de ser reestimados e colocados de volta no *Product Backlog*.

Todo o processo de cancelamento acarreta custos e recursos, pois torna-se necessário reagrupar numa nova Reunião de Planeamento para iniciar um novo *Sprint*. O cancelamento de um *Sprint* deve ser evitado a todo o custo devido ao seu potencial efeito devastador para a equipa de *Scrum*.

### **Product Owner**

O *Product Owner* representa os interesses associados aos *Stakeholders* no projeto. É ele o responsável pela definição dos itens relativos aos requisitos do projeto que irão constituir a lista presente no *Product Backlog*.

## **Product Backlog**

É uma estrutura hierárquica e dinâmica de requisitos, sendo o primeiro da lista, o requisito com maior prioridade e importância. O *Product Owner* é único responsável pela manutenção dessa estrutura identificando e priorizando as alterações, correções, melhorias e/ou funcionalidades idealizadas para serem acrescentadas ao produto.

À medida que um produto é utilizado e ganha cada vez mais valor, ocorre um maior *feedback* do mercado e, por sua vez, o *Product Backlog* torna-se numa lista maior e mais exaustiva. Desta forma, como supramencionado, o *Product Backlog* nunca se encontra completo, encontrando-se em constante mudança de forma a garantir que o produto tem o que é preciso para ser competitivo, apropriado, proveitoso e útil. Coexiste com o produto evoluindo lado a lado com o meio em que estão inseridos, pelo que este se vai desenvolvendo e modificando à medida que se vai conhecendo melhor o produto e os seus utilizadores. O *Product Backlog* é a exclusiva fonte de alterações a serem realizadas no produto contendo estimativas, descrições e uma ordem.

É durante a Reunião de Planeamento de *Sprint* (*Sprint Planning Meeting*), que o *Product Owner* prioriza os itens do *Product Backlog* e explica a sua pertinência à equipa de desenvolvimento. A equipa de desenvolvimento determina e identifica os itens que será capaz de completar durante o *Sprint* que está por começar. Esses itens identificados pela equipa de desenvolvimento como realizáveis no tempo do *sprint* que está por vir são transferidos do *Product Backlog* para o *Sprint Backlog*. Após essa transferência, a equipa de desenvolvimento divide cada item transferido do *Product Backlog* em tarefas de menor dimensão, facilitando a divisão do trabalho entre os membros da equipa.

A preparação do *Product Backlog* consiste em adicionar detalhes, estimativas e ordem aos itens, denominando-se este processo de *Grooming* (preparação).

O *Grooming* é uma atividade em tempo parcial durante um *Sprint* entre o *Product Owner* e a equipa de desenvolvimento. Este é um processo contínuo, onde os itens são todos analisados e revistos, podendo ser atualizados a qualquer momento pelo *Product Owner* ao seu critério. Neste processo, as equipas de desenvolvimento colaboram na elaboração dos detalhes dos itens e são responsáveis por desenvolver todas as estimativas. O *Product Owner* pode influenciar a equipa, ajudando a entender e a escolher os *Trade-offs*, mas são os executores que elaboram a estimativa final.

### **Sprint Planning Meeting**

A *Sprint Planning Meeting* é uma reunião onde estão presentes o *Product Owner*, o *Scrum Master* e a equipa de desenvolvimento. A sua duração deve ser limitada a oito horas quando se trata de um *Sprint* de quatro semanas, sendo proporcional para *Sprints* de menor duração. Esta Reunião de Planeamento de *sprint* pode ser dividida em duas fases, nomeadamente a primeira refletindo o que será entregue como resultado do *Sprint* seguinte e a segunda relativa a como será alcançado o trabalho necessário para obter esse resultado.

Na primeira fase da Reunião de Planeamento do *Sprint*, o *Product Owner* especifica as funcionalidades prioritárias para a equipa de desenvolvimento, ou seja, descreve a forma de como todo o trabalho será realizado e desencadeado, refletindo o *Product Backlog*. A equipa *Scrum* tem como função garantir que as tarefas foram compreendidas na sua totalidade pela equipa de desenvolvimento.

A equipa desenvolvimento prossegue a reunião, planeando e avaliando minuciosamente o trabalho necessário para a execução das tarefas dentro da duração do *Sprint* e procede à seleção dos itens que consegue desenvolver dentro dessa duração. Após esta seleção, a equipa *Scrum* define a meta do *Sprint* que engloba estas tarefas, constituindo o *Sprint Backlog*. Esta corresponde à segunda fase da reunião onde a presença do *Product Owner* é facultativa e onde outras pessoas podem ser convidadas a participar como assessoras técnicas ou conselheiras especializadas.

### **Sprint Backlog**

É constituído pelo conjunto de tarefas selecionadas do *Product Backlog*, que foram selecionadas para o *Sprint* em conjunto com o planeamento da sua execução. Tem como intuito dar uma perspetiva em tempo real do trabalho que se espera conseguir durante o *Sprint*. O *Sprint Backlog* é gerido única e exclusivamente pela equipa de desenvolvimento e encontra-se em constante mutação ao longo do *sprint*. Essa renovação prende-se com o facto da aquisição de maior conhecimento da equipa sobre o trabalho em progresso o que pode gerar novas tarefas, sendo estas adicionadas ao *Sprint Backlog*. Sempre que uma tarefa é concluída o trabalho necessário estimado para as restantes é atualizado.

### **Daily Scrum**

O *Daily Scrum* é um evento onde a equipa de desenvolvimento avalia o trabalho realizado desde a *Daily Scrum* anterior e planeia o trabalho a realizar até à seguinte. Esta reunião tem duração máxima de 15 minutos e ocorre diariamente. A responsabilidade pela aptidão da equipa em

efetuar esta reunião dentro do tempo limite é do *Scrum Master*, que deve orientar e dotar a equipa das boas práticas. O principal objetivo desta reunião é eliminar obstáculos à execução das tarefas, melhorando a comunicação da equipa e com o intuito de fornecer à equipa as *Soft Skills* necessárias para lidar com o imprevisto e atingir a meta do *Sprint*.

### **Revisão do *Sprint***

É uma reunião que ocorre no final de cada *Sprint* com o único propósito de avaliar se a meta do *Sprint* foi atingida. Com duração máxima de quatro horas para *Sprints* de quatro semanas, é uma reunião informal onde o *Product Owner* verifica o que se tornou um *Done* e esclarece o estado do *Product Backlog*. A equipa de desenvolvimento demonstra o trabalho realizado, esclarece alguma dúvida que surja e discute as dificuldades encontradas e resoluções seguidas. Contribui também, como preparação para a reunião de planeamento de *Sprint* seguinte. Após esta reunião, a equipa de desenvolvimento faz uma Retrospectiva do *Sprint* identificando melhorias ao que correu menos bem e estratégias a utilizar futuramente de modo a evitar e solucionar problemas já verificados.

### **Potentially Shippable Product Increment**

Um incremento é o resultado de um *Sprint* onde todas as tarefas planeadas pela equipa desenvolvimento foram terminadas com sucesso, em conjunto com incrementos resultantes de todos os *Sprints* anteriores. Independentemente da decisão do *Product Owner* em o lançar, este deve estar em condições comerciáveis e respeitar a definição de *Done* elaborada pela equipa *Scrum*.

### **Definição de *Done***

A definição de *Done* difere entre as várias equipas *Scrum*, mas a sua definição e significado assumem extrema importância quando o item do *Product Backlog* ou um Incremento é descrito como *Done*.

Apesar da definição *Done* variar, existe uma compreensão consensual e partilhada entre os membros da equipa *Scrum* de forma a promover uma linguagem comum e transparente. Esta definição serve para avaliar quando o trabalho de um incremento de um produto está completo. É essa mesma definição que orienta as equipas de desenvolvimento para saber o número de itens do *Product Backlog* que podem ser selecionados numa Reunião de Planeamento do *Sprint*. Com o aumento da experiência da equipa *Scrum*, a própria definição de *Done* é largamente expandida e garante uma maior qualidade, incluindo critérios mais rigorosos.

## A equipa Scrum

Como este projeto é elaborado por apenas uma pessoa, não é relevante detalhar e definir as tarefas do *Scrum Master* e da equipa de desenvolvimento neste capítulo do projeto. No entanto estas encontram-se detalhadas e definidas no Anexo C – Complemento Scrum.

### 3.9 – Resumo do Capítulo

Quando se começa o desenvolvimento de uma nova solução é importante fazer uma revisão sistemática da literatura teórica e técnica, permitindo efetuar uma pesquisa aprofundada e cuidada sobre os conceitos existentes, tal como das necessidades presentes para que sejam atingidos os objetivos com sucesso. Ao longo deste capítulo, foi possível identificar as abordagens de desenvolvimento existentes e suas características, a diferença entre as plataformas *mobile* Android, IOS e Windows Phone e perceber o conceito de *Web Services* e os tipos existentes. Foi também retratado o UML e seus elementos que serão utilizados neste projeto e especificada metodologia *SCRUM*, que será adaptada e auxiliará o desenvolvimento do produto.

A análise das abordagens de desenvolvimento existentes e da informação que dela resultou serviu de auxílio à empresa na tomada de decisão, clarificando sobre as vantagens e desvantagens associadas às diferentes abordagens. Isto levou a empresa a mudar os seus requisitos e a uma transformação dos objetivos primariamente definidos. Inicialmente, uma única solução seria desenvolvida e suportada nas três plataformas. Após esta análise, a empresa optou pelo desenvolvimento de soluções nativas para as diferentes plataformas.

A alteração relativa à abordagem de desenvolvimento, obrigou a uma reformulação das necessidades e dos objetivos para o mesmo período de tempo. O desenvolvimento nativo exige um *know-how* específico e especializado para cada uma das plataformas. Foi então necessário decidir para qual das três plataformas *mobile* seria iniciado o desenvolvimento, tendo a empresa optado pela plataforma Android, devido à sua popularidade. Esta plataforma utiliza o JAVA como linguagem de programação.

Após definida a abordagem de desenvolvimento a utilizar e a plataforma *mobile* alvo, é necessário perceber como será realizada a comunicação da solução cliente com o servidor. Esta comunicação prende-se com a utilização de *Web Services*, existindo duas alternativas: o SOAP e o RESTful. Apesar dos *Web Services* do estilo RESTful serem suportados nativamente pela Google, fabricante da plataforma Android e do seu SDK, estarem associados a diversas vantagens, a empresa optou pela utilização de *Web Services* do tipo SOAP, o que conferiu um

aumento de dificuldade considerável devido à necessidade de utilização de uma biblioteca criada por terceiros, denominada ksoap2, e à falta de suporte e documentação associada à sua utilização.

Assente tudo isto, subsistia a necessidade de explicar a linguagem de modelação utilizada e a *framework* de desenvolvimento. Como linguagem de modelação será utilizado o UML e elaborados diagramas de casos de uso, atividades e sequência. A *framework* de desenvolvimento consiste numa *framework* ágil denominada Scrum que também foi explicada neste capítulo.

## 4 – Contexto da Empresa e Enquadramento do Produto

Esta secção incide no contexto da empresa e enquadramento do produto. Apesar do grande foco ser o produto e o projeto não ter qualquer intervenção em processos da organização, é importante que seja feita uma breve caracterização da empresa e do sector de atividade, sendo este o primeiro ponto da secção. Como seguimento e segundo ponto, encontra-se a descrição detalhada do produto, das suas características e funcionalidades, tal como a identificação de possíveis produtos concorrentes e suas características. Para finalizar esta quarta secção, será perspetivado o posicionamento do produto no mercado relativamente aos concorrentes identificados.

### 4.1 - Descrição da Empresa

A Doctron, Sistemas de Informação, Lda., é uma empresa sediada em Guimarães, que se encontra no mercado desde 2002. Iniciou a sua atividade desenvolvendo sistemas para gestão documental baseadas em tecnologia IBM. No ano de 2004, desenvolveu um sistema de gestão com vocação à indústria têxtil de vestuário denominado CHAIN (Doctron, Lda, 2014).

### 4.2 - Descrição do Produto

O CHAIN é um sistema de gestão que apresenta uma estrutura muito flexível por forma a adaptar-se, de uma forma simples e rápida aos diversos modelos de negócio do seu mercado alvo. Em conceito, toda a aplicação se resume a três elementos básicos, todos eles fortemente configuráveis:

- Terceiros - Onde são representados Clientes, Fornecedores, Bancos, Armazéns entre outros.
- Artigos - Criação de artigos e produtos com estrutura MRP (*Material Requirement Plan*)
- Documentos - Representa qualquer documento como Compra, Venda, Ordem de Fabrico, Fatura, Guias, e transações mais específicas como Ordem de Tingimento, Não Conformidade, entre outros.

Através de uma configuração (que é sempre específica para cada cliente) destas três estruturas, consegue-se fazer frente a todos os problemas de gestão comercial, contas correntes, gestão de produção e de armazéns dos seus clientes alvo.

Tecnicamente, o CHAIN é desenvolvido em JAVA (J2EE), num modelo de 3 camadas, usando uma base de dados MS SQL e um *Application Server JRun* (em mudança para TomCat). Usa ainda o Jasper para produção de relatórios. Tem como principais pontos fortes:

- Multilingue  
Concebido de raiz para poder disponibilizar os interfaces nas várias línguas europeias. Permite que dois utilizadores estejam a trabalhar sobre os mesmos dados com interfaces em línguas diferentes
- É certificado pela Direção Geral das Contribuições e Impostos (DGCI) que agora se chama Autoridade Tributária e aduaneira (AT)  
Todos os documentos são certificados e garantem as normas definidas pela DGCI, agora AT.
- Utiliza um servidor *Lightweight Directory Access Protocol* (LDAP) para autenticação  
Guarda informação relativa aos vários perfis.
- Possibilita a gestão da relação de documentos  
Todos os documentos são relacionáveis e a relação entre eles pode ser uma simples cópia por explosão de necessidades segundo regras MRP, por satisfação de quantidades ou por liquidação financeira.
- Faz controlo da versão de documentos  
Regista todas as alterações realizadas nos documentos, quando estas são possíveis, criando sempre uma nova versão com a informação atualizada.
- Admite anexos em documentos e artigos  
Permite anexar qualquer informação que o utilizador pretenda em documentos e artigos, como por exemplo: afoto do produto ou a digitalização da fatura.. Na Figura 17, podemos verificar os campos da criação de um documento, com a possibilidade de adicionar mais do que um anexo.

**Alterar Documento**

Detalhe do Documento  
 Modificar os campos - Este documento já tem 3 reproduzido(s). Ver. 1

**Documento**

Tipo: Compras de Tecidos e Malhas Id: 120044  
 Série: CTC Data: 2012-01-27  
 An. 1:    
 An. 2:

**Terceiro**

Fornecedores: INDITEX SA  
 Documento: 1724048 (B206)

**Dados Entrega**

Data: 2012-01-27  
 Moradas: AVDA DE LA DIPUTACION

**Dados Financeiros**

Cond. Pgt: Pagamento 90 dias Agente: Moeda: EUR Câmbio: 1.0 Incoterm: CF

**Linhas do Documento**

Modelo	Descrição	Cor	Lote	Quantidade	Preço Unitário	Total Líquido	Data Entrega	Ação
001964	TEC. SONNY F.558	P611		2500.000	8.500	16250.000	2012-01-23	<input type="checkbox"/>

Figura 17 - Exemplo de um documento (CHAIN)

- Permite gestão de stocks  
Permite a gestão de modo fácil e em tempo real de *stocks*.
- Possibilita manter a cadeia operacional informada das alterações (Workflow)  
Sempre que ocorre alguma alteração, os elementos abrangidos por essa alteração são notificados.
- Controlo de qualidade  
Do suporte ao controlo de qualidade, oferecendo informações sobre padrões de qualidade, registando o grau do produto inserido.
- Contas correntes  
Permite uma gestão simples e fácil de contas correntes.
- 100% Web Based  
Compatível com vários *Browsers* (Internet Explorer, Firefox, Safari, Chrome), tirando partido de tecnologias como JSP (Java Server Pages) e Web 2.0, minimizando a largura de banda necessária, viabilizando a utilização do sistema com a infraestrutura comum na internet. Não necessita de instalações adicionais para ser utilizado.
- Saber da rentabilidade de um negócio  
Permite, como podemos verificar na Figura 18, calcular a rentabilidade de, por exemplo, uma encomenda através de cálculos de subtração dos gastos na matéria-prima e outras despesas ao valor recebido pelo produto final.

Documento	Quantidade	Clide Satisfeita	Valor	Agregado	V Unit.
ENC 110580 / 1 E LTD	100.0	100.0	0.00	160.69	1.61
PLMAT 110828 / 2 PMAT LTD	12.36	11.0	0.00	202.35	16.66
COTECENC 110559 / 0 CTC 6 DIAS - TEXTEIS INTERNACIONAIS, UNIPESSOAL, LDA	11.0	96.0	0.00	202.35	18.46
RECOTECENC 110656 / 2 RTC 6 DIAS - TEXTEIS INTERNACIONAIS, UNIPESSOAL, LDA	96.0	0.0	0.00	0.00	0.00
FACFOR 112025 / 2 A 6 DIAS - TEXTEIS INTERNACIONAIS, UNIPESSOAL, LDA	96.0	0.0	202.35	202.35	2.16
PLMAT 110832 / 1 PMAT LTD	76.0	76.0	0.00	0.00	0.00
PLMAT 110914 / 1 PMAT LTD	116.04	116.0	0.00	14.73	0.13
OK 120002 / 1 OK	100.0	0.0	0.00	573.00	5.73
FACTLI 120059 / 0 F	100.0	0.0	0.00	1237.66	12.38
FACTLI 120059 / 0 F	100.0	0.0	1237.66	1237.66	12.38

Expandir Todos | Fechar Todos

Figura 18 - Controlo de Custos (CHAIN)

- Informa quando há problemas (Management by Exception)  
Possibilita a configuração de sensores de alerta quando detetam situações anómalas. São exemplos disso, atrasos nos prazos, desvio dos valores medidos no processo de controlo de qualidade, quantidades recebidas não correspondentes às encomendadas, entre outros personalizáveis.
- Permite saber sempre onde está, o que está feito, o que falta fazer

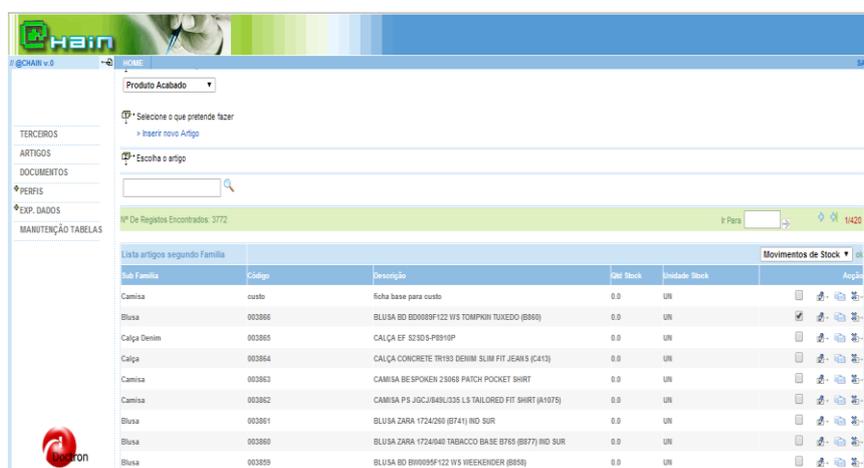
Permite a qualquer altura que o utilizador obtenha informações relativamente aos processos da empresa.

- Totalmente adaptável a diferentes formas de trabalhar

Permite uma personalização quase completa para o cliente de acordo com as suas necessidades em tempo real.

- Manter a facilidade de utilização e simplicidade da interface gráfica

Sempre com a mesma estrutura independentemente das necessidades do cliente. As várias famílias de artigos, os diversos grupos de fornecedores e os diversos tipos de documento partilham dos mesmos interfaces. Na Figura 19, verificamos o interface dos artigos e a forma como eles se apresentam.



The screenshot displays the CHAIN software interface. At the top, there is a navigation bar with the 'HOME' button. Below it, a dropdown menu for 'Produto Acabado' is visible. The main content area features a search bar and a table titled 'Lista artigos segundo Família'. The table has columns for 'Sub Família', 'Código', 'Descrição', 'Cód. Stock', 'Unidade Stock', and 'Ações'. The table lists several items, including 'Blusa', 'Camisa', and 'Calça Denim', with their respective codes and descriptions. A sidebar on the left contains navigation options: 'TERCEIROS', 'ARTIGOS', 'DOCUMENTOS', 'PERFIS', 'EXP. DADOS', and 'MANUTENÇÃO TABELAS'. The bottom left corner shows a red circular logo with the number '10'.

Sub Família	Código	Descrição	Cód. Stock	Unidade Stock	Ações
Blusa	custo	folha base para custo	0.0	UN	[Icons]
Blusa	003866	BLUSA BD BD0089F122 W/S TOMPKIN TUXEDO (B880)	0.0	UN	[Icons]
Calça Denim	003865	CALÇA EF 523D5-P8918P	0.0	UN	[Icons]
Calça	003864	CALÇA CONCRETE TR193 DENIM SLIM FIT JEANS (C413)	0.0	UN	[Icons]
Camisa	003863	CAMISA BE SPOKER 25068 PATCH POCKET SHIRT	0.0	UN	[Icons]
Camisa	003862	CAMISA P5 JGCJ84BL335 L/S TAILORED FIT SHIRT (A1075)	0.0	UN	[Icons]
Blusa	003861	BLUSA ZARA 17240268 (B741) MID SUR	0.0	UN	[Icons]
Blusa	003860	BLUSA ZARA 17240400 TABACCO BASE 0765 (B877) MID SUR	0.0	UN	[Icons]
Blusa	003859	BLUSA BD BW0095F122 W/S WEEKENDER (B850)	0.0	UN	[Icons]

Figura 19 - Interface dos Artigos (CHAIN)

Estes artigos podem ser agrupados por famílias, facilitando a procura e mantendo a organização. Na Figura 20, verifica-se o procedimento de seleção de uma família de artigos.

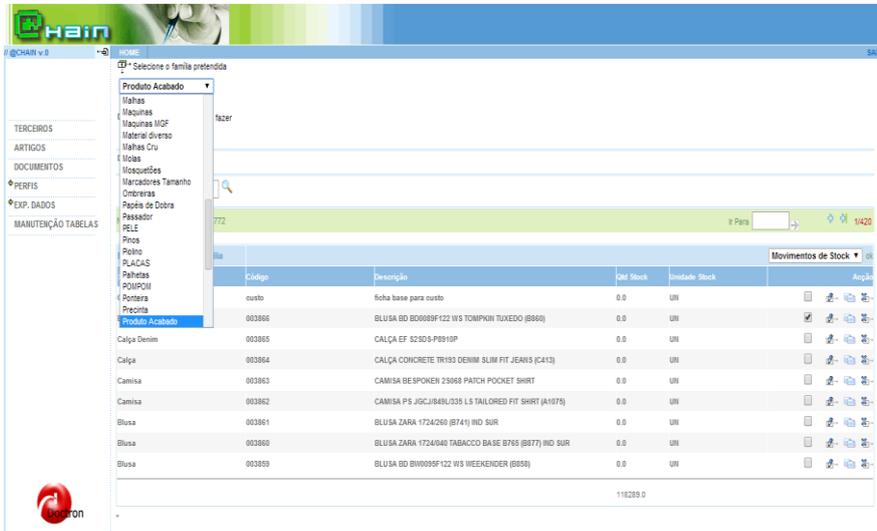


Figura 20 - Interface de Seleção de Família (CHAIN)

A nível da administração, é muito fácil configurar as necessidades do cliente. Enquanto administrador, é possível aceder à opção manutenção de tabelas, permitindo gerir e personalizar, de forma rápida e fácil, tudo o que o cliente vai ter ao seu dispor. Na Figura 21, encontra-se a interface de manutenção de tabelas, assim como o método de seleção da tabela que queremos personalizar, disponibilizando-a de acordo com as necessidades e pretensões do cliente.

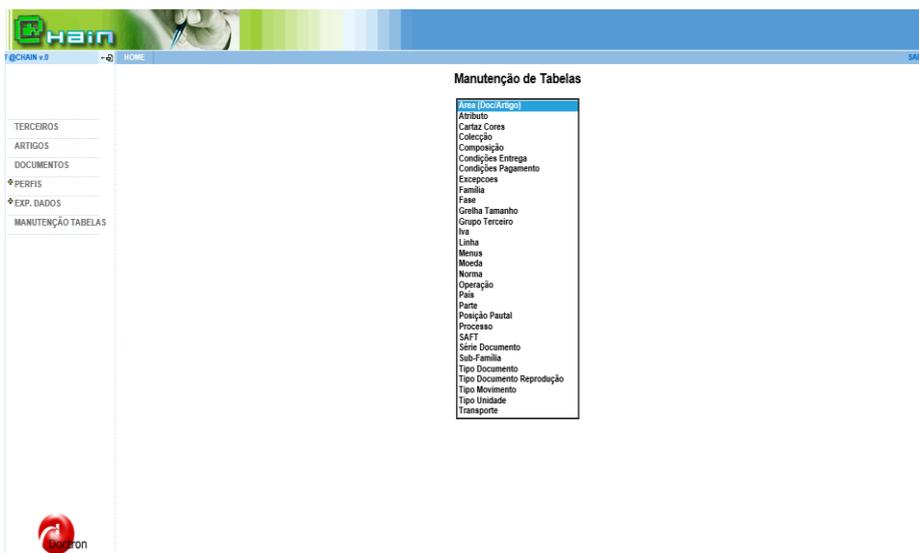


Figura 21 - Manutenção de Tabelas (CHAIN)

#### 4.2.1 – Análise da Concorrência

Neste capítulo, será levantado o estado da arte referente a soluções concorrentes ao CHAIN. Em complemento será efetuado, para cada produto concorrente, um levantamento dos serviços no que refere a aplicações *mobile*, qual a empresa que oferece esse produto, com o intuito de perceber como elas se posicionam a nível de soluções para dispositivos *mobile*.

A Doctron, solicitou que esse estudo se focasse em sete empresas com as seguintes características: empresas consideradas concorrentes que atuem em Portugal e tenham grande foco no setor da indústria têxtil e vestuário. Essas empresas são: Primavera BSS, PHC, Macwin, Vanguarda, Infos, Hydra IT e F3M. Todas as informações que se seguem foram retiradas da página *Web* das respetivas empresas.

##### 4.2.1 – PRIMAVERA BSS

A PRIMAVERA Business Software Solutions é uma empresa Portuguesa fundada em 1993 estando a sua sede situada em Braga. Apesar de oferecer soluções para outros setores, desenvolveu e comercializa, tal como a Doctron, um produto com grande foco no setor têxtil chamado PRIMAVERA Manufacturing (PRIMAVERA BSS, 2014).

No âmbito de extensões *mobile*, a PRIMAVERA Business Software Solutions comercializa dois produtos, nomeadamente:

- ELEVATION MOBILE (PRIMAVERA BSS, 2014)

A ELEVATION MOBILE é uma aplicação na *cloud*, que permite ao cliente aceder a determinadas informações estatísticas e de análise no âmbito da tomada de decisão. A aplicação é suportada pelo sistema Android, IOS e Windows Phone, fazendo integração com o ERP da PRIMAVERA BSS. Permite a exploração de informação nos módulos Vendas e Aprovações (PRIMAVERA BSS, 2014).

- PRIMAVERA MOBILE BUSINESS (PRIMAVERA BSS, 2014)

É um produto especializado da PRIMAVERA BSS, o único encontrado capaz de criar diversos tipos de documento, nomeadamente: Fatura, Orçamento, Recebimento, Encomenda entre outros. Estes documentos criados, são integrados diretamente no ERP PRIMAVERA sem necessidade de qualquer intervenção humana extra. Tanto o sistema no terminal fixo como no dispositivo *mobile* estão em constante sincronização, enquanto for garantida a ligação de dados. Tem como limitações:

A nível dos dispositivos móveis (PRIMAVERA BSS, 2014):

- Subscrição de um serviço móvel de dados 3G/GPRS;
- Windows Phone a partir da versão 5 e uma resolução de ecrã de 320x480;

A nível do sistema Servidor (PRIMAVERA BSS, 2014):

- Windows Server 2003/2008, Windows XP Pro, Windows Vista (exceto Home Edition);
  - SQL Server 2005/2008;
- O servidor ter permissões para ser acedido do exterior;

#### 4.2.2 - PHC

A PHC é uma empresa que se dedica à comercialização e desenvolvimento de sistemas de gestão. Fundada em 1989, oferece soluções para vários setores em Portugal, abrangendo a indústria têxtil e concorrendo diretamente com a Doctron através da solução PHC Manufactor CS (PHC, 2014).

Não existe qualquer referência a oferta de soluções *mobile* por parte da PHC.

#### 4.2.3 - MacWin

A MacWin é uma empresa que oferece soluções de gestão maioritariamente presente na indústria têxtil e vestuário. Fundada em 1998, assume que é a esse sector que se dedica diariamente no sentido de dar melhor resposta às novas exigências diárias (MacWin, 2014).

Relativamente à comercialização de aplicações *mobile*, a MacWin não faz referência a qualquer oferta.

#### 4.2.4 - Vanguarda

A Vanguarda é uma empresa que oferece soluções integradas para diversos setores, entre os quais, o setor da industria têxtil. Dedicar-se ao desenvolvimento e comercialização de soluções de gestão personalizadas, apelidando as suas funções como eficazes, eficientes e de excelência (Vanguarda, 2014).

No âmbito da oferta de soluções *mobile*, a Vanguarda não faz referência a qualquer solução.

#### 4.2.5 - Infos

A Infos é uma empresa sediada em Leça da Palmeira que desde a sua formação, 1990, deu grande foco ao setor têxtil. Concorre nessa área com a Doctron através do produto INFOS ITV. Sendo esta uma das soluções que a Infos oferece, entre outras soluções integradas de gestão e consultoria (INFOS, 2014).

Como soluções *mobile*, a Infos apresenta a solução X-Mobile. Esta plataforma é exclusiva Windows Phone, funciona mais como uma *framework* para aplicações específicas. Relativamente à aplicação no Windows Phone, ela restringe-se, tal como as outras aplicações *mobile* encontradas, a consultas, nomeadamente: vendas, encomendas, clientes, clientes novos, guia de circulação, entre outras funcionalidades, requerendo algumas delas, *hardware* próprio da Infos (INFOS, 2014).

#### 4.2.6 - Hydra IT

A Hydra IT é uma empresa da área das tecnologias de informação, atuando no mercado português e oferece soluções integradas de TI baseadas em Microsoft Dynamics Nav e CRM. Apresenta soluções para vários setores, nomeadamente para a indústria têxtil (Hydra IT, 2014).

A nível de oferta de soluções *mobile*, a Hydra IT não refere qualquer oferta.

#### 4.2.7 - F3M

A F3M é uma empresa Portuguesa, fundada em 1987, especializada em tecnologias da informação e comunicação. É uma empresa que opera no setor têxtil, com ofertas de conceção, produção e implementação de *software* de gestão e consultoria (F3M, 2014).

A F3M oferece a nível *mobile*, uma aplicação chamada F3M Mobile. Tal como as soluções *mobile* estudadas anteriormente. Esta disponibiliza estatísticas e análises, limitando o cliente à exploração de informação. Ao contrário das anteriores, não há nenhuma indicação sobre a plataforma de suporte em que esta opera nem com que ERP integra (F3M, 2014).

#### 4.2.8 – Resumo da Concorrência

Das soluções analisadas, todas se baseiam na exploração de informação. As soluções da PRIMAVERA BSS são suportadas por três plataformas, nomeadamente: Windows Phone, Android e IOS. Enquanto a solução da Infos, esta encontra-se circunscrita a uma plataforma exclusiva (Windows Phone) e funciona mais como *framework* para aplicações específicas. Relativamente à X-MOBILE, não foram encontradas referências sobre as plataformas que suportam a solução da F3M. Na Tabela 7, encontra-se o resumo da análise efetuada à concorrência, identificando a solução *mobile*, caso disponível, o seu nome, ERP e plataformas suportadas para cada empresa.

Tabela 7 - Soluções MOBILE por Empresa

Empresa	Soluções Mobile	Solução	Sistema Operativo Suportado	ERP Suportado
F3M	Sim	F3M MOBILE	-	-
Hydra IT	Não	-	-	-
Infos	Sim	X MOBILE	Windows Phone	Multi; Dynamics Nav; PHC; Sage Gestexper; Primavera; Artsoft; 2-Soft; Vanguarda Software; Baan
Macwin	Não	-	-	-
PHC	Não	-	-	-
Primavera BSS	Sim	Elevation Mobile;	IOS; Android; Windows Phone	Primavera
		Primavera Mobile Business	Windows Phone	Primavera
Vanguarda	Não	-	-	-

O intuito deste projeto é identificar as funcionalidades do CHAIN que podem ser estendidas para dispositivos *mobile*, não apenas com a finalidade de explorar informação, como é oferecido por todas as soluções encontradas à exceção do PRIMAVERA Mobile Business, que apesar das suas limitações, nomeadamente a nível da plataforma de suporte, terminais suportados e resolução do ecrã, é a única que, para além de permitir exploração de informação, também possibilita a criação de documentos a partir de um dispositivo *mobile*, fazendo a integração com o ERP PRIMAVERA.

Este é um projeto um pouco mais ambicioso no sentido de dar ao utilizador uma efetiva alternativa ao interface tradicional no computador, tal como já faz o PRIMAVERA Mobile Business através da criação de uma aplicação que permita explorar informação e criar documentos, que seja integrável com o ERP CHAIN. Numa fase inicial, pretendeu-se num único desenvolvimento cobrir as plataformas Android, Windows Phone e IOS. Num estudo mais aprofundado, este tipo de desenvolvimento que recorre a abordagens multiplataforma, como se pode constatar no subcapítulo 3.2 Abordagem de Desenvolvimento, apresenta resultados mais limitados tendo a empresa optado por um desenvolvimento nativo que inicialmente incidirá na plataforma Android.

#### 4.2.2 – Posicionamento do Produto

De maneira a avaliar o impacto que a solução terá no posicionamento do produto, é importante perceber alguns pontos, nomeadamente a sua origem, o *know-how* da equipa envolvida no seu desenvolvimento e como foi concebido:

- Origem do Produto

À semelhança do que aconteceu com outras empresas, agora concorrentes, que nasceram e prosperaram aproveitando o *momentum* de uma nova plataforma de desenvolvimento, como aconteceu com a passagem para ambientes gráficos, a Doctron iniciou o processo de desenvolvimento do CHAIN numa altura em que o desenvolvimento aplicativo inteiramente na *web* estava imaturo.

- Know-How da equipa de desenvolvimento

A equipa envolvida neste desenvolvimento, era constituída por elementos com larga experiência em ERPs, particularmente dedicados à indústria têxtil e de vestuário, conhecedores das aplicações de referência a nível mundial nesse tempo, como PORINI, DATATEX, LINX. Possuíam ainda experiência em implementações nacionais como INFOS e inúmeros desenvolvimentos à medida que marcaram uma época.

- Desenvolvimento Online

Um cliente da Doctron abriu as suas portas para que, à medida que cada funcionalidade estivesse desenvolvida, pudesse ser implementada e experimentada em produção, dando uma robustez de análise muito grande.

Como resultado, o CHAIN é uma ferramenta muito ágil para consultores na área têxtil, que pretendam fazer implementações em tempo *record*. Concorrentes nacionais como PHC ou Primavera e internacionais como MOVEX, desde cedo tinham bons produtos na área de produção e durante anos tentaram, a muito custo, implementá-los na indústria têxtil. Apenas num passado recente, estas empresas começaram a aculturar-se às especificidades deste tipo de mercado.

Os clientes CHAIN sempre souberam das vantagens da utilização de uma aplicação na *cloud*. O desenvolvimento de um interface *mobile* não irá reposicionar o produto, mas oferecer facilidades adicionais ao seu mercado, mantendo-se competitivo.

A Informação comparativa presente em baixo foi obtida nos vários negócios em que o CHAIN ganhou e perdeu para a concorrência:

- Penetração no mercado  
A penetração do CHAIN é baixa devido à concorrência já existente e da numerosa oferta especializada e não especializada no mercado.
- Preço produto  
O preço do produto é baixo. O fato de ser suportado por algumas ferramentas *Open Source* possibilita a prática de um preço reduzido.
- Preço de implementação  
O preço de implementação é médio. Grande parte desse preço prende-se fundamentalmente com o custo da consultoria.
- Tempo de implementação  
A implementação é muito rápida e a formação necessária é muito breve.
- Débil em termos de exploração de informação (*Reports e Dashboards*).  
A ausência de *Dashboards* é atualmente uma das suas principais fraquezas. Relativamente aos *Reports* quase que se pode dizer que são feitos à medida para cada cliente.
- Vocação para a *cloud*  
Devido ao facto do CHAIN ser totalmente baseado na *web* e sem qualquer concessão, tem total vocação para a *cloud*.
- Vocação para o mercado têxtil e de vestuário  
O CHAIN é um produto especializado no mercado têxtil e de vestuário cobrindo toda a sua atividade. O que não acontece com outros produtos, que apesar de terem *Add-ons* para o setor têxtil, não cobrem a sua atividade na totalidade.

Segue-se agora uma análise do produto numa perspetiva de Marketing. Na Figura 22, encontra-se a representação do processo de Marketing.

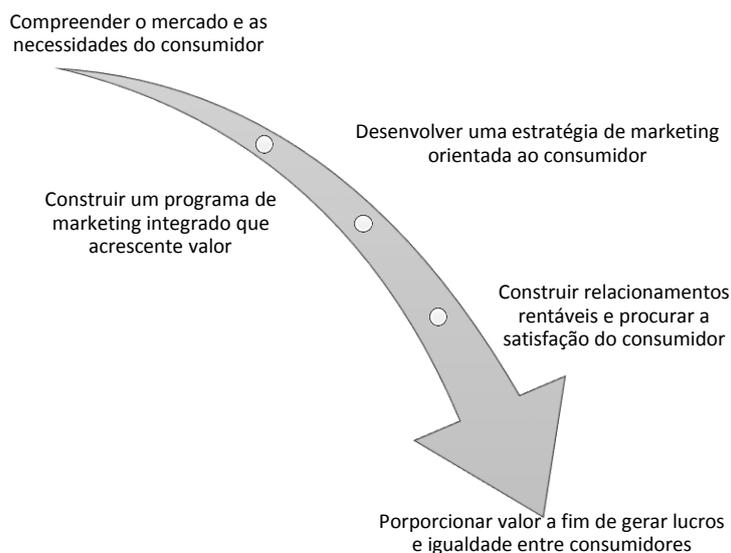


Figura 22 - Processo de Marketing adaptado de (Kotler & Armstrong , 2013)

É importante abordar este processo de forma a compreender a opção da Doctron pelo desenvolvimento deste produto. A Doctron, empresa que requisitou o desenvolvimento deste produto, é uma PME de tecnologias de informação com recursos limitados. Uma boa promoção do produto e distribuição potencializam novos consumidores, contudo estes canais não serão afetados pelo desenvolvimento do novo produto que consiste na extensão de funcionalidades de um produto já existente.

Para compreender o processo de comercialização deste produto é importante analisar a perspectiva do vendedor (Doctron) e do consumidor (potenciais clientes). Na Tabela 8, estão relacionadas essas duas perspectivas.

Tabela 8 - Perspetivas vendedor e consumidor adaptado de (Kotler & Armstrong , 2013)

Perspetiva do Vendedor (4 Ps)	Perspetiva do Consumidor (4 Cs)
Product (Produto)	Customer Solution (Solução para a necessidade do consumidor)
Price (Preço)	Customer Cost (Custo para o consumidor)
Place (Distribuição)	Convenience (Conveniência)
Promotion (Promoção)	Communication (Comunicação)

A perspetiva do vendedor em Marketing constitui o *Marketing Mix*. Kotler e Armstrong (2013, p. 76) definem o *Marketing Mix* como “tudo aquilo que a empresa pode fazer para influenciar a

procura pelo seu produto”. Para tal, uma boa análise sobre a perspetiva do consumidor é necessária. Na perspetiva dos “4 Ps”, estes referem-se aos termos ingleses de *Product*, *Price*, *Place* e *Promotion*. e na perspetiva dos “4 Cs” estes identificam o *Customer Solution*, *Customer Cost*, *Convenience* e *Communication* (Kotler & Armstrong , 2013).

Quando uma empresa desenvolve um produto, este tem como finalidade dar uma resposta a uma necessidade do consumidor, apresentando uma solução a que o consumidor atribuirá valor. O preço do produto também deve ser realista e variar em função do valor que o consumidor lhe dá e que está disposto a pagar. A conveniência pode gerar um novo consumidor. Uma boa cadeia de distribuição é fundamental para garantir essa conveniência. Por fim, relativamente à promoção, a empresa deve definir e utilizar estratégias que cativem o consumidor e o levem a comprar o produto. Assim sendo, a maneira como a informação do produto chega ao consumidor é fundamental.

A Doctron relativamente ao *Marketing Mix* apenas elaborou estratégias em função da política de produto. O CHAIN é um produto de especialidade para o setor têxtil e de vestuário que será alvo de extensão de algumas das suas funcionalidades para plataformas *mobile*. Este manterá o mesmo estatuto com a vantagem de ter as suas funcionalidades estendidas e otimizadas para um maior número de plataformas. A empresa procura manter e cativar potenciais clientes através da inovação tecnológica (extensão do produto para dispositivos *mobile*), acompanhando a concorrência, da especialização (focado no setor têxtil e vestuário) e qualidade superior (desenvolvido nativamente para cada plataforma) sem alterar outras políticas não relacionadas com o produto (Kotler & Armstrong , 2013). Não terá impacto no preço na perspetiva da empresa, na cadeia de distribuição, nem na política de comunicação. Na perspetiva do consumidor gera aumento de valor perante o aumento da oferta sem contrapartidas.

### 4.3 – Resumo do Capítulo

Ao longo deste capítulo foi dado grande foco ao produto devido ao facto de este ser o alvo da intervenção e não a empresa. Inicia com uma breve descrição da empresa e identificação do seu setor de atividade. Prossegue com a descrição detalhada do CHAIN, que é o produto para o qual vão ser estendidas as funcionalidades *mobile*, especificando as suas características e funcionalidades. É também feita uma análise à concorrência, no âmbito de encontrar produtos potencialmente concorrentes à solução a desenvolver, procedendo ao estudo das funcionalidades, que outras empresas concorrentes à Doctron já lhes embutiram e estenderam dos seus respetivos ERP’s. O capítulo termina com o impacto que a solução *mobile* terá no

produto CHAIN a nível de posicionamento de mercado, iniciando com uma breve história de como este foi desenvolvido e do *know-how* da equipa desenvolvimento.

## 5 – Desenvolvimento do Produto

Esta quinta secção projeta dois grandes conceitos: o pensar e o fazer. Para garantir o cumprimento dos requisitos do cliente podem ser utilizadas metodologias de auxílio à conceção de produtos. O primeiro ponto refletirá o pensar, onde através de modelação UML e da definição do interface gráfico do utilizador é possível planear ao detalhe a solução a desenvolver em conjunto com a documentação técnica necessária. Relativamente ao fazer, segundo e último ponto desta secção, serão adaptados e aplicados os conceitos e boas práticas evidenciadas numa *framework* ágil de desenvolvimento, Scrum, garantindo que a solução desenvolvida corresponde à solução pensada e requerida pelo cliente.

### 5.1 Conceção do Produto

É neste subcapítulo que se projeta o conceito pensar, estando decomposto na modelação UML da solução CHAIN Mobile e na definição da sua GUI e documentação necessária à sua utilização.

#### 5.1.1 Modelação UML

Aqui encontra-se a modelação elaborada utilizando UML, cingindo-se ao sistema de casos de uso do sistema CHAIN Mobile e do subsistema manusear documentos, abarcando os seus diagramas de atividade e de sequência para os diferentes cenários. Todos os restantes subsistemas podem ser encontrados detalhados e com os respetivos diagramas no Anexo A – Modelação UML.

##### 5.1.1.1 {U.0} Sistema CHAIN Mobile

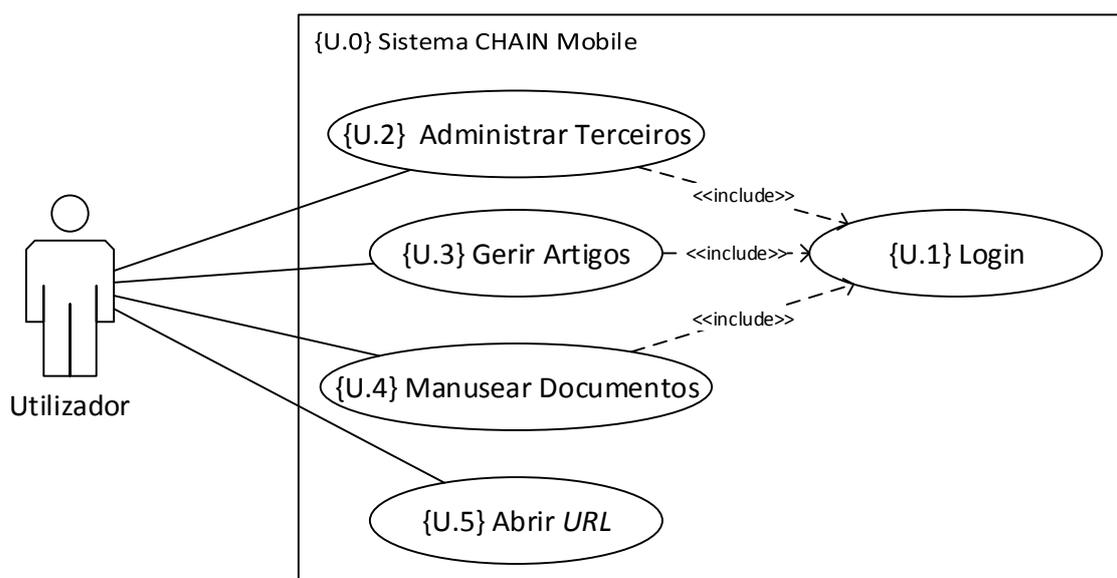


Figura 23 - Sistema CHAIN Mobile (UML-UC)

## Descrição

O sistema do CHAIN Mobile, tal como o CHAIN Web, será constituído por quatro módulos com funcionalidades distintas. O único ator do sistema é o utilizador da aplicação *mobile* que apenas terá acesso às funcionalidades facultadas por cada módulo se este efetuar com sucesso o login no CHAIN Web através da aplicação *mobile*.

### 5.1.1.5 {U.4} Manuseamento de Documentos

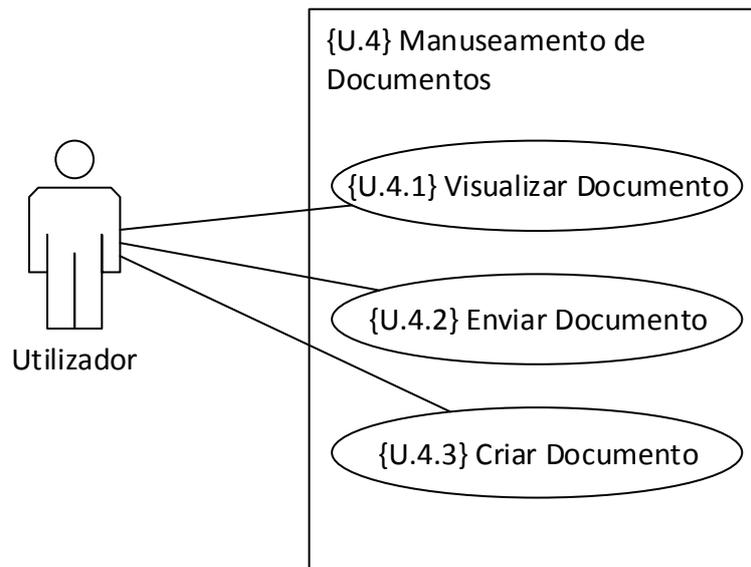


Figura 24 - Subsistema de Manuseamento de Documentos (UML-UC)

## Descrição

O subsistema de manuseamento de documentos possibilita ao utilizador visualizar qualquer documento que já se encontre no sistema, anexar documentos existentes e adicionar um novo documento.

## Pré Condições

- Login do utilizador efetuado com sucesso;
- Ter sido selecionado subsistema de Manuseamento de Documentos;

### {U4.1} Visualizar Documento

- Fluxo de eventos:
  1. O utilizador acede ao menu principal;
  2. O botão Documentos é selecionado;
  3. Existe ligação à internet;
  4. A lista de documentos é disponibilizada ao utilizador;

- Fluxo alternativo:
  1. Ligação à internet indisponível;
    - 1.1. Notificação “Sem acesso à internet, verifique a sua ligação”;
    - 1.2. Retorna ao menu principal;
- Diagrama de atividade:

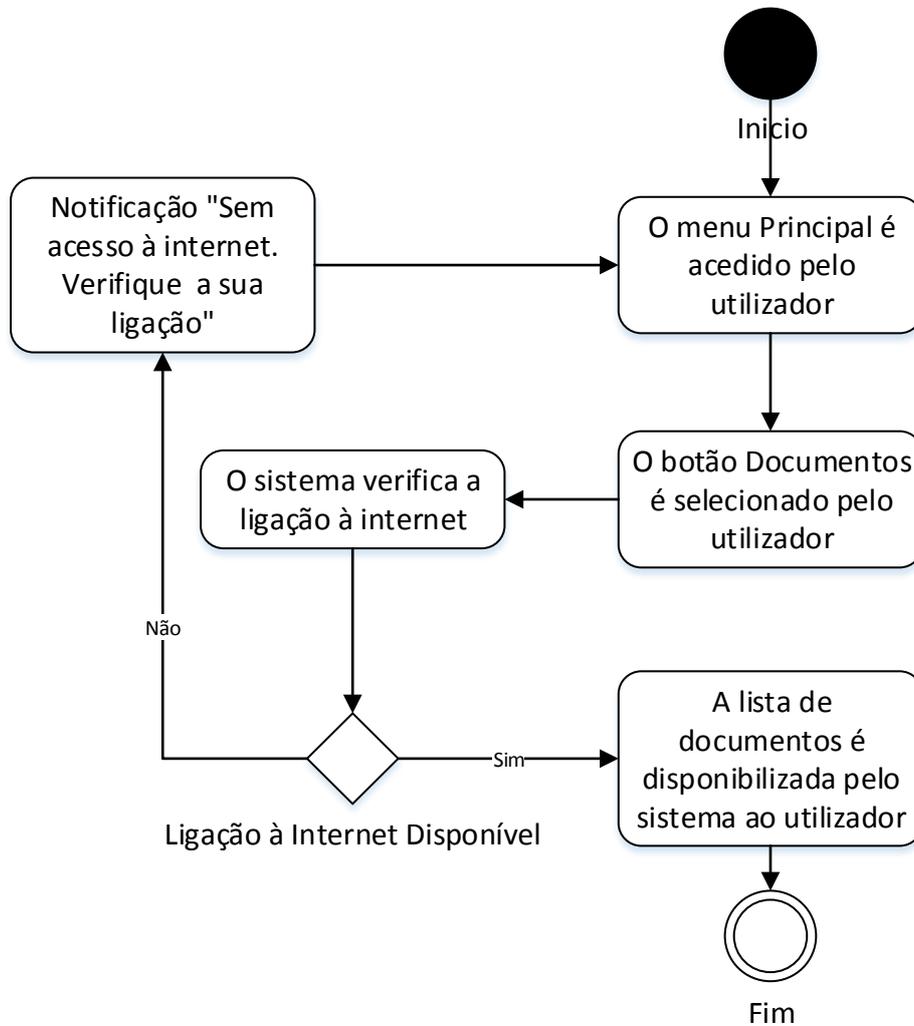


Figura 25 - Visualizar Documento (UML-DA)

- Diagramas de sequência:

1. **Cenário:** O utilizador vê os documentos com sucesso.

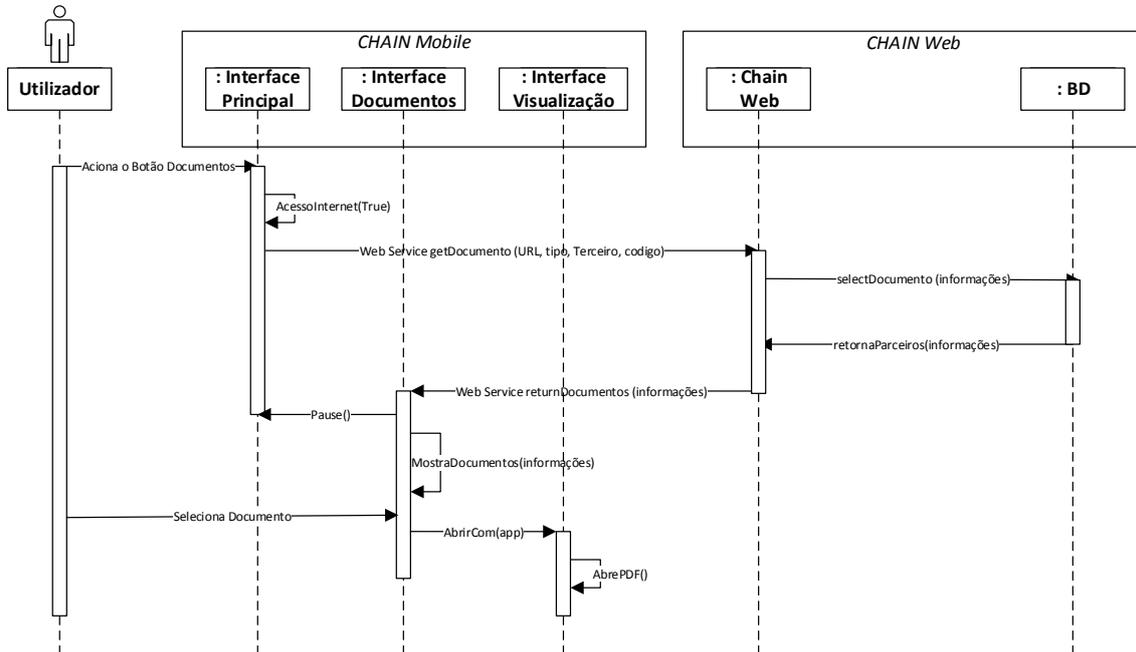


Figura 26 - O utilizador consegue visualizar os documentos com sucesso (UML-DS)

2. **Cenário:** O utilizador não consegue visualizar os documentos por falta de acesso à internet.

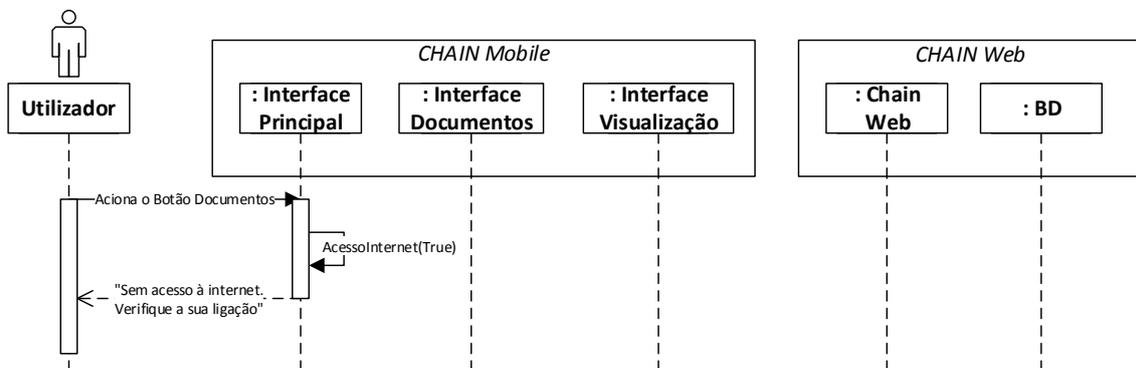


Figura 27 - O utilizador não consegue visualiza os documentos por falta de acesso à internet (UML-DS)

#### {U4.2} Enviar Documento

- Pré-Condição:

- O utilizador ter acesso à lista de documentos encontrando-se nesse menu;

- Fluxo de eventos:

1. A secção de enviar documento no documento é seleccionada;
2. O utilizador escolhe a aplicação que quer utilizar para enviar o documento;

3. A aplicação escolhida abre com o documento como anexo;

- Diagrama de atividade:

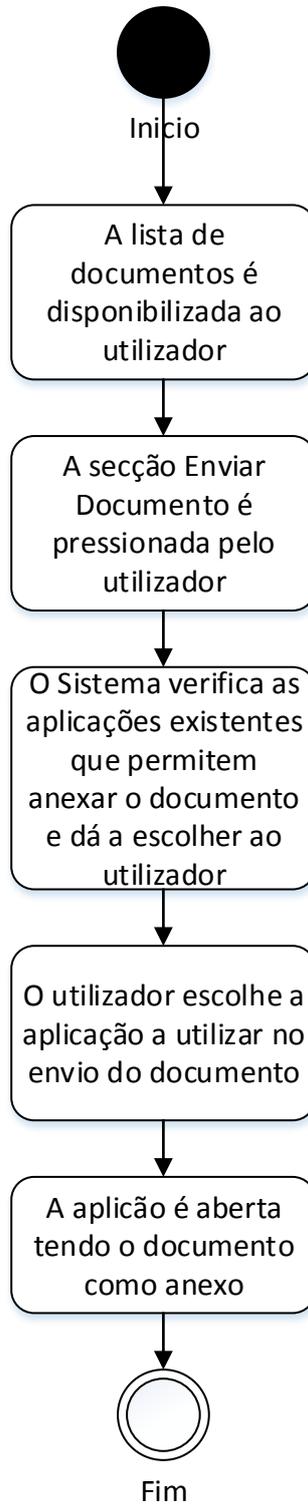


Figura 28 - Enviar Documento (UML-DA)

- Diagramas de sequência:

1. **Cenário:** O utilizador anexa um documento com sucesso.

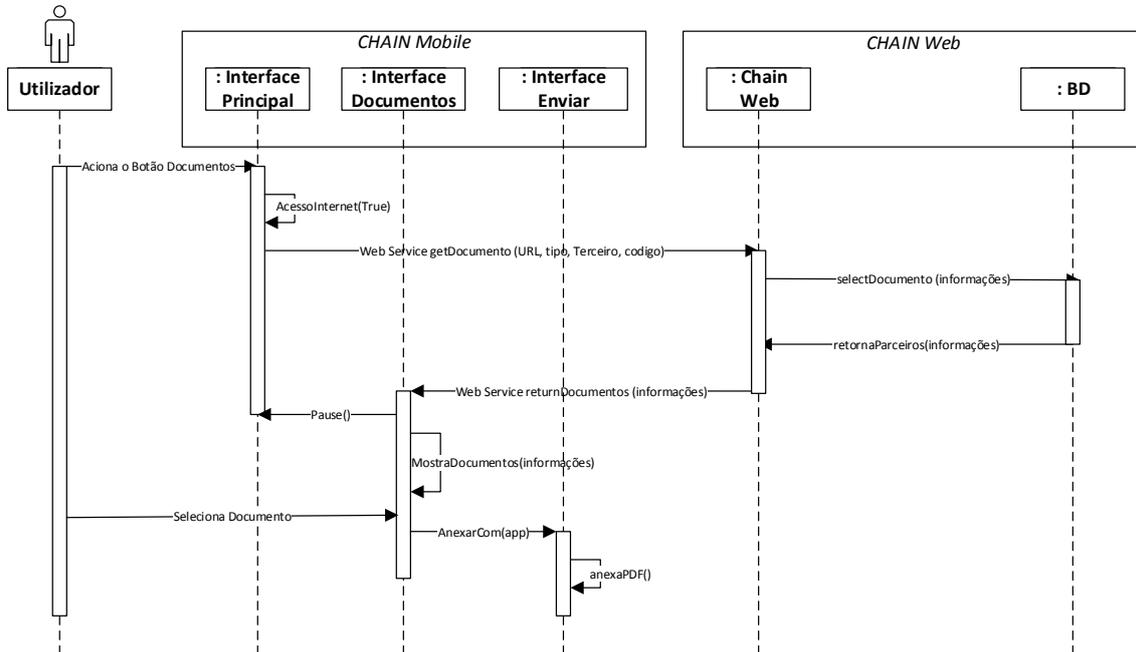


Figura 29 - O utilizador anexa um documento com sucesso (UML-DS)

2. **Cenário:** O utilizador não tem acesso à internet para anexar um documento.

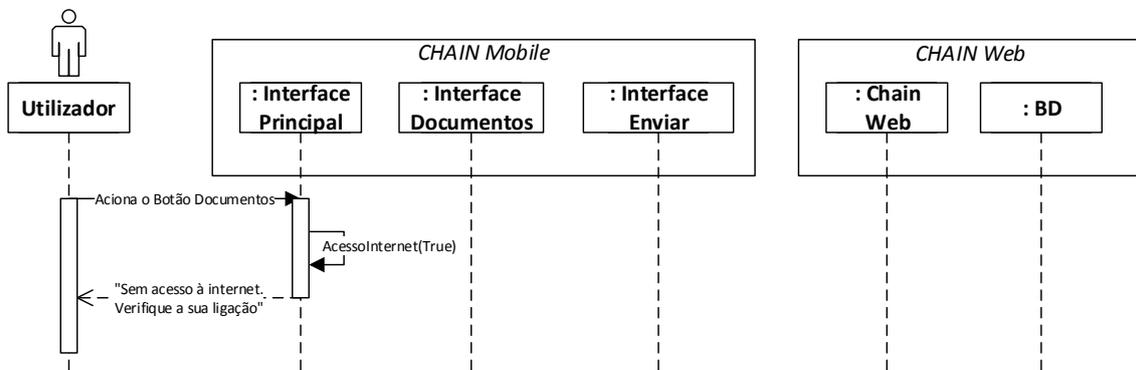


Figura 30 - O utilizador não consegue anexar um documento por falta de internet (UML-DS)

### {U4.3} Criar Documento

- Fluxo de eventos:

1. O menu Principal é acedido pelo utilizador;
2. O utilizador aciona o botão documentos;
3. Existe ligação à internet;
4. A lista de documentos é disponibilizada ao utilizador;
5. A secção de adicionar documento é seleccionada;

6. O sistema disponibiliza os tipos de documento e os terceiros disponíveis;
  7. O utilizador seleciona o tipo de documento e o terceiro;
  8. O utilizador adiciona observações ao documento;
  9. O utilizador seleciona o *tab* "Linha de Artigos";
  10. O utilizador carrega no *botão* adicionar artigos;
  11. O sistema verifica os artigos disponíveis;
  12. O utilizador seleciona o artigo pretendido;
  13. O utilizador adiciona a informação do artigo que pretende associar ao documento;
  14. O utilizador carrega no botão adicionar artigo;
  15. Os campos foram todos preenchidos;
  16. Notificação "o artigo foi inserido com sucesso";
  17. O utilizador carrega no *tab* "Linha de Artigos";
  18. O utilizador carrega no botão "Adicionar documento";
  19. A informação do novo documento é enviada para o CHAIN;
  20. Notificação "O documento foi inserido com sucesso";
- Fluxos alternativos:
    1. Campos não preenchidos
      - 1.1. Notificação "Todos os campos devem ser preenchidos";
      - 1.2. Retorna ao menu de inserção de informação;
    2. Ligação à internet indisponível
      - 2.1. Notificação "Sem acesso à internet, verifique a sua ligação";
      - 2.2. Retorna ao menu de inserção de informação;
    3. O utilizador quer adicionar mais artigos
      - 3.1. O utilizador seleciona o artigo;
      - 3.2. A informação relativa ao artigo é adicionada pelo utilizador;
      - 3.3. O botão adicionar artigo é acionado;
      - 3.4. Notificação "Artigo inserido com sucesso";
    4. O utilizador quer remover artigos
      - 4.1. O utilizador acede ao *tab* "Linha de Artigos";
      - 4.2. O utilizador aciona o artigo que quer remover;
      - 4.3. O atualizador acede ao menu atualizar linha de artigos;
      - 4.4. O utilizador carrega no botão remover;
      - 4.5. Notificação "O artigo foi removido com sucesso"
    5. O utilizador quer atualizar a informação de um artigo
      - 5.1. O utilizador acede ao *tab* "Linha de Artigos";

- 5.2. O utilizador aciona o artigo que quer atualizar;
- 5.3. O atualizador acede ao menu atualizar linha de artigos;
- 5.4. O utilizador altera a informação que pretende;
- 5.5. O utilizador carrega no botão atualizar;
- 5.6. Notificação "O artigo foi atualizado com sucesso"

- Diagrama de atividade:

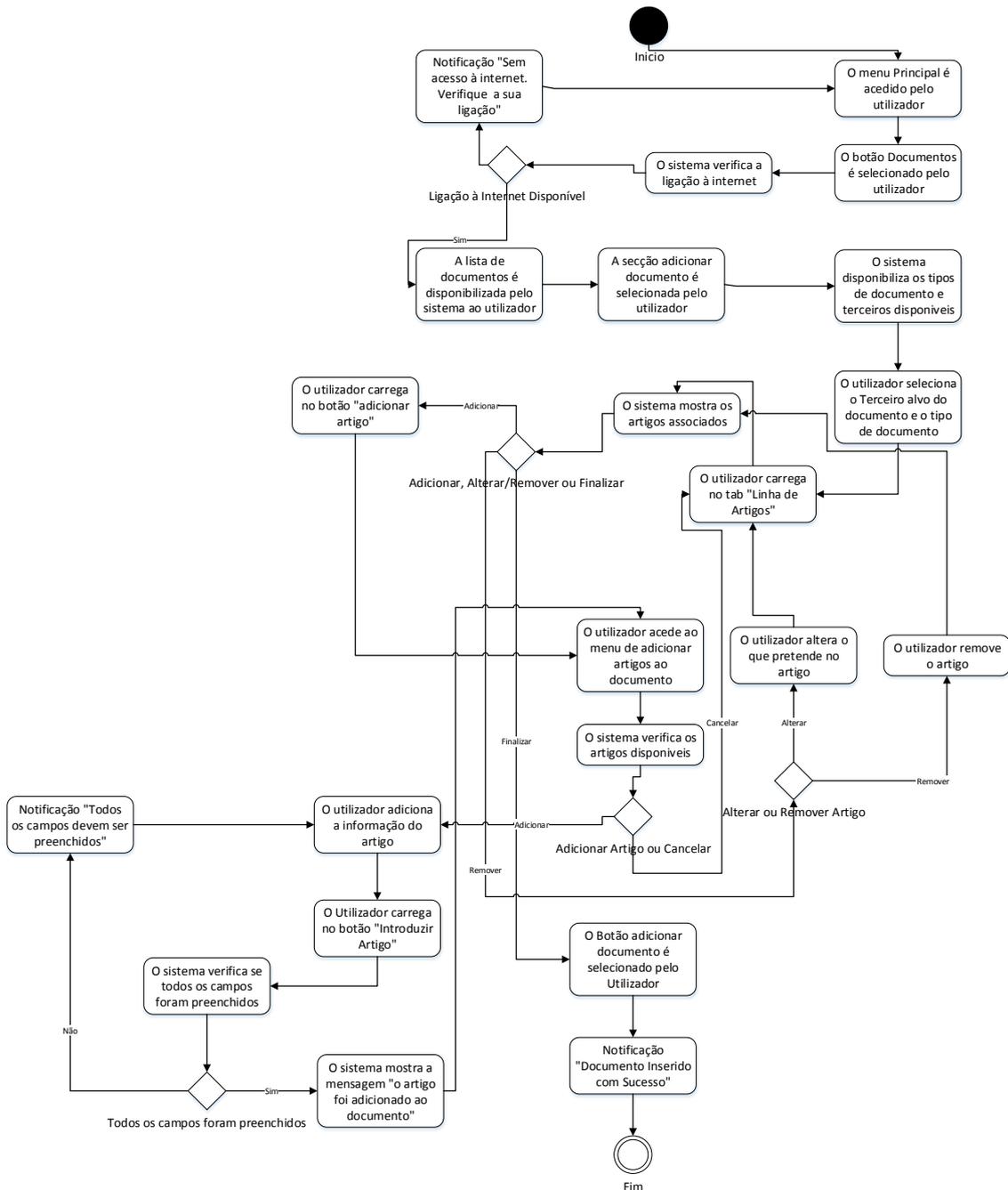


Figura 31 - Criar Documento (UML-DA)

- Diagramas de sequência:

1. **Cenário:** O utilizador cria um documento com uma linha de artigo.

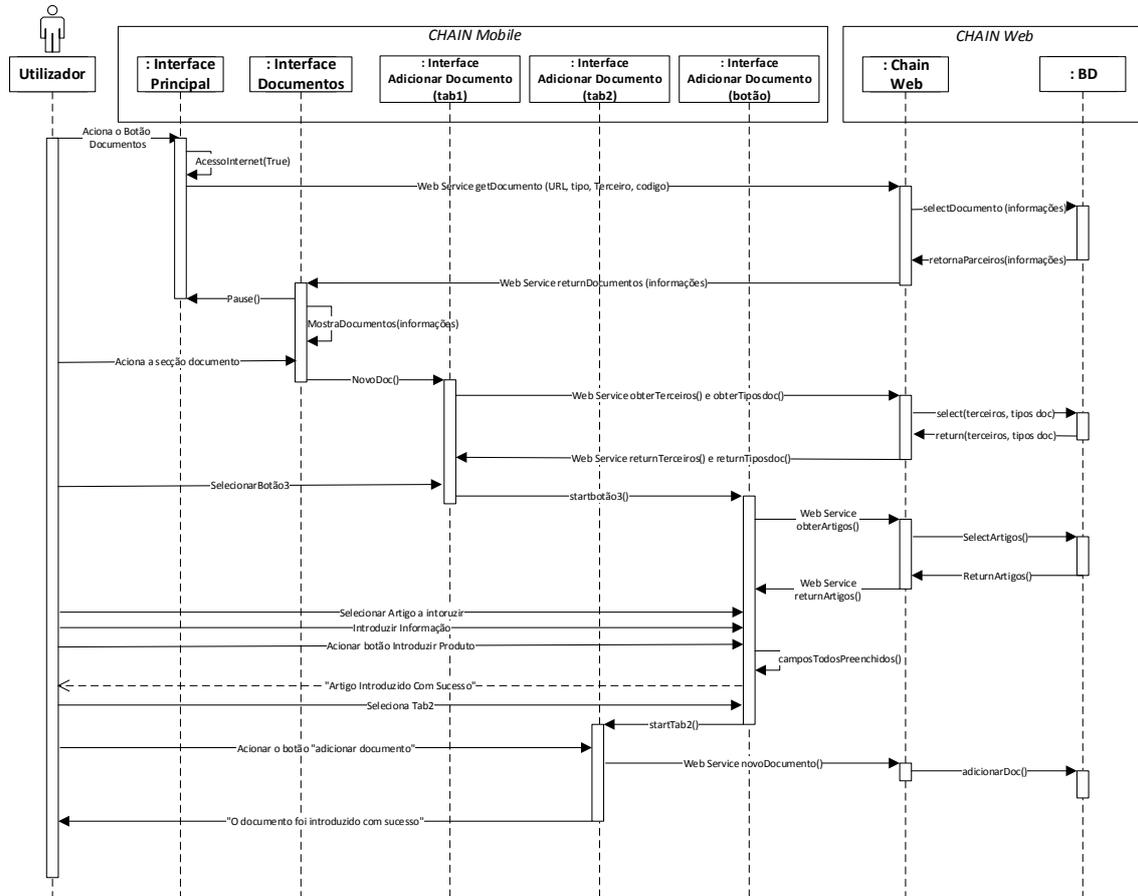


Figura 32 - Criar documento com um artigo associado (UML-DS)

2. **Cenário:** O utilizador adiciona dois artigos a um potencial documento.

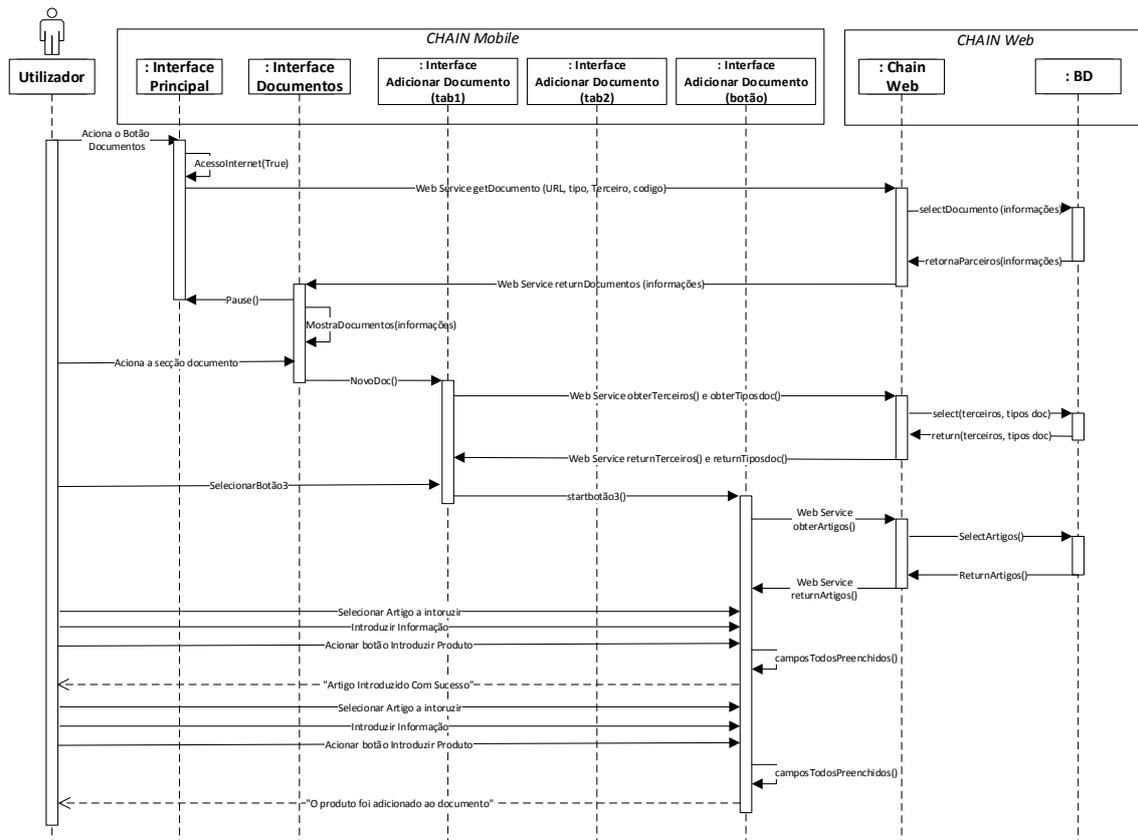


Figura 33 - Criar Documento com dois artigos (UML-DS)

3. **Cenário:** O utilizador remove um artigo de um potencial documento.

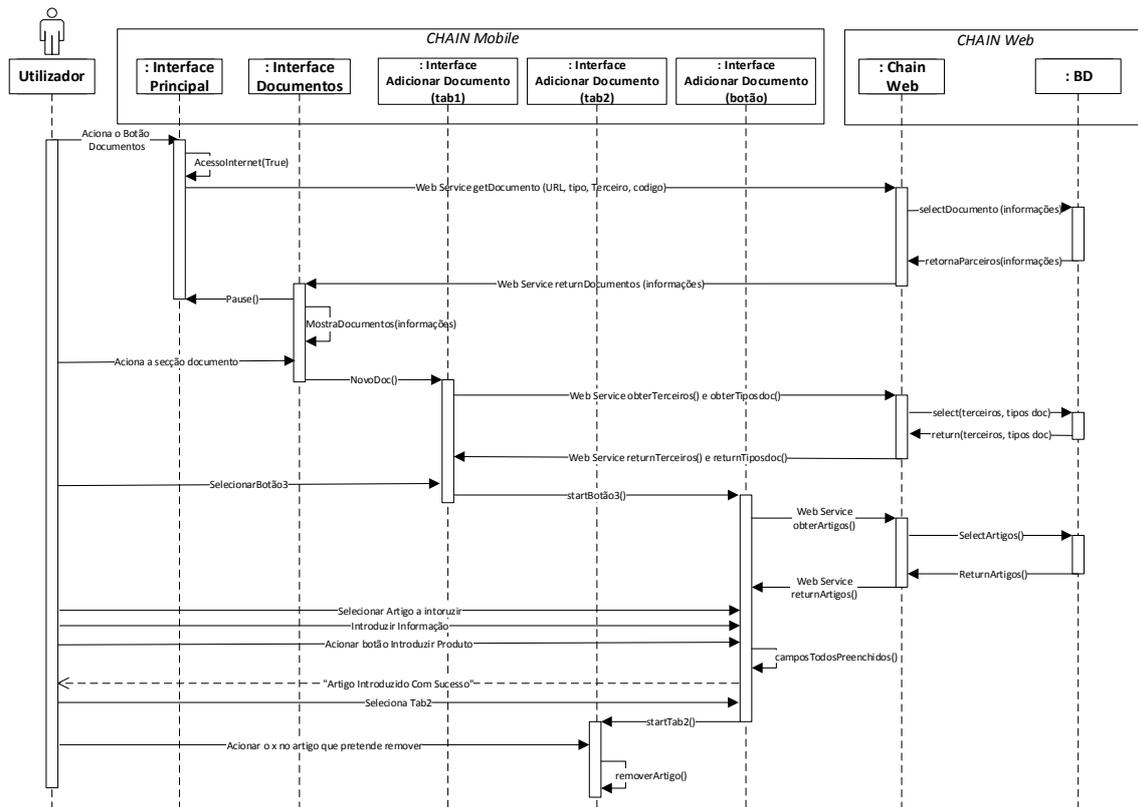


Figura 34 - Remover um artigo de um potencial documento (UML-DS)

4. **Cenário:** O utilizador não consegue aceder ao menu criar documento por falta de acesso à internet.

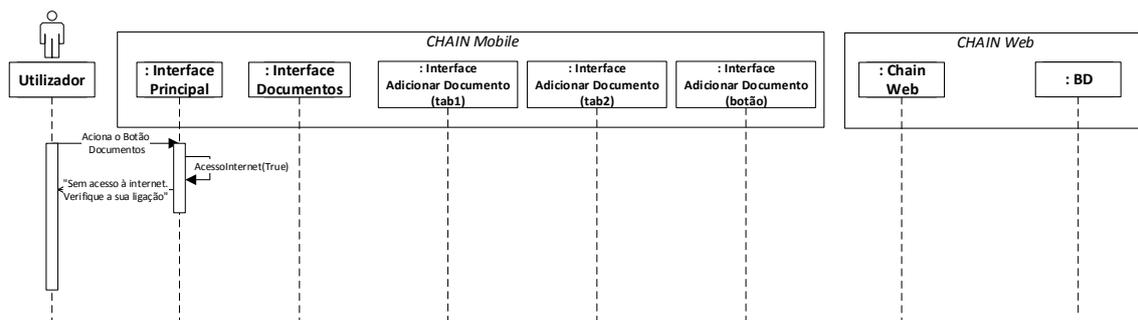


Figura 35 - Sem acesso à internet para aceder a documentos (UML-DS)

5. **Cenário:** O utilizador não consegue atribuir artigo ao documento por falta de preenchimento de algum dos campos.

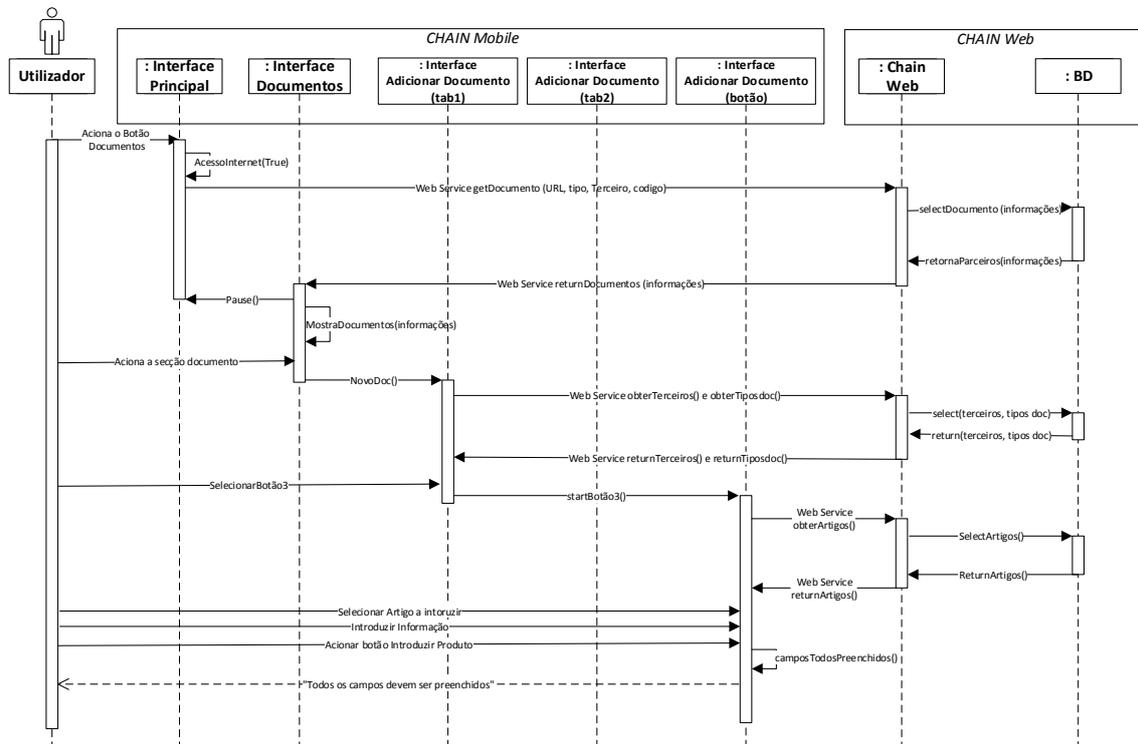


Figura 36 - Falha a associar um artigo a um documento (UML-DS)

A restante modelação referente a outros subsistemas do CHAIN Mobile pode ser consultada no Anexo A – Modelação UML.

### 5.1.2 Definição da Interface Gráfica do Utilizador (GUI)

A interface Gráfica do utilizador, que doravante também será referida como GUI, deriva do inglês de *Graphical User Interface*. Inicialmente, o projeto iria abordar o desenvolvimento de uma única solução suportada por três plataformas *mobile* distintas, nomeadamente no Android, IOS e Windows Phone e, como tal, foi concebido o GUI referente ao menu do *login* para essas três plataformas, com o objetivo de demonstrar algumas diferenças nos elementos de interface existentes e que caracterizam cada uma das plataformas *mobile*. Todos os restantes menus serão projetados para Android devido ao fato de ser esta a plataforma alvo da solução. Estes podem ser consultados no Anexo B – Interface Gráfico do Utilizador e Manual Técnico. A definição da interface gráfica do utilizador serviu para validar junto do cliente as funcionalidades a desenvolver. Tal como supracitado, o menu *login* foi desenhado para as plataformas Android, IOS e Windows Phone, com o intuito de denotar algumas diferenças a nível de interface, que caracterizam cada uma das plataformas. Apesar das diferenças se prenderem em detalhes relacionados com *design* dos componentes da interface, o utilizador pode estar mais

familiarizado com determinada plataforma, o que o pode levar a optar por essa plataforma devido a estas pequenas diferenças. Tendo a Doctron como futuro objetivo comercializar uma solução nativa para cada uma das plataformas acima referidas, foi desenhado um menu aludindo a essas diferenças.

#### 5.1.2.1 Android

A Figura 37, projeta a representação do menu de *login* na plataforma Android. A totalidade dos menus referentes à GUI da solução podem ser visualizados no Anexo B – Interface Gráfico do Utilizador e Manual Técnico.

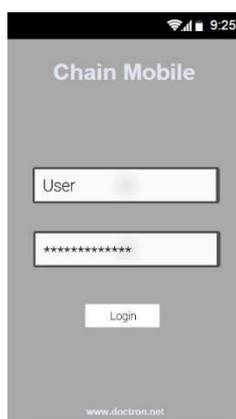


Figura 37 - Menu login (Android-GUI)

#### 5.1.2.2 IOS

A Figura 38, retrata o mesmo menu mas agora na plataforma IOS. Já é possível enumerar algumas diferenças relativas ao *design* dos componentes da interface gráfica do utilizador, como é o caso do botão de *login* e das caixas de introdução de texto.



Figura 38 - Menu login (IOS-GUI)

### 5.1.2.3 Windows Phone

Na Figura 39, tal como nas anteriores, é possível ver o menu *login* num dispositivo com a plataforma Windows Phone. Comparativamente ao *design* dos componentes, também denota algumas diferenças que caracterizam esta plataforma.



Figura 39 - Menu login (Windows Phone-GUI)

### 5.1.3 Documentação

Em conjunto com a GUI, foi também elaborada toda a documentação relacionada com a utilização dos menus.

#### 5.1.3.1 Menu Principal



Figura 40 - Menu principal (Android-GUI)

#### **Descrição:**

Na Figura 40, encontra-se o primeiro menu do CHAIN Mobile a que o utilizador apenas tem acesso após efetuar *login* no sistema. Neste menu, o utilizador pode aceder à página *Web* da Doctron através do *click* no URL que se encontra na parte inferior do menu. Quando o utilizador carrega no URL, tem a possibilidade de escolher o *browser* que pretende para aceder à página *Web* ou o sistema acede utilizando um *browser* pré-definido. Conforme necessidade pode

também aceder ao módulo de terceiros, artigos ou documentos através do *click* sobre os respetivos botões.

### 5.3.1.2 Menu Terceiros



Figura 41 - Menu Terceiros (Android-GUI)

#### **Descrição:**

A Figura 41, representa o menu de terceiros. Este menu surge quando o utilizador prime o botão terceiros no menu principal (Figura 40). Dentro deste menu, o utilizador tem acesso à lista de terceiros da sua organização. Para cada terceiro é possível realizar uma chamada para o seu contacto através de um clique no ícone de chamada ou navegar até à sua morada, seleccionando o ícone de localização. O sistema, através da comunicação por *Web Services* do tipo SOAP, recebe toda a informação necessária para as funcionalidades existentes neste menu. Carregando sobre o ícone adicionar (sinal representativo de uma soma no canto inferior direito), o utilizador acede ao menu referente à criação de terceiros representado na Figura 42. A qualquer altura o utilizador pode voltar ao menu principal, pressionando o botão *back* do seu dispositivo. Sempre que este menu é acedido a informação é refrescada.



Figura 42 - Criar Terceiro (GUI)

### **Descrição:**

Conforme o mencionado anteriormente, este é o menu referente à introdução de um novo terceiro. Através de *Web Services* do tipo SOAP, o sistema contacta o servidor do CHAIN no intuito de receber os grupos de terceiros existentes. Após selecionar o grupo pretendido, o utilizador introduz toda a informação necessária ao registo do novo terceiro. O acesso à internet e o preenchimento de todos os campos deve estar garantido de forma a ser possível completar a operação. Para terminar o processo, o utilizador carrega no botão adicionar terceiro e o CHAIN Mobile envia as informações relativas ao novo parceiro para o servidor do CHAIN. Tal como acontece em todos os menus, o utilizador pode a qualquer momento abortar a criação de um novo parceiro, carregando no botão *back* do dispositivo, neste caso, voltando ao menu Terceiros.

A restante documentação técnica pode ser consultada no Anexo B – Interface Gráfico do Utilizador e Manual Técnico.

## **5.2 Programação do Produto**

Este subcapítulo projeta o conceito fazer, abordando o desenvolvimento da solução CHAIN Mobile através da aplicação e adaptação da metodologia ágil Scrum. Retrata o desenvolvimento da aplicação após esta ser pensada, garantindo que tudo aquilo que foi pensado servirá como guia do desenvolvimento e que os seus requisitos serão rigorosamente cumpridos.

### **5.2.1 Aplicação da *Framework* Scrum**

Sendo o *sprint* a essência do Scrum, é importante começar pela definição da duração do *sprint*, tendo este a duração de quatro semanas. Neste projeto, efetivaram-se três *sprints*, derivando uma solução *mobile* potencialmente comercializável, que foi entregue à Doctron. Neste

subcapítulo, está contido o *Product Backlog* e o primeiro *sprint*, podendo os restantes ser consultados no anexo C – Complemento Scrum. Interessa também referir que, devido ao facto de ser um projeto realizado por uma só pessoa, foram nela concentrados os cargos de *Scrum Master*, *Product Owner* e de Equipa de Desenvolvimento em conjunto com as tarefas a eles associadas. O conceito de reunião deixou também de fazer sentido ao longo dos *sprints* pela mesma razão. Através da adaptação e simplificação da metodologia Scrum às condições do projeto, segue-se o *Product Backlog* em conjunto com o primeiro *sprint*:

**Product Backlog:**

O *Product Backlog* apresenta, de forma ordenada, a lista de requisitos do produto. Os requisitos apresentam-se sob a forma de tarefas e ordenado de acordo com a sua prioridade. A Tabela 9, representa o conteúdo do *Product Backlog* deste projeto.

Tabela 9 - Product Backlog (Scrum)

Prioridade	ID	Item	Quero	Para	Notas	Estado
Alta	1	Desenvolver Menus de Acordo com a GUI	Desenvolver Funcionalidade	Possibilitar a navegação ao utilizador		Terminado
	2	Realizar testes à GUI	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	3	Desenvolver Funcionalidade URL	Desenvolver Funcionalidade	Aceder à página Web da Doctron	Através do <i>click</i> o utilizador deve aceder ao à página <i>Web</i> da Doctron utilizando um <i>browser</i> predefinido ou por ele escolhido	Terminado
	4	Realizar testes ao URL	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	5	Funcionalidade Visualizar Terceiro	Desenvolver Funcionalidade	Ver todos os terceiros	O utilizador deve obter a informação relacionada com o terceiro, nomeadamente nome, contacto e morada	Terminado
	6	Realizar testes à funcionalidade visualizar terceiros	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
Média	7	Funcionalidade Visualizar Artigos	Desenvolver Funcionalidade	Ver todos os artigos, deve permitir o <i>download</i> individual	O utilizador deve obter a informação relacionada com o Artigo nomeadamente quantidade em stock, código e preço, mais detalhadamente poderá baixar um PDF	Terminado

Prioridade	ID	Item	Quero	Para	Notas	Estado
	8	Realizar testes a funcionalidade visualizar artigos	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	9	Funcionalidade Visualizar Documentos	Desenvolver Funcionalidade	Ver todos os documentos, deve permitir o <i>download</i>	O utilizador deve obter a informação relacionada com o documento através do <i>download</i> do PDF do documento	Terminado
	10	Realizar testes a funcionalidade visualizar documentos	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	11	Enviar Artigo	Desenvolver Funcionalidade	Anexar o Artigo	O utilizador deve poder anexar um artigo com qualquer aplicação no dispositivo que o permita	Terminado
Média	12	Realizar testes à funcionalidade enviar artigo	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	13	Enviar Documento	Desenvolver Funcionalidade	Anexar o documento	O utilizador deve poder anexar um documento com qualquer aplicação no dispositivo que o permita	Terminado
	14	Realizar testes à funcionalidade enviar Documento	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	15	Realizar Chamada	Desenvolver Funcionalidade	Realizar chamada para contacto recebido do terceiro	Através do contacto recebido o utilizador pode iniciar uma chamada	Terminado
	16	Realizar testes à funcionalidade realizar chamada	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	17	Obter Direções	Desenvolver Funcionalidade	Navegar até à morada do terceiro	Usando uma aplicação de navegação presente no dispositivo	Terminado
	18	Realizar testes à funcionalidade obter direções	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
Baixa	19	Criar Documento	Desenvolver Funcionalidade	Inserir um novo documento	O utilizador pode criar um documento e adicioná-lo à base de dados	Terminado
	20	Realizar testes à funcionalidade criar documento	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	21	Criar Terceiro	Desenvolver Funcionalidade	Inserir um novo terceiro	O utilizador pode criar um terceiro e adicioná-lo à base de dados	Terminado

Prioridade	ID	Item	Quero	Para	Notas	Estado
	22	Realizar testes à funcionalidade criar terceiro	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado
	23	Desenvolver Autenticação	Desenvolver Funcionalidade	Permitir autenticação dos diferentes utilizadores do CHAIN.	Através do login o CHAIN decidirá que conteúdos pode o utilizador aceder	Terminado
	24	Realizar testes à autenticação	Testar Funcionalidade	Verificar se faz o que é pretendido		Terminado

### **Primeiro Sprint:**

O objetivo do deste *sprint* foi terminar os 6 primeiros itens do *Product backlog* de acordo com a sua importância. Na Tabela 10, estão representadas as tarefas que ficaram decididas no planeamento do Sprint, dando origem ao *Sprint Backlog*. Ao fim do *sprint*, foi verificado o estado das tarefas, que equivale ao evento de revisão do *sprint*. Verificou-se que todas as tarefas definidas no *Sprint Backlog* foram concluídas com sucesso, transformando-se em incrementos ao produto potencialmente comercializável.

*Tabela 10 - Primeiro Sprint (Scrum)*

<i>Sprint Backlog</i>			Revisão do Sprint	Incremento
ID	Tarefas	Notas	Estado	Descrição
1	Desenvolver Menus de Acordo com a GUI	Elaborar os menus e tudo aquilo que é necessário à sua navegação.	Terminado	Definição dos menus e navegação entre eles
2	Realizar testes à GUI	Verificar se faz o que é pretendido.	Terminado	
3	Desenvolver Funcionalidade URL	O utilizador deve obter a informação relacionada com o terceiro, nomeadamente nome, contacto e morada.	Terminado	Acesso à página Web da Doctron no menu principal
4	Realizar testes ao URL	Verificar se faz o que é pretendido.	Terminado	
5	Funcionalidade Visualizar Terceiro	O utilizador deve obter a informação relacionada com o terceiro, nomeadamente nome, contacto e morada.	Terminado	Ver todos os Terceiros no menu terceiros
6	Realizar testes à funcionalidade visualizar terceiros	Verificar se faz o que é pretendido.	Terminado	

Como supracitado, os restantes *sprints* ocorridos ao longo do desenvolvimento, podem ser consultados no anexo C – Complemento Scrum.

### 5.3 Resumo do Capítulo

Este capítulo dividiu-se em dois grandes conceitos: o pensar e o fazer. Quando se pretende atingir um destino é importante escolher e definir o caminho a seguir para o alcançar. Para garantir que o destino é o pretendido, devem ser especificadas as direções e delineadas as necessidades a que é preciso responder para o percorrer e chegar a esse fim. O conceito pensar, reflete esta mesma visão, especificar as direções e as necessidades a fim de garantir que no fim, o destino pretendido foi alcançado. O fim desejado é o desenvolvimento da aplicação *mobile* para Android, que foi retratada na segunda parte do capítulo através da aplicação e adaptação da metodologia ágil de desenvolvimento, Scrum. A primeira parte deste capítulo foi referente à definição da interface gráfica do utilizador e à especificação dos requisitos da solução através da linguagem de modelação UML.

## 6 – Resultados

Deste projeto resultou uma solução *mobile* para Android, todo o seu planeamento e fundamentação teórica, que de acordo com os requisitos e necessidades do cliente veio estender as funcionalidades do seu ERP CHAIN. Este produto combina algumas das funcionalidades presentes nos dispositivos *mobile*, com suporte para a plataforma Android, com funcionalidades já presentes no CHAIN numa única solução. Este novo produto proporciona uma vantagem competitiva à Doctron, face aos seus concorrentes, através da extensão de funcionalidades do seu produto (ERP CHAIN) para dispositivos *mobile*, conferindo mobilidade e simplicidade aos seus clientes.

### 6.1 Impacto do Produto

Este subcapítulo aborda as vantagens que o produto desenvolvido irá trazer à organização (Doctron) e que garantidamente uma solução deste género trará aos seus clientes.

#### 6.1.1 Na organização

Em função da organização, foram várias as vantagens identificadas que o produto acrescenta, nomeadamente:

- **Vendas**  
Através de um público-alvo mais abrangente graças ao lançamento de extensões ao seu produto, as vendas são potencialmente aumentadas.
- **Serviços**  
Através do aumento do número de plataformas suportadas, com soluções otimizadas em função das funcionalidades do CHAIN, o seu leque de serviços é alargado e pode ser utilizado com maior eficiência.
- **Vantagem Competitiva**  
Oferecendo soluções para equipamentos *mobile* garantem à Doctron uma vantagem competitiva face aos seus concorrentes e permitem explorar e acompanhar novos segmentos de mercado.
- **Aumento da confiança e fidelização do cliente.**  
Através de novas funcionalidades a organização mostra-se interessada em corresponder às necessidades do cliente, oferecendo novas e melhores soluções.

#### 6.1.2 No cliente

Ainda não foi possível avaliar no cliente, mas olhando para um produto com estas características, garantidamente terá as seguintes vantagens:

- Aumento da confiança e fidelização do cliente  
Através de novas funcionalidades, a organização mostra-se interessada em corresponder às necessidades do cliente oferecendo novas e melhores soluções.
- Gestores que pelo seu nível hierárquico na organização não eram utilizadores do CHAIN passarão a sê-lo (principalmente para consultas)
- Desenvolvimento Nativo  
Proporciona uma aplicação desenvolvida especificamente para a plataforma Android, tirando partido de todas as vantagens associadas a este tipo de desenvolvimento.
- Um ambiente familiar para utilizadores de dispositivos Android  
Devido ao seu desenvolvimento nativo, integra elementos da interface gráfica do utilizador disponibilizados pelo SDK da Google.
- Mobilidade  
Sendo uma aplicação *mobile*, pode ser utilizada em qualquer altura e em tempo real por dispositivos que suportam Android.
- As mesmas funcionalidades em menos passos e em tempo real (melhorias no processo)  
Através do uso da aplicação, é possível aceder a funcionalidades do CHAIN em menos passos e mais rapidamente.
- Inovação  
Através da utilização de dispositivos *mobile* e dadas as suas características, apresenta uma inovação tecnológica.

## 6.2 Resumo do Capítulo

Ao longo deste capítulo podemos verificar que do projeto resulta uma aplicação *mobile* funcional para a plataforma Android, que permite estender funcionalidades do ERP CHAIN para dispositivos *mobile*. Essa extensibilidade oferece aos clientes da Doctron inúmeras vantagens, assim como potencializa as vendas e a exploração de novos segmentos de mercado.

## 7 – Conclusão

Este foi um projeto ambicioso que explorou uma área complementar à formação académica em Engenharia e Gestão de Sistemas de Informação. Foi um projeto realizado numa empresa, Doctron, e para a empresa, tendo como foco o seu produto CHAIN e a extensão deste para o mundo *mobile*.

Numa primeira fase a empresa pretendia expandir funcionalidades do seu produto CHAIN para dispositivos *mobile* e pretendia que essa expansão se desse de forma rápida, englobando as três plataformas *mobile* mais utilizadas em Portugal, nomeadamente o Android, IOS e Windows Phone. Desejava, desta forma, ganhar uma vantagem competitiva perante os seus concorrentes, através do alargamento das plataformas que suportam os seus serviços e atribuir-lhe uma maior mobilidade.

No entanto, esta era uma área nova para a empresa e para realizar essa extensão de funcionalidades a empresa considerou importante que fosse elaborada uma análise aos seus concorrentes e que oferta estes detinham para dispositivos *mobile*. Requisitou também o levantamento do estado da arte sobre as abordagens de desenvolvimento para dispositivos *mobile*, sobre as ferramentas existentes para cada abordagem, da comunicação e de como esta seria efetuada com o servidor do CHAIN e, ainda, uma análise do ponto de vista de mercado das três plataformas, de forma a compreender a sua adoção e evolução ao longo do tempo.

Através deste levantamento do estado da arte, a empresa resolveu que em vez de uma solução que fosse suportada pelas três plataformas alvo, através de uma abordagem multiplataforma, fossem, ao invés, desenvolvidas soluções nativas à medida para cada plataforma, tendo optado pela plataforma Android como alvo de desenvolvimento prioritário.

A partir desta decisão, foi então desenvolvida uma solução Android com extensão de funcionalidades do CHAIN, acompanhada de toda a fundamentação teórica requerida pela empresa e respetivo planeamento e modelação.

Este trabalho fornece todo o material necessário para o planeamento e desenvolvimento *mobile*, identificando os caminhos possíveis para atingir o objetivo, suas vantagens e desvantagens. Faculta as informações necessárias à tomada de decisão com vista ao desenvolvimento de aplicações *mobile* para as plataformas Android, IOS e Windows Phone.

## 7.1 Revisitar os Objetivos

Este Projeto atingiu todos os objetivos propostos. Relativamente ao plano do projeto, foram definidas as fases do projeto e o plano de atividades foi validado pelo responsável da empresa. Foi concebida também a análise de riscos precavendo qualquer situação adversa ou benéfica ao projeto.

Em função da empresa e do produto é contextualizada a empresa, definindo o setor de atividade, produto e a sua descrição. Sendo este projeto em função do produto que a empresa comercializa, não é relevante detalhar a empresa mas sim o produto. O produto da empresa é um ERP, denominado CHAIN, cujas funcionalidades foram descritas e explicadas. Procedeu-se a uma análise cuidada à concorrência de acordo com os concorrentes identificados pela Doctron, sendo analisadas as soluções que estes detinham no âmbito *mobile* e identificando as funcionalidades da sua oferta. De seguida, analisou-se o impacto que a solução *mobile* desenvolvida teria no posicionamento do produto, de acordo com as estratégias delineadas pela empresa.

Por fim, foram modeladas as funcionalidades e requisitos da solução *mobile*, procedendo-se à definição da GUI. O projeto terminou com o desenvolvimento da solução *mobile* para a plataforma Android, tendo sido o seu desenvolvimento efetuado com o auxílio da metodologia ágil Scrum.

Em suma, os objetivos propostos foram atingidos da seguinte forma:

### **Contexto do projeto:**

- **Plano do Projeto**
  - Definir as fases do projeto  
O projeto foi decomposto em fases contendo tarefas a realizar.
  - Elaborar o plano de atividades  
O trabalho foi dividido de forma cuidada e pensada ao longo do tempo.
  - Analisar riscos  
Foram identificados os potenciais riscos do projeto, supondo a sua probabilidade de ocorrência e impacto, definindo estratégias para lidar com a sua ocorrência.
- **Contextualização da empresa e enquadramento do produto**
  - Descrever a Empresa

A empresa foi caracterizada, tal como o seu produto, que foi identificado e detalhado.

- Traçar o Produto

O objetivo foi atingido através da identificação e descrição das funcionalidades do produto CHAIN.

- Analisar Concorrência

Foi efetuada a análise da concorrência e dos produtos *mobile* oferecidos, englobando a descrição de suas funcionalidades.

- Posicionar o Produto

Foram identificadas as estratégias de Marketing elaboradas pela empresa e o impacto no posicionamento do produto.

- Desenvolvimento da aplicação informática

- Conceção da Solução

Foi modelada e desenhada toda solução em função dos requisitos do cliente, tendo esta sido modelada através de UML e definida a sua GUI.

- Programar Produto

Foi desenvolvida uma aplicação informática Android com a capacidade de estender funcionalidades do ERP CHAIN.

### **Contexto Pessoal:**

- Integrar o Mercado de Trabalho

Foi possível conhecer novas realidades, enfrentar e ultrapassar dificuldades associadas ao mercado trabalho, aplicar a minha formação académica em contextos reais, desenvolvendo novas competências e conhecimentos.

- Complementar a formação Académica

Aquisição de competências na área do desenvolvimento *mobile*, através da programação de uma solução para a plataforma Android.

- Melhorar e Desenvolver as *Soft Skills*

Lidar com clientes e superiores hierárquicos, resolver problemas, enfrentar situações adversas, otimizar a gestão do tempo, melhorar a capacidade de comunicação e pensamento crítico, desenvolver atitude positiva e pró-ativa.

## **7.2 Oportunidades**

O desenvolvimento deste projeto permitiu o crescimento pessoal e académico, quer no contexto do mercado de trabalho, quer na engenharia e gestão de sistemas de informação. Foi

um projeto que incidiu sobre o desenvolvimento Android através da utilização da linguagem JAVA. Esta linguagem lidera o ranking de linguagens mais utilizadas em diversos contextos, especialmente no contexto empresarial. Através deste projeto, foi possível alargar e aprofundar os conhecimentos sobre a linguagem e ainda aprender a utilizá-la no contexto de desenvolvimento de aplicações para a plataforma Android.

Esta não foi a única oportunidade, existiram várias outras. Não fugindo da plataforma Android e do desenvolvimento que esta foi alvo, o desafio prendeu-se com a utilização de *Web Services* do tipo SOAP para realizar a comunicação. Este foi realmente o maior desafio de programação deste projeto. O fato de não ser suportado nativamente na plataforma Android, a falta de documentação e de suporte, a necessidade da utilização de uma biblioteca desenvolvida por terceiros, o fato de a comunicação através de *Web Services* do tipo SOAP na plataforma Android ser definida por muitos, como uma tarefa extremamente complicada e um desafio apenas ao alcance de programadores experientes e especializados na plataforma, este conseguiu ser ultrapassado, permitindo que a solução *mobile* e o servidor do CHAIN interagissem através de *Web Services* do tipo SOAP.

Permitiu, também, perceber de que forma pode ser benéfico para as empresas e seus clientes a extensão de funcionalidades de um ERP, integrando-o no mundo mobile, tirando partido de todas as funcionalidades e vantagens que este adiciona.

Outra oportunidade foi, a introdução a uma metodologia ágil de desenvolvimento e aplicação da mesma na criação da solução. Esta metodologia compreendia três grandes intervenientes no processo desenvolvimento: o *Product Owner*, *Scrum Master* e Equipa de Desenvolvimento. Sendo este um projeto realizado por apenas uma pessoa, as tarefas associadas a cada equipa convergiram para a mesma pessoa, tendo impacto direto no resultado dos *sprints* e na adaptação da metodologia.

Possibilitou a integração no mundo do trabalho e o contacto com todas as situações que a ele estão associadas. Foi importante verificar que o trabalho em equipa e a dependência entre funções complementares é fundamental para o sucesso da organização. Foi uma forma prática de interagir com o cliente, encarar as contrariedades, gerir os requisitos, que de reunião para reunião nem sempre se mantinham inalterados, gerir a mudança ao plano de projeto e redistribuir o trabalho, nomeadamente aquando da alteração da abordagem de desenvolvimento de multiplataforma para nativa, melhorar a comunicação, conhecer novas realidades e aplicar toda a formação adquirida ao longo do meu percurso académico.

Ao longo deste projeto, foi possível alargar os horizontes da minha linha de pensamento e ultrapassar dificuldades nunca antes sentidas. Foi uma aprendizagem contínua que complementou a minha formação académica, através da experiência obtida ao lidar com problemas que surgiram e poderão surgir no mercado de trabalho. Teve, também, um grande impacto nas ditas *soft skills*, permitindo o seu desenvolvimento e melhoria..

### 7.3 Trabalhos Futuros

Este projeto contém todo o planeamento e especificação do produto necessário à implementação da mesma solução para as plataformas *mobile* IOS e Windows Phone. Além destas extensões, futuramente podem ser adicionadas novas funcionalidades cujo planeamento não é coberto por este projeto.

## Referências Bibliográficas

- AlShahwan, F., Moessner, K., & Carrez, F. (2010). Evaluation of Distributed SOAP and RESTful Mobile Web Services. *International Journal on Advances in Networks and Services*, 447-467.
- APDC. (23 de Junho de 2014). *Notícias*. Obtido de APDC – Associação Portuguesa para o Desenvolvimento das Comunicações:  
[http://www.apdc.pt/Artigo.aspx?channel\\_id=3430D560-04A5-432A-B893-3224E5CEFBBC&content\\_id=6749DDDA-588E-4DE1-95C2-F22A9B97ECF6&lang=pt](http://www.apdc.pt/Artigo.aspx?channel_id=3430D560-04A5-432A-B893-3224E5CEFBBC&content_id=6749DDDA-588E-4DE1-95C2-F22A9B97ECF6&lang=pt)
- Apple. (20 de Outubro de 2014). *Apple*. Obtido de IOS8: <https://www.apple.com/pt/ios/>
- Bloomberg. (1 de Abril de 2014). *Google Buys Android for Its Mobile Arsenal*. Obtido de Bloomberg Business Week: <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>
- Cerami, E. (2002). *Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL* (First Edition ed.). O'Reilly.
- Chandio, A. A., Zhu, D., & Sodhro, A. H. (2012). International MultiConference of Engineers and Computer Scientists. *Integration of Inter-Connectivity of Information. I*, pp. 651-655. Hong Kong: Newswood Limited.
- Charland, A., & Leroux, B. (2011). Mobile application development: web vs. native. *Communications of the ACM*, 49-53.
- Ciman, M., Gaggi, O., & Gonzo, N. (2014). Cross-Platform Mobile Development: A Study on Apps with Animations. *Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC, 14*. Gyeongju, Korea.
- Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, (pp. 323 - 328). Sardinia. doi:10.1109/IWCMC.2013.6583580
- Darwin, I. F. (2014). *Java Cookbook, 3rd Edition*. O'Reilly Media.
- Dehlinger, J., & Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. *Workshop on Mobile Software Engineering*.
- Doctron, Lda. (2014). *Doctron - Sistemas de Informação, Lda*. Obtido em 02 de Janeiro de 2014, de Doctron - Sistemas de Informação, Lda: <http://www.doctron.net/>
- Exame Informatica. (25 de Outubro de 2014). *E as marcas que mais vendem smartphones em Portugal são....* Obtido de Exame Informatica:  
<http://exameinformatica.sapo.pt/noticias/mercados/2014-10-24-E-as-marcas-que-mais-vendem-smartphones-em-Portugal-sao>
- F3M. (2014). *F3M Information Systems*. Obtido em 27 de Fevereiro de 2014, de <http://www.f3m.pt/>
- Fischer, N., & Smolnik, S. (2013). The Impact of Mobile Computing on Individuals, Organizations, and Society – Synthesis of Existing Literature and Directions for Future

- Research. *46th Hawaii International Conference on System Sciences* (pp. 1082-1091). IEEE.
- FRED VOGELSTEIN. (1 de Abril de 2014). *How the Android Ecosystem Threatens the iPhone*. Obtido de WIRED: [http://www.wired.com/2011/04/mf\\_android/all/1](http://www.wired.com/2011/04/mf_android/all/1)
- Galvin, D. (2 de 2 de 2004). Protótipo de sistema CRM para dispositivos moveis utilizando a tecnologia .NET. Brazil. Obtido em 20 de 1 de 2014, de [http://www.bc.furb.br/docs/MO/2004/305302\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2004/305302_1_1.pdf)
- Gandhewar, N., & Sheikh, R. (2010). Google Android: An Emerging Software Platform For. *International Journal on Computer Science and Engineering*, 12-17.
- Gil, B. (1 de Julho de 2011). Monitorização Remota de Smartphones: Estudo e implementação de uma aplicação de controlo parental. Lisboa, Lisboa, Portugal. Obtido em 20 de Janeiro de 2014, de <http://repositorio-iul.iscte.pt/bitstream/10071/5998/1/Tese%20-%20Bruno%20Gil.pdf>
- Goadrich, M. H., & Rogers, M. P. (2011). Smart smartphone development: iOS versus android. *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 607-612). Nova Iorque, USA: ACM.
- Google. (2 de Outubro de 2014). *Android Phones*. Obtido de Android: <http://www.android.com/phones/>
- Google. (1 de Março de 2014). *Android Studio*. Obtido de Android Developers: <https://developer.android.com/sdk/installing/studio.html>
- Hall, S. P., & Anderson, E. (2009). OPERATING SYSTEMS FOR MOBILE COMPUTING. *CCSC: Rocky Mountain Conference* (pp. 64-71). Consortium for Computing Sciences in Colleges.
- Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2012). Comparing Cross-Plataform Development Approaches For Mobile Applications. *WEBIST 2012 - 8th International Conference on Web Information Systems and Technologies*, (pp. 299-311). Porto.
- Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2012). COMPARING CROSS-PLATFORM DEVELOPMENT. *WEBIST2012-8thInternational Conferenceon WebInformation Systemsand Technologies*, (pp. 299-311). Munster.
- Hsieh, C.-t. (2007). Mobile Commerce: Assessing New Business Opportunities. *Communications of the IIMA*, 9(1), 87-100.
- Hydra iT. (2014). Hydra iT Tecnologias de Informação e Conteudos LDA. Obtido em 27 de Fevereiro de 2014, de <http://www.hydra.pt/>
- IDC. (1 de Abril de 2014). *Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013*. Obtido de IDC - International Data Corporation: <http://www.idc.com/getdoc.jsp?containerId=prUS24676414>
- IEEE Spectrum. (12 de Setembro de 2014). *Top 10 Programming Languages - Spectrum's 2014 Ranking*. Obtido de IEEE SPECTRUM: <http://spectrum.ieee.org/computing/software/top-10-programming-languages>

- INFOS. (2014). INFOS Promovemos Eficiencia. Obtido em 27 de Fevereiro de 2014, de <http://www.infos.pt/main.php?id=4>
- Jendrock, E., Cervera-Navarro, R., Evans, I., Gollapudi, D., Haase, K., Markito, W., & Srivathsa, C. (Setembro de 2014). *The Java EE 7 Tutorial, Release 7 for Java EE Platform*. Obtido de Oracle: <http://docs.oracle.com/javaee/7/tutorial/doc/home.htm>
- Jin, X., Hu, X., Ying, K., Du, W., & Yin, H. (2014). Code Injection Attacks on HTML5-based Mobile Apps: Characterization, Detection and Mitigation. *21st ACM Conference on Computer and Communications Security (ACM CCS 2014)*. Scottsdale, Arizona, USA.
- JOŠT, G., HUBER, J., & HERIČKO, M. (2013). Using Object Oriented Software Metrics for Mobile. *Second Workshop on Software Quality Analysis, Monitoring, Improvement and Applications SQAMIA 2013*, (pp. 17- 27). Novi Sad, Serbia.
- Juntunen, A., Jalonen, E., & Luukkainen, S. (2013). HTML 5 in Mobile Devices – Drivers and Restraints. *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, (pp. 1053 - 1062). Wailea, Maui, HI. doi:10.1109/HICSS.2013.253
- Kalin, M. (2013). *Java Web Services: Up and Running* (Second Edition ed.). O'REILLY.
- Kleiner, C., & Schneider, T. (2011). Securing SOAP Web Services for Mobile Devices on Different Platforms. *MMS 2011: Mobile und ubiquitäre Informationssysteme. Proceedings der 6. Konferenz*, 185, pp. 25-38. Kaiserslautern, Alemanha.
- Kotler, P., & Armstrong, G. (2013). *Principles of Marketing (Global Edition)* (Fifteenth Edition ed.). PEARSON.
- ksoap2. (15 de Agosto de 2014). *ksoap2-android - A lightweight and efficient SOAP library for the Android platform*. Obtido de Google Project Hosting: <https://code.google.com/p/ksoap2-android/>
- Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9), 140 - 150.
- Lee, J., Lee, S.-J., & Wang, P.-F. (2014). A Framework for Composing SOAP, Non-SOAP and Non-Web Services. *Services Computing, IEEE Transactions on*, 1.
- Leite, L., Lago, N., Gerosa, M. A., & Kon, F. (2013). Um Middleware para Encenação Automatizada de Coreografias de Serviços Web em Ambientes de Computação em Nuvem. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 14.
- Liang, Y. D. (2014). *Introducing to JAVA PROGRAMMING: COMPREHENSIVE VERSION* (10 ed.). Prentice Hall.
- Liu, H.-C., Sung, W.-P., & Wenli, Y. (2013). Development of android software update system based on web service. *Information Technology and Computer Application Engineering: Proceedings of the International Conference on Information Technology and Computer Application Engineering (ITCAE 2013)* (pp. 249-252). CRC Press.
- Machado, A., Padoin, E. L., Salvadori, F., Righi, L., Campos, M., Sausen, P. S., & Dill, S. L. (2008). Utilização de Dispositivos Móveis, Web Services e Software Livre no Monitoramento Remoto de Pacientes. *CONGRESSO BRASILEIRO DE INFORMÁTICA NA SAÚDE. XI. Anais: CBIS*.

- MacWin. (2014). MacWin. Obtido em 24 de Fevereiro de 2014, de <http://www.macwin.pt/>
- Mallya, R., & Bhagat, S. (2013). Understanding Efficiency of Executable UML with Model Compilers. *National Conference on New Horizons in IT*, (pp. 112-116). Mumbai.
- Mazo, S., Otón, S., de-Marcos, L., García, A., & García, E. (2012). RESTful Service Oriented Architecture for Querying and Publishing Learning Objects in Repositories. *The Fourth International Conference on Mobile, Hybrid, and On-line Learning* (pp. 20-23). Valencia, Espanha: IARIA.
- Mengshoel, O. J., Iannucci, B., & Ishihara, A. (2013). Mobile Computing: Challenges and Opportunities for Autonomy and Feedback. *8th International Workshop on Feedback Computing*.
- Microsoft. (16 de Setembro de 2014). *Diagramas de atividade UML: referência*. Obtido de Microsoft Developer Network: <http://msdn.microsoft.com/pt-BR/library/dd409360.aspx>
- Microsoft. (17 de Setembro de 2014). *Diagramas de sequência UML: referência*. Obtido de Microsoft Developer Network: <http://msdn.microsoft.com/pt-BR/library/dd409377.aspx>
- Microsoft. (24 de Outubro de 2014). *Funcionalidades do Windows Phone*. Obtido de Windows Phone: <http://www.windowsphone.com/pt-PT/features>
- Microsoft. (1 de Julho de 2014). *Microsoft officially welcomes the Nokia Devices and Services business*. Obtido de News Center: <http://news.microsoft.com/2014/04/25/microsoft-officially-welcomes-the-nokia-devices-and-services-business/>
- Mumbaikar, S., & Padiya, P. (Maio de 2013). Web Services Based On SOAP and REST Principles. *International Journal of Scientific and Research Publications*, 3(5), 1-4.
- Murphy, M. L. (2014). *The Busy Coder's Guide to Android Development* (6.1 ed.). CommonsWare.
- OASIS. (1 de Março de 2014). *Basic Profile Version 2.0*. Obtido de OASIS - Advancing Open Standards for the Information Society: <http://docs.oasis-open.org/ws-brsp/BasicProfile/v2.0/BasicProfile-v2.0.html>
- Oliveira, F. C., & Castro, R. d. (2013). Desenvolvimento de API Android para Validador WSDL de WebServices SOAP em Java. *TIS - Tecnologias, Infraestrutura e Software*, 66-77.
- Oliveira, L. R., & Medina, R. D. (2007). Desenvolvimento de objetos de aprendizagem para dispositivos móveis: uma nova abordagem que contribui para a educação. *RENOTE - Revista Novas Tecnologias na Educação*, 5(1). Obtido de lumenagencia.
- OMG. (2014). *Unified Modeling Language (UML)*. (I. Object Management Group, Produtor) Obtido em 13 de Janeiro de 2014, de <http://www.uml.org/>
- Open Handset Alliance. (1 de Setembro de 2014). *Open Handset Alliance*. Obtido de Open Handset Alliance: <http://www.openhandsetalliance.com/index.html>
- ORACLE. (22 de Setembro de 2014). *JAVA Platform Standard Edition 8 Documentation*. Obtido de ORACLE: JAVA Documentation: <http://docs.oracle.com/javase/8/docs/>

- ORACLE. (22 de Setembro de 2014). *Obtenha Informações sobre a Tecnologia JAVA*. Obtido de Java: [http://www.java.com/pt\\_BR/about/](http://www.java.com/pt_BR/about/)
- Oxford Dictionaries. (10 de Setembro de 2014). Obtido de Oxford Dictionaries - Language matters: [http://www.oxforddictionaries.com/us/definition/american\\_english/goal?q=goals](http://www.oxforddictionaries.com/us/definition/american_english/goal?q=goals)
- Palmieri, M., Singh, I., & Cicchetti, A. (2012). Comparison of Cross-Platform Mobile Development Tools. *16th International Conference on Intelligence in Next Generation Networks (ICIN): Realising the Power of the Network* (pp. 179-186). Berlin, Alemanha: IEEE.
- Panetto, H., & Cecil, J. (Dezembro de 2013). Information Systems for Enterprise Integration, Interoperability and Networking: Theory and Applications. *Enterprise Information Systems*, 1-6. doi:10.1080/17517575.2012.684802
- Pastore, S. (2014). Developing mobile educational apps: development strategies, tools and business models. *ACSII Advances in Computer Science: an International Journal*, 3(1), 27-36.
- PHC. (2014). Software de Gestão PHC. Obtido em 23 de Fevereiro de 2014, de <http://www.phc.pt/portal/programs/cindex.aspx>
- PMI. (2013). A guide to the project management body of knowledge (PMBOK Guide). *Fifth Edition*. Obtido em 20 de Fevereiro de 2014
- Porto Editora. (10 de Setembro de 2014). *Dicionário da Língua Portuguesa - com Acordo Ortográfico*. Obtido de Infopédia - Enciclopédia e Dicionários Porto Editora: <http://www.infopedia.pt/lingua-portuguesa/objetivo>
- PRIMAVERA BSS. (2014). PRIMAVERA BSS Portugal. Braga, Portugal. Obtido em 23 de Fevereiro de 2014, de <http://www.primaverabss.com/pt/Home-pt%20-%20Homepage.aspx>
- Ravi, N., Mala, T., Srinivasan, M. K., & Sarukesi, K. (2013). Design and Implementation of VOD (Video on Demand) SaaS Framework for Android Platform on Cloud Environment. *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on* (pp. 171 - 176). Milan: IEEE.
- Rhiannon Williams. (3 de Setembro de 2014). *Apple iOS: a brief history*. Obtido de The Telegraph: <http://www.telegraph.co.uk/technology/apple/11068420/Apple-iOS-a-brief-history.html>
- Ronkainen, J., Eskeli, J., Urhema, T., & Koskela-Huotari, K. (2013). Experiences on Mobile Cross-Platform Application Development Using PhoneGap. *ICSEA 2013 : The Eighth International Conference on Software Engineering Advances*, (pp. 146-151). Venice, Italy.
- Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2014). Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges. *Communications Surveys & Tutorials, IEEE*, 16(1), 369 - 392.
- Schwaber, K., & Sutherland, J. (2011). *O decisivo guia Scrum: As regras do jogo*. (C. Oliveira, Trad.) Scrum Alliance.

- Schwaber, K., & Sutherland, J. (2013). *The Definitive Guide to Scrum: The Rules of the Game*. Scrum Alliance.
- Scrum Alliance. (02 de Janeiro de 2014). *Scrum Alliance*. Obtido de Scrum Alliance: <https://www.scrumalliance.org/>
- Sharif, R., Gull, S., & Nazir, M. (2013). A Model for Web developer to overcome the Cross Platform Dependency in Mobile Technologies. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, 2(1), 11-18.
- Sivamani, S., Kwak, K., & Cho, Y. (2014). A Design of Web-based Services Using RESTful API for Vertical Farm. *Life Science Journal 2014*, 763-767.
- Snadberg, R., & Rollins, M. (2013). *The Business of Android Apps Development Making and Marketing Apps that Succeed on Google Play, Amazon Appstore and More*. Nova Iorque: Apress.
- Tidwell, D., Snell, J., & Kulchenko, P. (2001). *Programming Web Services With SOAP* (First Edition ed.). O'Reilly.
- Upadhyaya, B., Zou, Y., Xiao, H., Ng, J., & Lau, A. (2011). Migration of SOAP-based Services to RESTful Services. *Web Systems Evolution (WSE), 2011 13th IEEE International Symposium on* (pp. 105 - 114). Williamsburg: IEEE.
- Vanguarda. (2014). Vanguarda Soluções de Gestão. Obtido em 26 de Fevereiro de 2014, de <http://www.vanguarda.com/>
- Verborgh, R., Steiner, T., Deursen, D. V., Roo, J. D., Walle, R. V., & Vallés, J. G. (2013). Capturing the functionality of Web services with functional descriptions. *Multimedia Tools and Applications*, 64(2), 365-387.
- W3C. (1 de Junho de 2014). *W3C XML Schema Definition Language (XSD)*. Obtido de W3C - World Wide Web Consortium: <http://www.w3.org/TR/xmlschema11-1/#intro1.1>
- W3Schools. (12 de Agosto de 2014). *Introduction to Web Services*. Obtido de w3schools.com: [http://www.w3schools.com/webservices/ws\\_intro.asp](http://www.w3schools.com/webservices/ws_intro.asp)
- W3schools. (1 de Fevereiro de 2014). *Web Services Tutorial*. Obtido de W3Schools.com: <http://www.w3schools.com/webservices/>
- Wang, R., Xing, L., Wang, X., & Chen, S. (2013). Unauthorized Origin Crossing on Mobile Platforms: Threats and Mitigation. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 635-646). Berlin, Alemanha: ACM CCS.
- Wasserman, A. I. (2010). Software Engineering Issues for Mobile Application Development. *FSE/SDP workshop on Future of software engineering research*, 397-400. doi:10.1145/1882362.1882443
- White, J. (2013). Going native (or not): Five questions to ask mobile application developers. *Australasian Medical Journal*, 7-14.
- WS-I. (1 de Março de 2014). *Profiles For Basic Web Services*. Obtido de WS-I - Web Services Interoperability Organization: <http://www.ws-i.org/Docs/Brochures/WS-I%20Profiles%20for%20Basic%20Web%20Services.pdf>

- Xu, H., & Yang, J. (2012). Do M-Commerce User's Expectations Reflect Reality. *International Journal of Electronic Business Management*, 10, 322-321.
- Xu, M., Yin, X., & Rong, J. (2013). Researchment and Realization Based on Android Database Application Technology. *Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA-13)* (pp. 442-445). Paris, França: Atlantis Press.
- Yoon, H.-J. (Abril de 2012). A Study on the Performance of Android Platform. *International Journal on Computer Science and Engineering*(4), 532-537.
- Zahra, S., Khalid, A., & Javed, A. (Maio de 2013). An Efficient and Effective New Generation Objective Quality Model for Mobile Applications. (MECS, Ed.) *International Journal of Modern Education and Computer Science (IJMECS)*, 5(4), 36-42.