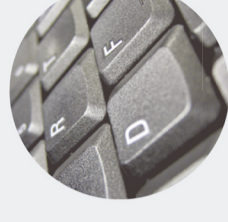
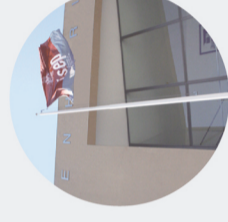




Desenvolvimento de aplicação móvel multiplataforma (Apache Cordova vs Xamarin)

DIOGO MIGUEL RODRIGUES E SILVA

Julho de 2018



Desenvolvimento de aplicação móvel multiplataforma (Apache Cordova vs Xamarin)

DIOGO MIGUEL RODRIGUES E SILVA
Julho de 2018



Desenvolvimento de aplicação móvel multiplataforma

Apache Cordova vs Xamarin

Diogo Miguel Rodrigues e Silva

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais**

Orientador: Elsa Maria De Carvalho Ferreira Gomes

Dedico este projeto aos meus pais.

Resumo

As tecnologias multiplataforma continuam a ser uma alternativa viável às tecnologias nativas, nomeadamente em organizações e equipas de desenvolvimento limitadas.

Partindo de uma plataforma já existente, o Portal do ISEP, o principal objetivo deste trabalho consistiu no desenvolvimento de uma aplicação móvel para toda a comunidade académica do ISEP ter acesso à informação desta plataforma de uma forma mais fácil e rápida.

Dado que a aplicação existente não satisfazia as necessidades dos alunos, funcionários e professores da instituição, e a sua manutenção era incomportável para a equipa de desenvolvimento existente, tornou-se necessária a adoção de uma tecnologia multiplataforma.

O trabalho realizado centrou-se no desenvolvimento de dois protótipos, em Apache Cordova e Xamarin, com os módulos Login, Notícias, Informação do utilizador e Logout implementados.

Após a realização de testes de usabilidade foi possível concluir que uma aplicação móvel do Portal funcional tornar-se-á numa mais-valia para a comunidade académica e para a instituição.

Palavras-chave: Tecnologias multiplataforma, Apache Cordova, Xamarin

Abstract

Cross-Platform Technologies remains a viable alternative to native technologies, mainly when we are dealing with organizations and development teams with limitations.

From an existent platform, Portal ISEP, the main goal of this project was to develop a mobile application to the whole ISEP academic community, offering a faster and easiest way to get necessary information from Portal.

Since the existent application didn't satisfy the necessities of the students, employees and teachers of the institution, and its maintenance was unaffordable for the existent development team, it became necessary to adopt a multiplatform technology.

The work done focused on the development of two prototypes, in Apache Cordova and Xamarin, with the modules Login, News, User information and Logout implemented.

After performing usability tests, it was possible to conclude that a mobile application of functional Portal will become an added-value for the academic community and to the institution.

Keywords: Cross-Platform Technologies, Apache Cordova, Xamarin

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao Eng.º José Carlos Barros Oliveira por me ter dado a oportunidade de desenvolver o projeto no DSI assim como ter disponibilizado recursos que possibilitaram a concretização do mesmo.

À Professora Elsa Gomes, pela sua orientação, disponibilidade, profissionalismo que lhe é característico, opiniões e sugestões, que foram o mote para que este projeto atingisse todos os objetivos.

Agradeço ao Eng.º Diogo Oliveira pelo acompanhamento dado, pela transmissão de conhecimentos e pela ajuda prestada sempre que necessário.

Não posso deixar de agradecer à equipa do DSI, que sempre me proporcionou um clima de bem-estar e interajuda.

Concomitantemente, um muito obrigado a todos os alunos, funcionários e professores por partilharem as suas experiências e mostrarem-se disponíveis a dar o seu contributo a este projeto.

Um especial agradecimento a toda a equipa do Gabinete de Relações Externas, Alexandra Trincão, Goreti e Paula, pelo incentivo, por estarem ligadas às minhas melhores experiências académicas e por me receberem sempre com amabilidade, sorrisos e boa disposição.

É também devido um agradecimento aos meus amigos que me acompanharam ao longo deste percurso, nomeadamente ao Eng.º Bruno Barbosa, Eng.º André Ponte e Eng.º Daniel Freitas.

Por último, o maior agradecimento vai para os meus pais e irmã por todo o suporte e educação.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Problema.....	2
1.3	Objetivos.....	2
1.4	Resultados Esperados	3
1.5	Análise de Valor	3
2	Contexto e Estado da Arte	4
2.1	Contexto e problema	4
2.2	Estado da arte.....	5
2.2.1	Tecnologias Multiplataforma	5
2.2.2	Trabalho anterior relacionado	9
2.3	Análise de valor	10
2.3.1	Modelo de desenvolvimento de novo conceito.....	10
2.3.2	Valor, valor para o cliente e valor percebido	12
2.3.3	Modelo Canvas	13
2.3.4	Método Analytic Hierarchy Process para comparação das soluções	16
3	Design.....	25
3.1	Design de software	25
3.2	Casos de uso	27
3.3	Estrutura Lógica	29
3.4	Arquitetura	30
3.4.1	Xamarin	30
3.4.2	Apache Cordova	32
4	Implementação da solução	34
4.1	WebServices	34
4.2	Interface	36
4.3	Bindings	38
5	Avaliação	41
5.1	Avaliação de usabilidade com utilizadores	41
5.1.1	Metodologia	42
5.1.2	Perfil da amostra	43
5.1.3	Análise dos resultados	43

5.2	Avaliação ao desenvolvimento	45
5.3	Métricas de desenvolvimento	47
5.3.1	Análise dos pontos de função	48
5.3.2	Conclusão da análise	57
6	Conclusão	59
6.1	Objetivos realizados	59
6.2	Limitações e trabalho futuro	59
6.3	Apreciação Final	61
7	Referências	62

Lista de Figuras

Figura 1 - New concept development model.....	10
Figura 2 – <i>Business Model Canvas</i>	15
Figura 3 - Valores de IR para matrizes quadradas de ordem n	19
Figura 4 - Diagrama hierárquico com importâncias relativas associadas	23
Figura 5 – Módulo Login (Padrão MVVM).....	26
Figura 6 – Casos de uso	27
Figura 7 – Estrutura analítica do projeto.....	30
Figura 8 – Arquitetura conceptual do método Shared Project	31
Figura 9 - Arquitetura conceptual do método Portable Class Library.....	32
Figura 10 – Arquitetura conceptual do Apache Cordova	33
Figura 11 – Métodos login e logout do serviço Session (Xamarin)	35
Figura 12 – <i>Callback</i> login do Serviço Session (Apache Cordova)	35
Figura 13 – Envelope login (Apache Cordova).....	36
Figura 14 – Função que permite receber resposta do pedido login	36
Figura 15 – Interface módulo notícias em Android (Xamarin).....	37
Figura 16 – Interface módulo Notas Finais em Windows (Apache Cordova).....	38
Figura 17 – <i>BindingContext</i> da classe LoginView (Xamarin).....	38
Figura 18 – <i>ListView</i> Notícias (Xamarin)	39
Figura 19 – Conversão de Array para WinJS.Binding.List (Apache Cordova)	40
Figura 20 – <i>ListView</i> Notícias (Apache Cordova)	40
Figura 21 – Entidades implementadas	50

Lista de Tabelas

Tabela 1 - Escala fundamental de Satty	17
Tabela 2 - Matriz de Comparação dos critérios do Segundo Nível	17
Tabela 3 – Matriz de Comparação dos critérios do Segundo Nível com soma das colunas	18
Tabela 4 - Matriz de Comparação dos critérios do Segundo Nível – Passo intermédio	18
Tabela 5 – Matriz Normalizada dos Critérios do Segundo Nível.....	18
Tabela 6 – Vetor de Prioridades da Matriz Normalizada	18
Tabela 7 – Matriz de Comparação do critério Interface	20
Tabela 8 - Matriz de Comparação do critério Interface com soma das colunas.....	20
Tabela 9 - Matriz de Comparação do critério Interface – Passo intermédio	20
Tabela 10 - Matriz Normalizada do critério Interface.....	20
Tabela 11 – Matriz de Comparação do critério Experiência	21
Tabela 12 - Matriz de Comparação do critério Experiência com soma das colunas.....	21
Tabela 13 - Matriz de Comparação do critério Experiência – Passo intermédio.....	21
Tabela 14 - Matriz Normalizada do critério Experiência.....	21
Tabela 15 - Matriz de Comparação do critério Ferramentas.....	22
Tabela 16 - Matriz de Comparação do critério Ferramentas com soma das colunas.....	22
Tabela 17 - Matriz de Comparação do critério Ferramentas – Passo intermédio	22
Tabela 18 - Matriz Normalizada do critério Ferramentas.....	23
Tabela 19 – Propriedades Compostas do problema	24
Tabela 20 - Especificação do caso de uso 1	27
Tabela 21 - Especificação do caso de uso 2	28
Tabela 22 - Especificação do caso de uso 3	28
Tabela 23 – Especificação do caso de uso 4.....	28
Tabela 24 – Caracterização dos participantes da amostra (N=36)	43
Tabela 25 – Nível de complexidade AIE	50
Tabela 26 – Nível de complexidade ALI	50
Tabela 27 – Nível de complexidade CE	51
Tabela 28 – Nível de complexidade EE.....	51
Tabela 29 – Nível de complexidade SE.....	51
Tabela 30 – Arquivo Lógico Interno	51
Tabela 31 – Arquivo de Interface Externa.....	52
Tabela 32 – Entrada Externa	53
Tabela 33 – Consulta Externa.....	53
Tabela 34 – Análise Pontos de Função Brutos	54
Tabela 35 – Características gerais do sistema	56

Acrónimos e Símbolos

Lista de Acrónimos

AHP	<i>Analytic Hierarchy Process</i>
DP	Desvio Padrão
DSI	Divisão de Sistemas Informáticos
HTTP	HyperText Transfer Protocol
ISEP	Instituto Superior de Engenharia do Porto
M	Média
NCD	<i>New Concept Development</i>
NPD	<i>New product development</i>
SOAP	<i>Simple Object Access Protocol</i>
TSG	<i>Technology stage gate</i>
UWP	<i>Universal Windows Plataform</i>
VSM	<i>Analytic Hierarchy Process</i>
XML	<i>eXtensible Markup Language</i>
LOC	<i>Lines of code</i>
EE	Entrada Externa
SE	Saída Externa
ALI	Arquivo Lógico Interno
AIE	Arquivo de Interface Externa
CE	Consulta Externa

1 Introdução

O capítulo da Introdução pretende apresentar sucintamente vários tópicos que são descritos ao longo do documento.

Inicia-se com uma introdução (secção 1.1), onde se procede à contextualização do tema, é apresentado um resumo do problema encontrado (secção 1.2) e de seguida é apresentado um resumo dos objetivos pretendidos com este trabalho (secção 1.3). Segue-se a apresentação dos resultados esperados (secção 1.4) e termina com um resumo da análise de valor (secção 1.5).

1.1 Enquadramento

Sendo o Instituto Superior de Engenharia do Porto (ISEP) uma instituição de referência voltada para a promoção da criação e partilha de conhecimento aplicado e orientado para o mundo real, e frequentado por mais de 6500 estudantes e 500 colaboradores [1], torna-se fundamental e pertinente desenvolver tecnologias para dar resposta às necessidades da comunidade académica.

Neste sentido, o ISEP desenvolve diversos produtos tendo como público alvo os alunos, professores e funcionários da instituição. Entre estes produtos encontra-se o Portal da instituição que possibilita a este público alvo a gestão da sua vida académica e profissional [2].

Inserida nesta plataforma, desenvolveram-se três aplicações móveis do Portal para as três principais plataformas móveis nativas, de forma que a consulta da informação seja facilitada. No entanto, todas as aplicações apresentam alguns problemas de funcionamento e percebeu-se que a manutenção e desenvolvimento das aplicações é insustentável para a equipa de desenvolvimento existente.

Tendo em conta as circunstâncias, definiu-se que seria necessário adotar uma tecnologia multiplataforma. Deve-se então proceder a um estudo aprofundado relativo a estas tecnologias de modo que a escolha de uma tecnologia multiplataforma seja facilitada e que o ISEP

possibilite à comunidade académica ter uma aplicação mais estável e disponível para todo o público alvo.

1.2 Problema

Focada na proximidade e na interação da instituição ISEP com os seus alunos, surge a necessidade de criar uma aplicação móvel com informações válidas relativas à vida académica destes.

Apesar da existência de aplicações móveis nativas (Universal Windows Platform (UWP), Android e IOS), a instituição não conseguiu ainda dar respostas definitivas e eficazes às necessidades totais dos seus alunos. Isto é, nem todas as aplicações estão funcionais e nem é viável manter/melhorar todas as aplicações para a equipa de desenvolvimento existente.

Como tal, surge a necessidade de adotar ferramentas de desenvolvimento multiplataforma. Entre estas ferramentas encontram-se o Apache Cordova e o Xamarin.

Qual destas tecnologias será capaz de ser a melhor alternativa às versões nativas tendo em conta o contexto e limitações da organização? Após testarem as duas tecnologias os utilizadores manifestam preferência por alguma delas?

1.3 Objetivos

O principal objetivo deste projeto é o desenvolvimento de uma aplicação móvel para que a comunidade do ISEP possa consultar informação relativa ao Portal, por exemplo: horário, notas parciais, notas finais, salas disponíveis, parque automóvel, entre outros.

Será então necessário avaliar qual a melhor tecnologia a adotar para o desenvolvimento da aplicação móvel tendo em conta as necessidades da organização.

Para tal, serão desenvolvidos dois protótipos de um ou mais módulos da aplicação nativa implementados em UWP, em Apache Cordova e Xamarin, para que seja possível comparar as duas implementações feitas, incluindo dificuldades obtidas, limitações de ambas, vantagens e desvantagens, comparação do tempo de implementação e curva de aprendizagem.

Também serão realizados testes de usabilidade com utilizadores, afim de se perceber qual das tecnologias é a preferida por este grupo.

Espera-se que esta dissertação seja um documento que auxilie equipas de desenvolvimento na tomada de decisões em relação à adoção de uma tecnologia multiplataforma.

1.4 Resultados Esperados

Uma vez que o Apache Cordova privilegia a integração de *frameworks* para a interface e estrutura de código, espera-se que em termos de tempo de desenvolvimento o Apache Cordova se superiorize ao Xamarin. No entanto, uma vez que o Xamarin é uma tecnologia Microsoft, será interessante perceber que há diversas vantagens e mais facilidade na implementação de várias funcionalidades, como por exemplo, utilização de *webservices*.

Espera-se, também, que a grande diferença entre as duas tecnologias em termos de complexidade e tempo de desenvolvimento esteja na implementação das interfaces para o utilizador.

Para além desse facto, avaliar o impacto da experiência da equipa de desenvolvimento nas diferentes linguagens de programação também será interessante de perceber.

1.5 Análise de Valor

Através do modelo New Concept Development (NCD), foi possível identificar que este projeto surge de uma oportunidade. Oportunidade essa que surge após o reconhecimento de vários problemas funcionais nas aplicações móveis do ISEP que se traduziam em descontentamento da comunidade académica.

Após a identificação da oportunidade, procedeu-se então à análise de cada elemento do modelo NCD até se definir o conceito do mesmo.

Em termos de valor para a comunidade académica do ISEP, a curto e a longo prazo, prevê-se que possam beneficiar de uma aplicação que lhe permite consultar informação do Portal de uma forma mais acessível e rápida, não se prevendo qualquer sacrifício para este público alvo.

Para a instituição ISEP, para além da valorização da instituição, prevê-se que tenha uma aplicação funcional para todas as plataformas e que fique com o *know-how* do projeto desenvolvido. A longo prazo, o ISEP poderá beneficiar da venda da aplicação a outras instituições de ensino. Contudo, é inerente a despesa de alguns recursos, nomeadamente, os custos da equipa de desenvolvimento e o tempo despendido no desenvolvimento deste novo projeto.

Por fim, elaborou-se um modelo Canvas do projeto e através do método AHP foi possível escolher qual das melhores alternativas tecnológicas seriam utilizadas no projeto.

2 Contexto e Estado da Arte

No desenvolvimento de um projeto, torna-se imprescindível refletir sobre as várias condicionantes que possam afetar a viabilidade do nosso projeto. Quer a nível das necessidades da organização, quer a nível daquilo que já existe no mercado.

Este capítulo começa uma análise ao contexto e problema (2.1), onde se procede à contextualização e se explica a razão que levou ao desenvolvimento deste tema, segue-se uma apresentação da análise de valor (2.3), e por fim apresenta-se o estado da arte (2.2), no qual é demonstrado o estudo realizado sobre tecnologias e soluções existentes.

2.1 Contexto e problema

Localizado na cidade do Porto e com mais de 160 anos de existência, o Instituto Superior de Engenharia do Porto é uma instituição de sucesso no ensino e inovação em engenharia. O ISEP oferece catorze licenciaturas e doze mestrados, na área da engenharia. Entre estes cursos, destacam-se, por exemplo, as licenciaturas em engenharia informática e engenharia mecânica, cursos com o selo de qualidade EUR-ACE¹ [3].

Para além dos vários cursos e formações que oferece, o ISEP dispõe de dez grupos de investigação e cinco centros de prestação de serviços. Entre estes centros de prestação de serviços, a Divisão de Sistemas Informáticos (DSI) é responsável pelo desenvolvimento de vários projetos como o site e o portal da instituição e a aplicação do portal [1].

A aplicação do portal é uma aplicação nativa implementada para as três plataformas: Windows, Android e iOS. Como tal, existem três projetos distintos em linguagens de programação que são características destas plataformas. A aplicação do portal é desenvolvida por uma equipa constituída por dois elementos: um supervisor e um programador.

¹ Selo de qualidade atribuído em Portugal pela Ordem dos Engenheiros

Apesar da existência destas aplicações móveis nativas, a instituição ainda não conseguiu dar respostas definitivas e eficazes às necessidades totais da comunidade académica. Isto é, nem todas as aplicações estão funcionais e nem é viável manter/melhorar ambas as aplicações com a equipa de desenvolvimento existente. Surge então a necessidade de adotar ferramentas de desenvolvimento multiplataforma. Entre estas ferramentas encontram-se o Apache Cordova e o Xamarin.

É neste contexto que surge o problema desta investigação. Qual destas tecnologias será capaz de ser a melhor alternativa às versões nativas tendo em conta o contexto e limitações da organização; de modo a garantir o desenvolvimento de um produto que maximize o bem-estar dos utilizadores?

2.2 Estado da arte

O desenvolvimento de projetos requer uma análise ao estado da arte. Neste sentido, foi necessário fazer um levantamento de tecnologias já existentes e de que forma é que essas tecnologias dão resposta a este público alvo. Este estudo exploratório permitiu não só o conhecimento mais aprofundado daquilo que existe no mercado, mas também o surgimento de novas ideias. Todos estes aspetos permitiram orientar o aluno na melhor direção para o desenvolvimento do seu próprio projeto. Nas páginas que se seguem, o aluno apresenta de forma resumida, o resultado dessa pesquisa.

2.2.1 Tecnologias Multiplataforma

O Mercado das aplicações móveis está a crescer rapidamente. A introdução de novas tecnologias, plataformas e linguagens de programação proporcionaram um grande impulso na área do desenvolvimento *mobile*. Com o aumento da popularidade das aplicações móveis, estas acabam por se tornar cada vez mais importantes para as organizações e para as suas necessidades do negócio [4].

Embora seja necessário ter alguma cautela no desenvolvimento em ambientes multiplataforma, existem várias razões que levam as empresas a adotar este tipo de tecnologia [5]:

- Uma equipa de desenvolvimento pode desenvolver soluções para a organização independentemente do tipo de dispositivo móvel.
- É mais fácil aprender com usar uma ferramenta de desenvolvimento do que várias.
- A maioria das ferramentas de desenvolvimento multiplataforma têm vindo a atingir um ponto de maturação bastante rápido. A maioria das funcionalidades que a organização necessita já se encontram incluídas no *core* destas ferramentas.
- A possibilidade de criar uma aplicação que pode ser distribuída muito rapidamente em todas as plataformas. Este aspeto é muito importante para pequenas e médias empresas que não tenham capacidade financeira para suportar uma equipa de

programadores *mobile* para Android, iOS e Windows. Estas tecnologias têm um custo de produção bastante baixo.

No entanto, quando se depende de uma plataforma de software para a criação de uma aplicação multiplataforma é necessário ter em consideração alguns aspetos. Em primeiro lugar, devido ao facto das plataformas Android, iOS e Windows terem *designs* e interfaces distintas, a aplicação multiplataforma terá que sacrificar alguma usabilidade. Para além disso, existirá sempre uma dependência em relação às plataformas nativas, dependência essa que surge aquando de *updates* aos SDKs nativos, *bug fixes*, e de novas versões dos sistemas operativo e *frameworks*. Por fim, poderá existir uma maior complexidade na implementação de uma funcionalidade específica.

Matthew David [6] estabeleceu algumas diretrizes em relação à escolha de uma tecnologia multiplataforma dependendo da empresa:

- Pequenas empresas (constituídas por menos de 50 pessoas)
Estas empresas devem focar-se em ferramentas *mobile* que possam ser lançadas para o mercado rapidamente, sem que seja necessário suportar o custo de uma equipa de desenvolvimento. Por norma, este tipo de organizações adota aplicações já disponíveis nas *app stores* públicas ou em produtos comercializados por terceiros.
- Pequenas e médias empresas (formadas até 500 trabalhadores)
Estas organizações deparam-se com o desafio de desenvolver aplicações para as diversas plataformas devido à variedade de dispositivos do staff e do consumidor final. Neste caso, recomenda-se que se adotem as tecnologias multiplataforma como por exemplo, o Xamarin.
- Grandes empresas
As necessidades destas organizações são obviamente mais complexas que as anteriormente referidas. Normalmente, apresentam sistemas de *back office* mais complexos e necessitam de segurança mais sofisticada. Desta forma, recomenda-se uma abordagem por camadas na qual a aplicação é desenvolvida numa ferramenta multiplataforma, mas as funcionalidades cruciais devem ser desenvolvidas através de ferramentas nativas. Aplicações voltadas para o cliente final devem ser desenvolvidas em ambientes nativos de forma a que seja possível utilizar as funcionalidades de segurança mais recentes presentes no sistema operativo do dispositivo.

Quando se pretende que a aplicação seja rapidamente disponibilizada para o público final, deve-se combinar os produtos comercializados por terceiros e as aplicações *template* que podem ser adaptadas pela equipa de desenvolvimento. Se o custo é a maior preocupação, a adoção de tecnologias *open source* serão a melhor opção, por exemplo o Apache Cordova.

De seguida, serão apresentadas algumas tecnologias multiplataforma. Escolheram-se estas quatro tecnologias uma vez que são *freeware*. O Xamarin é uma tecnologia que foi adquirida pela Microsoft [7]. No caso do Apache Cordova, é uma tecnologia estável e adotada por empresas internacionais [8]. Mais de dez mil empresas confiam no Sencha Touch, entre elas a

Airbus e Amazon [9]. Por fim, a ferramenta Appcelerator Titanium é também usada por diversas empresas de renome mundial, tais como, Cisco e T-Mobile [10].

2.2.1.1 Xamarin

Nos dias de hoje é inevitável incluir o Xamarin nas opções de escolha para uma tecnologia multiplataforma pois, após a compra por parte da Microsoft, passou a ser vista como uma alternativa viável no mercado e pelos profissionais da área das tecnologias de informação [11].

A Xamarin disponibiliza uma panóplia de produtos Xamarin para o desenvolvimento de aplicações móveis. Entre estes produtos podemos encontrar o Xamarin for Visual Studio [7], por exemplo, que nos permite desenvolver uma aplicação nativaⁱ para cada uma das três plataformas. O Xamarin Test Cloud [12] é uma *framework* de testes que permite correr testes automatizados em milhares de dispositivos reais na *cloud* e obter os relatórios desses testes. Através destes testes é possível descobrir, por exemplo, problemas de performance em cada sistema operativo. Outro produto que a Xamarin disponibiliza é o Xamarin Insights [13] que juntamente com a plataforma HockeyApp possibilita monitorizar a interação do utilizador com a app. Isto é, torna possível fazer uma análise e uma consulta das falhas da aplicação que são reportadas, que tipo de utilizadores a aplicação tem, e como estes interagem com a aplicação. O Xamarin.iOS e o Xamarin.Android é um conjunto de tecnologias que através do C# permite desenvolver aplicações nativas para as plataformas correspondentes (iOS e Android). Estas tecnologias fornecem controlos UI nativos e ferramentas de *build*. Isto tudo, respeitando o aspeto das aplicações nativas da aplicação [7]. Para além dos produtos referidos anteriormente, a Xamarin providencia o Xamarin.Forms [14] que permite implementar aplicações com uma UI nativa para iOS, Android e Windows, partilhando o mesmo código C# em todas as plataformas. Por exemplo, o controlo Entry em Xamarin.Forms é mapeado para um elemento UI de cada plataforma nativa. Neste caso será uma UITextView em iOS, uma EditText em Android e uma TextBox em Windows.

Uma das grandes vantagens no desenvolvimento de aplicações móveis em Xamarin é a facilidade no reaproveitamento de código permitindo ao programador partilhar a mesma camada de negócios (chamadas a serviços, classes de domínio, entre outros) para todas as plataformas. Assim sendo, o programador focar-se-á no desenvolvimento da interface e nas chamadas às APIs específicas de cada plataforma [7].

O uso do C# em Xamarin possibilita o recurso a *features* como funções *async/await* e expressões lambda [11]. Atualmente já é possível encontrar aplicações móveis em Xamarin em quase todas as áreas de negócio: finanças, governo, transportes, *media*, energia, saúde, sistemas sociais, entre outros [15].

2.2.1.2 Apache Cordova

O Apache Cordova é uma plataforma de desenvolvimento de aplicações móveis através de HTML, JavaScript e CSS, e este desenvolvimento poderá ser auxiliado por *frameworks mobile* como JQuery Mobile, Ionic, WinJS, entre outras. Estas *frameworks* auxiliam o programador *mobile* no desenho da interface com o utilizador [16].

Quando se desenvolve uma aplicação nesta tecnologia, está inerente a criação de uma aplicação híbrida na qual o código é compilado para diversas plataformas, como por exemplo Android, iOS, Windows Phone, Ubuntu, entre outras. Também são instaladas todas as APIs e *plugins* necessários para aceder a funções nativas do dispositivo, funções estas que podem ser a câmara, a localização e o acelerómetro, por exemplo [17].

A aplicação em Apache Cordova será uma WebView que ocupa o ecrã por completo e que executa num *container* nativo. Assim sendo, as páginas web manter-se-ão as mesmas independentemente da mudança do *container* que depende do sistema operativo [17].

Como referido anteriormente, uma das grandes vantagens é o desenvolvimento de uma app que abrange qualquer plataforma móvel. Apesar disso, existem problemas relacionados com o desempenho, associado ao facto de existir um *browser* embebido [18].

2.2.1.3 Sencha Touch

Uma das tecnologias mais utilizadas para o desenvolvimento de aplicações web, é o Sencha Touch. Baseado no padrão de software Model – View - Controller (MVC) em JavaScript, esta *framework* permite o desenvolvimento de uma aplicação móvel em HTML5 para dispositivos Android, iOS, Windows e BlackBerry [9].

Uma das vantagens da adoção desta tecnologia é que possibilita o desenvolvimento da aplicação em tempo reduzido e uma maior personalização dos componentes da interface com o utilizador [9]. Só é necessária a criação de um ficheiro HTML e JavaScript, apresentando um comportamento idêntico ao da tecnologia Apache Cordova em termos de tradução de cada página para cada plataforma [19].

2.2.1.4 Appcelerator Titanium

Appcelerator é uma ferramenta *open-source* que permite criar aplicações para tablets, desktops e dispositivos móveis, em plataformas como iOS, Android e Windows Phone a partir do mesmo código JavaScript [10].

Apesar de aparentar ser uma tecnologia equivalente ao Apache Cordova, as aplicações Appcelerator Titanium traduzem a interface desenvolvida para controlos nativos da plataforma do dispositivo. Basicamente, esta tecnologia disponibiliza ao utilizador uma experiência nativa em termos de UI. Para tal, em vez de ser manipulado um HTML DOM, como é feito em Apache Cordova, o JavaScript interage com a ferramenta Appcelerator Titanium [20].

Esta tecnologia disponibiliza objetos UI, como por exemplo botões e listas, e estes objetos são então representados pelos respectivos controlos nativos. No entanto, nem todas as particularidades de cada plataforma conseguem ser abstraídas levando à necessidade de desenvolver algum código específico para cada plataforma [21].

Tal como o Xamarin, a Appcelerator disponibiliza uma panóplia de produtos além do SDK Titanium para desenvolver aplicações móveis. Também é possível usufruir de alguns serviços de *cloud* que poderão ser utilizados como *backend* da aplicação, de testes e de capacidades analíticas para avaliar possíveis problemas que os utilizadores possam ter [22].

2.2.2 Trabalho anterior relacionado

No ano de 2011, no âmbito da disciplina de Projeto/Estágio da licenciatura em engenharia informática no ISEP, foi desenvolvido um projeto semelhante à aplicação móvel do Portal, “ISEP Mobile” [23]

O principal objetivo deste projeto foi a implementação de três módulos: parque de estacionamento, horário e salas disponíveis.

Através da funcionalidade do parque de estacionamento, os utilizadores em tempo real teriam acesso ao número de lugares disponíveis, informação sobre a tendência do estado do mesmo e consulta do histórico. Com o módulo do horário, o utilizador poderia consultar a informação relevante sobre as suas aulas. Por último, o módulo das salas disponíveis permitiria saber a disponibilidade de uma determinada sala, numa hora específica.

As tecnologias usadas neste projeto foram o MySQL como sistema de gestão de base de dados, o *toolkit* NuSoap para desenvolvimento de *webservices* e a aplicação foi desenvolvida em PhoneGap, atual Apache Cordova.

Neste trabalho, os alunos desenvolveram a aplicação móvel, ficando o ISEP responsável pela disponibilização dos serviços com informação real. No entanto, numa fase inicial, dado que os serviços não tinham sido disponibilizados foi necessário implementar uma estrutura de *backend* que permitisse aos alunos testarem o a sua implementação.

Foi relatado que o principal objetivo foi cumprido e o desenvolvimento dos três módulos foi feito com sucesso. Contudo, o principal contratempo que obtiveram foi a indisponibilidade dos servidores do ISEP durante um largo período. Para além disso, não lhes foi possível testar o módulo horário e salas disponíveis com *data* real visto que os serviços destes módulos não tinham sido desenvolvidos.

Apesar deste trabalho ser uma aplicação do Portal ISEP embrionária, esta aplicação não esteve acessível para a comunidade académica, nem foram feitos testes de funcionalidade com utilizadores reais. Para além disso, não foi feito um estudo aprofundado de que tecnologia a adotar para o desenvolvimento da mesma. Foi notório que, a nível de infraestruturas informáticas, o ISEP ainda não estava preparado para oferecer este produto à sua comunidade.

2.3 Análise de valor

Esta secção é composta por quatro subsecções que têm como principal objetivo auxiliar na análise de valor do tema.

Em primeiro lugar, baseado no modelo de Peter Koen, identifica-se os cinco elementos chave do modelo de acordo com o tema (secção 2.3.1). De seguida, apresenta-se o conceito de valor, valor para o cliente e valor percebido, e que benefícios e prejuízos o cliente terá (secção 2.3.2). Segue-se a apresentação da ideia de negócio segundo o modelo Canvas (secção 2.3.3) e por fim, com recurso ao Método AHP, foi possível escolher duas tecnologias entre quatro alternativas (secção 2.3.4).

2.3.1 Modelo de desenvolvimento de novo conceito

Segundo Peter Koen, o modelo de desenvolvimento de novo conceito (NCD) é composto por três partes essenciais (ver Figura 1):

- O *engine* é a liderança, a cultura, e a estratégia empresarial da organização que orientam os cinco elementos chave do modelo, é algo que é controlado pela organização.
- Pelos cinco elementos chave do modelo: a identificação da oportunidade (*opportunity identification*), a análise da oportunidade (*opportunity analysis*), a geração da ideia e enriquecimento (*idea generation and enrichment*), a seleção da ideia (*idea selection*) e a definição do conceito (*concept definition*).
- Os fatores influenciadores é algo que a organização dificilmente consegue controlar e que afetam todo o processo de inovação até à comercialização. Estes fatores são as capacidades organizacionais da empresa, os clientes, a concorrência, influências do mundo externo, e desenvolvimento de tecnologias que possam estar envolvidas.

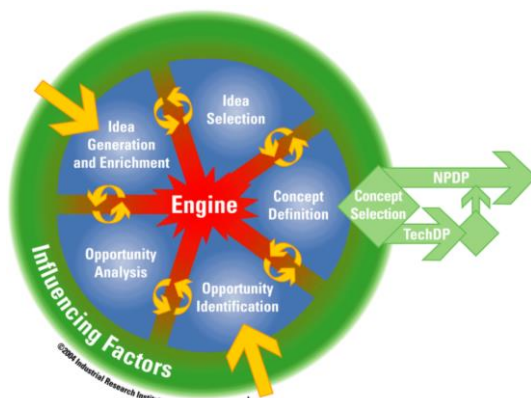


Figura 1 - New concept development model

Fonte: [29]

O facto de o modelo ter uma forma circular, sugere que é expectável que as ideias e conceitos fluam, circulem e iterem entre e ao longo dos cinco elementos. Além disso, o fluxo deve abranger os elementos em qualquer ordem e combinação e um elemento ou mais elementos poderão ser usados mais que uma vez. Tal como se pode ver na figura acima, o modelo é composto por setas que apontam para o modelo e outras que apontam para fora do modelo. Estas representam o início do projeto/ponto inicial do modelo e a saída do conceito para o novo desenvolvimento do produto ² (NPD) ou para o processo tecnologia por etapas ³ (TSG), respetivamente.

A identificação da oportunidade deste projeto ocorreu quando o ISEP reconheceu que existiam vários problemas funcionais nas aplicações móveis do portal que se traduziam num descontentamento da comunidade académica.

Numa análise à oportunidade anteriormente referida surgiram algumas questões pertinentes: as aplicações estão atualizadas? Há uma equipa de desenvolvimento móvel? Se sim, como é caracterizada essa equipa? Há solução para os problemas encontrados?

E foi através destas questões que se percebeu que não havia equipa de desenvolvimento móvel há algum tempo e como tal as aplicações não estavam atualizadas. Além disso, a previsão era de que a equipa de desenvolvimento fosse constituída por um programador e esse programador teria que encontrar soluções para os problemas existentes nas três plataformas, independentemente da sua experiência na área e sem qualquer auxílio dos anteriores programadores.

Passou-se então para uma fase na qual surgia uma ideia e fazia-se a análise dessa ideia sucessivamente até encontrar uma solução adequada.

Ideia 1: um programador melhorar, atualizar e continuar o desenvolvimento das três aplicações. Análise à ideia 1: pelas dificuldades acima referidas, um programador não conseguiria dar resposta às exigências na continuação do desenvolvimento das três aplicações. Trata-se de desenvolvimento contínuo em três linguagens de programação distintas e tudo que acarreta a manutenção de três projetos é muito complexa para ser realizada por um único programador, por exemplo: desenvolvimento de novas funcionalidades, testes, análise e resolução de bugs, entre outros.

Ideia 2: adicionar mais elementos à equipa de desenvolvimento.

Análise à ideia 2: esta ideia foi rejeitada visto que implicaria um maior investimento de recursos financeiros por parte da organização.

Ideia 3: contratar uma empresa outsourcing para o desenvolvimento das aplicações.

Análise à ideia 3: a organização defendeu que existem recursos de qualidade dentro desta para o sucesso do projeto. Além disso, o outsourcing apresenta diversas questões que devem ser

² *New product development*

³ *Technology stage gate*

tidas em conta para o ISEP, como por exemplo: ficaria sem o conhecimento que é adquirido ao longo do desenvolvimento destes projetos; não teria qualquer controlo no desenvolvimento; perda de confidencialidade; criação de uma dependência excessiva; elevados custos associados; má qualidade do serviço prestado;

Ideia 4: mudar a abordagem para um desenvolvimento multiplataforma após o um estudo das tecnologias existentes.

Análise à ideia 4: esta ideia pareceu a ideal dadas as circunstâncias.

No elemento final, na definição de conceito, deu-se início ao desenvolvimento de dois projetos em Apache Cordova e Xamarin, para depois ser escolhido um, e deste modo substituir os três projetos existentes.

Os principais critérios de avaliação foram o tempo de desenvolvimento, que seria mais reduzido e os baixos custos de desenvolvimento nestas tecnologias.

2.3.2 Valor, valor para o cliente e valor percebido

A análise de valor é um estudo organizado que se foca em perceber de que forma o produto pode ser melhorado ao mais baixo custo, sem sacrificar a fiabilidade e a qualidade do mesmo [24].

O valor de um produto poderá ser interpretado de diferentes formas conforme os clientes. As suas características comuns são, o elevado nível de performance, potencial, impacto emocional, estética, entre outros, relativo ao seu custo. Esta ideia também pode ser expressa maximizando a função do produto relativo ao seu custo:

$$Valor = \frac{Performance + Potencial}{Custo} = \frac{Funcionalidade}{Custo}$$

O valor de um produto nem sempre está associado a uma redução de custo. Nalguns casos o valor pode ser aumentado, aumentando a sua funcionalidade e o seu custo desde que o aumento do custo seja menor que o aumento da funcionalidade [25].

Segundo Woodruff [26], existem dois apertos para caracterizar o valor para o cliente: valor desejado e valor percebido. O valor desejado refere-se àquilo que o cliente deseja de um produto ou serviço; o valor percebido é o benefício que o cliente acredita que recebeu de um produto após a sua compra. Ou seja, o valor pode ser observado em diferentes níveis. A baixo nível, o valor para o cliente depende dos atributos desse mesmo produto e do significado que essas características têm para o cliente. Num nível mais elevado, o valor para o cliente pode ser visto como uma recompensa emocional e a realização de um objetivo ou de um desejo.

Percebe-se então que o valor percebido é a opinião que o cliente formula sobre um produto tendo em vista a satisfação das necessidades e exigências consideradas pelo cliente [27].

Baseado nos conceitos acima referidos, prevê-se que a comunidade académica do ISEP vá beneficiar de uma aplicação que lhe permite consultar informação do portal da instituição de uma forma mais acessível e rápida. Não se prevêem custos associados dado que o uso da aplicação é gratuita e estará funcional em todas as plataformas. Na mesma perspetiva de não trazer despesa aos alunos, professores e funcionários, a longo prazo poder-se-á desenvolver mais funcionalidades e incrementá-las a esta aplicação, facilitando assim a rotina diária dos mesmos.

Para a instituição ISEP, os benefícios são vários. Está previsto que na conclusão do desenvolvimento desta aplicação, o ISEP tenha uma aplicação funcional para todas as plataformas, fique com o *know-how* do projeto desenvolvido e conseqüente valorização da instituição. Numa perspetiva de futuro, desde que não existam alterações tecnológicas significativas e profundas, o ISEP poderá contar com uma aplicação estável, fiável e funcional. Poderá ainda colocar-se a hipótese da venda da aplicação a outras instituições similares.

No entanto, o ISEP deve estar preparado para suportar alguns investimentos, nomeadamente, os custos implicados na equipa de desenvolvimento e do tempo despendido no desenvolvimento deste novo projeto. A longo prazo estes sacrifícios manter-se-ão uma vez que as alterações e melhoramentos serão uma constante.

2.3.3 Modelo Canvas

O quadro de modelo de negócio⁴ é um modelo de gestão estratégica para o desenvolvimento de novos modelos de negócio. Este modelo permite tornar as propostas de valor do negócio visíveis e tangíveis, e fáceis de discussão e controlo. Integrar a proposta de valor num modelo de negócio viável é uma forma de adquirir valor para a organização [28].

Integrando perfeitamente o *Business Model Canvas* com a proposta de valor, as empresas focar-se-ão nos detalhes de como criar valor para os clientes. O modelo Canvas está representado num gráfico visual com elementos que descrevem a proposta de valor de uma empresa ou produto [28].

No total, o modelo Canvas é composto por nove elementos:

- Segmentos de clientes ou *Customer Segments*
São grupos de clientes e/ou organizações que uma organização pretende servir e criar valor, através de uma proposta de valor orientada aos mesmos.
- Propostas de valor ou *Value Propositions*
São baseadas numa panóplia de produtos ou serviços que criam valor para um determinado segmento de clientes.
- Canais ou *Channels*

⁴ *Business Model Canvas*

Descrevem como uma proposta de valor é comunicada e entregue a um segmento de mercado, através da comunicação, distribuição e canais de venda.

- **Relacionamento com o Cliente ou *Customer Relationships***
Determina que tipo de relação será estabelecida e mantida com cada segmento de clientes e nos quais é explicado como se obtêm novos clientes e se mantêm.
- **Fluxos de rendimento ou *Revenue Streams***
Resulta do sucesso da proposta de valor oferecida a um segmento de cliente. É a forma como a organização obtém valor com um preço que os clientes estão dispostos a pagar.
- **Recursos-chave ou *Key Resources***
São os recursos necessários para oferecer e entregar os elementos anteriormente descritos.
- **Atividades-chave ou *Key Activities***
São as atividades mais relevantes que a organização precisará de realizar com sucesso.
- **Parceiros-chave ou *Key Partnerships***
Refere-se à rede de fornecedores e parceiros que trarão recursos e atividades externas.
- **Estrutura de custos ou *Cost Structure***
Menciona todos os custos necessários para o funcionamento do modelo de negócio.

De seguida será apresentado o *Business Model Canvas* do modelo de negócio correspondente à aplicação do Portal do ISEP (ver Figura 2).

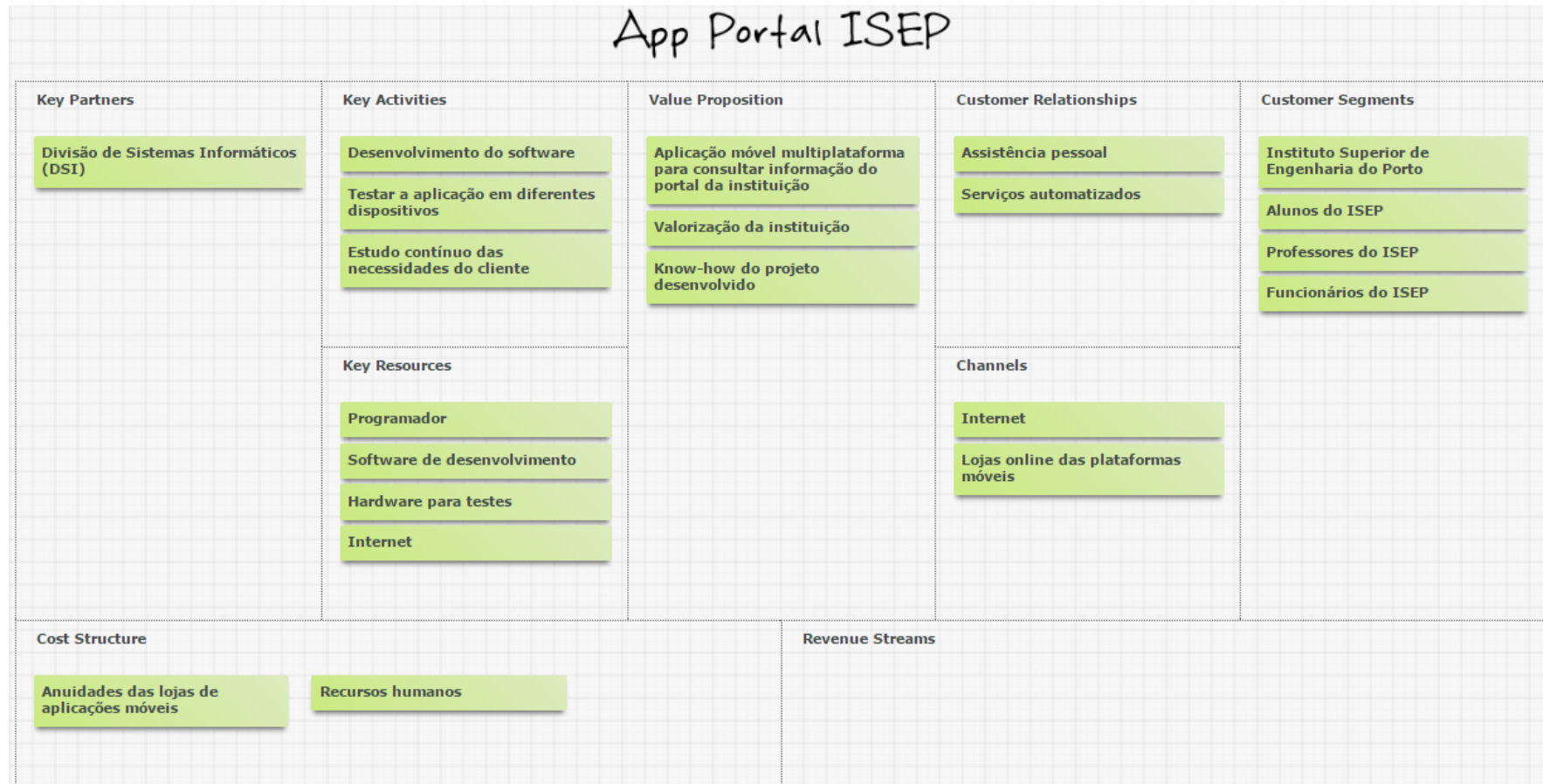


Figura 2 – Business Model Canvas

De acordo com o modelo, e uma vez que este projeto se destina à comunidade académica do ISEP, definiu-se que os segmentos de mercado seriam o próprio ISEP, os alunos, os funcionários e os professores da instituição.

O desenvolvimento de uma aplicação móvel multiplataforma relativa ao portal, é uma das propostas de valor que este projeto apresenta à comunidade académica e à instituição. Consequentemente a instituição fica mais valorizada ao mesmo tempo que adquire o conhecimento do projeto. Esta aquisição de *know-how* permitirá ao ISEP não estar dependente de terceiros para resolução de problemas e implementações de novas funcionalidades na aplicação.

Para além da internet, os canais que se definiram para chegar ao cliente foram as lojas de aplicações online, Apple Store, Google Play e Windows Store, nas quais será possível fazer-se o download da aplicação para o dispositivo. Em termos de relações com o cliente, pretende-se fornecer assistência técnica sempre que um problema for reportado pelo ISEP e através de serviços automatizados será possível fazer *updates* à aplicação.

Uma vez que se trata de um serviço totalmente gratuito para a comunidade académica, não se prevê qualquer fluxo de rendimento. Uma possível alternativa para rentabilizar as aplicações seria colocar publicidade nas mesmas. No entanto, apesar de terem sido sugeridas pelo aluno, a instituição reforçou que se trataria de um investimento para melhorar o quotidiano da comunidade académica.

Os recursos necessários para o desenvolvimento deste projeto serão o programador, o software de desenvolvimento, os dispositivos móveis e a internet. Através destes, será possível desenvolver a aplicação, testá-la e estudar as necessidades do cliente. Estas serão as atividades fulcrais deste projeto.

É imprescindível criar uma parceria com o DSI uma vez que é neste departamento que se encontra toda a informação e dados do portal. Dado que o software de desenvolvimento é gratuito e que o hardware utilizado nos testes serão os dispositivos móveis pessoais, os únicos custos associados ao projeto serão a anuidade que é necessário pagar para disponibilizar as aplicações nas lojas, e o custo do programador.

2.3.4 Método Analytic Hierarchy Process para comparação das soluções

Para fazer uma análise de decisão multicritério, recorreu-se ao método de análise hierárquica criado pelo professor Thoma L. Saaty. O método *Analytic Hierarchy Process* (AHP) deste projeto foi elaborado de acordo com as sete fases definidas pelo professor Saaty [29].

Em primeiro lugar, definiu-se o problema e estruturou-se o diagrama hierárquico. O problema real estudado é a escolha das duas plataformas multiplataforma para o desenvolvimento deste projeto. Os critérios estabelecidos para avaliar as hipóteses foram a interface com o utilizador, a experiência da equipa de desenvolvimento e as ferramentas disponíveis para auxiliar no desenvolvimento da aplicação. Após alguma pesquisa e através dos critérios definidos,

delinearam-se as quatro tecnologias possíveis, Xamarin, Apache Cordova, Sencha Touch e Appcelerator [30]. Na secção do estado da arte (ver Secção 2.2) as quatro tecnologias são apresentadas.

Passou-se para a segunda fase, a fase da comparação das alternativas e critérios. Com o auxílio da escala fundamental definida por Satty (ver Tabela 1), foi possível estabelecer prioridades entre os elementos para cada nível da hierarquia (ver Tabela 2) [30].

Tabela 1 - Escala fundamental de Satty

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2, 4, 6, 8	Valores Intermediários	Quando se procura uma condição de compromisso entre duas definições

Tabela 2 - Matriz de Comparação dos critérios do Segundo Nível

	Interface	Experiência	Ferramentas
Interface	1/1	1/4	1/2
Experiência	4/1	1/1	2/1
Ferramentas	2/1	1/2	1/1

Através da matriz de comparação dos critérios (ver Tabela 2) é possível perceber, por exemplo, que a experiência nas tecnologias é um fator mais importante do que a interface com o utilizador.

Na fase número três, procedeu-se à normalização dos valores da matriz de comparações. Para tal, calculou-se o total de cada coluna (ver Tabela 3) e dividiu-se cada valor da matriz por esse

total (ver Tabela 5). Na Tabela 4 é apresentado o passo intermédio que demonstra o procedimento da fase três [30].

Tabela 3 – Matriz de Comparação dos critérios do Segundo Nível com soma das colunas

	Interface	Experiência	Ferramentas
Interface	1/1	1/4	1/2
Experiência	4/1	1/1	2/1
Ferramentas	2/1	1/2	1/1
Soma	7	7/4	7/2

Tabela 4 - Matriz de Comparação dos critérios do Segundo Nível – Passo intermédio

	Interface	Experiência	Ferramentas
Interface	1/7	(1/4)/(7/4)	(1/2)/(7/2)
Experiência	4/7	1/(7/4)	2/(7/2)
Ferramentas	2/7	(1/2)/(7/4)	1/(7/2)

Tabela 5 – Matriz Normalizada dos Critérios do Segundo Nível

	Interface	Experiência	Ferramentas
Interface	1/7	4/28	2/14
Experiência	4/7	4/7	4/7
Ferramentas	2/7	4/14	2/7

Nesta fase, é necessário obter o vetor de prioridades que tem como principal objetivo a identificação da ordem de importância de cada critério. Para tal, procedeu-se ao cálculo da média aritmética dos valores de cada linha da matriz normalizada (ver Tabela 6) [30].

Tabela 6 – Vetor de Prioridades da Matriz Normalizada

	Interface	Experiência	Ferramentas	Prioridade Relativa
Interface	1/7	1/7	1/7	0.1429
Experiência	4/7	4/7	4/7	0.5714
Ferramentas	2/7	2/7	2/7	0.2857

Através dos resultados obtidos na Tabela 6 o critério “Experiência” surge em primeiro lugar, seguido do critério “Ferramentas” e “Interface”.

Na fase quatro, pretendeu-se avaliar a consistência das prioridades relativas. É então necessário calcular a Razão de Consistência (RC) de forma a avaliar se os julgamentos foram consistentes. Segundo Saaty, se o RC for menor que 10% então os julgamentos são coerentes [30].

Considera-se $[Ax = \lambda_{max} x]$, no qual x é o vetor próprio.

$$Ax = \lambda_{max} x = \begin{pmatrix} 1 & 1/4 & 1/2 \\ 4 & 1 & 2 \\ 2 & 1/2 & 1 \end{pmatrix} \begin{pmatrix} 0.14 \\ 0.57 \\ 0.29 \end{pmatrix} \cong \lambda_{max} \begin{pmatrix} 0.14 \\ 0.57 \\ 0.29 \end{pmatrix} =$$

$$= \begin{pmatrix} 0.43 \\ 1.71 \\ 0.86 \end{pmatrix} \cong \lambda_{max} \begin{pmatrix} 0.14 \\ 0.57 \\ 0.29 \end{pmatrix}$$

Assim sendo, o valor próprio é calculado da seguinte forma:

$$\lambda_{max} = average \left\{ \frac{0.43}{0.14} + \frac{1.71}{0.57} + \frac{0.86}{0.29} \right\} = 3.012$$

Após o cálculo de λ_{max} , calcula-se o Índice de Consistência (IC) para que seja possível proceder ao cálculo da Razão de Consistência (RC), em que n é o número de critérios e o número de critérios.

$$IC = \frac{\lambda_{max} - n}{n - 1} = \frac{3.01 - 3}{3 - 1} = 0.005$$

A razão é calculada através da seguinte fórmula: $RC = \frac{IC}{IR}$

O valor de IR é valor obtido de acordo com a Figura 3.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Figura 3 - Valores de IR para matrizes quadradas de ordem n

Uma vez que já temos calculado o valor de IC e uma vez que a nossa matriz é constituída por três critérios ($IR = 0.58$), é então possível calcular o valor de RC.

$$RC = \frac{IC}{IR} = \frac{0.005}{0.58} = 0.009$$

Uma vez que o valor RC é menor que 0.1 (10%), pode-se concluir que os valores das prioridades relativas do exemplo utilizado estão consistentes.

Na fase cinco foi necessário fazer a construção da matriz de comparação paritária para cada critério em relação às alternativas selecionadas [30].

Tal como anteriormente descrito na fase dois, construiu-se a Matriz de Comparação para o critério "Interface" (ver Tabela 7), ao cálculo da soma de cada coluna (ver Tabela 8), e dividiu-

se cada valor da matriz por esse total (ver Tabela 9). Obteve-se então a tabela normalizada para o critério Interface (ver Tabela 10).

Tabela 7 – Matriz de Comparação do critério Interface

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	3/1	2/1	5/1
Sencha Touch	1/3	1	2/1	2/1
Apache Cordova	1/2	1/2	1	3/1
Appcelerator	1/5	1/2	1/3	1

Tabela 8 - Matriz de Comparação do critério Interface com soma das colunas

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	3/1	2/1	5/1
Sencha Touch	1/3	1	2/1	2/1
Apache Cordova	1/2	1/2	1	3/1
Appcelerator	1/5	1/2	1/3	1
Soma	61/30	10/2	16/3	11

Tabela 9 - Matriz de Comparação do critério Interface – Passo intermédio

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1/(61/30)	3/5	2/(16/3)	5/11
Sencha Touch	(1/3)/(61/30)	1/5	2/(16/3)	2/11
Apache Cordova	(1/2)/(61/30)	(1/2)/5	1(16/3)	3/11
Appcelerator	(1/5)/(61/30)	(1/2)/5	(1/3)/(16/3)	1/11

Tabela 10 - Matriz Normalizada do critério Interface

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator	Prioridade Relativa
Xamarin	30/61	3/5	6/16	5/11	0.48
Sencha Touch	10/61	1/5	6/16	2/11	0.23
Apache Cordova	15/61	1/10	3/16	3/11	0.20
Appcelerator	6/61	1/10	1/16	1/11	0.09

Seguindo os mesmos passos para o cálculo da Matriz Normalizada do critério Interface, foi possível calcular a Matriz Normalizada do critério Experiência (ver Tabela 14). Elaborou-se a

matriz de comparação do critério Experiência (ver Tabela 11) e os respectivos cálculos para chegar à matriz normalizada (ver Tabela 12 e Tabela 13).

Tabela 11 – Matriz de Comparação do critério Experiência

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	5/1	1/2	5/1
Sencha Touch	1/5	1	1/7	1/2
Apache Cordova	2/1	7/1	1	7/1
Appcelerator	1/5	2/1	1/7	1

Tabela 12 - Matriz de Comparação do critério Experiência com soma das colunas

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	5	2	5
Sencha Touch	1/5	1	1/7	1/2
Apache Cordova	2	7	1	7
Appcelerator	1/5	2	1/7	1
Soma	17/5	15	23/7	27/2

Tabela 13 - Matriz de Comparação do critério Experiência – Passo intermédio

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	$1/(17/5)$	$5/15$	$2/(23/7)$	$5/(27/2)$
Sencha Touch	$(1/5)/(17/5)$	$1/15$	$(1/7)/(23/7)$	$(1/2)/(27/2)$
Apache Cordova	$2/(17/5)$	$7/15$	$1/(23/7)$	$7/(27/2)$
Appcelerator	$(1/5)/(17/5)$	$2/15$	$(1/7)/(23/7)$	$1/(27/2)$

Tabela 14 - Matriz Normalizada do critério Experiência

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator	Prioridade Relativa
Xamarin	5/17	1/3	14/23	10/27	0.40
Sencha Touch	1/17	1/15	1/23	1/27	0.05
Apache Cordova	10/17	7/15	7/23	14/27	0.47
Appcelerator	1/17	2/15	1/23	2/27	0.08

Uma vez mais recorreu-se aos passos anteriormente descritos para obtermos a matriz normalizada do critério Ferramentas (ver Tabela 18). Nas tabelas seguintes são apresentados os passos necessários para a obtenção da matriz normalizada (ver Tabela 15, Tabela 16 e Tabela 17).

Tabela 15 - Matriz de Comparação do critério Ferramentas

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	3/1	1/3	3/1
Sencha Touch	1/3	1	1/4	1/2
Apache Cordova	3/1	4/1	1	4/1
Appcelerator	1/3	2/1	1/4	1

Tabela 16 - Matriz de Comparação do critério Ferramentas com soma das colunas

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1	3	1/3	3
Sencha Touch	1/3	1	1/4	1/2
Apache Cordova	3	4	1	4
Appcelerator	1/3	2	1/4	1
Soma	14/3	10	11/6	17/2

Tabela 17 - Matriz de Comparação do critério Ferramentas – Passo intermédio

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator
Xamarin	1/(14/3)	3/10	(1/3)/(11/6)	3/(17/2)
Sencha Touch	(1/3)/(14/3)	1/10	(1/4)/(11/6)	(1/2)/(17/2)
Apache Cordova	3/(14/3)	4/10	1/(11/6)	4/(17/2)
Appcelerator	(1/3)/(14/3)	2/10	(1/4)/(11/6)	1/(17/2)

Tabela 18 - Matriz Normalizada do critério Ferramentas

	Xamarin	Sencha Touch	Apache Cordova	Appcelerator	Prioridade Relativa
Xamarin	3/14	3/10	6/33	6/17	0.26
Sencha Touch	1/14	1/10	6/44	1/17	0.09
Apache Cordova	9/14	2/5	6/11	8/17	0.52
Appcelerator	1/14	1/5	6/44	2/17	0.13

Através do diagrama demonstrado em seguida é possível observar a importância relativa de cada uma das alternativas que compõe a estrutura hierárquica do problema em questão (ver Figura 4).

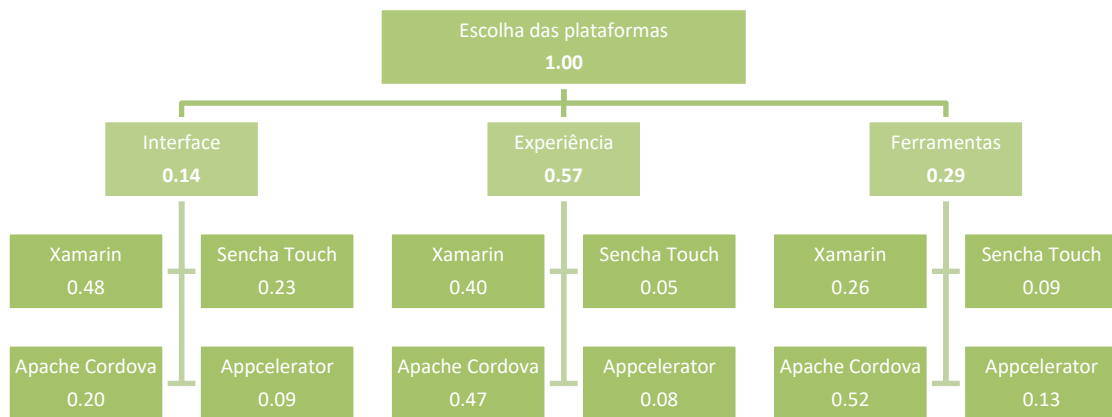


Figura 4 - Diagrama hierárquico com importâncias relativas associadas

Na etapa seis, obtemos as prioridades compostas das alternativas multiplicando os valores anteriores e os das prioridades relativos, obtidos no início do método. As linhas da matriz correspondem às alternativas e as colunas aos critérios adotados. Xamarin, Sencha Touch, Apache Cordova e Appcelerator é a ordem das alternativas e Interface, Experiência e Ferramentas a ordem dos critérios [30].

$$\begin{bmatrix} 0.48 & 0.40 & 0.26 \\ 0.23 & 0.05 & 0.09 \\ 0.20 & 0.47 & 0.52 \\ 0.09 & 0.08 & 0.13 \end{bmatrix} * \begin{bmatrix} 0.14 \\ 0.57 \\ 0.29 \end{bmatrix} = \begin{bmatrix} 0.37 \\ 0.09 \\ 0.45 \\ 0.10 \end{bmatrix}$$

Por último, na fase sete escolhe-se a opção mais adequada de acordo com os critérios definidos inicialmente e as suas respetivas importâncias.

Tabela 19 – Propriedades Compostas do problema

	Prioridades Compostas
Xamarin	0.37
Sencha Touch	0.09
Apache Cordova	0.45
Appcelerator	0.10

Através dos resultados da Tabela 19, é possível perceber que as melhores opções entre Xamarin, Sencha Touch, Apache Cordova e Appcelerator são o Xamarin e o Apache Cordova.

Através de uma forma estruturada que é o método AHP, após a definição de vários critérios, esta ferramenta demonstrou ser essencial no auxílio à tomada de decisão entre as quatro tecnologias escolhidas inicialmente.

3 Design

Neste capítulo será abordado o design da solução para o problema deste trabalho, apresentação da estrutura lógica da solução, análise arquitetural do projeto e uma comparação das várias alternativas.

3.1 Design de software

Tal como o desenvolvimento de aplicações *desktop*, o desenvolvimento de aplicações móveis partilha princípios de design de software. No entanto, as aplicações móveis devem ter em atenção o processo de gestão do ciclo de vida da aplicação móvel. De seguida serão apresentados alguns padrões de software móvel úteis na construção, manutenção e compreensão de aplicações móveis [31].

- Model, View, Controller (MVC)
O MVC [32] consiste na separação da interface do utilizador (View), da camada que lida com as interações (Controller) e dos dados que compõe a aplicação (Model). O foco principal neste padrão prende-se na camada View e Controller, uma vez que a camada Model é opcional.
- Model, View, ViewModel (MVVM)
Padrão similar ao padrão MVC, no qual o Controller é substituído pelo ViewModel. Assim, o ViewModel que trabalha numa camada inferior à da UI, permite expor a data e comandos que a View necessita e que são obtidos através do Model. Uma das vantagens deste padrão é que, através dos bindings, possibilita uma comunicação bidirecional [33].
- Singleton
Este padrão assegura que uma classe é apenas instanciada uma vez e fornece um ponto de acesso global para essa instancia. Verifica-se a necessidade do uso deste padrão, por exemplo, quando se usa SQLite numa aplicação móvel visto que só desejamos que exista uma instancia da base de dados [31].

- **Provider**
Trata-se de um padrão que tem como principal a reutilização de códigos em aplicações Silverlight, WPF e WinForms. O código partilhado é escrito através de classes interfaces e abstratas, enquanto implementações concretas da plataforma são desenvolvidas e passadas quando o código é usado [31].
- **Async**
O padrão async é usado quando existem tarefas *long-running* que necessitam de ser executadas, sem bloquearem a UI ou o processo corrente. Estas *tasks* devem executar numa outra *thread* enquanto a *thread* corrente continua a processar e a esperar por uma resposta do processo em *background*. Quando a informação é retornada, a *thread* corrente atualiza a UI [31].

Uma vez que em Xamarin é mais comum desenvolverem-se aplicações baseadas no padrão MVVM, para ambos os projetos deste trabalho adotou-se este padrão. Para a implementação do mesmo no projeto Apache Cordova, recorreu-se ao Typescript.

Na Figura 5 é possível observar a estrutura deste padrão no módulo Login do projeto. O LoginView é um elemento visual responsável pela definição e estrutura de toda a interface, assim como na recolha de toda a informação sobre a interação do utilizador com a aplicação. Através da LoginViewModel é possível implementar propriedades, comandos, entre outros, sendo responsável pela implementação de toda a camada de negócio. A LoginModel é a classe que representa o modelo de dados deste módulo. Com a LoginModel é possível estabelecer comunicação com *webservices* e/ou base de dados.

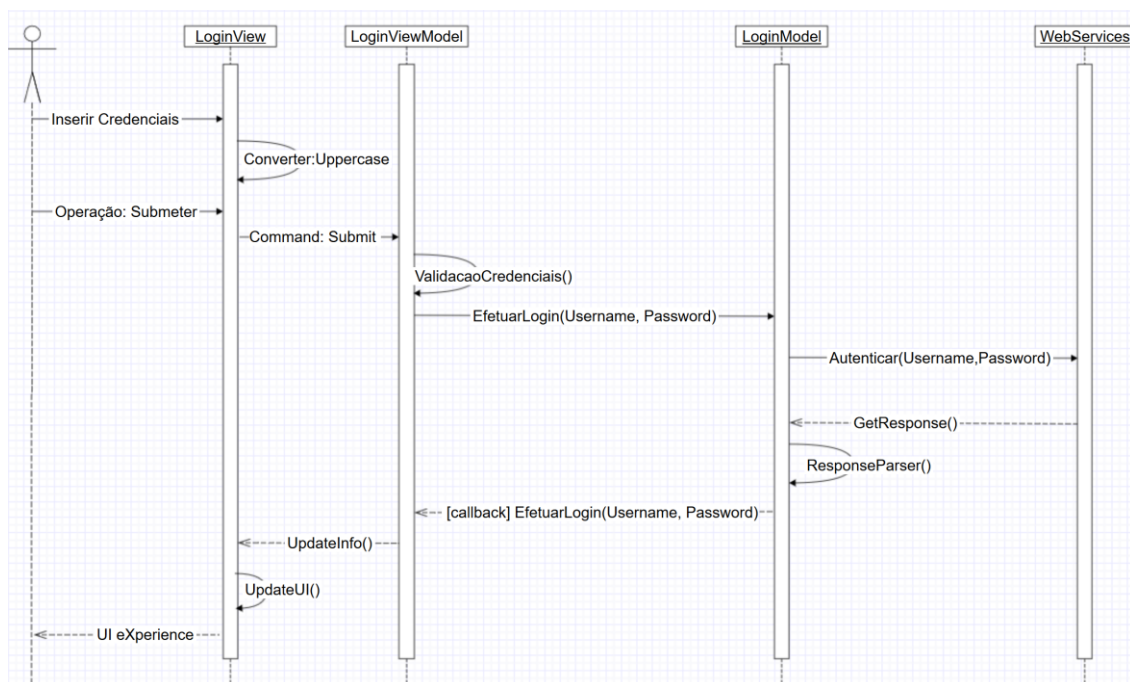


Figura 5 – Módulo Login (Padrão MVVM)

Desta forma, é criada uma estruturação clara na qual é possível separar por camadas, a interface, a lógica de negócio e os seus dados.

3.2 Casos de uso

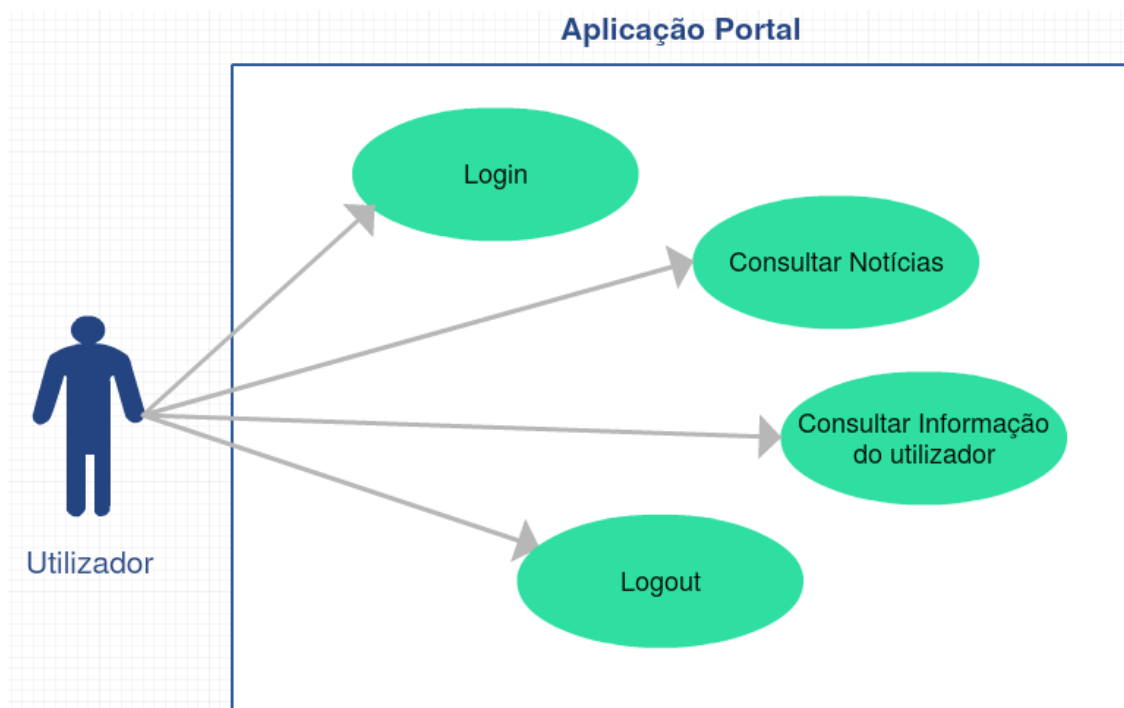


Figura 6 – Casos de uso

No diagrama de casos de uso apresentado na anteriormente (ver Figura 6), são apresentados os requisitos funcionais que o aluno desenvolveu. Como referido anteriormente, o aluno foi responsável pelo desenvolvimento das funcionalidades da aplicação. Maioritariamente as funcionalidades têm como principal objetivo a consulta de informação do Portal.

Tabela 20 - Especificação do caso de uso 1

Caso de uso 1 – Login	
Atores	Utilizador
Pré-Condições	Utilizador ter acesso ao Portal do ISEP
Pós-Condições	
Fluxo Básico de Eventos	
Ações do ator	Ações do sistema
1. Abre a aplicação	1.1 Aplicação apresenta página Login
2. Inserir credenciais	2.1 Login efetuado com sucesso

Neste primeiro caso de uso (ver Tabela 20 - Especificação do caso de uso), o utilizador poderá fazer login na aplicação do Portal depois de inserir as suas credenciais.

Tabela 21 - Especificação do caso de uso 2

Caso de uso 2 – Consultar Notícia	
Atores	Utilizador
Pré-Condições	Utilizador ter acedido à aplicação com sucesso. Página principal aberta.
Pós-Condições	
Fluxo Básico de Eventos	
Ações do ator	Ações do ator
1. Seleciona opção “Notícias” no menu principal	1.1 Lista todas as notícias
2. Seleciona notícia pretendida	2.1 Apresenta a notícia selecionada

Um das funcionalidades da aplicação do Portal é poder consultar notícia. O requisito especificado na Tabela 21 permite que o utilizador leia uma determinada notícia.

Tabela 22 - Especificação do caso de uso 3

Caso de uso 3 – Consultar Informação do utilizador	
Atores	Utilizador
Pré-Condições	Utilizador ter acedido à aplicação com sucesso. Página principal aberta.
Pós-Condições	
Fluxo Básico de Eventos	
Ações do ator	Ações do ator
1. Seleciona opção “Informação do utilizador” no menu principal	1.1 Apresenta foto, nome, estatuto e mail do utilizador

A funcionalidade representada na Tabela 22 permite ao utilizador consultar a sua informação pessoal.

Tabela 23 – Especificação do caso de uso 4

Caso de uso 4 – Logout	
Atores	Utilizador
Pré-Condições	Utilizador ter acedido à aplicação com sucesso. Página principal aberta.

Pós-Condições	Sessão terminada Cache da aplicação limpa
Fluxo Básico de Eventos	
Ações do ator	Ações do ator
1. Seleciona opção “Terminar Sessão” no menu principal	1.1 Apresenta uma caixa de diálogo perguntando ao utilizador se tem a certeza.
2. Utilizador pressiona o botão “Sim”	2.1 Termina sessão

Este caso de uso (ver Tabela 23) representa a funcionalidade de terminar sessão da aplicação. Após o término da sessão, o sistema deve apagar toda a informação guardada ao longo do funcionamento da aplicação.

3.3 Estrutura Lógica

No diagrama (ver Figura 7), é apresentada toda a estrutura analítica do projeto desenvolvido pelo aluno. Esta estrutura está dividida em duas subsecções, o *frontend* e o *backend*.

O *backend* é a área que tem como principal objetivo o acesso a base de dados através dos diversos serviços consumidos pela Web API e, no outro lado, a área de *frontend* é composto pela aplicação móvel com uma estrutura modular.

Dado que o projeto se enquadra no portal da instituição, todo o projeto desenvolvido pelo aluno consistiu em implementar a aplicação (*frontend*), aproveitando os recursos de *backend* disponibilizados e já desenvolvidos para o Portal do ISEP. Como tal, o aluno não tratou do desenvolvimento base de dados e *webservices*.

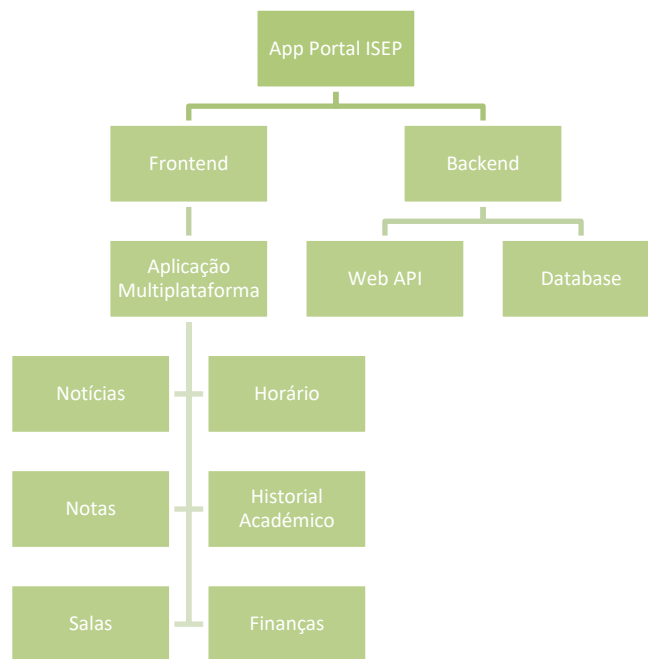


Figura 7 – Estrutura analítica do projeto

3.4 Arquitetura

3.4.1 Xamarin

Uma das grandes vantagens do uso de Xamarin é a possibilidade de partilha de código entre as aplicações de cada plataforma. Assim sendo, a Xamarin desenvolveu dois métodos distintos para o efeito: Shared Projects e Portable Class Libraries (PLC) [34].

Apesar de ambos os métodos permitirem desenvolver código que possa ser partilhado por várias aplicações, tornou-se imprescindível estudar a melhor opção para o projeto. De seguida será apresentada a arquitetura de cada método, assim como suas vantagens e desvantagens.

O método Shared Asset Projects [35] ou Shared Project utiliza um conjunto de ficheiros e oferece uma rápida e simples forma de partilhar código numa solução. Por norma, é necessário recorrer a diretivas de compilação para que seja possível fazer a separação do código para cada plataforma, como no exemplo:

```

#if __Android__
  //código correspondente à plataforma Android
#endif
#if __IOS__
  //código correspondente à plataforma Android
#endif
  
```

Código 1 – Exemplo de diretivas de compilação

Fonte: [36]

Assim sendo, uma vantagem deste método é a centralização dos códigos específicos de cada plataforma no projeto partilhado. Para além disso, é possível incluir referências específicas de cada plataforma. Mas esta vantagem acarreta uma desvantagem que é o facto de só conseguirmos obter os erros de uma determinada plataforma na compilação e nos testes dessa mesma plataforma. Isto porque a diretiva de compilação só é ativada na compilação [36].

Tal como podemos observar na Figura 8, neste método não é gerada qualquer biblioteca uma vez que o código é acoplado no executável de cada plataforma, o que impossibilita a execução de testes unitários, por exemplo [36].

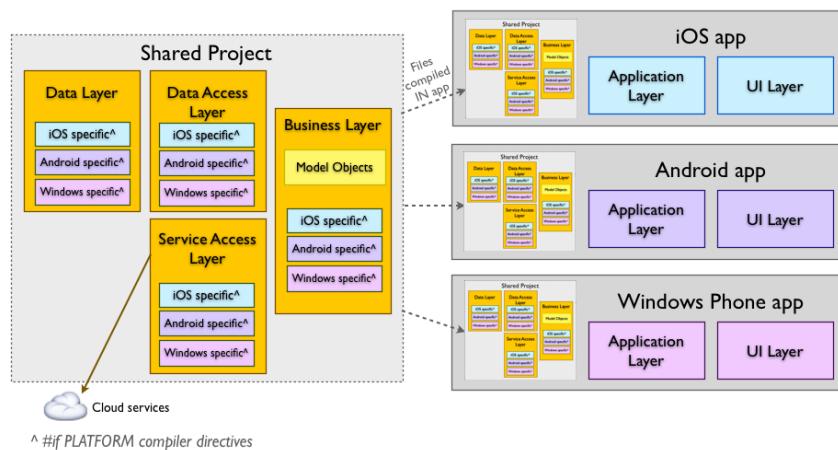


Figura 8 – Arquitetura conceptual do método Shared Project
Fonte: [35]

A alternativa à abordagem Shared Project é o uso da Portable Class Libraries [37]. Através do uso deste método é criada uma DLL que é restrita para executar para a plataforma específica para a qual foi criada. A criação desta biblioteca torna possível que outros projetos a referenciem e possibilita a criação de um projeto de testes unitários para se testar unicamente o projeto *portable*, sem que seja necessário ter em conta cada plataforma [38].

As soluções *portable* caracterizam-se por serem mais complexas em relação às soluções *shared*, uma vez que os projetos de cada plataforma têm códigos independentes e necessitam de um design mais bem estruturado, pois esta abordagem lida com interfaces e injeções de dependência (ver Figura 9) [36]. Além disso, com esta abordagem ainda não é possível usufruir de todos os recursos da plataforma .NET [37].

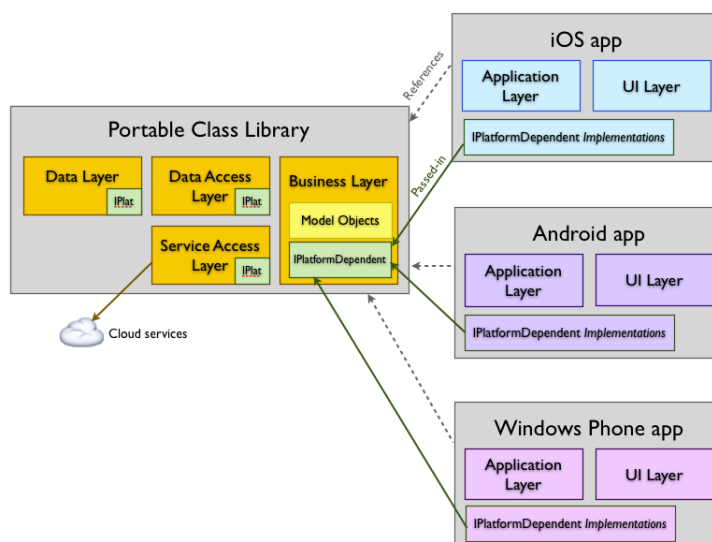


Figura 9 - Arquitetura conceptual do método Portable Class Library
 Fonte: [37]

Apesar deste projeto se tratar de uma única aplicação e não haver interesse em usar código noutros projetos, após a análise feita às duas alternativas, neste projeto optou-se pela abordagem Portable Class Libraries, uma vez que desta forma, evitou-se a implementação específica para cada plataforma [39]. Para além disso, possibilita a implementação de código mais estruturado. O mesmo código é utilizado em todas as aplicações.

Ainda assim, noutros projetos, poderá ser necessário fazer um novo estudo para a escolha da abordagem que melhor se adequa a este, visto que esta abordagem devido à sua versatilidade, o código é de fácil desenvolvimento, mas de difícil manutenção [40]. A abordagem *shared* torna-se insustentável quando estamos perante projetos de grandes dimensões e com equipas de desenvolvimento com vários programadores, enquanto a abordagem Portable Class Libraries pode ser preterida no caso de aplicações específicas para cada plataforma.

3.4.2 Apache Cordova

Em termos de arquitetura (Figura 10), uma aplicação desenvolvida em Apache Cordova, ao contrário do desenvolvimento em Xamarin, não apresenta alternativas significativas. Uma aplicação em Apache Cordova é composta por vários componentes [8].

O componente WebView [8] permite que o utilizador usufrua de toda a interface da aplicação.

O Web App [8] é o componente no qual podemos encontrar o código da aplicação desenvolvido (ficheiros CSS, JavaScript e HTML), recursos necessários (imagens, por exemplo) e um ficheiro config que disponibiliza informação e especificações sobre a aplicação.

Os Plugins [8] são parte integrante de uma aplicação em Cordova. Através destes é possível ter acesso a funcionalidades nativas como a bateria, camera, acelerómetro, entre outros.

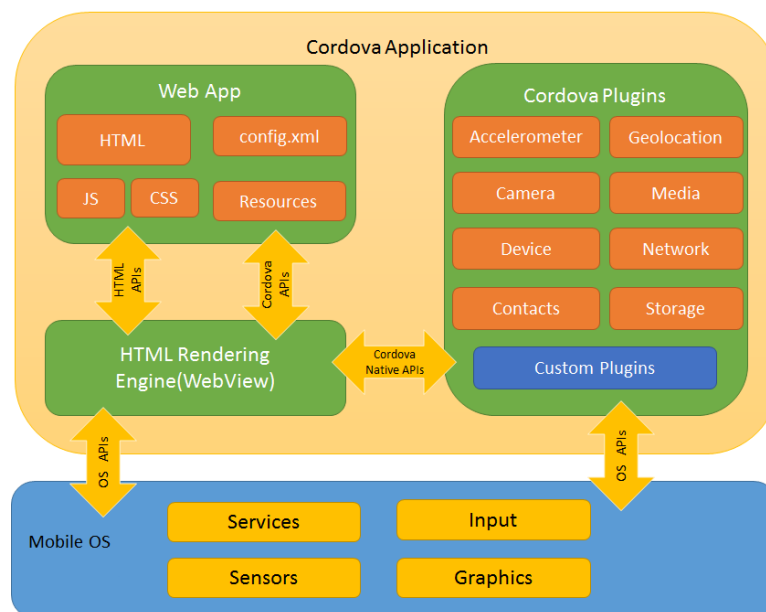


Figura 10 – Arquitetura conceitual do Apache Cordova
Fonte: [8]

Uma das questões relativas ao Apache Cordova é que este não fornece nenhuma biblioteca UI *widget* ou *MV* frameworks*. Por isso, caso se queira usar alguma destas tecnologias ter-se-á que as incluir na aplicação [8].

As *MV* frameworks* [41] permitem aos programadores implementarem padrões como o MVC e MVVM, enquanto as bibliotecas UI auxiliam o desenvolvimento da interface da aplicação. Como exemplos para *MV* frameworks* temos o TypeFramework [42], Backbone.js [43], entre outros. No caso de bibliotecas UI existem, por exemplo, o WinJS, Onsen UI e Ionic [44].

Tal como referido anteriormente, neste projeto utilizou-se o WinJS para o desenvolvimento da interface. Desta forma, todos os componentes da interface em HTML já estão criados sendo unicamente necessário alterar os CSS's caso queiramos alterar a parte estética desses componentes, como por exemplo cores, tipo de letra, entre outros.

4 Implementação da solução

Dado que este trabalho requereu o desenvolvimento de duas aplicações móveis através de ferramentas e linguagens distintas, torna-se necessário explicar algumas particularidades no que toca à implementação das mesmas.

De seguida abordar-se-á a comunicação das aplicações com os *webservices* existentes, desenvolvimento das interfaces, como se fez o tratamento de erros e a importância dos *Bindings* nas aplicações.

4.1 WebServices

Tal como mencionado anteriormente, os *webservices* consumidos pelas aplicações já estavam implementados, sendo unicamente necessário ter acesso aos WSDL's⁵ dos serviços.

Sendo imprescindível o acesso à informação de cada utilizador para efetuar login, obter as notícias e horário, entre outros, foi necessário implementar as aplicações a partir destes serviços baseados em *SOAP*⁶.

Desta forma, o primeiro passo foi estabelecer a comunicação entre as aplicações e estes serviços.

No caso do projeto em Xamarin, foi necessário adicionar uma referência web através dos WSDL's e desta forma foi gerada uma classe com todos os métodos disponíveis no serviço específico. Por exemplo, através do serviço *Session* é possível efetuar o login e logout. Após a criação da referência web foi gerada uma classe *ISession*. Para que fosse possível aceder aos métodos login e logout desse serviço foi necessário instanciar um objeto da classe *ISession*. De

⁵ Web Services Description Language (WSDL) descreve em formato XML a informação sobre um determinado serviço no protocolo SOAP.

⁶ SOAP é um protocolo baseado em XML que permite a comunicação entre aplicações e/ou serviços.

notar que ambos os métodos são assíncronos permitindo à aplicação realizar outros processos enquanto não for recebida a resposta ao pedido efetuado (ver Figura 11).

Após a obtenção da resposta foi necessário proceder à análise sintática e criação do modelo de dados de acordo com a informação recebida.

```
2 references
public System.Threading.Tasks.Task<string> LoginAsync(string user, string pwd)
{
    return base.Channel.LoginAsync(user, pwd);
}

2 references
public System.Threading.Tasks.Task<string> LogoutAsync(string session_guid, string session_code)
{
    return base.Channel.LogoutAsync(session_guid, session_code);
}
```

Figura 11 – Métodos login e logout do serviço Session (Xamarin)

Por outro lado, apesar dos princípios serem idênticos, a implementação da comunicação dos serviços com a aplicação em Apache Cordova foi mais complexa.

De salientar que em *JavaScript* o assincronismo é feito através de *callbacks*. As *callbacks* são porções de código que são executadas quando é recebida uma determinada mensagem. Isto é, durante a execução de um programa, a *callback* fica a aguardar por uma determinada mensagem para que possa também executar o seu código.

Posto isto, para se efetuar o login na aplicação em Apache Cordova foi necessário criar uma função (*callback*) que ficava a aguardar pela mensagem “webserviceLogin” para ser executada (ver Figura 12).

```
$(document).on("webserviceLogin", function (responseJson: any, result: string, hashResult: string, codeResult: string): void {
    if (result == "ok") {
        session._Hash = hashResult;
        session._Code = codeResult;
    } else {
        document.getElementById('errorMessage').innerHTML = hashResult;
        var errorDialog = document.getElementById("errorDialog").winControl;
        errorDialog.show();
        return;
    }
});
```

Figura 12 – *Callback* login do Serviço Session (Apache Cordova)

Esta *callback* aguarda que a resposta seja recebida e dependendo se o resultado contiver algum erro ou não, procede-se à criação do modelo de dados da sessão.

Com a *callback* já criada, procedeu-se à implementação do pedido HTTP. Para tal, foi necessário criar o envelope SOAP com a informação devida (ver Figura 13), enviar o pedido, e criar uma função para receber e tratar a informação da resposta recebida (ver Figura 14).

```
var sr =
    '<?xml version="1.0" encoding="utf-8"?>\n' +
    '<soapenv:Envelope ' +
    'xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ' +
    'xmlns:api="http://127.0.0.1/Integratics/Enswitch/API" ' +
    'xmlns:xsd="http://www.w3.org/2001/XMLSchema" ' +
    'xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">\n' +
    '<soapenv:Body>\n' +
    '<Login xmlns="http://portal.isep.ipp.pt/">\n' +
    '<user xsi:type="xsd:string">' + user + '</user>\n' +
    '<pwd xsi:type="xsd:string">' + passwd + '</pwd>\n' +
    '</Login>\n' +
    '</soapenv:Body>\n' +
    '</soapenv:Envelope>';
```

Figura 13 – Envelope login (Apache Cordova)

```
xmlhttp.onreadystatechange = function (): void {
    if (xmlhttp.readyState == 4) {
        if (xmlhttp.status == 200) {
            var response: string = xmlhttp.response;
            var parser = new DOMParser();
            var xmlDoc = parser.parseFromString(response, "text/xml");
            var responseString = xmlDoc.getElementsByTagName("LoginResult")[0].innerHTML;
            var responseJson = JSON.parse(responseString);
            $(document).trigger("webserviceLogin", responseJson);
        }
    }
}
```

Figura 14 – Função que permite receber resposta do pedido login

Através da Figura 14 é possível observar que a função “trigger” permite que a nossa *callback* possa ser executada visto que a mensagem “webserviceLogin” foi enviada. Para além disso, a *callback* receberá também a resposta já tratada através da variável “responseJson”.

4.2 Interface

Dado que para o desenvolvimento da interface no projeto Xamarin recorreu-se ao Xamarin.Forms, foi possível criar uma interface nativa para cada plataforma. Por exemplo, na criação de um botão em Xamarin.Forms, no momento em que o código é compilado para uma plataforma específica, esse botão é transformado para um botão nativo.

Assim sendo, o Xamarin.Forms fornece diversas ferramentas para a criação de uma interface, nomeadamente no tipo de páginas que pretendemos, por exemplo: MasterDetailPage, NavigationPage, CarouselPage, entre outros.

A escolha do tipo de página depende do gosto do cliente/programador e como tal, para este projeto optou-se pela *NavigationPage*. Este tipo de página gere a navegação entre páginas através de uma pilha de outras páginas. Uma das vantagens do uso deste tipo de página é a total adaptação da opção retroceder (*backbutton*), quer em termos de interface (gestos e botões), quer no botão físico do dispositivo (ver Figura 15). No entanto, é necessário ter em atenção à gestão da pilha das páginas.

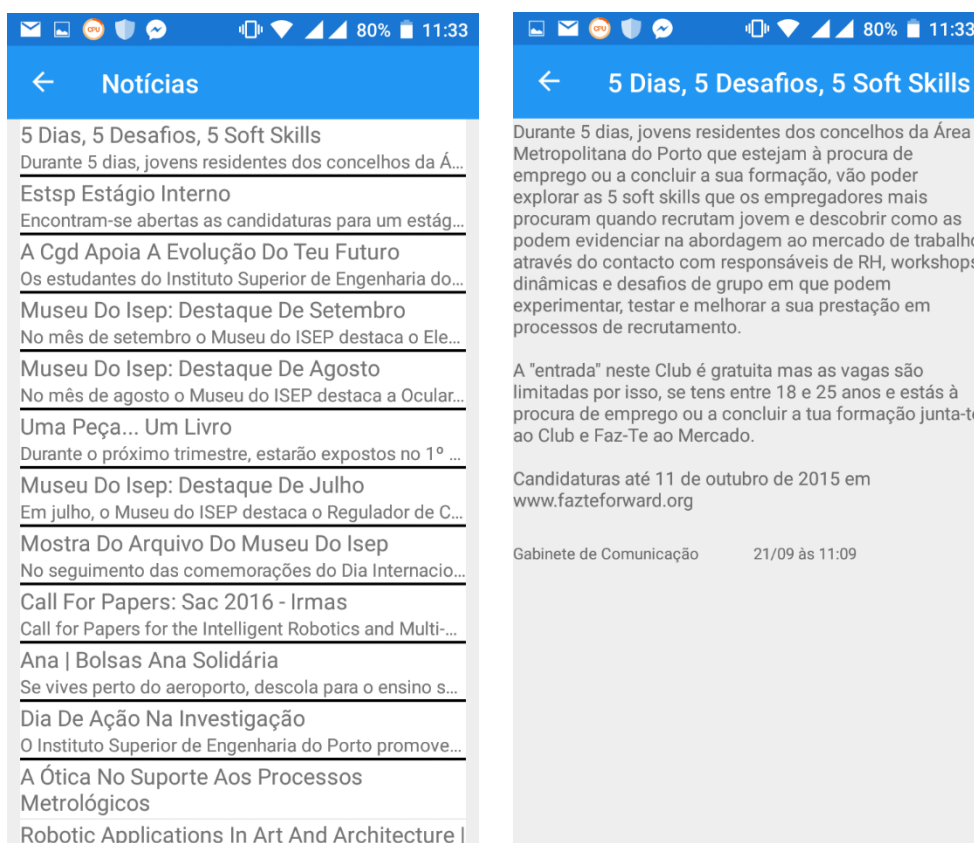


Figura 15 – Interface módulo notícias em Android (Xamarin)

No projeto Apache Cordova, o desenvolvimento da interface depende das ferramentas escolhidas para o efeito. Neste caso, optou-se pela biblioteca WinJS que fornece os controlos necessário para o desenvolvimento de uma interface, por exemplo: *listview*, *toolbar*, *flipview*, entre outros.

Foi através da criação de uma *splitview* que se desenvolveu a interface desta aplicação. O utilizador através do gesto de pressionar no botão hambúrguer, poderá escolher a página pretendida através do menu de opções aberto (ver Figura 16).

Importa referir que nesta aplicação, a interface desenvolvida não é adaptada ao ambiente onde está a ser executada. A aplicação em Android terá a mesma interface que na versão iOS, por exemplo.

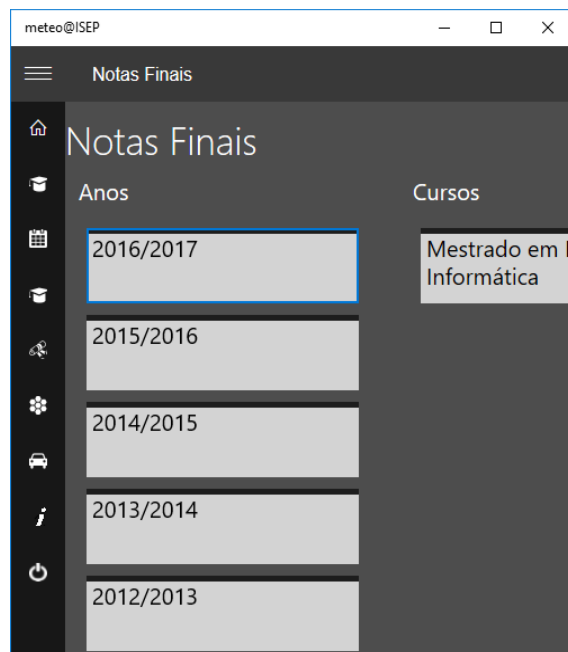


Figura 16 – Interface módulo Notas Finais em Windows (Apache Cordova)

4.3 Bindings

Os *Bindings* são uma ferramenta essencial na implementação do padrão MVVM uma vez que permitem que a informação da *ViewModel* possa ser apresentada na *View* (ver Figura 5). Isto facilita a equipa de desenvolvimento na divisão de tarefas dado que a implementação de ambas as camadas é praticamente independente. A *View* só necessita de saber o nome da propriedade ou comando da *ViewModel*, não tendo qualquer conhecimento do que está implementado nessa camada.

Para tal, em Xamarin, na implementação dos *Bindings* é necessário ter em atenção a alguns aspetos:

- É necessário indicar o *BindingContext* da *View*, sendo que este é a instância do objeto da classe *ViewModel* (ver Figura 17). Por outro lado, para que todo o sistema funcione, é necessário que a classe *ViewModel* implemente a interface *INotifyPropertyChanged*. Através desta interface, quer a *ViewModel*, quer a *View* sabem quando uma propriedade ou comando foi alterado ou acionado durante a execução da aplicação.

```
if (BindingContext == null)
    BindingContext = new LoginViewModel();
```

Figura 17 – *BindingContext* da classe LoginView (Xamarin)

- Existem três modos de *Bindings*: *OneWay*, *OneWayToSource* e *TwoWay*. O *Binding* do modo *OneWay* indica que a comunicação é unidirecional (da *ViewModel* para a *View*).

O modo *OneWayToSource* é o inverso da anterior (da *View* para a *ViewModel*).

Enquanto o modo *TwoWay* permite uma troca de informação bidirecional.

- As *ListView*'s que são muito usadas em aplicações móveis para listar informação, tem a particularidade de implementarem um *Binding* específico. '

Por exemplo, na classe *NewsViewModel* é necessário criar uma propriedade que contenha as notícias que serão apresentadas ao utilizador. Esta propriedade é do tipo *ObservableCollection* e cada item dessa coleção corresponde a uma *NewsModel*.

Desta forma, já é possível fazer o *Binding* à coleção para preencher a *ListView*. Como podemos observar na Figura 18, o *Binding* é feito à propriedade "News" que está implementado na *NewsViewModel* e através de *Binding* também é possível indicar à mesma qual a notícia que o utilizador escolheu ("SelectedItem").

```
<ListView x:Name="listview" ItemsSource="{Binding News}" SelectedItem="{Binding NewsSelected, Mode=TwoWay}">
```

Figura 18 – *ListView* Notícias (Xamarin)

- É também possível fazer *Binding* a *Value Converters*. Os *Converters* são um recurso criado para executar determinadas tarefas que são da responsabilidade da *View*. Por exemplo, uma das regras de negócio desta aplicação determina que o *username* do utilizador seja o número mecanográfico (alunos) ou a sigla em letra maiúscula (professores e funcionários). De modo a evitar que o utilizador seja obrigado a escrever a sigla em letras maiúsculas, criou-se um converter que transforma automaticamente a sigla para o estado pretendido sem que o utilizador tenha necessidade de efetuar qualquer alteração.

Tal como referido anteriormente, o desenvolvimento de aplicações em Apache Cordova está totalmente dependente das ferramentas que se acrescentam ao projeto.

Uma vez que se recorreu à biblioteca WinJS, a implementação de *bindings* foi mais limitado. Todos os comandos corresponderam a eventos e toda a informação foi passada pelos mesmos. Por exemplo, no módulo login, quando o utilizador pressiona o botão "Login" é disparado um evento "onclick" e na implementação deste evento obtém-se o texto do *username* e *password*.

No entanto, as *ListView*'s em Apache Cordova também têm uma implementação particular. Tal como em Xamarin, é necessário criar uma propriedade com todas as notícias que neste caso corresponde a um *Array* de Notícias. Tendo esse *array* preenchido ("parent.news"), procede-se há conversão desse *array* para um "WinJS.Binding.List" ("dataListNews"). Após a conversão, determina-se que os itens da *ListView* da interface ("listnews") é igual aos itens da lista convertida (ver Figura 19).

```

function fillList() {
    dataListNews = new WinJS.Binding.List(parent.news);
    listnews.itemDataSource = dataListNews.dataSource;
    WinJS.UI.processAll();
}

```

Figura 19 – Conversão de *array* para WinJS.Binding.List (Apache Cordova)

Após a criação da *ListView* em HTML, tal como demonstra a Figura 20, é através de um *template* que poderemos listar a informação pretendida. Por exemplo, no caso da listagem de notícias só pretendemos mostrar o título e parte do corpo da mesma. Então procede-se à criação do “smallListIconTextTemplate” e dado que cada item corresponde a uma notícia, basta indicar que o texto é o atributo “title” ou “body”.

```

<div id="listDivNews"
    data-win-control="WinJS.UI.ListView"
    data-win-options="{
itemTemplate: select('.smallListIconTextTemplate'),
layout: { type: WinJS.UI.ListLayout },
selectionMode: 'single',
tapBehavior: 'directSelect',
oniteminvoked : handler,
}">

```

Figura 20 – *ListView* Notícias (Apache Cordova)

5 Avaliação

Tal como referido anteriormente, desenvolveram-se dois projetos distintos nas tecnologias Apache Cordova e Xamarin (ver Secção 2.3.4). Estes projetos tinham como principal objetivo a implementação de uma aplicação móvel sobre o Portal do ISEP.

Dado que a avaliação é essencial para que o autor perceba as limitações e vantagens da aplicação, procedeu-se à realização de testes de usabilidade com utilizadores.

5.1 Avaliação de usabilidade com utilizadores

Segundo Rubin [45] existem quatro tipos de testes: teste exploratório, teste de avaliação, testes de validação e o teste comparativo.

Apesar de se pretender comparar dois protótipos distintos, estes testes consistiram na realização de testes de validação, pois tinham como principal objetivo verificar qual o produto que correspondia às expectativas e como é que cada produto se comportou como um todo. Esta avaliação, para além de permitir a definição dos vários perfis de cada utilizador e avaliar os módulos implementados, possibilitou também perceber qual das aplicações os utilizadores preferiam em termos de interface, funcionamento e adaptação à mesma.

Atendendo à necessidade de uma avaliação eficaz e atendendo às estratégias adotadas, foi necessário pedir a colaboração de alunos, funcionários e professores do ISEP, pois para além de serem o público alvo deste trabalho, são os utilizadores efetivos do Portal da instituição.

Esta avaliação de usabilidade foi composta por dois momentos distintos, mas complementares. O primeiro foi a realização de um inquérito sobre questões gerais do utilizador e a aplicação existente do Portal; no segundo momento realizaram-se os testes aos dois protótipos desenvolvidos e finalização do inquérito.

Os testes foram realizados em condições idênticas àquelas nas quais as aplicações se iriam utilizar, isto é, os utilizadores testaram a aplicação num smartphone com o mesmo sistema operativo do seu telemóvel, e testaram as funcionalidades do login, notícias, informação do utilizador e Logout.

5.1.1 Metodologia

O primeiro passo do processo foi a elaboração de um questionário que fosse objetivo e de resposta rápida, estando dividido em três secções (ver Anexo 1). A primeira secção permitiu ao investigador traçar os vários perfis dos inquiridos através das questões habituais tais como a idade e o género. A segunda caracterizava-se por apresentar questões gerais sobre a aplicação do ISEP, conhecimento da aplicação existente e da satisfação com a mesma, por exemplo. Na última secção, após os testes aos protótipos, os inquiridos poderiam expressar a sua preferência e partilhar sugestões de melhoria para cada aplicação. O questionário foi disponibilizado na plataforma Google Forms e foi respondido na presença do investigador.

Para que os testes com os utilizadores fossem realizados com sucesso foi necessário instalar a aplicação Xamarin em dois smartphones, Android e Windows Phone e instalar a aplicação Apache Cordova no computador do investigador.

Procedeu-se então ao recrutamento presencial e contacto direto na cantina do ISEP para o caso dos alunos; nos gabinetes dos professores para os mesmos; e no departamento de relações externas do ISEP para os funcionários.

Após a primeira abordagem e os voluntários disponíveis para responder ao questionário, seguiu-se uma breve explicação sobre os objetivos e propósitos do estudo. Também lhes foram explicadas quais as etapas que deveriam seguir e foi-lhes ainda dito que, apesar da existência de um questionário, haveria um espaço para que cada um pudesse ter a liberdade de expressar a sua própria opinião.

Terminadas as respostas às duas primeiras secções do questionário, seguiu-se o teste da Aplicação A (Xamarin) no *smartphone* com o sistema operativo correspondente ao telemóvel do utilizador. Pediu-se então que testasse a funcionalidade login, consultasse uma notícia à sua escolha, visse a informação pessoal e que terminasse sessão.

Depois de terminado o teste à primeira aplicação solicitou-se ao utilizador que realizasse os passos anteriores no teste à Aplicação B (Apache Cordova).

Por fim, o utilizador respondeu à terceira e última parte do questionário, na qual pôde partilhar dúvidas e opiniões relativas aos testes realizados.

Na análise estatística foram utilizadas medidas de estatística descritiva (frequências absolutas e relativas, médias e respetivos desvios-padrão) e estatística inferencial. O nível de significância foi fixado em $\alpha \leq .05$ [46]. Utilizou-se o teste Q de Cochran uma vez que se pretendia comparar três variáveis do tipo nominal (1- Sim; 2- Não) e comparar a razão da preferência de uma

determinada aplicação escolhida (Facilidade na obtenção da informação; Melhor Design; Rapidez do sistema, na Aplicação A e na Aplicação B).

A “facilidade na obtenção da informação” determina se é simples alcançar uma determinada página, por exemplo, se o utilizador quiser consultar a página de notícias não terá de passar por demasiadas páginas. Por “melhor design” trata-se do aspeto visual da aplicação como por exemplo, cores e ícones. “Rapidez do sistema” permite perceber se a aplicação é mais lenta ou mais rápida numa determinada funcionalidade.

O tratamento dos dados foi efetuado no programa SPSS (*Statistical Package for the Social Sciences*) versão 24.0 para Windows.

5.1.2 Perfil da amostra

Os utilizadores que realizaram os testes aos dois protótipos foram alunos, funcionários e professores do ISEP. Estipulou-se que os utilizadores deveriam ter idades compreendidas entre os 18 e os 65 anos e ter acesso ao portal da instituição. Apesar dos protótipos estarem unicamente em português, aceitaram-se utilizadores de outras nacionalidades sendo que o investigador efetuou a tradução dos mesmos para a língua inglesa. As pessoas que tivessem o sistema IOS no seu telemóvel foram excluídas deste inquérito visto que uma das aplicações não estava disponível neste sistema operativo.

Neste trabalho a amostra foi constituída por 36 participantes sendo que a idade mínima foi de 19 anos e a máxima de 53 anos (M=27.03, DP = 9.55). No que se refere ao género os valores são iguais (ver Tabela 24)⁷.

Maioritariamente são estudantes (N=29, 80.6%) e os inquiridos que utilizam o sistema operativo Android são 30 (83.3%) e 6 (16.7%) utilizam o sistema operativo Windows Phone⁷.

Tabela 24 – Caracterização dos participantes da amostra (N=36)

Variáveis	Valores correspondentes	N(%)
Género	1 - Masculino	18(50)
	2 - Feminino	18(50)
Estatuto	1 - Estudante	29(80.6)
	2 - Professor	4(11.1)
	3 - Funcionário	3(8.3)

5.1.3 Análise dos resultados

Na análise dos resultados dos inquéritos, constatou-se que todos os participantes são a favor da existência de uma aplicação que facilite o acesso a diversas funcionalidades do portal do

⁷ Para ver os dados estatísticos consulte o Anexo 2.

ISEP. Contudo, nem todos os utilizadores tinham conhecimento da existência da aplicação do portal, sendo que 33 utilizadores (91.7%) tinham conhecimento⁷.

De notar que os módulos que os utilizadores julgam ser imprescindíveis na aplicação, são o módulo horário (N=34, 94.4%), notas parciais (N=26, 72.2%), notas finais (N=28, 77.8%) e salas disponíveis (N=25, 69.4%). Apesar do valor estatístico da opção “outras” ser estatisticamente insignificante, através desta, foi possível recolher novas ideias para implementar no futuro, como por exemplo: entradas nas salas de aula com o telemóvel, pedidos de certificados, consultar faltas, marcação de férias, entre outras⁷.

No que se refere à frequência do uso da aplicação existente, verifica-se que a maioria dos participantes nunca a utiliza. Dos inquiridos, só 10 usam a aplicação de uma a cinco vezes por semana (27.8%) e só 5 a utilizam diariamente⁷.

Dos participantes que nunca utilizam a aplicação, 11 (30.6%) não têm opinião no que toca à sua satisfação. Durante os testes foi possível perceber que a maioria destes 11 utilizadores nunca instalou a aplicação⁷.

Dos que responderam que nunca utilizam a aplicação e não estão totalmente satisfeitos com a mesma, concluiu-se que 5 dos utilizadores tinham problemas de funcionamento (23.8%) e 7 (33.3%) mencionaram outros problemas. Daqueles que utilizam a aplicação, 11 (73,3%) referem problemas de funcionamento e 6 (40%) referem outros motivos⁸.

Foram apontados diversos motivos para a não total satisfação com a aplicação existente: “acesso e carregamento de informação lentos”, “problemas na memorização das credenciais”, “poucas funcionalidades”, “muitas vezes não funciona, falta de memória no telemóvel”, “não apresenta as notas corretamente”, “as salas vazias nem sempre estão disponíveis para estudar” e “não tem grande utilidade”⁸.

Dos 36 participantes, 18 (50%) preferem a aplicação A, 16 (44.4%) a aplicação B e 2 (5.6%) não têm preferência, sendo que na maioria dizem que preferem a aplicação devido ao melhor design, à facilidade na obtenção da informação e à rapidez do sistema. No entanto, observou-se na aplicação B que 1 participante (2.8%) referiu que a aplicação B deixou de funcionar⁸.

Relativamente à Aplicação A, verificou-se que 50% dos participantes (N=14) escolheram esta aplicação devido à facilidade na obtenção da informação. As diferenças observadas entre as percentagens de respostas são estatisticamente significativas ($Q(2)=6.118$; $p=.047$, $N=18$). Por outro lado, na Aplicação B observou-se que a opção “Melhor Design” foi a razão atribuída com maior frequência de respostas (N= 11, 31% das respostas). As diferenças observadas entre as percentagens de respostas também são estatisticamente significativas ($Q(2)=10.714$; $p=.005$, $N=16$). Na aplicação A, a opção “Melhor Design” apresentou uma frequência mais elevada do que o “Sistema mais rápido” e na aplicação B as diferenças demonstráveis “Melhor design” foi superior à “Facilidade na obtenção da informação” seguida de “Sistema mais rápido”⁸.

⁸ Para ver os dados estatísticos consulte o Anexo 2.

Importa também analisar as duas últimas questões do inquérito, uma vez que eram questões abertas para o utilizador sugerir melhorias às aplicações. No caso da aplicação A sugeriram que o tamanho de letra fosse aumentado ou que fosse criada a funcionalidade do *zoom*; o *layout* com cores relacionadas ao ISEP; fossem inseridos ícones adequados a cada módulo; inserir a imagem associada à notícia; melhorar o menu principal; desenvolver mais módulos; disponibilizar a aplicação em inglês; destaque entre as diferentes notícias (com cores, por exemplo); e ordenar o menu por ordem de frequência de consulta. Para a aplicação B sugeriram fazer o *decode* do texto da notícia; menu com mais ícones; títulos mais apelativos; carregamento mais otimizado; facilidade de acesso mais explícita; criação de uma página de login; informação visual do *loading*; simplificação da interface; guardar credenciais de acesso; e ordenar o menu por ordem de frequência.

5.2 Avaliação ao desenvolvimento

Em termos de avaliação ao desenvolvimento realizou-se uma análise comparativa após o desenvolvimento dos dois protótipos da aplicação. Foram avaliadas as dificuldades obtidas, limitações de ambas as tecnologias, vantagens e desvantagens, foi feita uma comparação em termos de tempo para o desenvolvimento das várias funcionalidades, entre outras.

Ao longo do desenvolvimento das aplicações foi necessário recolher informações segundo os critérios definidos anteriormente.

De seguida serão então apresentados esses critérios e a sua análise fundamentada com base na experiência prática.

Em termos de linguagens de programação o foco não se centrou na comparação do C# (Xamarin) e JavaScript (Apache Cordova). Pois a escolha depende da experiência e preferência do programador com ambas as linguagens.

No que toca ao desenvolvimento da interface, constatou-se que o desenvolvimento em HTML e CSS (Apache Cordova) é mais simples e rápido do que o XAML (Xamarin). Para além disso, o facto de existirem bibliotecas e *frameworks* em JavaScript, HTML e CSS, permite evitar a criação de uma interface de raiz, algo que não se verifica em XAML. Neste projeto, o WinJS foi a biblioteca escolhida para o desenvolvimento da interface no projeto Apache Cordova e através da consulta do site⁹ da mesma foi possível testar e retirar todos os componentes necessários para a construção da interface: *listview*, *splitview*, entre outros. Isto possibilita ao programador ou ao responsável pela interface, preocupar-se unicamente com a adaptação dos CSS's se necessário. Claro que estes tipos de bibliotecas acarretam outras desvantagens que podem variar consoante a escolha da mesma. Neste caso, uma das desvantagens encontradas no WinJS foi o facto desta tecnologia não suportar o *Two way Data Binding*. Esta característica permite

⁹ Site do WinJS - <http://try.buildwinjs.com/playground/>

que qualquer alteração num componente da interface se reflita noutra componente no código JavaScript da aplicação.

Apesar do desenvolvimento da interface em Apache Cordova ser intuitivo através do uso de Bibliotecas ou Frameworks, o Xamarin permite criar uma interface nativa individual para cada sistema operativo. Outra possibilidade é o uso de Xamarin.Forms no qual é possível criar páginas e navegação entre elas que depois serão compilados para as várias plataformas, adaptando-se ao sistema operativo correspondente. Isto é, as páginas do projeto Xamarin deste trabalho correspondem NavigationPages simplesmente por preferência do aluno. Poder-se-ia escolher outra alternativa, como por exemplo a MasterDetailPage ou TabbedPage, entre outras. Com estas páginas, o programador tem a página adaptada ao sistema operativo e, deste modo, através de código em C#, tem controlo sobre a navegação nas mesmas. Caso o utilizador pressione o botão de retroceder no hardware, por exemplo, a aplicação já sabe que comportamento deve ter. Desta forma, o projeto Xamarin apresenta uma interface mais responsiva.

O suporte às tecnologias também é um pouco diferente. Apesar da comunidade *online* em Xamarin ser boa e ainda estar em crescimento, notou-se que o Apache Cordova tem uma comunidade mais ativa e maior em comparação com a tecnologia anterior. Provavelmente isto deve-se ao facto do Xamarin ser uma tecnologia mais recente e ainda estar a entrar no mercado do desenvolvimento *mobile*.

Apesar do suporte ao Apache Cordova ser mais estável, os erros de compilação nesta tecnologia são muitas vezes ambíguos. Para o mesmo erro é possível encontrar várias soluções na internet sendo elas totalmente distintas e muitas vezes não solucionarem o problema.

Outro aspeto a ter em conta é o acesso à API do dispositivo visto que em Xamarin o acesso é total, enquanto em Apache Cordova existe a dependência do uso de plugins para o mesmo efeito. O facto de haver estas dependências pode levar ao aumento significativo de erros na aplicação dependendo do dispositivo utilizado.

Uma vantagem muito relevante em Xamarin é o suporte aos padrões de arquitetura como o MVC e MVVM (ver Secção 3.1). Para além de respeitar a boa prática de desenvolvimento de software, possibilita a reutilização de cerca de 80% do código entre várias aplicações. O uso destes padrões aumenta a complexidade do desenvolvimento de aplicações principalmente para quem não esteja familiarizado com estes. No caso do Apache Cordova, o suporte a estes padrões de arquitetura de software é possível através do uso de frameworks externas à tecnologia.

Relativamente à performance das aplicações, apesar de não ter sido feito qualquer teste específico para avaliar esta característica, foi possível observar que ambas as tecnologias apresentam performances diferentes. Rösler [47] já defendia que as aplicações nativas apresentavam uma performance maior que as aplicações híbridas. Neste trabalho, foi possível observar que a aplicação em Apache Cordova apresentava uma performance mais reduzida em relação ao Xamarin principalmente na apresentação de uma grande quantidade de informação

nas listas, devido à lenta renderização das mesmas. O facto do Xamarin suportar multithreading poderá ter influência neste aspeto enquanto que o multithreading ou a execução paralela em Apache Cordova é suportado por plugins.

No que diz respeito ao tempo de desenvolvimento das funcionalidades para ambas as aplicações, pelas razões evidenciadas anteriormente, a aplicação Apache Cordova foi desenvolvida mais rapidamente. Este aspeto prende-se essencialmente com o acesso a ferramentas externas à aplicação que facilitam o desenvolvimento da mesma.

Tendo em conta que a aplicação em Apache Cordova é a mesma para todas as plataformas, testando num qualquer smartphone é possível perceber como se comporta a aplicação. No entanto, dado que a aplicação varia dependendo do sistema operativo, no caso do Xamarin, se se quiser testar a aplicação em IOS, é necessário um Mac para se poder fazer o *deploy* para um dispositivo IOS.

Ainda que no desenvolvimento deste projeto não se tenha verificado este problema, caso uma empresa pretenda desenvolver uma aplicação em Xamarin, terá um custo associado. No caso do Apache Cordova, o desenvolvimento é totalmente gratuito.

5.3 Métricas de desenvolvimento

As métricas de desenvolvimento são ferramentas que promovendo a melhoria continua de um projeto e auxiliando as equipas de desenvolvimento na tomada de decisões estratégicas, permitem que sejam atingidos os resultados esperados [48].

Usadas na indústria de desenvolvimento de *software*, estas ferramentas possibilitam um estudo aprofundado sobre novas abordagens e que tecnologias adotar, acompanhamento e aperfeiçoamento do processo de produção, avaliação de investimentos e retornos, e com base em factos concretos melhoraram a gestão do projeto. Desta forma torna-se tangível a garantia de qualidade de um produto ou aplicação [49].

Existem duas categorias de métricas: diretas e indiretas. As métricas diretas baseiam-se num único atributo. Um atributo é uma propriedade física ou abstrata de uma entidade. Alguns exemplos de métricas diretas são: custo de implementação, número de linhas de código, esforço, entre outros; as métricas indiretas referem-se a medidas baseadas em n atributos, como por exemplo, a produtividade de um programador que é a relação entre tamanho de código produzido e o tempo de desenvolvimento [48].

Neste projeto recorreu-se à análise de métricas diretas através da medição do tamanho do produto desenvolvido. Geralmente, o tamanho do produto está inteiramente relacionado com o esforço necessário para produzi-lo, e por norma é usado na fase de planeamento, acompanhamento e na definição de estimativas.

Para medir o tamanho do produto, a contagem de linhas de código (LOC) e a análise de pontos de função (APF) são as medidas mais aplicadas. Contudo, existem algumas particularidades que devem ser tidas em conta aquando da decisão da métrica a utilizar [48].

Tal como o nome indica, a contagem de linhas de código assenta na contagem de instruções existentes no código fonte do produto. A análise de pontos de função avalia o tamanho do produto com base nas funcionalidades que são efetivamente entregues ao utilizador final, independentemente da metodologia e/ou ferramentas utilizadas.

No primeiro caso, LOC, é baseada em elementos concretos e a recolha da informação é bastante simples, tornando a sua automatização facilmente implementada. Para além disso, está indubitavelmente associada ao esforço de desenvolvimento [50].

Ao contrário da LOC, a APF pode ser realizada em qualquer fase do desenvolvimento do produto, visto que está relacionada com a especificação dos requisitos funcionais. No entanto, esta métrica é baseada numa avaliação a diversos elementos que são relativos e dependentes de vários contextos.

A principal razão pela escolha da análise de pontos de função como métrica de desenvolvimento deste projeto, prende-se com o facto de ser independente à linguagem de programação adotada, permitindo a comparação com projetos desenvolvidos em tecnologias distintas. A contagem de linhas de código é totalmente dependente da linguagem de programação, por exemplo, por norma um projeto implementado em C terá mais linhas de código que o mesmo projeto implementado em C#, logo o esforço aplicado no projeto em C será maior. Assim sendo, torna-se inviável usar este método para comparação de tecnologias em linguagens distintas [48].

Dado que o principal objetivo deste trabalho é a implementação de uma aplicação em duas tecnologias diferentes e comparação das mesmas, a métrica APF é a mais indicada. Posto isto, de seguida será demonstrada a aplicabilidade da APF neste caso prático.

5.3.1 Análise dos pontos de função

O principal objetivo desta análise foi fazer uma previsão do custo, quer em termos monetários, quer em tempo de desenvolvimento projeto, medidas que podem determinar o tamanho de uma aplicação [51].

Tal como indicado anteriormente, a análise de pontos de função baseia-se em funções executadas pela aplicação do ponto de vista do utilizador e parte de cinco premissas [51]:

- Entrada Externa (EE)
Corresponde ao processamento de dados vindos de fora da aplicação, ou seja, representam o fluxo de informações originadas pelo utilizador e/ou outros sistemas.
- Saída Externa (SE)

Ao invés da entrada externa, este caso representa todos os processos de geração de dados e/ou informações de controlo que saem da aplicação. As saídas de dados apresentam a informação ao utilizador com base no processamento da mesma.

- Arquivo Lógico Interno (ALI)
Grupo lógico de dados mantida internamente na aplicação e cuja a sua manutenção é feita dentro da mesma.
- Arquivo de interface externa (AIE)
Grupo lógico de dados armazenados e mantidos fora da aplicação. Muitas vezes associados a *webservices*.
- Consulta externa (CE)
Representa o processamento de consulta de informação da aplicação, mas que não inclui qualquer processamento de dados.

Procede-se então à identificação dos três passos a ter em conta na análise de pontos de função [51]:

1. Identificar os requisitos funcionais da aplicação e seu modelo de dados.
2. Classificar cada um dos requisitos dependendo no seu nível de complexidade
3. Ajustar o número de pontos de função brutos ao nível de complexidade de processamento.

Passo 1:

Após a explanação dos casos de uso deste projeto (ver secção 3.2), é necessário identificar os requisitos funcionais da aplicação. Assim sendo, foram identificados os seguintes requisitos funcionais:

RF1 - O sistema permite efetuar *login* com a conta associada ao portal do ISEP.

RF2 - O sistema permite consultar lista de notícias.

RF3 - O sistema permite consultar uma notícia específica.

RF4 - O sistema permite consultar informação pessoal.

RF5 - O sistema permite fazer *logout*.

De seguida é apresentado o modelo de dados que terá um papel preponderante na classificação de cada requisito funcional (ver Figura 21).

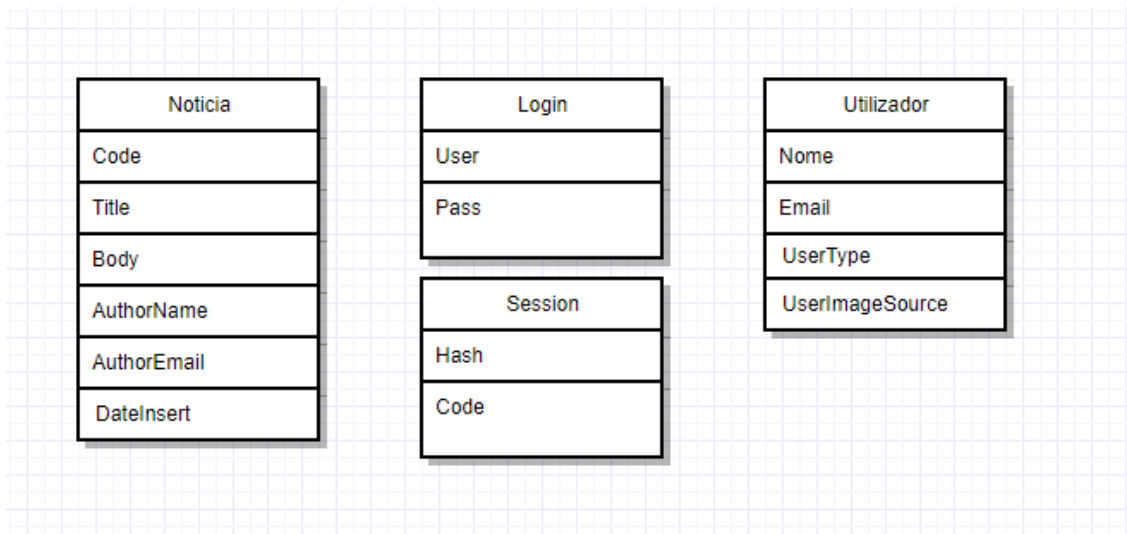


Figura 21 – Entidades implementadas

Passo 2:

Com base nas seguintes tabelas que representam o nível de complexidade associado a cada premissa, será possível determinar o nível de complexidade (simples, médio e complexo) para cada requisito funcional [52].

Tabela 25 – Nível de complexidade AIE

Dados Referenciados	1 a 19 itens	20 a 50 itens	51 ou mais itens
Registo Lógico			
1 tipo	Simples	Simples	Médio
2 a 5 tipos	Simples	Médio	Complexo
6 ou mais tipos	Médio	Complexo	Complexo

Tabela 26 – Nível de complexidade ALI

Dados Referenciados	1 a 19 itens	20 a 50 itens	51 ou mais itens
Registo Lógico			
1 tipo	Simples	Simples	Médio
2 a 5 tipos	Simples	Médio	Complexo
6 ou mais tipos	Médio	Complexo	Complexo

Tabela 27 – Nível de complexidade CE

Arquivo referenciado	Dados Referenciados	1 a 5 itens	6 a 19 itens	20 ou mais itens
0 ou 1 tipo		Simple	Simple	Médio
2 ou 3 tipos		Simple	Médio	Complexo
4 ou mais tipos		Médio	Complexo	Complexo

Tabela 28 – Nível de complexidade EE

Arquivo referenciado	Dados Referenciados	1 a 4 itens	5 a 15 itens	16 ou mais itens
0 ou 1 tipo		Simple	Simple	Médio
2 tipos		Simple	Médio	Complexo
3 ou mais tipos		Médio	Complexo	Complexo

Tabela 29 – Nível de complexidade SE

Arquivo referenciado	Dados Referenciados	1 a 5 itens	6 a 19 itens	20 ou mais itens
0 ou 1 tipo		Simple	Simple	Médio
2 ou 3 tipos		Simple	Médio	Complexo
4 ou mais tipos		Médio	Complexo	Complexo

Com suporte no modelo de dados é possível calcular os pontos de função correspondentes aos Arquivos Lógicos Internos e aos Arquivos de Interface externa [52].

Para o caso ALI, foram encontradas 3 entidades que são mantidas como arquivo interno: Notícia, Session e Utilizador. De notar que a tabela Login não é para ser mantida na aplicação visto que é informação privada do utilizador. De seguida, somam-se os atributos das 3 entidades (12 itens). Consultando a Tabela 26, tendo três arquivos referenciados e 12 dados referenciados, obtemos o nível de complexidade simples (ver Tabela 30).

Tabela 30 – Arquivo Lógico Interno

	Número de ocorrências	Peso	Resultado
Simple	3	7	21
Médio	0	10	0
Complexo	0	15	0

Total ALI	21
------------------	-----------

Dado que este projeto utiliza *webservices* que consomem informação externa à aplicação, foi necessário estabelecer os Arquivos de Interface externa (AIE). As entidades Notícia, Utilizador e Login correspondem à informação consumida pelos *webservices*. Uma vez mais, identificaram-se 3 arquivos que são mantidos externamente e que somando os seus atributos obtemos o valor de 12 itens. Consultando a Tabela 25, obtemos o nível de complexidade simples (ver Tabela 31).

Tabela 31 – Arquivo de Interface Externa

	Número de ocorrências	Peso	Resultado
Simple	3	5	15
Médio	0	7	0
Complexo	0	10	0
Total AIE			15

Tal como referido em secções anteriores, esta aplicação pretende unicamente consumir *webservices* não tendo qualquer funcionalidade de processamento de informação a ser enviada para o exterior. Deste modo, não se obteve qualquer SE.

A obtenção do nível de complexidade das entradas externas requiere uma análise à interface implementada ou a implementar. Através dos requisitos funcionais foram identificadas duas funcionalidades que seriam incluídas nesta premissa: *login* e *logout*.

Sabendo que a página *login* é constituída por duas *texto-box* (utilizador e password), um botão (para login) e uma caixa de erro caso este aconteça (com um botão “ok”), e como o único arquivo referenciado é a entidade Login, então procedeu-se à obtenção do nível de complexidade. Desta forma, em termos de arquivos referenciados são 1 (Entidade Login) e os dados referenciados correspondem aos itens de interface da página Login (2 *text-box* + 1 botão “*login*” + 1 botão “ok” = 4 itens referenciados). Obteve-se então o nível de complexidade simples (ver Tabela 28).

O mesmo procedimento aplica-se para o requisito funcional do *logout*. A página é constituída por 1 botão “logout” + 2 botões “tem a certeza que pretende fazer logout?” (Sim ou não) que é igual a 3 itens referenciados. O único arquivo referenciado que terá impacto com esta página será a Session, portanto o nível de complexidade é simples (ver Tabela 32Tabela 32).

Tabela 32 – Entrada Externa

	Número de ocorrências	Peso	Resultado
Simple	2	3	6
Médio	0	4	0
Complexo	0	6	0
Total EE			6

Por último, as consultas externas estão identificadas nos requisitos funcionais RF2, RF3 e RF4. Uma vez mais, recorre-se à interface da aplicação (ou protótipo) para se obter o nível de complexidade. O RF2 é representado por uma página em que são listadas as notícias. Cada item dessa lista é composto pelo Título e “Body” da notícia. Assim sendo, temos dois itens referenciados e um arquivo referenciado. Consultando a Tabela 27, obtemos o valor simples para o nível de complexidade. No caso do RF3, a página da notícia lista todos os atributos do arquivo referenciado “Noticia”. 6 itens referenciados e 1 arquivo referenciado. Também é nível de complexidade simples. O requisito funcional para consulta da informação do utilizador RF4, recorre unicamente ao arquivo referenciado Utilizador e lista a sua informação (4 itens). Nível de complexidade é simples (ver Tabela 33)

Tabela 33 – Consulta Externa

	Número de ocorrências	Peso	Resultado
Simple	3	3	9
Médio	0	4	0
Complexo	0	6	0
Total EE			9

Classificando cada um dos requisitos funcionais em termos de nível de complexidade, obtemos a tabela apresentada em seguida (ver Tabela 34).

Tabela 34 – Análise Pontos de Função Brutos

Função	Número de ocorrências	Complexidade	Peso	Resultado
Entrada Externa	2	Simple	x 3	6
	0	Médio	x 4	0
	0	Complexo	x 6	0
		Total EE		6
Saída Externa	0	Simple	x 4	0
	0	Médio	x 5	0
	0	Complexo	x 7	0
		Total SE		0
Arquivo Lógico Interno	3	Simple	x 7	21
	0	Médio	x 10	0
	0	Complexo	x 15	0
		Total ALI		21
Arquivo de Interface Externa	2	Simple	x 5	10
	0	Médio	x 7	0
	0	Complexo	x 10	0
		Total AIE		10
Consulta Externa	3	Simple	x 3	9
	0	Médio	x 4	0
	0	Complexo	x 6	0
		Total CE		9
Total (Bruto)				46

Conclui-se então que são 46 os pontos de função brutos (PFB).

Passo 3:

Calculados os pontos de função brutos, procede-se ao ajustamento do número de pontos de função brutos ao nível de complexidade de processamento. O fator de ajustamento tem por base as seguintes características gerais do sistema [51]:

- Comunicação de dados
Dados enviados e recebidos através de recursos de comunicação de dados.

- **Processamento distribuído**
Processamento é distribuído entre várias unidades de processamento.
- **Performance**
Elementos relacionados com a performance da aplicação que podem influenciar a sua implementação.
- **Configuração do sistema**
Se requer que o sistema seja configurado para a aplicação funcionar.
- **Volume de transações**
Corresponde à quantidade de transações que é possível fazer através da aplicação.
- **Entrada de dados *online***
Corresponde à quantidade de informação de sistemas externos que é recebida na aplicação.
- **Eficiência da interface para o utilizador final**
Se a interface da aplicação foi desenvolvida de modo a facilitar a interação do utilizador com a máquina.
- **Atualização de dados *online***
Caso a aplicação permita fazer atualização *online* dos arquivos lógicos internos.
- **Processamento complexo**
Processamento requerido pelo sistema para determinadas funcionalidades é elevado.
- **Reutilização do código fonte**
É relativo ao código que pode ser recuperado/melhorado de outras aplicações.
- **Facilidade de implementação**
É uma característica do sistema que permite identificar o nível da implementação da aplicação e ferramentas de conversão de dados.
- **Facilidade operacional**
Corresponde à automatização de procedimentos sem necessidade da intervenção manual dos operadores.
- **Múltiplos *sites***
Neste caso identificam-se os vários sistemas operativos e suas *stores*.
- **Facilidade de mudança**
Característica que determina se uma aplicação foi desenhada com o objetivo facilitar mudanças e gestão da informação por parte do utilizador.

Para cada característica é necessário determinar o nível de influência de 0 a 5, em que 0 significa que a aplicação não apresenta esta característica e 5 significa que é uma característica essencial (ver Tabela 35).

Tabela 35 – Características gerais do sistema

	Nível de influência
Comunicação de dados	5
Processamento distribuído	0
Performance	4
Configurações do sistema	0
Volume de transações	0
Entrada de dados <i>online</i>	5
Eficiência da interface para o utilizador final	3
Atualização de dados <i>online</i>	0
Processamento complexo	0
Reutilização de código	2
Facilidade de implementação	3
Facilidade operacional	0
Múltiplos <i>sites</i>	4
Facilidade de mudança	2
Nível de influência geral (Total)	28

Calculado o nível de influência geral (NIG), foi possível calcular o fator de ajuste (FA) com a seguinte expressão [52]:

$$FA = 0.65 + (NIG * 0.01) = 0.93$$

Depois procedeu-se ao cálculo do valor dos pontos de função ajustados (PFA) da seguinte forma [52]:

$$PFA = PFB * FA = 46 * 0.93 = 42.78$$

Desta forma, foi possível fazer uma previsão do tempo de implementação e do custo associado de cada aplicação. Para tal, estimou-se que o valor de pontos de função/ hora para implementação em C# (Xamarin) é de 12.31 e para o caso de JavaScript + Html (Apache Cordova) é de 9.84 [52].

O tempo de implementação da aplicação em Xamarin estimou-se da seguinte forma:

$$Txamarin = 42.78 * 12.31 = 526.62 \text{ horas (aprox. 13 semanas)}$$

O tempo de implementação da aplicação em Apache Cordova estimou-se da seguinte forma:

$$Tacordova = 42.78 * 9.84 = 420.96 \text{ horas (aprox. 10 semanas)}$$

Para calcular o custo de desenvolvimento da aplicação para este caso específico, dado que se trata de um programador sem grande experiência, estabeleceu-se o valor de 700€/mês como mensalidade. Sabendo que um mês tem aproximadamente 168,5 horas de trabalho, o preço por hora do programador com estas características equivale a 4.15€ por hora.

Assim sendo, a previsão do custo de desenvolvimento da aplicação em Xamarin corresponde a:

$$C_{xamarin} = 526.62 * 4.15 = 2187.74€$$

E a previsão do custo de desenvolvimento da aplicação em Apache Cordova equivale:

$$C_{cordova} = 420.96 * 4.15 = 1743€$$

5.3.2 Conclusão da análise

Tal como referido anteriormente, esta ferramenta (APF) é baseada numa avaliação a diversos elementos que são relativos e dependentes de vários contextos. Neste caso, os elementos de avaliação utilizados, tempo de desenvolvimento por ponto de função e custo, são valores apoiados com base na equipa de desenvolvimento do projeto.

Apesar de existir uma grande recolha de informação para serem obtidos os valores para o tempo de desenvolvimento, a experiência dos programadores com uma determinada tecnologia é um fator que pode influenciar bastante este elemento de avaliação. Para além disso, o conhecimento desta tecnologia é um valor que varia exponencialmente dado que o *know-how* do programador vai aumentando ao longo do tempo.

Por outro lado, o custo da aplicação varia conforme o número de elementos da equipa de desenvolvimento e salário que auferem.

Neste cenário, dado que a equipa de desenvolvimento era constituída por um único elemento e sem experiência em ambas as tecnologias, determinou-se o tempo de desenvolvimento por ponto de função com base em documentação e o seu salário foi o valor usado para a previsão do custo.

Por norma, e seguindo a recomendação de diversos autores, deve estar associado um erro de aproximadamente 30% à previsão feita. No entanto, os valores apresentados não apresentam qualquer erro adicionado.

Comparando a previsão feita inicialmente, com a informação real que foi obtida durante e depois do desenvolvimento de ambas as aplicações, foi possível constatar que o tempo de desenvolvimento da aplicação em Apache Cordova durou 14 semanas e 18 semanas.

Para uma análise à diferença entre a previsão e os dados reais em termos de tempo de desenvolvimento total, é necessário considerar vários aspetos que podem ter sido as principais influências para a obtenção deste resultado:

- A previsão foi calculada com base no número de horas brutas (8 horas diárias/ 5 dias por semana).
- Existiu dependência com os serviços disponibilizados pelo ISEP, por exemplo, obtenção da informação relativa a *webservices*.
- Tratamento de erros demorou mais tempo que o esperado.

Por outro lado, o custo financeiro ultrapassou o valor estimado dado que foi necessário um período de tempo dedicado à aprendizagem das tecnologias e configurações de todos os sistemas.

6 Conclusão

6.1 Objetivos realizados

O principal objetivo deste projeto centrou-se no desenvolvimento de uma aplicação móvel para que a comunidade do ISEP poder consultar informação relativa ao Portal. Com este projeto concluiu-se que a escolha de uma tecnologia multiplataforma é um desafio complexo, devido a todos os aspetos a ter em conta, quer em termos da equipa de desenvolvimento, quer em termos de uma avaliação das várias vantagens e desvantagens inerentes a cada tecnologia. Tornou-se necessário então estudar as abordagens existentes e perceber as necessidades do público alvo. Após este estudo exploratório, procedeu-se ao desenvolvimento de dois protótipos em duas tecnologias distintas: Apache Cordova e Xamarin. Apesar da complexidade e diversidade das funcionalidades existentes no Portal, com este projeto foi possível implementar os módulos login, notícias, informação do utilizador e Logout, tornando-se desta forma válidos os objetivos previamente definidos.

Desta forma, a implementação e realização de testes de usabilidade dos dois protótipos com utilizadores reais (alunos, professores e funcionários do ISEP); e as respostas aos questionários de satisfação permitiram perceber que todos os objetivos propostos neste projeto foram concretizados com sucesso.

6.2 Limitações e trabalho futuro

A avaliação das limitações do projeto teve como base os testes de usabilidade, realizados pelo público alvo e ainda pela análise ao desenvolvimento dos protótipos.

No que concerne aos testes de usabilidade o tamanho da amostra é relativamente baixo. Isto deveu-se à dificuldade no recrutamento de utilizadores e devido à complexidade dos testes. Assim sendo, a análise em termos de preferência de uma determinada aplicação por parte dos utilizadores é inconclusiva dado que dos 36 participantes, 50% preferiram a aplicação em Xamarin e 44.4% a aplicação em Apache Cordova.

Através destes testes foi notório perceber que a aplicação está mais direcionada para os alunos do que para a restante comunidade académica. Verificou-se que os alunos tinham conhecimento da aplicação existente e experiência na mesma, contrariamente ao que se verifica com alguns professores e funcionários que não tinham conhecimento da aplicação e/ou não a utilizavam no seu dia a dia. Este aspeto teve alguma relevância na resposta ao inquérito dado que os alunos sabiam referir problemas da aplicação existente, assim como sugerir melhorias e mais funcionalidades para uma futura aplicação.

Desta forma, caso seja do interesse da instituição, torna-se imprescindível implementar funcionalidades vocacionadas para este grupo (funcionários e professores).

Apesar das limitações apresentadas nos testes de usabilidade, o presente projeto deixa em aberto algumas funcionalidades para implementar no futuro:

- Associar o cartão de aluno, funcionário e professor à aplicação de forma a que seja permitida a entrada numa sala através do smartphone.
- Permitir pedidos de certificados, assim como consultar as equivalências e/ou estados dos pedidos.
- Consultar referência, entidade e valor de pagamentos multibanco.
- Marcar férias
- Justificar faltas.
- Criar um canal de comunicação entre professor e aluno.
- Consultar a assiduidade.
- Receber notificações.
- Desenvolver uma área de emprego na qual os alunos podem consultar propostas de emprego.
- Consultar horário de transportes públicos perto da instituição.
- Implementar uma área para a biblioteca onde seja possível verificar estado da requisição de livros e disponibilidade da mesma.
- Solicitar senha para a secretaria e consultar estado das senhas.
- Inscrição em exames.
- Contactos do ISEP.

Em termos de desenvolvimento é necessário perceber que este projeto se focou no desenvolvimento de duas aplicações estando elas dependentes dos serviços disponibilizados pelo Portal da instituição. Como tal, foi possível verificar que por vezes o acesso à base de dados apresenta um funcionamento aleatório. Há pedidos que demoram mais tempo que o esperado na obtenção da resposta, tendo um grande impacto na performance da aplicação. Para além deste facto a documentação existente é praticamente nula.

De notar que a experiência do aluno no desenvolvimento de aplicações móveis centrava-se em Windows Phone app e um pouco de Android. Como tal, foi necessário que o aluno tivesse que aprender quase na totalidade a programar para estas plataformas.

Outra limitação do projeto e consequente dificuldade obtida foi no desenvolvimento da aplicação em Apache Cordova. A meio do projeto foi necessário passar a solução para outra máquina e visto que esta tinha uma versão do Visual Studio diferente, a aplicação não corria. A solução encontrada foi instalar a versão do Visual Studio anterior. Este é também um aspeto a ter em conta quando se pretender fazer *updates* à versão do Apache Cordova e do Visual Studio.

No que diz respeito ao *deploy* das aplicações, no caso do Xamarin, dado que o aluno não possuía um Mac, não foi possível instalar e testar a aplicação Xamarin em IOS. Por outro lado, em Apache Cordova, não foi possível solucionar os diversos erros que aplicação dava quando se tentava fazer o *deploy* para os smartphones. Inclusivamente o dispositivo Android não era reconhecido no Visual Studio (projeto Apache Cordova) enquanto no projeto Xamarin não se verificou o mesmo problema.

Uma vez mais este projeto teve o teor de investigação e de exploração. Por isso, de modo a que seja rapidamente disponibilizada uma aplicação que satisfaça as necessidades da comunidade do ISEP, é necessário tomar uma decisão em relação à tecnologia a adotar e que melhor se adequa à equipa de desenvolvimento existente no ISEP.

6.3 Apreciação Final

No decorrer deste trabalho foi possível perceber que aplicação existente do Portal não satisfaz as necessidades da comunidade do ISEP. No entanto apresenta uma plataforma estável e com bastante qualidade.

Seria então necessário criar uma aplicação mais eficaz, no sentido de construir sinergias que facilitem a interação da comunidade do ISEP com o Portal, facilitando e tornando o acesso à informação mais rápida e funcional.

7 Referências

- [1] ISEP, "ISEP Brochura Empresas," [Online]. Available: https://issuu.com/isep/docs/isep_brochura_empresas_web. [Accessed 15 Janeiro 2017].
- [2] ISEP, "Portal ISEP," ISEP, [Online]. Available: <https://portal.isep.ipp.pt/intranet/Default.aspx>. [Accessed 15 Janeiro 2017].
- [3] ISEP, "Site ISEP," [Online]. Available: <https://www.isep.ipp.pt/Home>. [Accessed 15 Janeiro 2017].
- [4] Microsoft, "Cross-platform Hybrid and Native Technologies for Mobile App Development for iOS, Android and Windows," [Online]. Available: <https://info.microsoft.com/CO-BRDEV-CNTNT-FY16-02Feb-Cross-Platform-Mobile-App-Development-Registration.html>. [Accessed 1 Fevereiro 2017].
- [5] C. Wodehouse, "Using Cross-platform Technology to Build Flexible, Functional Mobile Apps," [Online]. Available: <https://www.upwork.com/hiring/mobile/cross-platform-mobile-app-technology/>. [Accessed 2017 Fevereiro 2].
- [6] M. David, "How to choose cross-platform app development tools," 10 Agosto 2015. [Online]. Available: <http://searchsoftwarequality.techtarget.com/feature/How-to-choose-cross-platform-app-development-tools>. [Accessed 2 Fevereiro 2017].
- [7] Xamarin, "Deliver native Android, iOS, and Windows apps, using existing skills, teams, and code.," [Online]. Available: <https://www.xamarin.com/platform>. [Accessed 22 Janeiro 2017].

- [8] A. Cordova, "Introduction," [Online]. Available:
] <https://cordova.apache.org/docs/en/latest/guide/overview/>. [Accessed 17 Fevereiro 2017].
- [9] Sencha, "Sencha Touch," [Online]. Available:
] <https://www.sencha.com/products/touch/#overview>. [Accessed 24 Janeiro 2017].
- [10] Appcelerator, "Build great mobile experiences faster," [Online]. Available:
] <https://www.appcelerator.com/>. [Accessed 23 Janeiro 2017].
- [11] Lambda3, "O que é Xamarin?," [Online]. Available:
] <https://blog.lambda3.com.br/2016/10/o-que-e-xamarin/>. [Accessed 21 Janeiro 2017].
- [12] Xamarin, "Mobile app testing made easy.," [Online]. Available:
] <https://www.xamarin.com/test-cloud>. [Accessed 22 Janeiro 2017].
- [13] Xamarin, "Xamarin.Insights + HockeyApp. Now the industry-leader.," [Online]. Available:
] <https://www.xamarin.com/insights>. [Accessed 22 Janeiro 2017].
- [14] Xamarin, "Xamarin.Forms," [Online]. Available: <https://www.xamarin.com/forms>.
] [Accessed 22 Janeiro 2017].
- [15] Xamarin, "Thousands of businesses worldwide use Xamarin to build, test, and monitor powerful apps.," [Online]. [Accessed 22 Janeiro 2017].
- [16] A. Cordova, "Cordova Tools," [Online]. Available: <https://cordova.apache.org/>. [Accessed
] 22 Janeiro 2017].
- [17] A. Cordova, "Documentation," [Online]. Available:
] <https://cordova.apache.org/docs/en/latest/guide/overview/>. [Accessed 22 Janeiro 2017].
- [18] E. F. A. Soares, "TelePalestra – Ensino em multicast," Porto, 2014.
]
- [19] SenchaDevelopers, "What is Sencha Framework?," [Online]. Available:
] <http://senchadevelopers.blogspot.de/2012/10/what-you-can-do-with-sencha.html>.
] [Accessed 24 Janeiro 2017].
- [20] P. Traeg, "Four Ways To Build A Mobile Application, Part 4: Appcelerator Titanium," [Online]. Available: <https://www.smashingmagazine.com/2014/03/4-ways-build-mobile-application-part4-appcelerator-titanium/>. [Accessed 23 Janeiro 2017].

- [21 K. D. Solutions, "What is Appcelerator Titanium?," [Online]. Available:
] <https://klocdigitalsolutions.wordpress.com/2015/08/11/what-is-appcelerator-titanium/>.
[Accessed 23 Janeiro 2017].
- [22 Appcelerator, "Native apps. Mobile APIs. Real-time analytics. One Platform.," [Online].
] Available: <https://www.appcelerator.com/mobile-app-development-products/>.
[Accessed 23 Janeiro 2017].
- [23 M. T. d. P. Sousa, Z. . M. M. Arruda and A. S. S. Fernandes , "Desenvolvimento de
] funcionalidades do Portal do ISEP para dispositivos móveis," ISEP, Porto, 2011.
- [24 L. F. P. D. Michael H. Morris, Business-to-Business Marketing: A Strategic Approach,
] Londres: SAGE, 2001.
- [25 N. Solutions, "Value Analysis and Function Analysis System Technique," [Online].
] Available: <http://www.npd-solutions.com/va.html>. [Accessed 13 Fevereiro 2017].
- [26 "Woodruff, R.B. J. of the Acad. Mark. Sci. (1997) 25: 139. doi:10.1007/BF02894350,"
] [Online]. Available: <http://link.springer.com/article/10.1007%2FBF02894350>. [Accessed
13 Fevereiro 2017].
- [27 N. Kokemuller, "What Is Customer Perceived Value?," [Online]. Available:
] <http://smallbusiness.chron.com/customer-perceived-value-23692.html>. [Accessed 13
Fevereiro 2017].
- [28 A. Osterwalder, Value Proposition Design: How to create products and services
] customers want, New Jersey: Wiley, 2014.
- [29 T. L. Saaty, "What is the analytic hierarcgy process?," *Mathematical models for decision
] support*, pp. 109-121, 1998.
- [30 S. Nicola, "Análise de Valor," [Online]. Available:
] [https://moodle.isep.ipp.pt/pluginfile.php/143572/mod_resource/content/1/An%C3%A1
ise_Valor_Aula4.pdf](https://moodle.isep.ipp.pt/pluginfile.php/143572/mod_resource/content/1/An%C3%A1lise_Valor_Aula4.pdf). [Accessed 15 Fevereiro 2017].
- [31 Xamarin, "Part 2 - Architecture," [Online]. Available:
] [https://developer.xamarin.com/guides/cross-
platform/application_fundamentals/building_cross_platform_applications/part_2_-
_architecture/](https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_2_-_architecture/). [Accessed 17 Fevereiro 2017].
- [32 Xamarin, "Common Patterns and Idioms," [Online]. Available:
] https://developer.xamarin.com/guides/mac/application_fundamentals/patterns/.
[Accessed 17 Fevereiro 2017].

- [33 J. Bennet, "MVVM: The Design Pattern for Xamarin Apps," 15 Janeiro 2017. [Online].
] Available: <https://dzone.com/articles/mvvm-the-design-pattern-for-xamarin-apps>.
[Accessed 17 Fevereiro 2017].
- [34 Xamarin, "Sharing Code Options," [Online]. Available:
] https://developer.xamarin.com/guides/cross-platform/application_fundamentals/code-sharing/. [Accessed 16 Fevereiro 2017].
- [35 Xamarin, "Shared Projects," [Online]. Available:
] https://developer.xamarin.com/guides/cross-platform/application_fundamentals/shared_projects/. [Accessed 16 Fevereiro 2017].
- [36 R. Andrade, "A diferença entre soluções Portable e Shared no Xamarin.Forms," [Online].
] Available: <http://www.randrade.net/2016/06/10/xamarin-forms-a-diferenca-entre-solucoes-portable-e-shared/>. [Accessed 16 Fevereiro 2017].
- [37 Xamarin, "Introduction to Portable Class Libraries," [Online]. Available:
] https://developer.xamarin.com/guides/cross-platform/application_fundamentals/pcl/introduction_to_portable_class_libraries/.
[Accessed 16 Fevereiro 2017].
- [38 J. Smith, "PCL or Shared Project?," 19 Janeiro 2016. [Online]. Available:
] <http://xfcomplete.net/general/2016/01/19/pcl-or-shared-project/>. [Accessed 16
Fevereiro 2017].
- [39 Xamarin, "Limitations of Portable projects vs. Shared projects," [Online]. Available:
] <https://forums.xamarin.com/discussion/17304/limitations-of-portable-projects-vs-shared-projects>. [Accessed 16 Fevereiro 2017].
- [40 Xamarin, "Why Shared Projects?," [Online]. Available:
] <https://forums.xamarin.com/discussion/59381/why-shared-projects>. [Accessed 16
Fevereiro 2017].
- [41 Stackoverflow, "What is a MV* framework?," [Online]. Available:
] <http://stackoverflow.com/questions/28930186/what-is-a-mv-framework>. [Accessed 17
Fevereiro 2017].
- [42 TypeFramework, "A MVC Web Framework for TypeScript," [Online]. Available:
] <http://typeframework.com/>. [Accessed 17 Fevereiro 2017].
- [43 Backbone.js, "Backbone.js," [Online]. Available: <http://backbonejs.org/>. [Accessed 17
Fevereiro 2017].

- [44 PhoneGapPro, "Best UI Framework for Phonegap / Apache Cordova," [Online]. Available:
] <https://phonegappro.com/tools/best-ui-framework-for-phonegap-apache-cordova/>.
] [Accessed 17 Fevereiro 2017].
- [45 J. Rubin, Handbook of usability testing: How to plan, design, and conduct effective tests,
] New York: Wiley, 1994.
- [46 J. Marôco, Análise Estatística com o SPSS Statistics, Pêro Pinheiro: ReportNumber, 2014.
]
- [47 F. Rösler, "Towards a Mobile Application Performance Benchmark," in *ICIW 2014 : The
] Ninth International Conference on Internet and Web Applications and Services*, Paris,
] 2014.
- [48 M. M. P. Gomes, "Métricas e medição no processo de desenvolvimento de Software,"
] ISCTE-IUL, Lisboa, 2010.
- [49 C. Kaner and W. P. Bond, "Software Engineering Metrics: What Do They Measure and
] How Do We Know?," 10TH International Software Metrics Symposium, Chicago, 2004.
- [50 G. Srilatha, R. Madhumathi, P. Sreshta and C. Thir, "Analysis of LOC attributes using code
] analyzer and correlation methods," in *International Conference on Trends in Electronics
] and Informatics (ICEI)*, Tirunelveli, 2017.
- [51 IFPUG, International Function Point - Function Point Counting Practices Manual,
] Michigan: IFPUG, 2005.
- [52 C. Jones, Software Economics and Function Point Metrics: Thirty years of IFPUG Progress,
] Rhode Island: Namcook Analytics LLC, 2017.
- [53 ISEP, "ISEP: Manual de Acolhimento dos Novos Alunos 2014-2015," 5 Setembro 2014.
] [Online]. Available: <https://issuu.com/isep/docs/isep.2014.manualacolhimentonovoalun>.
] [Acedido em 10 Janeiro 2017].
- [54 ISEP, "Portal ISEP," Google Play, [Online]. Available:
] https://play.google.com/store/apps/details?id=pt.ipp.isep.portal.portal_isep. [Accessed
] 12 Janeiro 2017].
- [55 ISEP, "Portal ISEP," Apple Store, [Online]. Available:
] <https://itunes.apple.com/pt/app/portal-isep/id758240892?l=pt&ls=1&mt=8>. [Accessed
] 12 Janeiro 2017].
- [56 ISEP, "Portal ISEP," Windows Store, [Online]. Available: [https://www.microsoft.com/pt-
\] pt/store/p/portal-isep/9wzdncrdjlpz](https://www.microsoft.com/pt-pt/store/p/portal-isep/9wzdncrdjlpz). [Accessed 12 Janeiro 2017].

[57 ISEP, “Meteorologia ISEP,” [Online]. Available: <http://meteo.isep.ipp.pt/>. [Accessed 13 Janeiro 2017].

[58 U. C. Portuguesa, “MyCatólica,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.moofwd.ucppt&hl=pt>. [Accessed 17 Janeiro 2017].

[59 U. Lusíada, “eLusiada,” U. Lusíada, [Online]. Available: <https://itunes.apple.com/pt/app/elusiada/id883802665?mt=8> . [Accessed 15 Janeiro 2017].

[60 AAFDL, “Associação Académica da Faculdade de Direito de Lisboa,” [Online]. Available: <https://itunes.apple.com/br/app/aafdl-associacao-academica/id1034864473?mt=8>. [Accessed 16 Janeiro 2017].

[61 B. Spot, “BG Spot,” [Online]. Available: <https://bgspotuc.wordpress.com/>. [Accessed 15 Janeiro 2017].

[62 IBM, “IBM Knowledge Center,” 18 1 2018. [Online]. Available: https://www.ibm.com/support/knowledgecenter/pt-br/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac34640_.htm.

[63 P. M. T. d. Sousa, Z. M. M. Arruda and A. S. S. Fernandes, “Desenvolvimento de funcionalidades do Portal do ISEP para dispositivos móveis,” Instituto Superior de Engenharia do Porto, Porto, 2011.

Portal ISEP mobile

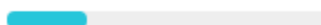
Este questionário realizado de forma anónima apresenta algumas questões sobre a demonstração em que participou e na qual pôde testar por si mesmo as funcionalidades de ambos os protótipos.

As questões estão divididas pelas seguintes categorias:

- a) Perfil do utilizador.
- b) Questões gerais sobre a aplicação do Portal ISEP.
- c) Avaliação dos protótipos desenvolvidos.

Agradeço desde já a sua participação!

SEGUINTE



Página 1 de 4

Nunca envie palavras-passe através dos Formulários do Google.

Perfil do utilizador

Idade *

A sua resposta

Género *

Masculino

Feminino

Estatuto *

Estudante

Professor

Funcionário

Sistema operativo do seu telemóvel *

Selecionar ▼

Questões gerais sobre a aplicação do Portal ISEP

O acesso a várias funcionalidades do Portal através de uma aplicação móvel seria do seu interesse? *

Sim

Não

Que funcionalidades gostaria de ter acesso na aplicação móvel?

*

Salas Disponíveis

Parque Automóvel

Notas Parciais

Notícias

Horário

Notas Finais

Finanças

Outra: _____

Tem conhecimento da existência da aplicação 'Portal ISEP'? *

Sim

Não

Com que frequência usa a aplicação? *

Selecionar ▼

Qual é o seu grau de satisfação com a aplicação existente? *

Numa escala em que "0" significa que se sente totalmente insatisfeito e 5 significa que se sente totalmente satisfeito.

0 1 2 3 4 5

Se a sua resposta é menor que 5, quais as razões que o levam a não estar totalmente satisfeito?

Não consigo aceder.

Problemas de funcionamento.

Outra: _____

Avaliação dos protótipos

Qual a aplicação que prefere? *

Aplicação A

Aplicação B

Indiferente

Porquê?

Apresenta melhor design

Facilidade em obter a informação

Sistema mais rápido

Outra: _____

Sugere alguma melhoria na Aplicação A?

A sua resposta

Sugere alguma melhoria na Aplicação B?

A sua resposta

Anexo 2 Dados estatísticos

Frequência de idades

N	Valid	36
	Missing	0
Mean		27,03
Std. Deviation		9,551
Minimum		19
Maximum		53

Gênero

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Masculino	18	50,0	50,0	50,0
	Feminino	18	50,0	50,0	100,0
	Total	36	100,0	100,0	

Estatuto

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Estudante	29	80,6	80,6	80,6
	Professor	4	11,1	11,1	91,7
	Funcionário	3	8,3	8,3	100,0
	Total	36	100,0	100,0	

Sistema Operativo

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Android	30	83,3	83,3	83,3
	Windows Phone	6	16,7	16,7	100,0
	Total	36	100,0	100,0	

App com acesso a várias funcionalidades do Portal

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	36	100,0	100,0	100,0

Notícias

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	10	27,8	27,8	27,8
	Não	26	72,2	72,2	100,0
	Total	36	100,0	100,0	

Horário

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	34	94,4	94,4	94,4
	Não	2	5,6	5,6	100,0
	Total	36	100,0	100,0	

Notas Parciais

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	26	72,2	72,2	72,2
	Não	10	27,8	27,8	100,0
	Total	36	100,0	100,0	

Notas Finais

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	28	77,8	77,8	77,8
	Não	8	22,2	22,2	100,0
	Total	36	100,0	100,0	

Salas Disponíveis

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	25	69,4	69,4	69,4
	Não	11	30,6	30,6	100,0
	Total	36	100,0	100,0	

Parque Automóvel

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	21	58,3	58,3	58,3
	Não	15	41,7	41,7	100,0
	Total	36	100,0	100,0	

Finanças

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	13	36,1	36,1	36,1
	Não	23	63,9	63,9	100,0
	Total	36	100,0	100,0	

Outras funcionalidades

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	11	30,6	30,6	30,6
	Não	25	69,4	69,4	100,0
	Total	36	100,0	100,0	

Conhecimento da existência da aplicação

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	33	91,7	91,7	91,7
	Não	3	8,3	8,3	100,0
	Total	36	100,0	100,0	

Frequência de uso da aplicação

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Diariamente	5	13,9	13,9	13,9
	3 a 5 vezes por semana	4	11,1	11,1	25,0
	1 a 2 vezes por semana	6	16,7	16,7	41,7
	Nunca	21	58,3	58,3	100,0
	Total	36	100,0	100,0	

Grau de satisfação (nunca utilizam app)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Totalmente Insatisfeito	1	4,8	4,8	4,8
	Muito Insatisfeito	2	9,5	9,5	14,3
	Insatisfeito	3	14,3	14,3	28,6
	Satisfeito	3	14,3	14,3	42,9
	Muito Satisfeito	1	4,8	4,8	47,6
	Sem Opinião	11	52,4	52,4	100,0
	Total	21	100,0	100,0	

Problemas de Funcionamento (nunca utilizam a app)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	5	23,8	23,8	23,8
	Não	16	76,2	76,2	100,0
	Total	21	100,0	100,0	

Outros problemas (nunca utilizam a app)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	7	33,3	33,3	33,3
	Não	14	66,7	66,7	100,0
	Total	21	100,0	100,0	

Problemas de Funcionamento (utilizam app)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	11	73,3	73,3	73,3
	Não	4	26,7	26,7	100,0
	Total	15	100,0	100,0	

Outros problemas (utilizam app)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sim	6	40,0	40,0	40,0
	Não	9	60,0	60,0	100,0
	Total	15	100,0	100,0	

Inacessível

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Não	36	100,0	100,0	100,0

Aplicação preferida

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid Aplicação A	18	50,0	50,0	50,0
Aplicação B	16	44,4	44,4	94,4
Sem preferência	2	5,6	5,6	100,0
Total	36	100,0	100,0	

Cochran Test

Frequências

Aplicação preferida		Value	
		1	2
Aplicação A	Melhor Design	8	10
	Facilidade na obtenção da informação	14	4
	Sistema mais rápido	6	12
Aplicação B	Melhor Design	11	5
	Facilidade na obtenção da informação	6	10
	Sistema mais rápido	1	15

Estatísticas do Cochran Test

Aplicação A	N	18
	Cochran's Q	6,118 ^a
	df	2
	Asymp. Sig.	,047
Aplicação B	N	16
	Cochran's Q	10,714 ^a
	df	2
	Asymp. Sig.	,005

Aplicação preferida por outras razões

			Outras		Total
			Sim	Não	
Aplicação preferida	Aplicação A	Count	0	18	18
		% of Total	0,0%	50,0%	50,0%
	Aplicação B	Count	1	15	16
		% of Total	2,8%	41,7%	44,4%
	Sem preferência	Count	0	2	2
		% of Total	0,0%	5,6%	5,6%
Total	Count	1	35	36	
	% of Total	2,8%	97,2%	100,0%	
