



Júlio Dinis Lopes de Barros

Aplicação Móvel Android de Apoio a
Gestão de Empréstimos e Reservas das
Bibliotecas da Universidade do Minho

Universidade do Minho
Escola de Engenharia





Universidade do Minho
Escola de Engenharia

Júlio Dinis Lopes de Barros

Aplicação Móvel Android de Apoio a
Gestão de Empréstimos e Reservas das
Bibliotecas da Universidade do Minho

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia de Telecomunicações e Informática

Trabalho efectuado sob a orientação do
Professor Doutor Ricardo J. Machado

DECLARAÇÃO

Nome: Júlio Dinis Lopes de Barros

Endereço eletrónico: juliolopesb@gmail.com

Telefone: 919577227

Número do Bilhete de Identidade: 77Z30Z499

Título dissertação: Aplicação Móvel Android de Apoio a Gestão de Empréstimos e Reservas das Bibliotecas da Universidade do Minho

Orientador: Professor Doutor Ricardo J. Machado

Ano de conclusão: 2015

Designação do Mestrado: Ciclo de Estudos Integrados conducentes ao Grau de Mestre em Engenharia de Telecomunicações e Informática

Área de Especialização: Engenharia de Telecomunicações e Informática

Escola: Escola de Engenharia da Universidade do Minho

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

Universidade do Minho, ___/___/_____

Assinatura: _____

*“In the great battles of life, the first step to victory
is the desire to win”*

Mohandas Karamchand Gandhi

Agradecimentos

Findo esta etapa muito importante da minha vida, gostaria de agradecer a todas as pessoas que de alguma forma contribuíram para a realização desta dissertação, às quais transmito os meus mais sinceros agradecimentos.

Primeiramente de forma particular, gostaria de agradecer ao Professor Doutor Ricardo J. Machado pela disponibilidade, a dedicação e a ajuda prestada durante a realização deste trabalho, visto que sem a sua contribuição não seria possível, um muito obrigado.

Um agradecimento muito especial aos meus colegas e amigos que me acompanharam ao longo desta caminhada académica, e principalmente ajudaram na minha integração a esta nova realidade. Também uma palavra de agradecimento ao Nuno Fernandes pela ajuda, disponibilidade e paciência e aos demais funcionários da Biblioteca da Universidade do Minho em Guimarães. Um obrigado a chefe da divisão de informação dos Serviços de Documentação da Universidade do Minho, Dr^a. Daniela Castro Ramalho pela oportunidade de trabalhar com os SDUM no desenvolvimento da aplicação. Não obstante, um obrigado especial ao Samuel Valente pela disponibilidade e ajuda.

A um nível mais pessoal, um profundo e sincero obrigado a minha família, em especial aos meus pais Venâncio Barros e Maria Barros e aos meus irmãos pela confiança e apoio incondicional durante todo o meu percurso académico. Também não podia deixar de agradecer a Elisabete Antunes, Edney Ramos e Péricles Hungria pela amizade, motivação e confiança, que fazem deles pessoas muito importantes na minha vida.

Para terminar, gostaria de destacar três pessoas com que partilhei muitos momentos e que foram excelentes companheiros ao longo da minha vida, Mário Évora, Elisa Furtado e Kleyder Mariano. Obrigado pelo vosso apoio e sobretudo pela amizade.

Resumo

Os dispositivos móveis, em particular os *tablets* e *smartphones*, alcançaram uma enorme popularidade ao longo dos últimos anos devido à sua grande versatilidade e multifuncionalidade, conquistando deste modo, meritoriamente um espaço de destaque no nosso dia-a-dia, tanto a nível pessoal como profissional. Neste contexto, os utentes das bibliotecas da Universidade do Minho não são uma exceção, e os SDUM (Serviços de Documentação da Universidade do Minho) no cumprimento da sua missão, definiu como uma das linhas gerais proporcionar aos utentes uma melhor qualidade de assistência, com o desenvolvimento de uma aplicação móvel de gestão de empréstimos e reservas de publicações em posse.

Contudo, existe um ainda um grande entrave no mercado do desenvolvimento de aplicações para dispositivos móveis, devido à sua fragmentação em termos de plataformas móveis utilizadas (iOs, Android, Windows Phone, etc.). Esta diversificação exige um maior esforço no desenvolvimento das aplicações, de modo que obriga o desenvolvimento das mesmas para cada plataforma móvel em particular. É neste sentido que as abordagens de desenvolvimento multiplataforma ganharam relevância, permitindo o desenvolvimento de aplicações para várias plataformas a partir de um único código fonte.

O principal objetivo desta dissertação é desenvolver uma aplicação de apoio a gestão de empréstimos e reservas nas bibliotecas da Universidade do Minho. Os objetivos intercalados são: realização de estudos sobre as abordagens e ferramentas de desenvolvimento multiplataforma, adotar métodos de engenharia de requisitos e conceção, implementação e teste da solução final (fundamental no processo de correção de falhas, de modo que a permitir uma solução final com maior qualidade). A primeira fase do modelo de processo de engenharia de requisitos consiste no levantamento/definição e priorização de requisitos, que tem como objetivo conhecer as técnicas de levantamento, assim como identificar e aplicar as que melhor se adequam a este projeto. Após a execução da fase de Análise e negociação, efetuou-se a documentação dos requisitos a um nível de detalhe apropriado, como consta no Anexo B – Documento de Especificação de Requisitos.

Palavras-chave: dispositivos móveis, *smartphones*, *tablets*, aplicações móveis, HTML5, plataformas móveis, desenvolvimento multiplataforma

Abstract

Mobile devices, especially smartphones and tablets, achieved an enormous popularity over the past few years, due to their great versatility and multi-functionality, so gaining a prominent space in our daily lives, both personally and professionally. In this context, the users of the libraries of University of Minho are not an exception, and the SDUM (Documentation services of the University of Minho) in the fulfilment of its mission, defined as one of the general lines provide users a better quality, assistance with the development of a mobile application loan management and booking of publications in possession.

However, there is still a big obstacle in developing applications for mobile devices, because of its fragmentation in terms of mobile platforms used (iOs, Android, Windows Phone, etc.). This diversification requires a greater effort in the development of applications, since it requires the development of same for each mobile platform in particular. It is in this sense that the cross-platform development approaches have gained importance, enabling the development of application for multiple platforms from a single source code.

The main objective of this dissertation aims at the development of a mobile application to support for the management of loans and reserves of publications in libraries of the University of Minho. The secondary objectives are: make a study of development tools and cross-platform approaches, adopt methods of requirements engineering, and implementation, conception and test of the final solution (fundamental in fault correction process, thereby to permit a higher quality of final solution). The first phase of this model of process of requirements engineering consists of the survey/definition and prioritization of requirements, which aims to meet the lifting techniques, as well as identify and apply the best suited to this project. After the execution of Analysis and negotiation phase, made-if the documentation of the requirements to a level of detail appropriate, as stated in the Anexo B – Documento de Especificação de Requisitos.

Keywords: mobile devices, smartphones, tablets, mobile applications, HTML5, mobile operating system, cross-platform development

Índice

Agradecimentos	v
Resumo	vii
Abstract	ix
Índice	xi
Lista de Figuras	xii
Lista de Tabelas	xiii
Lista de Siglas e Acrónimos	xv
1. Introdução	1
1.1. Enquadramento e Motivação	1
1.2. Descrição do Problema	2
1.3. Objetivos	3
1.4. Metodologia de Investigação e Desenvolvimento	3
1.5. Estrutura da Dissertação	6
2. Revisão da Literatura	9
2.1. Introdução	9
2.2. Dispositivos e Plataformas Móveis.....	11
2.3. Metodologias de desenvolvimento de aplicações móveis	23
2.4. Definição das tecnologias - Frameworks Multiplataforma.....	41
2.5. Conclusão.....	62
3. Requisitos e Arquitetura	63
3.1. Introdução	63
3.2. Engenharia de Requisitos.....	64
3.3. Caracterização genérica dos SDUM	74
3.4. Arquitetura da Solução	79
3.5. Conclusão.....	84
4. Implementação e testes	85
4.1. Introdução	85
4.2. Seleção da tecnologia da solução.....	85
4.3. Implementação da solução SDUM	87
4.4. Testes	106
4.5. Conclusão.....	114
5. Conclusão	117
5.1. Síntese do Trabalho	117

5.2. Trabalho Futuro	119
Referências	121
Anexos.....	127
Anexo A – Estrutura do Volere Requirements Specification Template	127
Anexo B – Documento de Especificação de Requisitos	137
Anexo C – Testes de Usabilidade	205
Anexo D – Lista de Serviços disponibilizados pelo Aleph X-Server	209

Lista de Figuras

Figura 1.1 - Ciclo da Investigação-Ação [3], [6].....	6
Figura 2.1 - Smartphone OS Market Share, Q1 2015 IDC [8].....	12
Figura 2.2 - Smartphone OS Market Share, Q1 2015 [8].....	12
Figura 2.3 - Processo do Dalvik Virtual Machine [14]	16
Figura 2.4 - Arquitetura do Android [11], [14]	17
Figura 2.5 - Arquitetura iOS [11], [20]	19
Figura 2.6 - Arquitetura do Windows Phone [11]	21
Figura 2.7 – Arquitetura do BlackBerry 10 OS [21]	22
Figura 2.8 – Arquitetura de aplicações WEB [27], [25].....	25
Figura 2.9 - Arquitetura de aplicações Híbridas [27]	27
Figura 2.10- Arquitetura de aplicações interpretadas [27]	29
Figura 2.11 - Arquitetura de aplicações Cross-platform [27].....	31
Figura 2.12 - Desenvolvimento Multiplataforma vs Nativa [38]	41
Figura 2.13 - Utilização das ferramentas Multiplataformas em 2014 [39]	42
Figura 2.14 - Esquema completo da arquitetura do Titanium [40]	43
Figura 2.15 - Interface da arquitetura Rhodes entre o Smartphone e os componentes do Rhodes [1]	45
Figura 2.16 - Building PhoneGap App [45]	46
Figura 2.17 - Camadas de Interface da arquitetura do PhoneGap [1]	47
Figura 2.18 - Esquema completo da arquitetura do PhoneGap [1]	48
Figura 2.19 - Principais critérios de seleção de frameworks [39]	50
Figura 2.20 - Arquitetura geral do desenvolvimento de aplicações multiplataforma [42]	57
Figura 2.21 - Padrão MVC [53].....	59
Figura 2.22 - Estrutura de uma mensagem SOAP [55], [58]	61
Figura 3.1 - Riscos no contexto de desenvolvimento de <i>software</i> [59].....	63
Figura 3.2 - Modelo do processo de engenharia de requisitos [60].....	65
Figura 3.3 - Domínio da aplicação (Levantamento de requisitos) [61].....	67
Figura 3.4 - Processo de Análise e negociação [61].....	70
Figura 3.5 - Requirements Shell [62]	72
Figura 3.6 - Organograma do SDUM [63]	74
Figura 3.7 - Arquitetura do sistema final.....	79
Figura 3.8 - Aleph WEB Service [64]	81

Figura 3.9 - Arquitetura multi-tier dos produtos do Ex Libris [64].....	83
Figura 3.10 - Interação com o X-Server [64]	84
Figura 4.1 - PhoneGap + Sencha Touch [47]	87
Figura 4.2 - Padrão MVC do Sencha Touch [47].....	88
Figura 4.3 - Estrutura dos componentes	89
Figura 4.4 - Utilização do SQLite Plugin	92
Figura 4.5 - Permissões	93
Figura 4.6 - Definição do idioma	100
Figura 4.7 - Autenticação no SDUMApp.....	101
Figura 4.8 - Interface da Pesquisa de publicações.....	102
Figura 4.9 - Lista de resultados da pesquisa de publicações	102
Figura 4.10 - Reserva de publicações	103
Figura 4.11 - Renovação de publicações na Lista de Empréstimos	104
Figura 4.12 - Lista de Empréstimos	104
Figura 4.13 - Calendário do dispositivo móvel	105
Figura 4.14 - Calendário de Eventos	105
Figura 4.15 – Pesquisa de publicações	106
Figura 4.16 - Lista de Preferências	106

Lista de Tabelas

Tabela 2.1 - Desenvolvimento de Aplicações Nativas [20]	24
Tabela 2.2 - Metodologia Multiplataformas e frameworks adotados [24]	32
Tabela 2.3 - Tipos de Aplicação e abordagem preferido [24]	35
Tabela 2.4 - Análise comparativa das abordagens de desenvolvimento multiplataforma [25], [30].....	36
Tabela 2.5 - Comparação entre a abordagem Nativa e Multiplataforma [18], [20], [31]40	
Tabela 2.6 - Comparação entre a compatibilidade de plataformas móveis e o suporte os do ambiente de desenvolvimento [1], [30], [36]	50
Tabela 2.7 - Comparação das características de desenvolvimento [1], [30], [36]	51
Tabela 2.8 - Comparação das principais APIs suportadas [1], [30], [36]	52
Tabela 2.9 - Comparação das características gerais [1], [30].....	53
Tabela 2.10 - Métrica da utilização da CPU [35]	54
Tabela 2.11 - Métrica da utilização da memória [35].....	56
Tabela 2.12 - Métrica do consumo de energia [35]	57
Tabela 3.1 - Processos de Engenharia de Requisitos [23].....	65
Tabela 3.2 - Lista dos serviços do Aleph [46].....	209
Tabela 4.1 – Acesso aos componentes de <i>Hardware</i>	89
Tabela 4.2 – Login como operador.....	94
Tabela 4.3 – Autenticação do utente	94

Tabela 4.4 – Pesquisa de publicações nos catálogos das bibliotecas da Universidade do Minho	96
Tabela 4.5 – Apresentação dos dados das publicações pesquisadas	96
Tabela 4.6 – Renovação de publicações	97
Tabela 4.7 – Reserva de publicações	98
Tabela 4.8 – Verificação do Empréstimo da publicação	98
Tabela 4.9 – Teste da funcionalidade	107
Tabela 4.10 – Teste de desempenho	108
Tabela 4.11 – Teste de interoperabilidade.....	109

Lista de Siglas e Acrónimos

AJAX – Asynchronous JavaScript and XML

API - Application Programming Interface

CPU – Central Processing Unit

CSS - Cascading Style Sheets

DVM – Dalvik Virtual Machine

GPS - Global Positioning System

GUI - Graphic User Interface

HTML - HyperText Markup Language

IEEE – Institute of Electrical and Electronic Engineers

MVC – Model-View-Controller

NFC - Near Field Communication

OS – Operating System

OTS - off-the-shelf

PDA – Personal Digital Assistant

PSS – Proporcional set size

RDBMS – Relational Database Management System

SDUM – Serviços de Documentação da Universidade do Minho

SOAP – Simple Object Access Protocol

SQL – Structured Query Language

SUS – System Usability Scale

TTS – Text to Speech

UI – User Interface

UDDI – Universal Description, Discovery, and Integration

URL - Uniform Resource Location

USS – Unique set size

WEB – World Wide Web

WSDL – Web Service Description Language

XML - Extensible Markup Language

1. Introdução

1.1. Enquadramento e Motivação

Atualmente, os *smartphones* estão a mudar a forma como as pessoas interagem entre si e com o mundo, devido à sua grande utilidade e às inúmeras aplicações disponíveis para estes aparelhos. O desenvolvimento destas aplicações tem ganho cada vez mais adeptos, formando-se até grandes comunidades de partilha de ideias. A utilidade das mesmas no nosso dia-a-dia é inquestionável, tanto como no contexto social como profissional. Com o tempo, a rigorosidade dos utilizadores é de tal maneira que estão sempre a espera de melhores aplicações, mais versáteis e em pouco tempo [1].

Desenvolver aplicações móveis de alta performance neste mercado competitivo torna-se um grande desafio, de modo a poder corresponder às expectativas dos clientes.

No entanto, existe ainda um grande entrave no mercado do desenvolvimento de aplicações para *smartphones* devido ao fato dos mesmos se encontrarem fragmentados em termos de plataformas utilizadas (iOs, Android, Windows Phone, etc.). Esta diversificação exige um maior esforço no desenvolvimento das aplicações, de modo que haja portabilidade para as várias plataformas, e abranger em grande escala o público-alvo. Para que estas dificuldades fossem colmatas, o desenvolvimento de aplicações baseadas em tecnologias WEB ganhou relevância [2].

Esta nova vertente é impulsionada principalmente pela implementação do HTML5 que, com as suas variadas funcionalidades, permite maior integração das aplicações com os dispositivos.

No contexto das bibliotecas, de forma a proporcionar uma melhor qualidade de assistência, o desenvolvimento desta aplicação vem facilitar a vida aos utilizadores, proporcionando uma melhor gestão das requisições e reservas de publicações efetuadas. É importante analisar que tarefas e como estas tarefas são realizadas nas bibliotecas da Universidade do Minho. Um produto sustentado na observação cuidada destes aspetos leva a que a utilidade das suas funcionalidades esteja acautelada.

1.2. Descrição do Problema

O problema proposto para a realização desta dissertação consiste no estudo de abordagens e tecnologias de desenvolvimento de aplicações multiplataformas e o desenvolvimento de uma aplicação útil, de objetivos práticos e funcional, de apoio a gestão de empréstimos e renovações nas bibliotecas da Universidade do Minho de acordo com o estudo proposto. É inquestionável a utilidade das aplicações móveis no dia-a-dia, tanto em contexto social como profissional. O motivo para o aumento desta utilidade deriva do crescimento acentuado e contínuo da venda de dispositivos móveis. Uma simples atividade como ler um livro ou um jornal, passou a ser mais cômodo a partir dos dispositivos móveis, de modo que tornámo-nos cada vez mais dependentes das funcionalidades que estas tecnologias nos proporcionam.

De uma forma mais pormenorizada, para esta dissertação pretende-se desenvolver uma aplicação móvel, que possa incorporar todas as funcionalidades existentes no *Website* dos SDUM, com a particularidade de incluir algumas funcionalidades novas (como por exemplo, a sincronização com o calendário dos dispositivos móveis, possibilitando assim a receção de notificações, alertando a data de entrega de publicações). Estas funcionalidades permitem que os utentes tenham melhor gestão das suas requisições e reservas. Esta aplicação vem proporcionar uma melhor assistência, e também facilitar a vida aos utentes das bibliotecas da Universidade do Minho.

As principais funcionalidades a implementar na solução SDUM são:

- Pesquisa de publicações nos catálogos das bibliotecas da Universidade do Minho;
- Reserva de publicações de acordo com a pesquisa efetuada;
- Renovação de publicações a partir da lista de publicações emprestados;
- Sincronização das publicações emprestadas com o calendário do dispositivo móvel.

1.3. Objetivos

O principal objetivo desta dissertação é desenvolver uma aplicação de apoio à gestão de empréstimos e renovações nas Bibliotecas da Universidade do Minho baseado na abordagem de desenvolvimento multiplataforma. Numa primeira fase, o objetivo é gerar e disponibilizar a aplicação para o sistema operativo Android.

Para esta dissertação, foi necessário ultrapassar um conjunto de desafios, nomeadamente o estudo das abordagens de desenvolvimento e das ferramentas de desenvolvimento de aplicações móveis, de modo que resultado final chegasse a um bom porto. Desta forma, os objetivos intercalados para a realização desta dissertação são:

1. Adotar as abordagens e tecnologias de desenvolvimentos multiplataformas de acordo com a análise comparativa e os critérios de seleção da metodologia e da tecnologia;
2. Adotar métodos de engenharia de requisitos capazes de suportar a descoberta, análise, documentação e verificação dos requisitos da solução SDUM e as suas restrições;
3. Conceção, implementação e teste a aplicação móvel da solução SDUM.

1.4. Metodologia de Investigação e Desenvolvimento

A metodologia de investigação e desenvolvimento que vai ser utilizada será a Investigação-Ação, que consiste em provar se a solução resolve o problema. A Investigação-Ação é um método de investigação criado desde do século XX, para ser utilizado nas ciências sociais e médicas. Perto do final da década de 1990, começou a ganhar popularidade nas investigações académicas em sistemas de informação. Este método produz resultados de investigação altamente relevantes, por basear-se em ações práticas, destinado em resolver um problema imediato [3].

1.4.1. Metodologia Investigação-Ação

A metodologia Investigação-Ação (*Action Research*), é conhecido por imensos nomes, inclusive *Participatory Research*, *Collaborative Inquiry*, *Emancipatory Research*, *Action Learning* e *Contextual Action Research*, mas todos estes são variantes sobre o tema. Simplificando,

Investigação-Ação é “aprender fazendo”, identifica-se um problema, faz-se algo para resolvê-lo, verifica-se se o esforço foi bem-sucedido, e se não ficar satisfeito, tente novamente [4].

A Investigação-Ação pode ser descrita como uma família de metodologias de investigação que incluem Ação (ou mudança) e investigação (ou compreensão) simultâneas, utilizando um processo cíclico ou em espiral, que alterna entre Ação e reflexão crítica.

Disponibiliza a possibilidade de intervenção na entidade sob investigação e a análise dos resultados e permite uma aproximação aberta ao campo de investigação sendo, desta forma, possível capturar informação que não pode, frequentemente, ser predeterminada. Esta estratégia leva o investigador a participar ativamente em qualquer que seja a mudança num sistema.

Ao longo do tempo, a constituição dos ciclos tem evoluído através de alterações que variam com os autores. Para **Susman** a composição mais consensual é a do ciclo de cinco fases. Segundo **Kock**, os passos envolvidos na pesquisa por Ação são: **diagnóstico, planeamento de Ação, execução de Ação, avaliação e aprendizagem específica.**

A primeira fase é diagnosticar, seguida de planear a Ação, tomar a Ação, avaliar e aprender/documentar. A partir da fase de aprender/documentar pode reiniciar-se um novo ciclo voltando-se à fase de diagnóstico [3], [5], [6]. Este ciclo encontra-se representado na Figura 1.1 a seguir apresentada.

O ciclo “Investigação-Ação” repete-se em várias iterações de toda a sequência de fases do método.

- a) **Diagnóstico** – numa forma geral, é a fase em que o problema é identificado e definido. Tem por principal objetivo, identificar uma oportunidade de solução ou melhoria num potencial problema previamente identificado. Durante esta fase importa ter uma visão global na tentativa de compreensão do problema como um todo integrado. Desta fase resultam, normalmente, assunções sobre a natureza e domínio do problema.
- b) **Planeamento de Ação** – soluções alternativas e as ações são consideradas com base na teoria. Consiste em planear o conjunto de ações a tomar na investigação e identificar a abordagem e objetivos da intervenção. No planeamento são tidas em conta as diferentes alternativas de ações a tomar, procede-se à seleção da alternativa mais adequada.

- c) **Execução de Ação** – Esta terceira fase, consiste em pôr em prática as ações consideradas no Planeamento de Ação, ou seja, realizar as ações que foram selecionadas na fase de planeamento.
- d) **Avaliação** – fase onde as consequências e o impacto da ação ou intervenção é estudada ou observada. Tem por objetivo verificar se as ações efetuadas tiveram o efeito esperado e se serviram para resolver os problemas iniciais. É muito útil incluir nesta avaliação uma análise crítica que avalie em que medida as ações tomadas foram as únicas responsáveis pelos efeitos obtidos. Tal despiste justifica-se pela possibilidade da existência de interferências originadas por ações intrínsecas ao ambiente em estudo.
- e) **Aprendizagem específica** – nesta fase são documentados os resultados da avaliação. A aprendizagem é alimentado em um novo ciclo que reinicia em uma nova fase de diagnóstico [5], [6].

Cada vez que é repetido um ciclo o conjunto de melhorias introduzidas, com vista à obtenção dos objetivos inicialmente enunciados, tende a estabilizar. Quando a estabilização for considerada significativa pode-se dar por concluída a intervenção ou partir para a redefinição de novos objetivos e recomeçar um novo ciclo. A maior virtude deste modelo reside em alertar para a necessidade de se cumprir um determinado conjunto de etapas e em garantir que, terminada uma iteração do ciclo, é preparada uma nova iteração que aproveite o conhecimento adquirido na iteração anterior [6].



Figura 1.1 - Ciclo da Investigação-Ação [3], [6]

1.5. Estrutura da Dissertação

A presente dissertação encontra-se dividida em cinco capítulos. Este capítulo apresenta uma introdução inicial ao tema da dissertação, bem como as principais motivações para o desenvolvimento do trabalho, os objetivos propostos e a metodologia de investigação e desenvolvimento adotada.

No **segundo capítulo** é apresentado o estado da arte (revisão da literatura) alusivo às abordagens de desenvolvimento de aplicações. É feita uma pequena abordagem aos dispositivos móveis e às plataformas móveis e a cota de mercado das principais plataformas móveis (sistemas operativos móveis). Efetuou-se ainda comparações entre as duas abordagens de desenvolvimento, bem como o critério de seleção e a análise comparativa entre as mesmas. Por fim, elaborou-se um estudo das ferramentas de desenvolvimento de aplicações multiplataformas e uma breve análise comparativa entre as mesmas.

O **terceiro capítulo** diz respeito ao processo de engenharia de requisitos, à arquitetura da solução e ao estudo da organização no qual a aplicação se destina (**Serviços de Documentação da Universidade do Minho** e também a área de ação da mesma dentro da Universidade do Minho, bem como todos os processos inerentes a esta mesma organização (Processo de Empréstimo, Reserva e Renovação de publicações). Nesse sentido, primeiramente é efetuado um estudo sobre

os Serviços de Documentação da Universidade do Minho, a área de intervenção, o organograma da organização, bem como a especificação dos serviços prestados, tais como o empréstimo e reserva de publicações. De seguida, é apresentado o modelo de processo de engenharia de requisitos e a especificação do produto, que inclui os requisitos funcionais e não funcionais em que a aplicação se deve sustentar. Por fim, é apresentada a arquitetura da solução final, que visa ilustrar todos os componentes do sistema.

No **quarto capítulo** é descrito a implementação de todo o sistema de uma forma pormenorizada. É explicado a tecnologia adotada, bem como o processo de aquisição e processamento dos dados. Por fim, são efetuados testes à aplicação e a posterior observação dos resultados, de modo a tirar ilações quanto à sua utilidade e ao modo como responde às exigências diárias.

O **quinto e último capítulo** expõe as conclusões obtidas e verificação dos objetivos previamente definidos. Adicionalmente, são apresentadas algumas sugestões de trabalho futuro.

2. Revisão da Literatura

2.1. Introdução

O mercado de dispositivos móveis explodiu nos últimos anos devido a sua popularidade e massificação, muito pela culpa dos *smartphones* e *tablets*, e surgiu um novo mercado para aplicação móveis para estes dispositivos. Hoje em dia, esse mercado já esta relativamente consolidado, bem como a nossa dependência por estas aplicações no nosso dia-a-dia. A prova disso é o estudo realizado pela **Flurry Analytics**, empresa de análises, que revelou um aumento de 115% no uso de aplicações em 2013.

Ao longo deste capítulo são abordados vários temas relacionados com o desenvolvimento e modelação de aplicações móveis. Inicialmente é efetuado uma breve introdução sobre os principais dispositivos e plataformas móveis, bem como a análise da cota de mercado dessas mesmas plataformas. Segue-se um estudo sobre as diferentes abordagens de desenvolvimento de aplicações móveis: desenvolvimento de aplicações nativas e multiplataforma, com maior destaque para o desenvolvimento de aplicações móveis multiplataforma, que se encontra dividida em 4 abordagens diferentes: Abordagem híbrida, WEB, compilação-cruzada e interpretada. Segue-se um estudo dos critérios de seleção e uma análise comparativa entre abordagens de desenvolvimento nativa e multiplataforma. De seguida um estudo sobre as ferramentas (*frameworks*) de desenvolvimento de aplicações multiplataformas, bem como uma breve análise comparativa entre as mesmas. Finaliza-se este capítulo com o estudo da arquitetura geral de desenvolvimento de aplicações móveis multiplataforma.

2.1.1. Trabalho relacionado

Esta subsecção consiste em apresentar algumas aplicações de apoio às bibliotecas de diferentes universidades. De modo a desenvolver a aplicação de apoio às bibliotecas da Universidade do Minho, surgiu a necessidade de conhecer e verificar possíveis funcionalidades a serem desenvolvidas a partir de aplicações já desenvolvidas para bibliotecas universitárias. Serão apresentadas quatro aplicações de apoio às bibliotecas: **Bibliotecas USP**, **Library Groningen University**, **Biblio USAL** e **Biblioteca Llanos** a seguir descritos nas seguintes tabelas: Tabela 2.1, Tabela 2.2, Tabela 2.3 e Tabela 2.4.

Tabela 2.1 – Aplicação das Bibliotecas da Universidade de São Paulo

Nome	Bibliotecas USP
Descrição	Aplicação das Bibliotecas da Universidade de São Paulo, Brasil
Funcionalidades	Pesquisa de publicações, Lista de Desejos, Lista de Referências, Lista de sugestões, Itens emprestados, Histórico de empréstimos, Lista de reservas, Renovação de publicações, Exportação de listas por <i>email</i> ou Google Drive ou DropBox Enviar email, Localização das Bibliotecas e Horários de funcionamento, Dados Pessoais, Ler códigos de barras de publicações para verificar os detalhes do mesmo.
Sistema Operativo	Android e iOS.

Tabela 2.2 – Aplicação da Biblioteca da Universidade de Groningen

Nome	Library Groningen University
Descrição	Aplicação da Biblioteca da Universidade de Groningen, Holanda
Funcionalidades	Pesquisas no catálogo, Renovação de publicações, Horário de funcionamento e localização da biblioteca, <i>Workstations</i> disponíveis, Lista de Empréstimos, Lista de Reservas, Notificações de data de entrega de publicações e de levantamento de publicações após a sua reserva.
Sistema Operativo	Android e iOS.

Tabela 2.3 – Aplicação das Bibliotecas da Universidade de Salamanca

Nome	Biblio USAL
Descrição	Aplicação das Bibliotecas da Universidade de Salamanca, Espanha
Funcionalidades	Localização das bibliotecas e Horários de funcionamento, Pesquisa de publicações no catálogo, agenda e novidades sobre as bibliotecas, Reserva e renovação de publicações, acesso à área pessoa a partir da camera para ler o código de barras do cartão de universitário, Lista de preferências, integração com as redes sociais (twitter e facebook).
Sistema Operativo	Android e iOS.

Tabela 2.4 – Aplicação da Biblioteca da Universidade de Llanos

Nome	Biblioteca Unillanos
Descrição	Aplicação das Bibliotecas da Universidade de Llanos, Colômbia
Funcionalidades	Localização das bibliotecas e Horários de funcionamento, Pesquisa de publicações no catálogo, Reserva e renovação de publicações, Lista de emprestados, Histórico de empréstimos, Lista de reservas, Renovação de publicações.
Sistema Operativo	Android e iOS.

2.2. Dispositivos e Plataformas Móveis

“Mobility can be defined as the capability of being able to move or be moved easily. In the context of mobile computing, mobility pertains to people’s use of portable and functionality powerful mobile devices that offer the ability to perform a set of application functions untethered, while also being able to connect to, obtain data from, and provide data to other users, applications, and systems” [7].

Os dispositivos móveis são caracterizados como dispositivos de pequenas dimensões, inteligentes e com grande destreza de computação. Estes dispositivos estão a tornar-se cada vez mais, numa ferramenta muito importante no nosso dia-a-dia, tanto a nível pessoal como profissional. São dispositivos práticos, poderosos e extensíveis e a sua evolução em *Hardware* e *Software* encontra-se entrelaçada, ou seja, melhor *Hardware* permite um melhor *Software*, e melhor *Software* faz com que os utilizadores queiram mais, levando a um melhor *Hardware*. Recursos como a camera, música, GPS, Wi-Fi, acelerómetros ganharam popularidade em alguns destes dispositivos, e estes são oferecidos pelos principais sistemas operativos móveis e mais modernos como Android, iOS, Symbian, Windows Phone, BlackBerry. O sistema operativo móvel é o gestor do *Hardware* no dispositivo móvel. Dos três maiores sistemas operativos móveis, o Android tem maior cota de mercado, seguido pelo iOS e o Windows Phone pela mesma ordem, tendo este o mercado mais pequeno dos três, como mostram as figuras (Figura 2.1 e Figura 2.2) a seguir apresentadas, um estudo realizado pela IDC em Agosto de 2015.

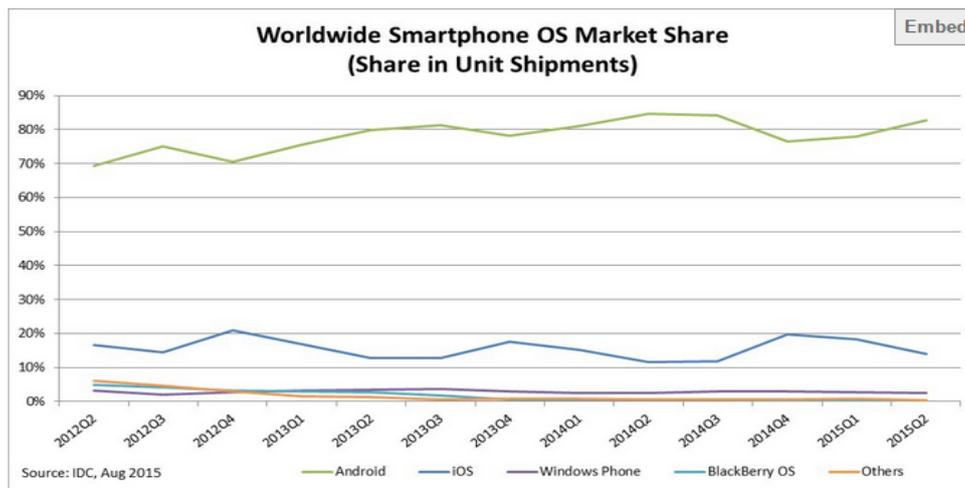


Figura 2.1 - Smartphone OS Market Share, Q1 2015 IDC [8]

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Figura 2.2 - Smartphone OS Market Share, Q1 2015 [8]

2.2.1. Dispositivos Móveis

São muitos os tipos de dispositivos móveis introduzidos desde a década de 1990, incluindo o *PDA*, *Smartphone*, *Tablet*, *Ultra-mobile PC* e *wearable computers*. No entanto, pretende-se especificar apenas três destes dispositivos móveis [9].

2.2.1.1. Personal Digital Assistant (PDA)

O PDA também conhecido por *palmtop* ou assistente de dados pessoais, é um dispositivo móvel que tem como função efetuar a gestão de dados pessoais.

Em 1984, Psion lançou o primeiro “PDA”, o **Organizer II**. Em 1991, seguido pelo Psion Serie 3, começaram a aparecer PDAs com estilos mais familiares [9]. Os PDAs atuais são dotados de grande capacidade de processamento, possuindo funções de agenda e de escritório, com possibilidade de acesso à internet, intranet com computadores e com redes informáticas através do Wi-Fi [10]. Apesar do crescimento elevado dos PDAs, estas passaram a ser consideradas obsoletas com a adoção generalizada de *smartphones*, um misto de telemóvel e de PDA, que pela sua maior portabilidade foi ganhando cada vez mais utilizadores [9].

2.2.1.2. Smartphone

O *Smartphone* é um telemóvel construído a partir um sistema operativo móvel, com mais capacidade computacional avançada e conectividade, do que o telefone.

O primeiro *Smartphone* foi a combinação do PDA com a funcionalidade do telemóvel. Algumas funções foram adicionadas em modelos posteriores, como *media players* portáteis, cameras digitais compactas *low-end*, cameras de vídeos, unidade de navegação GPS, ecrãs sensíveis ao toque de alta resolução e *browsers* para visualização de sites e páginas otimizadas para dispositivos móveis [10]. Além disso, o Wi-Fi fornece o acesso a dados de alta velocidade e banda larga móvel. Os sistemas operativos mais utilizados atualmente em *smartphones* modernos são: Android do Google, iOS da Apple e Windows Phone da Microsoft [9].

2.2.1.3. Tablet

Os *tablets* são dispositivos de dimensões maiores em relação aos telemóveis e PDAs. São tipos de dispositivos móveis integrados com uma tela de toque plana e operado principalmente pelo toque no ecrã [10]. Não tem teclado físico no dispositivo, mas utilizam o teclado virtual no ecrã. Os primeiros exemplos do conceito de *tablet* originou nos séculos XIX e XX, mas como protótipos e conceito de ideias. O primeiro *tablet* comercial baseado neste conceito apareceu no final do século XX.

A Apple lançou o iPad com sistema operativo e tecnologia *touchscreen* em 2010 e tornou-se no primeiro dispositivo móvel *tablet* a atingir fins comerciais em todo mundo. Após este sucesso, muitos outros fabricantes tem vindo a produzir os seus próprios *tablets*, como é o caso da Samsung, HTC, Motorola, RIM, Sony, Amazon, JP, Microsoft, entre outros [9].

2.2.2. Sistemas Operativos Móveis

“A mobile software platform is defined as the combination of an operating system for a collection of compatible mobile devices with a set of related software development libraries, application programming interfaces (APIs), and programming tools” [7].

Existem atualmente vários sistemas operativos para dispositivos móveis e atualmente os quatro mais utilizados são: Android, iOS, Windows Phone e BlackBerry. Além destes, ainda existem mais sistemas operativos móveis, tais como, o Symbian, RIM, Palm OS, Tizen e Bada.

2.2.2.1. Android

A plataforma móvel Android foi desenvolvido pela **Open Handset Alliance (OHA)**, liderado pelo Google e outras empresas. O **OHA** é um consórcio entre 65 empresas de *Hardware*, *Software* e telecomunicações, cujo objetivo foi desenvolver uma plataforma *standard* para os dispositivos móveis [9], [11], [12], [13], [14].

O Android é um sistema operativo móvel *open-source* baseado em *kernel* do Linux e oferece uma abordagem unificada no desenvolvimento de aplicações para dispositivos móveis, ou seja, as aplicações Android executam em dispositivos diferentes com OS Android [9], [15].

A primeira versão beta do **Android Software Development Kit (SDK)** foi lançada pelo Google em 2007, mas a sua primeira versão comercial, Android 1.0, só foi lançada em Setembro de 2008.

A plataforma Android para além de fornecer o sistema operativo móvel, incluindo o seu ambiente de desenvolvimento, também fornece uma máquina virtual (**Dalvik Virtual Machine**) para executar o código da aplicação, ou seja, as aplicações Android são executados numa instancia do Dalvik VM (máquina virtual Java) [11], [14], [16]. O **Dalvik Virtual Machine** foi redesenhado e otimizado pelo Google para os recursos de *Hardware* dos dispositivos móveis. Este pode ter várias instâncias num dispositivo e cada instância é executado separadamente em um processo Linux (as aplicações Android são executados numa instância do Dalvik VM) [16].

O Android OS consiste em 4 camadas como consta na Figura 2.4: **Application, Application Framework, Android Runtime and Libraries e Linux Kernel**.

1. Application Layer

É a camada superior da arquitetura Android e é utilizada pelas aplicações instaladas (Ex: telefone, correio eletrónico, entre outros). A maioria das aplicações são aplicações nativas como o Google Maps, Camera, *browser*, SMS, contactos, calendários [14], [17], [18].

2. Application Framework

A camada **Application Framework** permite o acesso às APIs utilizadas para criação de aplicações para Android. Nesta camada inclui classes e serviços necessários para o desenvolvimento de aplicações Android [14], [18], [19]. Os gestores (*managers*) presentes nesta camada permitem às aplicações o acesso aos dados. Estes gestores são:

- ✓ **Gestor de Atividades** – efetua a gerência e o controlo de todas as atividades, bem como a gerência do ciclo de vida das aplicações;
- ✓ **Gestor de Recursos** – permite o acesso aos *non-code resources* (ex. gráficos);
- ✓ **Gestor de Notificações** – permite a todas as aplicações a mostrarem alertas personalizadas na barra de estados;

- ✓ **Gestor de Localização** – possibilita a ativação de alertas quando um utilizador entra ou sai de uma localização geográfica em particular;
- ✓ **Gestor de *Packages*** – permite a recuperação dos dados dos *packages* (pacotes) instaladas no dispositivo;
- ✓ **Gestor do telefone** – trata das configurações de conexão à rede, bem como todas as informações dos serviços no dispositivo;
- ✓ **Gestor de Janelas** – permite a criação de *views* e *layouts* nas janelas;
- ✓ **Fornecedor de Conteúdos** – permite que uma aplicação tenha acesso aos dados de outras aplicações, ou seja, partilha de dados [14], [19].

3. Android Runtime and Libraries;

A camada **Android Runtime** é o responsável pela execução de todas as aplicações Android [19]. Embora as aplicações Android sejam desenvolvidas utilizando a linguagem Java, estas não são executadas em uma máquina virtual Java tradicional. O OS Android possui a sua máquina virtual, construída pelos engenheiros da Google de modo a obter um consumo mínimo de memória e isolamento de processos, denominado **DVM (Dalvik Virtual Machine)**. Esta máquina virtual executa os ficheiros em formato DEX (.dex) [14], [18]. A Figura 2.3 a seguir apresentado descreve tal execução.

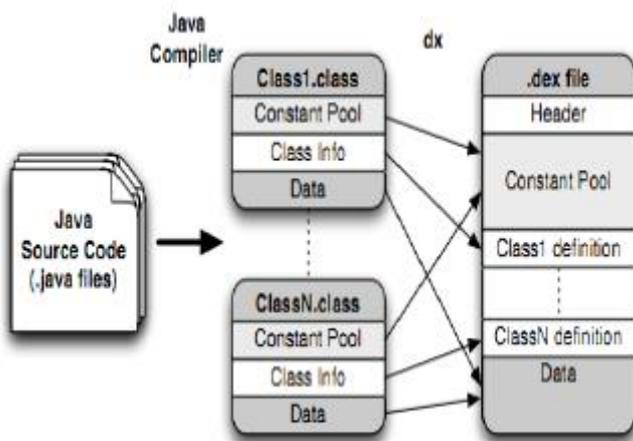


Figura 2.3 - Processo do Dalvik Virtual Machine [14]

A camada **Libraries** é composta por bibliotecas escritas em C/C++ utilizadas em vários componentes do sistema Android. A partir da camada **Application Framework**, pode-se ter acesso

a estas bibliotecas [14], [18], [19]. Existem inúmeras bibliotecas, como bibliotecas WEB para acesso aos *browsers* WEB, bibliotecas para formatos áudio e vídeo, bibliotecas 3D, SQLite, FreeType, entre outros [19].

4. Linux Kernel.

Esta camada é o núcleo da arquitetura Android. Fornece serviços como gerenciamento de memória, energia, segurança entre outros. Fornece um nível de abstração entre o *Hardware* do dispositivo e contém todos os *drivers* de *Hardware* essenciais como camera, teclado, display, e outros.

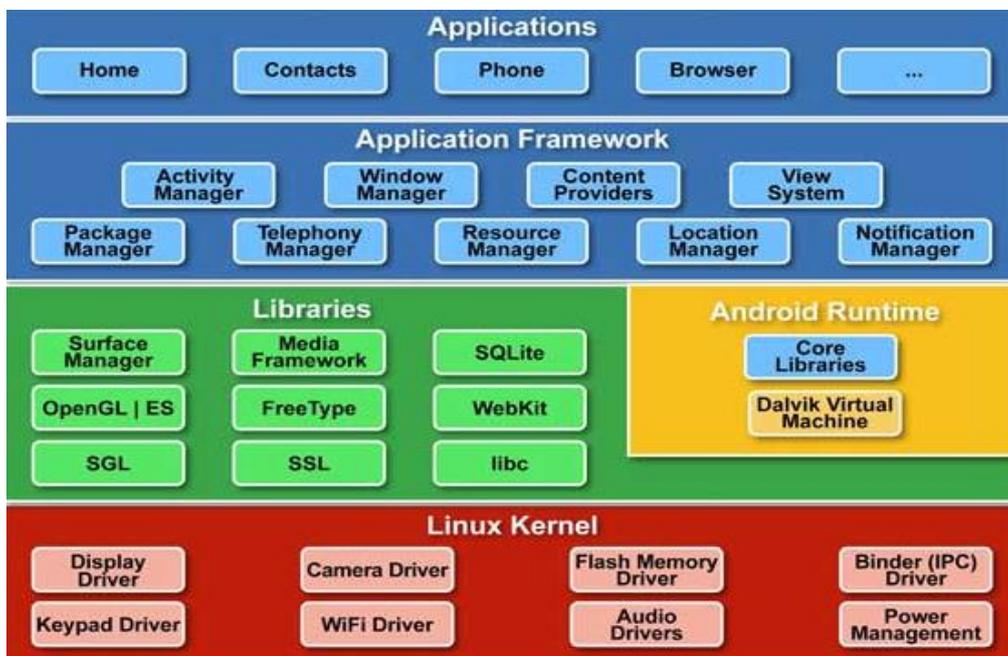


Figura 2.4 - Arquitetura do Android [11], [14]

2.2.2.2. iOS

O iOS é o sistema operativo desenvolvido pela Apple Inc., foi lançado em 2007 em vários dispositivos da Apple, como o iPhone e o iPod Touch, e depois mais tarde, estendido para iPads e Apple TV [7], [20]. Este sistema operativo gere o *Hardware* do dispositivo e fornece as tecnologias necessárias para a implementação e desenvolvimento de aplicações nativas.

As aplicações para iOs são implementadas em *Objective-C* utilizando a biblioteca *Cocoa Touch* [20]. O *Objective-C* é uma extensão da linguagem C, enquanto *Cocoa Touch* é uma coleção de classes [7], [12]. As linguagens C# e Java (utilizados para o desenvolvimento de aplicações Windows Phone e Android) são bastantes semelhantes na sintaxe, no entanto a biblioteca do *Objective-C* oferece uma alternativa diferente. O *Objective-C*, como suscita o nome, suporta programação orientada a objetos. É um superconjunto da linguagem de programação C, fornece recursos orientados a objetos e tempo de execução dinâmica. A programação orientada a objetos do *Objective-C* é baseada na passagem de mensagens para instâncias do objeto.

O *Objective-C* herda da linguagem de programação C a sintaxe, os tipos primitivos e declarações do controlo de fluxo e adiciona sintaxe para definir classes e métodos. Esta linguagem e plataforma têm melhorado continuamente ao longo dos anos, e uma mudança especialmente notável veio com a introdução do **ARC (Automatic Reference Counting)**. Esta melhoria proporcionou o gerenciamento automático da memória e fez com que a quantidade do código *boilerplate* (corresponde a códigos que podem ser reutilizados várias vezes com pouca ou nenhuma alteração) reduzisse e os vazamentos na memória geral fosse menos comum [11].

A implementação de tecnologias iOs pode ser vista como um conjunto de 4 camadas como ilustrado na Figura 2.5. As camadas inferiores contêm os serviços e tecnologias fundamentais dos quais todas as aplicações dependem, enquanto as camadas superiores fornecem serviços e tecnologias mais sofisticados. Estas camadas são:

1. Cocoa Touch

A camada **Cocoa Touch** contém os principais *frameworks* de desenvolvimento de aplicações para iOs (como por exemplo **Address Book UI Framework**, **EventKit UI Framework**, **GameKit Framework**, e outros). Esta camada também oferece uma estrutura básica à aplicação e o suporte às principais tecnologias como a multitarefa (*multitasking*), serviço de notificações *Apple push*, as entradas baseadas no toque (*Touch-based input*), notificações locais, TextKit, AirDrop entre outros serviços de alto nível do sistema [12], [20].

2. Media

Esta camada contém as tecnologias de gráficos (ex. **UIKit graphics**, **Core Animation**, etc.), áudio (ex. **Media Player framework**, **Core Audio**, etc.) e vídeo (**AVKit**, **Core Media**, etc.). Estas tecnologias foram projetadas para tornar mais fácil a implementação das aplicações multimédias [12], [20].

3. Core Services

A camada **Core Services** contém os serviços fundamentais do sistema das aplicações [12], [20]. A chave entre estes serviços são: o Core Foundation e o Foundation Framework, que definem os tipos básicos que todas as aplicações utilizam. As principais tecnologias nesta camada são: **Grand Central Dispatch**, **SQLite**, **XML Support**, **In-App Purchase**, **iCloud Storage**, entre outros [20].

4. Core OS

A camada **Core OS** contém as características de baixo nível que foram utilizadas para a implementação da maioria das tecnologias [12], [20]. Em situações que é necessário lidar explicitamente com a segurança ou comunicar com um acessório de *Hardware* externo, são utilizados os *frameworks* desta camada, como é o exemplo do **Accelerate Framework**, **External Accessory Framework**, **Local Authentication Framework**, entre outros *frameworks* [20].

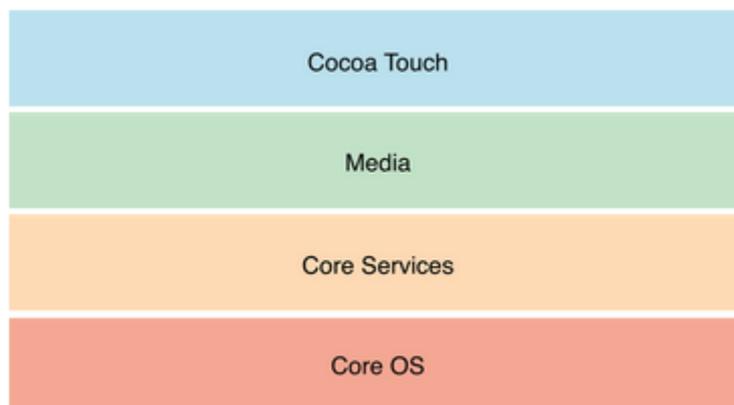


Figura 2.5 - Arquitetura iOS [11], [20]

2.2.2.3. Windows Phone

Anteriormente, o sistema operativo móvel criado pela Microsoft era chamado de Windows Mobile. Após as alterações introduzidas pela Apple (iOS) e pelo Google (Android) em 2007, a Microsoft decidiu tomar uma nova direção e criar o Windows Phone. Os programas para Windows Phone 7 são escritos em *.NET managed code*. O *managed code* é o código escrito em linguagens que estão

disponíveis para utilização com o *framework* Microsoft .NET, por exemplo o C#. O Windows Phone 7 suporta duas plataformas de programação populares, chamados **Silverlight** e **XNA**. **Silverlight** que é uma evolução do **Windows Presentation Foundation** (WPF). Estas fornecem aos programadores a capacidade de criar interfaces de utilizador sofisticadas. O **XNA** é uma plataforma de jogos da Microsoft e suporta tanto gráficos 2D como 3D. A plataforma de desenvolvimento do Windows Phone é o Visual Studio [13]. A versão mais recente do Windows Phone é o Windows Phone 8 lançada em Janeiro de 2014 e as principais mudanças reside no fato de ele suportar os CPUs *quad-core* e *octa-core* e também suportar resoluções mais altas [11]. A Figura 2.6 ilustra a arquitetura do Windows Phone 8. Este consiste também em 4 camadas. Na camada inferior contém os *drivers* necessários para suporte do *Hardware* do dispositivo.

A segunda camada é o *kernel*. O Windows Phone 8 alterou o *kernel* para Windows NT, que também é utilizado pelo Windows 8 para computadores, em vez do Windows CE utilizado anteriormente no Windows Phone 7. O sistema operativo e o *kernel* vem do Windows com algumas modificações específicas do Windows Phone. Isto simplifica o processo de portabilidade de aplicações entre os computadores de mesa e os dispositivos móveis. O Windows Phone 8 utiliza o **Core System** do Windows, que lida com os recursos do sistema, tais como o arranque, gerenciamento do *Hardware* e de recursos. O **Core System** é complementado pelo **Mobile Core**, um conjunto de binários específicos do Windows Phone para lidar com as tarefas específicas do telefone. Acima da camada do *kernel* são os **Shared Services** do sistema e APIs utilizados pelas aplicações para ter acesso às funcionalidades subjacentes. Esta camada contém muitos dos mesmos recursos utilizados pelo Windows 8, como o sistema de ficheiros NTFS, o motor de gráficos DirectX, elementos de segurança, entre outros. O desenvolvimento de aplicações é feito a partir do Windows Phone SDK e Driver Kit oferecido pela Microsoft.

A camada superior contém os *frameworks* de desenvolvimento de aplicações para Windows Phone. As aplicações são implementadas a partir da linguagem de programação C#, Visual Basic .NET ou C++.

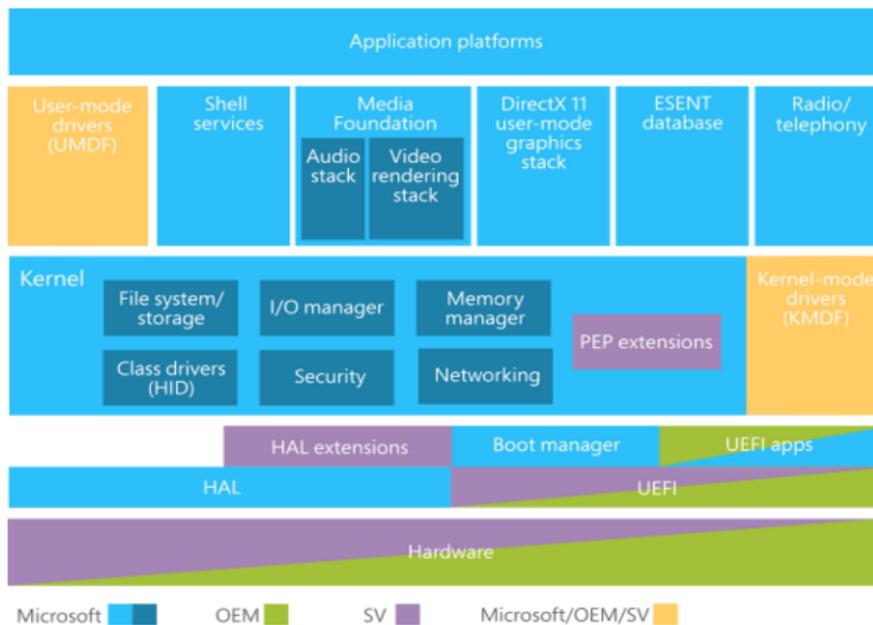


Figura 2.6 - Arquitetura do Windows Phone [11]

2.2.2.4. BlackBerry OS

O BlackBerry OS e a sua plataforma de desenvolvimento foram desenvolvidos pela empresa Canadiana **Research-In-Motion (RIM)** [9], [21]. Este sistema operativo oferece uma plataforma para desenvolvimento de aplicações que suporta somente o J2ME. A máquina virtual Java (JVM) do BlackBerry é baseado na implementação do J2ME da **Sun**, encontrando-se parcialmente escrito em C, C++ e *assembler*. É uma implementação nativa, e está localizada no *firmware* do dispositivo, o que torna muito difícil de alterá-la. Duas grandes vantagens para isto são:

- O sistema operativo não tem que ser compilado para o tipo de CPU do dispositivo;
- Ao mesmo tempo, fornece camadas de abstração de *Hardware* para outras funcionalidades de *Hardware* do dispositivo, como o botão de controlo do som, da comunicação radio, e outros. Teoricamente, dá um melhor desempenho ao dispositivo.

O BlackBerry OS tem suporte *multitasking* com integração de gestos (*gestures*) [13].

A Figura 2.7 a seguir apresentada ilustra os componentes da arquitetura do sistema operativo móvel BlackBerry 10. Estes componentes trabalham juntos para proteger a privacidade, a integridade e a confidencialidade da aplicação e dos seus dados. A Tabela 2.5 a seguir apresentada descreve cada um dos componentes da arquitetura do BlackBerry 10.

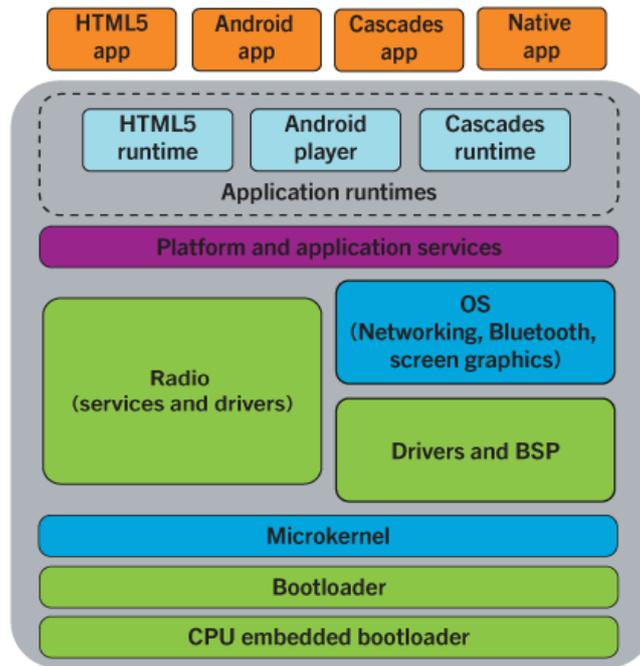


Figura 2.7 – Arquitetura do BlackBerry 10 OS [21]

Tabela 2.5 – Componentes da Arquitetura do BlackBerry 10 OS [21]

Componente	Descrição
CPU embedded bootloader	Verifica a assinatura digital do código <i>bootloader</i> antes de ser executado.
Bootloader	Verifica a assinatura digital do sistema operativo antes de ser executado.
Microkernel	É a quantidade mínima de <i>Software</i> que o sistema operativo requer para ser executado.
Radio	Inclui os <i>drivers</i> , pilhas (<i>stacks</i>) e serviços necessários para suporte dos subsistemas rádio para voz, dados e outros serviços.
Drivers and BSP	Incluem os <i>drivers</i> e a lógica <i>board ring</i> para suporte do Hardware do dispositivo.
OS	São os processos do OS que existem fora do <i>kernel</i> .
Platform and application services	Incluem o gerenciamento da segurança, gerenciamento da instalação de <i>Softwares</i> , os serviços de segundo plano para as aplicações, os serviços media, entre outros. Estes são

	necessários porque as aplicações não executam serviços em segundo plano ou obter acesso aos componentes e serviços protegidos.
Application runtimes	Incluem as máquinas virtuais, as bibliotecas, os serviços, o mapeamento das camadas, e outros. Todas as aplicações são executadas em caixas de proteção (<i>sandbox</i>) isoladas.
App	São as aplicações nativas introduzidas pela organização ou os instalados pelo utilizador.

2.3. Metodologias de desenvolvimento de aplicações móveis

Existem duas abordagens diferentes para o desenvolvimento de aplicações móveis para as diferentes plataformas móveis: A abordagem nativa e a abordagem multiplataforma.

A abordagem nativa consiste em desenvolver aplicações para uma determinada plataforma móvel, enquanto a abordagem multiplataforma consiste em desenvolver aplicações móveis que possam ser executadas em mais do que uma plataforma móvel. A abordagem multiplataforma encontra-se ainda dividida em 4 abordagens diferentes: Abordagem WEB, híbrida, interpretada e compilação-cruzada.

2.3.1. Abordagem Nativa

As aplicações nativas são aquelas projetadas para serem executadas numa plataforma móvel em específico. Por exemplo, no caso do Android, o desenvolvimento é baseado na linguagem Java e tem como IDE o Android SDK (ou atualmente o Android Studio) e para o iOS é baseado na linguagem de programação Objective-C e a IDE é o Xcode IDE para interface com o utilizador, ferramenta de análise de desempenho e iOS simulador [22], [23]. A Tabela 2.6 a seguir apresenta as diferentes ferramentas, linguagens, formatos e distribuição das aplicações nativas dos principais sistemas operativos móveis.

Tabela 2.6 - Desenvolvimento de Aplicações Nativas [24]

	Apple iOS	Android	BlackBerry OS	Windows Phone
Linguagem	Objective-C	Java	Java (J2ME)	C#, VB .NET e outros
IDE	Xcode	Android SDK Android Studio	BB Java Eclipse Plug-in	Visual Studio, Windows Phone tools
Formato	.app	.apk	.cod	.xap
App stores	Apple App Store	Google Play	BlackBerry App World	Windows Phone Marketplace

2.3.2. Abordagem multiplataforma

Esta abordagem consiste em desenvolver aplicações para várias plataformas móveis com um único código fonte. Existem 4 tipos de abordagem multiplataforma: **Abordagem WEB**, **Híbrida**, **Interpretada** e **Compilação-Cruzada**.

2.3.2.1. Abordagem WEB

A abordagem WEB também denominadas aplicações WEB, são aplicações projetadas para serem executadas nos *browsers* dos dispositivos móveis, de modo a obterem os conteúdos da internet. As aplicações móveis WEB são desenvolvidas utilizando tecnologias WEB como *HTML*, *CSS* e *JavaScript*, e dadas as novas características emergentes do HTML5 e CSS3, possibilitaram o desenvolvimento de aplicações mais ricas e complexas [25], [26].

Nesta abordagem, o dispositivo móvel não terá que armazenar conteúdos que são específicas para aplicações WEB, desde que os dados da aplicação são conduzidas pelo servidor. Uma vez que a aplicação é baseado no *browser*, a aplicação será independente da plataforma [27], [28]. A Figura 2.8 representa a arquitetura das aplicações WEB.

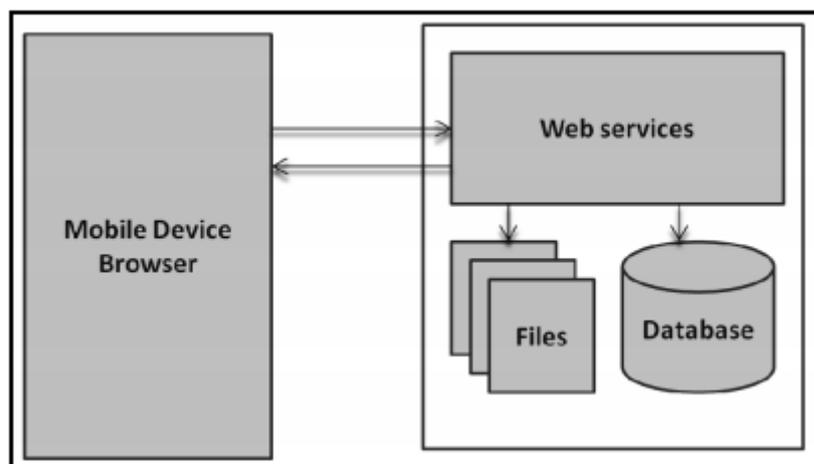


Figura 2.8 – Arquitetura de aplicações WEB [27], [25]

Algumas operações da aplicação são realizadas no lado do cliente, mas no entanto as operações críticas ou mais importantes são realizadas no servidor. O cliente alberga a interface do utilizador e a lógica de validação dos dados do utilizador, enquanto o servidor implementa a lógica de negócio (“*business logic*”) [27].

A principal desvantagem das aplicações WEB reside no fato de ter acesso limitado ao *Hardware* e aos dados do dispositivo subjacente. Outro problema resulta do tempo extra necessário para devolver a página WEB e o custo adicional para baixar a esta mesma página da internet. Ainda, uma vez que estas aplicações não são instaladas fisicamente no dispositivo móvel, existem situações, como por exemplo, quando os dispositivos encontra-se no modo “Avião”, as aplicações WEB são inacessíveis para o utilizador [25], [28], [29].

As aplicações WEB tem inúmeras vantagens, tais como:

- Não requererem nenhuma instalação no dispositivo móvel, visto que a aplicação é acedida a partir do URL introduzida no *browser* do dispositivo;
- Os dados da aplicação encontram-se no servidor, de modo que não é necessário nenhuma atualização da aplicação no dispositivo móvel;
- Uma vez que os *browsers* são bastantes padronizados, a interface do utilizador WEB pode ser reutilizada em algumas plataformas [27], [29].

Como tal, ainda existem alguns desafios relevantes a serem ultrapassadas no desenvolvimento deste tipo de aplicações, tais como:

- As aplicações WEB não são distribuídas nas lojas de aplicações móveis (*mobile application stores*), visto que são acedidas a partir de um *URL*. Hoje em dia, os utilizadores procuram as aplicações que desejam nas lojas de aplicações. A ausência destas aplicações nestas lojas pode ter um impacto negativo sobre a popularidade da aplicação.
- Podem ter um desempenho inferior devido a atrasos na conexão e atrasos da rede.
- Não tem acesso ao *Hardware* da plataforma, tais como, a Camera, o GPS, Bluetooth, Acelerómetro, NFC etc.,
- Ao contrário das Aplicações Desktop, as aplicações WEB tem que suportar diferentes resoluções do ecrã. É um dos fatores mais importantes a ser considerado durante o desenvolvimento deste tipo de aplicações móveis.
- O teste manual relativa a aparência da aplicação em resoluções de ecrã diferentes consomem uma quantidade de tempo considerável.
- O programador tem menos controlo sobre a forma como os diferentes *browsers* irão processar o conteúdo.
- As aplicações WEB são limitadas em potenciar os gestos (*gestures*) oferecidos pelas plataformas.
- Monitorizar uma aplicação WEB não é tão simples como uma aplicação nativa [27].

É mais simples começar a implementar uma aplicação WEB móvel do que desenvolver uma aplicação nativa. No entanto, devido ao fato de cada dispositivo ter dimensões e resoluções diferentes, implica a realização de testes em diferentes dispositivos.

2.3.2.2. Abordagem Híbrida

A abordagem híbrida, também conhecida como abordagem “*WEB to native wrapper*”, encontra-se entre a abordagem WEB e nativa. As aplicações híbridas são desenvolvidas utilizando tecnologias WEB e são executados nos dispositivos móveis como se fossem aplicações nativas, ou seja, é um médio termo entre a abordagem nativa e a abordagem WEB [30], [31]. Utiliza o motor WEBKit para exibir os controlos, botões e animações. O motor WEBKit é portanto, o responsável pelo desenho e gestão dos objetos da interface do utilizador [32], [25]. Os recursos do dispositivo poderão ser acedidos pela aplicação híbrida a partir das APIs especializados da camada de

abstração. A abordagem híbrida tira vantagem tanto do mecanismo de navegação, como também dos recursos do dispositivo. Ao contrário de aplicações WEB, as aplicações híbridas necessitam ser baixadas e instaladas no dispositivo móvel como se fossem aplicações nativas, mas a sua performance são muitas vezes inferiores a estas, de modo que a sua execução requer da execução do *browser* que faz parte da aplicação final [25], [26]. Comparativamente às aplicações nativas, estes são inferiores, visto que necessitam mais tempo de processamento e não alcançam completamente a aparência nativa [29]. Como um exemplo de um *framework* mais popular para a criação de aplicações híbridas é o PhoneGap ou Apache Cordova. A implementação do código para criar aplicações híbridas pode ser feito utilizando várias tecnologias e plataformas de desenvolvimento, e para conseguir uma interface parecida à nativa, ou seja, uma interface de utilizador simulada, é necessário a utilização de *frameworks* de desenvolvimento específico, tais como o JQuery ou o Sencha Touch (ferramentas que seguem a abordagem WEB), também designados *UI-frameworks* [27], [28]. A Figura 2.9 representa a arquitetura destas aplicações.

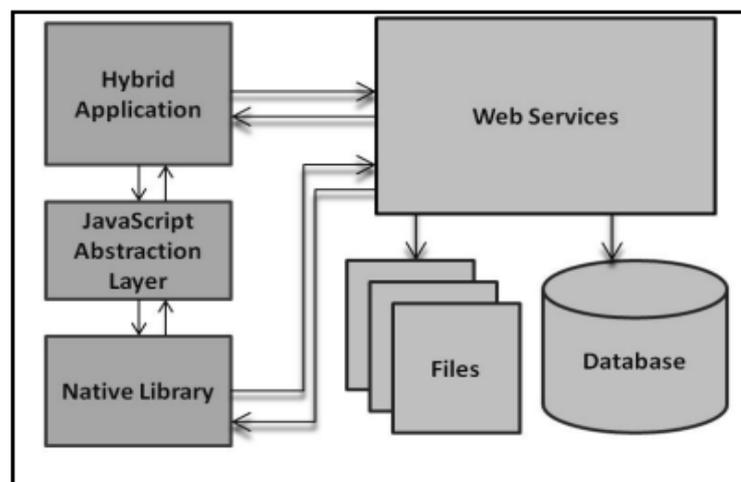


Figura 2.9 - Arquitetura de aplicações Híbridas [27]

Estas aplicações tem inúmeras vantagens, tais como:

- As aplicações híbridas podem ser distribuídas nas lojas de aplicações (*app stores*);
- A principal vantagem da abordagem híbrida reside no fato da interface do utilizador poder ser reutilizada em diferentes plataformas, utilizados recursos da plataforma nativa;
- Uma vez que os recursos da plataforma nativa são disponibilizadas através da camada de abstração de *Hardware*, a aplicação pode utilizar os recursos do dispositivo;

- Os recursos do *Hardware* e da plataforma do dispositivo podem ser acedida com APIs de *frameworks* específicas [27], [29].

Os grandes desafios para o desenvolvimento de aplicações híbridas, são:

- Estas aplicações tem desempenho inferior, comparativamente às aplicações nativas;
- Uma vez que as aplicações híbridas utilizam a linguagem de programação *JavaScript* na camada de abstração do *Hardware*, são submetidos às vulnerabilidades de comunicação em espaço cruzado, nomeadamente o **Cross-Site-Scripting** ou **XSS**, que consiste na injeção de código malicioso utilizando a linguagem de programação JavaScript, entre outros. As aplicações híbridas também sofrem de comportamentos específicos da plataforma *JavaScript* e incompatibilidades do *threading model* com *JavaScript*;
- Mesmo que a interface do utilizador possa ser reutilizado em diferentes plataformas, este não terá a aparência de uma aplicação nativa [27].

2.3.2.3. Abordagem Interpretada

No caso das aplicações *runtime* ou interpretada, o código da aplicação é implementado para o dispositivo móvel e este é interpretado logo a seguir, ou seja, a aplicação é desenvolvida numa linguagem específica (ex. JavaScript), mas no entanto, este mesmo código é interpretada (a partir de um interpretador) e posteriormente convertida em aplicações nativas para as diferentes plataformas. Existe um interpretador que executa o código em tempo de execução [26], [29]. Os recursos nativos são disponibilizados por intermédio de uma camada de abstração. O compilador interpreta o código fonte e converte-a ao mesmo tempo para aplicações de diferentes plataformas, e portanto, suporta o desenvolvimento de aplicações multiplataformas. A aplicação interpretada interage com a camada de abstração para aceder as APIs nativas e utilizam os componentes da interface do utilizador nativo da plataforma específica para interação com utilizadores. A lógica da aplicação é implementada independentemente utilizando várias tecnologias, tais como Java, Ruby, XML, entre outros.[31]. A principal vantagem desta abordagem é a eficiência, devido a sua interface de utilizador nativo, mas também tem desvantagens. A principal desvantagem reside no fato destas aplicações estarem totalmente dependentes do ambiente de desenvolvimento de *software*, ou seja, os novos recursos da plataforma específica podem estar disponível para

aplicações, somente quando e se for suportado pelo ambiente de desenvolvimento [26], [29]. Esta abordagem permite que o programador desenhe e desenvolva a interface final do utilizador, que é idêntica a interface do utilizador nativo sem qualquer linha de código adicional. Também, esta pode atingir um nível elevado de código reutilizável, mas no entanto poderá reduzir um pouco o desempenho, devido o processo de interpretação do mesmo [25]. Um exemplo do ambiente de desenvolvimento de *software* mais popular para a criação de aplicações interpretadas é o Appcelerator Titanium Mobile [25], [27], [28], [26]. A Figura 2.10 a seguir apresentada representa a arquitetura das aplicações interpretadas.

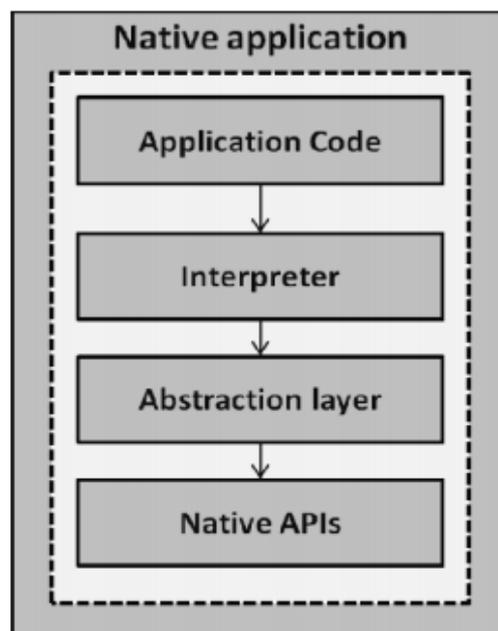


Figura 2.10- Arquitetura de aplicações interpretadas [27]

Estas aplicações também tem as suas vantagens, tais como:

- Fornece a aparência de aplicações nativas. A lógica do negócio pode ser reutilizada em diferentes plataformas [26], [29];
- O *Hardware* do dispositivo e os recursos da plataforma são envolvidos com API de *frameworks* específicos [27].

Além de vantagens também disponham um conjunto de desafios a serem ultrapassadas, tais como:

- A reutilização da interface do utilizador depende do nível de abstração do *framework*;
- A principal desvantagem das aplicações *runtime* reside no fato do desenvolvimento estar dependente de um conjunto de características fornecidas pelo *framework* escolhido;
- O desempenho da aplicação pode degradar um pouco devido ao processo de interpretação do código em tempo de execução [27].

2.3.2.4. Abordagem Compilação-Cruzada

No caso de aplicações de compilação-cruzada ou também designados aplicações *generated*, o *cross-compiler* (multicompilador) converte o código fonte para binários nativos. Este é o responsável por gerar o código executável para uma plataforma em específico. O programador pode escrever o código fonte em uma linguagem de programação comum e o *cross-compiler* compila o código fonte para o código nativo de uma plataforma em específico. A aplicação final utiliza a linguagem nativa, portanto, pode ser considerada uma aplicação nativa para todos os efeitos. Assim, esta abordagem permite que os programadores tenham acesso total a todos os recursos oferecidos pelo dispositivo móvel e o desempenho pode ser considerada semelhantes (senão mesmo igual) relativamente às aplicações nativas [29]. Alguns testes já realizados mostraram que para aplicações complexas, a solução nativa é melhor, desde que o código gerado dê piores desempenhos a aplicação resultante, em comparação com o código escrito pelo programador. Esta abordagem depende da eficiência e fiabilidade do *cross-compiler*. Como exemplos de *frameworks* de desenvolvimento de aplicações de compilação-cruzada são: o Applause e o Rhodes [27]. A Figura 2.11 a seguir ilustrada representa a arquitetura das aplicações de compilação-cruzada ou *Generated*.

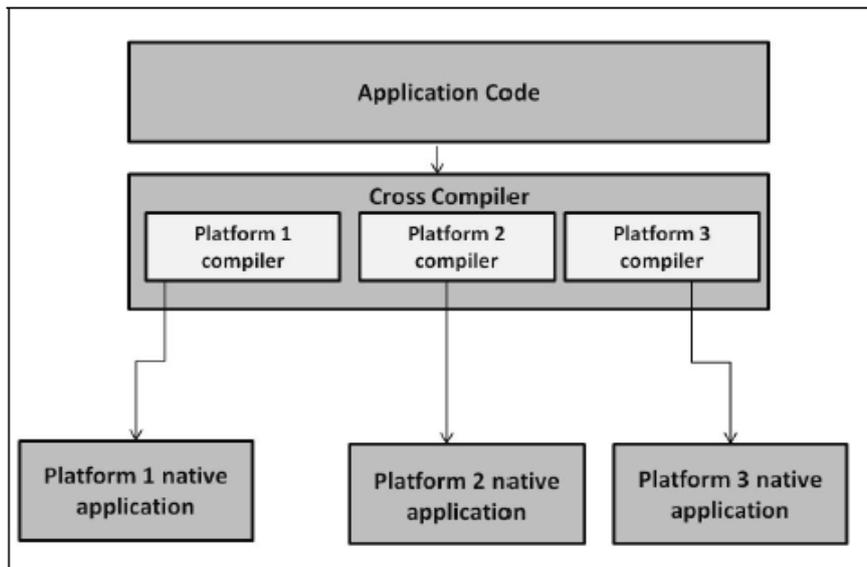


Figura 2.11 - Arquitetura de aplicações Cross-platform [27]

As principais vantagens desta abordagem são:

- As aplicações de compilação-cruzada fornecem todos os recursos que as aplicações nativas oferecem [25];
- O *Hardware* e o *Software* do dispositivo podem ser acedidos. Todos os componentes da interface do utilizador nativo podem ser utilizados.
- O desempenho é o principal destaque destas aplicações [27], [29].

Os desafios para esta abordagem são:

- A principal desvantagem das aplicações de compilação-cruzada reside no fato da interface do utilizador não poder ser reutilizada. Também os recursos específicos da plataforma, como o acesso a camera, serviços de localização, notificações locais, entre outros, não podem ser reutilizadas. Estas características são específicas das plataformas móveis e a forma de acede-las, varia de plataforma para plataforma.
- Esta abordagem deverá ser apropriada para aplicações simples, de modo que serão superados pela abordagem nativa para o desenvolvimento de aplicações sofisticados.

- Identificar e corrigir os problemas das fases da compilação múltipla é difícil para o programador da aplicação [27].

2.3.3. Critérios de Seleção entre as abordagens de desenvolvimento multiplataforma

O critério de seleção da abordagem multiplataforma a adotar depende principalmente da exigência da aplicação, mas também das plataformas alvos, do tipo de aplicação, do acesso aos dados e ao *Hardware*, da interface do utilizador e sua aparência, do seu desempenho e da sua distribuição no mercado. O suporte das ferramentas e dos *frameworks* também deve ser um fator a ser considerado ao selecionar uma abordagem multiplataforma [27].

Tabela 2.7 - Metodologia Multiplataformas e frameworks adotados [27], [33]

Abordagem Multiplataforma	<i>Frameworks</i> adotados	Linguagens utilizadas	Plataformas suportadas
WEB	Tecnologias WEB	HTML, CSS, JavaScript, etc.	Android, iOS, Windows, BlackBerry
Híbrido	PhoneGap	JavaScript	Android, iOS, Windows, BlackBerry
Interpretada	Titanium, JMango	JavaScript	Android, iOS, BlackBerry
Compilação-Cruzada	Rhodes, Applause	C#	Android, iOS, Windows

Cada aplicação móvel tem as suas próprias características, de modo que a seleção da abordagem multiplataforma apropriada deverá ser aplicada a cada uma das aplicações. As aplicações móveis são classificadas principalmente em aplicações de dados conduzidos pelo servidor, aplicações baseados em Sensor/IO, aplicações autónomas/independentes e aplicações cliente-servidor, e a escolha deve ser escolhida com sabedoria consoante o tipo de aplicação [34].

A Execução Nativa refere-se às abordagens compilação-cruzada e interpretada, uma vez que ambos executam no espaço nativo. O desempenho das aplicações interpretadas é inferior

comparativamente com as aplicações de compilação-cruzada, devido á sua interpretação dinâmica [27].

Classificação das Aplicações

➤ Aplicações de dados conduzidos pelo servidor

As aplicações de dados conduzidos pelo servidor são aplicações clientes, onde a lógica de negócio (*business logic*) reside no servidor. O cliente móvel permite a visualização de informações e a interação com o utilizador. A abordagem WEB seria adequado para estes tipos de aplicações. Seguindo a abordagem WEB, a interface do utilizador destas aplicações podem ser reutilizados. Devido ao fato da lógica de negócio da aplicação residir no servidor, o cliente móvel não precisa se preocupar com quaisquer atualizações da aplicação no dispositivo móvel. Se a aplicação desejar executar qualquer ação ao nível da plataforma, como por exemplo, agendar uma notificação local sobre a recuperação de dados, a abordagem hibrida seria a mais adequada [27].

➤ Aplicações baseadas em Sensores/IO

As aplicações baseadas em sensores utilizam principalmente o *Hardware* do dispositivo móvel. Algumas aplicações processam os dados dos sensores localmente, enquanto algumas dependem do servidor para processamento destes dados. Esta abordagem deve ser selecionada sabiamente com base na área de processamento para estes tipos de aplicações. As aplicações de compilação-cruzada seriam a melhor escolha, se os dados fossem processados no dispositivo móvel, devido ao fato de terem acesso ao *Hardware* nativo e também ótimo desempenho. As aplicações interpretadas poderiam ser a segunda opção, visto que oferecem um conjunto de recursos nativos a partir dos ambientes de desenvolvimento correspondente. No entanto, em termos de desempenho, as aplicações interpretadas são inferiores comparativamente às aplicações de compilação-cruzada.

Se uma aplicação tem um processamento intensiva e requer um servidor para processamento dos dados, então a abordagem hibrida poderia ser o mais significativo [27].

➤ Aplicações Autônomas

As aplicações autônomas aqui refere-se às aplicações que os dados são produzidos e processados no próprio dispositivo móvel. Também inclui aplicações onde os dados vem do servidor e processamento acontece no cliente. O cliente móvel processa localmente e exibe a visualização de informações, permitindo também a interação com utilizadores. Para as aplicações autônomas, a abordagem interpretada seria a escolha mais indica, uma vez que podem aproveitar os recursos nativos da plataforma móvel [27].

➤ Aplicações Cliente – Servidor

No caso de aplicações cliente-servidor, tanto o servidor como o cliente encontram-se envolvidos no processamento dos dados na aplicação [27]. Nas aplicações cliente-servidor, o cliente além de permitir a visualização de informações provenientes do servidor e a interação com o utilizador, também este efetua o processamento de dados localmente de modo a exibi-la posteriormente. Estes dados não são enviados para o servidor, visto que o cliente deverá efetuar a sua gerência e processamento, posteriormente serão disponibilizados aos utilizadores da aplicação. A abordagem híbrida será a mais apropriada aqui.

É muito importante a seleção de uma abordagem multiplataforma a ser seguido para um tipo de aplicação em particular. Na Tabela 2.8 a seguir apresentada, apresenta a classificação de cada abordagem para diferentes tipos de aplicações [27].

Os resultados da avaliação devem ser interpretados da seguinte forma:

- 1 – Não preferencial
- 2 – Preferencial, mas não é a abordagem perfeita
- 3 – Abordagem perfeita

Tabela 2.8 - Tipos de Aplicação e abordagem preferido [27]

Código da aplicação	WEB	Híbrida	Execução Nativa (Interpretada/ Compilação-Cruzada)
Aplicações de dados conduzidos pelo servidor	3	2	1
Aplicações autónomas	1	2	3
Aplicações baseadas em Sensores/IO (Dados processados no Dispositivo móvel)	1	2	3
Aplicações baseadas em Sensores (Dados processados no Servidor)	1	3	2
Aplicações Cliente-Servidor	1	3	2

2.3.4. Análise comparativa entre as abordagens de desenvolvimento multiplataforma

Pretende-se efetuar uma análise comparativa entre as abordagens de desenvolvimento multiplataforma de aplicações móveis, anteriormente descritas, com base num conjunto de características, convergentes com um conjunto de critérios. Os critérios para a análise comparativa entre as abordagens de desenvolvimento multiplataforma são os seguintes:

- **Distribuição (Market place deployment)** – avaliar a possibilidade e a facilidade na distribuição de aplicações nos *app stores* das plataformas móveis, como o Google Play ou o iTunes da Apple;
- **Tecnologias generalizadas** – avaliar se as aplicações podem ser desenvolvidas utilizando tecnologias generalizadas, como por exemplo, o JavaScript;
- **Acesso ao *Hardware* e aos dados** – avaliar o acesso ao *Hardware* do dispositivo e aos dados, ou seja, avaliar se as aplicações tem acesso total, não tem acesso ou tem acesso limitado ao *Hardware* e aos dados do dispositivo;

- **Interface do utilizador e a aparência** – avaliar se as aplicações suportam intrinsecamente os componentes da interface do utilizador nativo e se a aparência é simulada através de ferramentas, como o JQuery ou Sencha Touch;
- **Percepção utilizador relativamente ao desempenho** – avaliar se as aplicações tem baixo, médio ou alto desempenho de acordo com a percepção dos utilizadores finais (como o tempo de carregamento e a velocidade de execução) em comparação com uma aplicação nativa. Este critério é uma estimativa empírica aproximada com base na nossa experiência prática e nas informações publicadas na WEB. O desempenho global poderá ser afetada por inúmeros fatores [28], [30], [33], [35], [36].

Existem muitas mais características que são tão importantes como estes, que poderiam ser utilizados para enriquecer esta análise comparativa.

Tabela 2.9 - Análise comparativa das abordagens de desenvolvimento multiplataforma [28], [33]

	WEB	Híbrida	Interpretada	Compilação-Cruzada
Distribuição (<i>App Stores</i>)	Não	Sim	Sim	Sim
Tecnologias Generalizadas	Sim	Sim	Sim	Não
Acesso aos dados e Hardware	Limitado	Limitado	Limitado	Total acesso
Interface de utilizador e aparência	Simulada	Simulada	Nativa	Nativa
Percepção da performance	Baixo	Medio	Medio	Alto

As aplicações WEB são aplicações baseadas em *browsers* e não são instaladas fisicamente no dispositivo. O tempo extra necessário para baixar o código fonte e os recursos da WEB faz com que as estas aplicações tenham inevitavelmente um baixo desempenho em relação às aplicações nativas e também, podem tornar-se indisponíveis após a falha da rede. *Framework* como o JQuery Mobile prometem desenvolver aplicações com aparência nativa.

As aplicações híbridas simulam a aparência de uma aplicação nativa. As principais vantagens das aplicações híbridas residem no fato da sua capacidade de executar o código fonte em diversas

plataformas móveis, e do seu desenvolvimento ser efetuada utilizando tecnologias de desenvolvimento WEB amplamente utilizados (especialmente o JavaScript). Estas aplicações tem um desempenho médio, segundo a percepção dos utilizadores finais comparativamente com as aplicações nativas, uma vez que a interface do utilizador ainda é baseado na WEB, e não na utilização de componentes nativos otimizados. Por outro lado, as APIs especializadas são intersetadas para interpretar o acesso ao *Hardware* e a lógica de negócio. O acesso ao *Hardware* e aos dados da aplicação é limitado e no modo geral, as aplicações podem ser distribuídas via *app stores* (lojas de aplicações) [28], [30].

As aplicações interpretadas têm interface de utilizador nativo e a lógica da aplicação é implementada de forma independente. O desempenho geral segundo a percepção dos utilizadores finais é médio em comparação com aplicações nativas, visto que por um lado tem a interface do utilizador rápida, mas por outro lado o tempo extra necessário para interpretar a lógica da aplicação e a presença de APIs especializadas para o acesso aos componentes de *Hardware*, faz com que perca no desempenho. Estas aplicações tem acesso limitado ao *Hardware* e aos dados do dispositivo móvel. Em termos de fiabilidade, as APIs especializadas podem ser utilizados para armazenar o estado e os dados da aplicação (exemplo, SQLite API) e em geral as aplicações podem ser distribuídos sem nenhuma dificuldade.

As aplicações de compilação-cruzada atingem alto nível de desempenho como as aplicações nativas. Os ambientes de desenvolvimento de *Software* desta categoria suporta as principais plataformas móveis, como o iOS, Android e Windows Phone. Estes têm acesso total ao *Hardware* e aos dados do dispositivo, e tem interface de utilizador nativo [28], [30].

2.3.5. Análise comparativa entre o desenvolvimento de aplicações móveis nativas e multiplataforma

Ambas as abordagens de desenvolvimento tem as suas vantagens e desvantagens. A Tabela 2.10 a seguir apresentada resume a comparação entre as duas abordagens de acordo com os seguintes critérios:

- **Experiência do utilizador e interface de utilizador** – As aplicações nativas oferecem uma interface mais fluida e responsiva que as soluções multiplataforma, especialmente para

animações e gestos. Isso ocorre porque ao escrever o código com a linguagem nativa da plataforma, temos acesso às APIs de todos os dispositivos. Embora as soluções multiplataformas ofereçam APIs nativas para utilização, mas referem sempre a um subconjunto limitado de recursos específicos dos dispositivos e muitas das vezes é necessário esperar até que sejam disponibilizados de modo a serem utilizados.

- **Desempenho** – uma das principais vantagens da utilização de ferramentas de desenvolvimento nativo, reside no fato destas aplicações correrem mais suavemente em qualquer dispositivo móvel que utiliza este sistema operativo. Em vez disso, o processo multiplataforma é as vezes mais lento do que utilizando as ferramentas nativas para uma aplicação. Esta diferença pode ser notada facilmente durante na representação gráfica e animações.
- **Recursos específicos do dispositivo** – As ferramentas nativas permitem que os programadores/desenvolvedores de aplicações tenham vantagens sob todas as funcionalidades da plataforma. Pelo contrário, as aplicações multiplataformas fornecem todas as vantagens para um número limitado de funcionalidades. Por exemplo, com aplicações multiplataformas, o suporte aos gráficos *high-end* e 3D são muitas vezes limitadas.
- **Distribuição via *app store*** – podem haver requisitos rigorosos para admissão das aplicações nas lojas de aplicações (*app stores*) públicas. A Apple Inc., por exemplo, requer que os programadores submetam as aplicações para testes e verificação da compatibilidade com os seus dispositivos móveis antes de disponibilizar na *app store*. Neste sentido, as aplicações desenvolvidas utilizando ferramentas nativas são geralmente são aceites de forma mais fácil nas *app stores* do que aplicações multiplataformas, porque presume-se que as aplicações nativas foram desenvolvidos por programadores com conhecimentos da plataforma destino.
- **Custos de desenvolvimento para múltiplas plataformas** – Os desenvolvedores de aplicações nativas, ao direcionarem às suas aplicações para múltiplas plataformas móveis, devem adotar as boas práticas da arquitetura da aplicação e utilizarem o modelo de dados para otimizarem o esforço de desenvolvimento em múltiplas plataformas. No desenvolvimento de aplicações nativas, requer programadores com competências para as plataformas de destino. Com as aplicações multiplataformas, o custo de desenvolvimento é reduzido. Esta é talvez a maior vantagem, porque o

esquema de compilações múltiplas permite que as empresas ou marcas obtenham uma aplicação para as outras plataformas sem ter que investir em uma equipa ou programadores específicos para este ecossistema.

- **Suporte a programadores** – Os programadores em geral, preferem o desenvolvimento de aplicações nativas, visto que é mais fácil de obter ajuda a partir de *forums online* e obter respostas rápidas de pessoas que já desenvolvem estas aplicações a mais tempo. O desenvolvimento de aplicações multiplataforma por ser uma técnica mais recente poderá ser um pouco mais difícil de obter ajudas. No entanto, existem *frameworks* de desenvolvimento de aplicações multiplataformas que oferecem uma boa documentação de APIs e é mais fácil obter ajudas, como é o exemplo do **Sencha Touch**, e do **PhoneGap**, entre outros.
- **Segurança** – As aplicações multiplataformas apresentam maiores riscos de segurança do que as aplicações nativas. A razão está diretamente relacionada com os riscos do HTML5. Por exemplo, para citar alguns, vulnerabilidade de segurança URL, devido aos dados armazenados em cache no dispositivo móvel (no *browser*). Pelo contrário, os sistemas operativos como o iOS e o Android oferecem serviços de segurança integrados, como a criptografia de dados. As técnicas de ataque como a manipulação e *cookies* e *SQL injection* para aceder dados sensíveis de servidores *back-end* e para dispositivos móveis em si não são possíveis em aplicações nativas (bem contruídas).
- **Acesso atempado as novas inovações da Plataforma móvel** – Os *frameworks* multiplataformas podem não suportar todas as funcionalidades do sistema operativo móvel ou do dispositivo. Qualquer nova funcionalidade ou recurso adicionado no sistema operativo não fica imediatamente disponível no *framework* multiplataforma a utilizar. É necessário um tempo de espera, de modo que seja atualizado para poder suportar as novas funcionalidades.
- **Reutilização do código** – o código das aplicações multiplataformas são consideradas reutilizáveis. Em vez de ter que escrever uma ação específica ou uma sequência de ações para cada plataforma alvo em particular, o programador pode escrever o código apenas uma única vez e reutiliza-la em outras plataformas ou em outros projetos. Isso nem sempre é verdade, visto que alguns dos *frameworks* multiplataforma utilizam muitas vezes seus próprios subconjuntos *JavaScript*, o que significa que, para trocar de

plataforma o código escrito anteriormente não vai ser provavelmente reutilizável sem algumas alterações substanciais.

- **Desafios de Projeção** – Em desenvolvimento nativo, a projeção é simplificada pelo suporte e serviços disponibilizados pelo sistema operativo. O sistema operativo pode, por exemplo, notificar as aplicações sobre eventos, como a chegada de mensagens e também sobre o nível de energia. Em ambientes multiplataforma, os programadores precisaram adicionar estes recursos explicitamente. Além disso, com as *frameworks* multiplataformas, os programadores devem projetar como cada recurso que precisam tem que ser implementada em cada plataforma destino. Por exemplo, projetando uma aplicação para *iOs* é diferente do que projetar uma para *Android*. As convenções da interface do utilizador e da experiencia do utilizador são diferentes, e os *touch point* e menus funcionam de maneiras diferentes.
- **Disponibilidade do conhecimento de programação** – é reconhecido que há mais programadores WEB do que os programadores nativos. Como a maioria dos *frameworks* multiplataformas são baseadas em HTML5 e CSS3, e portanto é mais fácil encontrar os conhecimentos necessários entre os desenvolvedores WEB. As habilidades de programadores nativos normalmente custam mais e são mais difíceis de obter [22], [33].

Tabela 2.10 - Comparação entre a abordagem Nativa e Multiplataforma [22], [24], [37]

	Nativo	Multiplataforma
UI e Experiencia do Utilizador	Melhor	Pior
Desempenho	Melhor	Pior
Recursos específicos do dispositivo	Melhor	Pior
Distribuição via <i>app store</i>	Melhor	Pior
Custos de desenvolvimento para múltiplas plataformas	Melhor	Pior
Suporte a programadores	Melhor	Pior
Segurança	Melhor	Pior
Acesso atempado as novas inovações da Plataforma móvel	Melhor	Pior
Reutilização do código	Pior	Melhor
Desafios de Projeção	Pior	Melhor
Disponibilidade do conhecimento de programação	Pior	Melhor

A Figura 2.12 a seguir apresentada retrata de forma mais resumida e concisa a análise comparativa entre o desenvolvimento de aplicações nativas em relação as aplicações multiplataformas. As aplicações multiplataformas tem como objetivo a diminuição dos custos e do tempo de desenvolvimento, de modo que a partir de um único código fonte produzido, pode-se gerar aplicações para diferentes plataformas móveis. Por outro lado, as aplicações nativas apresentam melhores interfaces do utilizador (UI) e melhores desempenho.

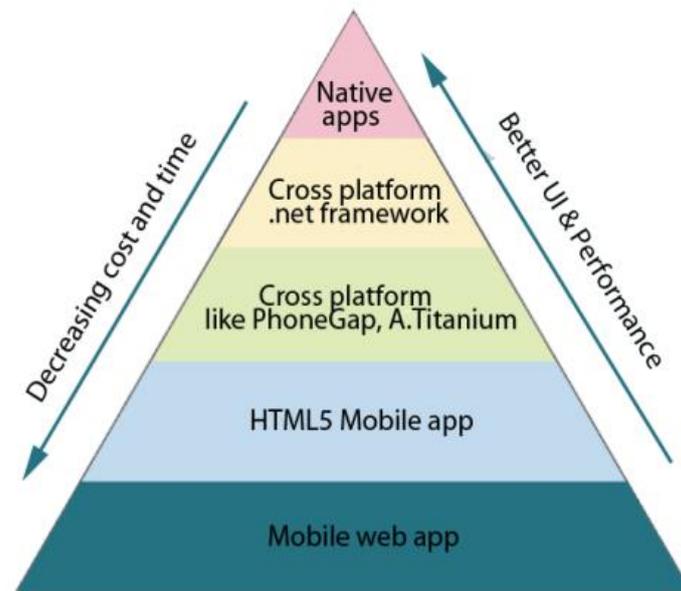


Figura 2.12 - Desenvolvimento Multiplataforma vs Nativa [38]

2.4. Definição das tecnologias - Frameworks Multiplataforma

Esta secção consiste em descrever cinco ferramentas de desenvolvimento diferentes que seguem abordagens de desenvolvimento multiplataforma diferentes. Estas ferramentas de desenvolvimento foram escolhidos com base nos seguintes critérios:

- Licença grátis – permitir o desenvolvimento e distribuição da aplicação livremente, ou seja, sem pagar pela licença. Existem outras ferramentas poderosas, mas devido ao fato de não disponibilizarem a licença gratuitamente, foram descartadas (Exemplo, o **Xamarin** – Figura 2.13).

- Suporte aos três sistemas operativos com mais cota de mercado atualmente (Figura 2.1 e Figura 2.2).

Estas ferramentas de desenvolvimento multiplataforma são: Appcelerator Titanium, PhoneGap, Rhodes, jQuery Mobile e Sencha Touch, sendo as duas últimas, ferramentas de desenvolvimento de aplicações WEB, que também são designados por **UI-frameworks**.

A Figura 2.13 a seguir apresentada ilustra as ações de mercado (*market shares*) das diferentes ferramentas de desenvolvimento multiplataforma em 2014. Embora existam dezenas de ferramentas disponíveis, apenas algumas são largamente utilizados.

A seguir são descritas estas ferramentas de desenvolvimento de aplicações móveis multiplataforma.

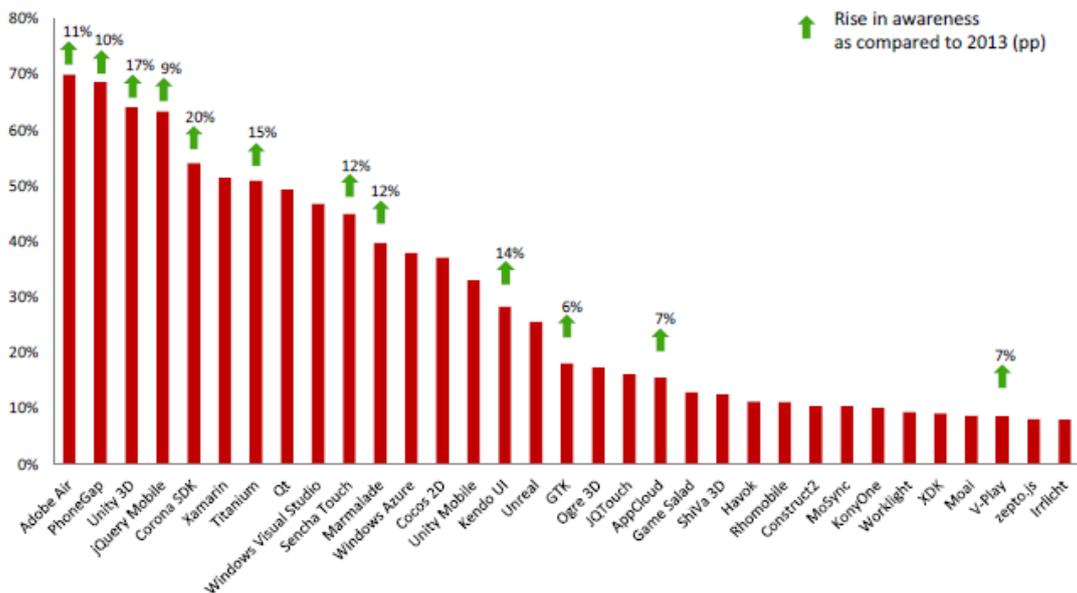


Figura 2.13 - Utilização das ferramentas Multiplataformas em 2014 [39]

2.4.1. Appcelerator Titanium Mobile

O Titanium é um *framework open-source* desenvolvido pelo Appcelerator Inc. sob a licença do Apache 2.0 que permite criar aplicações nativas multiplataformas utilizando JavaScript [40], [35].

O Titanium liga o JavaScript às bibliotecas nativas, compila-o em *bytecode* e em seguida, o SDK da plataforma (Android ou iOS) constrói o *package* para a plataforma destino desejado [41]. Esta *framework* suporta todas as principais plataformas móveis, como o iOS, Android, BlackBerry e Windows Phone.

O Titanium Studio é a IDE oficial, e é baseado na plataforma Eclipse [41], [42], [43]. Esta IDE funciona em todos os principais sistemas operativos, como o Windows, Mac OS e Linux. O Titanium utiliza o padrão **MVC (Model View Controller)** para desenvolver aplicações móveis. O padrão **MVC** possibilita criar aplicações de forma distribuída a partir das definições de dados (*models*), da lógica de negócio (*controllers*) e das interfaces (*views*), fornecendo ao mesmo tempo um ponto de conexão entre estes elementos [43]. O *model* utiliza o framework JavaScript backbone.js, o *view* utiliza o XML e TSS e o *controller* por sua vez utiliza o JavaScript simples.

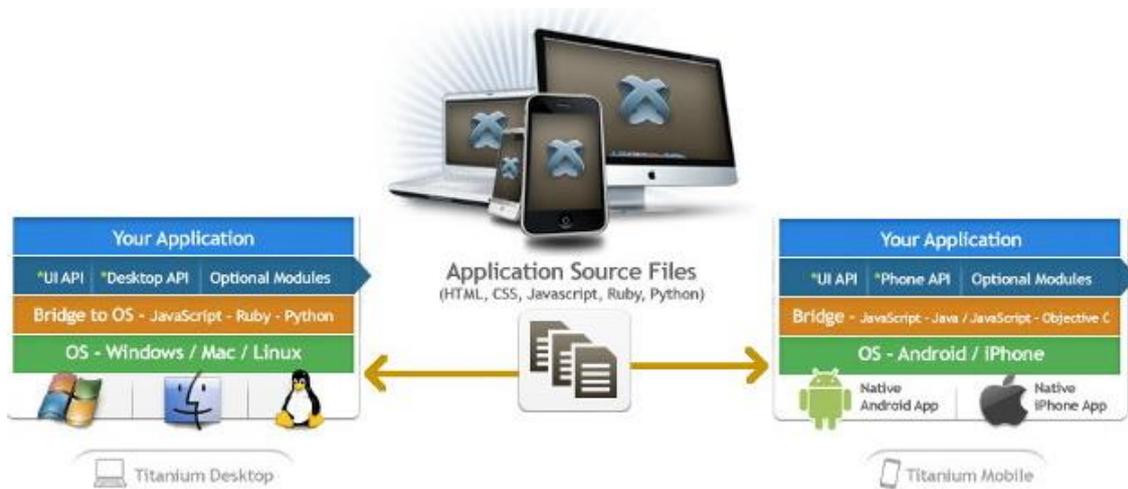


Figura 2.14 - Esquema completo da arquitetura do Titanium [40]

A Figura 2.14 acima apresentada representa a arquitetura do *framework* Titanium. A arquitetura consiste em duas partes principais: na primeira, as aplicações são “empacotados” juntamente com um interpretador JavaScript para executar o código da aplicação e na segunda, a biblioteca do Titanium oferece as APIs para as funcionalidades do dispositivo, como sensores, acesso ao sistema de ficheiros, componentes de interface nativo e outros [35].

2.4.2. Rhodes

O Rhodes é uma ferramenta de desenvolvimento de aplicações móveis multiplataforma desenvolvido pela **Rhobile**, que foi adquirida pela **Motorola Solutions Inc.** em **Massachusetts Institute of Technology (MIT)** em Outubro de 2011 [35], [43]. Foi desenvolvida com o objetivo de construir rapidamente aplicações nativas para todos os principais sistemas operativos móveis (iOS, Android, BlackBerry, Windows Phone e Symbian) [43]. O objetivo principal do Rhodes é fornecer um alto nível de produtividade e portabilidade na programação. O Rhodes é um ambiente de desenvolvimento móvel *open-source* baseado em Ruby [1].

O **RhoMobile** fornece um IDE chamado RhoStudio que é uma solução inovadora dedicada a utilizadores que querem desenvolver aplicações através de um IDE padrão [44], [43]. Esta solução pode ser utilizada nas plataformas Linux, Mac e Microsoft Windows. Alternativamente, RhoMobile oferece também a possibilidade de desenvolver aplicações com qualquer editor ou IDE, desde que tenha suporte ao HTML, CSS, JavaScript e Ruby [41]. Os editores mais populares são: o Eclipse, Visual Studio, Netbeans, IntelliJ e Textmate.

O Rhodes utiliza também o padrão **MVC (Model View Controller)** para desenvolver aplicações móveis. As linguagens utilizadas no elemento *view* são: o HTML, CSS e JavaScript para desenvolver aplicações móveis, enquanto no elemento *controller* a linguagem é Ruby, para fazer um suporte *backend*. Além disso, com a abordagem MVC é possível também escrever aplicações que utilizam apenas o elemento *view*. Obviamente, é realizável apenas para aplicações ou *Websites* que requerem um baixo nível de complexidade. O Rhodes fornece apenas três possibilidades para adicionar extensibilidades neste *framework*, o primeiro pode ser realizado ao adicionar bibliotecas externas da Ruby ao Rhodes, o segundo por criação de extensões nativas para um SDK de cada OS em específico e a última por fazer a extensão de *views* já existentes disponíveis no Rhodes [1].

Na Figura 2.15 a seguir apresentada corresponde a arquitetura do Rhodes. Os componentes **Controller**, **HTML templates** e **Source adapter** são as partes que os desenvolvedores de aplicações necessitam de implementar para a criação de aplicações, ao passo que os outros componentes são fornecidos pelo Rhodes, como por exemplo o **Rhodes App Generator** que é uma IDE que pode ser RhoStudio ou outro editor, o **Ruby Executor** que é o executador do código Ruby, o **Device Capabilities** que são as APIs, o **Rhom** que é a mini base de dados **ORM (Object Relational Mapper)**, o **RhoSync Client** que é a biblioteca que adiciona a capacidade de sincronização de dados para as

aplicações, e o *RhoSync* que simplifica o desenvolvimento da conectividade para aplicações empresariais de *backend* [1].

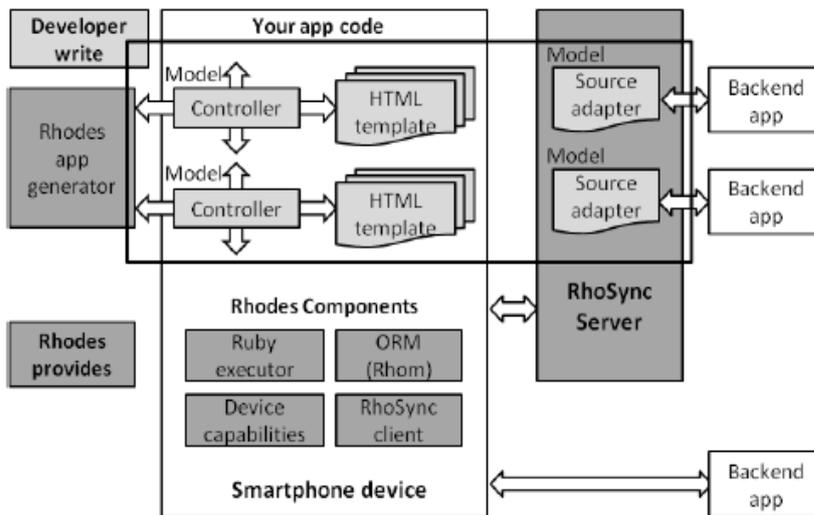


Figura 2.15 - Interface da arquitetura Rhodes entre o Smartphone e os componentes do Rhodes [1]

2.4.3. Phonegap

O PhoneGap é uma ferramenta de desenvolvimento móvel *open-source* desenvolvido pelo **Adobe System Inc.** sob a licença do Apache 2.0 [35], [43], [45]. Este permite aos programadores e empresas a desenvolverem aplicações grátis, comerciais e *open-source*, e dá-lhes também a possibilidade de utilizar qualquer combinação de licenças [1]. O ambiente de desenvolvimento é multiplataforma e permite a criação de aplicações para sistemas operativos Android, Bada, BlackBerry, iOS, Symbian, WebOs e Windows Phone [35].

O PhoneGap é uma solução útil para criar aplicações móveis utilizando linguagens de programação WEB modernos, tais como o HTML, HTML5, CSS, CSS3 e JavaScript, em vez de utilizar linguagens de programação menos conhecidos como o *Objective-C* ou outras linguagens [42], [43], [45]. Tem o benefício de trazer muitas vantagens para os programadores qualificados e especialmente para atrair programadores WEB.

Essencialmente, o PhoneGap é um “empacotador” que permite incluir aplicações escritos em linguagens de programação conhecidos, em aplicações nativas. As aplicações do PhoneGap são

híbridas, o que significa que não são puramente nativos e nem puramente baseado em WEB, ou seja, o fato de não ser puramente nativo advém do fato do *layout* da aplicação ser feita via *web-view* em vez da linguagem nativa do OS, e a falta de suporte do HTML em algumas funções, faz com que não seja puramente baseado em WEB [1].

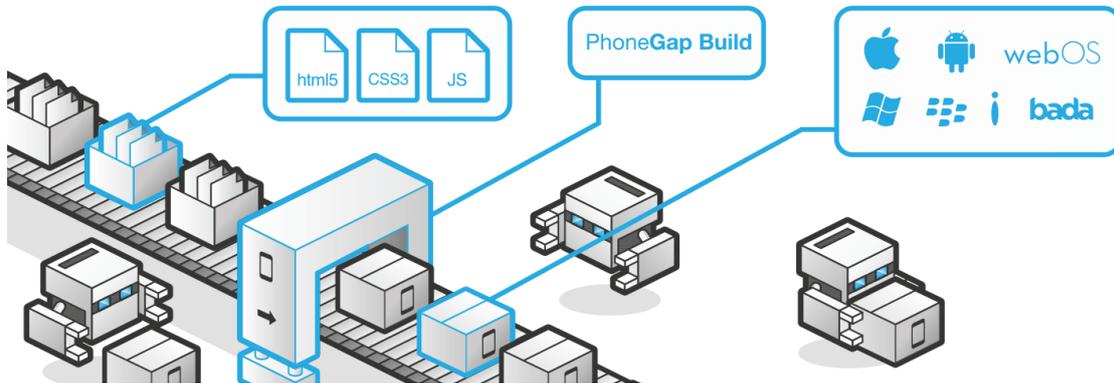


Figura 2.16 - Building PhoneGap App [45]

O PhoneGap não fornece uma IDE para desenvolvimento de aplicações, mas no entanto os programadores podem escrever o código fonte a partir de IDE's (ex. Eclipse para Android e XCode para iOS, etc...), ou então escrever o código num editor de texto (como por exemplo, o Sublime Text) e posteriormente compilar a partir dos comandos do PhoneGap, denominados **PhoneGap Build command** [42], [43]. Esta abordagem não permite que os programadores tenham um ambiente de desenvolvimento centralizado, de modo que o esforço necessário para compilar o código fonte e produzir a aplicação executável é alta.

A arquitetura do PhoneGap é constituída por 3 principais camadas: **WEB Application**, **PhoneGap** e **OS and native APIs**, como especificada na Figura 2.17 a seguir apresentada.

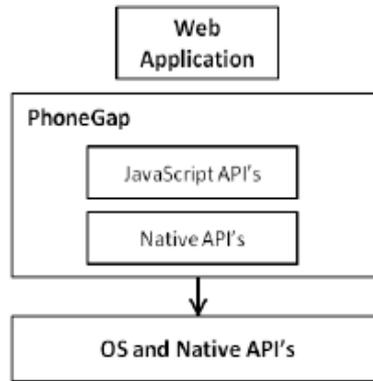


Figura 2.17 - Camadas de Interface da arquitetura do PhoneGap [1]

A camada superior representada na figura representa o código fonte da aplicação. A camada central é composta pelas APIs JavaScript e nativas e é a camada responsável pela interface entre as camadas **WEB Application** e **PhoneGap**. Para além disso, também cuida da interface entre as APIs JavaScript que são utilizadas pela aplicação e as APIs nativas de cada sistema operativo móvel. O PhoneGap fornece APIs JavaScript aos programadores de modo que tenham acesso às funcionalidades dos dispositivos móveis como o Acelerómetro, código de barras, Bluetooth, calendário, camera, contactos, conexão, arquivos, GPS, menu, NFC, etc.

A Figura 2.18 a seguir apresentada corresponde ao esquema da arquitetura do PhoneGap mais detalhada, fornecida pela IBM. Representa todos os componentes das camadas **WEB Application**, **HTML rendering engine**, **PhoneGap APIs** e **Mobile OS** [1].

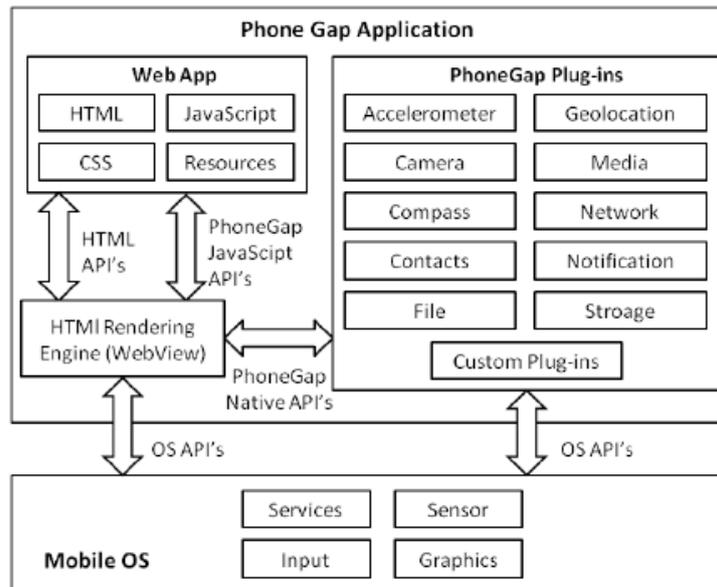


Figura 2.18 - Esquema completo da arquitetura do PhoneGap [1]

2.4.4. Sencha Touch

Sencha Touch é um *framework* MVC *open-source* desenvolvido pela **Sencha Inc.** em Junho de 2010. É um *framework* de aplicações móveis de alto desempenho, que permite desenvolver aplicações multiplataformas utilizando JavaScript. É compatível com Android, iOS, BlackBerry, Windows Phone e Tizen [37], [45], [46], [47].

2.4.5. JQuery Mobile

JQuery Mobile é um framework para desenvolvimento de interfaces do utilizador em dispositivos móveis, baseado no famoso jQuery e bibliotecas jQuery UI [46].

JQuery Mobile fornece um sistema de interface do utilizador unificado que funciona perfeitamente em todas as plataformas móveis. Este sistema inclui layouts (*lists, detail panes, overlays*) e um conjunto de controlos de formulário e *widgets* de interface do utilizador (*toggles, sliders, tabs*) [37].

Além de componentes da interface do utilizador, o jQuery Mobile oferece animações e suporte JavaScript para *touch events*. Este suporta todos os principais sistemas operativos móveis, como o Android, iOS, Windows Phone, BlackBerry OS [46].

2.4.6. Requisitos e Critérios de Seleção

Pretende-se elaborar e levantar uma lista de critérios de modo a avaliar as melhores soluções relativamente aos *frameworks* de desenvolvimento. Estes critérios foram escolhidos para ajudar os programadores a perceberem qual ferramenta poderia ser apropriado para as suas finalidades.

Os critérios a levar em consideração na escolha do *framework* apropriado são:

- **Sistema operativo móvel suportado** para perceber os possíveis efeitos nos respetivos modelos de negócio;
- **Licenças dos *frameworks*** oferecidos para avaliar as condições e termos de utilização;
- **Linguagens de programação** oferecida aos programadores para a criação de aplicações;
- **Disponibilidades de APIs** fornecidas com o objetivo de ter uma ideia das diferentes partes do *Hardware* acessíveis no sistema operativo;
- **Acessibilidade às APIs nativas** para comparar como é possível acedê-los em cada *framework*;
- **Arquitetura** fornecida ao processo de desenvolvimento da aplicação;
- **Ambiente de desenvolvimento integrado** disponível para desenvolver aplicações.
- **Desempenho (Utilização do CPU, da memória e consumo de energia)** para comparar melhores e piores desempenhos em aplicações desenvolvidas em cada *framework* [1].

Segundo o relatório do estudo realizado pelo **research2guidance** em 2014, os principais critérios definidos (ordenados por nível de importância) pelos desenvolvedores de aplicações móveis multiplataformas foram os seguintes (Figura 2.19):

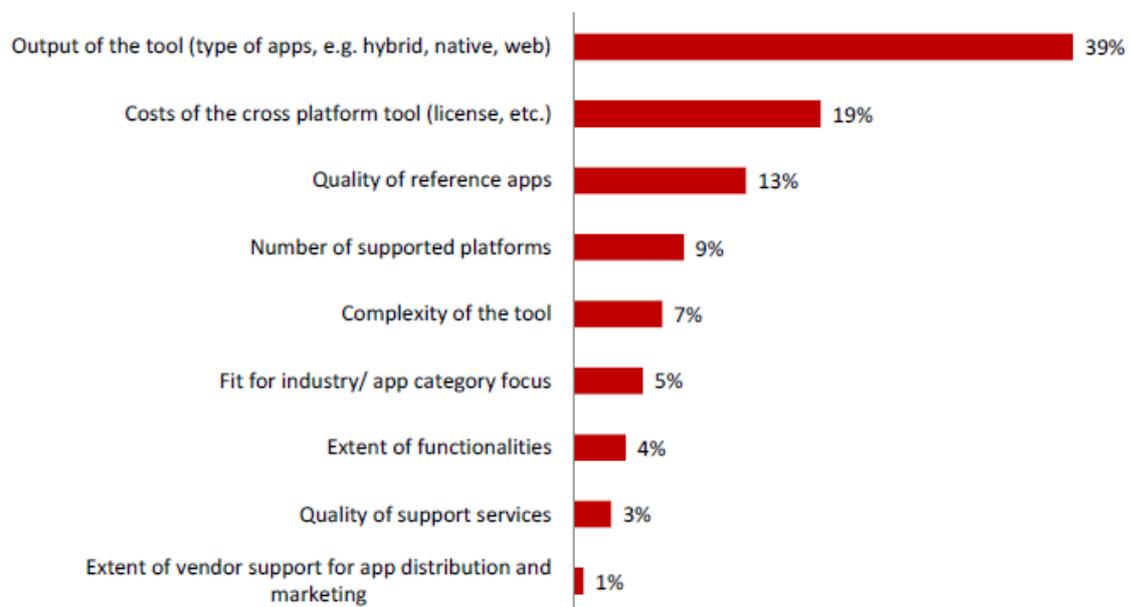


Figura 2.19 - Principais critérios de seleção de frameworks [39]

2.4.6.1. Compatibilidade com plataformas móveis

Na tabela a seguir, mostra os diferentes tipos de sistemas operativos móveis suportados por cada *framework*.

Tabela 2.11 - Comparação entre a compatibilidade de plataformas móveis e o suporte os do ambiente de desenvolvimento [1], [33], [43]

Framework	Suporte OS móvel	Suporte OS
Rhodes	Android, BlackBerry, iOS, Symbian, Windows Mobile, Windows Phone	Linux, Mac, Windows
PhoneGap	Android, BlackBerry, iOS, Symbian, WEBOS, Bada, Windows Phone	Linux, Mac, Windows
Titanium	Android, BlackBerry, iOS, Windows Phone, Tizen	Linux, Mac, Windows
Sencha Touch	Android, BlackBerry, iOS, Windows Phone, Tizen	Linux, Mac, Windows
jQuery Mobile	Android, BlackBerry, iOS, WEBOS, Windows Phone	Linux, Mac, Windows

De acordo com a Tabela 2.11, é fácil de perceber que o PhoneGap oferece mais compatibilidades para o desenvolvimento em diferentes sistemas operativos móveis, o que acaba por ser ótimo aos desenvolvedores e proprietários do modelo de negócio, de modo que permite ganhar um máximo de lucro [43].

2.4.6.2. APIs nativas disponibilizadas

A Tabela 2.12 descreve varias características dos *frameworks*, como as linguagens de programação suportadas, o acesso às APIs nativas, o IDE e *plug-in*.

O acesso às APIs nativas é um fator de grande importância, de modo que possibilita que os programadores tenham acesso aos recursos nativos do sistema operativo móvel no qual pretende-se desenvolver a aplicação.

Tabela 2.12 - Comparação das características de desenvolvimento [1], [33], [43]

Framework	Linguagem programação	Acesso as APIs nativas	IDE	Plug-in
Rhodes	HTML, HTML5, CSS, JavaScript	JavaScript	RhoStudio, RhoHub, *	Sim
PhoneGap	HTML, HTML5, CSS, CSS3, JavaScript	JavaScript	IDE nativo do sistema operativo (ex. Eclipse, Xcode)	Sim
Titanium	JavaScript, TSS	JavaScript	Titanium Studio	Sim
Sencha Touch	HTML, HTML5, CSS, CSS3, JavaScript	JavaScript *	Sencha Touch Architecture, *	Sim
jQuery Mobile	HTML, HTML5, CSS, CSS3, JavaScript	N.A.	*	Sim

*Outras IDEs alternativas, como Eclipse, Visual Studio, Netbeans, IntelliJ, Textemate, Sublime, etc.

Tabela 2.13 - Comparação das principais APIs suportadas [1], [33], [43]

Nome da API	Rhodes	PhoneGap	Titanium	Sencha Touch	jQuery Mobile
Acelerómetro	X	X	X	-	-
Código de Barras	X	-	X	-	-
Bluetooth	X	X		-	-
Calendário	X	X	X	-	-
Camera	X	X	X	-	-
Captura	X	X	X	-	-
Conexão	X	X	X	X	-
Contacto	X	X	X	-	-
Dispositivo	X	X	X	-	-
Ficheiro	X	X	X	-	-
Geolocalização	X	X	X	X	-
Menu	X	-	X	-	-
NFC	X	X	X	-	-
Notificação	X	X	X	X	-
Rotação do ecrã	X	X	X	X	-
Armazenamento	X	X	X	X	-

OBS: As APIs suportadas acima indicadas, não correspondem a todas as plataformas móveis suportadas pelos frameworks de desenvolvimento, ou seja, nem todas as plataformas móveis tem suporte a estes recursos nativos.

2.4.6.3. Licenças

A licença do *framework* é outro parâmetro útil para comparação. Como mostrado na Tabela 2.14, as licenças disponíveis para estes *frameworks* são MIT, Apache 2.0, GNU General Public License 2 (GPL2) e comercial. Todas as licenças apresentadas são gratuitas e além disso oferecem um suporte *open-source*, ou seja, grátis. Os *frameworks* Rhodes, Titanium e Sencha Touch são

frameworks com suporte MVC. Uma dos principais benefícios do padrão MVC é a possibilidade de desenvolver aplicações de forma distribuída a partir do *model*, *view* e *controller*, de modo a separar uma parte da outra, ou seja, a vantagem é a possibilidade de modificar cada parte de forma independente. Outro benefício que oferece o padrão MVC reside no fato de fornecer um ambiente que incorpora diferentes tecnologias em diferentes localizações e uma arquitetura com melhor suporte a escalabilidade. Relativamente ao tipo de abordagem de desenvolvimento, o *framework* Rhodes segue a abordagem compilação-cruzada, o PhoneGap a abordagem híbrida, o Titanium a abordagem interpretada e tanto o Sencha Touch como o jQuery seguem a abordagem WEB [1].

Tabela 2.14 - Comparação das características gerais [1], [33]

Framework	Licença	<i>Open-source</i>	Abordagem	MVC
Rhodes	MIT, comercial	Sim	Compilação-Cruzada	Sim
PhoneGap	Apache 2.0	Sim	Híbrido	Não
Titanium	Apache 2.0, comercial	Sim	Interpretada	Sim
Sencha Touch	GPL v3	Sim	WEB	Sim
jQuery Mobile	GPL/MIT	Sim	WEB	Não

2.4.6.4. Desempenho

Para examinar o desempenho, foram desenvolvidos aplicações de teste para o sistema operativo móvel Android utilizando *frameworks* de desenvolvimento multiplataforma. Os *frameworks* utilizados para o desenvolvimento destas aplicações foram:

- PhoneGap;
- PhoneGap + jQuery mobile;
- PhoneGap + Sencha Touch;
- Titanium.

As aplicações desenvolvidas para examinar o desempenho tem as seguintes características:

- A interface do utilizador tem botões em que cada um solicita um tipo de pedido. Cada pedido é efetuado a uma *WebService* (disponível na internet) a partir de tecnologias AJAX, REST e SOAP;
- Analisar e apresentar os diferentes formatos de respostas: Text, XML, JSON [42].

2.4.6.4.1. Consumo da CPU

Para medir a utilização da CPU, as aplicações foram desenvolvidas utilizando o PhoneGap e outros *frameworks*. O segmento de código para medir a utilização da CPU são adicionados ao Android *activity* através do PhoneGap. Mas no entanto, nenhuma aplicação pode ser desenvolvido utilizando o *framework* Titanium, uma vez que não permite adicionar o *activity* [42]. Poderia ser desenvolvido um *plugin* para este efeito, mas é um processo demorado. São seguidos dois caminhos diferentes para registar a utilização da CPU:

- A primeira abordagem, efetuasse uma captura instantânea da CPU em cada estado do ciclo de vida da *Activity* (ex. *onCreate*, *onStart*, *onPause*, *onStop* e *onDestroy*) das aplicações.
- A outra abordagem é ler o resultado “*top*” a cada segundo durante todo o ciclo de vida das aplicações. Em seguida, calculasse a média para cada estado do ciclo de vida do *Activity*.

As aplicações desenvolvidas executam as mesmas funcionalidades dos testes anteriores. A Tabela 2.15 a seguir retrata os resultados do teste da utilização da CPU [42].

Tabela 2.15 - Métrica da utilização da CPU [42]

Framework utilizado	Utilização da CPU a partir da abordagem da captura instantânea	Utilização da CPU a partir da abordagem do comando “ <i>top</i> ”
PhoneGap + HTML + CSS	81.92771%	Max: 10% Min: 0% Média: 2%

PhoneGap + jQuery + HTML	80.26316%	Max: 42% Min: 0% Média: 10%
PhoneGap + Sencha Touch	44.0%	Max: 32% Min: 0% Média: 8%

Os valores obtidos a partir da abordagem de captura instantânea é calculado quando a aplicação encontrasse a fazer muita computação e é assim representada utilizando valores muito altos. Também é de notar que estes valores podem variar muito em um milésimo de segundo para outra, uma vez que são instantâneos na utilização da CPU em cada pequeno período de tempo.

Os valores obtidos a partir do resultado “*top*” mostra variadas utilizações da CPU. O valor mínimo é sempre zero, uma vez que a aplicação obtém a página requisita e apresenta-o, a aplicação não utiliza qualquer espera de CPU para a próxima entrada do sistema. O valor médio é calculado utilizando o tempo total decorrido. A partir desta abordagem, é claro que a primeira aplicação utilize muito menos CPU, mas a interface do utilizador não é muito sofisticada (não parecida com a UI nativa). Quando é utilizando o Sencha Touch com o PhoneGap, a utilização da CPU é maior, mas no entanto a UI é significativamente melhor (UI nativa simulada, ou seja, tem a interface do utilizador parecido à nativa) [42].

2.4.6.4.2. Consumo da Memória

A informação da utilização da memória é obtida a partir da ferramenta DDMS do ADT.

- ✓ **PSS (Proportional set size)** – PSS é a quantidade de memória partilhada com outros processos. Esta é a memória que não seria libertado caso um processo terminar, mas é indicativo da quantidade que este processo está a contribuir para memória total carregada;
- ✓ **USS (Unique set size)** - USS é um conjunto de páginas exclusivas de um processo. Esta é a quantidade de memória que seriam libertado caso a aplicação encerrasse [42].

Tabela 2.16 - Métrica da utilização da memória [42]

Framework utilizado	PSS	USS
PhoneGap	12091	6036
PhoneGap + jQuery Mobile	14730	9424
PhoneGap + Sencha Touch	24526	20164
Titanium	17500	8676

De acordo com a Tabela 2.16 acima apresentado, pode-se verificar que a aplicação desenvolvida com o *framework* PhoneGap tem menor valor de **PSS** e **USS**. Isto é devido ao fato do PhoneGap ser projetado para não utilizar nenhum elemento de estilo ou ferramenta para o aperfeiçoamento da UI.

Por outro lado, a integração do PhoneGap com o jQuery Mobile ou com o Sencha Touch corresponde a um ambiente completo para uma melhor UI no desenvolvimento de aplicações. Neste caso, a memória utilizada aumenta com a utilização de ficheiros HTML e JavaScript. O Titanium dispõe de um SDK completo e a utilização da memória é portanto, elevada [42].

2.4.6.4.3. Consumo de Energia

O consumo de energia das aplicações móveis tem recebido muita atenção por parte dos pesquisadores atualmente. Para utilizar eficazmente a bateria do dispositivo móvel, as aplicações desenvolvidas utilizando *frameworks* multiplataforma devem ser eficientes de energia, ou seja, gastar menos energia possível. A mediação do consumo de energia das aplicações foi obtida utilizando o “*Power Tutor*”. O *Power Tutor* é uma aplicação Android muito popular que relata o consumo de energia de cada aplicação individualmente em um dispositivo móvel. A Tabela 2.17 a seguir apresentada representa os resultados, e estes valores correspondem a média do consumo de energia das aplicações [42].

Tabela 2.17 - Métrica do consumo de energia [42]

Framework utilizado	Consumo de energia (mW)
PhoneGap + HTML + CSS	107
PhoneGap + jQuery + HTML	168
PhoneGap + Sencha Touch	120

Mais uma vez, o resultado mostra que a primeira aplicação consome menos energia entre as três aplicações. Isto reside no fato desta primeira aplicação ter uma UI muito simples. A aplicação desenvolvida com PhoneGap e Sencha Touch funciona de forma eficiente, mesmo consumindo 120mW de energia [42].

2.4.7. Arquitetura Geral do desenvolvimento de Aplicações Multiplataformas

Desenvolver uma aplicação que execute em múltiplas plataformas como iOs, Android, RIM, Symbian, WP7 e Bada ou outros, pode ser uma tarefa difícil considerando o número de implementações tecnológicas que vem em cada plataforma móvel. Mas foram feitos esforços para fazer a execução de aplicações multiplataformas uma possibilidade [42]. A Figura 2.20 a seguir apresentada mostra a arquitetura geral do desenvolvimento de aplicações multiplataformas.

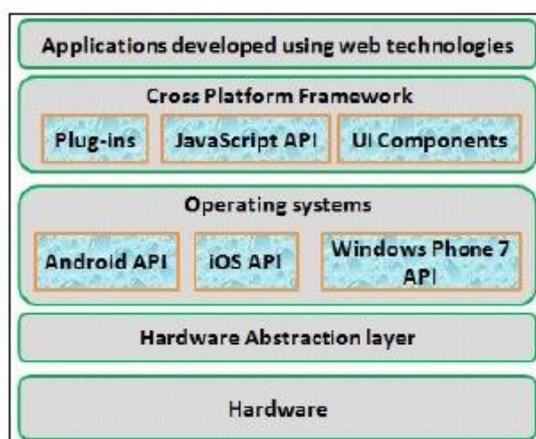


Figura 2.20 - Arquitetura geral do desenvolvimento de aplicações multiplataforma [42]

O programador implementa a lógica de negócio ou as funcionalidades da aplicação utilizando tecnologias WEB. Os *frameworks* multiplataformas permitem a implementação da interface do utilizador, fácil acesso ao armazenamento e os recursos do dispositivo móvel (sensores, camera e contatos) que interagem com APIs JavaScript. Estas, por sua vez, interagem com APIs nativas das plataformas móveis [42].

Os *frameworks* de desenvolvimento de aplicações móveis multiplataformas estão cada vez mais sofisticados, de modo que também tem melhorado o desempenho das aplicações e a tem-se efetuado esforços de para obter uma UI quase tão bom como à nativa.

2.4.8. Padrão MVC

O padrão *Model-View-Controller*, mundialmente conhecido por MVC, bastante utilizado no mundo das aplicações WEB, foi introduzido a fim de gerir o aumento da complexidade destas aplicações. Este começou como um *framework* desenvolvido pelo **Trygve Reenskaug** para a plataforma **Smalltalk** no final da década de 1970. Desde então, tem desempenhado um papel fundamental na maioria dos *UI-frameworks* (*frameworks* de desenvolvimento da interface do utilizador) [48]. O padrão MVC promove a separação da lógica da aplicação e da apresentação, e consiste em decompor a aplicação em três componentes que comunicam entre si (*model*, *view* e *controller*) [49], [50]. Cada um destes componentes tem um conjunto de tarefas específicas a desempenhar. A Figura 2.21 a seguir apresentada consiste em ilustrar a estrutura deste padrão, bem como a interação entre os componentes do mesmo.

1. **Model** – efetua o gerenciamento dos dados do domínio da aplicação e do seu comportamento, responde aos pedidos de informação do seu estado (geralmente efetuado pela *view*) e responde às instruções para alteração do estado (geralmente efetuado pelo *controller*);
2. **View** – efetua o gerenciamento da apresentação da informação ao utilizador;
3. **Controller** – escuta e interpreta as entradas dos utilizadores (a partir do teclado, rato ou gestos), e posteriormente instruir o *model* ou o *view* a executar ações baseadas nestas entradas [49], [50], [51], [52].

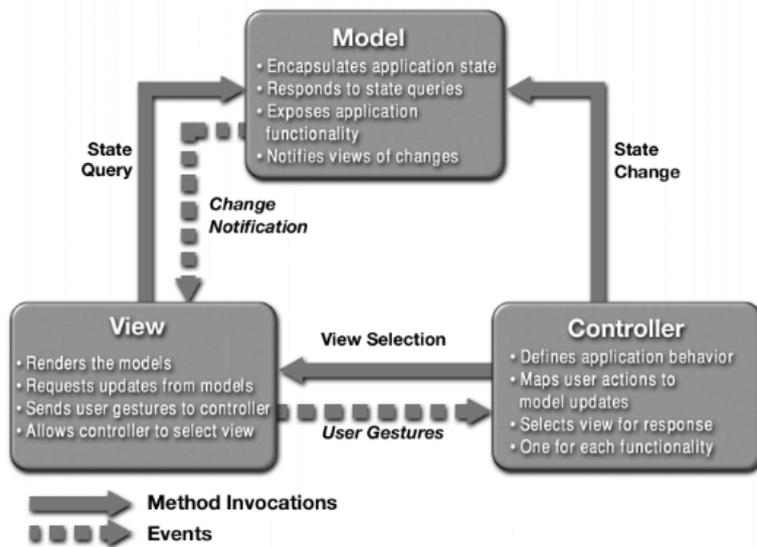


Figura 2.21 - Padrão MVC [53]

Por outras palavras, o *model* representa os dados no sistema, o *view* é alguma forma de visualização do estado do *model* (é possível que um *model* tenha vários *views*), e o *controller* oferece facilidades (a partir dos *event handlers*) na alteração do estado do *model*[49]. É importante notar que, tanto a *view* como o *controller* dependem do *model* e o *model* não depende da *view* e nem do *controller*. Esta é um dos principais benefícios desta separação. Esta separação permite que o *model* seja contruído e testado independentemente da apresentação visual [51], [52]. Entre as diversas vantagens deste padrão estão a separação entre a apresentação e lógica da aplicação, a fácil manutenção do sistema, reaproveitamento do código, alteração na *view* sem afetar o *controller*, entre outras.

2.4.9. Web Services

Os *Web Services* são aplicações WEB com capacidade para interagir entre si, permitindo a automatização de tarefas que só podiam ser feitas através da interação dos humanos, ou seja, uma norma que define formas de interação aplicação-aplicação recorrendo a formatos abertos

[54], [55]. A utilização de protocolos normalizados proporciona a capacidade de intercâmbio de informação entre aplicações em ambientes eminentemente heterogéneos.

Ao longo de vários anos outras iniciativas tem vindo a serem desenvolvidas no âmbito dos sistemas distribuídos resultando em diversos protocolos como DCOM (*Distributed Component Model*), CORBA (*Common Object Request Broker*) e J2EE (*Java 2 Platform, Enterprise Edition*). No entanto estas soluções são proprietárias, de difícil implementação e caras, o que, num ambiente heterogéneo como a Internet, as torna inadequadas. Os Web Services, por outro lado, aparecem como uma plataforma independente para suporte ao desenvolvimento de aplicações distribuídas sobre Internet respondendo a todas as questões [56].

Sendo aplicações modulares, auto descritivas, acessíveis através de um URL, independentes das plataformas de desenvolvimento e que permitem a interação entre aplicações sem intervenção humana, os *Web Services* apresentam-se como a solução para os atuais problemas de integração de aplicações. Estas suas características devem-se em grande parte ao facto de se basearem em normas *standard*, de entre as quais se destacam: XML, SOAP, WSDL e UDDI [55],[57].

1. **XML** - metalinguagem de anotação na qual estão definidas todas as outras normas que servem de base aos *Web Services*.
2. **SOAP** - linguagem de anotação com a qual se pode descrever o protocolo de comunicação, responsável pela troca de mensagens de e para os *Web Services* (uma mensagem SOAP é um documento XML).
3. **WSDL** - linguagem de anotação definida em XML e que tem como objetivo descrever a API de um *Web Service*.
4. **UDDI** - linguagem de anotação definida em XML com a qual se cria a meta informação característica de um *Web Service*; vários registos UDDI são agrupados em repositórios que possuem uma interface/API de pesquisa para permitir a uma aplicação cliente pesquisar e localizar um serviço [55].

2.4.9.1. SOAP

O SOAP acrônimo de *Simple Object Access Protocol*, é uma linguagem XML que permite descrever as mensagens trocadas entre duas aplicações. Baseia-se em normas como o *XML Schema* e *XML Namespace*, que lhe confere independência em relação ao sistema operativo, linguagem de programação e aplicação. Este protocolo permite uma disponibilização de serviços estruturados, sob a forma de *WSDL (Web Services Description Language)*, tendo como vantagens uma fácil comunicação através de HTTP e Firewall, assim como a sua versatilidade e independência de plataformas e linguagens. Embora existam algumas limitações como a performance em grandes dimensões de dados na troca de mensagens, quando comparado com outros protocolos. O SOAP troca informação, através de mensagens (SOAP Messages), sobre uma estrutura de envelopes onde na origem é construído um envelope devidamente estruturado e no destino é extraída a informação. Os envelopes são constituídos em duas partes principais designadas por *header* e o *body*, onde apenas o *body* é obrigatório. Ambas as partes são constituídas por um ou mais blocos que assumem uma estrutura definida pelo WSDL [57], [58].

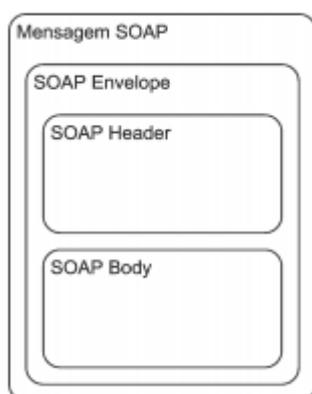


Figura 2.22 - Estrutura de uma mensagem SOAP [55], [58]

O SOAP assume a existência de uma entidade que envia o envelope e outra que recebe. Entre este processo podem existir intermediários que reenviam o envelope, onde podem processar o *header* caso exista. As mensagens SOAP apresentadas na Figura 2.22 anteriormente apresentada são estruturadas através de documentos XML que contêm a informação necessária para as entidades que comunicam. No SOAP *body* é possível adicionar todo o tipo de informação que deverá estar descrita no XML Schema de forma a validar o conteúdo

2.5. Conclusão

Neste capítulo efetuou-se a revisão da literatura de alguns aspetos e conceitos importantes relacionados com o desenvolvimento de aplicações móveis. Esta revisão da literatura permitiu concluir que a seleção de uma abordagem de desenvolvimento multiplataforma depende principalmente da exigência da aplicação, mas também das plataformas alvos, do tipo de aplicação do acesso aos dados e recursos (hardware), da interface do utilizador e aparência, do desempenho e da distribuição no mercado. A seleção da abordagem de desenvolvimento multiplataforma encontra-se interligada a seleção de um *framework* de desenvolvimento multiplataforma, ou seja, a seleção de uma abordagem de desenvolvimento multiplataforma obriga-nos a selecionar um *framework* de desenvolvimento dentro desta mesma abordagem de desenvolvimento e caso for efetuado a seleção de um *framework* de desenvolvimento, estaremos a seguir uma dada abordagem onde este *framework* se insere.

Concluindo, de acordo com a análise, a seleção da abordagem de desenvolvimento multiplataforma deve ser efetuado o tipo de aplicação no qual pretende-se desenvolver, ou seja, deve-se selecionar em função dos vários fatores que encontram relacionados com os critérios anteriormente descritos.

3. Requisitos e Arquitetura

3.1. Introdução

O desenvolvimento de aplicações e de *Software* em geral tem um conjunto de riscos associados, e para mitigá-los é necessário efetuar uma gestão destes riscos. Um risco é qualquer ameaça à realização de um, ou mais, objetivos fundamentais do produto final. O risco faz parte de qualquer atividade humana, não podendo nunca ser eliminado. Este por si só, não é mau, de modo que é essencial ao progresso, e o insucesso constitui, muitas vezes, numa componente fundamental de aprendizagem. No entanto, devemos aprender a equilibrar as possíveis consequências negativas do risco, com os benefícios potenciais da respetiva oportunidade associada. Para a gestão do risco é necessário identificar (descobrir os riscos específicos associados ao desenvolvimento do software), avaliar (analisar os atributos dos riscos – probabilidade e impacto), planear a resposta (decidir o que fazer com os riscos), monitorizar (medir a eficácia das medidas tomadas) e controlar o risco (agir sobre os riscos, consoante os resultados da monitorização). Segundo **Charette**, a gestão do risco não diz respeito a decisões futuras, mas sim a consequências futuras das decisões atuais [59]. O desenvolvimento de *Software* envolve um número elevado de áreas de risco, como ilustrado na Figura 3.1 a seguir apresentada.

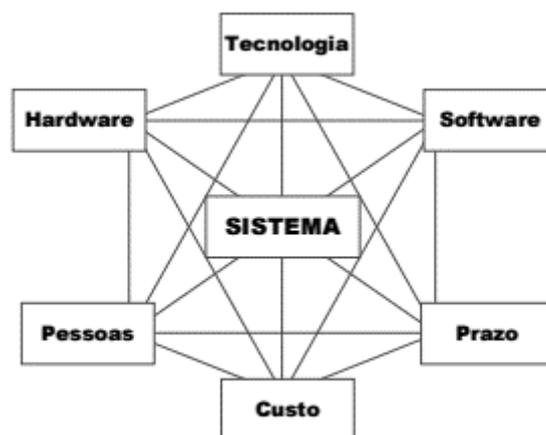


Figura 3.1 - Riscos no contexto de desenvolvimento de *software* [59]

Ao longo deste capítulo são abordados vários temas relacionados com o conhecimento do domínio do problema. Em primeiro lugar é efetuado um estudo sobre a engenharia de requisitos e a

execução do modelo de processo de engenharia de requisitos. A engenharia de requisitos é um processo de descobrir, analisar, documentar e verificar os requisitos do sistema a desenvolver e as suas restrições. Logo a seguir segue-se com o estudo dos SDUM, bem como o estudo de todos os processos e serviços prestados pelos mesmos. Estes serviços prestados são: serviços de empréstimos, renovação e reserva de publicações e pretende-se especificar todos esses serviços particularmente. É de real importância estudar a estrutura da organização (organograma), e de igual modo a sua área de ação. Por fim, é ilustrado a arquitetura da solução, que resume o modelo e os componentes desenvolvidos.

3.2. Engenharia de Requisitos

A Engenharia de requisitos é tradicionalmente conhecida como análise de sistema. É o processo de descobrir, analisar, documentar e verificar os requisitos de um sistema e as suas restrições, ou seja, é um processo de identificação das características do sistema a desenvolver, de assegurar que essas características correspondem aos objetivos do negócio e por fim verificar se o sistema desenvolvido satisfaz ou não as características identificadas. Um requisito é uma característica do sistema, ou a descrição de algo que o sistema é capaz de efetuar para satisfazer os seus objetivos. Os requisitos versam sobre o espaço do problema (**o quê**), e não sobre o espaço da solução (**o como**), contudo poderá haver casos que os requisitos colocam restrições ao espaço da solução.

A engenharia de requisitos é uma atividade crítica no processo de desenvolvimento de sistemas, por ser uma etapa inicial e cujas falhas terão efeitos e cadeia nas etapas subsequentes assim como no produto final. A determinação incorreta dos requisitos levará à obtenção e disponibilização de sistemas informáticos inadequados aos sistemas de informação e ao sistema organizacional [60].

A Figura 3.2 a seguir apresentada o modelo do processo de engenharia de requisitos. A um nível de abstração grande, um processo de engenharia de requisitos pode ser descrito como a obtenção/definição, análise e negociação, documentação e validação de requisitos.

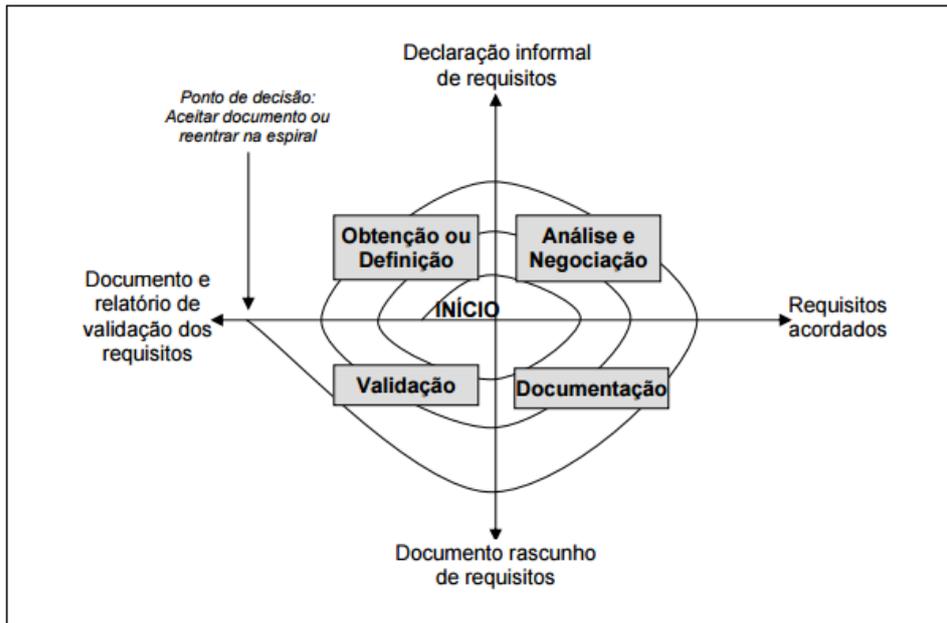


Figura 3.2 - Modelo do processo de engenharia de requisitos [60]

Esta figura mostra que as diferentes atividades na engenharia de requisitos devem ser repetidas até que seja tomada a decisão de aceitação do documento de requisitos. Se for encontrado um problema no rascunho documento de requisitos, reentra-se na espiral. Isto continua até ser produzido um documento aceitável ou até fatores externos tais como pressões de calendarização ou falta de recursos forcem o término do processo de engenharia de requisitos [60].

A Tabela 3.1 a seguir apresentada descreve as fases do modelo de processo de engenharia de requisitos, bem como as suas técnicas e ferramentas.

Tabela 3.1 - Processos de Engenharia de Requisitos [35]

Fases	Descrição	Técnicas e Ferramentas
Obtenção/ Definição/ Levantamento	Após a identificação dos <i>stakeholders</i> identificam-se os requisitos que o sistema tem de satisfazer	Entrevistas, inquéritos, cenários, <i>brainstorming</i> em <i>workshops</i> , protótipos, simulações, <i>focus groups</i> , categorização de <i>stakeholders</i> , etc...
Análise e Negociação	Garantir a unicidade, consistência e completude dos requisitos,	Mapas relacionais, diagrama de contexto, tabela evento-resposta,

	identificando anomalias, e procurando resolvê-las; priorização dos requisitos	casos de uso, modelização de dados, priorização de requisitos
Documentação	O registo da documentação deve descrever, além dos requisitos, o <i>background</i> do sistema, o domínio e contexto do problema, um glossário, a descrição dos <i>stakeholders</i> e qualquer outra informação relevante.	Templates, <i>layouts</i> , linguagem natural, linguagem natural controlada, linguagem formal, regras de documentação, estruturação da documentação, etc...
Validação	Garantir que a documentação e especificação representam de um modo preciso as necessidades dos <i>stakeholders</i> ; avaliar a perenidade lógica dos requisitos	Inspeções, análises formais, animações, simulações, protótipos, etc...
Gestão de Requisitos	Garantir a rastreabilidade dos requisitos; analisar a maturidade e estabilidade dos requisitos; gerir mudanças (eliminação, alteração ou adição) de requisitos durante a fase de desenvolvimento ou manutenção do sistema, de modo a minimizar o impacto e risco que daí advém.	Políticas e procedimentos de controlo de mudança, definir atributos dos requisitos, matrizes de rastreabilidade, etc...

3.2.1. Definição/Levantamento de Requisitos

Num processo de engenharia de requisitos geralmente reconhecem-se dois grandes tipos de requisitos: funcionais e não funcionais. Os requisitos funcionais representam as funções do sistema e o que este deverá fazer, e os requisitos não funcionais representam os atributos do sistema e as qualidades gerais do mesmo [60].

Sugere-se que para a engenharia de requisitos sejam considerados três objetivos principais: “antecipação, investigação e especificação de requisitos”. A antecipação diz respeito aos requisitos que são antecipáveis pela equipa de desenvolvimento em funções dos seus *know-how* e experiência, a investigação é o ponto fulcral da análise de requisitos, visto que cuida de utilizar ferramentas e técnicas para explorar e documentar os requisitos e por fim a especificação de requisitos diz respeito à especificação dos requisitos, ou seja, a descrição das características desejadas no novo sistema.

O processo de levantamento de requisitos é iniciado com a compreensão do domínio da aplicação para posteriormente identificar o problema a ser resolvido no contexto do negócio da organização, de acordo com as necessidades e restrições dos *stakeholders* do sistema, como mostra a figura a seguir apresentada [60].

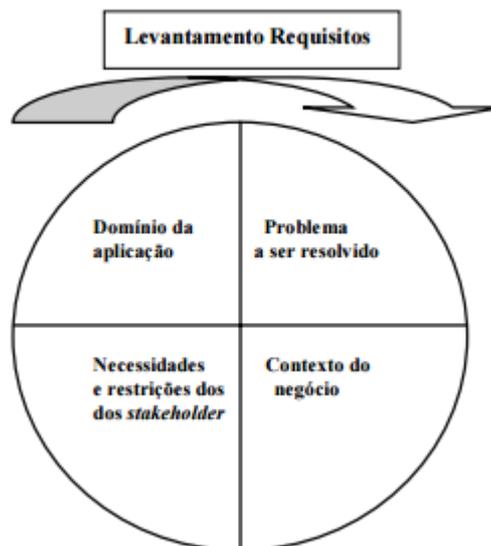


Figura 3.3 - Domínio da aplicação (Levantamento de requisitos) [61]

Nesta dissertação, para o levantamento de requisitos foram utilizados três técnicas: *Workshop*, *brainstorming* e prototipagem.

O *Workshop* consiste numa reunião para recolha de requisitos em que participam os engenheiros de requisitos e intervenientes com interesse no sistema (*stakeholders*) que representam a organização e o contexto em que o sistema será utilizado. A lógica do *workshop* inclui explicar o método, explicar os objetivos, mostrar o conjunto inicial de requisitos (*draft*), encorajar a crítica construtiva e fazer de imediato as correções à lista de requisitos [60].

O *Brainstorming* é uma técnica para gerar e explorar ideias provenientes de várias fontes. É utilizado normalmente em *workshops*. Os princípios desta técnica resumem-se em:

- Todas as ideias são potencialmente boas. Todas as ideias são iguais, uma vez que encerram um determinado potencial para conduzir a uma solução inesperada;
- Deixar fluir o pensamento em primeiro lugar e ponderar depois, ou seja, as ideias devem ser expressas sem qualquer inibição;
- Dar a palavra a cada pessoa, de forma rotativa, para que expresse a sua ideia;
- Um ambiente descontraído e sem tensões favorece a geração de novas ideias.

O *Brainstorming* tem por intenção alargar as fronteiras do espaço do problema dos participantes e obter soluções não convencionais (pela manipulação de “quadros de experiência” individuais ou definições internas contextuais de eventos e situações) para permitir aceder as informações e ideias de outra forma indisponíveis por causa dos seus “quadros” da situação atual [60].

A construção de protótipos é um processo que possibilita a criação de modelos de *Software*, os quais podem assumir uma das seguintes formas:

- Construção de um protótipo em papel ou de um modelo que descreva a interação homem-máquina, mostrando ao utilizador a perspetiva dessa interação;
- Construção de um protótipo que implemente alguns subconjuntos da aplicação desejada;
- Melhoria de um programa já existem com o objetivo de lhe adicionar características específicas da nova aplicação.

A prototipagem pode ser entendida como uma forma de simular rapidamente a aplicação e garantir nas diferentes fases de desenvolvimento a coerência da informação. Durante a análise de

requisitos, a prototipagem dá forma concreta aos requisitos e assiste à sua clarificação e determinação [60].

3.2.1.1. Requisitos Funcionais

Os requisitos funcionais descrevem as funções, tarefas e subtarefas que se espera que o sistema realize. Incluem tudo que os utilizadores e engenheiros de requisitos esperam que o sistema faça. A identificação e definição de requisitos funcionais não é um exercício de como o sistema suportará as funções, atividades e tarefas, mas sim um exercício detalhado de como o sistema contemplará e perceberá os utilizadores [60].

O levantamento de requisitos funcionais pode ser consultada com maior nível de detalhe no documento de especificação de requisitos no Anexo B – Documento de Especificação de Requisitos.

3.2.1.2. Requisitos não funcionais

Os requisitos não funcionais são aqueles requisitos que não dizem especificamente respeito às funcionalidades de um sistema, ou seja, colocam restrições no produto a desenvolver e no processo a utilizar, e especificam as restrições externas às quais o produto deve ir de encontro. Os requisitos não funcionais incluem, entre outros, requisitos de fiabilidade, segurança, adaptabilidade, portabilidade e desempenho. Por vezes há necessidade de sacrificar requisitos funcionais para ir de encontros aos não funcionais, nomeadamente por limitações de tecnologia [60].

O levantamento de requisitos não funcionais pode ser consultada com maior nível de detalhe no documento de especificação de requisitos no Anexo B – Documento de Especificação de Requisitos.

3.2.2. Análise e Negociação

O principal objetivo deste processo é realizar a análise de requisitos e formular descrições dos requisitos para que estes possam ser interpretados da maneira mais clara pelos *stakeholders* envolvidos no projeto. Este processo pode ser feita em paralelo com o processo de definição de requisitos, pois estes dois processos encontram-se ligadas. Durante a definição de requisitos, alguns problemas podem ser descobertos, tais como requisitos incompleto, ambíguos, entre outros, e a análise de requisitos visa resolver estes problemas. A análise distribui os requisitos em categorias, explora as relações entre eles, e classifica a importância de cada um dos requisitos de acordo com as necessidades do *stakeholders*. Os requisitos são negociados para decidir quais devem ser aceites, de forma a obter um consenso [61].

A Figura 3.4 a seguir apresentada o os passos do processo de análise e negociação de requisitos.

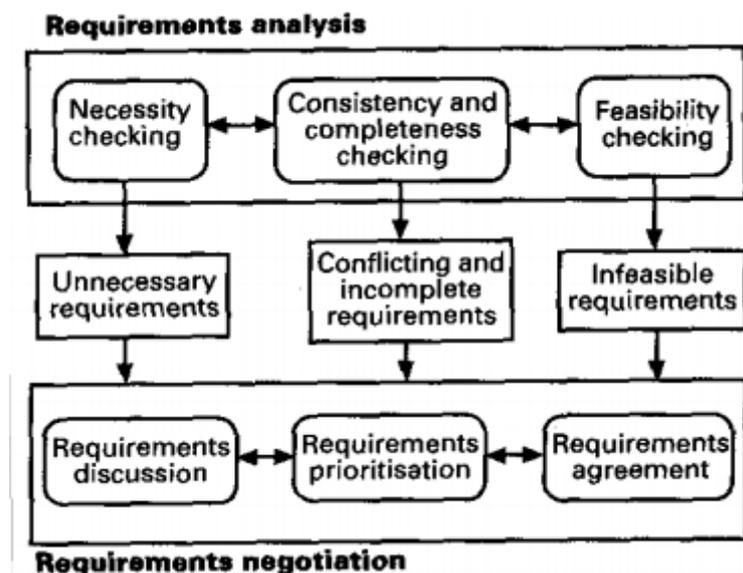


Figura 3.4 - Processo de Análise e negociação [61]

Para especificar a análise de requisitos utilizou-se modelos de casos de uso descrito pelo padrão UML (lógica de negócio). O modelo de casos de uso permite descrever e detalhar com facilidade os requisitos, bem como o aproveitamento destas especificações nos testes ao produto. A construção de um diagrama de casos de uso começa com a identificação dos atores do sistema e posteriormente identificar os principais casos de uso em que cada um interage com o sistema. Os diagramas construídos encontram-se no Documento de especificação de requisitos no Anexo B – Documento de Especificação de Requisitos.

3.2.3. Documentação

Os requisitos acordados são modelados e documentados a um nível de detalhe apropriado. Geralmente, há necessidade de ser produzido um documento de especificação de requisitos, de forma que todos os *stakeholders* possam compreendê-lo. Este documento de especificação deve ser de fácil percepção às pessoas envolvidas no processo de engenharia de requisitos, assim como deve descrever os requisitos funcionais, não funcionais, organizacionais e outros aspetos relevantes do sistema com um grande nível de detalhe. Tais aspetos são características inerentes a cada projeto e podem variar segundo a natureza do *Software* a ser desenvolvido [60].

Geralmente, os requisitos são documentados e comunicados através de um documento formal, elaborado a partir de templates já existentes, como por exemplo, **IEEE Std 830-1998 “Recommended Practice for Software Requirements Specifications”** e **Volere Requirements Specification Template**. Porém, não existe um padrão nem uma designação universal para este documento, podendo cada organização criar o seu próprio modelo ou adaptar um modelo já existente às suas necessidades.

Para a documentar e modelar dos requisitos, optou-se produzir um documento de especificação de requisitos utilizando o **“Volere Requirements Specification Template Edition 17 - 2015”**.

3.2.3.1. Volere Requirements Specification Template

Volere é um modelo de especificação de requisitos que é utilizado como base nas especificações de requisitos. Ele fornece secções para cada um dos tipos de requisitos, adequados aos sistemas de *Software* atuais. Pode-se adaptar este modelo às suas necessidades, como uma ferramenta de levantamento de requisitos. A primeira edição do Volere foi lançado em 1995 e desde então, tem sido utilizado em milhares de projetos em centenas de países. As organizações de todo mundo tem economizado tempo e dinheiro utilizando este modelo como base de especificação dos requisitos. Este é o resultado de muitos anos de prática, consultaria e investigação em Engenharia de Requisitos e análise de negócios e atualmente encontra-se na edição 17.

Este modelo é um modelo que define uma descrição completa das funcionalidades e capacidades dos produtos. O modelo em si é composto por cinco sessões diferentes: **Project Drivers**, **Project Constraints**, **Functional Requirements**, **Nonfunctional Requirements** e **Project Issues** [62].

Também permite escrever os requisitos individualmente, com um conjunto de componentes relacionados com um requisito específico, como mostra a Figura 3.5 a seguir apresentada.

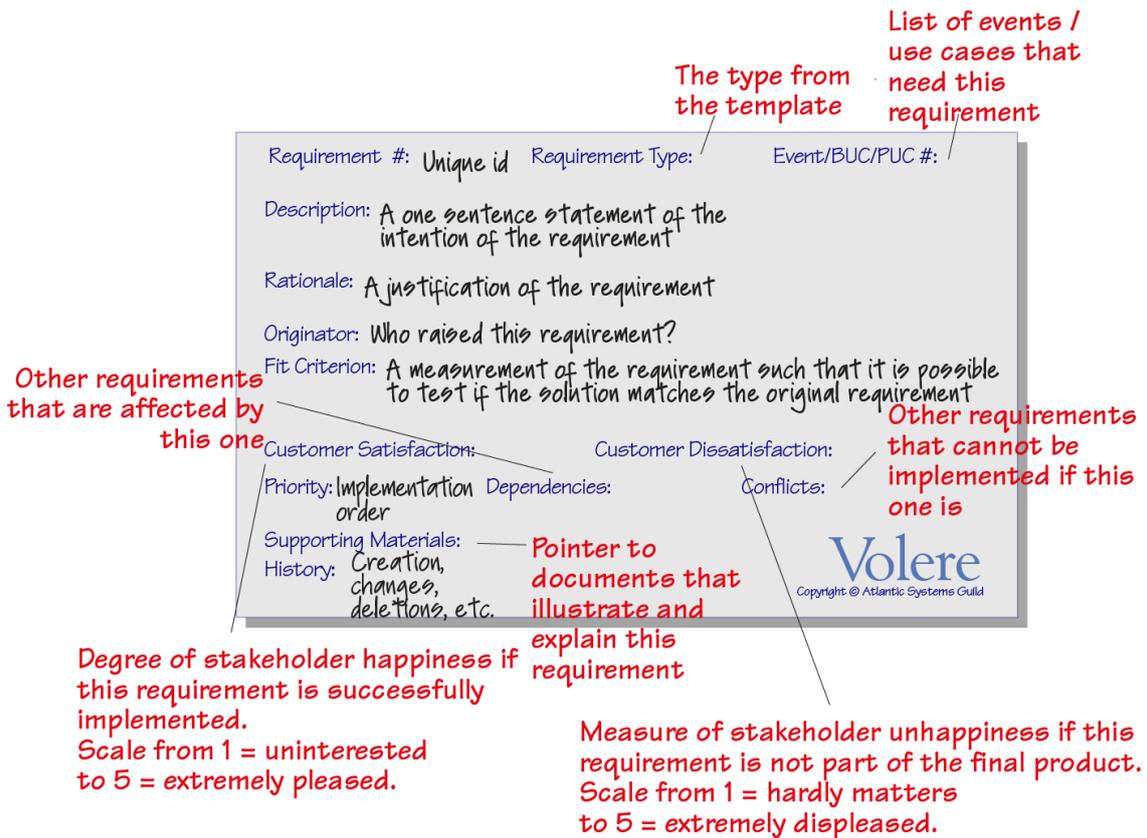


Figura 3.5 - Requirements Shell [62]

Este *template* apresenta um conjunto de recomendações de boas práticas para a especificação do produto e propõe a estrutura especificada no Anexo B – Documento de Especificação de Requisitos.

3.2.4. Validação

A validação de requisitos é definida como o processo que determina se a especificação de requisitos é coerente com a definição de requisitos, ou seja, certifica se o modelo de requisitos é consistente com as necessidades e intenções dos clientes e dos utilizadores. A validação retrata um processo contínuo, ocorrendo na maior parte do tempo em paralelo com as fases de análise, negociação e documentação [60], [61]. Este processo pretende detetar problemas no documento dos requisitos, antes de ser utilizado como base nas fases seguintes do desenvolvimento.

Segundo o modelo do processo de engenharia de requisitos representados na Figura 3.2 acima apresentada, as diferentes atividades devem ser repetidas até que seja tomada a decisão de aceitação do documento de requisitos. Caso seja encontrado um problema no rascunho do documento de requisitos, reentra-se na espiral obtenção/definição, análise e negociação, documentação e avaliação. Isto continua até ser produzido um documento aceitável ou até fatores externos tais como pressões de calendarização ou falta de recursos forçarem o término do processo de engenharia de requisitos. Um documento final de requisitos deve ser produzido. Quaisquer mudanças adicionais aos requisitos farão então parte de um processo de gestão de requisitos [60].

3.2.5. Gestão de requisitos

A gestão de requisitos é o processo de gestão das mudanças dos requisitos de um sistema. Os requisitos de um sistema mudam sempre como reflexo das mudanças das necessidades dos utilizadores, do ambiente onde o sistema vai funcionar, do negócio, das leis, etc. [60].

3.3. Caracterização genérica dos SDUM

Os Serviços de Documentação (SDUM) são uma importante estrutura de apoio às atividades desenvolvidas na Universidade do Minho. Têm por missão recolher, gerir, tratar, organizar, fornecer, preservar e facultar a informação ou recursos informativos relevantes para as atividades educativas e de investigação científica e tecnológica que decorrem na universidade. Os SDUM mantêm e disponibilizam, através da WEB, o catálogo bibliográfico da Universidade do Minho que referencia toda a bibliografia, e outros recursos informativos, existentes nesta Universidade [63].

Entre outras atividades, os SDUM garantem a gestão das bibliotecas da Universidade (funcionamento diário), tanto em Braga como em Guimarães, e desenvolvem ações que visam promover e facilitar o acesso às mais diversas fontes de informação.

De acordo com o Regulamento Orgânico da Universidade do Minho, os Serviços de Documentação constituem uma Direção de Serviços dirigida por um diretor, diretamente dependente do Reitor. A Direção dos Serviços de Documentação constitui um sistema integrado que engloba todas as unidades funcionais de biblioteconomia e informação bibliográfica e todas as bibliotecas da Universidade, como mostra o organograma a seguir apresentado (Tabela 3.1) [63].



Figura 3.6 - Organograma do SDUM [63]

3.3.1. Bibliotecas da Universidade do Minho

Os Serviços de Documentação (SDUM) dispõem de vários pontos de serviço. Conheça o endereço, horários de funcionamento e meios de contacto das diversas bibliotecas.

Aceda ao catálogo bibliográfico de cada uma das bibliotecas e esteja a par das novidades bibliográficas [63].

- **BGUM** - Biblioteca Geral da Universidade do Minho
- **BPG** - Biblioteca da Universidade do Minho em Guimarães
- **BCE** - Biblioteca de Ciências de Educação
- **BEC** - Biblioteca do Edifício dos Congregados
- **BECS** - Biblioteca da Escola de Ciências da Saúde
- **BNP** - Biblioteca Nuno Portas
- **CDEUM** - Centro de Documentação Europeia da Universidade do Minho
- Outras Bibliotecas

3.3.2. Serviço de Empréstimo

Os utilizadores das bibliotecas da Universidade do Minho têm acesso a mais de 414.350 volumes de monografias (livros), a mais de 349.658 fascículos de publicações periódicas (revistas) e cerca de 17.502 outros documentos (audiovisuais, mapas, etc.), distribuídos por cerca de 316.655 títulos referenciados no catálogo bibliográfico.

Este vasto fundo documental pode ser consultado presencialmente, ou utilizado no exterior da biblioteca, por empréstimo [63].

3.3.2.1. Empréstimo Domiciliário

A requisição de publicações é facultada individualmente a cada utilizador inscrito nas bibliotecas da Universidade do Minho. Para requisitar uma publicação, o utilizador deverá recolhê-la na

estante e dirigir-se ao balcão de atendimento, apresentando o cartão de estudante ou documento pessoal que o identifique perante a biblioteca. Em cada biblioteca/piso existe um funcionário disponível para prestar todo o apoio necessário. Todas as publicações são requisitáveis, exceto publicações periódicas, obras de referência, material audiovisual, publicações do fundo de reservas e outras obras devidamente identificadas no catálogo bibliográfico. Em qualquer circunstância, o utilizador é sempre o único responsável pela publicação requisitada, tendo de indemnizar a biblioteca em caso de dano ou perda da mesma.

O empréstimo para leitura domiciliária realiza-se por um período variável, entre 3 e 60 dias, de acordo com o estatuto das publicações e tipologia do utilizador. Segundo a tipologia de utilizadores, podem reter em sua posse o seguinte número de publicações:

- Os docentes, investigadores, alunos de terceiro ciclo e trabalhadores não docentes da Universidade do Minho - até 30 publicações;
- Os alunos de segundo ciclo - até 15 publicações;
- Os alunos do primeiro ciclo - até 10 publicações;
- Os restantes utilizadores - até 6 publicações.

O empréstimo interbibliotecas obedece às mesmas regras do empréstimo domiciliário, com as seguintes exceções:

- O prazo de cedência pode realizar-se até o máximo de 30 dias, a contar da data do envio das publicações para a biblioteca requisitante;
- Os pedidos de empréstimos interbibliotecas têm de ser assinados pelo responsável da biblioteca requisitante;
- No empréstimo interbibliotecas, a biblioteca requisitante funciona sempre como única responsável pelas publicações requisitadas.

Além de existirem empréstimos por um período variável, entre 3 e 60 dias, também existem empréstimos de longa duração, ou seja, empréstimos de publicações por períodos superiores a 60 dias. Os utilizadores que podem beneficiar do empréstimo de longa duração das publicações são as unidades orgânicas de ensino e investigação e respetivas subunidades, os projetos de investigação, as unidades culturais e as unidades de serviços da Universidade do Minho. Estes empréstimos podem também ser aplicados a publicações adquiridas no âmbito do apoio aos projetos de ensino, devendo, neste caso, a requisição ser feita pelo diretor de curso e aplicam-se

somente a publicações compradas especificamente para esse fim pela entidade requisitante, ou publicações cuja taxa de utilização nas bibliotecas da Universidade seja muito baixo, cabendo, neste caso a autorização ao chefe de divisão de biblioteconomia. O empréstimo de longa duração pode ser interrompido, por períodos definidos, se a biblioteca necessitar das publicações para satisfazer pedidos de outros utilizadores e caso a entrega não for procedida implica a cessação do direito a este regime de empréstimo até que a situação esteja regularizada. O empréstimo deve ser renovado no final de cada ano letivo, devendo o responsável da entidade requisitante efetuar o pedido até ao 1º dia útil do mês de Setembro e cada entidade ou utilizador não pode deter mais de 50 publicações simultaneamente (regime especial) [63].

A renovação de requisição de publicações pode ser feita até ao último dia do prazo de empréstimo, pessoalmente ou por telefone nos balcões de atendimento das bibliotecas da Universidade do Minho, através da Internet (no catálogo bibliográfico), utilizando o seu número de utilizador e *password*, ou via correio eletrónico, através do endereço bibliotecas@sdum.uminho.pt. Os pedidos via correio eletrónico, se efetuados no último dia do prazo de empréstimo, apenas serão satisfeitos se enviados durante o horário de funcionamento das bibliotecas e só podem ser considerados efetivos após receção de mensagem de resposta dos SDUM.

O utilizador perde o direito à renovação do prazo de empréstimo se deixar ultrapassar a data de devolução da publicação requisitada, se a biblioteca dela necessitar para satisfazer outros pedidos, ou se atingir o limite de 6 renovações.

A devolução de publicações fora do prazo de requisição está sujeita às seguintes penalizações:

- Atrasos até 15 dias - suspensão do direito de requisitar e renovar publicações, enquanto não é efetuada a devolução da (s) publicação (ões), e pagamento de penalização monetária no valor de €0.50, por cada dia útil e por cada publicação retida;
- Atrasos superiores a 15 dias: penalização monetária atrás mencionada e suspensão do direito de requisição durante 15 dias, contados a partir da data de devolução da publicação.
- Atrasos superiores a 30 dias: penalização monetária atrás mencionada e suspensão do direito de requisição durante 30 dias, contados a partir da devolução da publicação [63].

3.3.2.2. Empréstimo Domiciliário à Distância

O serviço de empréstimo domiciliário à distância destina-se exclusivamente aos utentes inscritos nas bibliotecas da U.M., circunscreve-se ao território nacional e aplica-se a publicações requisitáveis e disponíveis para requisição da Biblioteca Geral da Universidade do Minho (BGUM), da Biblioteca da Universidade do Minho em Guimarães (BPG), da Biblioteca da Escola de Ciências da Saúde (BECS) e da Biblioteca Nuno Portas (BNP), nas condições definidas no Regulamento das Bibliotecas da U.M. relativamente ao empréstimo de publicações. Os pedidos de envio de publicações, em regime de empréstimo domiciliário à distância, devem ser solicitados através do preenchimento de formulário próprio, que deverá ser enviado para o endereço de *email* do Gabinete de Difusão de Informação (dif@sdum.uminho.pt) [63].

3.3.2.3. Reserva de Publicações

A reserva de publicações requisitadas por outros utilizadores pode ser solicitada pessoalmente no balcão de atendimento, por telefone, via *email* ou internet, mediante login no catálogo bibliográfico com o número de utilizador e código pessoal atribuído pela Universidade do Minho. Os utilizadores podem deter até 3 reservas em simultâneo.

Quando a publicação reservada é devolvida na biblioteca, o utilizador é imediatamente notificado via *email*. A publicação mantém-se reservada em seu nome até ao final do dia útil seguinte, para o respetivo levantamento.

O contacto via *email*, de docentes, investigadores, alunos e funcionários da Universidade do Minho, será sempre efetuado para o endereço de correio eletrónico atribuído pela universidade [63].

3.4. Arquitetura da Solução

A arquitetura da solução resume o modelo desenvolvido para o sistema de gestão de empréstimos e reservas das Bibliotecas da Universidade do Minho (solução SDUM). Tal como é ilustrado na Figura 3.7, a arquitetura global contempla dois blocos fundamentais: a aplicação móvel designada por **SDUMApp** e o **servidor**.

- ✓ **Aplicação Móvel** – responsável pelo processamento de dados provenientes do servidor e oferecer as funcionalidades para a interação entre o utilizador e a aplicação.
- ✓ **Servidor** – responsável por disponibilizar os serviços alojados, de modo a permitir o correto funcionamento da aplicação móvel.

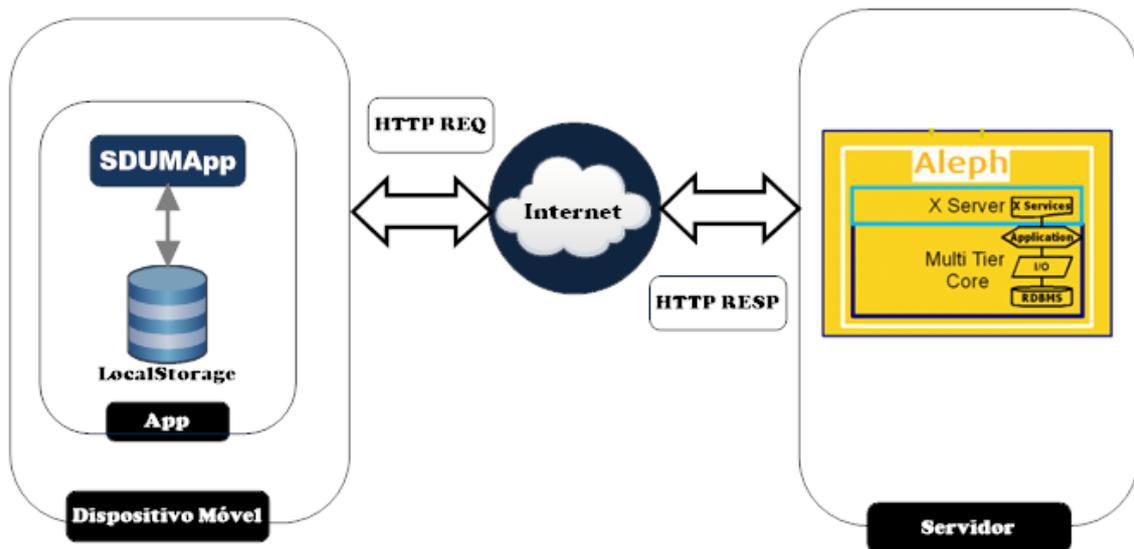


Figura 3.7 - Arquitetura do sistema final

Esta arquitetura segue a paradigma cliente-servidor, em que o cliente é a aplicação **SDUMApp** desenvolvida para o sistema operativo móvel Android e o servidor é o **Aleph X-server**. A protocolo de comunicação entre estes blocos é o protocolo HTTP, o cliente efetua um pedido ao servidor a partir de um sintaxe URL e a resposta é devolvida em XML pelo servidor.

3.4.1. Componentes da Arquitetura

3.4.1.1. Aplicação

A aplicação é o componente fundamental de interação com o utilizador final, ou seja, permite aos utilizadores a interação com o sistema e utilizar as suas funcionalidades. A aplicação foi desenvolvida para dispositivos móveis com sistema operativo Android. O funcionamento do mesmo depende da conexão do dispositivo móvel à internet (Wi-fi ou Redes móveis), visto que é necessário efetuar pedidos ao servidor para que os utilizadores possam visualizar os dados. As respostas dos pedidos efetuados ao servidor são rececionados em formato padrão XML e a aplicação antes apresentar os resultados efetua o tratamento destes dados e são armazenados temporariamente. Quando o utilizador efetuar o *logout* da aplicação, os dados armazenados são eliminados. Ao instalar a aplicação, o utilizador deverá aceitar um conjunto de permissões, de modo que a aplicação tenha acesso a alguns recursos do dispositivo móvel. A tabela representa a lista de permissões de utilização.

As funcionalidades mais pertinentes desta aplicação são:

- Pesquisa de publicações;
- Reserva de publicações;
- Renovação de publicações;
- Visualização de dados pessoais;
- Visualização das publicações em posse;
- Visualização da lista de publicações reservadas;
- Visualização de um calendário de eventos com os livros requisitados;
- Sincronização de publicações requisitados com o calendário do dispositivo móvel.

3.4.1.2. Servidor

O **Aleph X-Services** é o servidor onde são armazenadas todas as informações administrativas, bibliográficas, relativas aos leitores e empréstimos interbibliotecas das bibliotecas da Universidade do Minho. Desde modo, a aplicação interage diretamente com um servidor já existente e acede aos dados alocados no mesmo.

3.4.1.2.1. Aleph WebServices

O **Aleph WebServices** fornece APIs que suportam interações diretas com outras aplicações via HTTP utilizando mensagens em XML que seguem o padrão **SOAP**.

As descrições das operações oferecidas pelo Aleph WebService são escritas em **WSDL (WEB Services Description Language)**.

O X-Server é um componente da estrutura **Multi tier Core** dos produtos Ex Libris e permite a conectividade com o sistema ALEPH a partir da interface do padrão XML [64].

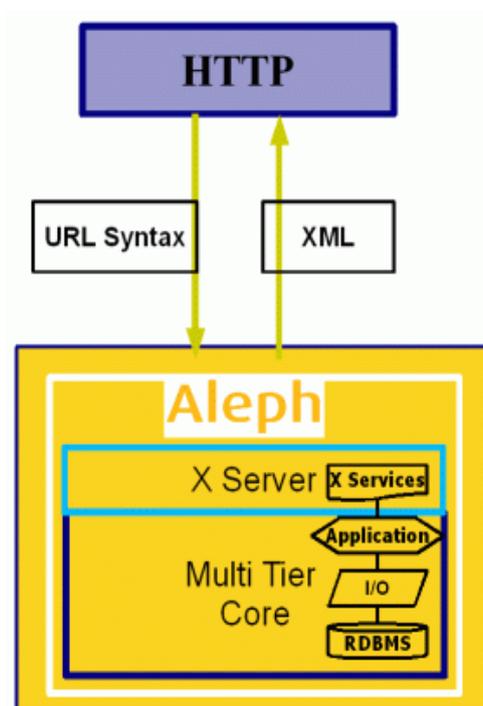


Figura 3.8 - Aleph WEB Service [64]

A Tabela (consta em Anexo D – Lista de Serviços disponibilizados pelo Aleph X-Server) representa a lista de serviços disponibilizados pelo Aleph. Esta lista é atualizada para Aleph 20.

3.4.1.2.1. X-Server

O X-Server é um componente da arquitetura **multi-tier** dos produtos do **Ex Libris** que permite a conectividade com o sistema **Aleph**, bem como os sistemas **MetaLib** e **DigiTool** a partir da interface do padrão XML.

A arquitetura **multi-tier** dos produtos do **Ex Libris** pode ser dividida nas seguintes camadas como mostra a Figura 3.9:

1. A camada **Data Logic and Services** (serviços e lógica dos dados) interage com a base de dados e fornece dados para os níveis mais elevados. Esta camada é composta pelo componente **RDBMS**, que lida com o acesso físico à base de dados, e também pelo **I/O Engine**, que permite a comunicação entre a camada da aplicação e os dados efetivos.
2. A camada **Application Logic** (lógica da aplicação) fornece serviços funcionais à camada Lógica da apresentação, com base nos dados que ela acede a partir da camada **Data Logic and Services**. Esta camada é composta pelo componente **Application Atomic Services** (serviços atômicos da aplicação), que fornece serviços básicos, e pela camada **X-Server**. A camada X-Server fornece serviços compostos baseados na acumulação de vários serviços independentes numa operação funcional abrangente ou no pedido de informação [64].

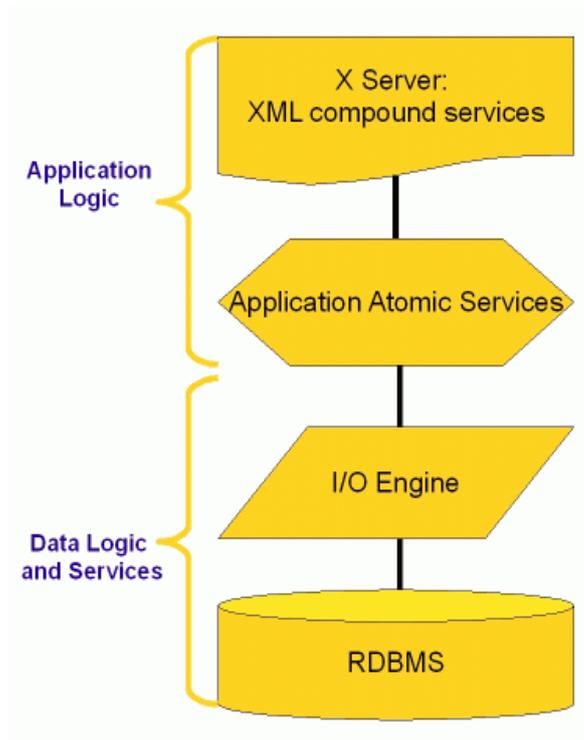


Figura 3.9 - Arquitetura multi-tier dos produtos do Ex Libris [64]

O componente X-Server tem as seguintes vantagens:

- O X-Server permite que os produtos do **Ex Libris** sejam totalmente acessíveis por outros sistemas. A interação de entrada com o X-Server é efetuada a partir da sintaxe URL e de saída a partir do XML, como é ilustrada na Figura 3.9.
- Os serviços fornecidos pelo X-Server são serviços compostos e completos que fornecem uma interação simples e direta com os sistemas **Ex Libris**. Não é necessário o conhecimento do formato interno, dos protocolos e da estrutura dos sistemas.
- O X-Server permite que os componentes da Lógica de apresentação dos produtos Ex Libris sejam independentes da Lógica da Aplicação. A Lógica da apresentação lida com as interações entre o utilizador e a apresentação dos dados [64].

O X-Server recebe a entrada em forma de uma sintaxe URL *standard* e responde em XML, como mostra a Figura 3.10 a seguir apresentada.

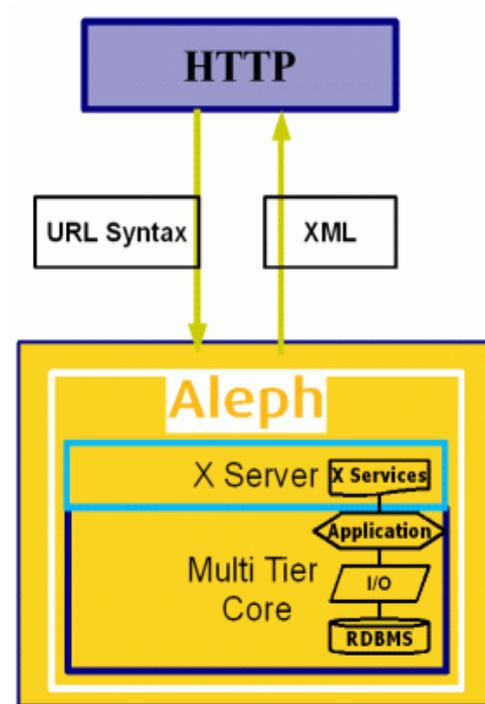


Figura 3.10 - Interação com o X-Server [64]

3.5. Conclusão

Neste capítulo foi aplicado o modelo de processo de engenharia de requisitos, de modo a identificar e especificar as características do produto a desenvolver e analisar os riscos associados. A engenharia de requisitos define, sem dúvida, um dos mais importantes conjuntos de atividades a serem realizadas em projetos de desenvolvimento de *Software*. Embora não garanta a qualidade dos produtos gerados, é um pré-requisito básico para que obtenhamos sucesso no desenvolvimento do projeto.

Deste modo, conclui-se que a engenharia de requisitos depende muito da interação entre os clientes e os engenheiros de requisitos, de modo que seja minimizado qualquer problema na definição de requisitos por parte do cliente. Por mais que não se deseje, os requisitos estão sempre em mudança durante o desenvolvimento de um sistema, e quão melhor for o processo de engenharia de requisitos desenvolvido, menores serão os problemas encontrados em função de toda a dificuldade que envolve esta importante parte da análise.

4. Implementação e testes

4.1. Introdução

Neste capítulo são especificados a implementação da solução SDUM, bem como as tecnologias utilizadas para a implementação das funcionalidades presentes no mesmo. Também são apresentados e discutidos os resultados obtidos resultantes dos testes efetuados, assim como decisões relevantes para o desempenho da aplicação. A implementação das funcionalidades da aplicação foi baseado no Caderno de Especificação de Requisitos (Anexo B – Documento de Especificação de Requisitos), elaborado a partir da execução do modelo de processo de engenharia de requisitos especificados no capítulo anterior.

4.2. Seleção da tecnologia da solução

Para o desenvolvimento da aplicação foi necessário escolher uma ferramenta de desenvolvimento de aplicações multiplataformas mais adequado às necessidades da aplicação final, de acordo com o estudo efetuado anteriormente. Das ferramentas estudadas, foi escolhida o **PhoneGap + Sencha Touch**.

Primeiramente efetuou-se a escolha da abordagem de desenvolvimento multiplataforma e posteriormente a escolha de uma ferramenta de desenvolvimento correspondente à mesma. A seleção de uma abordagem multiplataforma depende principalmente da exigência da aplicação, mas também das plataformas alvos, do tipo de aplicação, do acesso aos dados e recursos (hardware), da interface do utilizador e aparência, do desempenho e da distribuição no mercado, segundo os critérios de seleção de abordagens multiplataformas.

Para este projeto, o principal critério utilizado para a seleção da abordagem foi a exigência e o tipo de aplicação. Cada aplicação móvel tem as suas próprias características, e a seleção da abordagem deve ser determinado em função destas. Existem 4 tipos de aplicações: aplicações de dados conduzidos pelo servidor, aplicações baseados em sensores/IO, aplicações autónimas/independentes e aplicações cliente-servidor. Destes tipos de aplicações, a aplicação a

desenvolver corresponde ao tipo cliente-servidor, visto que tanto o cliente como o servidor encontram-se envolvidos no processamento de dados na aplicação. O cliente para além de interação com o utilizador e de permitir a visualização das informações provenientes do servidor, também efetua o processamento de dados localmente de modo a serem posteriormente exibidos. Estes dados são processados localmente de modo que não são enviados ao servidor, visto que o cliente deverá efetuar a sua gestão e processamento. Como tal para este tipo de aplicações, a abordagem de desenvolvimento híbrida é a mais indicada. (Tabela 2.8). O PhoneGap é considerado um *framework* de referência no desenvolvimento de aplicações híbridas e oferece licença grátis, daí a escolha do mesmo (Figura 2.13).

O PhoneGap pode ser conjugado com alguns **UI-frameworks** (Ex. Sencha Touch, jQuery mobile), de modo a desenvolver uma melhor interface do utilizador. Dos UI-frameworks estudados anteriormente, o escolhido foi o Sencha Touch. Esta escolha foi determinada de acordo com a análise comparativa do desempenho dos *frameworks* de desenvolvimento multiplataforma. Segundo esta mesma análise, as aplicações desenvolvidas pelas ferramentas PhoneGap + Sencha Touch ocupam muito menos CPU do que as aplicações desenvolvidas pelas ferramentas PhoneGap + jQuery mobile, e também consomem menos energia, no entanto as aplicações desenvolvidas pelas ferramentas PhoneGap + jQuery mobile ocupam menos memória. (Tabela 2.15, Tabela 2.16 e Tabela 2.17). O Sencha Touch fornece um conjunto de componentes **GUI (Graphic User Interface)** otimizados para *input touch* que podem ser utilizados nas aplicações WEB móveis. Estes componentes vão desde botões, a elementos de formulário, *sliders*, listas, ícones, separadores móveis, mapa com suporte a *multi-touch*, barras de ferramentas e outros menus, etc. Contém também vários efeitos de transição e suporta os gestos *Touch* mais comuns.



Figura 4.1 - PhoneGap + Sencha Touch [47]

4.3. Implementação da solução SDUM

Nesta secção pretende-se analisar a implementação da aplicação ao nível dos componentes da arquitetura geral descrita no capítulo anterior, nomeadamente a implementação da aplicação móvel e a descrição de URL's de interligação com o servidor (pedido de dados), bem como as decisões de implementação. Pretende-se ainda efetuar testes à solução final de modo a identificar as falhas da solução, de modo a proceder à sua correção.

4.3.1. Aplicação Móvel

Para o desenvolvimento da solução SDUM foi escolhida o framework de desenvolvimento **PhoneGap** e é um “empacotador” que permite incluir aplicações escritos em linguagens de programação conhecidos em aplicações nativas, e o framework **Sencha Touch** é um framework de alto desempenho para desenvolvimento de interfaces do utilizador em dispositivos móveis, que permite desenvolver aplicações multiplataformas utilizando JavaScript. É um framework MVC (*Model-View-Controller*) que consiste numa abordagem que facilita o desenvolvimento e estruturação da aplicação, de modo que existe separação entre a interface e a camada lógica. Desta forma, as alterações feitas na interface da aplicação não afeta a manipulação de dados, e este poderá ser reorganizado sem alterara a interface da aplicação. Desenvolver uma aplicação

recorrendo a este framework de desenvolvimento traz muitas vantagens, de modo que permite o desenvolvimento da aplicação através do padrão MVC. No Sencha Touch este padrão consiste no desenvolvimento em um ou mais *Model*, *View*, *Controller*, *Store* e *Profile*, como pode-se verificar na Figura 4.2 a seguir apresentada.

As *Views* tem duas funções: o primeiro consiste em mostrar as representações de dados dos *Models* e a outra função reside no fato de possibilitarem a obtenção dos dados de entrada introduzida pelo utilizador para os *Controllers*. Os *Controllers* transformam estas entradas de dados em alterações no comportamento da aplicação. O *Model* é a camada que possui a lógica da aplicação. Ele é responsável pelas regras de negócio, lógica e funções. O *Profile* permite uma fácil personificação do UI para tablets e telemóveis. A *View* pode ser qualquer saída de representação de dados. O *Controller* efetua a mediação da entrada, e converte-a em comandos para o modelo ou visão.

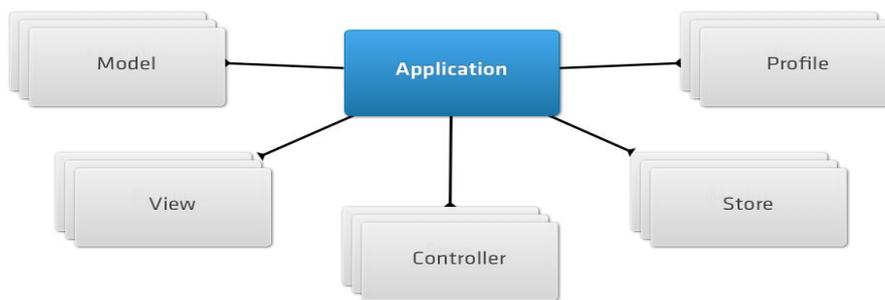


Figura 4.2 - Padrão MVC do Sencha Touch [47]

O desenvolvimento de aplicações a partir de frameworks que utilizam este padrão, é vantajoso de modo que permite a separação dos diferentes aspectos da aplicação (lógica de entrada, lógica de negócio e UI). Para grandes projetos, facilita o seu desenvolvimento e manutenção.

A aplicação é constituída por 55 *Views*, 9 *Controllers*, 8 *Models* e 7 *Stores*, e dada a extensão da estrutura da aplicação, estes não serão abordados individualmente. A Figura 4.3 representa a estrutura da aplicação.

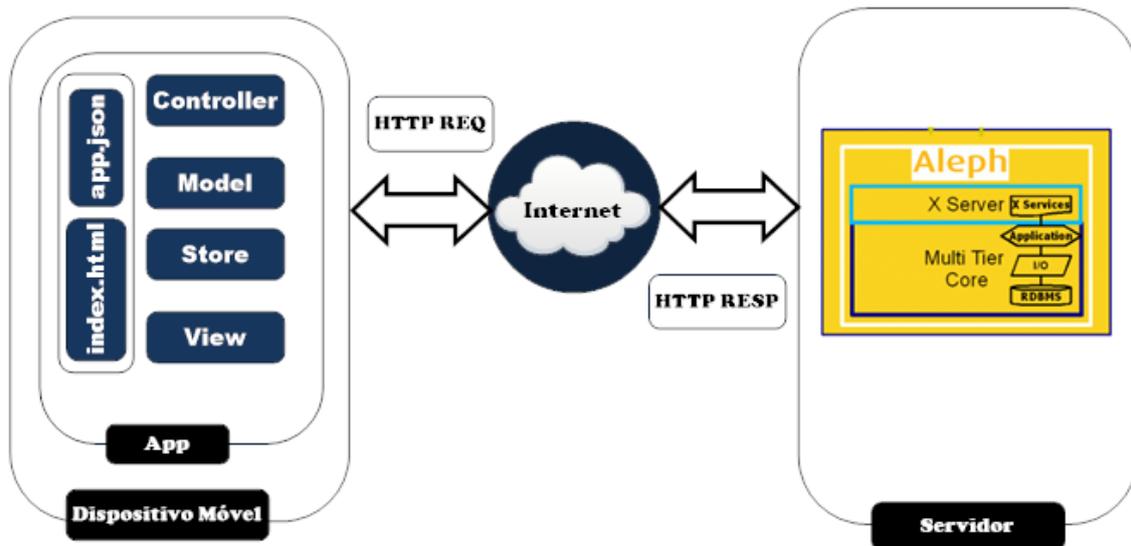


Figura 4.3 - Estrutura dos componentes

4.3.1.1. Configuração de Plugins Cordova

O PhoneGap tem um conjunto de *plugins*, que permitem o acesso aos recursos nativos (tanto a *Hardware* do dispositivo móvel, bem como as aplicações nativas). Estes *plugins* são implementados para cada plataforma de formas diferentes e podem não estar disponível para todas as plataformas móveis. De modo a cumprir com os requisitos propostos, efetuou-se a importação de dois *plugins*, como consta na Tabela 4.1.

Para efetuar a importação de *plugin* para um determinado projeto, deve-se seguir dois passos:

- O primeiro passo, consiste em importar o código nativo para o ficheiro **config.xml**. É necessário introduzir a seguinte tag **<gap:plugin>**;
- No passo seguinte referencia-se o código JavaScript do *plugin*.

Tabela 4.1 – Acesso aos recursos nativos

Plugin	Recurso nativo	Descrição
Network	INTERNET	Permite aceder à internet.
Information	ACCESS_NETWORK_STATE	Permite aceder ao estado da rede Wi-Fi.

	ACCESS_WIFI_STATE	Permite saber o estado da interface Wi-Fi do dispositivo móvel.
Calendar	READ_CALENDAR	Permite ler o calendário do dispositivo móvel
	WRITE_CALENDAR	Permite escrever no calendário do dispositivo móvel.
SQLite	READ_EXTERNAL_STORAGE	Permite o acesso à interface nativa do SQLite (base de dados).
	WRITE_EXTERNAL_STORAGE	
Email Composer	N/A	Permite enviar compor e enviar <i>emails</i> . Acesso a aplicação nativa Email.
SpeechSynthesis	N/A	Permite efetuar a síntese de textos para voz. Alinhado com a W3C Web Speech API.
Vibration	VIBRATE	Permite a vibração do dispositivo móvel. Encontra-se alinhado com a W3C vibration specification.
Whitelist	N/A	Permite a implementação de uma política <i>whitelist</i> para navegar na aplicação webview.

Deste modo, neste projeto, os códigos nativos importados para o ficheiro *config.xml* foram os especificados na Tabela 4.2 a seguir apresentada:

Tabela 4.2 – Código nativo importado para o ficheiro *config.xml*

Plugin	Código nativo importado
Network Information	<gap:plugin name="org.apache.cordova.network-information" />
Calendar	<gap:plugin name="cordova-plugin-calendar" source="npm" />
SQLite	<plugin name="SQLitePlugin" value="org.pgsqlite.SQLitePlugin" />
Email Composer	<gap:plugin name="de.appplant.cordova.plugin.email-composer" version="0.8.2" />

SpeechSynthesis	<gap:plugin name="org.apache.cordova.speech.speechsynthesis" version="0.1.0" />
Vibration	<plugin name="cordova-plugin-vibration" spec="1.2.0" source="pgb" />
Whitelist	<plugin name="cordova-plugin-whitelist" version="1" />

Os códigos nativos referenciados foram os especificados na Tabela 4.3 a seguir apresentada:

Tabela 4.3 – Códigos nativos referenciados

Plugin	Código nativo importado
Network	<script src="Connection.js"></script>
Information	<script src="network.js"></script>
Calendar	<script src="Calendar.js"></script>
SQLite	<script src="SpeechSynthesis.js"></script>
	<script src="SpeechSynthesisEvent.js"></script>
	<script src="SpeechSynthesisUtterance.js"></script>
	<script src="SpeechSynthesisVoice.js"></script>
	<script src="SpeechSynthesisVoiceList.js"></script>
Email Composer	<script src="email_composer.js"></script>
SpeechSynthesis	<script src="SQLitePlugin.js"></script>
Vibration	<script src="vibration.js"></script>
Whitelist	<script src="whitelist.js"></script>

O funcionamento da aplicação (solução SDUM) depende da conexão do dispositivo móvel à internet (Wi-fi ou Redes móveis). Para tal, os utilizadores da mesma serão notificados sempre que os seus dispositivos móveis não tiverem ou perderem a conexão com à internet. Desta forma, para a realização deste requisito efetuou-se a importação do *plugin Network Information*, de modo a permitir a verificação do estado de conexão do dispositivo móvel à internet. A partir da função *document.addEventListener("offline", onOffline, false)*, sempre que o dispositivo móvel perder ou não tiver conexão com à internet, o utilizador é alertado sobre esta falta de conexão.

A sincronização do calendário de eventos da aplicação com o calendário do dispositivo móvel é um dos requisitos para a aplicação imposta pelos SDUM. Desta forma, efetuou-se a importação do *plugin Calendar* para permitir o acesso ao calendário do dispositivo móvel. A partir da função `window.plugins.calendar.createEvent(title,eventLocation,notes,startDate,endDate,success,error)`, é possível criar um novo evento no calendário do dispositivo. A importância desta funcionalidade é inquestionável, visto que possibilita que os utentes das bibliotecas da Universidade do Minho recebam notificações de entrega ou renovação de publicações em posse para evitar futuras penalizações monetárias, devido à entrega de publicações fora da data de entrega.

De modo que alguns dados importantes possam ser guardadas permanentemente, efetuou-se a importação do *plugin SQLite* que permite o acesso à base de dados racional SQL, sem a necessidade de executar um processo RDBMS separado e sem grandes *overheads*. O acesso à base de dados é efetuado a partir da função `window.sqlitePlugin.openDatabase()`. Todas as operações na base de dados são executadas a partir da função `executeSql()` e estas só poderão ser efetuados na função `transaction()` como exemplificado na Figura 4.4 a seguir apresentada.

```
var db = window.openDatabase("Preferencia_DB", "1.0", "Just a Preferences DB", 20000); //will create database Dummy_DB or open it
db.transaction(queryDB,errorCB);
function queryDB(tx){
  tx.executeSql("DELETE FROM Preferencia WHERE codbar="+codigoBarras+"",[], querySuccess, errorCB);
}
function errorCB(err) {
  alert("Error processing SQL: "+err.code);
}
```

Figura 4.4 - Utilização do SQLite Plugin

De modo a possibilitar que os utilizadores enviem questões rápidas relacionadas com os recursos bibliográficos e informativos disponíveis na Universidade do Minho ou questões relacionados com as condições de utilização das bibliotecas U.M foi importado o *plugin Email Composer*. Este permite enviar emails a partir de uma conta pré-configurada na aplicação nativa **Email**. Desta forma, antes do envio do *email* é verificado se este serviço encontra-se disponível no dispositivo móvel. Para tal utiliza-se a função `cordova.plugins.email.isAvailable()` para verificar tal disponibilidade, e em caso afirmativo efetua-se o envio do respetivo *email* a partir da função `cordova.plugins.email.open()`.

Uma das novidades da solução SDUM reside no fato de disponibilizar a ajuda áudio. Consiste em auxiliar os utentes sobre os passos e os dados da aplicação. Esta funcionalidade necessita de ser ativada na aplicação. Para a inclusão desta funcionalidade, foi necessário efetuar a importação do

plugin SpeechSynthesis. Este *plugin* suporta TTS básico, que consiste numa técnica de sintetização da fala humana, que converte texto em linguagem natural. A partir da função *speechSynthesis.speak()* é possível efetuar a conversão de um determinado texto em fala. É possível também a seleção de um idioma para a fala. Na solução SDUM o idioma pré-definido é o Português, mas no entanto poderá ser alterado para o Inglês.

O *plugin Vibrate* foi utilizado para vibrar o dispositivo móvel, sempre que o mesmo não tiver ou perder a conexão com a internet. A vibração do dispositivo móvel é possível a partir da função *navigator.vibrate()*.

Por fim, o *plugin Whitelist* é um modelo de segurança que controla o acesso aos domínios externos. A política de segurança padrão é bloquear todo o acesso à rede. No entanto, a partir deste *plugin*, é possível declarar o acesso a domínios e subdomínios de redes específicos.

A Figura 4.5 a seguir apresentada ilustra as permissões que o utilizador deverá aceitar para a aplicação. Aquando da instalação da aplicação, o utilizador deverá aceitar este conjunto de permissões, de modo a permitirem o acesso aos recursos nativos do sistema operativo móvel, neste caso, o sistema operativo móvel Android.



Figura 4.5 - Permissões

4.3.1.2. Protocolo de Comunicação

Consiste na especificação dos serviços de interface de dados entre o servidor (Aleph X-Server) e a aplicação móvel.

Tabela 4.4 – Login como operador

Métodos	POST
URL	http://aleph.sdum.uminho.pt/X?op=login&user_name={xxxx}&user_password={yyyy}&library=MIN50
Path	login
Parâmetros de Entrada	<i>Username</i> (xxxx) e <i>password</i> (yyyy) do operador
Parâmetros de Saída	Session-id
Descrição	Autenticação como operador
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

Em primeiro lugar é necessário ter uma conta de operador no servidor, de modo a possibilitar a integração da aplicação móvel com o servidor. Esta conta de operador foi facultada pelos SDUM. Efetua-se o pedido de serviço “login” a partir do URL disponibilizada na Tabela 4.4 a seguir apresentada, com a especificação do *username* e *password* de operador previamente facultada. Após a verificação a conta de operador, o servidor devolve a um código de sessão (*session-id*) que será utilizada para as próximas interações com o servidor. Caso o resultado dessa verificação for negativa, é devolvida uma mensagem de erro.

Tabela 4.5 – Autenticação do utente

Métodos	GET
URL	http://aleph.sdum.uminho.pt/X?op=bor-auth&library=min50&bor_id=a9999&verification=X9TR99&session={xxxx}
Path	bor-auth

Parâmetros de Entrada	<i>Session id</i> , numero de aluno e <i>password</i>
Parâmetros de Saída	Z303, Z304, Z305, <i>item-I</i> (Z36, Z30, Z13, <i>Current-fine</i> , <i>Due-date</i>), <i>Balance</i> e <i>Fine</i> .
Descrição	Consulta de dados do utente
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

O pedido de autenticação “bor-auth” de utentes é efetuado a partir do URL disponibilizada na Tabela 4.5. Recebe como entrada os credenciais dos utentes das bibliotecas da Universidade do Minho e o código de sessão disponibilizada anteriormente pelo servidor. A partir daqui, o servidor irá verificar a existência deste utente na base de dados. Após a verificação positiva, o servidor responde com um conjunto de informação:

- **Z303** – corresponde aos dados de processamento do utente (nome, numero de telefone, género, irregularidades de inscrição, data de inserção, etc...);
- **Z304** – dados de correspondência do utente (morada, email);
- **Z305** – dados de inscrição do utente (tipo de utente, data de inscrição, data de expiração da inscrição, bloqueios por irregularidades, permissões, etc...);
- **Z36** – dados de empréstimo de publicações (numero de sistema do documento – id, sub-biblioteca, data e hora de empréstimo, data de devolução, etc...);
- **Z30** – dados do exemplar emprestado (código de barras, catalogadores, etc...);
- **Z13** – dados da obra (autor, titulo, ano, editor, *isbn*, etc...);
- ***Current-fine*** – multa corrente do empréstimo;
- ***Due-date*** – data de devolução do empréstimo;
- ***Balance*** – valor total das multas em dívida;
- ***Fine*** – histórico de multas.

Como o resultado da verificação for negativa, é devolvida uma mensagem de erro.

Tabela 4.6 – Pesquisa de publicações nos catálogos das bibliotecas da Universidade do Minho

Métodos	GET
URL	http://aleph.sdum.uminho.pt/X?op=find&code=campo_pesquisa &request=termo_pesquisa&base=base
Path	find
Parâmetros de Entrada	Campo de pesquisa, termo de pesquisa e catálogo
Parâmetros de Saída	<i>Session-id</i> (id da sessão), <i>set-number</i> (identificação do set de resultados encontrados) e <i>no-records</i> (numero de registos no set de resultados).
Descrição	Pesquisa de publicações
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

O URL do pedido de pesquisa “*find*” de publicações encontra-se especificada na Tabela 4.6 acima apresentada. Recebe como entrada os campos para efetuar uma pesquisa, nomeadamente o campo de pesquisa, o termo de pesquisa e o catálogo de pesquisa. O servidor efetua a pesquisa na base de dados do campo de pesquisa apenas do catalogo de pesquisa e devolve o numero de sessão, a identificação do set de resultados encontrados (*set-number*) e o numero de registos no *set* de resultados (*no-records*). Caso não existir nenhuma publicação com o campo de pesquisa determinado, é retornado uma mensagem de insucesso. Este serviço apenas efetua a pesquisa do número de publicações existentes de acordo com a pesquisa efetuada.

Tabela 4.7 – Apresentação dos dados das publicações pesquisadas

Métodos	GET
URL	http://aleph.sdum.uminho.pt/X?op=present&set_no=set_numero &set_entry=registro_inicial-registro_final&format=marc
Path	present
Parâmetros de Entrada	<i>Set-number</i> e <i>no-records</i>
Parâmetros de Saída	<i>Doc_number</i> , ISBN ou ISSN, título, complemento de título, autor, códigos de barras de exemplares, localizações de exemplares, etc...

Descrição	Pesquisa de publicações
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

Este serviço funciona em conformidade com o serviço especificado anteriormente. Enquanto o anterior efetua a pesquisa das publicações que correspondem à pesquisa efetuada e devolve a *session-id*, o *set-number* e o *no-records*, este serviço devolve a lista de resultados, ou seja, uma lista de publicações com os seus detalhes inerentes (*doc-number*, ISBN ou ISSN, título, complemento ao título, autor, exemplares das publicações, etc...). Recebe como parâmetro de entrada o *set-number*, o *set-number* e o *no-records*. Na

Tabela 4.7 é apresentada a URL do pedido de apresentação dos dados de publicações “*present*”.

Tabela 4.8 – Renovação de publicações

Métodos	POST
URL	http://aleph.sdum.uminho.pt/X?op=renew&item_barcode=código_o_barras&bor_id=id_utente&library=min50&session=session_id
Path	renew
Parâmetros de Entrada	Código de barras do exemplar, id do utente e <i>session-id</i>
Parâmetros de Saída	N/A
Descrição	Renovação de publicações
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

O pedido de renovação de uma publicação é efetuado a partir do URL especificado na Tabela 4.8 acima apresentada. Recebe como parâmetro de entrada, o código de barras da publicação, a identificação do utente e o código de sessão devolvida ao efetuar o pedido de autenticação. O servidor verifica a possibilidade de renovar a publicação, e caso seja possível envia uma mensagem

de confirmação e devolve a data de devolução. Caso contrário, é enviada uma mensagem de erro, especificando assim a indisponibilidade em renovar a publicação.

Tabela 4.9 – Reserva de publicações

Métodos	POST
URL	http://aleph.sdum.uminho.pt/X?op=hold-req&item_barcode=codigo_barras&bor_id=id_utente&library=min50&session=session-id
Path	hold-req
Parâmetros de Entrada	Código de barras do exemplar, id do utente e <i>session-id</i>
Parâmetros de Saída	N/A
Descrição	Reserva de publicações
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

Para reservar uma publicação efetua-se o pedido de serviço “hold-req” como especificado na Tabela 4.9. A URL recebe como parâmetro de entrada o código de barras, a identificação do utente (*username*) e o código de sessão. No entanto, antes de proceder a reserva da publicação, efetua-se a verificação do empréstimo da publicação, ou seja, se encontra ou não na posse de um determinado utente, especificada na Tabela 4.10 a seguir apresentada. Caso a publicação encontrar-se na posse de um determinado utente, é efetuada a renovação e o servidor devolve uma mensagem de confirmação. Caso contrário é enviado uma mensagem de erro, ou seja, da impossibilidade de reservar a publicação.

Tabela 4.10 – Verificação do Empréstimo da publicação

Métodos	GET
URL	http://aleph.sdum.uminho.pt/X?op= item-data &doc-number=id&base=min01&session=session-id
Path	item-data

Parâmetros de Entrada	Código do documento (doc-number) e o código de sessão (<i>session-id</i>)
Parâmetros de Saída	N/A
Descrição	Verificação do Empréstimo das publicações
Código/Dados de retorno	Em caso de sucesso é enviada uma mensagem com os dados de empréstimo da publicação, em caso de insucesso é enviada uma mensagem de erro.

4.3.1.3. Processamento de dados

As respostas dos pedidos efetuados ao servidor são rececionados no formato padrão XML e a aplicação antes apresentar os resultados efetua o tratamento destes dados e posteriormente são armazenados temporariamente. Foi necessário desenvolver um mecanismo de extração dos dados relevantes ao ficheiro XML rececionado. Para tal, utilizou-se o JQuery para efetuar o *parser* do ficheiro. O jQuery oferece uma gama de funções para trabalhar as requisições a partir do AJAX. Quando se trabalha com o AJAX, utiliza-se um objeto nativo dos navegadores que permite a comunicação assíncrona com o servidor. Desta gama de funções disponibilizadas pelo jQuery, utilizou-se a função “**find**” para extração dos dados necessários do XML. Esta função possibilita a obtenção de descendentes de cada elemento selecionado, filtrados por um seletor, objeto jQuery, ou elemento.

4.3.1.4. Definição de Idiomas

Atualmente a exigência no desenvolvimento de aplicações móveis é de tal forma que, não é exigido somente uma aplicação de qualidade, mas também a necessidade de desenvolver aplicações com mais do que um idioma, fruto da grande globalização existente. Por exemplo, a Universidade do Minho recebe mais de 600 alunos em mobilidade (ERASMUS), oriundos de universidades e países diferentes. Grande parte destes alunos são utentes das bibliotecas da Universidade do Minho. A definição de apenas do idioma nativo (Português) na aplicação, acaba por ser insuficiente neste caso. Deste modo, para que não houvesse este problema, implementou-se dois idiomas na

aplicação: o nativo (Português) e a alternativa (Inglês), de modo a abranger o máximo de utentes. Para a implementação, foi necessário identificar todos os componentes de texto nas interfaces da aplicação e associar esses componentes a variáveis que os representam nestas duas línguas diferentes. A alteração do idioma poderá ser efetuado pelo utilizador da aplicação após a inicialização da mesma, e como os textos ficam associados às variáveis que os representam, após a alteração passam a conter o texto do idioma selecionado. A Figura 4.6 a seguir apresentada ilustra estas mesmas variáveis com os respetivos textos.

```

//DEFINIÇÃO DO IDIOMA -> PORTUGUÊS
PANEL: {
  WELCOMEHOME: 'Home',
  WELCOMEAT: 'Atividades',
  WELCOMEDEV: 'Eventos',
  WELCOMEUT: 'Utilizador',
  WELCOMEEMORE: 'Mais',
  WELCOMEOUT: 'Sair',
  WELCOMEABOUT: 'Sobre',
  WELCOMEEMAP: 'Localização Bibliotecas UM',
  WELCOMEEMLOC: 'Localização Bibliotecas',
  WELCOMEEMCALEV: 'Calendário',
  WELCOMEEMPREST: 'Lista de Empréstimos',
  WELCOMEEMPRESTD: 'Detalhes do Empréstimo',
  WELCOMEPEPESQEX: 'Lista de Resultados',
  WELCOMEPEPESQ: 'Pesquisar',
  WELCOMEPEPESQTIPO: 'Campo',
  WELCOMEPEPESQCAT: 'Catálogo',
  WELCOMEPEPESQGER: 'Catálogo Geral',
  WELCOMEPEPESQDOC: 'Documento',
  WELCOMEPEPESQDOCALL: 'Todos',
  WELCOMEEMHISTOR: 'Histórico de Multas',
  WELCOMEEMHISTORD: 'Detalhes da Multa',
  WELCOMEEMAP: 'Localização',
  WELCOMEEMAP1: 'Localização BEC',
  WELCOMEEMAP2: 'Localização BGUM',
  WELCOMEPEPESQLIV: 'Pesquisa Básica',
  WELCOMEPEPESQLIVD: 'Detalhes do Livro',
  WELCOMEPEPESQEXD: 'Detalhes do Exemplar',
  WELCOMEEMRESERV: 'Lista de Reservas',
  WELCOMEEMRESERVD: 'Detalhes da Reserva'
},

//DEFINIÇÃO DO IDIOMA -> INGLÊS
PANEL: {
  WELCOMEHOME: 'Home',
  WELCOMEAT: 'Activities',
  WELCOMEDEV: 'Events',
  WELCOMEUT: 'User',
  WELCOMEEMORE: 'More',
  WELCOMEOUT: 'Log Out',
  WELCOMEABOUT: 'About',
  WELCOMEEMAP: 'UM Libraries Localization',
  WELCOMEEMLOC: 'Library Localization',
  WELCOMEEMCALEV: 'Calendar',
  WELCOMEEMPREST: 'Loan List',
  WELCOMEEMPRESTD: 'Loan Details',
  WELCOMEPEPESQEX: 'Copies Search',
  WELCOMEPEPESQ: 'Search',
  WELCOMEPEPESQTIPO: 'Field',
  WELCOMEPEPESQCAT: 'Catalog',
  WELCOMEPEPESQGER: 'General Catalog',
  WELCOMEPEPESQDOC: 'Document',
  WELCOMEPEPESQDOCALL: 'All',
  WELCOMEEMHISTOR: 'Fines History',
  WELCOMEEMHISTORD: 'Fines Details',
  WELCOMEEMAP: 'Localization',
  WELCOMEEMAP1: 'BEC Localization',
  WELCOMEEMAP2: 'BGUM Localization',
  WELCOMEPEPESQLIV: 'Basic Search',
  WELCOMEPEPESQLIVD: 'Book Detail',
  WELCOMEPEPESQEXD: 'Copy Detail',
  WELCOMEEMRESERV: 'Hold Request List',
  WELCOMEEMRESERVD: 'Hold Request Detail'
},

```

Figura 4.6 - Definição do idioma

4.3.1.5. Funcionalidades da solução SDUM

4.3.1.5.1. Autenticação

Ao iniciar a aplicação, esta é a primeira interface apresentada, como ilustra a Figura 4.7. Efetua-se de igual modo a verificação do estado de conexão do dispositivo móvel à internet e caso for negativo (ou seja, não tiver nenhuma conexão), o utilizador será logo notificado da falta de conexão. Para que um utilizador se autentique na aplicação, deverá preencher os campos obrigatórios do

utilizador e *password* com os respetivos credenciais. Após a submissão deste dados, a aplicação envia um pedido de autenticação do utilizador (*username* e *password*), como especificado na Tabela 4.5 apresentado anteriormente. Caso a autenticação for efetuada, a aplicação direciona o utilizador a interface principal (*Home*). A figura a seguir apresentada representa o ecrã da autenticação.

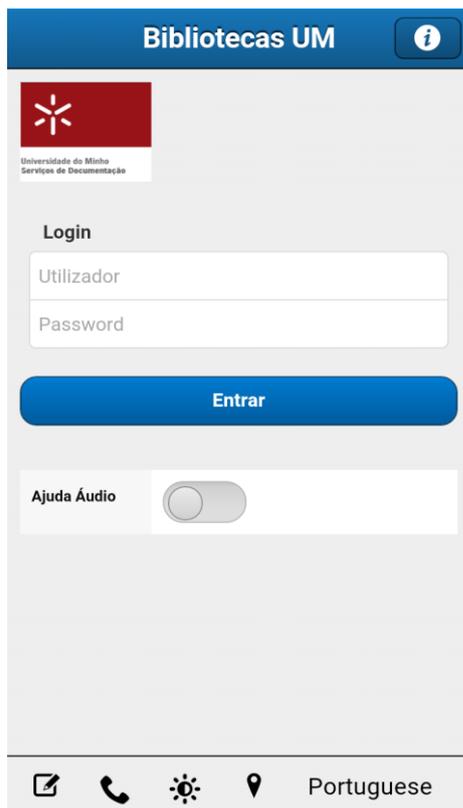


Figura 4.7 - Autenticação no SDUMApp

Nesta interface existem ainda alguns botões de informação e a opção de alteração do idioma. O idioma pré-definido é o Português, mas no entanto poderá ser alterada sempre que o utilizador assim desejar (opção “Portuguese” da Figura 4.7). Os botões de informação informam os utilizadores tanto do processo de autenticação, bem como de informações relativas às bibliotecas da Universidade do Minho (nomeadamente, os horários de funcionamento e a localização destas mesmas bibliotecas).

4.3.1.5.2. Pesquisa de publicações

A pesquisa de publicações é uma das funcionalidades presentes na solução SDUM. Possibilita a pesquisa de publicações mediante o preenchimento dos campos obrigatórios da pesquisa, nomeadamente o que se pretende pesquisa, o campo da pesquisa, o catálogo (o catálogo pré-definido é o catálogo geral, no entanto poderá ser alterada pelos utilizadores) e o documento (predefinição da pesquisa de todos os tipos de documentos, entretanto também poderá ser alterada por uma outra opção). Após efetuar a pesquisa é devolvida uma lista de publicações, que também pode-se verificar os dados de cada um. O pedido de serviço efetuado por este serviço encontra-se especificado na Tabela 4.6 e Tabela 4.7.

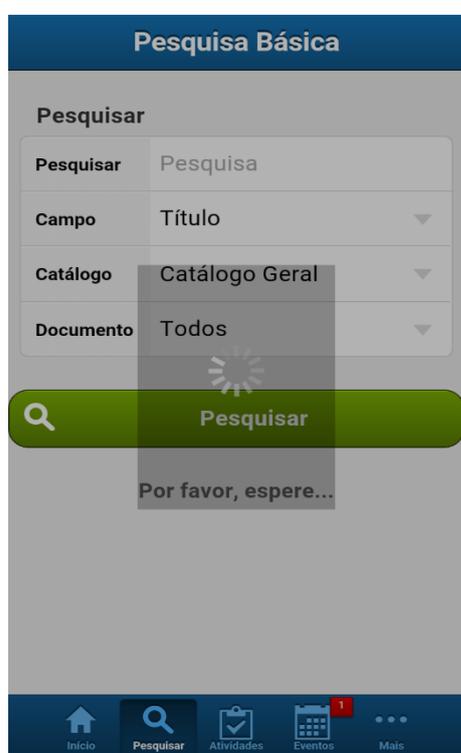


Figura 4.8 - Interface da Pesquisa de publicações



Figura 4.9 - Lista de resultados da pesquisa de publicações

4.3.1.5.3. Reserva de publicações

A interface permite aos utentes efetuarem a reserva de uma determinada publicação. Para tal, é necessário efetuar a pesquisa desta publicação e selecioná-la, de modo a visualizar os seus dados

detalhados. Como especificado anteriormente a reserva de publicações só será efetuada, caso a publicação encontra-se em empréstimo. Caso contrário notifica-se o utilizador da não possibilidade de reservar a publicação em questão. Este pedido de serviço encontra-se especificado na Tabela 4.9 acima apresentado.



Figura 4.10 - Reserva de publicações

Em caso afirmativo da reserva, é possível verificar na Lista de Reservas, no qual inclui esta publicação reservada, mas no entanto é necessário atualizar os dados associados ao utilizador. Deste modo existe um botão de atualização, que possibilita a atualização destes dados.

4.3.1.5.4. Lista de Empréstimos e Renovação de publicações

A lista de empréstimos para além de permitir os utilizadores visualizarem as publicações em sua posse, também permite que estes efetuem pedidos de renovação destas mesmas publicações. O pedido de renovação de publicações encontra-se especificados na Tabela 4.8 acima apresentada.

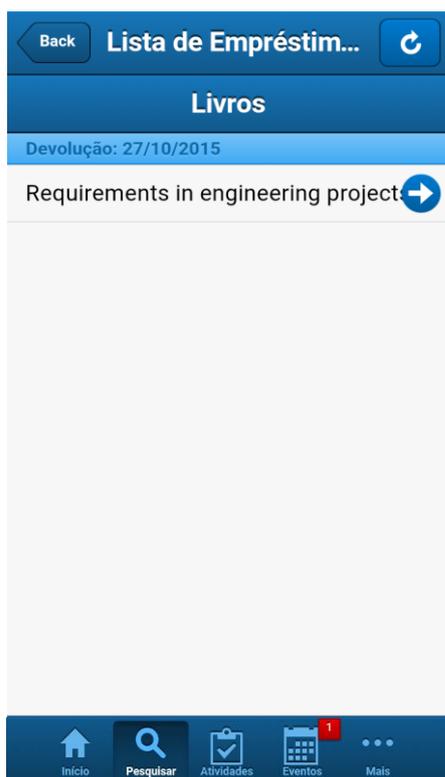


Figura 4.12 - Lista de Empréstimos



Figura 4.11 - Renovação de publicações na Lista de Empréstimos

4.3.1.5.5. Calendário de Eventos e Sincronização

Uma das funcionalidades de grande importância e relevância é sem dúvida o de Calendário de eventos e sincronização deste com o calendário do dispositivo móvel. Esta funcionalidade, para além de possibilitar os utilizadores a visualização das publicações com as respetivas datas de entrega, também permite a criação de eventos no calendário do dispositivo móvel, viabilizando dessa forma a receção de notificações para entrega ou renovação de publicações.

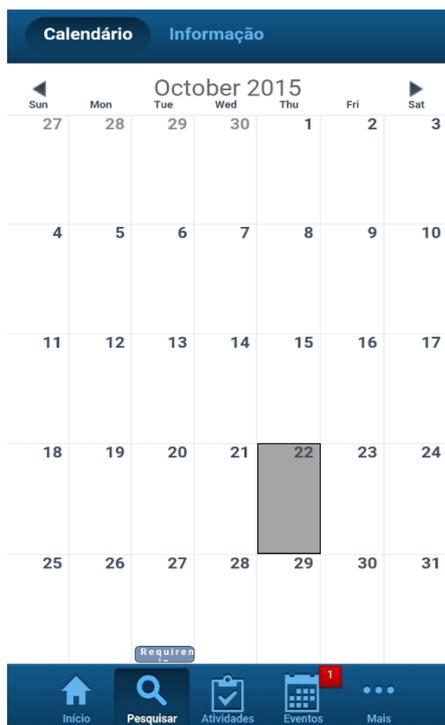


Figura 4.14 - Calendário de Eventos



Figura 4.13 - Calendário do dispositivo móvel

4.3.1.5.6. Lista de Preferências

Esta interface permite que os utentes visualizem as publicações de preferência. É mais uma valia de modo que esta lista permite que estas publicações possam ser visualizadas após a sua pesquisa. Após a pesquisa de publicações é devolvida uma lista. De seguida é selecionado uma determinada publicação da lista de modo a visualizar os seus dados. Existe um botão cujo nome é “Preferências” e caso seja habilitada, este incluirá esta determinada publicação na lista de preferências, como ilustrada na Figura 4.15 a seguir apresentada.

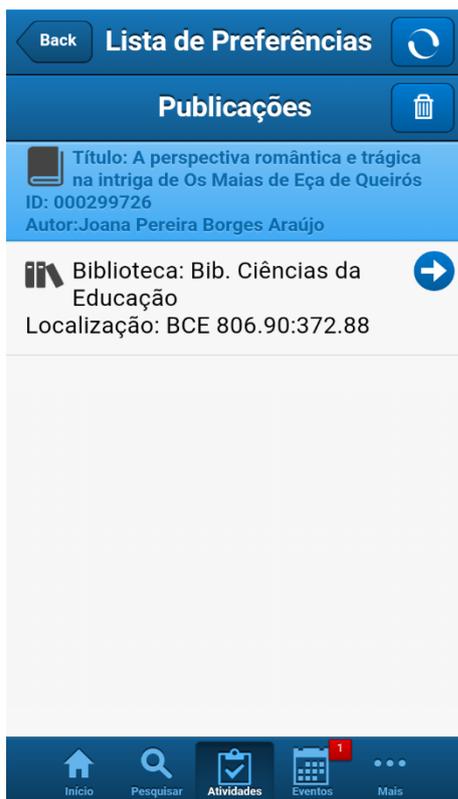


Figura 4.16 - Lista de Preferências



Figura 4.15 – Pesquisa de publicações

4.4. Testes

O teste de *Software* é um processo de execução de um produto que visa determinar se este cumpre as suas especificações e funciona corretamente no ambiente para o qual foi projetado. O objetivo desta fase é revelar todas as falhas do sistema, para que as mesmas possam ser identificadas e corrigidas antes da entrega final, ou seja da passagem a produção. Serão especificados todos os testes realizados e a sua importância no processo de garantia de qualidade.

4.4.1. Testes da funcionalidade

Este teste tem como objetivo testar o que a aplicação deve fazer, de acordo com os requisitos funcionais do sistema.

A Tabela 4.11 a seguir apresentada especifica o teste de funcionalidade efetuada na solução SDUM, de acordo com os requisitos funcionais delineados e especificados no Anexo B – Documento de Especificação de Requisitos. A avaliação é efetuada de acordo com a seguinte nomenclatura:

F - Funciona; FR - Funciona com algumas Restrições; NF - Não funciona; NT - Não foi testada.

Tabela 4.11 – Teste da funcionalidade

Requisitos	Resultado
1. Visualização do horário de funcionamento das bibliotecas da Universidade do Minho	F
2. Visualização da localização das bibliotecas	F
3. Pesquisa de publicações	F
4. Reserva de publicações	F
5. Renovação de publicações	F
6. Visualização da Lista de Empréstimos	F
7. Visualização da Lista de Reservas	F
8. Visualização dos dados pessoais	F
9. Visualização do calendário de eventos	F
10. Visualização das penalizações monetárias por entrega de publicações fora do prazo	F
11. Sincronização das datas de devolução de publicações com o calendário do dispositivo móvel	F

Todos os requisitos delineados foram cumpridos e testados com sucesso. O objetivo deste teste passa por percorrer todos os serviços e funcionalidades da aplicação, na procura de anomalias ou pequenos erros, e identificá-los no documento para correção futura.

4.4.2. Testes de desempenho

O teste de desempenho tem como objetivo verificar se a aplicação satisfaz os requisitos de desempenho específicos, por exemplo, a capacidade e o tempo de resposta dos pedidos efetuados pelos utilizadores.

A Tabela 4.12 especifica um conjunto de testes relativamente ao tempo de resposta, ou seja, para cada um dos pedidos efetuados pelos utilizadores, foram medidos os tempos de realização dos mesmos.

Para este teste foram utilizados dois dispositivos móveis: o tablet **Samsung Galaxy Tab 4 7'** (Modelo: SM-T230NU) com 1.5GB de memória e 8GB de espaço de armazenamento, o *smartphone* **Samsung Galaxy S3** (Modelo: i9300) com 1GB de memória RAM e 16GB de espaço de armazenamento, o *smartphone* **OnePlus X** (Modelo: E1003) com 3GB de memória RAM e 16GB de espaço de armazenamento e o *smartphone* **Alcatel One Touch Pix 3 (4)** (Modelo: 4013X) com 512MB de memória RAM e 4GB de espaço de armazenamento ligados a rede *eduroam*.

Tabela 4.12 – Teste de desempenho

Pedidos	Tempo de resposta				Observação
	Galaxy Tab 4	Galaxy S3	OnePlus X	One Touch	
Execução da aplicação	12s	12s	6s	16s	Os tempos de resposta já incluem o tempo de acesso à rede
Autenticação	10s	4s	3s	13s	
Pesquisa de publicações (Lista de resultados < 20 exemplares)	10s	6s	4s	18s	
Pesquisa de publicações (Lista de resultados > 100 exemplares)	25s	16s	14s	52s	
Reserva de publicações	5s	2s	2s	13s	
Renovação de publicações	5s	2s	2s	13s	
Sincronização das datas de entrega de publicações com o calendário do dispositivo móvel	3s	2s	2s	11s	
Atualização dos dados	9s	2s	2s	6s	

4.4.3. Testes de Interoperabilidade

A interoperabilidade é a habilidade de 2 ou mais sistemas ou componentes de trocar informação e usar a informação que foi trocada. Também pode ser definida como a capacidade de comunicar e executar programas através de várias unidades funcionais utilizando-se linguagens e protocolos comuns. Este teste tem como objetivo testar e avaliar a comunicação entre a aplicação com o servidor Aleph X-Server. A Tabela 4.13 a seguir apresentada corresponde aos resultados obtidos do teste de interoperabilidade aplicados à solução SDUM. A troca de dados com o servidor é efetuada a partir do protocolo HTTP. O pedido de um serviço é efetuado a partir de uma sintaxe URL e a resposta é devolvida em XML pelo servidor. A avaliação é efetuada de acordo com a seguinte nomenclatura:

F - Funciona; FR - Funciona com algumas Restrições; NF - Não funciona; NT - Não foi testada.

Tabela 4.13 – Teste de interoperabilidade

Pedido	Descrição	Protocolo	Resultado
login	Autenticação como operador	HTTP	F
bor-auth	Autenticação	HTTP	F
find	Pesquisa de publicações	HTTP	F
renew	Renovação de publicações	HTTP	F
hold-req	Reserva de publicações	HTTP	F

4.4.4. Testes de Compatibilidade

Os testes de compatibilidade tem como objetivo garantir a compatibilidade das aplicações entre diferentes ambientes e plataformas. O objetivo destes testes é garantir a interação e a comunicação da aplicação com as diferentes dispositivos e versões de sistemas operativos.

A solução SDUM foi desenvolvida para dispositivos móveis com sistema operativo móvel Android. Deste modo, pretende-se efetuar o teste de compatibilidade da aplicação em diferentes dispositivos móveis com OS Android e versões diferentes do OS Android.

Os dispositivos móveis utilizados para o teste foram:

- **Samsung Galaxy S3** (Modelo: i9300) – Android 5.1
- **Samsung Galaxy S4** (Modelo: i9500) – Android 5.1
- **Samsung Tab 4 7'** (Modelo: SM-T230NU) – Android 4.4.2 (Kit Kat)
- **OnePlus X** (Modelo: E1003) – Android 5.1.1
- **Alcatel One Touch** (Modelo: 4013X) – Android 4.2.2

Em todos estes dispositivos móveis nos quais foram testados a aplicação, não foram detetados erros. A aplicação funciona normalmente, no entanto nota-se alguma lentidão fruto também do fato das aplicações híbridas terem pior desempenho em relação às aplicações nativas.

4.4.5. Teste de Usabilidade

O teste de usabilidade tem como foco a experiência do utilizador, a facilidade e possibilidade de realizar as ações pretendidas, ou seja, tem como objetivo verificar a facilidade que a aplicação possui, de ser claramente compreendida e manipulada pelo utilizador. Este tipo de teste é uma técnica de pesquisa utilizada para avaliar um determinado produto. Os testes são realizados com utilizadores representativos do público-alvo. Cada participante irá tentar realizar as tarefas propostas e avalia-los, bem como efetuar a avaliação da usabilidade geral da aplicação, respondendo um questionário que avalia a sua concordância com a aplicação. Para a avaliação da usabilidade geral foi utilizada o método *System Usability Scale (SUS)* que propõe medir a usabilidade de forma confiável. Este método foi desenvolvido por **John Brooke** em 1986 e consiste num questionário composto por 10 afirmações com cinco opções de resposta, que os participantes avaliam o seu nível de concordância com a aplicação numa escala de 0-4.

Metade das questões está redigida de forma positiva e a outra metade está redigida de forma negativa, para obrigar o utilizador a estar atento quando estiver a responder ao questionário. Deste modo, a avaliação das tarefas é feita de acordo com o nível de dificuldade:

1 – Muito fácil; 2 – Fácil; 3 – Neutro; 4 – Difícil; 5 – Muito difícil.

A avaliação da usabilidade geral é efetuada de acordo com o nível de concordância:

- Para as afirmações ímpares a avaliação é efetuada da seguinte forma: 0 – Discordo totalmente; 1 – Discordo; 2 – Neutro; 3 – Concordo; 4 – Concordo totalmente.
- Para as afirmações pares a avaliação é efetuada de forma contrária: 0 – Concordo totalmente; 1 – Concordo; 2 – Neutro; 3 – Discordo; 4 – Discordo totalmente.

A soma das contagens deve ser multiplicada por 2.5 para obter-se o valor global da usabilidade da aplicação. Deste modo, as pontuações do SUS tem uma gama de 0 – 100. Pontuações abaixo do limiar de 60 pontos representam sistemas com experiências relativamente pobres e insatisfeito para os utilizadores, e pontuações acima dos 80 pontos representam experiências muito boas, com alto índice de satisfação dos utilizadores.

Foi solicitado a um conjunto de 20 voluntários que nunca tiveram contacto prévio com a aplicação, à realização das principais tarefas da solução SDUM, de modo a avaliarem o quão fácil a aplicação pode ser compreendida, aprendida, utilizada e atrativa para o utilizador. O questionário de avaliação da usabilidade e da avaliação das tarefas pode ser consultada com maior nível de detalhe no Anexo C – Testes de Usabilidade.

Os resultados da avaliação da usabilidade das tarefas obtidos foram os seguintes:

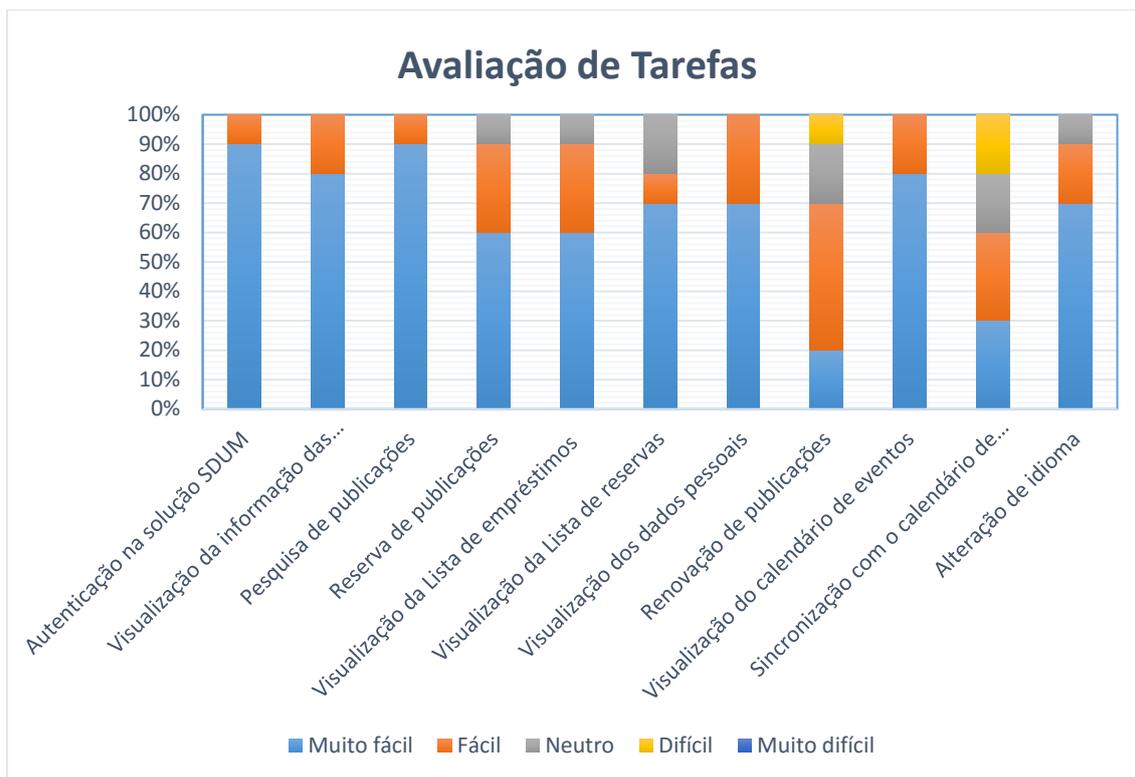


Gráfico 1 - Avaliação das tarefas presentes na solução SDUM

Pode-se verificar no Gráfico 1 acima apresentado, que quase todas as funcionalidades da solução SDUM são fáceis ou muito fáceis de serem realizadas. A tarefa 10 apresenta maior dificuldade aos utilizadores, visto que é uma funcionalidade nova introduzida na solução SDUM e para a sua realização é necessário a leitura das informações associadas a esta. Na tarefa 1, 90% dos utilizadores consideraram que é muito fácil de ser realizado, enquanto os restantes 10% avaliaram a tarefa como sendo fácil. Nas tarefas 2, 70% dos utilizadores consideram que é uma tarefa muito fácil de ser realizada e apenas 30% consideram ser uma tarefa fácil. Na tarefa 3, 90% dos inquiridos consideram ser uma tarefa muito fácil e apenas 10% consideraram ser uma tarefa fácil. Nas tarefas 4 e 5, 60% dos utilizadores consideraram serem tarefas muito fáceis, e os restantes 40% foram classificados como fáceis e neutros (30% consideram ser uma tarefa fácil, enquanto que os restantes 10% consideraram que a tarefa não é nem fácil nem difícil). A tarefa 6 é considerada muito fácil pela maioria dos utilizadores (70%), e das restantes 30%, 10% consideraram ser uma tarefa fácil e 20% consideraram o meio-termo (nem fácil, nem difícil). Já a tarefa 7, 70% consideraram ser uma tarefa muito fácil e os restantes 30% classificaram como uma tarefa fácil. A tarefa 8 teve algum nível de dificuldade por parte dos utilizadores, de modo que apenas 20% consideraram ser uma tarefa muito fácil, 50% consideraram ser uma tarefa fácil, 20% neutros e

10% consideraram ser uma tarefa difícil de ser efetuada (esta dificuldade resultado do fato de não saberem os passos necessários para a renovação de publicações). A tarefa 9 foi considerada muito fácil por 80% dos utilizadores, enquanto os restantes 20% consideraram ser fácil. A tarefa 10 com maior nível de dificuldade encontrada pelos utilizadores, de modo que 20% consideraram como sendo difícil, outros 20% consideraram não ser uma tarefa difícil ou fácil, e os restantes 60% (30% dos utilizadores consideraram ser uma tarefa fácil e os outros 30% consideraram ser muito fácil). Para terminar, a tarefa 11 foi considerada pela maioria dos utilizadores como sendo uma tarefa muito fácil, com 70% das classificações dos utilizados e os restantes 30% (20% consideraram ser uma tarefa fácil e os restantes 10% consideraram ser neutros, não sendo fácil e nem difícil).

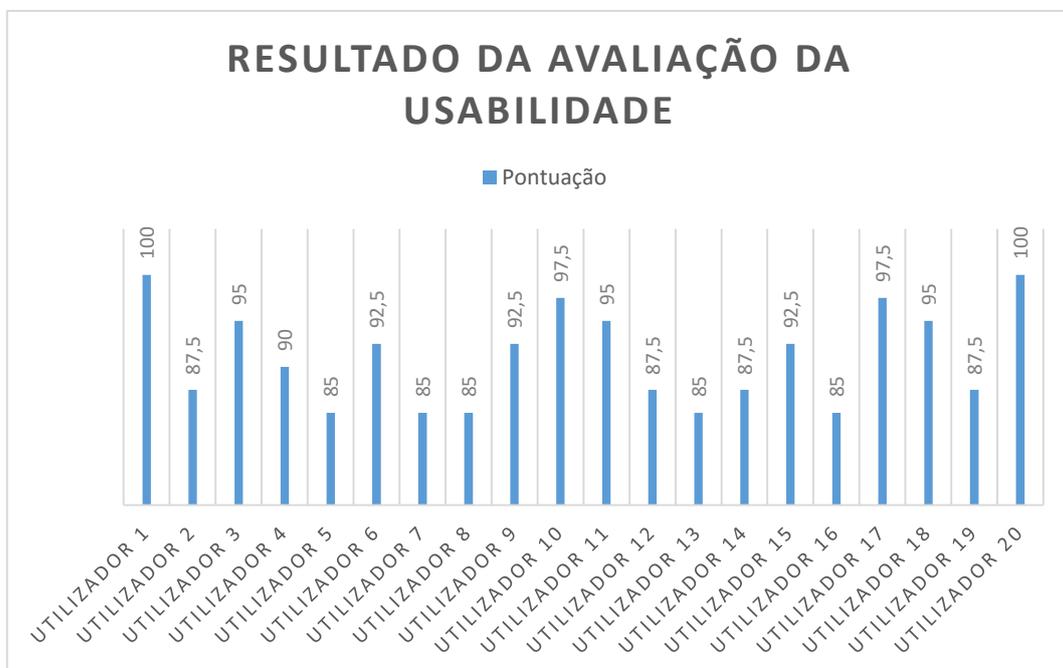


Gráfico 2 - Avaliação da usabilidade geral da solução SDUM

Média das pontuações das avaliações SUS: 91,125 pontos

Como especificado anteriormente, as pontuações abaixo do limiar de 60 pontos representam sistemas com experiências relativamente pobres e com alto nível de insatisfação dos utilizadores, e pontuações acima dos 80 pontos representam experiências muito boas, com alto índice de satisfação dos utilizadores. Como tal, de acordo com a média das pontuações das avaliações SUS realizadas pelos 20 utilizadores (91,125 pontos), pode-se verificar que a aplicação (solução SDUM)

vai ao encontro das inspirações destes mesmos utilizadores, ou seja, os utilizadores ficaram satisfeitos com a experiência do utilizador da aplicação, bem como a facilidade e possibilidade de realizar as ações pretendidas. O questionário de avaliação da usabilidade geral da aplicação pode ser consultada com maior nível de detalhe no Anexo C – Testes de Usabilidade.

4.5. Conclusão

Neste capítulo foram especificados a tecnologia selecionada para o desenvolvimento da solução final, as decisões de implementação da solução SDUM, bem como os testes aplicados à solução final. A seleção da tecnologia foi efetuada de acordo com estudo efetuado no anteriormente, e posteriormente efetuou-se a seleção de uma ferramenta de desenvolvimento que seja esta mesma abordagem de desenvolvimento. Deste modo, foi selecionado as ferramentas Sencha Touch + PhoneGap, o Sencha Touch para o desenvolvimento da interface de utilizador e o PhoneGap para gerar a aplicação híbrida e o acesso aos recursos do dispositivo móvel. Na implementação da solução SDUM foram especificados a decisões de implementação do componente Aplicação Móvel da arquitetura da solução. Foi especificado o protocolo de comunicação entre os dois componentes da arquitetura (Aplicação + Servidor), como foi efetuado o acesso aos recursos nativos dos dispositivos móveis, como é efetuado o processamento dos dados oriundos do servidor, a definição dos idiomas e as funcionalidades com maior relevância. Por fim, efetuou-se um conjunto de testes, de modo a identificar possíveis erros, de modo que fossem corrigidas, bem como testes ao nível do utilizador. Os resultados do teste de funcionalidade foram um sucesso, de modo que todas funcionalidades propostas foram cumpridas e funcionam na sua plenitude, de igual forma, os testes de interoperabilidade efetuados. Os testes de desempenho aplicados foram os esperados em aplicação híbridas, bem como os testes de compatibilidade efetuados em diferentes dispositivos com sistema operativo móvel android. De acordo com os 20 utilizadores que avaliaram a solução SDUM, acharam a aplicação fácil de ser compreendida e manipulada. De acordo com a avaliação da usabilidade geral da aplicação, os utilizadores classificaram a aplicação com 91,125 pontos em 100 possíveis.

Deste modo, concluiu-se que todos os desafios de implementação foram abordadas e ultrapassados com sucesso de acordo com as soluções adequadas encontradas para os resolver,

bem como todos os requisitos propostos pelos Serviços de Documentação da Universidade do Minho.

5. Conclusão

5.1. Síntese do Trabalho

Finalizada a implementação da aplicação, é tempo de fazer o balanço e retrospectiva das ideias gerais que transpareceram ao longo do processo de análise e conceção da aplicação. As tecnologias de informação são áreas que estão em constante em evolução, praticamente todos os dias surge uma novidade tecnológica. Uma das áreas que tem beneficiado com esta evolução é do desenvolvimento de aplicações móveis.

Esta dissertação teve como objetivo principal desenvolver uma aplicação móvel de apoio a gestão empréstimos e reservas de publicações nas bibliotecas da Universidade do Minho, de acordo com a abordagem teórica e aplicada às abordagens de desenvolvimento de aplicações móveis, mais em concreto da abordagem de desenvolvimento de aplicações móveis multiplataforma. Foi realizado um pequeno estudo sobre os dispositivos e plataformas móveis. Logo a seguir foi efetuado um estudo comparativo sobre as abordagens de desenvolvimento de aplicações móveis, com maior ênfase sobre as abordagens de desenvolvimento multiplataforma. Também foi efetuado um estudo sobre os critérios de seleção destas abordagens de desenvolvimento multiplataforma, bem como sobre os frameworks de desenvolvimento. Com esta revisão literária, permitiu concluir que a seleção de uma abordagem de desenvolvimento multiplataforma depende principalmente da exigência da aplicação, mas também das plataformas alvos, do tipo de aplicação do acesso aos dados e recursos (hardware), da interface do utilizador e aparência, do desempenho e da distribuição no mercado, ou seja, a seleção da abordagem de desenvolvimento multiplataforma depende do tipo aplicação e das suas necessidades, de modo que cada caso é um caso, e a seleção de uma abordagem de desenvolvimento multiplataforma poderá ser a mais adequada a uma aplicação e não ser a mais adequada para outras aplicações. Para esta aplicação foi selecionada a abordagem de desenvolvimento multiplataforma híbrida, visto que em relação ao tipo de aplicações corresponde ao cliente-servidor, de modo que tanto o cliente como o servidor encontram-se envolvidos no processamento de dados na aplicação. Para tal, para este tipo de aplicações a melhor abordagem a ser selecionada é a abordagem híbrida. Relativamente à estratégia de desenvolvimento de aplicações híbridas, embora ainda extremamente recente e com

algumas limitações, o desenvolvimento destas aplicações tem um enorme potencial. No entanto, há ainda muito por fazer para que se torne uma estratégia sólida e confiável por grande parte da comunidade informática. A melhoria progressiva da capacidade dos dispositivos móveis, e a evolução do HTML5, são sem dúvida dois fatores que podem tornar esta estratégia mais popular junto da comunidade.

Depois seleção da abordagem de desenvolvimento multiplataforma foi fundamental elaborar e executar o modelo de processo de engenharia de requisitos, que consiste em 4 fases: levantamento de requisitos, análise e negociação, documentação e validação. Efetuou-se o levantamento e negociação dos requisitos da aplicação com a chefe da divisão de informação dos SDUM e posteriormente produziu-se um documento de especificação de requisitos, como consta no Anexo B – Documento de Especificação de Requisitos. A produção do Documento de Especificação de Requisitos é muito importante porque estabelece uma base de acordo entre o cliente e os desenvolvedores do *Software* sobre as funções a serem desempenhadas pelo sistema e as necessidades a serem atendidas, uma vez que nele deverá constar uma descrição completa de todos os requisitos identificados e validados por ambas as partes. A elaboração deste documento reduz também os riscos associados ao desenvolvimento do *Software*, pois obriga a uma negociação dos requisitos antes de iniciar a construção do sistema, evitando assim inconsistências ou erros. Depois foi efetuada a arquitetura da solução final, que especifica os componentes do sistema.

Ultrapassada esta fase, procedeu-se à seleção de um framework de desenvolvimento de aplicações híbridas e posteriormente a implementação do sistema. O framework selecionado foi o Sencha Touch + PhoneGap, de acordo com alguns critérios considerados pertinentes ao bom funcionamento da aplicação final, nomeadamente à baixa ocupação da CPU e consumo de energia, quando executada.

Após a conclusão da implementação do sistema, a aplicação foi testada com o objetivo de identificar erros e bugs, de modo que efetuar-se os ajustes finais.

Uma vez concluído este trabalho, verifica-se que os desafios e requisitos iniciais foram abordadas e que foram implementados com sucesso.

5.2. Trabalho Futuro

Esta secção é dedicada ao trabalho futuro que pode ser desenvolvido na sequência desta dissertação. Os objetivos definidos inicialmente foram atingidos com sucesso, mas ainda assim há margem de melhoramento em alguns aspetos que podem ser explorados e aprofundados, com o objetivo de cativar todos os utentes das bibliotecas da Universidade do Minho.

As sugestões de melhoramento a seguir apresentadas são diretivas, que podem ser utilizadas para expandir o trabalho:

- A interface gráfica da aplicação poderia ser melhorada, tornando-a mais atrativa aos utentes;
- Gerar esta mesma aplicação para outras plataformas móveis com maior cota no mercado (iOs, Windows Phone e BlackBerry OS), de modo a possibilitar que todos os utentes utilizem esta aplicação. Esta aplicação foi desenvolvida para dispositivos com sistema operativo móvel Android numa primeira fase. Posteriormente consoante a popularidade da mesma e a satisfação dos utentes, será também distribuída esta aplicação para as outras plataformas móveis.
- Introdução de novos idiomas de forma a abranger o maior número de potenciais utilizadores. Atualmente a aplicação suporta dois idiomas: o nativo – português e a alternativa – inglês. Presume-se que estes dois idiomas satisfazem todos os utentes, mas a introdução de novos idiomas é sempre uma vantagem;
- Desenvolver a aplicação com um *framework* de desenvolvimento de aplicações híbridas, que permite melhores índices de desempenho. Ex. Xamarin. O único senão reside no fato deste framework de desenvolvimento permitir o desenvolvimento de aplicações para apenas três dos sistemas operativos móveis com maior cota de mercado (Android, iOs e Windows Phone). Atualmente a Xamarin anunciou que a sua plataforma passaria a ser gratuita a estudantes, no entanto os interessados deverão preencher um formulário e comprovar o seu estatuto de estudante;
- Maior nível de segurança entre a aplicação e o servidor. As aplicações híbridas são aplicações desenvolvidas utilizando tecnologias WEB (HTML5, CSS3, JavaScript) e são executadas nos dispositivos móveis como se fossem aplicações nativas, ou seja, é um médio termo entre a abordagem nativa e a abordagem WEB. O HTML5 traz melhorias significativas para a WEB, e com isso trás novas preocupações de segurança. Garantir

uma comunicação segura entre a aplicação e o servidor é dos pontos mais importantes. Os dados que circulam entre ambos não são críticos, mas merecem um tratamento seguro e confidencial;

- Funcionalidade de leitor de código de barras de modo a possibilitar verificar os detalhes de uma determinada publicação. Para esta funcionalidade seria auxiliada pela camera do dispositivo móvel de modo a efetuar o *scanner* e reconhecer a informação destes códigos;
- Disponibilização de capas e índices de publicações após uma pesquisa;
- Integração da aplicação com o *repositorium* da Universidade do Minho, de modo a que os utentes tenham também acesso aos diversos tipos de documentos em formato digital (artigos de revistas científicas, as comunicações a congressos e conferências, as teses de doutoramento e as dissertações mestrado) a partir da aplicação;
- Incorporação de RSS *feeds* das bibliotecas da Universidade do Minho;
- Disponibilização do mapa da localização de publicações (mapa da sala com a localização aproximada da aplicação), após efetuação de uma pesquisa nos catálogos das bibliotecas da Universidade do Minho;
- A disponibilização dos inquéritos aos utilizadores das bibliotecas da Universidade do Minho a partir da aplicação;
- Finalmente seria interessante analisar detalhadamente a usabilidade da aplicação com a realização de um conjunto de questões (inquéritos) aos utentes das bibliotecas da Universidade do Minho, a fim de obter uma opinião dos mesmos sobre a aplicação. Foi feita um teste de usabilidade com um número reduzido de amostras, de modo que seria interessante efetuar um teste de usabilidade para um número maior de utentes.

Referências

- [1] M. Palmieri, I. Singh, A. Cicchetti, Comparison of cross-platform mobile development tools, 2012 16th Int. Conf. Intell. Next Gener. Networks. (2012) 179–186. doi:10.1109/ICIN.2012.6376023.
- [2] A. Charland, B. Leroux, mobile application Development : Web vs . native, Commun. ACM. 54 (2011) 0–5. doi:10.1145/1941487.
- [3] R.L. Baskerville, Investigating information systems with action research, Commun. AIS. 2 (1999) 4. doi:http://www.cis.gsu.edu/~rbaskerv/CAIS_2_19/CAIS_2_19.html.
- [4] R. O'Brien, An overview of the methodological approach of action Research, Univ. Toronto. (1998) 1–15. http://www.web.ca/~robrien/papers/arfinal.html (accessed January 28, 2015).
- [5] M. Greeff, L. Coetzee, Using action research for complex research initiatives, Adapt. Sci. Technol. 2009. ICAST 2009. 2nd Int. Conf. (2009) 331–338. doi:10.1109/ICASTECH.2009.5409703.
- [6] V. Santos, L. Amaral, H. Mamede, Utilização do método Investigação-Ação na investigação em Criatividade no Planeamento de Sistemas de Informação Using the Action-Research Method in Information Systems Planning Creativity research, (n.d.). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6615743&tag=1.
- [7] I. Hernández, A. Viveros, E. Rubio, Analysis for the design of open applications on mobile devices, Ieeexplore.Ieee.Org. (2013) 126–131. doi:10.1109/CONIELECOMP.2013.6525772.
- [8] IDC – International Data Corporation, Smartphone OS Market Share, Q2 2015, (2015). http://www.idc.com/prodserv/smartphone-os-market-share.jsp (accessed March 20, 2015).
- [9] M. Nosrati, Mobile Computing : Principles , Devices and Operating Systems, (2012) 399–408. doi:10.1002/wcm.1203/abstract.
- [10] M.A. Goud, V. Bhaskar, R.A.O. Prof, V.P. Rao, Mobile Devices Overview, (2013) 103–107. http://www.interscience.in/IJCSI_Vol3Iss2/103-107.pdf.
- [11] T.-M. Gronli, J. Hansen, G. Ghinea, M. Younas, Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS, 2014 IEEE 28th Int. Conf. Adv. Inf. Netw. Appl. (2014) 635–641. doi:10.1109/AINA.2014.78.
- [12] H.Y. Cho, C.S. Nam, D.R. Shin, A compariosn of open and closed mobile platforms, ICEIE 2010 - 2010 Int. Conf. Electron. Inf. Eng. Proc. 2 (2010) 141–143. doi:10.1109/ICEIE.2010.5559730.
- [13] A. Hammershoj, A. Sapuppo, R. Tadayoni, Challenges for mobile application development, 2010 14th Int. Conf. Intell. Next Gener. Networks (ICIN 2010) Second Int. Work. Bus. Model. Mob. Platforms (BMMP 2010). (2010) 1–8. doi:10.1109/ICIN.2010.5640893.

- [14] P. Kaur, S. Sharma, Google Android a mobile platform: A review, *Eng. Comput. Sci. (RAECS)*, 2014 Recent Adv. 10 (2014) 1–5.
doi:10.1109/RAECS.2014.6799598.
- [15] E. Oliver, A survey of platforms for mobile networks research, *ACM SIGMOBILE Mob. Comput. Commun. Rev.* 12 (2009) 56.
doi:10.1145/1508285.1508292.
- [16] J. Liu, J. Yu, Research on Development of Android Applications, 2011 4th Int. Conf. *Intell. Networks Intell. Syst.* (2011) 69–72. doi:10.1109/ICINIS.2011.40.
- [17] Google Inc., Android Technology Overview, (n.d.).
<http://source.android.com/source/index.html> (accessed February 6, 2015).
- [18] G.S. Chandel, P.K. Singh, *International Journal of Advanced Research in Computer Science and Software Engineering*, Int. J. 4 (2014) 62 – 70.
http://www.ijarcsse.com/docs/papers/July2012/Volume_2_issue_7/V2I700161.pdf
http://www.ijarcsse.com/docs/papers/9_September2012/Volume_2_issue_9/V2I90140.pdf.
- [19] C. Wang, The research of Android System architecture and application programming, *Proc. 2011 Int. Conf. Comput. Sci. Netw. Technol.* (2011) 785–790. doi:10.1109/ICCSNT.2011.6182081.
- [20] A. Inc., iOS Technology Overview, (2014).
https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898-CH1-SW1 (accessed February 6, 2015).
- [21] Research In Motion Limited, BlackBerry Technology Overview, (2013).
<https://help.blackberry.com/en/blackberry-security-overview/latest/blackberry-security-overview-html/awi1402929620791.html> (accessed February 6, 2015).
- [22] P. Scandurra, M. Rosario, Native versus Cross-platform frameworks for mobile application development, (n.d.).
- [23] H. Heitkötter, S. Hanschke, T. a Majchrzak, Comparing Cross-Platform Development Approaches for Mobile Applications, *Web Inf. Syst. Technol.* 140 (2013) 120–138. doi:10.1007/978-3-642-36608-6.
- [24] IBM Software, Native, WEB or hybrid mobile-app development, n.d.
<ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>.
- [25] M. Ciman, O. Gaggi, Evaluating impact of cross-platform frameworks in energy consumption of mobile applications, *WEBIST14 - 10th Int. Conf. Web Inf. Syst. Technol.* (2014). <http://www.math.unipd.it/~gaggi/doc/WEBist14battery.pdf>.
- [26] L. Delia, N. Galdamez, P. Thomas, L. Corbalan, P. Pesado, Multi-Platform Mobile Application Development Analysis, *IEEE Softw.* (2015) 0–5.
[http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7128878&filter=AND\(p_IS_Number:7128853\)](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7128878&filter=AND(p_IS_Number:7128853)).
- [27] C.P. Rahul Raj, S.B. Tolety, A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach, 2012 Annu. *IEEE India Conf. INDICON 2012.* (2012) 625–629.
doi:10.1109/INDCON.2012.6420693.

- [28] S. Xanthopoulos, S. Xinogalos, A comparative analysis of cross-platform development approaches for mobile applications, Proc. 6th Balk. Conf. Informatics - BCI '13. (2013) 213. doi:10.1145/2490257.2490292.
- [29] B.S. Thakare, State of Art Approaches to Build Cross Platform Mobile Application, Int. J. Comput. Sci. Eng. 107 (2014) 22–23. <http://research.ijcaonline.org/volume107/number20/pxc3900389.pdf>.
- [30] T.A.M. Henning Heitkötter, Sebastian Hanschke, Evaluation of cross-platform development for mobile devices, (2014).
- [31] S. Charkaoui, Z. Adraoui, E.H. Benlahmar, Cross-platform mobile development approaches, 2014 Third IEEE Int. Colloq. Inf. Sci. Technol. (2014) 188–191. doi:10.1109/CIST.2014.7016616.
- [32] D. Sin, E. Lawson, K. Kannoopatti, Mobile web apps - The non-programmer's alternative to native applications, Int. Conf. Hum. Syst. Interact. HSI. (2012) 8–15. doi:10.1109/HSI.2012.11.
- [33] N. Serrano, J. Hernantes, G. Gallardo, Mobile Web Apps, IEEE Softw. 30 (2013) 22–27. doi:10.1109/MS.2013.111.
- [34] W.S. El-Kassas, B.A. Abdullah, A.H. Yousef, A.M. Wahba, Taxonomy of Cross-Platform Mobile Applications Development Approaches, Ain Shams Eng. J. (2015). doi:10.1016/j.asej.2015.08.004.
- [35] A. Sommer, S. Krusche, Evaluation of cross-platform frameworks for mobile applications, Proc. 1st Eur. Work. Mob. Eng. (2013) 363–376. <http://subs.emis.de/LNI/Proceedings/Proceedings215/363.pdf>.
- [36] H. Heitkötter, S. Hanschke, T. a Majchrzak, Comparing Cross-Platform Development Approaches for Mobile Applications, Web Inf. Syst. Technol. 140 (2013) 120–138. doi:10.1007/978-3-642-36608-6.
- [37] P. Smutný, Mobile development tools and cross-platform solutions, Proc. 2012 13th Int. Carpathian Control Conf. ICC 2012. (2012) 653–656. doi:10.1109/CarpathianCC.2012.6228727.
- [38] AllianceTek Empowering IT Solution, Comparison Chart for Mobile App Development Methods, (2012) 1–4. <http://www.alliancetek.com/downloads/article/comparison-chart-for-mobile-app.pdf> (accessed April 5, 2015).
- [39] C.T. Tool, C. Tool, L. Enterprise, Cross-Platform Tool Benchmarking Find the right tool for your app project July 2014, (2014). <http://www.research2guidance.com/r2g/Cross-Platform-Tool-Benchmarking-Report-2014.pdf>.
- [40] A. Inc., Appcelerator Titanium, (2015). <http://www.appcelerator.com/product/> (accessed April 5, 2015).
- [41] J. Ohrt, V. Turau, Cross-platform development tools for smartphone applications, Computer (Long. Beach. Calif). 45 (2012) 72–79. doi:10.1109/MC.2012.121.
- [42] I. Dalmaso, S.K. Datta, C. Bonnet, N. Nikaein, Survey, comparison and evaluation of cross platform mobile application development tools, 2013 9th Int. Wirel. Commun. Mob. Comput. Conf. (2013) 323–328.

doi:10.1109/IWCMC.2013.6583580.

- [43] A. Ribeiro, A.R. da Silva, Survey on Cross-Platforms and Languages for Mobile Apps, 2012 Eighth Int. Conf. Qual. Inf. Commun. Technol. (2012) 255–260. doi:10.1109/QUATIC.2012.56.
- [44] Z.T. Inc., Rhodes/RhoMobile, (2015). <http://docs.rhomobile.com/en/2.2.0/rhoelements/rhoelements-introduction> (accessed April 5, 2015).
- [45] Adobe Systems Inc., PhoneGap, (2015). <http://phonegap.com/> (accessed April 5, 2015).
- [46] H. Heitkötter, T. Majchrzak, B. Ruland, T. Weber, Evaluating Frameworks for Creating Mobile Web Apps, Wi.Uni-Muenster.De. (2013) 209–221. doi:10.5220/0004356702090221.
- [47] Sencha Inc., Sencha Touch, (2015). <https://www.sencha.com/products/touch/#overview> (accessed April 5, 2015).
- [48] M. Fowler, Patterns of Enterprise Application Architecture, 2003. doi:10.1119/1.1969597.
- [49] Q. Mahmoud, Z. Maamar, Applying the MVC Design Pattern to Multi-Agent Systems, 2006 Can. Conf. Electr. Comput. Eng. (2006) 2420–2423. doi:10.1109/CCECE.2006.277427.
- [50] D.M. Selfa, M. Carrillo, M. Del Rocío Boone, A database and web application based on MVC architecture, Proc. 16th IEEE Int. Conf. Electron. Commun. Comput. CONIELECOMP 2006. 2006 (2006) 48. doi:10.1109/CONIELECOMP.2006.6.
- [51] F.A. Masoud, D.H. Halabi, ASP.NET and JSP Frameworks in Model View Controller Implementation, 2006 2nd Int. Conf. Inf. Commun. Technol. 2 (2006) 3593–3598. doi:10.1109/ICTTA.2006.1684998.
- [52] Microsoft, Model-View-Controller, (2015). <https://msdn.microsoft.com/en-us/library/ff649643.aspx> (accessed December 21, 2015).
- [53] Design Patterns: Model View Controller (MVC) Pattern - 2015, (2015). http://www.bogotobogo.com/DesignPatterns/mvc_model_view_controller_pattern.php (accessed December 21, 2015).
- [54] K. Gottschalk, S. Graham, H. Kreger, J. Snell, Introduction to Web services architecture, IBM Syst. J. 41 (2002) 170–177. doi:10.1147/sj.412.0170.
- [55] M.A. Araújo, Web services na informação geográfica, (2005) 123. <http://repositorium.sdum.uminho.pt/handle/1822/4570> (accessed December 22, 2015).
- [56] O. Wuermli, A. Wrobel, Web Service Overview, (2003) 1–28.
- [57] C.J.F. Lopes, J.C. Ramalho, Web Services : Metodologias de Desenvolvimento, (2004). <http://repositorium.sdum.uminho.pt/handle/1822/559> (accessed December 22, 2015).
- [58] P.B.L. de Abreu, Coordenação de RESTful Web Services, (2010) 67. http://mei.di.uminho.pt/sites/default/files/dissertacoes/eeum_di_dissertacao_pg10

972.pdf.

- [59] A.G. Miguel, O Risco e a Gestão do Risco em Projectos de Desenvolvimento de Sistemas de Informação, Universidade do Minho, 2002.
<http://repositorium.sdum.uminho.pt/handle/1822/26594>.
- [60] A. Rocha, Influência da maturidade da função sistema de informação na abordagem à engenharia de requisitos, Universidade do Minho, 2000.
<http://hdl.handle.net/1822/182>.
- [61] Ian Sommerville, Gerald Kotonya, Requirements Engineering: Process and Techniques, New York, NY, USA, 1998.
<http://dl.acm.org/citation.cfm?id=552009>.
- [62] S.R. Robertson, James, Volere Requirements Specification Template, (2015).
<http://www.volere.co.uk/template.htm> (accessed June 21, 2015).
- [63] SDUM, Serviços de Documentação - Universidade do Minho, (n.d.).
<http://www.sdum.uminho.pt/> (accessed November 14, 2015).
- [64] Ex Libris Ltd, Introduction to Aleph X-Services, (2015).
<https://developers.exlibrisgroup.com/aleph/apis/Aleph-X-Services/introduction-to-aleph-x-services> (accessed February 17, 2015).

Anexos

Anexo A – Estrutura do Volere Requirements Specification Template

Tabela A.1 - Estrutura do Volere Requirements Specification Template [62]

Estrutura	Descrição
The Purpose of the Project	O propósito do projeto está relacionado com os objetivos do projeto e lida com a razão fundamental que o cliente pediu para construir o produto, ou seja, descreve o problema de negócio que o cliente enfrenta e explica como o produto destina-se a resolver o problema.
The Stakeholders	Esta seção descreve os <i>stakeholders</i> (as pessoas que tem interesse no produto).
Constraints	As restrições do projeto representam as restrições ao desenvolvimento da aplicação, ou seja, são as restrições gerais que limitam as opções durante o desenvolvimento dos requisitos.
Naming Conventions and Terminology	Geralmente os projetos tem o seu próprio vocabulário, geralmente contendo uma variedade de siglas e abreviaturas. Por vezes a falha em entender corretamente esta nomenclatura específica do projeto leva inevitavelmente a mal entendidos e a falta de comunicação entre os membros da equipa. Esta seção consiste na elaboração de uma lista de siglas e abreviaturas utilizados no projeto.
Relevant Facts and Assumptions	Os fatos relevantes são fatores externos que tem um efeito sobre o produto, mas não são limites de requisitos obrigatórios. Eles não são necessariamente traduzidos em requisitos, mas poderiam ser. Os fatos relevantes alerta os desenvolvedores/programadores às condições e fatores que tenham incidência nos requisitos.
The Scope of the Work	Tem como objetivo determinar o trabalho que precisa ser realizar para que se possa entregar o produto com as características e funções específicas, ou seja, consiste em determinar o que irá ser feito ate ao término do projeto.
Business Data Model	Esta sessão tem como objetivo a especificação do assunto essencial, dos objetos de negócios, das entidades e classes que são pertinentes ao produto.

and Data Dictionary	
The Scope of the Product	Consiste em apresentar as características e funções que caracterizam o produto.
Functional Requirements	Representa a especificação de cada requisito funcional.
Non-functional Requirements	São as propriedades comportamentais que as funções especificadas devem ter, tais como, o desempenho, a usabilidade, etc.
Project Issues	Representam e definem as condições em que o projeto será realizado. Apresenta uma imagem coerente de todos os fatores que contribuem para o sucesso ou o fracasso do projeto.

Tabela A.2 - Estrutura detalhado do Volere Requirements Specification Template

Estrutura		Descrição
The Purpose of the Project	The User Business or Background of the Project Effort	Consiste numa breve descrição do negócio que está a ser efetuado, o seu contexto, bem como a situação que desencadeou o esforço de desenvolvimento.
	Goals of the Project	Esta seção consiste em descrever as especificações relativamente ao produto (o que se pretende que o produto faça) e que vantagens que trará para os objetivos gerais do trabalho.
The Stakeholders	The Client	Determina o nome do cliente (por vezes referido como o patrocinador).
	The Customer	Refere-se a pessoa que destina-se a comprar o produto. No caso de desenvolvimento interno, o <i>client</i> e o <i>customer</i> são provavelmente a mesma pessoa.

	Other Stakeholders	Consiste em determinar as funções e possível, os nomes de pessoas e organizações que são afetadas pelo produto.
	The Hands-On Users of the Product	<p>Uma lista de tipos especiais de <i>stakeholders</i> – potenciais utilizadores do produto. Para cada categoria de utilizados, é necessário fornecer as seguintes informações:</p> <ul style="list-style-type: none"> ✓ Nome do utilizador/categoria; ✓ Função do utilizador; ✓ Experiencia Tecnológica; ✓ Experiencia do assunto; ✓ Outras características do utilizador
	Personas	Uma história sobre uma pessoa inventada que inclua: o nome da pessoa, a idade, o trabalho, família, <i>hobbies</i> , morada, música favorita, etc...
	Priorities Assigned to Users	<p>Priorizar (nível de importância) cada tipo de utilizadores. Isto identifica a importância e a excelência dos utilizadores. A priorização dos utilizadores deve ser da seguinte forma:</p> <ul style="list-style-type: none"> ✓ Utilizadores chave – São essenciais para o sucesso contínuo do produto. É necessário dar maior prioridade aos requisitos gerados por este tipo de utilizadores. ✓ Utilizadores secundários – irão utilizar o produto, mas a sua opinião sobre o mesmo não tem nenhum efeito sobre o sucesso a longo prazo do produto. ✓ Utilizadores sem importância – para esta categoria de utilizadores é dada a prioridade mais baixa. Neste inclui utilizadores esporádicos, não qualificados e não autorizados
	User Participation	Anexar a categoria de utilizadores uma indicação de participação, ou seja, descreve a contribuição que é esperada desses utilizador para fornecer os requisitos (por exemplo: o conhecimento do negócio, prototipagem da interface ou requerimentos de usabilidade).

	Maintenance Users and Service Technicians	Utilizadores de manutenção são um tipo especial de utilizadores <i>hands-on</i> que tem requisitos específicos para a manutenção e alteração do produto.
Constraints	Solution Constraints	Especifica as restrições sobre a forma como o problema deve ser resolvido. Descreve a tecnologia ou a solução obrigatório. Você deve explicar a razão para utilizar a tecnologia.
	Implementation Environment of the Current System	Descreve o ambiente físico e tecnológico no qual o produto será instalado.
	Partner or Collaborative Applications	Descreve as aplicações que não fazem parte do produto, mas com a qual o produto irá colaborar. Eles podem ser aplicações externas, pacotes comerciais ou aplicações internos pré-existentes.
	Off-the-Shelf Software	Descreve os <i>softwares</i> comerciais, <i>open sources</i> ou qualquer outro software off-the-shelf (OTS) que deve ser utilizado para implementar algum dos requisitos para o produto.
	Anticipated Workplace Environment	Este descreve o local de trabalho em que os utilizadores estão a trabalhar e utilizar o produto. Deve descrever quaisquer características do local de trabalho que poderia ter um efeito sobre o <i>design</i> do produto, os aspetos sociais e culturais do local do trabalho.
	Schedule Constraints	Quaisquer prazos conhecidos deve ser indicado aqui.
	Budget Constraints	Esta seção mostra o orçamento para o projeto, expressa em dinheiro ou recursos disponíveis.
	Enterprise Constraints	Esta seção contém requisitos que são específicos para a empresa que está a fazer o investimento no projeto.
Naming Conventions	Glossary of All Terms, Including Acronyms, Used	O glossário contém os significados de todos os nomes, siglas e abreviaturas utilizados pelos <i>stakeholders</i> .

and Terminology	by Stakeholders Involved in the Project	
Relevant Facts and Assumptions	Relevant Facts	Fatores que tem um efeito sobre o produto, mas não são requisitos de restrições obrigatórios.
	Business Rules	Estas são regras de negócios que possam ter um impacto sobre o trabalho/negocio/domínio que é a fonte dos requisitos. As regras do negócio será a alavanca para os requisitos.
	Assumptions	Uma lista de suposições que os desenvolvedores estão fazendo. Estas suposições podem ser sobre o ambiente operacional pretendido, mas pode ser qualquer coisa que tenha efeito sobre o produto. Como parte de gestão de expectativas, as suposições também contém declarações sobre o que o produto não vai fazer.
The Scope of the Work	The Current Situation	Esta é uma análise dos processos de negócio existentes, incluindo o manual e os processos automatizados que podem ser substituídos ou modificados pelo novo produto.
	The Context of the Work	O diagrama de contexto do trabalho identifica os limites do trabalho que é preciso investigar para ser capaz de construir o produto.
	Work Partitioning	Uma lista que mostra todos os eventos de negócio aos quais o trabalho responde. Eventos de negócios são acontecimento no mundo real que afetam o trabalho. A resposta para cada evento é denominado de business use case (BUC) . A lista de eventos inclui os seguintes elementos: <ul style="list-style-type: none"> ✓ Nome do evento; ✓ Entrada ou triggering data flow do sistema adjacente (idêntico ao nome no diagrama de contexto); ✓ Saída (s) para sistemas adjacentes (idêntica ao nome (s) no diagrama de contexto); ✓ Breve resumo do business use case (caso de usos);

		✓ Classes dos dados de negócio relevantes para este evento;
	Specifying a Business Use Case (BUC)	A especificação dos detalhes de como um Business Use Case (BUC) responde a um evento de negócios.
Business Data Model and Data Dictionary	Business Data Model	A especificação do assunto essencial, os objetos de negócio, entidades e classes que são pertinentes ao produto. Pode assumir a forma de um modelo de entidades e relacionamentos, diagrama de classes first-cut , ou qualquer outro tipo de modelo de dados.
	Data Dictionary	O glossário descrito anteriormente é o ponto de partida para o estabelecimento de um entendimento comum da terminologia. Assim que começar a definir o âmbito do inquérito é possível definir as entradas e saída de dados em um dicionário de dados formal.
The Scope of the Product	Product Boundary	Um diagrama de caso de usos identifica os limites entre os utilizadores (atores) e o produto.
	Product Use Case Table	O diagrama do escopo do produto é um resumo útil de todas as interfaces entre o produto e outros sistemas automatizados, organizações e utilizadores.
	Individual Product Use Cases	É onde é definida os detalhes sobre os casos de usos individuais do produto (PUCs) listados na tabela.
Functional Requirements	Functional Requirements	Esta é uma especificação de cada um dos requisitos funcionais atômicos. Como todos os tipos de requisitos atômicos (funcionais, não funcionais e de restrição), utiliza-se o Requirements Shell (figura 26) como guia para que cada atributo seja especificado.
	Look and Feel Requirements	Contém requisitos relacionados com o espírito do produto (ex: cor), bem como do estilo e a aparência.

<p>Non-functional Requirements</p>	<p>Usability and Humanity Requirements</p>	<p>Descreve as aspirações do cliente para a facilidade na utilização do produto, a preferência dos utilizadores na escolha da linguagem de preferência (Idioma), bem como o quão fácil deve ser para as pessoas com deficiências comuns o acesso ao produto. A usabilidade do produto é derivado das habilidades dos utilizadores sobre o produto e da complexidade das funcionalidades do mesmo.</p> <p>Os requisitos de usabilidade devem cobrir as propriedades, tais como:</p> <ul style="list-style-type: none"> ✓ Eficiência da utilização; ✓ Facilidade em lembrar como funciona o produto; ✓ Taxas de erro; ✓ Satisfação geral na utilização do produto; ✓ Feedback;
	<p>Performance Requirements</p>	<p>Especifica a quantidade de tempo necessário para que uma tarefa seja completada (Requisitos de velocidade e latência), a precisão produzido pelo produto (Requisitos de precisão e exatidão), a confiabilidade e disponibilidade necessário do produto (Requisitos de Confiabilidade e disponibilidade), a robustez do produto para continuar a funcionar em condições anormais (Requisitos de Robustez ou Tolerância a falhas), os volumes que o produto deve ser capaz de lidar e com a quantidade de dados armazenados no produto (Requisitos de Capacidade), a escalabilidade do produto e a capacidade do produto em lidar com novos volumes (Requisitos de escalabilidade e extensibilidade), e o tempo de vida esperado ao produto (Requisitos de Longevidade).</p>
	<p>Operational and Environmental Requirements</p>	<p>Especifica o ambiente físico no qual o produto vai operar e descreve os requisitos para interface com aplicações parceiros ou dispositivos que o produto necessita para operar com sucesso.</p>

	Maintainability and Support Requirements	Descreve o tempo necessário para efetuar alterações específicos ao produto, a especificação da entidade que fica encarregado de efetuar futuras manutenções ao produto (Requisitos de Manutenção), o nível de suporte que o produto requer (Requisitos de Suportabilidade), bem como a adaptabilidade do produto para outras plataformas (Requisitos de Adaptabilidade).
	Security Requirements	Especifica quem está autorizado a aceder ao produto (tanto a funcionalidades como a dados), em que circunstâncias que este acesso é concedido, e que partes do acesso do produto é permitido (Requisitos de Acesso), a integridade exigida ao produto. Especifica também o que o produto tem que fazer para garantir a privacidade dos indivíduos, cujas informações são armazenadas (Requisitos de Privacidade), os requisitos que o produto deve fazer para proteger-se contra <i>softwares</i> maliciosos, tais como, <i>virus, worms, malware, spyware</i> (Requisitos de Imunidade)
	Cultural Requirements	Contém requisitos que são específicos aos fatores socioculturais que afetam a aceitabilidade do produto.
	Compliance Requirements	Uma declaração especificando os requisitos legais para este sistema (Requisitos de Conformidade Legal).
Project Issues	Open Issues	Uma declaração de fatores que são incertos e podem fazer diferença significativa no produto.
	Off-the-Shelf Solutions	Esta sessão examina as soluções disponíveis e resume sua aplicabilidade aos requisitos. Consiste num pequeno estudo de viabilidade sobre as alternativas
	New Problems	Especifica as interfaces entre os novos sistemas e os existentes (Efeitos na instalação de sistemas), os potenciais problemas dos utilizadores, identificação dos problemas a lidar, bem como descobrir potenciais conflitos que não poderiam ser realizados até a data de implementação.

	Tasks	Que medidas devem ser tomadas para entregar o produto? Esta sessão destaca o esforço necessário para desenvolver o produto e descreve a especificação de cada fase de desenvolvimento.
	Migration to the New Product	Descreve os dados que vão ver modificados ou traduzido para o novo produto, bem como a lista de atividades convertidos e o calendário de implementação.
	Risks	Todos os projetos envolvem riscos, ou seja, o risco que qualquer coisa vai dar errado. O risco não é necessariamente uma coisa má, como nenhum processo é feito sem arriscar um bocado. O risco é apenas uma coisa má, se forem ignoradas e tornam-se problemas.
	Costs	O outro custo dos requisitos é a quantidade de dinheiro ou esforço que tem que ser gasto no produto. O custo pode ser avaliada a partir de um dos métodos de estimativa.
	User Documentation and Training	Esta sessão especifica a documentação do utilizador que será produzido como parte do esforço de construção do produto.
	Waiting Room	O Waiting Room detém os requisitos que não irão, por uma razão ou outra, fazer parte da versão inicial do produto.
	Ideas for Solutions	As ideias para a solução não são obviamente requisitos, mas é praticamente impossível após o levantamento de requisitos não ter ideias de como implementa-los.

Documento de Especificação de Requisitos Volere Requirements Specification Template v17

SDUMApp

1. The Purpose of the Project

1.1. The User Business or Background of the Project Effort

Os Serviços de Documentação (**SDUM**) são uma importante estrutura de apoio às atividades desenvolvidas na Universidade do Minho. Estes mantêm e disponibilizam, através da WEB, o catálogo bibliográfico da Universidade do Minho que referência toda a bibliografia, e outros recursos informativos, existentes nesta Universidade. Entre outras atividades, os SDUM garantem a gestão das bibliotecas da Universidade (funcionamento diário), tanto em Braga como em Guimarães.

A maioria dos utentes das bibliotecas da Universidade do Minho questionam sobre a existência de uma aplicação móvel, para que possam efetuar as renovações das suas requisições de uma forma mais rápida, eficaz e em qualquer altura e lugar, resultante também da nossa dependência, na utilização de aplicações móveis no dia-a-dia. Segundo um estudo efetuado pelo **Flury Analytics**, só em 2013 houve um aumento da utilização de aplicações em 115%.

O principal fator de incidência de multas aos utentes é a entrega fora de prazo, por motivos adversos. As entregas fora de prazo resultam basicamente do “esquecimento” dos utentes e da não visualização do aviso de devolução ou renovação do prazo de empréstimo (devido a não utilização do e-mail institucional da Universidade).

Daí, o feedback dos utentes no sentido que houvesse uma aplicação de fácil utilização, que permitisse, para além de efetuar as renovações das requisições em sua posse, também visualizar a partir de um calendário a data de entrega dessas requisições e receber notificações no dispositivo móvel alertando a data de entrega.

O objetivo deste projeto é fornecer aos utentes, uma aplicação móvel para proporcionar uma melhor gestão dos empréstimos e reservas nas bibliotecas da Universidade do Minho. Esta aplicação poderá ser instalada por todos os utentes com dispositivos móveis com sistema operativo Android.

Se a aplicação for bem-sucedido, os funcionários das bibliotecas vão economizar tempo, energia e esforço (tempo que poderá ser utilizado na realização de outras tarefas, nomeadamente na catalogação de novas publicações, bem como na maior assistência à sala de leitura, entre outras tarefas possíveis), de modo que os utentes passaram a efetuar as renovações ou reservas das publicações de forma mais cómoda e mais rápida, sem a necessidade de dirigirem aos balcões

das bibliotecas ou de efetuarem chamadas para esse fim. Os utentes por sua vez, também vão ter a vida facilitada, visto que além de economizar tempo e esforço, poderão evitar multas por entregas fora de prazo.

1.2. Goal of the Project

Os objetivos deste projeto são:

a)

Proposta	Vantagem
Permitir que os utentes efetuem a gestão de Empréstimos e Reservas de livros e outras publicações em qualquer lugar, desde que tenha um dispositivo móvel com acesso à internet.	Facilitar a vida dos funcionários das bibliotecas (técnicos da biblioteca), bem como dos utentes. Comodidade e rapidez em efetuar renovações ou reservas de publicações.

b)

Proposta	Vantagem
Disponibilização de um calendário com as datas de entregas das publicações. Possibilidade de sincronizar este mesmo calendário com o calendário do dispositivo móvel, de modo a receber notificações de aviso de devolução ou renovação.	Reduzir as incidências de multas (principalmente devido a não utilização do e-mail institucional da Universidade e consequentemente a não visualização do aviso de devolução ou renovação do prazo de empréstimo).

c)

Proposta	Vantagem
Permitir pesquisas rápidas e eficientes nos catálogos das bibliotecas da universidade e também reservas de exemplares.	Facilidade e rapidez em pesquisas nos catálogos.

2. The Stakeholders

2.1. The Client

O cliente é a pessoa ou organização que patrocinou o desenvolvimento do sistema. Neste projeto, o cliente é o **Serviço de Documentação da Universidade do Minho (SDUM)**. O desenvolvimento da aplicação foi proposto pela chefe de divisão de informação dos SDUM – Dr^a. Daniela Castro Ramalho.

2.2. The Customer

Os *customers* (consumidores) deste sistema são os utentes (alunos, professores, funcionários, investigadores, etc...) das bibliotecas da Universidade do Minho. Neste caso, o cliente e o consumidor, não são mesma pessoa, visto que o sistema a desenvolver não será para consumo interno dos funcionários das bibliotecas, mas sim para todos os utentes.

2.3. Other Stakeholders

Consiste em determinar as funções e possível, os nomes de pessoas e organizações que são afetadas pelo produto e engloba também todas as pessoas que de alguma forma podem influenciar no sucesso do projeto. A Figura B representa um mapa de *stakeholders* genérico.

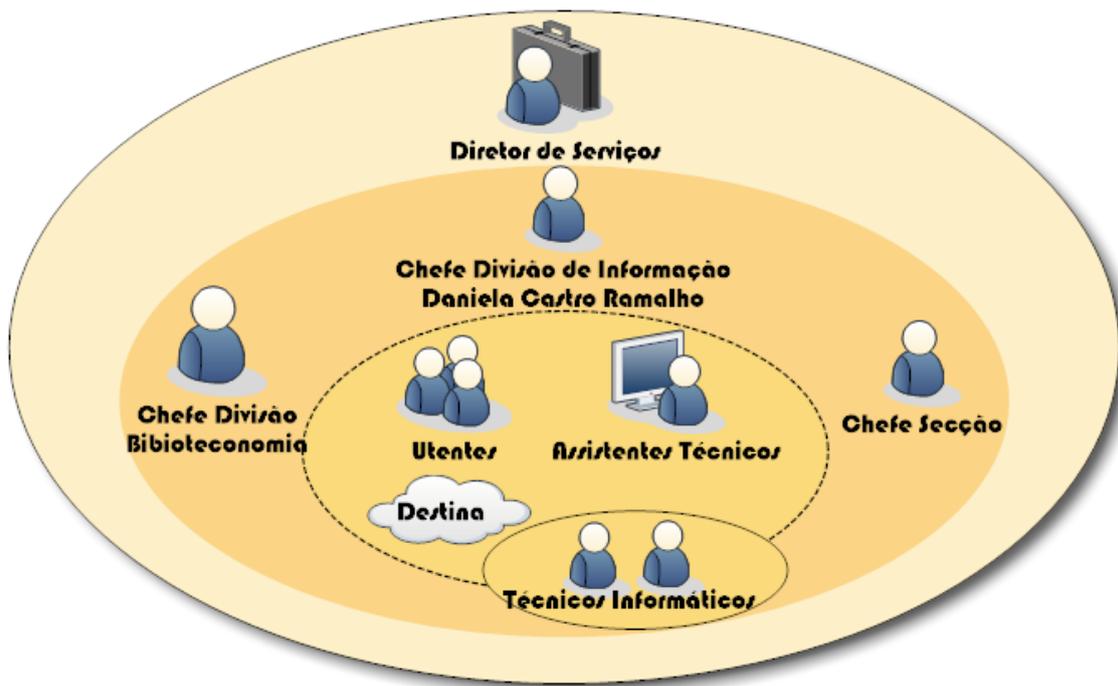


Figura B.1 - Mapa de Stakeholders (SDUM + Utentes)

Tabela B.1 - Utentes

<i>Stakeholder</i>	
Identificação	Utentes (<i>End Users</i>)
Conhecimento necessário para o projeto	Conhecimentos básicos de utilização da aplicação.
Conhecimento que contribuirá para o projeto	Baixo. Os utentes não envolvem no desenvolvimento da aplicação.
Grau de envolvimento necessário do <i>stakeholder</i>	Baixo/Nenhum.
Grau de influência do <i>stakeholder</i>	Alta. A aplicação a desenvolver consiste em satisfazer os utentes.

Tabela B.2 – Técnico Informático

<i>Stakeholder</i>	
Identificação	Técnico Informático (Nuno Fernandes)
Conhecimento necessário para o projeto	Competências de computação e tecnologias de Internet, testes de requisitos e especificações, especificação de requisitos e modelação de negócio.
Conhecimento que contribuirá para o projeto	Alto. Conhecimentos do servidor e dos processos de empréstimo e reserva de publicações.
Grau de envolvimento necessário do <i>stakeholder</i>	Alto. Disponibilização de URLs para ligação com o servidor, envolvimento na definição da engenharia de requisitos e nos testes das funcionalidades da aplicação.
Grau de influência do <i>stakeholder</i>	Médio. Envolve em todo o processo de especificação de requisitos, mas não tem poder de decisão sobre os mesmos.

Tabela B.3 – Chefe Divisão de Informática

<i>Stakeholder</i>	
Identificação	Chefe Divisão de Informação (Dr. ^a Daniela Castro Ramalho)
Conhecimento necessário para o projeto	Requisitos da aplicação e plano de projeto.
Conhecimento que contribuirá para o projeto	Baixo.
Grau de envolvimento necessário do <i>stakeholder</i>	Médio. Estabelece os critérios e as datas.
Grau de influência do <i>stakeholder</i>	Alto.

Tabela B.4 – Técnicos Assistentes de Biblioteca (Divisão de Biblioteconomia)

<i>Stakeholder</i>	
Identificação	Assistentes Técnicos (Divisão de Biblioteconomia).
Conhecimento necessário para o projeto	Básico.
Conhecimento que contribuirá para o projeto	Médio-Baixo. Conhecimento do funcionamento dos processos de gestão de empréstimos e renovações de publicações, mas não participam no desenvolvimento da aplicação.
Grau de envolvimento necessário do <i>stakeholder</i>	Baixo.
Grau de influência do <i>stakeholder</i>	Baixo. Não tem poder de decisão e influencia.

2.4. The Hands-On Users of the Product

Os utilizadores deste produto serão os utentes das bibliotecas da Universidade do Minho.

Tabela B.5 - Utentes

Stakeholder	
Nome utilizador/Categoria	Utentes
Função do utilizador	Utilização no dia-a-dia (na biblioteca ou qualquer lugar, desde que tenha conexão à internet).
Experiencia no assunto	Pode variar de iniciante a mestre
Experiencia tecnológica	Mestre – avançado
Habilidades intelectuais	Mestre – avançado
Géneros	Masculino e Feminino
Grupo etário	16 – 65 (Estudantes, Professores, Funcionários).

2.5. Personas

N/A

2.6. Priorities Assigned to Users

Os utentes são utilizadores chaves, visto que são essenciais para o sucesso contínuo do produto.

2.7. User Participation

As tabelas a seguir apresentadas representam a participação dos utilizadores no desenvolvimento do produto. Consiste em determinar o conhecimento expectável que estes utilizadores possam fornecer.

Tabela B.6 – User Participation (Chefe de Divisão de Informação)

User Participation	
Stakeholder	Chefe de Divisão de Informação
Nome	Dr. ^a Daniela Castro Ramalho
Contribuição	Prototipagem da interface, descrição dos requisitos (Requisitos funcionais e não funcionais).

Tabela B.7 – User Participation (Técnico Informático)

User Participation	
Utilizador	Técnico informático
Nome	Nuno Fernandes
Contribuição	Prototipagem da interface, descrição de requisitos (Requisitos funcionais e não funcionais), descrição do funcionamento do servidor (Aleph X-Server), especificação de URLs para pedido de serviço ao servidor, descrição das <i>tags</i> do XML devolvido pelo servidor, validação dos requisitos (realização de testes).

Tabela B.8 - User Participation (Assistentes Técnicos)

User Participation	
Utilizador	Assistentes Técnicos.
Nomes	Paula Cristina Claro, Cláudia Ribeiro, Rui Martins, Jorge Oliveira, Eduardo Santos e Maria Paula Marques.
Contribuição	Descrição do processo de gestão de empréstimos e reserva de publicações. Funcionamento de pesquisas no catálogo das bibliotecas da Universidade do Minho.

2.8. Maintenance Users and Service Technicians

O utilizador de manutenção será o técnico informático da Biblioteca da Universidade do Minho em Guimarães (BPG), Nuno Fernandes. Este deverá lidar com as futuras manutenções e alterações na aplicação.

3. Constraints

3.1. Solution constraints

As seguintes restrições da solução foram criadas baseadas no produto:

Tabela B.9 – Restrição da Solução 1

Restrições	
Descrição	O produto deve ser desenvolvimento para dispositivos móveis
Lógica	O produto desenvolvido para os utentes do Serviço de Documentação da Universidade do Minho. Facilita a gestão de empréstimos e reservas e pode ser acedido em qualquer lugar (desde que tenha acesso à internet).
Critério de ajuste	O produto deve funcionar corretamente em dispositivos móveis com sistema operativo Android.

Tabela B.10 – Restrição da Solução 2

Restrições	
Descrição	O produto deve interligar com o servidor já existe (Aleph X-Server). O Neste servidor é armazenada todas as informações administrativas, bibliográficas, relativas aos leitores e empréstimos interbibliotecas das bibliotecas da Universidade do Minho. O objetivo é o produto utilizar o mesmo servidor.
Lógica	O cliente não vai pagar para ter um novo servidor, nem perder tempo com a introdução dos dados novamente.
Critério de ajuste	Os dados que o produto terá acesso, são os mesmos armazenados no servidor.

3.2. Implementation Environment of Current System

Na Figura é especificado o sistema que pretende-se implementar.

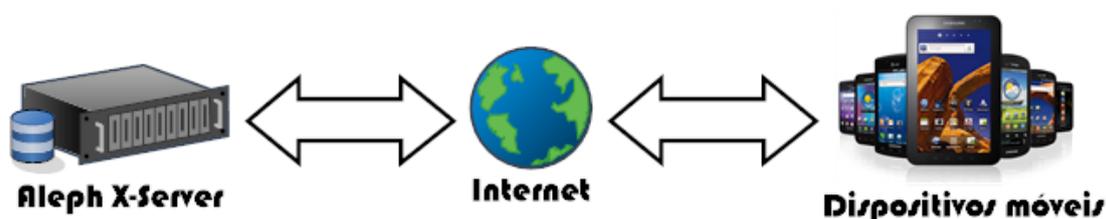


Figura B.2 – Sistema

O Aleph X-Server é o servidor utilizado atualmente pelos SDUM para armazenamento e gestão dos dados e continuará a ser utilizado pela aplicação. O suporte ao servidor é efetuado pela equipa de informática dos SDUM. A comunicação da aplicação com o servidor depende da conexão do dispositivo móvel com a internet (Wi-fi ou Redes móveis), possibilitando assim troca de informação entre os mesmos.

3.3. Partner or Collaborative Applications

A Tabela descreve uma aplicação que não faz parte do produto, mas este colabora com o mesmo.

Tabela B.11 – Aleph X-Server (Collaborative Application)

Collaborative Application	
Descrição	O produto deve interagir com o servidor já existente de modo a disponibilizar os dados aos utilizadores. O X-Server é um componente da arquitetura multi-tier dos produtos do Ex Libris que permite a conectividade com o sistema Aleph . A arquitetura multi-tier encontra-se dividido em duas camadas: Data Logic and Services e Application Logic . A primeira camada interage com a base de dados (Oracle RDBMS).
Modelo/Referência	Aleph X-Server (X-Server é constituído pela base de dados – Oracle RDBMS).
Lógica	O servidor recebe pedidos por parte da aplicação e devolve a resposta em formato pré-definido.

A Figura representa a interação da aplicação com o *collaborative application*.

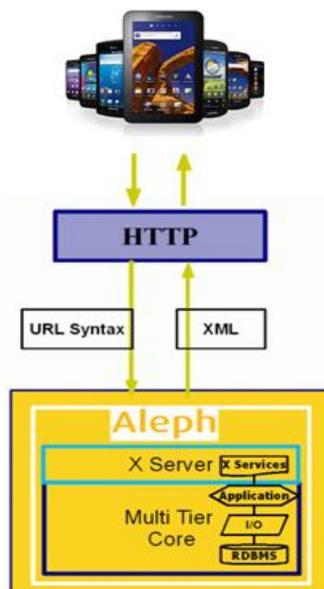


Figura B.2 - Collaborative Application

3.4. Off-the-Shelf Software

O único *software off-the-shelf* (OTS) necessário para implementar requisitos da aplicação será o Aleph X-Server.

3.5. Anticipated Workplace Environment

A aplicação poderá ser utilizada em qualquer ambiente, mas apenas em dispositivos moveis com sistema operativo Android.

3.6. Schedule Constraints

Não existe nenhum prazo a ter em conta. Esta aplicação deverá ser desenvolvida no âmbito de uma dissertação de mestrado e estará disponível para utilização depois deste estiver concluído.

3.7. Budget Constraints

O desenvolvimento desta aplicação não deverá ter nenhum custo associado.

3.8. Enterprise Constraints

N/A

4. Naming Conventions and Terminology

A seguir encontram-se o vocabulário com siglas e abreviaturas que foram identificadas antes deste projeto.

4.1. Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Tabela B.12 – Glossário de termos

Termos/Acrónimos	Descrição
Utentes	Todos os utilizadores inscritos nos Serviços de Documentação da Universidade do Minho
Catálogo	Inventário ou lista ordenado de publicações
ISBN	<i>Internation Standard Book Number</i> . É um indicador único para publicações.
SDUM	Serviços de Documentação da Universidade do Minho
Biblioteconomia	Área de conhecimento que estuda as praticas, perspetivas e as aplicações de representação e gestão do conhecimento em diferentes ambientes de informação (ex. Bibliotecas)
CDU	Classificação Decimal Universal. É um esquema internacional de classificação de documentos.
Publicações	Documento publicado por um determinado autor (Livros, dissertações, relatórios, etc...)
Exemplares	Cópias de publicações
BGUM	Biblioteca Geral da Universidade do Minho
BPG	Biblioteca da Universidade do Minho em Guimarães
BCE	Biblioteca de Ciências de Educação
BECS	Biblioteca da Escola de Ciências da Saúde
BD	Biblioteca de Direito
BNP	Biblioteca Nuno Portas
CDEUM	Centro de Documentação Europeia da Universidade do Minho

5. Relevant Facts and Assumptions

5.1. Assumptions

A seguir são apresentados um conjunto de pressupostos sobre o produto:

- Os utentes devem ter acesso à internet sempre que quiserem utilizar a aplicação;
- Todos os utentes devem ter conhecimento do idioma português ou inglês;
- Todos os utentes tem conhecimento do seu *username* e *password*;
- Os utentes devem instalar a aplicação nos respetivos dispositivos móveis;
- A aplicação irá funcionar em dispositivos móveis com sistema operativo Android;
- A aplicação deverá permitir que os utentes sincronizem o calendário de eventos (data de entrega das publicações) com o calendário do dispositivo móvel;
- A aplicação estará disponível para testes, no máximo 6 meses depois da data de início da sua implementação

6. The Scope of the Work

6.1. The Current Situation

Atualmente os SDUM não dispõe de nenhuma aplicação móvel de gestão de empréstimos e reservas de publicação e pesquisas no catálogo para os utentes das bibliotecas da Universidade do Minho, de modo que o único recurso disponível é o respetivo *Website* (www.sdum.uminho.pt). Este *Website* também poderá ser acedida no modo mobile, adequando assim as características dos dispositivos móveis (ecrã de pequenas dimensões).



Figura B.3 - Área pessoal Website SDUM

A área pessoal, após a autenticação de utentes, permite visualizar as publicações em sua posse, como tal também possibilita a sua renovação, as suas reservas de publicações, um histórico de publicações requisitadas, bem como visualizar o montante de penalizações monetárias devido à entrega de publicações fora do prazo.



Figura B.4 - Pesquisa no catálogo dos SDUM no Website

6.2. The Context of the Work

A Figura a seguir apresentada representa o diagrama de contexto do sistema de gestão de bibliotecas dos SDUM. Os diagramas de contexto mostram as relações estabelecidas entre o sistema e os interlocutores (Técnico Auxiliar e os utentes).



Figura B.5 - Diagrama de Contexto

6.3. Work Partitioning

Este corresponde às atividades realizadas nos SDUM, todas as relações estabelecidas com o sistema. Consiste numa apreciação geral do que já existe e quais as ações que são realizadas no sistema de gestão da biblioteca. Posteriormente na Seção 8 será efetuada a contextualização do produto final (solução SDUM).

Tabela B.13 – Lista de Eventos de Negócio

Nº	Nome do Evento	Entrada	Saída	Resumo do BUC
1	Utentes e os Técnicos Auxiliares da Biblioteca efetuam	Pesquisa publicações	Resultado da pesquisa	Pesquisa de publicações nos catálogos das bibliotecas da Universidade do Minho

	pesquisas de publicações			
2	Utentes e Técnicos Auxiliares efetuam reservas de publicações	Reserva publicações	Confirmação da reserva	Os utentes e os técnicos auxiliares podem reservar publicações a partir da pesquisa previamente efetuada.
3	Utentes e Técnicos Auxiliares efetuam renovações de publicações	Renovação publicações	Confirmação da renovação	Os utentes podem renovar às publicações em sua posse ou então solicitam os técnicos auxiliares a procederem a renovação destas mesmas publicações.
4	Os Técnicos Auxiliares efetuam o empréstimo de Publicações	Empréstimo Publicações	Confirmação do empréstimo	O empréstimo de publicações aos utentes é efetuado pelos técnicos auxiliares.
5	Os técnicos auxiliares efetuam a catalogação de publicações	Catalogação de publicações	Confirmação da tarefa	Os técnicos auxiliares poderão efetuar a inserção, a remoção ou a alteração de publicações nos catálogos das bibliotecas da Universidade do Minho, ou seja, efetuam a catalogação bibliográfica das publicações.
6	Os técnicos auxiliares podem efetuar a gestão de novos utentes	Gestão de novos utentes	Confirmação da tarefa	Os leitores externos devem proceder à sua inscrição nos serviços de documentação, de modo a poderem requisitar publicações. Esta

				inscrição é efetuada pelos Técnicos Auxiliares com o aval do chefe de divisão. Também se necessário, é possível a alteração dos dados dos utentes.
7	Os utentes efetuam a devolução de publicações em sua posse	Devolução de publicações	Confirmação de devolução	A devolução de uma publicação (no sistema) só pode ser efetuada por um Técnico Auxiliar da biblioteca, de modo que os utentes devem dirigir a uma biblioteca da Universidade do Minho e entregar a publicação para a devolução.
8	Os utentes efetuam o pagamento das penalizações monetárias (Multas)	Pagamento de Multas	Confirmação de pagamento	O pagamento do montante das multas referentes à entrega da publicação fora de prazo é efetuada nas bibliotecas. Deste modo, os utentes devem dirigir a uma das bibliotecas e efetuar o pagamento do montante. Enquanto o pagamento não for efetuado, o utente fica impossibilitado de efetuar a requisição de publicações

6.4. Specifying a Business Use Case (BUC)

Tabela B.14 – Especificação do BUC (Business Event 1)

Business Event 1: Utentes e os Técnicos Auxiliares da Biblioteca efetuam pesquisas de publicações

Business Use Case: Pesquisar de publicações

Trigger: Pesquisa publicações

Preconditions:

Interested Stakeholders: Utentes (alunos, professor, investigadores, funcionários, etc...), SDUM

Active Stakeholders: Utentes, Técnico Auxiliar

1. Caso 1

- O utente dirige-se a um balcão de umas das bibliotecas da Universidade do Minho e pede ajuda ao técnico auxiliar na pesquisa de uma publicação, fornecendo deste modo o nome e o autor, ou até mesmo um deles.
- O técnico auxiliar efetua a pesquisa da publicação no catálogo *online*.
- O técnico auxiliar entrega ao utente a localização do exemplar da publicação (CDU e biblioteca) e o código de barras do mesmo.
- Caso o exemplar pertença à biblioteca em que se encontra, o utente poderá dirigir-se a sala de leitura e procurar pela publicação. Se tiver dificuldade na procura, poderá pedir auxiliar ao técnico auxiliar.

2. Caso 2

- O utente efetua a autenticação (portal WEB) a partir do seu *email* (email institucional) e do password atribuído. Os credenciais dos alunos são os mesmos atribuídos pelos Serviços Académicos ou Serviços de Pós-graduação no ato de inscrição (Alunos). Os funcionários, professores e demais utentes, também devem ter o seu *username* e *password* para puderem efetuar requisições de publicações nas bibliotecas da Universidade do Minho.
- O utente preenche os campos, nomeadamente o tipo de pesquisa, tipo de documento, o catálogo e o tipo de documento, de modo a efetuar a pesquisa de publicações.

- O sistema retorna uma lista de publicações e seus exemplares (caso existam). É possível visualizar as informações detalhadas de cada publicação, nomeadamente a localização da publicação na biblioteca (CDU e biblioteca) e o código de barras.

Outcome: Resultado da pesquisa (lista de publicações e respetivas informações detalhadas).

Ex. Título, Autor, ID, Biblioteca, CDU, etc...

Tabela B.15 – Especificação do BUC (Business Event 2)

Business Event 2: Utentes e Técnicos Auxiliares efetuam reservas de publicações

Business Use Case: Reservar de publicações

Trigger: Reserva publicações

Preconditions: Necessária autenticação no portal WEB (www.sdum.uminho.pt) ou então entregar um documento de identificação com foto ao Técnico Auxiliar da Biblioteca; a publicação e seus exemplares devem estar indisponíveis, ou seja, em posse de algum utente. Caso exista exemplares nas estantes das bibliotecas, a reserva não é efetuada; não deverá ter multas para pagar, caso contrário fica ilegível para reservar livros.

Interested Stakeholders: Utentes, SDUM

Active Stakeholders: Utentes, Técnico Auxiliar

1. Caso 1

- O utente dirige-se a um dos balçães das bibliotecas e solicita a reserva de uma publicação ao Técnico Auxiliar. Caso já tenha efetuado a pesquisa do exemplar, facilita a vida ao Técnico. Caso contrário, é necessário efetuar a pesquisa da publicação.
- O Técnico Auxiliar solicita ao utente o cartão de aluno ou um documento de identificação com foto. Caso o utente disponibilize um documento de identificação, deve também disponibilizar o seu número de aluno ou de utente (restantes utentes).
- O Técnico Auxiliar procede a reserva da publicação.
- O sistema efetua a reserva da publicação caso a publicação encontrar-se na posse de um outro utente e também caso o utente não tiver nenhuma situação irregular com a biblioteca. O Técnico Auxiliar comunica o utente que a reserva da publicação foi efetuada com sucesso.

- Caso contrário, o sistema recusa a reserva da publicação. Neste caso, o Técnico Auxiliar comunica o utente que a reserva não pode ser efetuada e explica o porquê do sucedido de acordo com a notificação do sistema.

2. Caso 2

- O utente efetua a autenticação a partir do seu *username* e *password*.
- Efetua a pesquisa de uma publicação.
- É retornado uma lista de publicações e seus exemplares.
- Visualiza-se as informações do exemplar e a partir daí é possível efetuar o pedido de reserva do mesmo.
- O sistema verifica se o utente tem alguma multa por pagar. Caso tenha, é negada a reserva da publicação (as multas só podem ser pagas nos balcões das bibliotecas da Universidade do Minho).
- Caso contrário, é efetuada a reserva e o utente é notificado que o processo foi efetuado com sucesso.

Outcome: É efetuada ou não a reserva da publicação e o utente é notificado.

Tabela B.16 - Especificação do BUC (Business Event 3)

Business Event 3: Utentes e Técnicos Auxiliares efetuam renovações de publicações

Business Use Case: Renovar de publicações

Trigger: Renovação publicações

Preconditions: Necessária autenticação no portal WEB (www.sdum.uminho.pt) ou então entregar um documento de identificação com foto ao Técnico Auxiliar da Biblioteca; Ter pelo menos uma publicação em sua posse; Não ter nenhum caso de irregularidade (Multas ou Livros por entrega fora da data de entrega).

Interested Stakeholders: Utentes, SDUM

Active Stakeholders: Utentes, Técnico Auxiliar

1. Caso 1

- O utente dirige-se a um dos balcões das bibliotecas e solicita a renovação de publicações em sua posse ao Técnico Auxiliar.

- O Técnico Auxiliar solicita ao utente o cartão de aluno ou um documento de identificação com foto. Caso o utente disponibilize um documento de identificação, deve também disponibilizar o seu número de aluno ou de utente (restantes utentes).
- O Técnico Auxiliar procede a renovação da publicação.
- O sistema efetua a renovação da publicação caso o utente não tiver nenhuma situação irregular com a biblioteca. O Técnico Auxiliar comunica o utente que a renovação da publicação foi efetuada com sucesso.
- Caso contrário, o sistema recusa a renovação da publicação. Neste caso, o Técnico Auxiliar comunica o utente que a renovação não pode ser efetuada e explica o porquê do sucedido de acordo com a notificação do sistema.

2. Caso 2

- O utente efetua a autenticação a partir do seu *username* e *password*.
- Efetua a visualização da Lista de Empréstimos na Área Pessoal.
- Escolhe uma publicação requisitada e visualizar os seus dados. A partir daí é possível efetuar o pedido de renovação desta mesma publicação, ou então poderá também renovar todas as publicações em sua posse sem ter que visualizar os dados de uma publicação em concreto.
- O sistema verifica se a data de entrega já foi ultrapassada ou não. Caso a data já tenha sido ultrapassada, o utente é notificado que o processo de renovação não foi efetuado com sucesso. O utente deverá entregar a publicação numa dos balçoes das bibliotecas da Universidade do Minho. Caso tenha ultrapassado um dia, o utente não pagará nenhuma multa, visto que a multa é efetuada após um dia de tolerância.
- Caso contrário, é verificado se o livro já foi reservado. Se estiver reservado, o utente deverá entregar a publicação num dos balçoes das bibliotecas até a data que termina a requisição.
- Se a publicação não estiver reservada e o utente não tiver nenhum caso de incumprimento com a biblioteca, a publicação é renovada e este é notificado sobre o sucesso do processo de renovação.

Outcome: É efetuada ou não a renovação da publicação e o utente é notificado.

Tabela B.17 - Especificação do BUC (Business Event 4)

<p>Business Event 4: Os Técnicos Auxiliares efetuam o empréstimo de Publicações</p> <p>Business Use Case: Requisição de Publicação</p> <p>Trigger: Empréstimo Publicações</p> <p>Preconditions: Necessária autenticação (entregar um documento de identificação com foto ao Técnico Auxiliar da Biblioteca); Não ter nenhuma situação irregular (Multa ou Bloqueado).</p> <p>Interested Stakeholders: Utentes, SDUM</p> <p>Active Stakeholders: Técnicos Auxiliares</p> <ul style="list-style-type: none">- O utente dirige-se a um dos balçães das bibliotecas e solicita a requisição de publicações ao Técnico Auxiliar.- O utente entrega a publicação que pretende requisitar ao Técnico.- O Técnico Auxiliar solicita ao utente o cartão de aluno ou um documento de identificação com foto. Caso o utente disponibilize um documento de identificação, deve também disponibilizar o seu número de aluno ou de utente (restantes utentes).- O Técnico Auxiliar procede o empréstimo de publicações.- O sistema efetua o empréstimo de publicações caso o utente não tiver nenhuma situação irregular com a biblioteca. O Técnico Auxiliar assina e entrega ao utente o comprovativo de requisição (duplicado) para assinar também. A biblioteca fica com o original e o utente fica com um o duplicado.- Caso contrário, o sistema recusa o empréstimo da publicação se o utente tiver alguma situação irregular. Neste caso, o Técnico Auxiliar comunica o utente que a requisição da publicação não pode ser efetuada e explica o porquê do sucedido de acordo com a notificação do sistema. <p>Outcome: É efetuada o empréstimo de publicações ao utente</p>

Tabela B.18 - Especificação do BUC (Business Event 5)

<p>Business Event 5: Os técnicos auxiliares efetuam a catalogação de publicações</p> <p>Business Use Case: Catalogar publicações</p> <p>Trigger: Catalogação de publicações</p> <p>Preconditions: O Técnico Auxiliar deve ter autorização (privilégios no sistema) para efetuar a catalogação de publicações. Necessária autenticação do Técnico Auxiliar no sistema.</p> <p>Interested Stakeholders: SDUM</p> <p>Active Stakeholders: Técnico Auxiliar</p> <ul style="list-style-type: none">- O Técnico Auxiliar efetua a autenticação a partir do seu <i>username</i> e <i>password</i>.- Efetua a catalogação das publicações consoante o assunto. <p>Outcome: É efetuada a catalogação da publicação</p>

Tabela B.19 - Especificação do BUC (Business Event 6)

<p>Business Event 6: Os técnicos auxiliares podem efetuar a gestão de novos utentes</p> <p>Business Use Case: Inserir novos utentes, alteração dos dados pessoais dos novos utentes</p> <p>Trigger: Gestão de novos utentes</p> <p>Preconditions: O Técnico Auxiliar deve ter autorização (privilégios no sistema) para efetuar a catalogação de publicações. Necessária autenticação do Técnico Auxiliar no sistema.</p> <p>Interested Stakeholders: SDUM</p> <p>Active Stakeholders: Técnico Auxiliar</p> <ul style="list-style-type: none">- O individuo dirige-se a um dos balções das bibliotecas e solicita o seu registos nos Serviços de Documentação da Universidade do Minho. Se for um antigo aluno da Universidade do Minho, pode continuar a ser utentes das bibliotecas da universidade.- O Técnico Auxiliar entrega-o uma ficha de inscrição para que o mesmo preencha. Caso for um membro da Associação dos Antigos Estudantes da UM, ao abrigo do protocolo estabelecido entre os SDUM e a AAEUM, após o preenchimento da ficha de utilizador e a apresentação do cartão da AAEUM, ficará inscrito como utente. Se não for membro da AAEUM, deverá então apresentar um documento comprovativo da sua ligação à UM. Como antigo aluno da UM, o seu pedido será despachado favoravelmente, mas terá de pagar a taxa de emissão de cartão de leitor externo (7.50€).

- Caso não for membro da comunidade da Universidade, e necessitar de utilizar as bibliotecas da universidade, deverá solicitar a inscrição como leitor externo. Deverá também preencher a ficha de utilizador e fazê-lo acompanhar por uma exposição das razões que justificam a sua necessidade de utilizar as bibliotecas da UM. No caso do seu pedido ser despachado favoravelmente, terá ainda de proceder ao pagamento da taxa de emissão de cartão de leitor externo (70€).

Outcome: É efetuado a inscrição de novos utentes

Tabela B.20 - Especificação do BUC (Business Event 7)

Business Event 7: Os utentes efetuam a devolução de publicações em sua posse

Business Use Case: Devolver publicações

Trigger: Devolução de publicações

Preconditions: Ter publicações em sua posse;

Interested Stakeholders: Utentes

Active Stakeholders: Utentes, Técnico Auxiliar

- O utente dirige-se a um dos balçoes das bibliotecas e solicita a devolução de publicações ao Técnico Auxiliar. Entrega as publicações que pretender devolver.
- O Técnico Auxiliar procede a devolução da (s) publicação (ões) no sistema.
- O sistema efetua o devolução de publicações. Verifica se a (s) publicação (ões) foram entregues fora de data. Caso afirmativo, se foram entregues um dia após a data de devolução, não é aplicada nenhuma multa. A publicação é devolvida com sucesso e o Técnico Auxiliar entrega ao utente o comprovativo de entrega, assinado pelo mesmo. A biblioteca fica com uma cópia deste mesmo comprovativo.
- Caso a entrega for efetuada com mais de um dia de atraso, o sistema efetua a alerta de que o utente tem uma multa a pagar devido a entrega fora do tempo. O Técnico Auxiliar por sua vez, só tem que notificar o utente que tem uma multa por pagar devido a entrega fora de tempo e o valor do mesmo.

Outcome: É efetuada a entrega da publicação

Tabela B.21 - Especificação do BUC (Business Event 8)

<p>Business Event 8: Os utentes efetuam o pagamento das penalizações monetárias (Multas)</p> <p>Business Use Case: Pagar Multas</p> <p>Trigger: Pagamento de Multas</p> <p>Preconditions: Ter multa para pagar devido a entrega de publicações após a data de entrega</p> <p>Interested Stakeholders: Utentes</p> <p>Active Stakeholders: Utentes, Técnico Auxiliar</p> <ul style="list-style-type: none">- O utente dirige-se a um dos balçães das bibliotecas e solicita o pagamento das multas em seu nome ao Técnico Auxiliar.- Fornece o número de utente ao Técnico auxiliar e paga a quantia da multa.- O Técnico Auxiliar assina e entrega ao utente o comprovativo de pagamento da multa. <p>Outcome: É efetuado o pagamento da multa de publicações entregues fora de tempo</p>
--

7. Business Data Model and Data Dictionary

7.1. Business Data Model

Consiste em determinar o diagrama de classes da solução final.

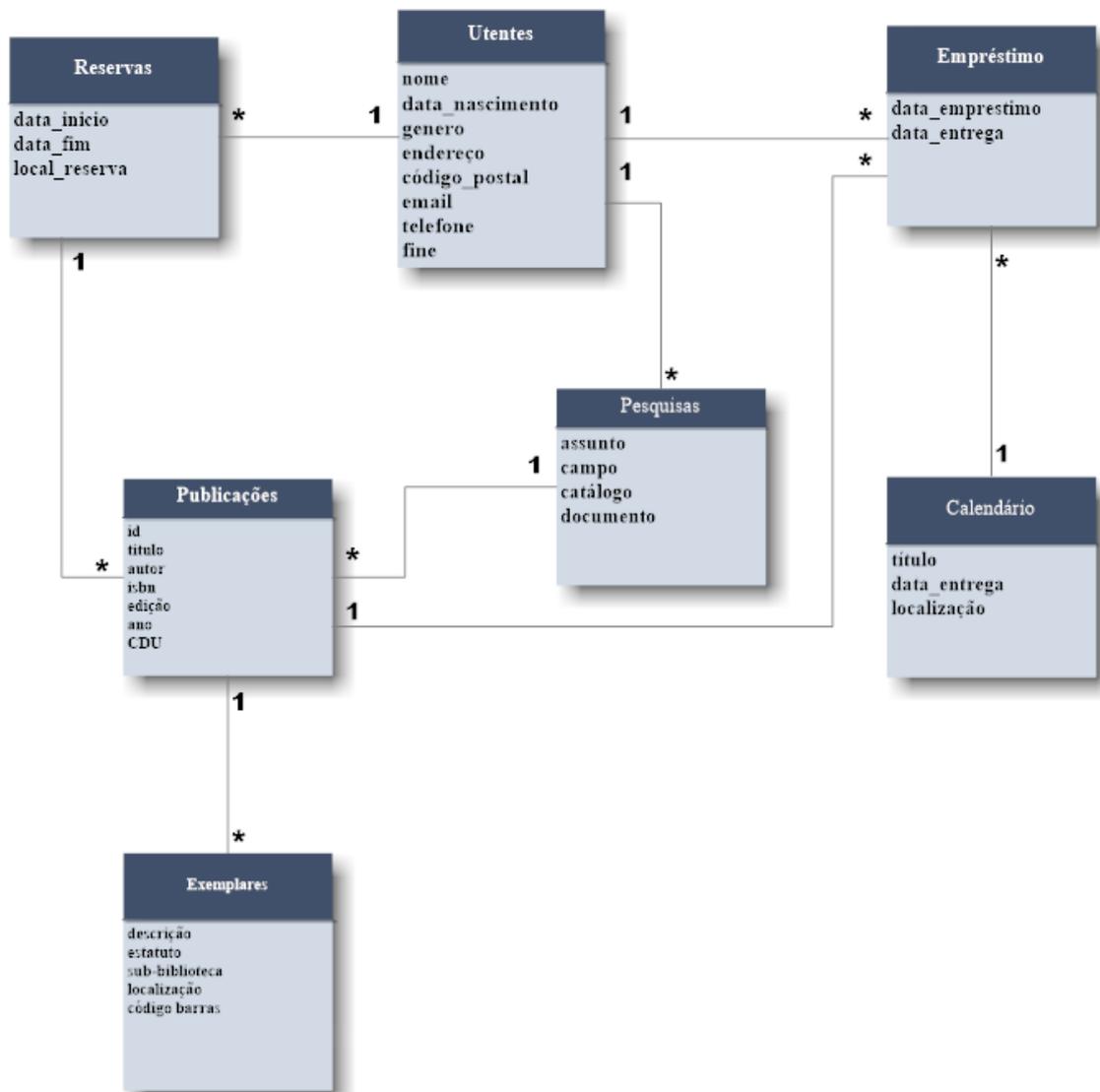


Figura B.6 - Diagrama de Classes

8. The Scope of the Product

8.1. Product Boundary

O diagrama de contexto a seguir apresentado na Figura mostra as relações estabelecidas entre a aplicação móvel e os utentes das bibliotecas. As entradas da aplicação são produzidas pelas entidades externas (utentes) e as saídas pelo próprio sistema.



Figura B.7 - Diagrama de Contexto do produto

A Figura a seguir apresentada descreve um cenário que mostra as funcionalidades da solução SDUM do ponto de vista do utilizador.

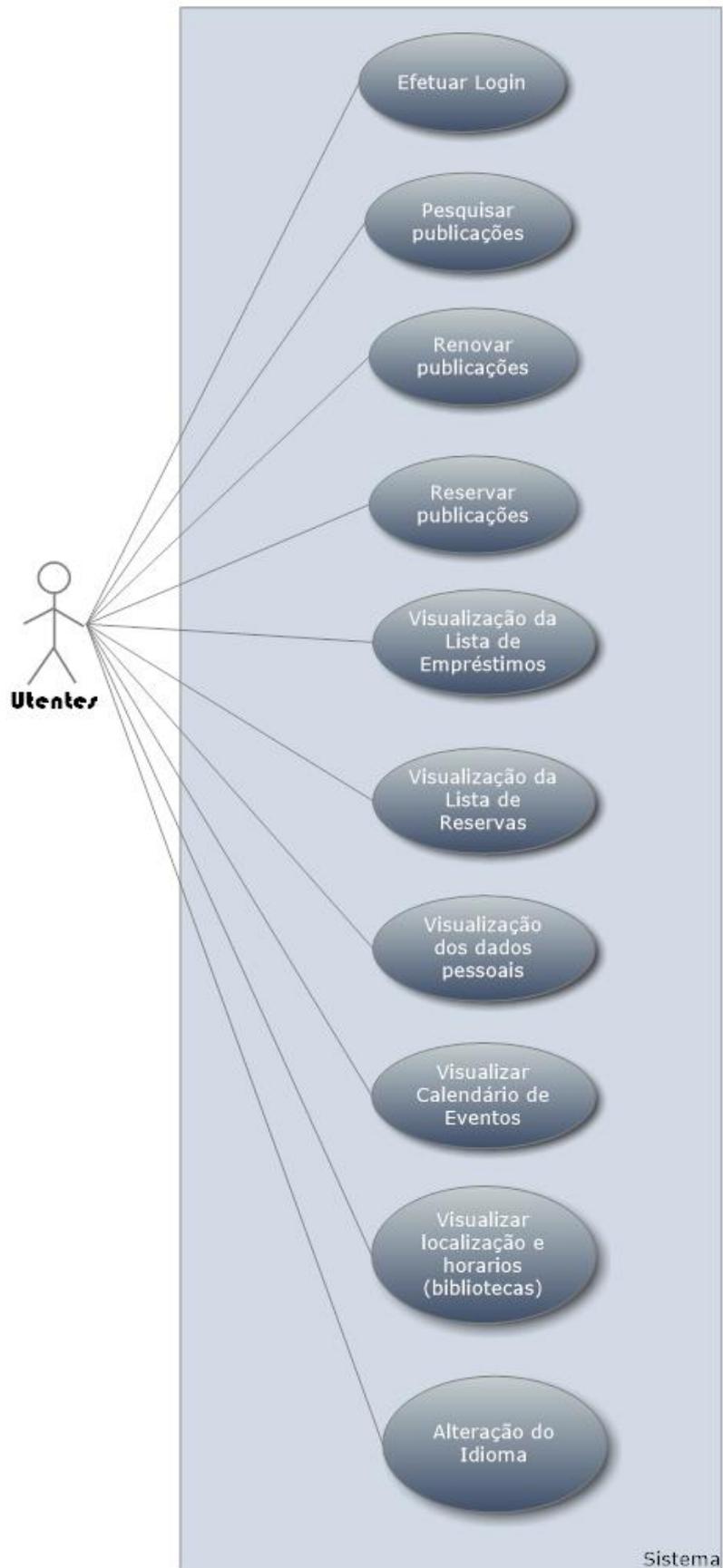


Figura B.8 - Diagrama de Casos de Uso do produto

8.2. Product Use Case Table

A Tabela a seguir apresentada correspondem ao caso de uso do produto. É um resumo de todas as interfaces entre o produto e os utilizadores.

Tabela B.22 – Tabela de Caso de Usos do Produto

Nº PUC	Nome PUC	Atores	Input	Output
1	Autenticação	Utentes	<i>Username</i> e <i>password</i>	Confirmação do acesso à conta
2	Visualizar dados pessoais	Utentes	Informação pessoal	Dados pessoais do utente
3	Visualizar Lista de Empréstimos	Utentes	Lista de Empréstimos	Publicações requisitados
4	Visualizar Lista de Reservas	Utentes	Lista de Reservas	Publicações reservadas
5	Efetuar a renovação de publicações	Utentes	Renovar publicações	Confirmação da renovação
6	Efetuar a reserva de publicações	Utentes	Reservar publicações	Confirmação da reserva
7	Efetuar pesquisas nos catálogos das bibliotecas	Utentes	Pesquisa de publicações	Lista de exemplares e respetivas informações
8	Visualizar o calendário de eventos	Utentes	Calendário de eventos	Calendário com datas de entrega de publicações
9	Visualizar os horários de funcionamento e as localizações das bibliotecas da Universidade do Minho	Utentes	Horário de funcionamento e localização	Lista de bibliotecas (Horário de funcionamento e localização)
10	Alteração do idioma	Utentes	Idioma	Idioma alterada

8.3. Individual Product Use Cases

8.3.1. Formal Use Case Specification including exception and alternatives

Tabela B.23 - Especificação do PUC (Product Use Case 1)

Product Use Case	
Use Case	UC1
Especificação do Use Case	Autenticação
Ator	Utentes
Descrição	Os utentes efetuam a autenticação utilizando o seu <i>username</i> e <i>password</i> , de modo a poderem aceder às funcionalidades da aplicação.
Pré-condições	<ol style="list-style-type: none">1. A aplicação deve estar em execução;2. O dispositivo móvel deve ter acesso à internet (Wi-fi ou Redes Móveis);
Cenário primário	<ol style="list-style-type: none">1. O utente introduz o <i>username</i> e <i>password</i> para autenticação;2. O servidor verifica se o utente existe na base de dados e valida a sua autenticação;
Cenário alternativo	<ol style="list-style-type: none">1. O <i>username</i> do utente não se encontra no formato (ex: A9999, F999, D99, B999, PG9999). O <i>username</i> corresponde ao número de utente e é composto por uma ou mais letras que designam o seu tipo de utilizador e por números;2. Autenticação negada;
Pós-condições	N/A

Tabela B.24 - Especificação do PUC (Product Use Case 2)

Product Use Case	
Use Case	UC2
Especificação do Use Case	Visualizar dados pessoais
Ator	Utentes

Descrição	Utentes poderão visualizar os seus dados pessoais, mediante a sua autenticação no sistema.
Pré-condições	1. Autenticação efetuada;
Cenário primário	1. O utente efetua a autenticação; 2. Seleciona a opção para visualização dos dados do utilizador; 3. Dados do utente (verificado e disponibilizado pelo servidor quando efetuou a autenticação);
Cenário alternativo	1. Autenticação negada;
Pós-condições	N/A

Tabela B.25 - Especificação do PUC (Product Use Case 3)

Product Use Case	
Use Case	UC3
Especificação do Use Case	Visualizar Lista de Empréstimos
Ator	Utentes
Descrição	Os utentes poderão visualizar a lista das publicações que têm em sua posse. Estas publicações podem ser renovadas à qualquer altura. Caso não for possível a renovação das mesmas, o utente é notificado pela aplicação.
Pré-condições	1. Autenticação efetuada; 2. Ter publicações em sua posse, ou seja, publicações requisitados;
Cenário primário	1. O utente efetua a autenticação; 2. Seleciona a opção para visualizar a lista de empréstimos; 3. Publicações e informações detalhadas (disponibilizado pelo servidor quando o utente efetuou a autenticação)
Cenário alternativo	1. Autenticação negada; 2. Não ter nenhuma publicação requisitado;
Pós-condições	1. As publicações na lista de empréstimos podem ser renovadas;

Tabela B.26 - Especificação do PUC (Product Use Case 4)

Product Use Case	
Use Case	UC4
Especificação do Use Case	Visualizar Lista de Reservas
Ator	Utentes poderão visualizar as publicações reservadas. Estas publicações poderão ser levantadas logo após a sua entrega.
Descrição	Os utentes poderão visualizar a lista das publicações reservadas.
Pré-condições	<ol style="list-style-type: none"> 1. Autenticação efetuada; 2. Ter publicações reservadas;
Cenário primário	<ol style="list-style-type: none"> 1. O utente efetua a autenticação; 2. Seleciona a opção para visualizar a lista de reservas; 3. Publicações e informações detalhadas (disponibilizado pelo servidor)
Cenário alternativo	<ol style="list-style-type: none"> 1. Autenticação negada; 2. Não ter nenhuma publicação reservada;
Pós-condições	N/A

Tabela B.27 - Especificação do PUC (Product Use Case 5)

Product Use Case	
Use Case	UC5
Especificação do Use Case	Efetuar a renovação de publicações
Ator	Utentes
Descrição	Os utentes poderão renovar as publicações em sua posse. A renovação só é efetuada caso a publicação não esteja reserva por outros utentes ou se não tiverem nenhuma situação irregular com os SDUM (não tiverem multa por entrega de publicações fora da data prevista para entrega ou não estiverem bloqueados devido ao mau comportamento nas bibliotecas da Universidade do Minho ou outros casos de maior gravidade);

Pré-condições	<ol style="list-style-type: none"> 1. Autenticação efetuada; 2. Ter livros em sua posse; 3. Não ter nenhuma situação irregular com os SDUM
Cenário primário	<ol style="list-style-type: none"> 1. O utente efetua a autenticação; 2. Seleciona a opção para visualizar a lista de empréstimos; 3. Seleciona uma publicação da lista de empréstimos; 4. Efetua o pedido de renovação da publicação. 5. O servidor verifica se é possível renovar a publicação e valida a renovação;
Cenário alternativo	<ol style="list-style-type: none"> 1. Autenticação negada; 2. Não ter nenhuma publicação requisitada; 3. Não ter nenhuma publicação requisitada; 4. Renovação não efetuada (Publicação reservada por outro utente ou situação irregular com os SDUM);
Pós-condições	N/A

Tabela B.28 - Especificação do PUC (Product Use Case 6)

Product Use Case	
Use Case	UC6
Especificação do Use Case	Efetuar a reserva de publicações
Ator	Utentes
Descrição	Os utentes poderão reservar as publicações. A reserva só poderá ser efetuada mediante à pesquisa de publicações nos catálogos das bibliotecas da Universidade do Minho. O processo de reserva só é concluído caso o utente não tiver nenhuma situação irregular com os SDUM ou se a respetiva publicação encontrar-se nas estantes (a reserva só é efetuada caso a publicação encontrar-se na posse de um outro utente).
Pré-condições	<ol style="list-style-type: none"> 3. Autenticação efetuada; 4. Efetuar pesquisa de publicações nos catálogos das bibliotecas;

Cenário primário	<ol style="list-style-type: none"> 1. O utente efetua a autenticação; 2. Seleciona a opção para pesquisar publicações; 3. Introduce os dados da pesquisa e efetua a pesquisa; 4. O servidor disponibiliza uma lista de publicações de acordo com a pesquisa efetuada; 5. O utente seleciona uma publicação e efetua o pedido de reserva. O servidor verifica a possibilidade de reserva da publicação e valida o mesmo;
Cenário alternativo	<ol style="list-style-type: none"> 1. Autenticação negada; 2. N/A 3. Dados introduzidos incorretamente 4. Publicação inexistente 5. Reserva não pode ser efetuada;
Pós-condições	N/A

Tabela B.29 - Especificação do PUC (Product Use Case 7)

Product Use Case	
Use Case	UC7
Especificação do Use Case	Efetuar pesquisas nos catálogos das bibliotecas
Ator	Utentes
Descrição	Os utentes poderão efetuar pesquisas de publicações nos catálogos das bibliotecas da Universidade do Minho
Pré-condições	<ol style="list-style-type: none"> 1. Autenticação efetuada;
Cenário primário	<ol style="list-style-type: none"> 1. O utente efetua a autenticação; 2. Seleciona a opção para pesquisar publicações; 3. Introduce os dados da pesquisa e efetua a pesquisa; 4. O servidor disponibiliza uma lista de publicações de acordo com a pesquisa efetuada;

Cenário alternativo	<ol style="list-style-type: none"> 1. Autenticação negada; 2. N/A 3. Dados introduzidos incorretamente 4. Publicação inexistente
Pós-condições	<ol style="list-style-type: none"> 1. Possível reservar publicações

Tabela B.30 - Especificação do PUC (Product Use Case 8)

Product Use Case	
Use Case	UC8
Especificação do Use Case	Visualizar o calendário de eventos
Ator	Utentes
Descrição	Os utentes poderão visualizar um calendário de eventos com as publicações e as respetivas datas de entrega. A partir deste calendário é possível sincronizar estes eventos com o calendário do dispositivo móvel, que irá possibilitar a obtenção de notificações.
Pré-condições	<ol style="list-style-type: none"> 1. Autenticação efetuada; 2. Ter publicações requisitadas;
Cenário primário	<ol style="list-style-type: none"> 1. O utente efetua a autenticação; 2. Seleciona a opção para visualizar calendário de eventos; 3. Calendário com publicações e respetivas datas de entrega (disponibilizadas pelo servidor);
Cenário alternativo	<ol style="list-style-type: none"> 1. Autenticação negada; 2. Não ter publicações requisitadas;
Pós-condições	<ol style="list-style-type: none"> 1. Possibilidade de sincronizar o calendário de eventos com o calendário do dispositivo móvel

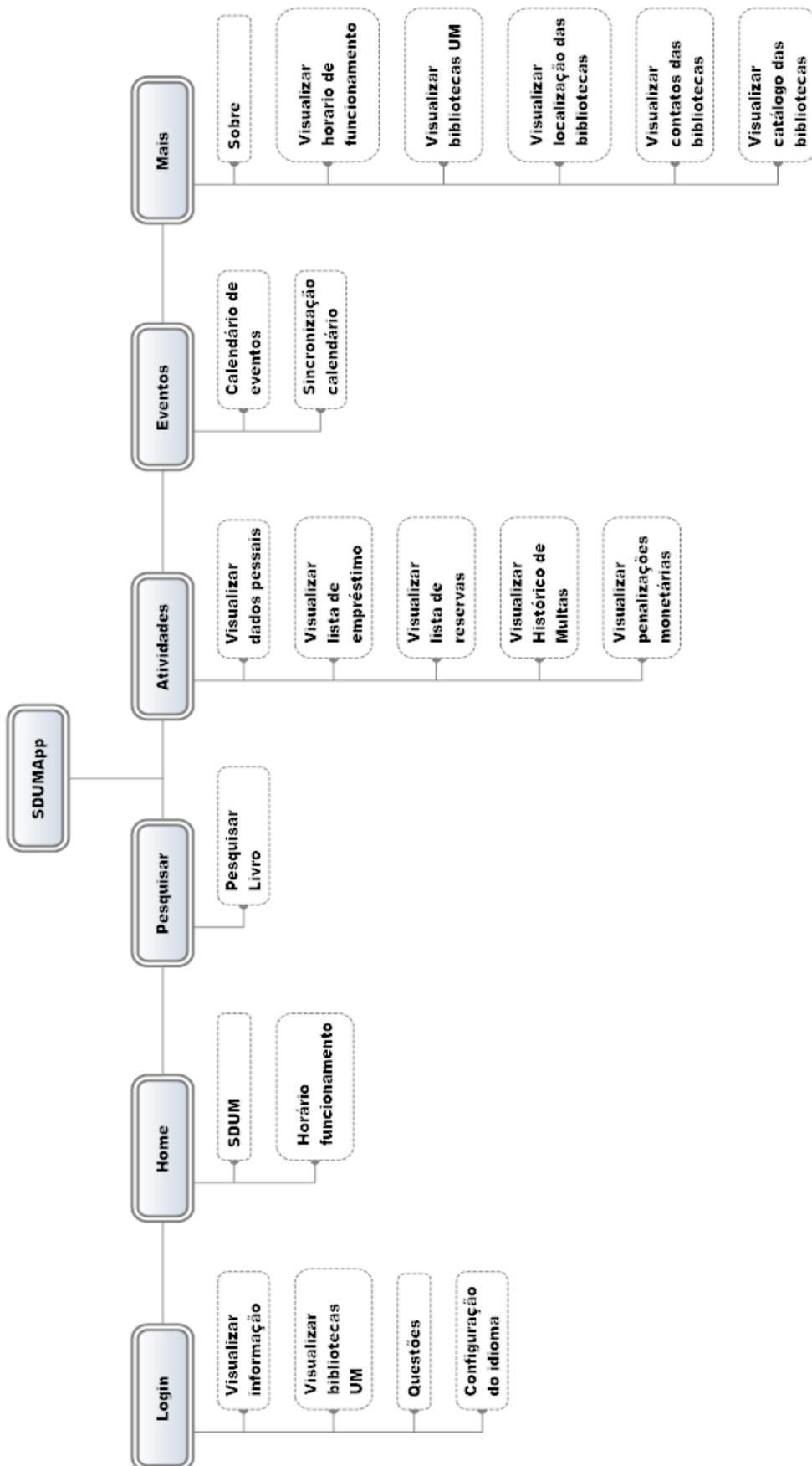
Tabela B.31 - Especificação do PUC (Product Use Case 9)

Product Use Case	
Use Case	UC9
Especificação do Use Case	Visualizar os horários de funcionamento e as localizações das bibliotecas da Universidade do Minho
Ator	Utentes
Descrição	Os utentes poderão visualizar as localizações das bibliotecas da Universidade do Minho, bem como o seu horário de funcionamento
Pré-condições	1. O dispositivo móvel deve ter acesso à internet;
Cenário primário	1. O utente seleciona a opção de visualização das bibliotecas UM e o seu horário de funcionamento;
Cenário alternativo	1. Sem conexão a internet
Pós-condições	N/A

Tabela B.32 - Especificação do PUC (Product Use Case 10)

Product Use Case	
Use Case	UC10
Especificação do Use Case	Alteração do idioma
Ator	Utentes
Descrição	Os utentes alterar o idioma da aplicação. O idioma pré-definido é o Português, mas também no entanto pode ser alterada para o Inglês.
Pré-condições	1. A aplicação deve estar em execução;
Cenário primário	1. O utente seleciona a opção do idioma (Português ou Inglês);
Cenário alternativo	N/A
Pós-condições	N/A

8.3.2. Storeboard



8.3.3. Hi-fi Prototype



Figura B.10 - Janela Login

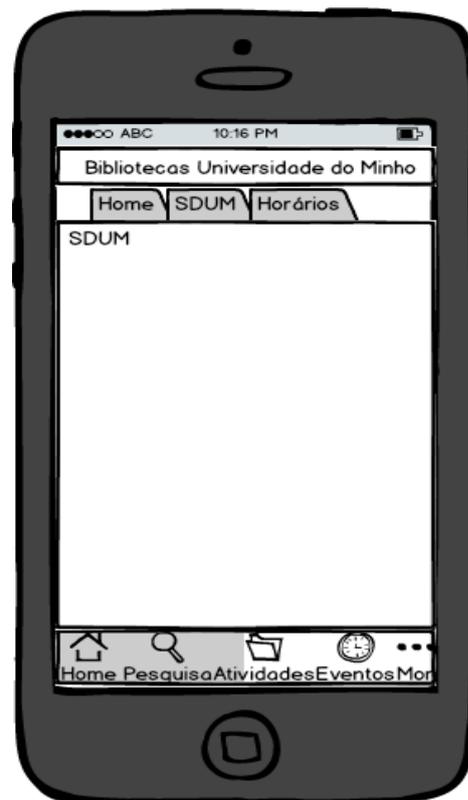


Figura B.9 - Janela Home

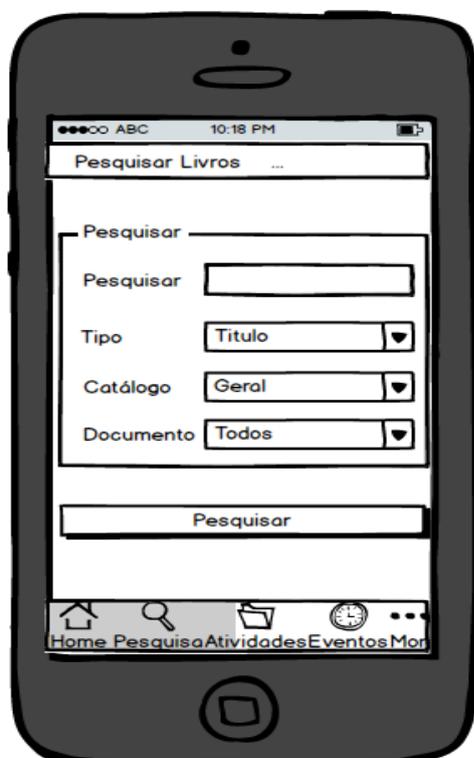


Figura B.11 - Janela Pesquisar

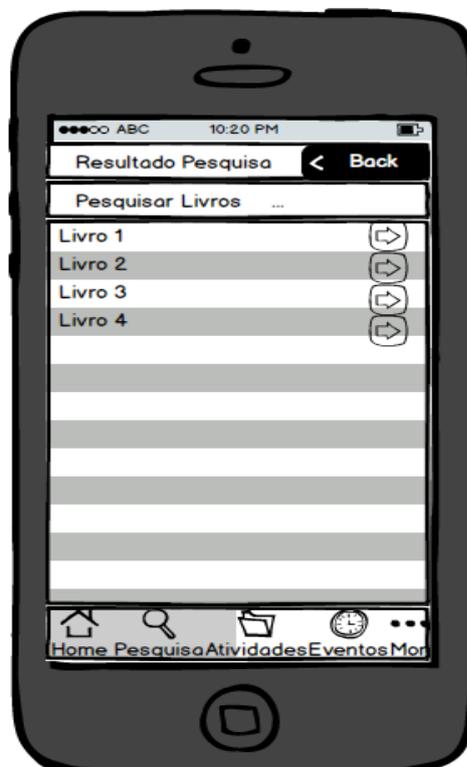


Figura B.12 - Janela com resultado da pesquisa



Figura B.14 - Dados pessoais do utente



Figura B.13 - Reserva de publicações



Tabela B.15 - Lista de Empréstimos

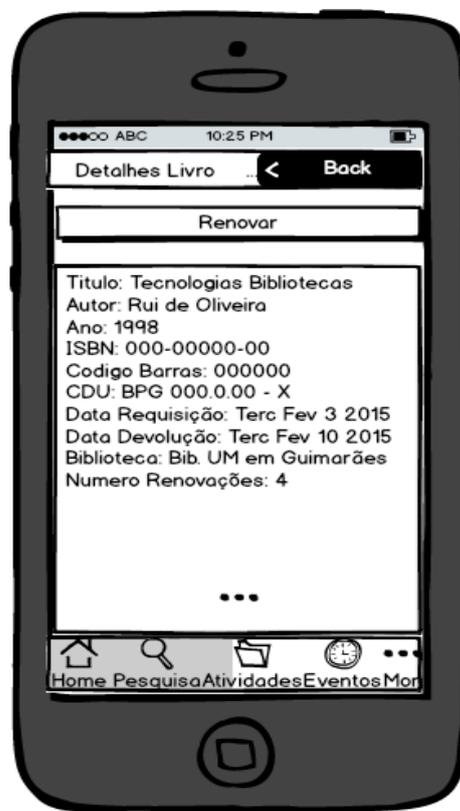


Figura B.16 - Renovar publicações

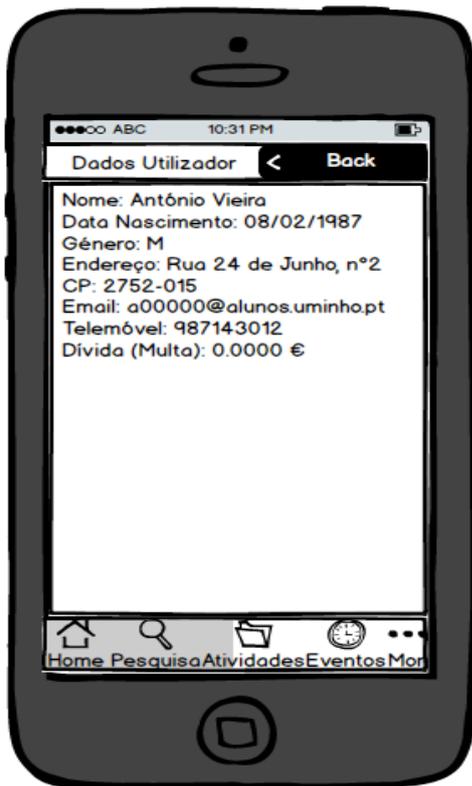


Figura B.18 - Visualização dos dados pessoais

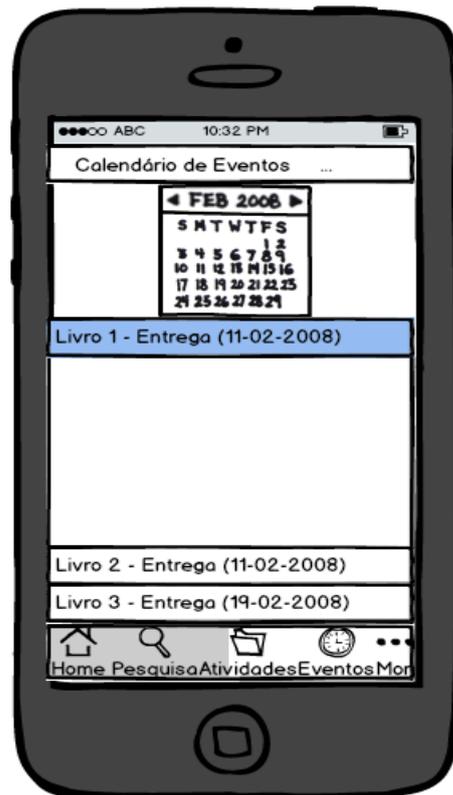


Figura B.17 - Calendário de Eventos

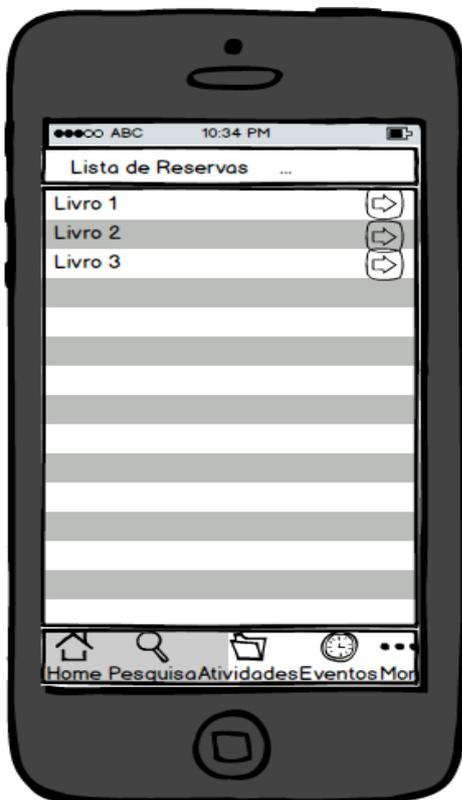


Figura B.20 - Lista de Reservas

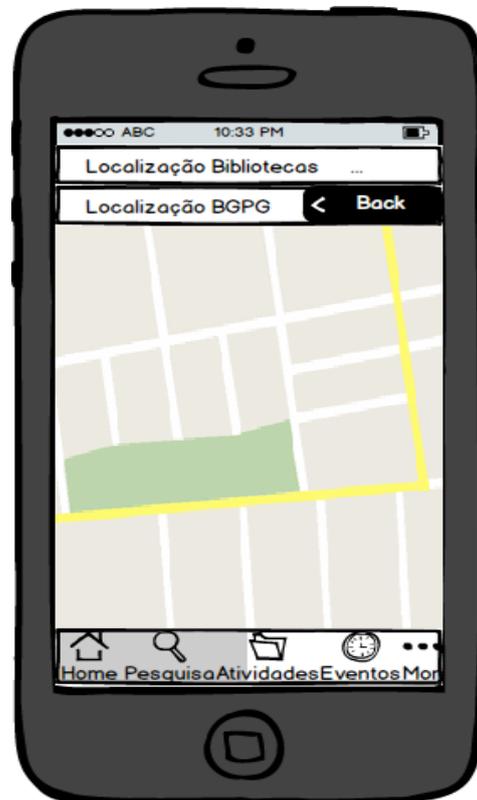


Figura B.19 - Localização das bibliotecas

8.3.4. Sequence Diagram

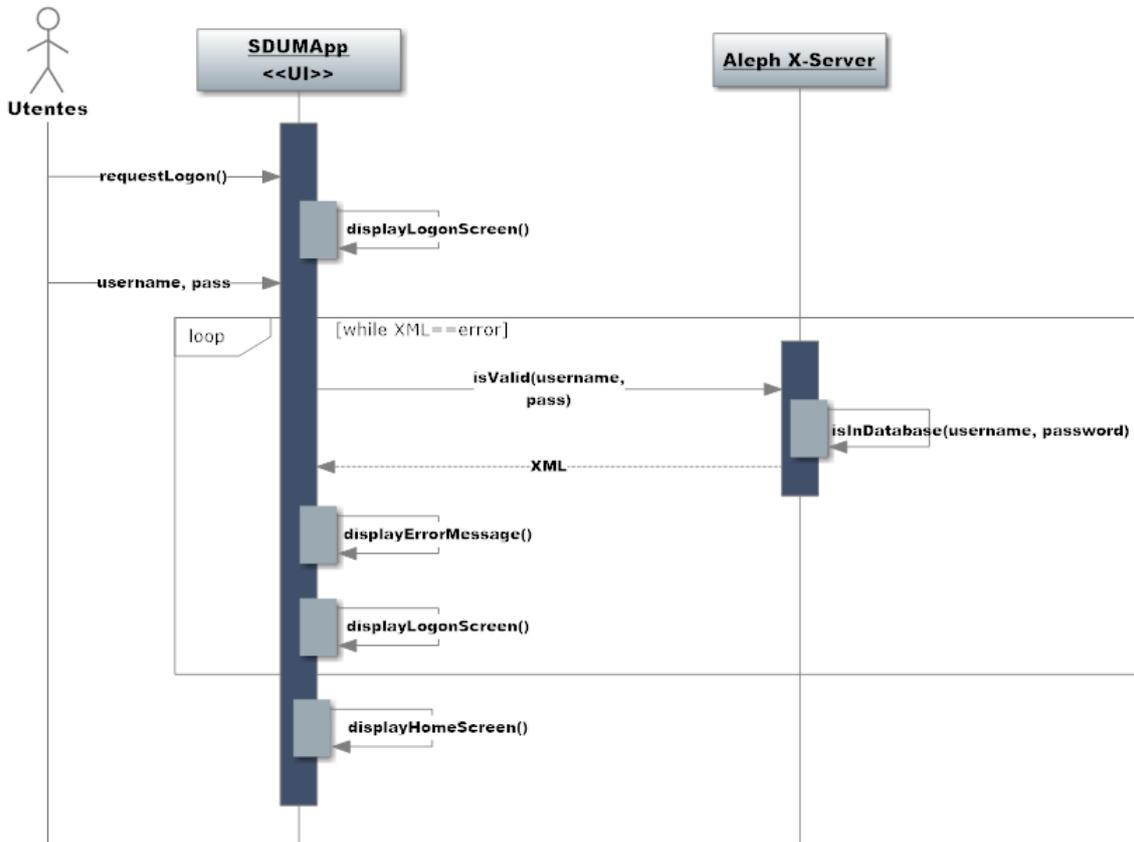


Figura B.21 – Diagrama de Sequencia (Autenticação)

A Figura representa o diagrama de sequência para a autenticação. O utilizador deverá introduzir os seus credencias e de seguida efetuar o pedido de autenticação. A aplicação por sua vez, efetua o pedido de autenticação do utilizador ao servidor. O servidor verifica a existência deste mesmo utilizador na base de dados e em caso afirmativo retorna um XML com os dados pessoais, os dados referentes às publicações na posse do utilizador, as publicações reservas, entre outros. Em caso de falha, o servidor retorna um XML de erro e a aplicação alerta o utilizador sobre a impossibilidade de efetuar a autenticação do mesmo.

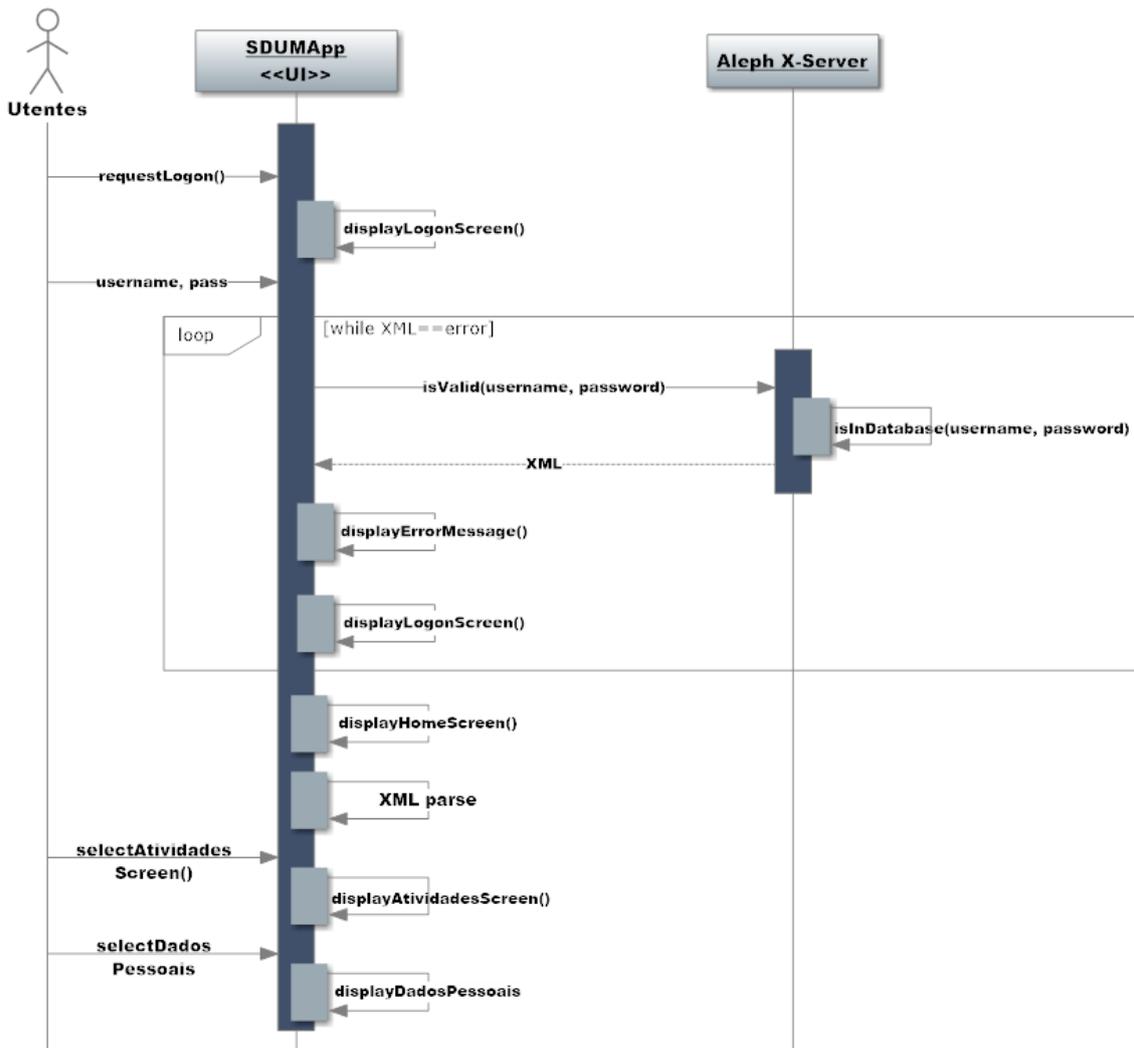


Figura B.22 – Diagrama de Sequência (Dados Pessoais)

A Figura corresponde ao diagrama de seqüência da visualização dos dados do utilizador. Para que o utilizador tenha acesso aos seus dados pessoais, é necessário que este efetue a sua autenticação na aplicação. Após efetua a autenticação, este deverá selecionar a janela Atividades e posteriormente selecionar a opção “Dados Pessoais”. Para a visualização dos dados pessoais, tanto como a verificação da lista de empréstimos e reservas não haverá necessidade de se efetuar o pedido ao servidor, de modo que após a autenticação do utente, o servidor devolve um XML com os dados pessoais, os dados referentes às publicações na posse do utilizador, as publicações reservas, entre outros. Estes dados são armazenados localmente, de modo que possam ser visualizados pelos utentes.

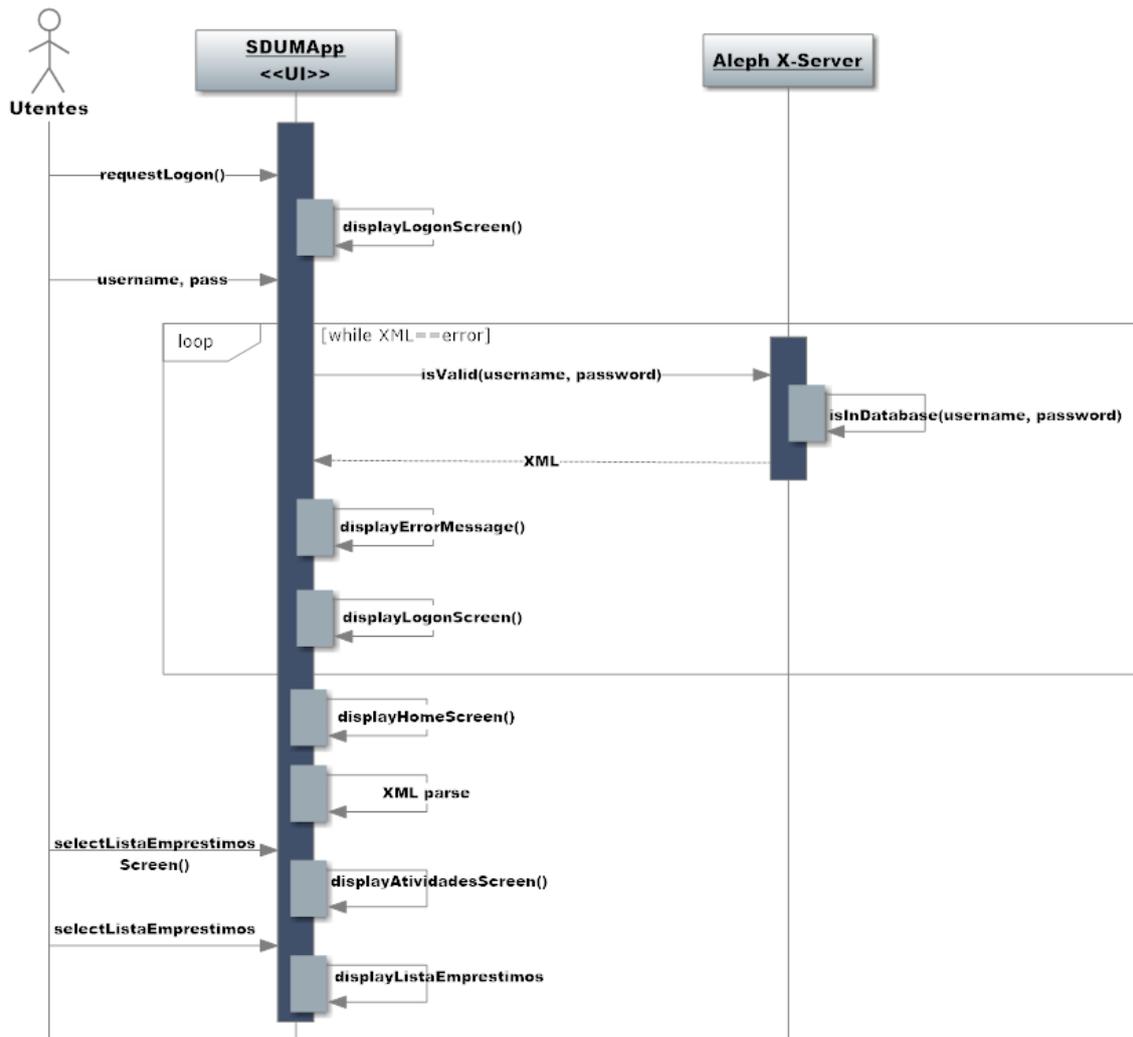


Figura B.23 – Diagrama de Sequencia (Lista de Empréstimos)

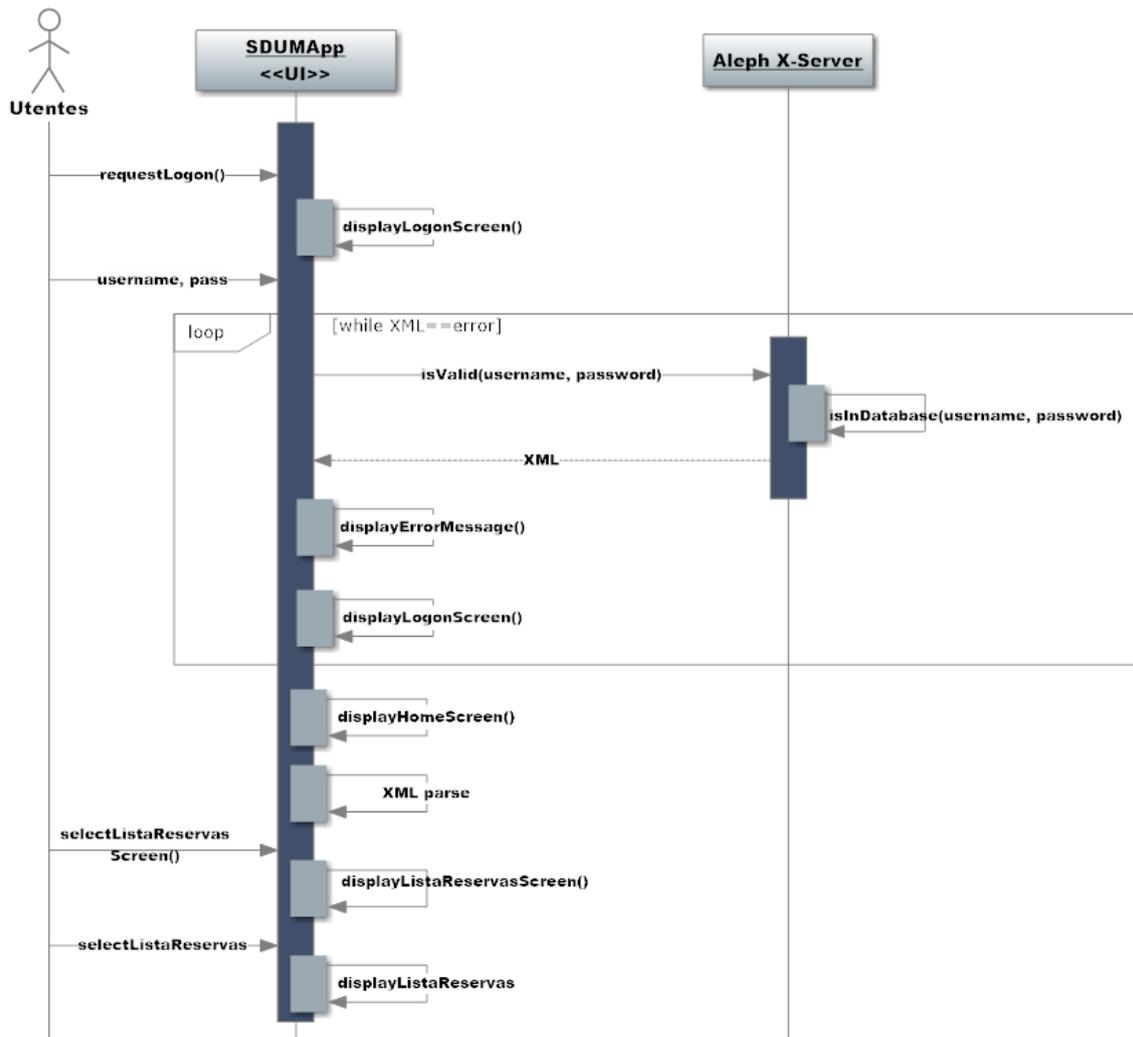


Figura B.24 – Diagrama de Sequencia (Lista de Reservas)

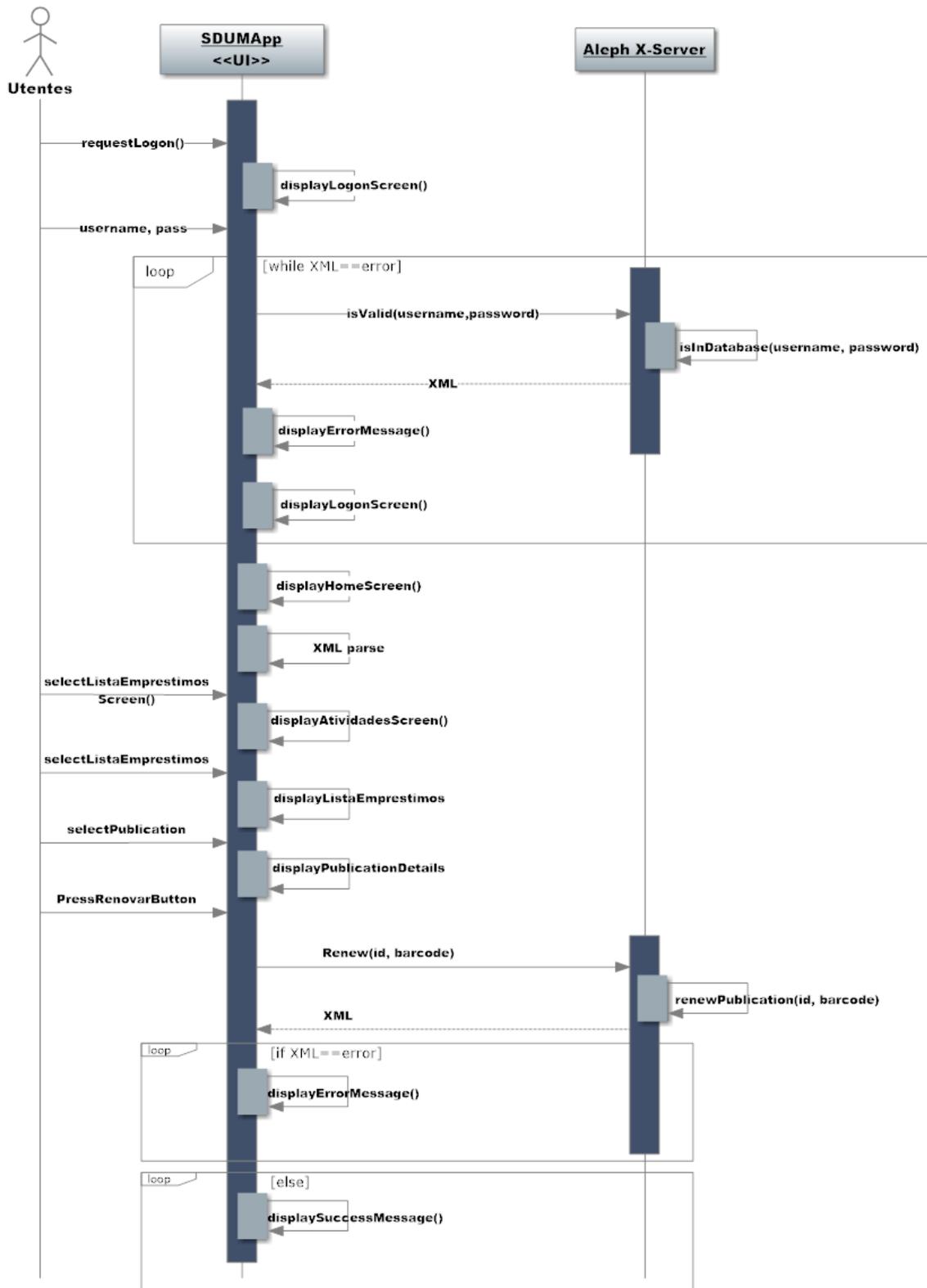


Figura B.25 – Diagrama de Sequencia (Renovação de Publicações)

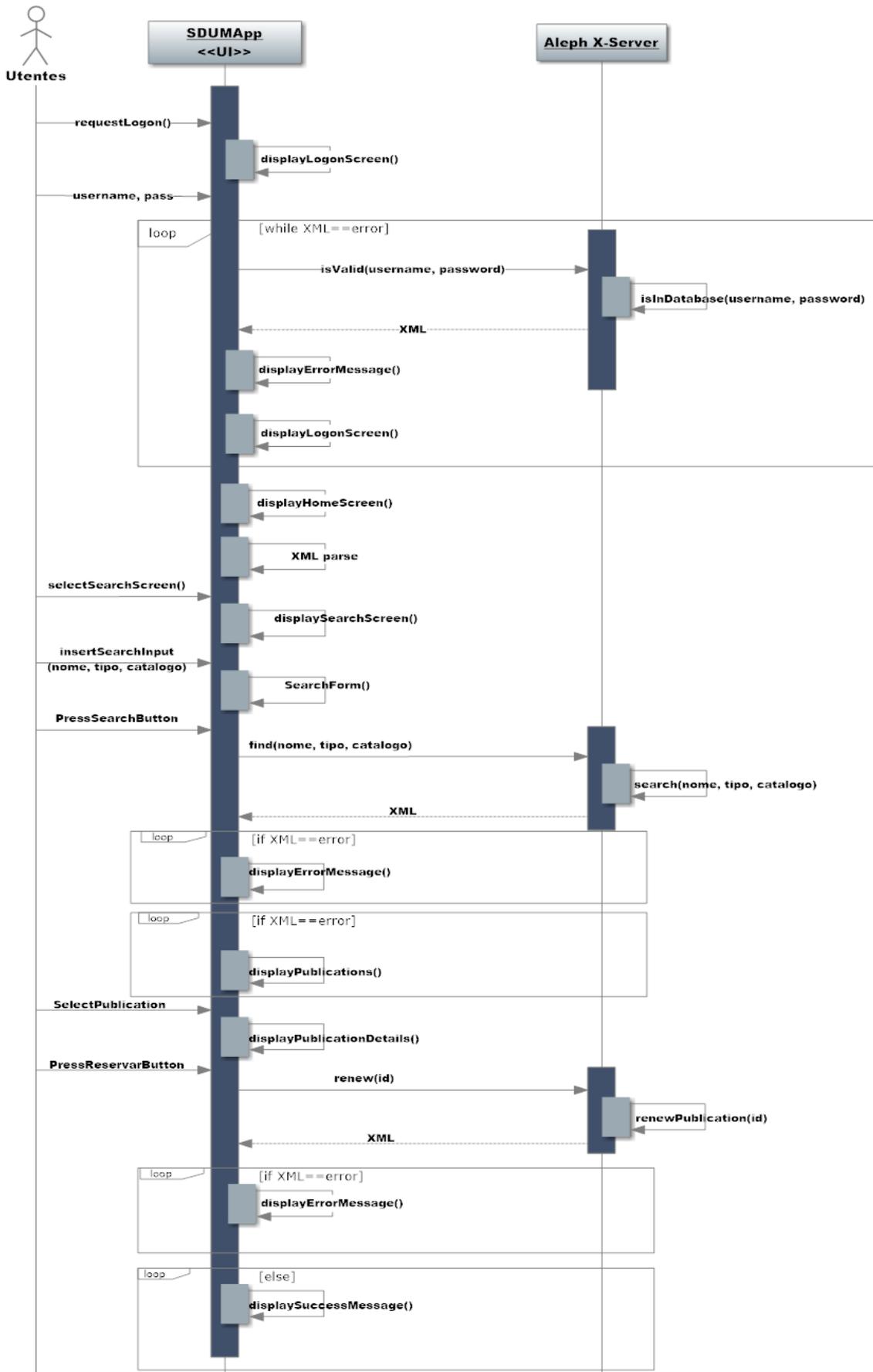


Figura B.26 – Diagrama de Sequencia (Reserva de Publicações)

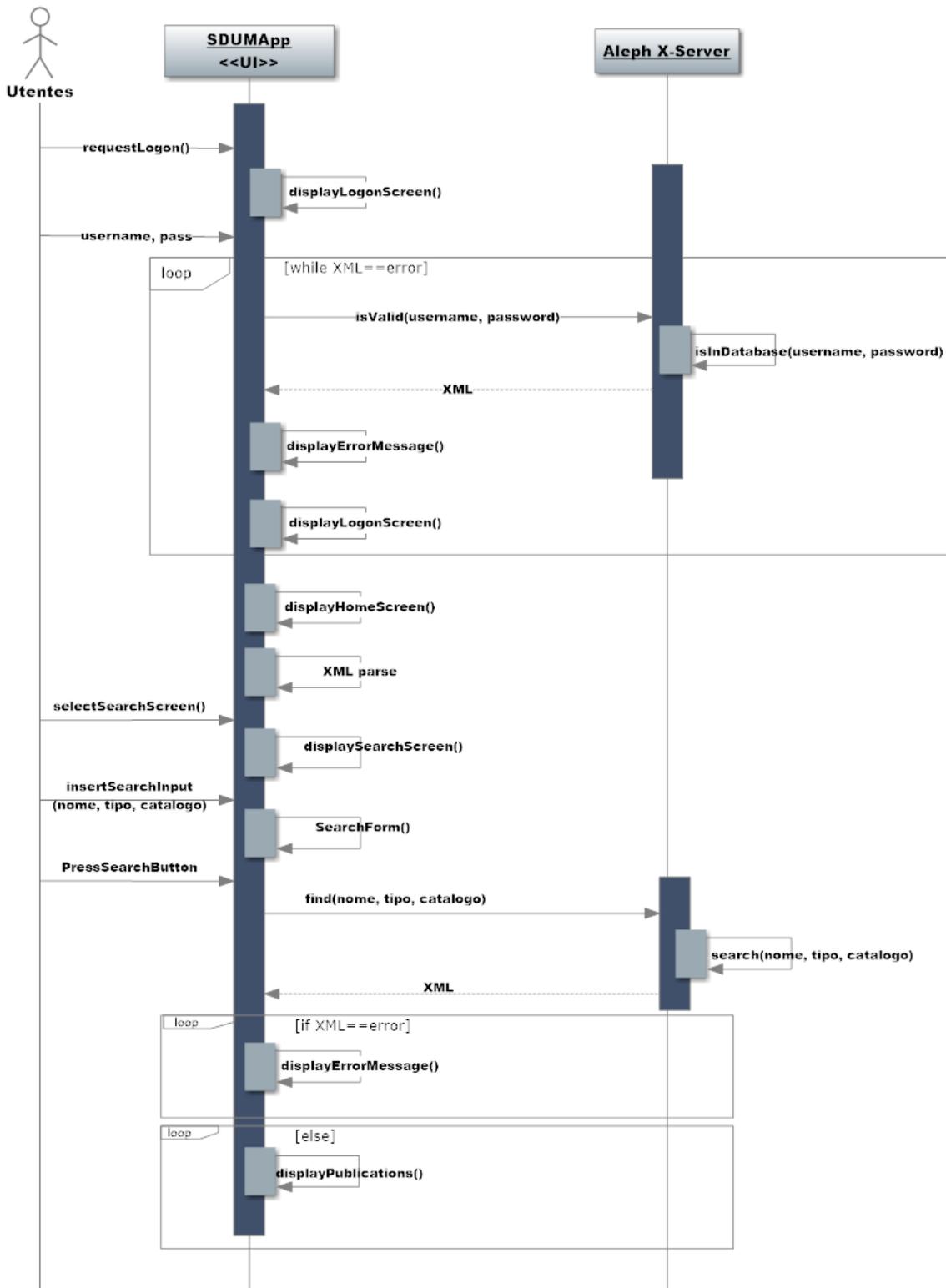


Figura B.27 – Diagrama de Sequencia (Pesquisas nos Catálogos)

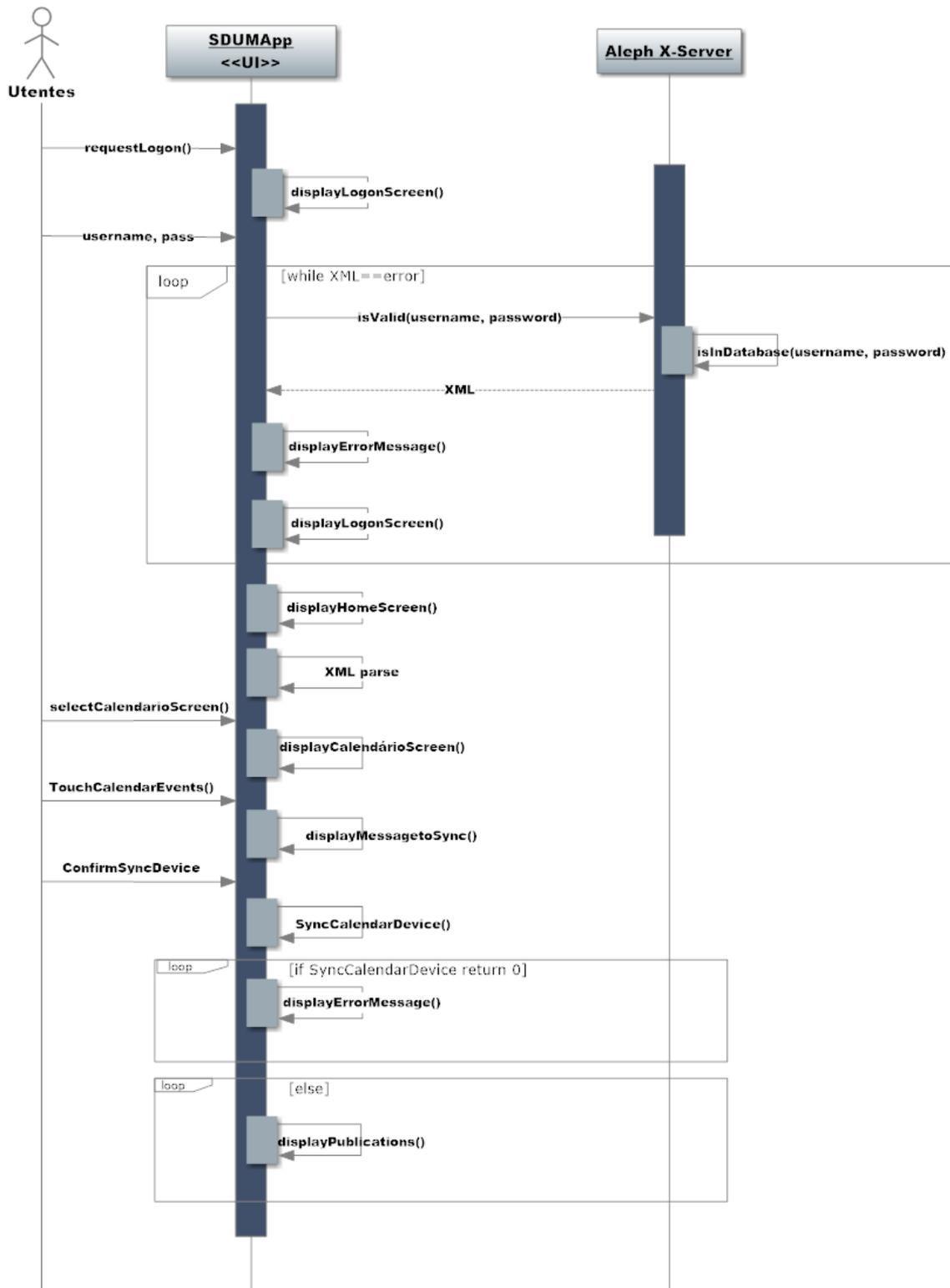


Figura B.28 - Diagrama de Sequencia (Calendário de Eventos)

8.3.5. Activity Diagram

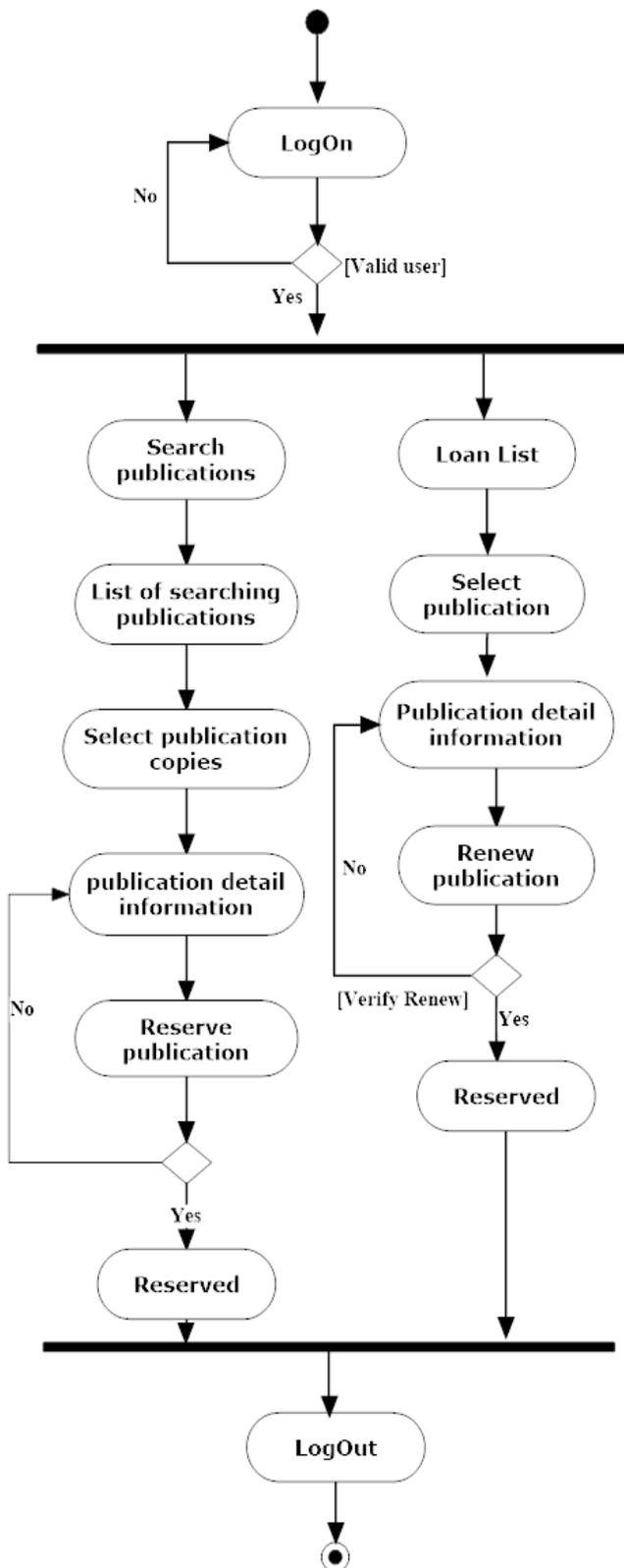


Figura B.29 - Activity Diagram (Renovar e Reservar Publicações)

9. Functional Requirements

Tabela B.33 – Requisito Funcional 1

Requirement #: 1	Requirement Type: Functional	Event/BUC/PUC #: UC1
Description: Autenticação de utentes		
Rationale: Os utentes efetuam a autenticação utilizando o seu <i>username</i> e <i>password</i> , de modo a poderem aceder às funcionalidades da aplicação.		
Originator: SDUM – Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: O processo de autenticação não deve ultrapassar 3s		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC1), Diagrama de Sequencia (Autenticação)		
History: Criado em 12 de Novembro de 2014		

Tabela B.34 - Requisito Funcional 2

Requirement #: 2	Requirement Type: Functional	Event/BUC/PUC #: UC2
Description: Visualização de dados pessoais		
Rationale: Utenes poderão visualizar os seus dados pessoais.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: É possível visualizar os dados pessoais do utente.		
Customer Satisfaction: 3		Customer Dissatisfaction: 1
Priority: Low	Dependencies: Autenticação	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC2), Diagrama de Sequencia (Dados Pessoais)		
History: Criado em 12 de Novembro de 2014		

Tabela B.35 - Requisito Funcional 3

Requirement #: 3	Requirement Type: Functional	Event/BUC/PUC #: UC3
Description: Visualização da Lista de Empréstimos		
Rationale: Os utentes poderão visualizar a lista das publicações requisitadas, ou seja, uma lista no qual consta todas as publicações em sua posse, requisitadas nas bibliotecas da Universidade do Minho. A partir desta lista para além da possibilidade de verificar os detalhes destas publicações requisitadas, é possível também renovar estas mesmas publicações.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: É possível visualizar a lista de publicações em posse		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: Autenticação	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC3), Diagrama de Sequencia (Lista de Empréstimos)		
History: Criado em 12 de Novembro de 2014		

Tabela B.36 - Requisito Funcional 4

Requirement #: 4	Requirement Type: Functional	Event/BUC/PUC #: UC4
Description: Visualização da Lista de Reservas		
Rationale: Os utentes poderão visualizar a lista das publicações reservadas. Nesta lista consta todas as publicações reservadas pelo utente, sendo possível também verificar os detalhes das mesmas.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: É possível visualizar a lista de publicações reservadas.		
Customer Satisfaction: 5		Customer Dissatisfaction: 4
Priority: Low	Dependencies: Autenticação, Reserva de Publicações	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC4), Diagrama de Sequencia (Lista de Reservas)		
History: Criado em 12 de Novembro de 2014		

Tabela B.37 - Requisito Funcional 5

Requirement #: 5	Requirement Type: Functional	Event/BUC/PUC #: UC5
Description: Renovação de Publicações		
Rationale: Os utentes poderão renovar as publicações em sua posse. A renovação só será efetuada, caso a publicação não esteja reservada por outros utentes ou caso o utente não tenha nenhuma situação irregular com os SDUM (multa por entrega de publicações fora da data prevista para entrega, bloqueados devido ao mau comportamento nas bibliotecas da Universidade do Minho ou outros casos de maior gravidade);		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: A aplicação deve permitir a renovação de publicações aos utentes.		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: Autenticação, Lista de Empréstimos	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC5), Activity Diagram (Renovar e Reservar Publicações), Diagrama de Sequencia (Renovação de Publicações)		
History: Criado em 12 de Novembro de 2014		

Tabela B.38 - Requisito Funcional 6

Requirement #: 6	Requirement Type: Functional	Event/BUC/PUC #: UC6
Description: Reserva de Publicações		
Rationale: Os utentes poderão efetuar a reserva de publicações. A reserva só poderá ser efetuada mediante à pesquisa nos catálogos das bibliotecas da Universidade do Minho. O processo de reserva só será concluído caso o utente não tiver nenhuma situação irregular com os SDUM ou caso a publicação não encontrar-se nas estantes (a reserva só é efetuada caso a publicação encontrar-se na posse de um outro utente).		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: Após a pesquisa de publicações, é possível efetuar o pedido de reserva da mesma.		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: Autenticação, Pesquisa de Publicações	Conflicts:

Supporting Material: Digrama de Casos de Uso (UC6), Activity Diagram (Renovar e Reservar Publicações), Diagrama de Sequencia (Reserva de Publicações)
History: Criado em 12 de Novembro de 2014

Tabela B.39 - Requisito Funcional 7

Requirement #: 7	Requirement Type: Functional	Event/BUC/PUC #: UC7
Description: Pesquisa de Publicações		
Rationale: Os utentes poderão efetuar pesquisas de publicações nos catálogos das bibliotecas da Universidade do Minho.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: Autenticação	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC7), Diagrama de Sequencia (Pesquisas nos Catálogos)		
History: Criado em 12 de Novembro de 2014		

Tabela B.40 - Requisito Funcional 8

Requirement #: 8	Requirement Type: Functional	Event/BUC/PUC #: UC8
Description: Visualização do Calendário de Eventos		
Rationale: Os utentes poderão visualizar o calendário de eventos com as publicações e as respetivas datas de entrega. A partir deste calendário é possível sincronizar estes eventos com o calendário do dispositivo móvel, que irá possibilitar a obtenção de notificações.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: É possível a partir do calendário de eventos sincronizar o mesmo com o calendário do dispositivo móvel.		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: Autenticação, Lista de Empréstimos	Conflicts: N/A

Supporting Material: Digrama de Casos de Uso (UC8), Diagrama de Sequencia (Calendário de Eventos).
History: Atualizado em 08 de Janeiro de 2015

Tabela B.41 - Requisito Funcional 9

Requirement #: 9	Requirement Type: Funcional	Event/BUC/PUC #: UC9
Description: Visualização de horário de funcionamento das bibliotecas da Universidade do Minho e a sua localização		
Rationale: Os utentes poderão visualizar as localizações das bibliotecas da Universidade do Minho, bem como o seu horário de funcionamento.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 4
Priority: Medium	Dependencies: N/A	Conflicts: N/A
Supporting Material: Digrama de Casos de Uso (UC9)		
History: Criado em 12 de Novembro de 2014		

10. Non-functional Requirements

10.1. Look and Feel Requirements

10.1.1. Appearance Requirements

Tabela B.42 - Requisito não-funcional 1

Requirement #: 10	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Aparência do produto		
Rationale: O produto deve ter uma interface atrativa para todos os utentes das bibliotecas da Universidade do Minho.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A

Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM
History: Criado em 12 de Fevereiro de 2015

10.2. Usability and Humanity Requirements

10.2.1. Ease of Use Requirements

Tabela B.43 - Requisito não-funcional 2

Requirement #: 12	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Facilidade de utilização do produto		
Rationale: Facilidade na utilização do sistema, ou seja, o sistema deve ser simples e intuitivo, permitindo aos utilizadores realizarem as tarefas com pouco tempo de treinamento.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.44 - Requisito não-funcional 3

Requirement #: 13	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Eficiência de utilização do produto		
Rationale:		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.45 - Requisito não-funcional 4

Requirement #: 14	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Informação de utilização e validação de dados		
Rationale: Na interface do produto deverá conter instruções e ajudas ao utilizador de modo a evitar que estes cometam erros		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.46 - Requisito não-funcional 5

Requirement #: 13	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Satisfação geral na utilização do produto		
Rationale: Os utilizadores deverão ficar satisfeitos com o produto, visto que produto será desenvolvido para o interesse de todos os utentes dos SDUM, ou seja, o produto deverá ser útil. O produto deverá envolver características que possam atrair os potenciais utilizadores do mesmo, e deverá incluir desde a adequação das informações aos utilizadores até os requisitos visuais utilizados na sua interface gráfica.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.47 - Requisito não-funcional 6

Requirement #: 13	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: <i>feedback</i> do produto		
Rationale: A interface deve reagir às ações do utilizador. Por exemplo, ao pressionar um botão, espera-se que uma certa funcionalidade seja ativada e o tempo de resposta apropriado e consistente com a funcionalidade envolvida;		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.2.2. Personalization and Internationalization Requirements

Tabela B.48 - Requisito não-funcional 7

Requirement #: 15	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Configuração do idioma		
Rationale: O sistema deverá permitir a configuração do idioma de acordo com a preferência do utilizador. Deve ter disponíveis pelo menos dois idiomas: Português e Inglês . A universidade do Minho recebe anualmente um número considerável de estudantes em mobilidade (Erasmus), daí a necessidade do Inglês para além do Português.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: O idioma pré-definido para o produto deverá ser o Português.		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM, Portal SDUM (www.sdum.uminho.pt)		
History: Criado em 12 de Novembro de 2014		

10.2.3. Learning Requirements

Tabela B.49 - Requisito não-funcional 8

Requirement #: 16	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Fácil aprendizagem		
Rationale: O sistema deve ser de fácil aprendizagem, ou seja, os utilizadores devem ser capazes de utilizar o produto sem nenhum treinamento prévio. Deve ser intuitivo.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 4		Customer Dissatisfaction: 4
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.2.4. Understandability and Politeness Requirements

N/A

10.2.5. Accessibility Requirements

Tabela B.50 - Requisito não-funcional 9

Requirement #: 18	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Acessibilidade do produto		
Rationale: O produto deverá também ser utilizado por utilizadores com difidências físicas e auditivas.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 3
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.3. Performance Requirements

10.3.1. Speed and Latency Requirements

Tabela B.51 - Requisito não-funcional 10

Requirement #: 19	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Respostas imediatas		
Rationale: O sistema deve proporcionar respostas imediatas aos cliques dos utilizadores.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.52 - Requisito não-funcional 11

Requirement #: 20	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Velocidade das respostas aos pedidos		
Rationale: A apresentação dos dados das pesquisas não devem demorar mais do que que 10 segundos para 90% das solicitações;		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

Tabela B.53 - Requisito não-funcional 12

Requirement #: 21	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Latência das respostas		
Rationale: A resposta deve ser rápida o suficiente para evitar interrupção no fluxo de pensamento do utilizador		

Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.3.2. Reliability and Availability Requirements

Tabela B.54 - Requisito não-funcional 13

Requirement #: 22	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Disponibilidade do produto		
Rationale: O produto deverá esta disponível 24 horas por dia, durante 365 dias por ano		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.3.3. Rebusness or Fault-Tolerance Requirements

Tabela B.55 - Requisito não-funcional 14

Requirement #: 23	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Robustez do produto		
Rationale: Caso houver perda de conexão com a internet, a aplicação deverá indicar um alerta ao utilizador. A partir desse momento não será possível efetuar pedidos até que a conexão ser retomada.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		

Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 3
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.3.4. Requirements for Interfacing with Adjacent Systems

Tabela B.56 - Requisito não-funcional 15

Requirement #: 24	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Interface com o Aleph X-Server		
Rationale: O Servidor Aleph X-Server é o componente fundamental para o funcionamento do produto. É o servidor onde são armazenadas todas as informações administrativas, bibliográficas, relativas aos leitores e empréstimos interbibliotecas dos SDUM.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.3.5. Productization Requirements

Tabela B.57 - Requisito não-funcional 16

Requirement #: 25	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Instalação do produto		
Rationale: O produto deverá ser de fácil instalação por utilizadores inexperientes		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		

Customer Satisfaction: 5		Customer Dissatisfaction: 1	
Priority: High	Dependencies: N/A	Conflicts: N/A	
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM			
History: Criado em 12 de Novembro de 2014			

Tabela B.58 - Requisito não-funcional 17

Requirement #: 26	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: <i>Download</i> do produto		
Rationale: O produto deverá ser grátis aos utentes, isto é, qualquer utilizador poderá baixar o produto de forma grátis.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 1
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.4. Maintainability and Support Requirements

10.4.1. Maintenance Requirements

Tabela B.59 - Requisito não-funcional 18

Requirement #: 27	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Manutenção do produto		
Rationale: O produto deverá ser submetido a manutenção sempre que for necessário. Deverá ser submetido a manutenção das informações dos horários de funcionamento das bibliotecas da Universidade do Minho, bem como a sua localização. Esta manutenção será efetuada por		

um dos elementos equipa de informática dos SDUM, mais especificamente pelo Nuno Fernandes.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.4.2. Adaptability Requirements

Tabela B.60 - Requisito não-funcional 19

Requirement #: 28	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Sistema operativo móvel de funcionamento		
Rationale: O produto deverá ser compatível com todos os dispositivos móveis que possuam o sistema operativo Android.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.5. Security Requirements

10.5.1. Access Requirements

Tabela B.61 - Requisito não-funcional 20

Requirement #: 29	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Autenticação		
Rationale: O acesso ao sistema é efetuado a partir de autenticação. O utilizador introduz os seus credenciais (<i>username</i> e <i>password</i>) e caso seja utente dos SDUM terá acesso às funcionalidades do sistema.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.5.2. Integrity Requirements

Tabela B.62 - Requisito não-funcional 21

Requirement #: 30	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Integridade		
Rationale: O produto deverá garantir a integridade dos dados, ou seja, garantir que os dados dos utilizadores estão protegidos de acessos não autorizados.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

10.5.3. Privacy Requirements

Tabela B.63 - Requisito não-funcional 22

Requirement #: 31	Requirement Type: Non-functional	Event/BUC/PUC #:
Description: Acesso aos dados dos utilizadores		
Rationale: Os dados de cada utilizador são privados. Só os mesmos é que podem ter acesso aos seus dados pessoais.		
Originator: SDUM - Chefe divisão de Informação Dr ^a Daniela Castro Ramalho		
Fit Criterion: N/A		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Priority: High	Dependencies: N/A	Conflicts: N/A
Supporting Material: <i>Workshops</i> com a chefe de divisão de informação dos SDUM		
History: Criado em 12 de Novembro de 2014		

11. Risks

A Tabela a seguir apresentada tem como objetivo especificar um conjunto de riscos de desenvolvimento da solução final. Para a definição da probabilidade de ocorrência e do impacto foi especificada a seguinte nomenclatura:

ME – Muito Elevada; E - Elevada; M - Moderada; B – Baixa; MB – Muito Baixa.

Tabela B.64 – Lista de Riscos associados ao desenvolvimento da solução final

Nº	Descrição do Risco	Probabilidade de ocorrência	Impacto	Plano de mitigação
1	Má interpretação dos requisitos	ME	ME	Execução do modelo de processo de engenharia de requisitos.
2	Mau conhecimento do domínio do problema	ME	E	Estudo detalhado do conhecimento do domínio do problema.

3	Requisitos mal especificados	ME	ME	Execução do modelo de processo de engenharia de requisitos.
4	Falta do domínio tecnológico (framework de desenvolvimento multiplataforma)	ME	ME	Estudo intensivo da tecnologia antes de começar a implementar a aplicação.
5	Incumprimento de datas e prazo de entrega	ME	M	Execução dos métodos de engenharia de requisitos e realização de um plano de trabalho.
6	Lógica de negócio mal estruturadas	ME	ME	Estudo do domínio do problema.
7	Dificuldade na integração da aplicação com o servidor e no processamento de dados provenientes do mesmo	E	ME	Reuniões com a equipa de informáticos dos SDUM.
8	Mudanças e alterações contínuas dos requisitos	ME	ME	Execução do modelo de processo de engenharia de requisitos.
9	Má definição da arquitetura da solução	ME	E	Execução do modelo de processo de engenharia de requisitos.
11	Baixa produtividade	M	M	Aumentar o nível de produtividade
12	Dependência em relação à equipa de	B	B	Reuniões com a equipa de informáticos dos SDUM periodicamente.

	informáticos dos SDUM			
13	Conflitos de ideias/interesses com o cliente (SDUM)	E	M	Execução do modelo de processo de engenharia de requisitos.

A avaliação dos atributos de um risco tem como objetivo obter uma melhor compreensão do próprio risco, através da determinação da probabilidade, do impacto esperado e do momento em que se prevê que este venha a ocorrer. A conjugação destes dois atributos dá origem a denominada Matriz de Exposição ao Risco, instrumento muito utilizado na gestão dos riscos, pelo simplicidade e visão de conjunto que propicia. A Tabela mostra uma Matriz de Exposição ao Risco, que é utilizado pela Força Aérea dos EUA nos seus projetos de desenvolvimento de *Software*.

Tabela B.65 – Matriz de Exposição ao Risco

Impacto	Probabilidade		
	Elevada	Média	Baixa
Catastrófico	Elevada	Elevada	Moderada
Crítico	Elevada	Moderada	Moderada
Marginal	Moderada	Moderada	Baixa
Negligenciável	Moderada	Baixa	Baixa

Legenda	Descrição
	Necessária uma estratégia de mitigação ativa e um Plano de contingência detalhado.
	Necessária uma estratégia de mitigação ativa.
	Risco documentado e tratado como um pressuposto do projeto.

Anexo C – Testes de Usabilidade

Teste de Usabilidade da solução SDUM

Consiste em avaliar a usabilidade da aplicação de apoio a gestão de empréstimos e renovações nas bibliotecas da Universidade do Minho - SDUMApp, ou seja, pretende-se avaliar o quão fácil é a aplicação de ser compreendida e manipulada pelos utilizadores.

1. Autenticação na solução SDUM

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

2. Visualização dos dados das bibliotecas da Universidade do Minho

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

3. Pesquisa de publicações

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

4. Reserva de publicações

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

5. Visualização da Lista de empréstimos

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

6. Visualização da Lista de reservas

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

7. Visualização dos dados pessoais

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

8. Renovação de publicações

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

9. Visualização do calendário de eventos

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

10. Sincronização com o calendário de dispositivo móvel

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

11. Alteração de idioma

Classifique-a relativamente ao grau de dificuldade
Marcar apenas uma oval.

	0	1	2	3	4	
Muito fácil	<input type="radio"/>	Muito difícil				

Figura C.1 - Teste de usabilidade das tarefas

Avaliação da usabilidade

consiste num questionário composto por 10 afirmações com cinco opções de resposta, que os participantes avaliam o seu nível de concordância com a aplicação numa escala de 0-4.

1. Sinto que vou usar esta aplicação com frequência.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente				

2. Acho a aplicação desnecessariamente complexa.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Concordo Totalmente	<input type="radio"/>	Discordo Totalmente				

3. Achei a aplicação fácil de utilizar.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente				

4. Acho que vou precisar de ajuda para poder utilizar esta aplicação.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Concordo Totalmente	<input type="radio"/>	Discordo Totalmente				

5. As funcionalidades da aplicação estão bem integradas.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente				

6. Achei a aplicação muito inconsistente.

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Concordo Totalmente	<input type="radio"/>	Discordo Totalmente				

7. **Penso que a maioria das pessoas aprenderiam a utilizar a aplicação rapidamente.**

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente				

8. **Achei a aplicação muito complicada de utilizar.**

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Concordo Totalmente	<input type="radio"/>	Discordo Totalmente				

9. **Senti-me confiante a utilizar a aplicação.**

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente				

10. **Preciso aprender muitas coisas para utilizar bem esta aplicação.**

Avalie esta afirmação
Marcar apenas uma oval.

	0	1	2	3	4	
Concordo Totalmente	<input type="radio"/>	Discordo Totalmente				

Figura C.2 - Avaliação da usabilidade

Anexo D – Lista de Serviços disponibilizados pelo Aleph X-Server

Tabela D.1 - Lista dos serviços do Aleph [64]

Serviços oferecidos pelo Aleph	
Serviços	Descrição
HoldReqCancel	Este serviço elimina o hold request record (Z37) dado uma determina chave designado hold request key (docnumber + item sequence + cancel sequence)
CircStatus	Este serviço devolve o estado de circulação (circulation status) de cada document number introduzido pelo utilizador. <ul style="list-style-type: none"> ✓ Informação do exemplar (Z30); ✓ Informação de empréstimos (Z36); ✓ Estado de empréstimo (Tab15), data entrega, hora entrega;
AdminService [*]	Explicação geral do X-Service
SortSet	Este serviço reordena um conjunto específico de documentos de acordo com os parâmetros fornecido pelo utilizador;
BorAuth	Este serviço devolve o registo global ou dados de processamento (Z303), os dados de inscrição (Z305) e os dados de correspondência (Z304) de uma determinado utente, caso o ID e a código de verificação coincidirem.
GetHolding	Este serviço devolve holdings (HOL) information para um determinado document number do catálogo, fornecido pelo utilizador.
ItemData	O serviço devolve o document number do utilizador. Para cada um dos document's items ele devolve: <ul style="list-style-type: none"> ✓ Dados do exemplar (Z30); ✓ Dados do empréstimo (Z36); ✓ Uma indicação se o pedido está em espera;
FindDoc	Devolve o formato XML OAI de um expanded document determinado pelo utilizador.
UpdateBor	Este serviço realiza diversas ações (Atualização/Inserção/Delete) envolvendo a atualização dos dados do utente. Estas ações afetam as tabelas Z303, Z304, Z305, Z308 com base na entrada XML especificando

	que a ação é tomada em cada tabela. O XML inclui todos os dados das tabelas e é apenas para um único utente.
Find	Este serviço devolve o set number e o number of records que respondem a uma solicitação de pesquisa inserido pelo utilizador.
Renew	Este serviço renova o empréstimo de um determinado exemplar. O X-Service renova o empréstimo somente se ele pode ser feito. Por exemplo, uma solicitação de renovação de um exemplar cuja data de entrega já se encontra ultrapassada, o serviço não procede a renovação do empréstimo.
HoldReq	Este serviço cria um hold-request record (Z37) para um determinado exemplar após a realização das verificações iniciais.
BorByKey	Este serviço devolve o ID interno (do Z308) para um determinado Borrower_id , caso o ID fornecido e a verificação do código coincidirem. Caso não coincidirem, é retornado uma mensagem de erro.
UpdateDoc	O serviço realiza ações (Atualizar/Inserir/Excluir) relacionados com a atualização do documento. (O serviço utiliza o pc_cat_c0203 que atualiza o documento através do GUI).
BorCourse	Este serviço devolve todos os course records (Z107) pertencentes a um dado Borrower_Id caso o ID fornecido e a verificação do código coincidirem. Caso não se confirme, é retornado uma mensagem de erro.
Demo [*]	Explicação geral do X-Service
Version [*]	Explicação geral do X-Service
BorInfo	Este serviço devolve todas as informações relacionadas a um determina utente: dados de processamento do utente, dados de correspondência do utente, dados de inscrição do utente, dados de empréstimos, dados do exemplar requisitado, dados da obra, multa corrente do empréstimo, data de devolução do empréstimo, valor total de multas em dívida, etc..., caso o ID e o código de verificação fornecidos coincidirem. Caso contrário, é retornado uma mensagem de erro.
GetSortCodes	Este serviço formata o tab_sort table residente numa determinada biblioteca numa resposta em XML.

Present	O serviço devolve o formato XML OAI do expanded documents . Os documents podem ser visualizadas de acordo com os locais dentro de um determinado set number específico.
----------------	--