

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



DESENVOLVIMENTO DE SOLUÇÕES MOBILE
PARA O BANCO DE PORTUGAL

Luís Duarte de Jesus Santos

PROJECTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Engenharia de Software

2014

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



DESENVOLVIMENTO DE SOLUÇÕES MOBILE
PARA O BANCO DE PORTUGAL

Luís Duarte de Jesus Santos

PROJECTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Engenharia de Software

Trabalho orientado pela Prof. Doutora Cátia Luísa Santana Calisto Pesquita
e coorientado por Abílio Neves Brás

2014

Agradecimentos

Agradeço a minha família, em especial aos meus pais e irmãs, pelo carinho e apoio incessante que sempre dedicaram durante esta jornada académica e ao longo da minha vida.

Agradeço à minha namorada Marlene Freitas todo o carinho, amizade e amor, foram um apoio muito importante.

Agradeço à professora Cátia Pesquita, minha orientadora, toda a sua disponibilidade, empenho e conhecimento que foram fundamentais na escrita desta dissertação.

Agradeço ao Banco de Portugal e ao meu coorientador Abílio Neves Brás, a oportunidade e o apoio que me foi dado.

Agradeço ao António Pires, toda a ajuda, disponibilidade, ensinamentos e paciência, foram extremamente importantes.

Agradeço aos meus colegas de trabalho Tiago Sousa, André Grilo, Vicente Oliveira e Tânia Diniz, as horas de paciência, a capacidade de integração e o apoio dado no projeto, foram sem dúvida fulcrais na minha iniciação na vida profissional. Um agradecimento especial para a Tânia Diniz que foi incansável na ajuda e orientação de todo o projeto.

Agradeço à todos os colegas de faculdade que sempre estiveram presentes em todas as etapas da minha vida académica, dando força e apoio, os quais se tornaram verdadeiros amigos, em especial o Cláudio Saramagaio, Rafael Oliveira, Luís Rochinha e Vasco Tareco, foram sem dúvida fundamentais.

Agradeço aos meus amigos da terra natal, que sempre me acompanharam e ajudaram quando precisei, em especial ao Aurélio Pita, Miguel Quintal, Carlos Gomes, Alex Correia, Célia e Ivo da Silva.

E a todos que contribuíram direta ou indiretamente para a realização deste trabalho.

Dedico este trabalho às pessoas mais importantes da minha vida pelo que me ensinaram e transmitiram e pelo apoio incondicional e infundável

Aos meus pais

Às minhas irmãs

À minha namorada

Resumo

Atualmente, com a mobilidade que os *smartphones* e *tablets* proporcionam, cada vez mais assistimos a uma migração das empresas e das pessoas para o mundo *mobile* na procura de divulgar as suas marcas ou produtos, nesta nova realidade que continua a crescer a passos largos. O Banco de Portugal, no cumprimento da sua missão, definiu como uma das linhas gerais de orientação estratégica a promoção de um Banco aberto e respeitado pela sociedade, contribuindo para uma sociedade mais informada através da disponibilização de informação útil, compreensível e acessível a todos os utilizadores, pelo que o desenvolvimento de soluções *mobile* tornou-se fundamental. No entanto, existe uma grande segmentação de plataformas no mercado, o que dificulta o desenvolvimento de aplicações para estes dispositivos. É neste sentido que começaram a aparecer estratégias baseadas em tecnologias *web* como HTML5, JavaScript e CSS3, que vieram agilizar o trabalho dos programadores quando procuram desenvolver aplicações móveis multiplataforma.

Este Projeto de Mestrado em Engenharia Informática aborda o desenvolvimento de aplicações *web* móveis usando as abordagens *Responsive Web Design* e Aplicações Híbridas. Estas foram utilizadas no desenvolvimento de duas soluções: um protótipo da página inicial do *site* Institucional do Banco de Portugal, e uma aplicação híbrida para *smartphones* e *tablets* que suporta alguns dos principais serviços existentes no atual *site*.

Foram ainda realizados diversos testes de software no decorrer dos diversos estádios de desenvolvimento das aplicações, os quais revelaram-se fundamentais no processo de correção de falhas, permitindo uma entrega final com maior qualidade.

Palavras-chave: aplicações móveis, desenvolvimento web, aplicações híbridas, web design, responsive web design

Abstract

Smartphones and tablets are pushing people and businesses into the mobile world to advertise their brands and products. The Bank of Portugal, in the fulfillment one of its mission, has defined as strategic guidelines the promotion of a respected Bank, contributing to a more informed society by providing useful information, understandable and accessible to all users. The development of mobile solutions is crucial in accomplishing these goals. However, there is a segmentation of platforms on the market, which hinder the development of applications for these devices. To address this strategies based on web technologies like HTML5, CSS3 and JavaScript, have been created aiding the programmers work in developing multi-platform mobile applications.

This Master's project in Computer Engineering focuses on developing mobile web applications using Responsive Web Design and Hybrid Applications approaches. These approaches were used to develop a prototype for the Banco de Portugal website homepage and a hybrid application to provide some of the main services in the current institutional website of Banco de Portugal. We carried out several software tests along the various stages of applications development, being these fundamental in the error correction process, allowing a final delivery with higher quality.

Keywords: mobile applications, web development, hybrid applications, responsive web design, design

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Motivação	1
1.2	Objetivos.....	2
1.3	Notação adotada	2
1.4	Organização do documento	3
Capítulo 2	Trabalho relacionado.....	4
2.1	Estado da Arte	4
2.1.1	Aplicações Nativas vs. <i>Web</i> vs. Híbridas.....	4
	Aplicações Nativas.....	5
	Aplicações <i>Web</i>	5
	Aplicações Híbridas	6
	Comparação	7
2.1.2	Aplicações <i>web</i> (<i>Responsive vs. Adaptive Design</i>).....	9
	<i>Responsive Web Design</i> (RWD)	9
	<i>Adaptive Web Design</i> (AWD).....	10
	Comparação	11
2.1.3	HTML5.....	12
	Suporte <i>offline</i> e armazenamento.....	12
	Canvas.....	12
2.1.4	<i>Frameworks</i> para desenvolvimento <i>mobile</i>	13
	PhoneGap/Apache Cordova.....	13
	Sencha Touch.....	13
	Twitter Bootstrap	13
	jQuery Mobile.....	14
	Modernizr.....	14
2.2	Casos de Estudo.....	14
2.2.1	Banco de França	14

2.2.2	Banco da Suécia	16
2.2.3	Banco da Bélgica e do Japão.....	17
2.2.4	Banco da Suíça	18
Capítulo 3	Enquadramento.....	19
3.1	Instituição de acolhimento.....	19
3.2	Processo de Desenvolvimento.....	21
3.3	Calendarização do Projeto.....	21
Capítulo 4	Análise e especificação de requisitos	24
4.1	Análise de conteúdos e serviços	24
4.1.1	<i>Site</i> Institucional do Banco de Portugal	24
4.1.2	Maior número de acessos	25
4.1.3	Apetência natural para <i>mobile</i>	26
4.1.4	Centralização de outras aplicações <i>mobile</i>	26
4.1.5	Conclusão.....	26
4.2	Decisões técnicas face as estratégias de desenvolvimento.....	27
4.3	Documentação produzida	27
4.3.1	Especificação do sistema.....	28
	Componente <i>web</i>	28
	Componente <i>mobile</i>	29
	API <i>Web</i> do Banco de Portugal.....	29
4.3.2	Requisitos funcionais e de teste	29
4.3.3	Requisitos não funcionais.....	32
4.3.4	Casos de Uso	33
4.3.5	Esboços.....	36
	38
	38
	Esboços de alta-fidelidade	39
Capítulo 5	Desenho da solução.....	40
5.1	Arquitetura do sistema.....	40

5.1.1	Racional de decisões chave	41
5.2	Desenho e Modelação do sistema.....	42
5.2.1	Diagrama de módulos.....	42
	Descrição de Módulos.....	43
5.2.2	Diagrama de classes	44
5.2.3	Diagramas de sequência	46
Capítulo 6	Implementação	48
6.1	<i>Web Services</i>	48
6.1.1	API RESTful	49
	Autenticação	50
	Criação de pedidos	50
	<i>Parsing</i> dos dados e resposta	51
6.2	Aplicação <i>mobile</i>	51
6.2.1	Cordova.....	52
6.2.1	<i>Web Page</i>	53
6.2.2	Módulo de configuração da aplicação.....	54
6.2.3	<i>Model View Controller</i> (MVC).....	54
	<i>Views</i>	56
	<i>Controllers</i>	56
	<i>Cordova Plugins</i>	57
6.3	Protótipo do <i>site</i>	58
Capítulo 7	Testes.....	61
7.1	Testes de sistema	61
7.2	Testes de carga.....	62
7.2.1	Teste 1 (<i>Web services</i> sem implementação de <i>cache</i>).....	63
7.2.2	Teste 2 (<i>Web services</i> com correções após o teste 1).....	64
7.2.3	Teste 3 (Quatro vezes mais carga relativamente ao teste 2)	65
7.3	Testes de usabilidade	66
7.3.1	Sucesso na realização de tarefas.....	67

7.3.2	Avaliação de usabilidade.....	67
7.3.3	Resultados	68
7.4	Testes de Compatibilidade.....	70
Capítulo 8	Conclusões	72
8.1	Considerações Finais	72
8.2	Contribuições.....	73
8.3	Perspetivas Futuras	74
Bibliografia	75
Capítulo 9	Anexos.....	78
Anexo A	78
A.1.	Relatório de análise	78
Anexo B	90
B.1.	Desenho e modelação do sistema.....	90
Descrição das classes	92
Anexo C	96
C.1.	Testes de sistema.....	96
Anexo D	100
D.1.	Testes de carga	100
Anexo E	116
E.1.	Testes de usabilidade	116
Anexo F	119
F.1.	Layouts WIZ.....	119
Anexo G	124
G.1.	Método AuthenticateSPS com o modelo de autenticação FBA da classe Authenticate.....	124
G.2.	Excerto do código usado na criação de um pedido dos <i>web services</i> com a camada de dados	125
G.3.	Código do objeto CreateRequest responsável pela criação de pedidos entre as camadas de serviços e dados	125

G.4. Código da estrutura de dados que permite guardar a resposta a um pedido do serviço TaxasJuro	126
G.5. Método CreateResponse da classe TaxasJuro.....	126
G.6. Excerto do código do módulo principal da aplicação híbrida.....	127
G.7. Código da API JavaScript do <i>plugin</i> desenvolvido e também do módulo nativo para iOS	128

Lista de Figuras

Figura 1 – Aplicação RWD InflexionInteractive.com em execução num desktop, tablet e smartphone.....	9
Figura 2 – Aplicação AWD AmericanAirlines.com em execução num desktop, tablet e smartphone.....	10
Figura 3 – Vista de menus da aplicação do Banco de França.....	15
Figura 4 – Vista dos discursos do governador da aplicação do Banco de França ..	15
Figura 5 – Vista de um desktop da página inicial do Banco Central da Suécia.....	16
Figura 6 – Vista do site <i>mobile</i> do banco do Japão quando acedido por um <i>smartphone</i>	17
Figura 7 – Vista do site <i>mobile</i> do Banco da Bélgica quando acedido por um <i>smartphone</i>	17
Figura 8 – Vista da página <i>mobile</i> do Banco da Suíça quando acedida por um <i>smartphone</i>	18
Figura 9 – Organograma geral do Banco de Portugal.....	20
Figura 10 – Diferentes fases do projeto no Modelo em V.....	21
Figura 11 – Mapa de Gantt do projeto	23
Figura 12 – Diagrama geral de casos de uso do protótipo do <i>site</i> e da aplicação móvel	34
Figura 13 - Esboço da vista de comunicados num <i>smartphone</i> da aplicação móvel	37
Figura 14 - Esboço da vista de comunicados num <i>tablet</i> da aplicação móvel.....	37
Figura 15 – Protótipo de baixa fidelidade da <i>homepage</i> num desktop	38
Figura 16 – Protótipo de baixa fidelidade da <i>homepage</i> num <i>tablet</i>	38
Figura 17 – Protótipo de baixa fidelidade da <i>homepage</i> num <i>smartphone</i>	38
Figura 18 – Diferentes componentes que compõem o sistema.....	40
Figura 19 – Diagrama de módulos do sistema.....	43
Figura 20 – Diagrama de classes da camada de serviços.....	45
Figura 21 – Diagrama de sequência "Fazer simulação de taxas de juro"	46

Figura 22 – Diagrama de sequência "Visualizar comunicados no idioma inglês" .	46
Figura 23 – Diagrama de sequência "Fazer conversão de moeda"	47
Figura 24 – Tabela com os detalhes de cada operação dos serviços RESTful desenvolvidos	49
Figura 25 – Estrutura dos diferentes componentes da aplicação <i>mobile</i>	52
Figura 26 – Estrutura do projeto da aplicação <i>mobile</i> no IDE de desenvolvimento Eclipse	53
Figura 27 – Estrutura do projeto da aplicação <i>mobile</i> no IDE de desenvolvimento Xcode.....	53
Figura 28 – Diagrama com todas as <i>Views</i> , <i>Controllers</i> e <i>Services</i> criados na aplicação.	55
Figura 29 – Aplicação <i>mobile</i> quando acedida por um <i>tablet</i>	56
Figura 30 – Aplicação <i>mobile</i> quando acedida por um <i>smartphone</i>	56
Figura 31 – Protótipo do site desenvolvido quando acedido num desktop.....	60
Figura 32 – Protótipo do <i>site</i> desenvolvido quando acedido num <i>tablet</i>	60
Figura 33 – Protótipo do <i>site</i> desenvolvido quando acedido num <i>smartphone</i>	60
Figura 34 – Gráfico do nível de sucesso das diferentes tarefas baseada na dificuldade para completar uma tarefa	68
Figura 35 – Gráfico com as pontuações SUS de todos os utilizadores que responderam ao questionário.....	69

Lista de Tabelas

Tabela 1 – Comparação das diferentes características para as três abordagens de desenvolvimento de aplicações <i>mobile</i> apresentadas (Nativa, <i>Web</i> e Híbrida).....	8
Tabela 2 – Diferenças entre as abordagens de desenvolvimento <i>web</i> (AWD e RWD).	11
Tabela 3 – Calendarização do Projeto.....	22
Tabela 4 – Requisitos funcionais e de teste da componente <i>web</i>	30
Tabela 5 – Requisitos funcionais e de teste da componente <i>mobile</i>	31
Tabela 6 – Requisitos funcionais e de teste da API <i>Web</i>	32
Tabela 7 – Condições de execução do primeiro teste de carga.....	63
Tabela 8 – Resultados da execução do primeiro teste de carga	64
Tabela 9 – Detalhes dos erros do primeiro teste de carga.....	64
Tabela 10 – Condições de execução do segundo teste de carga	65
Tabela 11 – Resultados da execução do segundo teste de carga	65
Tabela 12 – Condições de execução do terceiro teste de carga	65
Tabela 13 – Resultados da execução do terceiro teste de carga.....	66

Capítulo 1

Introdução

Neste capítulo são abordadas as principais motivações deste Projeto de Mestrado em Engenharia Informática (PEI) e os seus principais objetivos. É também apresentada a notação adotada na estruturação do documento e um sumário dos capítulos seguintes.

1.1 Motivação

O atual contexto do *site* institucional do Banco de Portugal é substancialmente diferente do projetado em 2008 quando foi desenvolvido. Nessa altura, a disponibilização de um único *site* garantia a qualidade e acessibilidade de informação a todo o público-alvo, no entanto, por força dos avanços tecnológicos no sector móvel, isto já não se verifica. Atualmente as pressões exercidas sobre o Banco decorrem de novos utilizadores, novas necessidades e novos suportes de informação, obrigando a que a informação seja acessível, compreensível e útil a todos.

O crescimento dos dispositivos móveis foi o principal fator que contribuiu para estas mudanças, pois muitos dos novos utilizadores visitam *sites* usando um *smartphone* ou um *tablet* e o *site* institucional não está preparado para este tipo de acessos. A atual fragmentação de plataformas e a necessidade do Banco em acompanhar as tendências e os avanços tecnológicos foram outros fatores igualmente importantes e serviram de alavanca para o trabalho a desenvolver no decurso deste PEI.

Com o desenvolvimento de soluções *mobile* prevê-se uma melhoria significativa na interação do Banco com os seus utilizadores. Desta forma, podemos entender a disponibilização das aplicações numa lógica de serviço público.

Contudo, as principais motivações deste PEI passam por, primeiro, apontar as diretrizes para a reestruturação *web* prevista para 2015 e 2016 apresentando um protótipo do *site* institucional, e segundo, disponibilizar aos utilizadores uma nova forma de

aceder a conteúdos do Banco de Portugal, através da disponibilização de uma aplicação móvel.

1.2 Objetivos

São dois os objetivos principais do trabalho conducente a este PEI. O primeiro é desenvolver um protótipo do *site* institucional do Banco de Portugal focado apenas na implementação de alguns serviços da página inicial. Deste modo, o protótipo do *site* deverá cumprir com os requisitos propostos, e ser capaz de se adaptar a diferentes ambientes, não limitando os seus utilizadores a sistemas operativos ou dispositivos, ou seja, fornecer uma experiência de visualização otimizada e uma leitura e navegação fáceis com um mínimo de redimensionamento e visionamento para uma ampla gama de dispositivos. Este servirá como prova de conceito para uma reestruturação *web* futura.

O segundo é desenvolver uma aplicação para disponibilizar nas *App Stores* que funcione de forma independente do *site* institucional, mas ofereça alguns dos seus principais serviços. A aplicação deverá ser portátil nas principais plataformas do mercado móvel (iOS e Android), permitindo assim um novo meio de acesso a conteúdos do Banco de Portugal.

Todo o trabalho desenvolvido deverá ter em conta as principais regras de usabilidade e navegabilidade, o que implica testar as aplicações em algumas fases de desenvolvimento com utilizadores de forma a obter feedback.

Grande parte do trabalho envolverá essencialmente a camada de apresentação ou seja *front-end*, no entanto o acesso a serviços tanto no protótipo como na aplicação será feito através de *web services* que também serão desenvolvidos no decorrer do projeto, criando assim um terceiro objetivo complementar.

1.3 Notação adotada

A notação adotada na escrita deste documento foi a língua portuguesa com o novo acordo ortográfico. As expressões que estejam noutra idioma estarão em itálico.

Para a bibliografia e citações utilizou-se o estilo descrito na Norma ISO 690 com o uso de referência numérica.

1.4 Organização do documento

Este documento encontra-se dividido em oito capítulos principais, cada um abordando diferentes aspetos do projeto.

No primeiro capítulo são descritas as motivações deste PEI, a importância dos projetos e os objetivos principais de cada solução desenvolvida.

No segundo capítulo é apresentado o estado da arte no desenvolvimento de aplicações móveis e quatro casos de estudo de aplicações similares desenvolvidas por outros Bancos Centrais.

No terceiro capítulo é apresentada a instituição de acolhimento, o processo de desenvolvimento do projeto e a calendarização feita na realização do mesmo.

O quarto capítulo descreve a fase de análise e levantamento de requisitos e toda a documentação produzida durante esse processo.

No quinto capítulo são apresentados todos os diagramas produzidos para auxiliar a fase de implementação, bem como o desenho do sistema.

O capítulo seis é dedicado à fase de implementação desde as decisões na conceção do sistema, escolha de plataformas e metodologias e o processo de desenvolvimento.

No sétimo capítulo é descrita a fase de testes, detalhando todos os testes realizados e resultados obtidos.

O oitavo e último capítulo é dedicado às conclusões finais do projeto, incluindo as limitações verificadas e uma perspetiva de trabalho a ser realizado no futuro.

Capítulo 2

Trabalho relacionado

Neste capítulo é apresentado o estado da arte no desenvolvimento de soluções móveis e ainda quatro casos de estudo de outros Bancos Centrais.

2.1 Estado da Arte

Nesta secção são apresentadas as diferentes abordagens no desenvolvimento e *design* de soluções móveis, as quais correspondem ao ponto de partida do trabalho a realizar no decorrer desta dissertação. Como tal, interessa conhecê-las assim como às suas vantagens e desvantagens. Adicionalmente, são descritas tecnologias relevantes como HTML5 (e as principais novidades face às versões anteriores) e algumas *frameworks* que facilitam e simplificam o desenvolvimento para *mobile*.

2.1.1 Aplicações Nativas vs. Web vs. Híbridas

Nos dias que correm, cada vez mais as organizações reconhecem a necessidade de desenvolver soluções para *mobile*. No entanto, o processo de escolha da abordagem de desenvolvimento (nativa, *web* ou híbrida), assenta em múltiplas variáveis, tais como, orçamento, requisitos, público-alvo, tipo de funcionalidades entre outros. Todas as abordagens têm benefícios e limitações quando comparadas, e encontrar a estratégia que melhor se enquadra com as necessidades de um projeto não é um processo determinístico.

Abaixo são apresentadas as principais vantagens e desvantagens das diferentes abordagens de desenvolvimento [1-3].

Aplicações Nativas

As aplicações nativas são especificamente concebidas para serem executadas num sistema operativo de um determinado dispositivo móvel e necessitam ser instaladas. São aplicações desenvolvidas exclusivamente para uma determinada plataforma (Android, iOS, Windows Phone, etc...), o que permite explorar ao máximo a potencialidade de cada uma destas e de cada dispositivo. Utilizam os componentes básicos da interface de utilizador de cada plataforma, e aumentam a complexidade e os custos de desenvolvimento, pois não permitem a reutilização de código entre plataformas. Uma equipa que desenvolve aplicações para iOS, normalmente não tem a mesma proficiência em Android ou Windows Phone. Muitas vezes uma aplicação multiplataforma pode exigir ter mais do que uma equipa trabalhando em conjunto no desenvolvimento destas aplicações para que a experiência do utilizador seja a mais próxima possível nas diferentes plataformas. Basicamente, o volume de trabalho, manutenção, correção de erros, atualizações, entre outros, vai aumentando à medida que queremos maior portabilidade entre plataformas.

O desenvolvimento de aplicações nativas é usado normalmente quando:

- Existem elevados requisitos de desempenho e usabilidade;
- É necessário usar grandes quantidades de animação, como por exemplo no desenvolvimento de jogos;
- É necessário aceder a APIs específicas dos dispositivos, como por exemplo o acelerómetro;
- A aplicação é exclusivamente para os utilizadores de uma determinada plataforma;
- Existem recursos suficientemente qualificados nas diferentes plataformas e o orçamento é suficiente para pagar estes recursos.

Aplicações Web

As aplicações *web* são sempre acedidas através de um *browser* e normalmente podem ser utilizadas em todos os dispositivos móveis, ou seja, não é necessário a sua instalação, apenas requerem um *browser* e acesso à internet, o que simplifica muito o processo de atualizações. No entanto, estas aplicações necessitam reconhecer o ambiente em que são executadas e ajustarem-se a este, isto é, a visualização deve ser diferente de dispositivo para dispositivo. São de mais fácil desenvolvimento do que as aplicações nativas porque são automaticamente multiplataforma, ou seja, não necessitam de conhecimento de diferentes linguagens de programação, apenas linguagens *web* (HTML, CSS e JavaScript) o que reduz naturalmente os custos.

As principais desvantagens destas aplicações são o facto do acesso a recursos de hardware ser limitado, apenas permitem integração através de HTML5 de alguns recursos como a câmara e o GPS, e não suportarem o modo *offline* nem o modo *Full-Sreen* por necessitarem sempre de um *browser* e acesso a internet.

O desenvolvimento de aplicações *web* normalmente é usado quando:

- A aplicação necessita operar em diferentes plataformas, e existem restrições de tempo e orçamento;
- É necessário colocá-la no mercado num curto espaço de tempo;
- Não existem recursos qualificados para o desenvolvimento nativo;
- O conhecimento de desenvolvimento *web* garante o cumprimento de todos os requisitos;
- Existe pouco conhecimento relativamente às características das diferentes plataformas e dispositivos móveis.

Aplicações Híbridas

A terceira alternativa são as aplicações híbridas, que combinam o desenvolvimento nativo com a tecnologia *web*. A parte nativa da aplicação usa as APIs da plataforma para criar um motor de renderização HTML que funciona como ponte entre o *browser* e as APIs do dispositivo. Esta ponte permite aproveitar quase todos os recursos que os dispositivos modernos oferecem, como por exemplo o acelerómetro e a câmara. O desenvolvimento da parte nativa pode ser feito com codificação própria, ou através do uso de *frameworks*. Se for feita com codificação própria exige programar o encapsulamento da componente *web* para que depois corra nas diversas plataformas, por outro lado o uso de *frameworks* simplifica o processo e apenas requer o desenvolvimento da componente *web*. A parte *web* por sua vez é desenvolvida com HTML5, CSS3 e JavaScript, sendo depois este código empacotado e armazenado localmente no dispositivo fazendo com que as aplicações pareçam nativas.

A *Framework* mais popular para o desenvolvimento destas aplicações é o PhoneGap [4] que consiste num conjunto de APIs que permite que a aplicação aceda aos recursos nativos dos dispositivos, no entanto, existem muitas outras *frameworks* similares no mercado. Uma característica importante deste tipo de aplicações é o facto de estas conseguirem aproveitar o melhor das duas abordagens anteriores, por um lado, a possibilidade de apenas utilizar tecnologias *web*, e por outro, a possibilidade de usar os recursos intrínsecos de cada dispositivo, conseguindo um equilíbrio a nível de desempenho, recursos e orçamento comparativamente às outras duas abordagens.

O desenvolvimento de aplicações híbridas normalmente é usado quando:

- É necessário aceder apenas a alguns recursos do dispositivo, como por exemplo o acelerómetro ou GPS;
- Queremos apenas desenvolver a aplicação uma vez para diferentes plataformas;
- Pretendemos conseguir o melhor das duas abordagens anteriores;
- Queremos ter uma aplicação que use recursos nativos, mas o ciclo de desenvolvimento seja curto.

Comparação

Em suma, só um bom levantamento de requisitos e um bom conhecimento das diferentes abordagens de desenvolvimento pode garantir uma escolha acertada que cumpra com todos os requisitos. No entanto, conhecer as diferentes abordagens implica compará-las segundo um conjunto de características [1-3]. Na Tabela 1 é feita a comparação das três abordagens de desenvolvimento segundo as características abaixo descritas.

- **Monetização** – As aplicações nativas e híbridas têm um modelo de negócio já implementado pelas *App Stores*, uma vez que podem ter ou não opções de compra, o que pode ser mais vantajoso para o distribuidor que não tem de se preocupar com este fator. Já as aplicações *web* requerem que se implementem os mecanismos para a sua comercialização.
- **Distribuição** – As aplicações nativas e híbridas são distribuídas nas *App Stores* das diferentes plataformas. As aplicações *web* não dependem de terceiras entidades, apenas são disponibilizadas sob a forma de uma página *web*. Neste aspeto as aplicações *web* têm vantagem pois a distribuição é centralizada.
- **Manutenção** – As aplicações nativas e híbridas requerem que se atualize nas *App Stores* as novas versões das aplicações. As aplicações *web* garantem que os utilizadores estão sempre a utilizar a versão mais recente da aplicação, o que é mais vantajoso.
- **Portabilidade** – As aplicações nativas apenas estão disponíveis para as plataformas para as quais foram desenvolvidas, enquanto as aplicações *web* são compatíveis com todas as plataformas. Por sua vez, as aplicações híbridas podem estar disponíveis para um grande número de plataformas.
- **Funcionalidade** – Atualmente as aplicações nativas têm mais funcionalidades que as aplicações *web* e híbridas, no entanto, esta desvantagem está a ser atenuada por força do trabalho realizado pelo *World Wide Web Consortium* (W3C), na realização de APIs que oferecem cada vez mais funcionalidades dos dispositivos às aplicações *web* e híbridas. Portanto, neste aspeto, as aplicações

nativas conseguem utilizar mais recursos dos dispositivos e implementar mais funcionalidades, enquanto as aplicações híbridas apenas conseguem usar parte destes recursos e as aplicações *web* um número ainda mais reduzido.

- **Ligação à Internet** – Se por um lado as aplicações *web* estão totalmente dependentes do acesso à internet, as aplicações nativas e híbridas podem funcionar com ou sem esta ligação dependendo do tipo de aplicação.
- **Desempenho** – O desempenho pode ser analisado de três pontos de vista: latência, fluidez e tempo de execução. Relativamente à latência, as aplicações nativas e híbridas têm um melhor desempenho, pois já estão presentes no dispositivo e não é necessário qualquer transferência. A nível de fluidez, as aplicações nativas apresentam a vantagem de não necessitar de qualquer *browser* na sua utilização, no entanto, as aplicações híbridas também não usam diretamente o *browser*, mas sim o seu motor, pelo que não são tão afetadas como as aplicações *web*. Em termos de tempo de execução, as aplicações nativas possuem valores mais baixos que as outras duas, pois a eficiência de uma implementação nativa supera a de uma implementação *web*.
- **Orçamento** – Quando as aplicações têm de ser compatíveis com múltiplas plataformas, as aplicações nativas exigem mais volume de trabalho, manutenção, correção de erros e atualizações, portanto o desenvolvimento destas é mais caro relativamente as aplicações híbridas e *web*. Por outro lado quando apenas se desenvolve para uma única plataforma, o valor é semelhante em todas as abordagens.
- **Know-how** – As aplicações *web* exigem apenas que a equipa de desenvolvimento saiba tecnologias *web* (HTML5, CSS e JavaScript), enquanto que as aplicações nativas exigem o conhecimento específico das linguagens de cada plataforma. Por sua vez as aplicações híbridas podem utilizar só tecnologias *web*, ou adicionalmente codificação em outras linguagens.

	Aplicações Nativas	Aplicações Híbridas	Aplicações Web
Monetização	Melhor	Melhor	Pior
Distribuição	Pior	Pior	Melhor
Manutenção	Pior	Pior	Melhor
Portabilidade	Pior	Intermédio	Melhor
Funcionalidade	Melhor	Intermédio	Pior
Ligação a internet	Melhor	Melhor	Pior
Desempenho	Melhor	Intermédio	Pior
Orçamento	Pior	Melhor	Melhor
Know-how	Pior	Intermédio	Melhor

Tabela 1 – Comparação das diferentes características para as três abordagens de desenvolvimento de aplicações *mobile* apresentadas (Nativa, Web e Híbrida).

2.1.2 Aplicações web (*Responsive vs. Adaptive Design*)

Atualmente existem duas abordagens principais quando se fala em *design* de aplicações web para *mobile*: *responsive design* e *adaptive design*. Ambas são focadas na aparência visual e acessibilidade de informação dependendo das dimensões e características do dispositivo de acesso. No entanto, existem algumas diferenças que são importantes especificar.

Responsive Web Design (RWD)

Responsive Web Design [5,6] é uma abordagem de *design web* que permite desenvolver *sites* que forneçam uma experiência de visualização e navegação otimizada com um mínimo de redimensionamento, para uma ampla gama de dispositivos. Tem a característica de alterar a forma como os elementos de um *site* são exibidos, no entanto, não remove elementos nem altera as suas funcionalidades. Um bom projeto RWD corre em qualquer plataforma, seja qual for o dispositivo onde está a ser visualizado, independentemente da resolução ou orientação do mesmo. Um exemplo de um projeto RWD é a aplicação InflexionInteractive que adapta os seus conteúdos para todo o tipo de dispositivos, como ilustrado na Figura 1.



Figura 1 – Aplicação RWD InflexionInteractive.com em execução num desktop, tablet e smartphone.

Um projeto RWD adapta a sua exibição para o ambiente de visualização através do uso de grades proporcionais e fluídas, imagens flexíveis e *media queries* CSS3. As grades fluídas são estruturas de duas dimensões compostas por eixos verticais e horizontais que se cruzam e permitem a um *designer* alinhar e organizar os elementos de uma aplicação web. Requerem que as dimensões dos elementos sejam definidas por unidades relativas como percentagem, em vez de unidades absolutas. Por outro lado o uso de imagens flexíveis é muito importante. Devem ser usadas sempre imagens que tenham dimensões em unidades relativas, de modo a prevenir que estas saiam fora dos

seus elementos HTML. Por último, o uso de *media queries* é essencial pois permite aplicar diferentes regras de estilo CSS com base nas características de cada dispositivo onde o *site* está a ser exibido. Basicamente a ideia por detrás do conceito é, “Um *site*, um design, um conteúdo, em incontáveis dispositivos” [7]. Tem ainda a curiosidade de ser a abordagem recomendada pela Google [8] principalmente por questões de indexação, uma vez que apenas existe um URL facilitando assim o trabalho dos motores de pesquisa.

Principais Características

- Melhora a visibilidade pelos motores de pesquisa;
- Otimiza a ergonomia, pois mantém intacta a qualidade de informação;
- Requer a reconstrução dos *sites* já existentes;
- Pode ter mau desempenho em alguns dispositivos;
- Mantém o mesmo ficheiro HTML para qualquer dispositivo.

Adaptive Web Design (AWD)

Adaptive Web Design [9,10] é uma abordagem de *design* que segue exatamente os mesmos objetivos de RWD, ou seja, oferecer uma experiência de visualização e navegação otimizadas com um mínimo de redimensionamento para uma ampla gama de dispositivos, contudo a metodologia por detrás de um projeto adaptativo é bastante diferente. Um exemplo de um projeto AWD é a aplicação AmericanAirlines ilustrada na Figura 2.



Figura 2 – Aplicação AWD AmericanAirlines.com em execução num desktop, tablet e smartphone.

Esta metodologia pode usar o servidor ou o cliente para detetar o dispositivo, no entanto, quando a deteção é feita no lado do servidor, são fornecidos ficheiros HTML otimizados para os diferentes dispositivos, o que permite o carregamento das páginas mais rápido. O uso desta abordagem é muito útil quando já existe um *site* tradicional e queremos uma solução *mobile* que não implique refazer o que está feito. Normalmente

as aplicações desenvolvidas segundo esta abordagem têm dois *templates* distintos para cada tipo de acesso, um para acesso normal, e outro para acesso móvel, no entanto, podem ainda ser mais otimizadas e oferecer diferentes *templates* para cada dispositivo.

Principais Características

- Não requer a reconstrução dos *sites* já existentes;
- Requer conhecimentos avançados de JavaScript e CSS;
- Obtém boa performance em quase todos os dispositivos;
- Fornece informação otimizada para cada dispositivo;
- Deteção do dispositivo feita tanto no servidor como no cliente.

Comparação

Em suma, ambas as abordagens visam melhorar a qualidade e acessibilidade de informação, no entanto, com estratégias diferentes. Abaixo na Tabela 2 são apresentadas as principais diferenças entre as abordagens *Responsive* e *Adaptive Design* [11,12].

	AWD	RWD
Identificação do dispositivo	Pode ser feita tanto do lado do cliente como do servidor.	É feita apenas no lado do cliente através de <i>media queries</i> CSS3.
Ficheiros HTML	Diferentes ficheiros HTML consoante o dispositivo.	Apenas é fornecido um ficheiro HTML.
Otimização de Conteúdos	Descarregados apenas os conteúdos necessários.	Todos os conteúdos são descarregados.
Otimização dos dispositivos	Diferentes <i>templates</i> para os diferentes dispositivos.	Um único <i>template</i> para todos os dispositivos.
Performance	Página carrega rapidamente, porque apenas os elementos necessários são descarregados.	Página carrega mais devagar, porque existem mais elementos para descarregar.
Implementação	Permite implementar em <i>sites</i> que já existem.	É necessário reconstruir.
Desenvolvimento	Exige conhecimentos avançados de JavaScript para manipular de forma eficiente o DOM (<i>Document Object Model</i>).	Maiores dificuldades nas fases de teste, porque o mesmo <i>template</i> serve todos os dispositivos. Implica testar iterativamente todas as funcionalidades nos diferentes dispositivos.

Tabela 2 – Diferenças entre as abordagens de desenvolvimento *web* (AWD e RWD).

2.1.3 HTML5

O HTML5 [13,14] é uma linguagem de marcação utilizada para estruturar e representar conteúdo para a *World Wide Web*. Trata-se da versão mais recente da linguagem HTML e trouxe várias mudanças que se têm verificado serem essenciais no mundo *web*. Uma das principais novidades, nesta nova versão, é a inclusão de novas funções sintáticas, como por exemplo as tags <video>, <audio>, <canvas> e <svg> que vieram enriquecer a linguagem e tornar mais fácil a inclusão e manipulação de conteúdos gráficos e multimédia na *web*, sem ter de usar outras APIs ou *plugins* proprietários. No entanto, outros novos elementos foram também adicionados, como a inclusão das tags, <header>, <section>, <article>, <nav>, <aside> e <footer> que vieram enriquecer o conteúdo semântico dos documentos, uniformizando assim a criação de secções específicas. Relativamente às APIs e aos modelos de objetos de documentos (DOM), estes são agora uma parte fundamental da especificação do HTML5. É ainda possível nesta nova versão definir com algum detalhe o processamento necessário para que erros de sintaxe em documento inválidos sejam tratados uniformemente em todos os *browsers*. Contudo, existem outras novidades que potenciam o HTML5 no desenvolvimento de soluções para *mobile*, pelo que de seguida são descritas as mais importantes e significativas no contexto deste projeto.

Suporte *offline* e armazenamento

O suporte *offline* e armazenamento permite que as aplicações possam trabalhar sem necessidade de acesso constante à internet. Basicamente o HTML indica ao *browser* quais os elementos que são necessários guardar localmente, como por exemplo imagens, dados do utilizador, entre outros, permitindo assim que a aplicação funcione em modo *offline*.

Canvas

Canvas é uma área delimitada pelo uso da tag <canvas> que especifica uma área para renderização dinâmica para imagens interativas, tabelas, gráficos, componentes de jogos, entre outros, através de linguagens de programação dinâmica como por exemplo JavaScript. Esta nova possibilidade é muito enriquecedora para o desenvolvimento *mobile* na medida em que evita o uso de outros *plugins* ou tecnologias como o flash que não funcionam muito bem neste contexto.

Desde a sua primeira especificação em 2008 que o HTML5 tem vindo a evoluir e ganhar novas funcionalidades. A inclusão de novas APIs, o controle embutido de conteúdo multimédia, o suporte *offline* e armazenamento local são algumas das novidades que estão a tornar o HTML5 um padrão consensual entre os grandes *players* do mercado *mobile*. No entanto, apesar do HTML5 já ser uma das linguagens mais utilizadas no contexto *web* e de muitas novas funcionalidades resolverem algumas limitações da versão anterior no desenvolvimento para *mobile*, nem tudo é positivo e ainda existem muitos problemas e limitações que precisam ser resolvidos para que se estabeleça como uma linguagem definitiva. Com este objetivo em mente o *World Wide Web Consortium (W3C)* [13], estipulou para o final de 2014 a conclusão do HTML5, sendo que, tudo o que não for concluído até então, será incluído nas versões seguintes.

2.1.4 Frameworks para desenvolvimento *mobile*

Nesta secção são apresentadas de forma breve algumas das principais *frameworks* e bibliotecas baseadas em tecnologias *web* que facilitam e simplificam o desenvolvimento para *mobile*.

PhoneGap/Apache Cordova

O phoneGap [4] é uma *framework* que permite desenvolver aplicações híbridas usando apenas código HTML5, CSS3 e JavaScript. Possui suporte para alguns recursos dos dispositivos como: acelerómetro, câmara, GPS, rede, armazenamento, entre outras. Trata-se de uma *framework* adquirida pela Adobe Systems, mas que foi disponibilizada à Apache com o nome Apache Cordova, para que continue a evoluir como um projeto *Open Source*.

Sencha Touch

Sencha Touch [15] é uma biblioteca JavaScript, que permite desenvolver interfaces de utilizador. É usada essencialmente para desenvolver interfaces de utilizador de aplicações *web* móveis que se parecem com aplicações nativas. É totalmente baseada em padrões da *web*, como HTML5, CSS3 e JavaScript.

Twitter Bootstrap

O Twitter Bootstrap [16] é uma *framework Open Source* criada para facilitar o desenvolvimento *web*. Possui diversas funcionalidades como *layouts responsive*, sistema de grades fluídas, e diversos modelos HTML e CSS, como formulários, botões, navegação, e muitos outros componentes da interface, permitindo a todo o tipo de

programadores desenvolver interfaces de utilizador agradáveis. É também uma excelente *framework* de desenvolvimento de aplicações móveis uma vez que oferece diversas funcionalidades que ajudam a criação deste tipo de interfaces.

jQuery Mobile

É uma *framework* para desenvolvimento móvel construída sobre a sólida fundação do jQuery que permite desenvolver interfaces de utilizador simples para todo o tipo de aplicações. Oferece alguns modelos já desenvolvidos com HTML5, CSS3 e JavaScript que têm uma aparência quase idêntica a modelos nativos [17].

Modernizr

Modernizr [18] é uma pequena biblioteca JavaScript que deteta a disponibilidade de implementações dos recursos HTML5 e CSS3 nos diferentes *browsers*, ou seja, permite identificar se um *browser* implementa ou não um determinado recurso. O uso desta biblioteca ajuda os programadores a tirar proveito dos novos recursos que os *browsers* suportam e ainda ajuda a criar salvaguardas para *browsers* que não possuem esse suporte.

2.2 Casos de Estudo

Como o projeto conducente a este PEI consiste em desenvolver soluções *mobile* para o Banco de Portugal, surgiu a ideia de se fazer uma análise de *benchmarking* junto de alguns dos principais Bancos Centrais no mundo com o objetivo de identificar soluções e estratégias tomadas na disponibilização de conteúdos móveis.

Foram feitas pesquisas em diversos Bancos Centrais, no entanto, apenas alguns desenvolveram soluções móveis, os quais são apresentadas abaixo.

2.2.1 Banco de França

O Banco de França a nível de disponibilização de conteúdos para acesso móvel, tem uma aplicação *mobile* nativa, disponível tanto no Google Play como na App Store. Trata-se de uma aplicação que oferece apenas alguns dos principais serviços disponíveis no *site* institucional. A aplicação disponibiliza ao utilizador as seguintes funcionalidades:

- **Notícias** – Apresenta uma lista das últimas notícias;
- **Discursos do governador** – Apresenta uma lista com os discursos do governador;
- **Publicações e estatísticas** – Permite consultar os principais artigos publicados e documentos estatísticos;
- **Defesa do consumidor** – Informação de direitos e deveres do consumidor;
- **Informações práticas** – Consultar a localização geográfica de todas as agências e filiais, bem como os contactos das mesmas.
- **Taxas** – Permite consultar os valores das diferentes taxas e a sua evolução diária. E ainda disponibiliza um conversor de moeda que permite fazer o câmbio de uma moeda para outra.

É uma aplicação apenas de consulta que disponibiliza alguns dos conteúdos mais importantes do ponto de vista do negócio, disponibiliza também um conversor de moeda que permite efetuar os câmbios de uma moeda para outra. Relativamente à interface, todos os menus são claros e intuitivos, de acordo com as regras básicas de usabilidade, o que permite uma aprendizagem rápida. Em suma, é o tipo de aplicação que se enquadra nas pretensões do Banco de Portugal, relativamente aos objetivos do estágio.

Nas Figuras 3 e 4 estão ilustradas duas vistas da aplicação nativa do Banco de França, a vista dos discursos do Governador e a vista de menus.



Figura 3 – Vista de menus da aplicação do Banco de França



Figura 4 – Vista dos discursos do governador da aplicação do Banco de França

2.2.2 Banco da Suécia

O Banco Central da Suécia desenvolveu uma aplicação *web responsive*, para a disponibilização dos seus conteúdos quer em *desktops* quer em dispositivos móveis. Esta aplicação serve qualquer dispositivo, adaptando-se sempre às dimensões do mesmo, no entanto apenas considera duas quebras de *layout*, ou seja, dispositivos com tamanho superior a 866px, e dispositivos com tamanho inferior.

Na prática, quando acedido em *desktops* ou portáteis, os conteúdos são apresentados em cinco secções horizontais e quatro colunas verticais. Na primeira secção é exibido o menu e a barra de pesquisa, na segunda são apresentados quatro conteúdos diferentes referentes a informações de taxas e juro e outras taxas, na terceira os destaques, na quarta as notícias e a agenda e na última secção o rodapé que contém alguns links importantes relativamente ao *site*. Quando a aplicação é acedida em dispositivos móveis o menu é ocultado ficando visível apenas um menu *drop-down*, e os conteúdos passam a estar dispostas em apenas duas colunas, uns abaixo dos outros, sendo que, quanto menor for o dispositivo, menor é o tamanho das fontes de texto apresentadas.

Em suma, é uma aplicação estruturada para ser redimensionada e exibida em qualquer dispositivo, pois todos os elementos incluindo imagens e vídeos tem tamanhos proporcionais para que possam ser visualizados sem recurso a zoom. É uma abordagem que agrada às pretensões do Banco de Portugal, no entanto, no curto prazo é de difícil implementação, pois toda a organização precisa ser pensada implicando refazer todo o *site* institucional. Na Figura 5 está ilustrada a vista da página inicial do Banco da Suécia quando acedido num *desktop*.



Figura 5 – Vista de um desktop da página inicial do Banco Central da Suécia

2.2.3 Banco da Bélgica e do Japão

Os Bancos da Bélgica e do Japão seguiram uma estratégia semelhante para disponibilizar aos seus utilizadores acesso a conteúdos do *site* institucional através de dispositivos móveis. A estratégia passou por desenvolver um *site* otimizado para *mobile* diferente do *site* principal. Na prática ambos os Bancos apenas desenvolveram uma página com menus pensados para acesso móvel, pois ao aceder a um item destes menus o utilizador é redirecionado para o *site* principal que não está preparado para este tipo de acessos.

Este tipo de estratégia é extremamente fraca porque apenas resolve o problema de acesso a informação, já que a informação é sempre visualizada com recurso ao zoom, ou seja, é uma estratégia que não resolve os problemas de usabilidade e interação com o utilizador, pelo que nem deverá ser considerada.

Nas Figuras 6 e 7 abaixo estão ilustradas as páginas *mobile* do Banco da Bélgica e do Japão respetivamente.



Figura 6 – Vista do site *mobile* do banco do Japão quando acedido por um *smartphone*

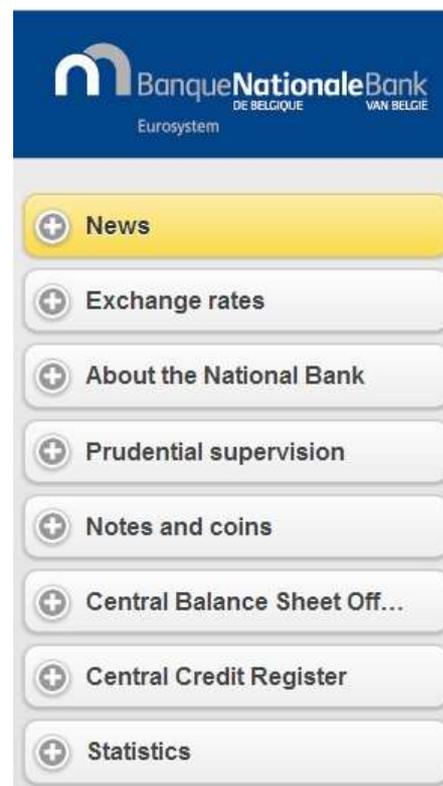


Figura 7 – Vista do site *mobile* do Banco da Bélgica quando acedido por um *smartphone*

2.2.4 Banco da Suíça

O *site* do Banco da Suíça disponibiliza os seus conteúdos para acesso móvel, através de uma aplicação *web adaptive*, na qual existem duas versões do *site*, uma versão normal e uma versão *mobile*.

Quando o acesso é feito num dispositivo móvel, o servidor deteta o dispositivo e fornece uma página HTML diferente apenas com as seguintes funcionalidades: notícias; estatísticas; notas e moedas; recrutamento; taxas de câmbio; taxas de juro. Por outro lado, se o acesso for com um *desktop* o servidor fornece as páginas normais com toda a informação.

Trata-se de uma página simples de navegar, e que funciona de forma igual em todos os dispositivos móveis, uma vez que, também existe uma componente *responsive* que permite o ajuste no lado do cliente para os diferentes tamanhos de ecrã. Toda a informação está facilmente acessível e é fácil de ler, o que torna a página fácil de aprender e navegar de acordo com as regras da boa usabilidade.

Na Figura 8 está ilustrada a imagem da vista da página inicial do Banco da Suíça quando acedido num dispositivo móvel.

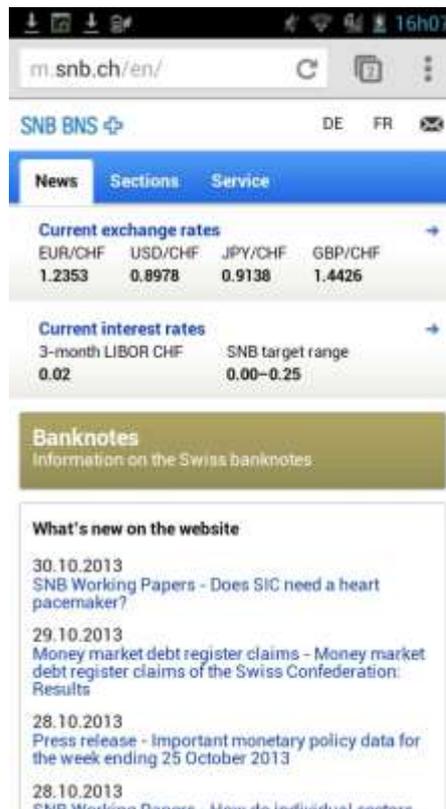


Figura 8 – Vista da página *mobile* do Banco da Suíça quando acedida por um *smartphone*

Capítulo 3

Enquadramento

Neste capítulo é apresentada a instituição de acolhimento, o processo de desenvolvimento e a calendarização do projeto.

3.1 Instituição de acolhimento

O Banco de Portugal, é o Banco Central da República Portuguesa e foi fundado em 19 de Novembro de 1846. Integra desde 1998 o Sistema Europeu de Bancos Centrais (SEBC) que é constituído pelo Banco Central Europeu (BCE) e pelos bancos nacionais da União Monetária (UEM) [19,20].

Compete ao Banco a supervisão prudencial das instituições de crédito e das sociedades financeiras, a emissão de notas e a colocação em circulação das moedas metálicas. Compete também ao Banco regular, fiscalizar e promover o bom funcionamento dos sistemas de pagamentos, gerir as disponibilidades externas do País e agir como intermediário das relações monetárias internacionais do Estado, bem como aconselhar o Governo nos domínios económico e financeiro. Cabe ainda ao Banco a recolha e elaboração das estatísticas monetárias, financeiras, cambiais e de balança de pagamentos.

O Banco de Portugal está organizado internamente segundo o organograma apresentado abaixo na Figura 9.

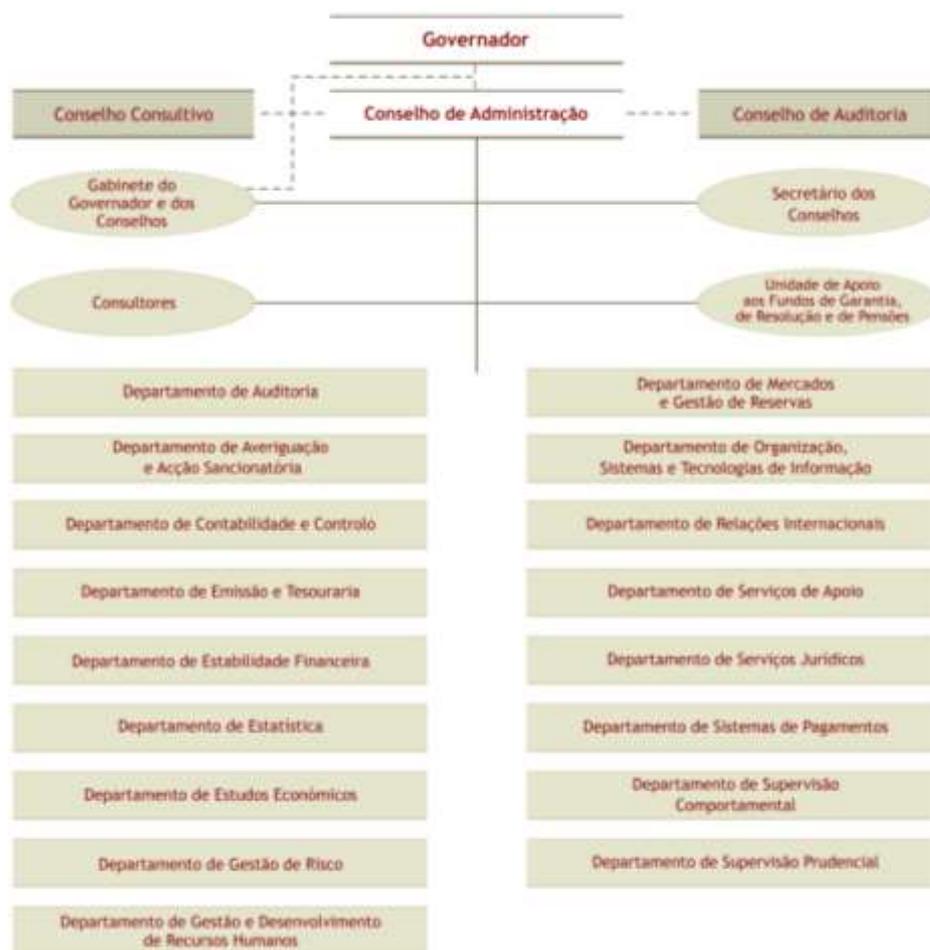


Figura 9 – Organograma geral do Banco de Portugal

Este PEI foi realizado no Departamento de Organização, Sistemas e Tecnologias de Informação que tem como missão prestar serviços ao Banco em matéria de Informação, numa perspetiva tecnológica e organizacional, prosseguindo esforços para garantir a sua qualidade, disponibilidade e coerência. Este PEI está integrado no Núcleo de Desenvolvimento e Evolução de Processos e Sistemas Empresariais do qual três pessoas estarão diretamente envolvidas com o mesmo:

- **Coordenador de Núcleo** – Responsável por aprovar todas as decisões tomadas no decorrer do projeto;
- **Dois Técnicos Assistentes** – Responsáveis pela orientação do projeto, ajuda e suporte na tomada de decisões.

3.2 Processo de Desenvolvimento

O processo de desenvolvimento adotado para os projetos propostos segue o modelo em V (*V-Model*) [21]. A adoção deste modelo de desenvolvimento fundamenta as melhores práticas em todas as fases do projeto, desde a análise, desenho, construção testes e manutenção. Na Figura 10 estão ilustradas as diferentes fases do projeto no modelo em V.

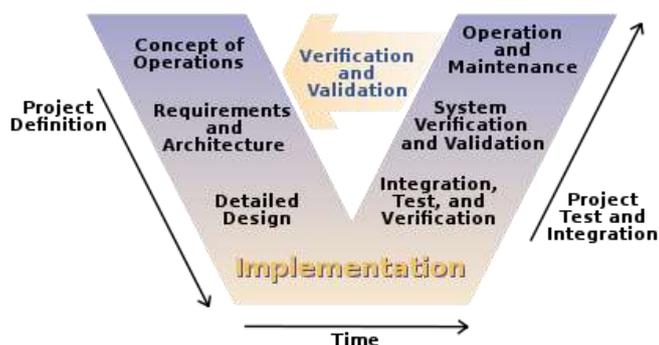


Figura 10 – Diferentes fases do projeto no Modelo em V

Trata-se de um modelo que enfatiza a importância de considerar os testes em todas as fases do processo, ao invés de realizar testes só no final. Deste modo garante um processo de testes mais rápido e uma entrega final com menos defeitos.

O uso deste modelo oferece ainda um esquema detalhado de como concretizar o projeto, e permite maior flexibilidade e iteratividade nos melhoramentos em cada etapa, possibilitando ter um produto final mais sólido e com melhor qualidade.

3.3 Calendarização do Projeto

A calendarização do projeto precisou ser reajustada relativamente à feita inicialmente, para que os objetivos propostos fossem cumpridos. A primeira calendarização juntava a fase de análise e desenho, o que provocou um atraso nessa fase. Assim, foi necessário dividir as várias tarefas em mais fases com tempos mais realistas, e reorganizar o restante planeamento para que o projeto fosse entregue nos tempos previstos. A nova calendarização definiu alguma folga nas fases de teste, de forma a prevenir atrasos em tarefas que não corressem como planeado. Esta folga revelou-se extremamente útil já que a passagem a produção dos *web services* foi um processo complicado que obrigou a refazer muito código pré-existente atrasando o plano.

Em suma, a calendarização serviu de linha orientadora ajudando a que os atrasos não afetassem a entrega final do projeto, que acabou por ser cumprida nos tempos

previstos, no entanto, a última tarefa do plano (tarefa 23) não foi ainda realizada, porque o Banco de Portugal decidiu reagendá-la para mais tarde depois da apresentação da aplicação ao Governador do Banco de Portugal. Abaixo, na Tabela 3 e Figura 11 estão ilustradas a calendarização e o mapa de Gantt do projeto.

Id	Nome da tarefa	Duração	Início	Fim	Precedências
1	Análise de requisitos	32 Dias	Seg. 07/10/13	Sex. 08/11/13	
2	Análise de Benchmarking e tecnologias de desenvolvimento	15 Dias	Seg. 07/10/13	Ter. 22/10/13	
3	Levantamento de Requisitos	13 Dias	Ter. 22/10/13	Seg. 04/11/13	2
4	Escrita do relatório de requisitos	4 Dias	Seg. 04/11/13	Sex. 08/11/13	3
5	Desenho técnico do sistema	12 Dias	Sex. 08/11/13	Qua. 20/11/13	1
6	Criação de documentação de desenho	12 Dias	Sex. 08/11/13	Qua. 20/11/13	1
7	Desenvolvimento <i>Web services</i>	79 Dias	Qua. 20/11/13	Sex. 07/02/14	
8	Desenvolvimento de todos os serviços	65 Dias	Qua. 20/11/13	Sex. 24/01/14	
9	Testes unitários aos serviços	14 Dias	Sex. 24/01/14	Sex. 07/02/14	8
10	Desenvolvimento Componente <i>web</i>	31 Dias	Sex. 07/02/14	Seg. 10/03/14	7
11	Criação da aplicação <i>web</i> responsiva	20 Dias	Sex. 07/02/14	Qui. 27/02/14	
12	Testes de compatibilidade	6 Dias	Qui. 27/02/14	Qua. 05/03/14	11
13	Integração com os <i>web services</i>	5 Dias	Qua. 05/03/14	Seg. 10/03/14	11
14	Desenvolvimento Componente <i>mobile</i>	85 Dias	Seg. 10/03/14	Seg. 02/06/14	7
15	Criação da aplicação <i>web</i> responsiva	43 Dias	Seg. 10/03/14	Qui. 22/05/14	
16	Integração com os <i>web services</i>	15 Dias	Qui. 22/05/14	Sex. 06/06/14	15
17	Deploy e testes para iOS e Android	20 Dias	Sex. 06/06/14	Seg. 26/06/14	16
18	Passagem a produção	7 Dias	Seg. 26/06/14	Seg. 02/06/14	17
19	Testes	28 Dias	Seg. 02/06/14	Seg. 30/06/14	14
20	Testes de usabilidade	11 Dias	Seg. 02/06/14	Sex. 13-06/14	
21	Testes de sistema	6 Dias	Sex. 13-06/14	Ter. 19/06/14	
22	Testes de carga	11 Dias	Ter. 19/06/14	Seg. 30/06/14	
23	Colocação da aplicação nas Stores	9 Dias	Seg. 30/06/14	Qua. 09/07/14	19

Tabela 3 – Calendarização do Projeto

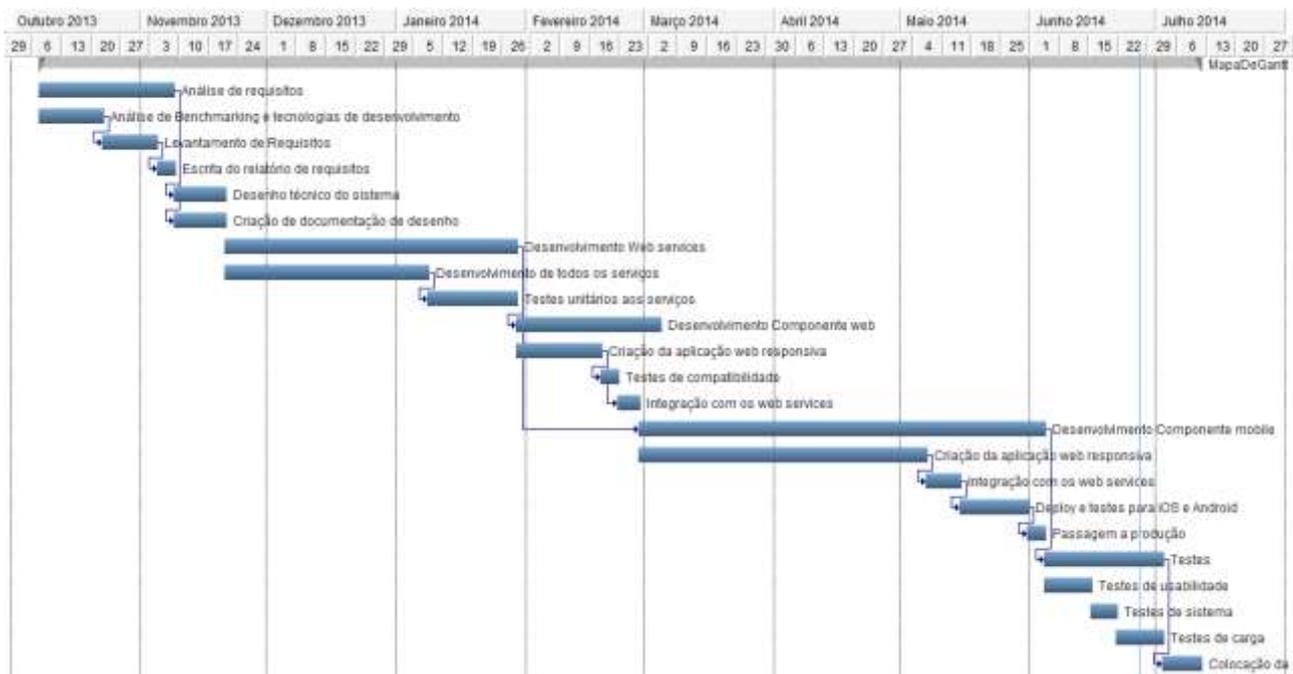


Figura 11 – Mapa de Gantt do projeto

Capítulo 4

Análise e especificação de requisitos

O completo entendimento desta fase é um ponto fundamental que pode determinar o sucesso ou fracasso de um projeto de software. Neste capítulo serão descritas as diferentes fases de análise e ainda toda a documentação produzida durante a mesma.

4.1 Análise de conteúdos e serviços

A primeira tarefa conducente à fase de análise foi fazer uma avaliação dos atuais serviços / conteúdos disponibilizados nos vários portais do Banco de Portugal e identificá-los como possíveis candidatos a disponibilizar num contexto móvel.

O Banco de Portugal é responsável por oito portais, que oferecem um extenso número de serviços, como tal, e para que a avaliação fosse o mais objetiva possível, foi necessário definir alguns critérios. Foram definidos quatro critérios fundamentais para a identificação destes serviços pela seguinte ordem de prioridade:

1. Serviços / conteúdos no Portal do *site* Institucional;
2. Serviços / conteúdos com maior número de acesso;
3. Serviço com apetência natural a incluir num contexto móvel;
4. Centralização de outras aplicações *mobile* existentes.

4.1.1 *Site* Institucional do Banco de Portugal

A identificação dos serviços mais importantes do *site* institucional, começou nas primeiras reuniões junto das áreas de negócio, nas quais foi obtida a informação de que os serviços mais importantes das diferentes valências do Banco de Portugal estavam destacados com uma presença na página inicial. A seleção destes conteúdos é da responsabilidade do Gabinete de Comunicação Institucional que se insere no Departamento do Gabinete de Apoio ao Governador (GAB), pelo que os conteúdos

identificados nesta fase também deverão ser aprovados pelo mesmo. Foram então identificados os seguintes serviços:

- Comunicados;
- Publicações;
- Taxas de câmbio;
- Taxas de juro;
- Indicadores económicos;
- Projeções económicas;
- Agenda;

4.1.2 Maior número de acessos

Na segunda fase da análise foi consultada a lista dos indicadores internet disponíveis no *site* da Intranet, com o objetivo de identificar os serviços mais acedidos pelos utilizadores. Após esta análise foram identificados os cinco serviços seguintes ordenados de forma descendente:

1. Conversor de Moeda;
2. Serviço para obter mapa de responsabilidades de crédito;
3. A página da área de Domínios e estatísticas cambiais, onde são exibidos as Taxas de referência Diárias e as Cotações do Ouro em barra.
4. A página das Intervenções Públicas;
5. A página dos Comunicados.

O conversor de moeda e a página da área de Domínios estatísticos estão incluídos no âmbito das taxas de câmbio já identificados na secção 4.1.1 acima, tal como os comunicados, o que reforça a importância dos mesmos para os utilizadores. Por outro lado, os outros dois serviços identificados nesta secção poderiam também ser vistos como candidatos, no entanto, apenas o serviço de intervenções públicas ficou na lista, uma vez que o serviço para obter mapa de responsabilidades de crédito, não se enquadra no contexto dos serviços móveis. Trata-se de um serviço de consulta sobre os créditos concedidos aos seus clientes e também sobre as responsabilidades de crédito potenciais que representam compromissos irrevogáveis, ou seja é um serviço que além de estar fora do contexto obriga a grandes requisitos de segurança, difíceis de garantir num ambiente móvel e fora do âmbito deste projeto.

4.1.3 Apetência natural para *mobile*

Na identificação dos serviços relativos ao critério 3, em primeiro lugar foi incluído o conversor de moeda já identificado na secção 4.1.2 acima. É o serviço mais acedido no *site* Institucional, e o tipo de serviço normalmente disponibilizado por outras instituições Bancárias. Foram ainda identificados os simuladores do Portal do Cliente Bancário listados abaixo:

- Simulador de Crédito a habitação;
- Simulador de Crédito aos consumidores;
- Simulador de Depósitos bancários;
- Simulador de Taxas de juro.

A inclusão destes serviços numa aplicação móvel permite que estejam disponíveis em modo *offline*, o que é uma mais-valia relativamente ao portal internet.

4.1.4 Centralização de outras aplicações *mobile*

O Banco de Portugal tem direcionado alguns esforços para garantir uma cobertura nos novos meios de comunicação, nomeadamente o *mobile*. Como parte destes esforços foi desenvolvido o BPstat Mobile (Sistema móvel de acesso online à Base de Dados Estatística Interativa do Banco de Portugal) e agora o projeto conducente a esta tese de mestrado. Uma forma de simplificar o acesso destes serviços é a disponibilização dos mesmos numa única plataforma. A aplicação a desenvolver deverá oferecer mecanismos de acesso fácil ao BPstat Mobile ou até mesmo incorporá-lo no contexto de uma única aplicação.

4.1.5 Conclusão

Esta análise foi extremamente importante porque permitiu conhecer o negócio e estruturar a lista dos potenciais serviços a disponibilizar num contexto móvel. No entanto, nem todos os serviços identificados foram incluídos na lista de requisitos, pois iriam aumentar a lista de trabalhos, tornando assim irrealista a conclusão do projeto no tempo previsto.

Os dados recolhidos foram então analisados em cooperação com o GAB, com o objetivo de elaborar a lista final de requisitos funcionais e não funcionais do sistema. A lista resultante desta análise não inclui todos os serviços pretendidos pelo GAB, no entanto o desenvolvimento do projeto deverá prever a inclusão de mais funcionalidades nas versões futuras. A lista final dos requisitos pode ser visualizada na secção 4.3 abaixo.

4.2 Decisões técnicas face as estratégias de desenvolvimento

Analisadas as vantagens e desvantagens das diferentes estratégias de desenvolvimento e os conteúdos e serviços a disponibilizar nas aplicações pretendidas, secções 2.1 e 4.1 respetivamente, o passo seguinte no desenvolvimento deste projeto foi escolher as melhores estratégias de desenvolvimento para as aplicações pretendidas para este Projeto de Mestrado.

Relativamente ao protótipo do *site*, eram duas as estratégias de desenvolvimento possíveis, *Responsive* e *Adaptive Web Design*. Face às características dos serviços, e aos objetivos pretendidos, a escolha recaiu na estratégia *responsive*. O protótipo servirá apenas como prova de conceito para uma reestruturação *web* futura, ou seja, a construção de um novo *site* internet. O *responsive design* oferece mais vantagens relativamente à estratégia *adaptive*, porque apenas é desenvolvida uma solução que se adapta a todo o tipo de dispositivos, acabando por facilitar e minimizar os custos de manutenção. Por exemplo, qualquer alteração é apenas feita uma vez numa única solução, contrariamente a estratégia *adaptive*, onde é necessário alterar em todas as soluções que existem para cada tipo de dispositivo.

No caso da aplicação móvel a opção de desenvolvimento *web* foi descartada, pois o uso desta estratégia não permitia o uso da aplicação em modo *offline*, que era um dos requisitos. Assim, apenas as estratégias de desenvolvimento nativo e híbrido foram analisadas. Por um lado o desenvolvimento nativo obrigava a criação de um projeto por cada plataforma e a necessidade de conhecer diferentes linguagens de desenvolvimento (Objective C para iOS e Java para Android), e por outro o desenvolvimento híbrido que contrariamente ao nativo requer apenas conhecimentos em linguagens *web*, pois um único projeto servirá as duas plataformas. A escolha recaiu na estratégia híbrida, pois esta oferecia mais vantagens comparativamente a estratégia nativa, nomeadamente, menos tempo e custo no desenvolvimento e manutenção. Outro fator importante na decisão foi o fato da aplicação não necessitar usar grandes recursos nativos de cada dispositivo, evitando assim grandes perdas de desempenho, onde claramente a estratégia nativa é melhor.

4.3 Documentação produzida

De forma a garantir requisitos bem definidos e claros para todos os intervenientes no projeto, foi criada documentação durante a análise e especificação de requisitos onde foram descritos todos os dados e decisões relevantes durante a fase de análise. A realização de documentação nesta fase foi extremamente útil pois ajudou a clarificar todos os dados e decisões tomadas entre os diversos intervenientes do projeto. No final

da fase de análise foi criado um relatório de análise com base na documentação gerada, onde são detalhados os seguintes pontos:

- “Especificação do sistema” – Descreve de forma clara todos os componentes do sistema, explicando os diferentes atores e interações possíveis entre os mesmos;
- “Requisitos funcionais e de teste” – Descrevem as funcionalidades e serviços que o sistema deve oferecer;
- “Requisitos não funcionais” – Descrevem as restrições aos serviços, qualidades e características que o sistema deve oferecer;
- “Casos de uso” – Permitem descrever detalhadamente as interações dos utilizadores com todas as funcionalidades da aplicação;
- “Esboços da interface” – Oferecem uma visualização gráfica da interface, permitindo ao cliente entender como ficará a solução final.

De seguida são descritos todos os pontos presentes na versão final do documento e é explicado como todos os dados foram adquiridos. Este relatório pode ser visto no Anexo A.

4.3.1 Especificação do sistema

A especificação do sistema é uma parte importante na fase inicial dos projetos, pois proporciona um meio de comunicação com o cliente, simplificando assim a compreensão das diferentes componentes do sistema sem recurso a descrições mais técnicas. Os dados para a elaboração desta especificação foram adquiridos com base nas várias reuniões iniciais com o cliente (GAB), onde foi apresentado o propósito e objetivos do projeto. De seguida será descrita esta especificação.

O sistema será dividido em três componentes. Uma aplicação *web responsive*, designada por componente *web*, uma aplicação híbrida designada por componente *mobile*, e uma API *Web* que disponibiliza acesso aos serviços / conteúdos do Banco de Portugal e será utilizada tanto pelas componentes *web* como *mobile*.

Componente *web*

A componente *web* será apenas uma prova de conceito, ou seja, o objetivo é desenvolver um protótipo do *site* institucional focado apenas na página inicial, que mostre uma possível solução de como o novo *site* institucional deverá ser implementado. Esta componente não será colocada em produção. Apenas deve

demonstrar as principais funcionalidades do ponto de vista do utilizador. Deverá ser capaz de se adaptar automaticamente aos diferentes tamanhos de ecrã, proporcionando aos utilizadores uma ótima visualização e navegabilidade em qualquer dispositivo.

Uma vez que se trata de um protótipo, apenas terá um tipo de utilizador, que terá acesso total aos conteúdos.

Componente *mobile*

A componente *mobile* tem como objetivo a disponibilização de alguns dos principais serviços do Banco de Portugal num contexto móvel. Esta componente deverá ser colocada em produção aquando da sua conclusão. Os serviços a disponibilizar deverão ser identificados e analisados em cooperação com o GAB. A aplicação numa primeira fase deverá ser desenvolvida para as plataformas Android e iOS de forma otimizada para *smartphones* e *tablets*, no entanto deverá ser levado em conta a possibilidade de ser desenvolvida para Windows phone no futuro.

A aplicação apenas terá um tipo de utilizador que terá acesso total à aplicação, não sendo requerido nenhum *login*. A aplicação deverá ainda ser utilizável em modo *offline*.

API Web do Banco de Portugal

A API *Web* terá como objetivo a disponibilização dos serviços / conteúdos do Banco de Portugal. Será desenvolvida apenas para o uso nas aplicações acima descritas, no entanto poderá ser disponibilizada no futuro para programadores que queiram usar estes mesmos serviços. Deverá ser levado em conta a possibilidade de esta API crescer e disponibilizar mais serviços no futuro.

4.3.2 Requisitos funcionais e de teste

A especificação dos requisitos funcionais e de teste é uma das componentes mais importantes da fase de análise, na medida que uma má especificação de requisitos poderá ter um impacto negativo e oneroso no decorrer do projeto. Descrevem o que o produto de software deve fazer ou que ações processuais devem tomar, ou seja, as principais funcionalidades que a aplicação deverá ter.

Foram identificados seis requisitos para a componente *web*, sete para a componente *mobile* e oito para a API *Web* ou seja vinte e um no total. Para representar estes requisitos foi criada uma tabela por componente com as seguintes colunas:

identificador do requisito, componente a que pertence, nome do requisito, descrição, e observações. Na Tabela 4 são apresentados os requisitos funcionais e de teste da componente *web*, na Tabela 5 os requisitos da componente *mobile* e na Tabela 6 os requisitos da API *Web*.

Id	Componente	Nome	Descrição	Observações
RF01	<i>Web</i>	Visualizar as Taxas de Juro	Apresenta o valor das Taxas de Juro de referência na página inicial.	Os valores das taxas de juro devem ser os mais atuais. Não é necessário qualquer input por parte do utilizador.
RF02	<i>Web</i>	Visualizar a última publicação	Apresentar informações referentes ao último artigo publicado na página inicial.	Deverá ser apresentada a capa e o título da última publicação.
RF03	<i>Web</i>	Visualizar as Taxas de câmbio	Apresenta as cotações das principais moedas na página inicial.	Deverá apresentar as cotações das principais moedas do dia atual. As principais moedas são: Dólar dos Estados Unidos, Iene Japonês, Libra Esterlina, Franco Suíço, Real do Brasil.
RF04	<i>Web</i>	Conversor de moeda	Permite efetuar uma conversão com base nas taxas do dia requerido.	Deverá efetuar o cálculo de conversão entre qualquer uma das moedas listadas na data requerida. Todos os campos são de preenchimento obrigatório.
RF05	<i>Web</i>	Visualizar os comunicados mais recentes	Apresenta a lista dos comunicados mais recentes.	Deverá apresentar uma lista dos oito últimos comunicados ordenados por ordem cronológica. A data de publicação deverá ser exibida.
RF06	<i>Web</i>	Visualizar as intervenções públicas mais recentes	Apresenta a lista das últimas intervenções públicas	Deverá apresentar uma lista das oito últimas intervenções públicas ordenadas por ordem cronológica. A data de publicação deverá ser exibida.

Tabela 4 – Requisitos funcionais e de teste da componente *web*

Id	Componente	Nome	Descrição	Observações
RF07	<i>Mobile</i>	Simulador de Taxas de Juro	Permite efetuar simulações de Taxas de Juro	Deverá simular as Taxas de Juro, e apresentar os resultados do juro composto e do juro simples. Todos os campos são obrigatórios. Deverá ser possível fazer a simulação em modo offline.
RF08	<i>Mobile</i>	Conversor de moeda	Permite efetuar uma conversão com base nas taxas do dia requerido.	Deverá efetuar o cálculo de conversão entre qualquer uma das moedas listadas na data requerida. Todos os campos são de preenchimento obrigatório.
RF09	<i>Mobile</i>	Visualizar os comunicados mais recentes.	Apresenta a lista dos comunicados mais recentes.	Deverá apresentar uma lista dos 50 últimos comunicados ordenados por ordem cronológica.
RF10	<i>Mobile</i>	Visualizar as principais intervenções do governador.	Apresenta a lista das últimas intervenções públicas	Deverá apresentar uma lista das 50 últimas intervenções públicas ordenadas por ordem cronológica.
RF11	<i>Mobile</i>	Visualizar os destaques específicos para a aplicação <i>mobile</i> .	Apresentar uma lista de 3 destaques na vista inicial da aplicação.	Deverá ser possível fazer “swipe” ao ecrã para visualizar o destaque seguinte.
RF12	<i>Mobile</i>	Aceder ao BPstat <i>Mobile</i> na aplicação.	Incorpora o BPstat <i>Mobile</i> numa <i>web view</i> na aplicação.	O BPstat <i>Mobile</i> deverá manter a sua identidade gráfica.
RF13	<i>Mobile</i>	Aplicação bilingue	Permite escolher usar a aplicação em inglês ou português.	Todo o texto na aplicação deverá ser coerente com a linguagem selecionada.

Tabela 5 – Requisitos funcionais e de teste da componente *mobile*

Id	Componente	Nome	Descrição	Observações
RF14	API Web	Devolve os últimos comunicados ordenados por data	Retorna a lista dos últimos 50 comunicados.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF15	API Web	Devolver as últimas intervenções públicas ordenadas por data	Retorna a lista das últimas 50 intervenções na linguagem pretendida	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF16	API Web	Devolver o número especificado de intervenções públicas	Retorna a lista de intervenções, com o número de itens especificado como parâmetro.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF17	API Web	Devolver o número especificado de comunicados	Retorna a lista de comunicados, com o número de itens especificado como parâmetro.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF18	API Web	Devolver as taxas de juro	Devolve a lista das principais taxas de juro de referência.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF19	API Web	Devolver as taxas de câmbio do dia corrente	Devolve a lista das taxas de câmbio para a data introduzida.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF20	API Web	Devolver as últimas taxas de câmbio	Devolve a lista das últimas taxas de câmbio.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF21	API Web	Devolver as taxas de câmbio de uma moeda	Devolve a lista das taxas de câmbio para o código introduzido.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.

Tabela 6 – Requisitos funcionais e de teste da API Web

A elaboração destes requisitos foi feita com base na especificação de requisitos, e na análise dos serviços e conteúdos do *site* Institucional do Banco de Portugal. Todo o processo de levantamento de requisitos funcionais teve sempre o acompanhamento do cliente.

4.3.3 Requisitos não funcionais

Os requisitos não funcionais são aqueles que caracterizam o ambiente onde o sistema será implementado, e apesar de não fazerem diretamente parte do sistema,

devem ser tidos em conta, pois especificam as principais características que o mesmo terá. A identificação destes requisitos foi feita com base na especificação do sistema apresentada na secção 4.3.1

Os requisitos identificados são os seguintes:

- Usabilidade – As três componentes do sistema deverão ser intuitivas e fáceis de usar;
- Extensibilidade – A arquitetura do sistema deverá ser o mais flexível possível, de forma a permitir a adição de novos módulos e funcionalidades no futuro;
- Modificabilidade – Deverá ser possível alterar os diferentes módulos do sistema sem comprometê-lo num todo;
- Testabilidade – Deverá ser possível testar cada elemento em separado.
- Confidencialidade – Não serão necessários quaisquer dados do utilizador para a utilização de qualquer componente do sistema;
- Portabilidade – Quer a componente *web* quer a *mobile* deverão ser compatíveis com diversos dispositivos.

4.3.4 Casos de Uso

O diagrama de casos de uso contempla de uma forma simples e intuitiva as principais funcionalidades que as aplicações devem possuir, descrevendo a sequência de ações que o utilizador terá de fazer para concluir uma tarefa. Nesta secção foi definido um caso de uso por cada requisito da componente *web* e *mobile*.

Na Figura 12 estão ilustrados todos os casos de uso definidos no relatório de análise.

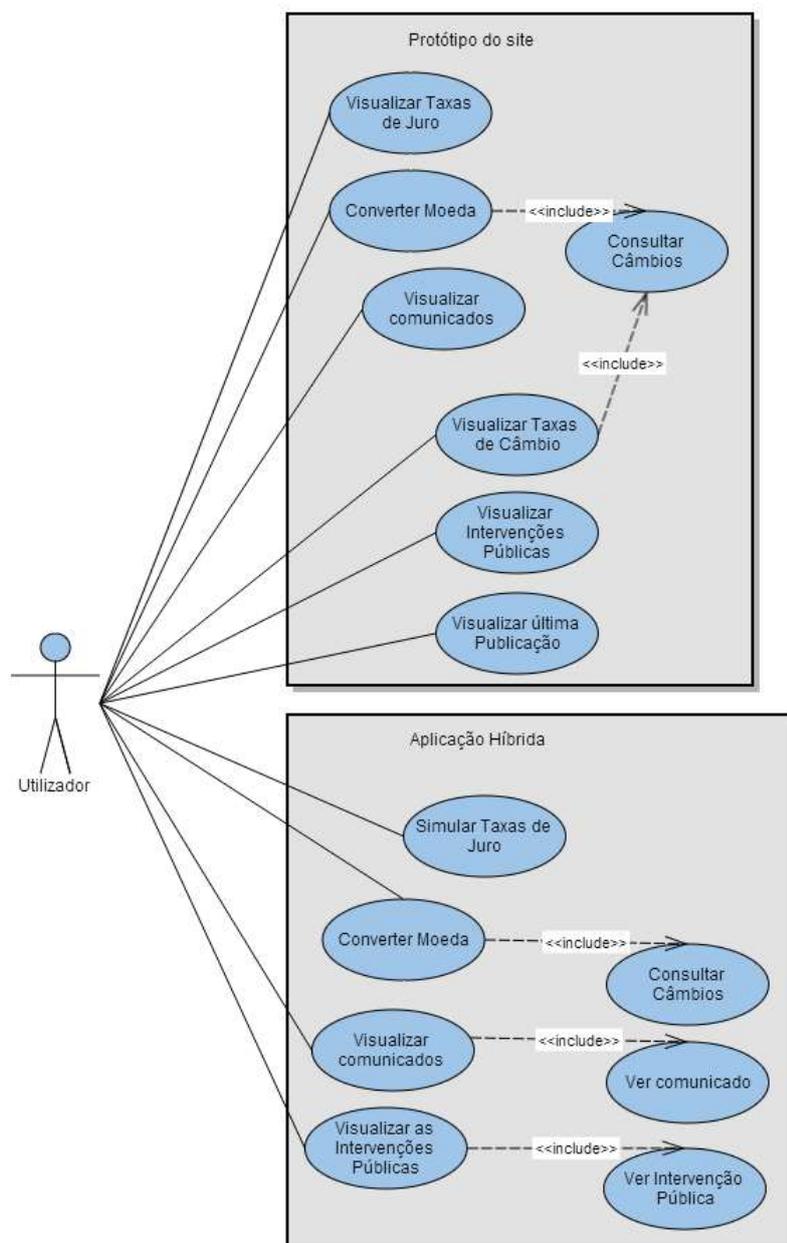


Figura 12 – Diagrama geral de casos de uso do protótipo do *site* e da aplicação móvel

Uma forma de melhorar a compreensão dos casos de uso é descrevê-los de forma detalhada. Como tal, cada um destes foi decomposto na seguinte estrutura: objetivo, componente, utilizador, fluxo de ações, resultado expectável, e estado de erro.

Abaixo estão ilustrados três exemplos, nos quais é fácil perceber a sequência de ações necessária para a realização de cada caso de uso.

Caso de uso – Simular taxas de juro

Componente – *Mobile*

Objetivo – Efetuar uma simulação de taxas de juro.

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder à aplicação
2. Abrir o menu
3. Selecionar o item “Simulador de taxas de juro”
4. Preencher todos os campos:
 - a. Capital inicial – Número decimal positivo;
 - b. Prazo – Número inteiro positivo;
 - c. Tipo de prazo – Escolher Anos, Meses ou Dias;
 - d. Taxa anual – Número decimal positivo;
 - e. Capitalizações – Número inteiro positivo
5. Clicar no botão calcular

Resultado Expectável – É apresentada uma nova vista onde são apresentados os resultados dos juros e capitais simples/compostos no final do prazo.

Estado de erro – É apresentado um alerta com informação dos *inputs* inválidos.

Caso de uso – Conversor de moeda

Componente – *Mobile*

Objetivo – Fazer a conversão de 5 Euros para o Dólar Australiano no dia 12-12-2013.

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder à aplicação;
2. Abrir o menu;
3. Selecionar o item “Conversor de moeda”;
4. Preencher todos os campos;
 - a. Valor – Número decimal;
 - b. Data – O formato é apresentado;
 - c. De – Escolher na lista apresentada;
 - d. Para – Escolher na lista apresentada;
5. Clicar no botão calcular.

Resultado Expectável – É apresentada uma nova vista onde são apresentados os resultados da conversão, com os dados de *input*, e o resultado da conversão.

Estado de erro – É apresentado um alerta com informação dos *inputs* inválidos.

Caso de uso – Ver os comunicados em Inglês

Componente – *Mobile*

Objetivo – Mudar a linguagem da aplicação para Inglês e ver os comunicados mais recentes.

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder à aplicação;
2. Abrir o menu;
3. Selecionar o item “Opções”;
4. Mudar a linguagem para Inglês;
5. Voltar a abrir o menu;
6. Selecionar o item “Comunicados”;

Resultado Expectável – É apresentada uma nova vista com a listagem dos últimos 50 comunicados. Cada comunicado tem associado a data de publicação, um título, a informação a indicar se é um PDF e a referência para o conteúdo desse mesmo item.

Estado de erro – Sem ligação a internet.

Os outros casos de uso podem ser vistos no Anexo A referente ao relatório de análise.

4.3.5 Esboços

A obtenção de toda a informação na fase de análise permite por fim criar os esboços da interface das aplicações.

Os esboços permitem uma visualização gráfica da interface das aplicações e ajudam a entender o comportamento nos diferentes ambientes e estados, ajudando a correção de detalhes gráficos de acordo com o gosto do cliente.

Foram desenhados esboços para todas as vistas da aplicação móvel, e esboços para as diferentes versões da aplicação *web*, nomeadamente para a versão *desktop*, *tablet* e *smartphone*. Os mesmos foram apresentados e discutidos com o cliente com o objetivo de projetar uma interface de acordo com as ideias do mesmo. Durante este processo foram solicitadas alterações ao menu, nomeadamente a ordenação dos diversos itens, e aos conteúdos exibidos na página inicial da aplicação móvel, as quais foram prontamente alterados e posteriormente aprovados.

Nas Figuras 13 e 14 estão ilustrados dois exemplos dos esboços de baixo nível da aplicação móvel apresentados ao cliente, outros esboços podem ser vistos no Anexo A referente ao relatório de análise.



Figura 13 - Esboço da vista de comunicados num *smartphone* da aplicação móvel



Figura 14 - Esboço da vista de comunicados num *tablet* da aplicação móvel

Na Figura 13 está ilustrado o esboço da vista dos Comunicados na versão *smartphone*, enquanto na Figura 14 é apresentado o esboço na versão *tablet*. Em qualquer uma das figuras o conteúdo exibido será igual, no entanto a aplicação será apresentada de forma otimizada consoante o dispositivo. No caso dos *tablets*, o menu estará sempre visível no lado esquerdo contrariamente aos *smartphones* onde o menu está escondido.

Relativamente ao protótipo do *site* foram desenhados em primeiro lugar esboços de baixa fidelidade (desenhados a lápis) com as ideias base para à página inicial em *desktops*, *tablets* e *smartphones* e depois os *mockups* de média-fidelidade que foram apresentados e aprovados pelo cliente. Abaixo nas Figuras 15, 16 e 17 estão ilustrados os esboços de baixa fidelidade criados para a página inicial do *site* institucional.



Figura 15 – Protótipo de baixa fidelidade da *homepage* num desktop

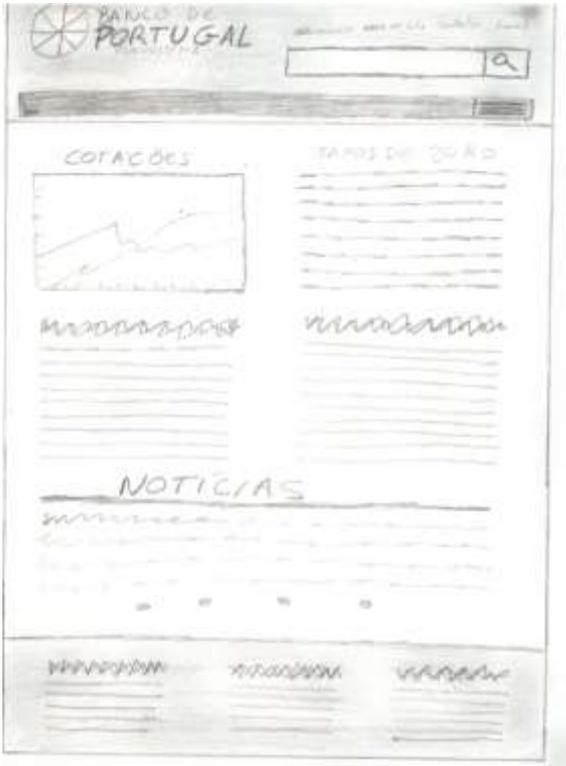


Figura 16 – Protótipo de baixa fidelidade da *homepage* num tablet



Figura 17 – Protótipo de baixa fidelidade da *homepage* num *smartphone*

Esboços de alta-fidelidade

Os esboços da interface acima descritos foram igualmente importantes para o desenho gráfico da aplicação, que foi desenvolvido por uma empresa externa. Foram facultados todos os esboços de baixa e média-fidelidade à WIZ Interactive, para que, com base nos mesmos fosse criada uma identidade gráfica para a aplicação.

A WIZ é uma empresa de *design* gráfico com alguma experiência em projetos no Banco de Portugal, pelo que já são conhecedores da identidade gráfica que o Banco pretende para todos os seus portais e aplicações. Assim, o trabalho desenvolvido por estes passou por desenhar os *mockups* de alta-fidelidade, os guias de cores, as fontes, os tamanhos e os espaçamentos a utilizar nas aplicações. Os *mockups* e guias fornecidos pela WIZ Interactive podem ser vistos no Anexo F.

Capítulo 5

Desenho da solução

A fase de desenho tem como objetivo modelar o sistema, atendendo aos requisitos levantados na fase de análise, e prepará-los para a implementação. Nesta fase, a criação de documentação adequada facilita a comunicação entre os *stakeholders*, permite melhor compreensão dos processos de codificação e contribui para que o código produzido tenha maior qualidade.

Neste capítulo será apresentada a documentação gerada, detalhando a arquitetura do sistema e os diagramas criados.

5.1 Arquitetura do sistema

Em primeiro lugar é importante descrever a arquitetura do sistema, ou seja, todos os componentes que serão desenvolvidos e como estes comunicam entre si. Como descrito acima no Capítulo 4 secção 4.3.1 o sistema será constituído por 3 componentes, uma aplicação *web*, uma aplicação *mobile* e uma API RESTful. A arquitetura do sistema está ilustrada na Figura 18.

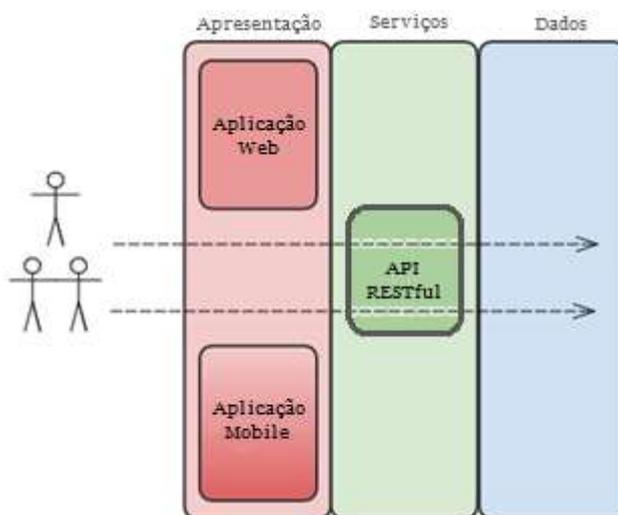


Figura 18 – Diferentes componentes que compõem o sistema

A arquitetura pode ser descrita como uma arquitetura típica em camadas (N-Tier), onde cada camada é autocontida (o suficiente) com responsabilidades específicas, de forma a ser independente de todas as outras. Cada camada interage apenas com a camada adjacente, ou seja, a alteração de uma camada não afeta as outras desde que os contratos sejam cumpridos.

O sistema está dividido em 3 camadas:

- **Camada de dados** – Responsável pela persistência dos dados. Única camada que acede as bases de dados;
- **Camada de serviços** – Camada prestadora de serviços, fornece acesso aos dados como um serviço da *web*, ou seja, atua como canal de comunicação entre clientes e a camada de dados. Os clientes interagem com a camada de dados através da camada de serviços;
- **Camadas de apresentação** – As aplicações cliente são os consumidores dos serviços. Oferecem uma interface para os utilizadores interagirem com o sistema. Alguns clientes podem também implementar lógica de negócio.

5.1.1 Racional de decisões chave

A camada de dados estava totalmente implementada, no entanto, existem algumas particularidades importantes a apresentar nesta fase.

No início do projeto foi explicado que o acesso às bases de dados não seria permitido por razões de segurança, e que os dados apenas deveriam ser consumidos pelos serviços que já existiam na plataforma SharePoint. Foi também transmitido que os serviços do SharePoint apenas podem ser acedidos por aplicações que corram nos servidores internos do Banco de Portugal, pois o acesso do exterior está totalmente bloqueado por *firewall*. Estas particularidades foram determinantes na escolha da arquitetura do sistema, justificando assim a necessidade da camada de serviços.

A camada de serviços é a única que acede aos dados e está disponível na internet, razão que obrigará a cuidados redobrados de segurança aquando da sua implementação. Será extremamente importante fazer o tratamento de todo o tipo de exceções e validar todo o tipo de *inputs*, de forma a não expor os serviços da camada de dados e evitar usos mal-intencionados por terceiros.

5.2 Desenho e Modelação do sistema

Apresentada a arquitetura, a tarefa seguinte passa por modelar todo o sistema de forma a simplificar o trabalho na fase de implementação, e garantir que os diversos módulos se interligam de forma correta. Nesta fase é necessário rever todos os dados recolhidos durante a análise e especificação de requisitos e adaptá-los de acordo com os objetivos e características do sistema proposto.

Foi criado o documento “Desenho e modelação do sistema” onde são descritos todos os módulos do sistema e explicado como os mesmos devem ser implementados. Nesta fase o uso de diagramas UML (*Unified Modeling Language*) foi essencial para ilustrar o desenho do sistema e como os diferentes componentes se interligam. Este documento é composto pelos diagramas de módulos, de classes e de sequência apresentados nas secções que se seguem.

Esta fase tem o objetivo de auxiliar à implementação, ou seja criar as bases para uma implementação mais simples e intuitiva. Outros detalhes mais específicos serão apresentados no Capítulo 6 referente à implementação.

5.2.1 Diagrama de módulos

Uma das melhores formas de descrever um sistema é através de um diagrama de módulos ou pacotes. Estes diagramas têm como objetivo descrever os módulos ou pedaços do sistema divididos em agrupamentos lógicos mostrando todas as suas dependências. O uso deste tipo de diagramas ajuda a entender a comunicação entre módulos e as classes que os constituem.

Na Figura 19 está ilustrado o diagrama de módulos do sistema. O diagrama está organizado por camadas de acordo com a arquitetura descrita na Secção 5.1

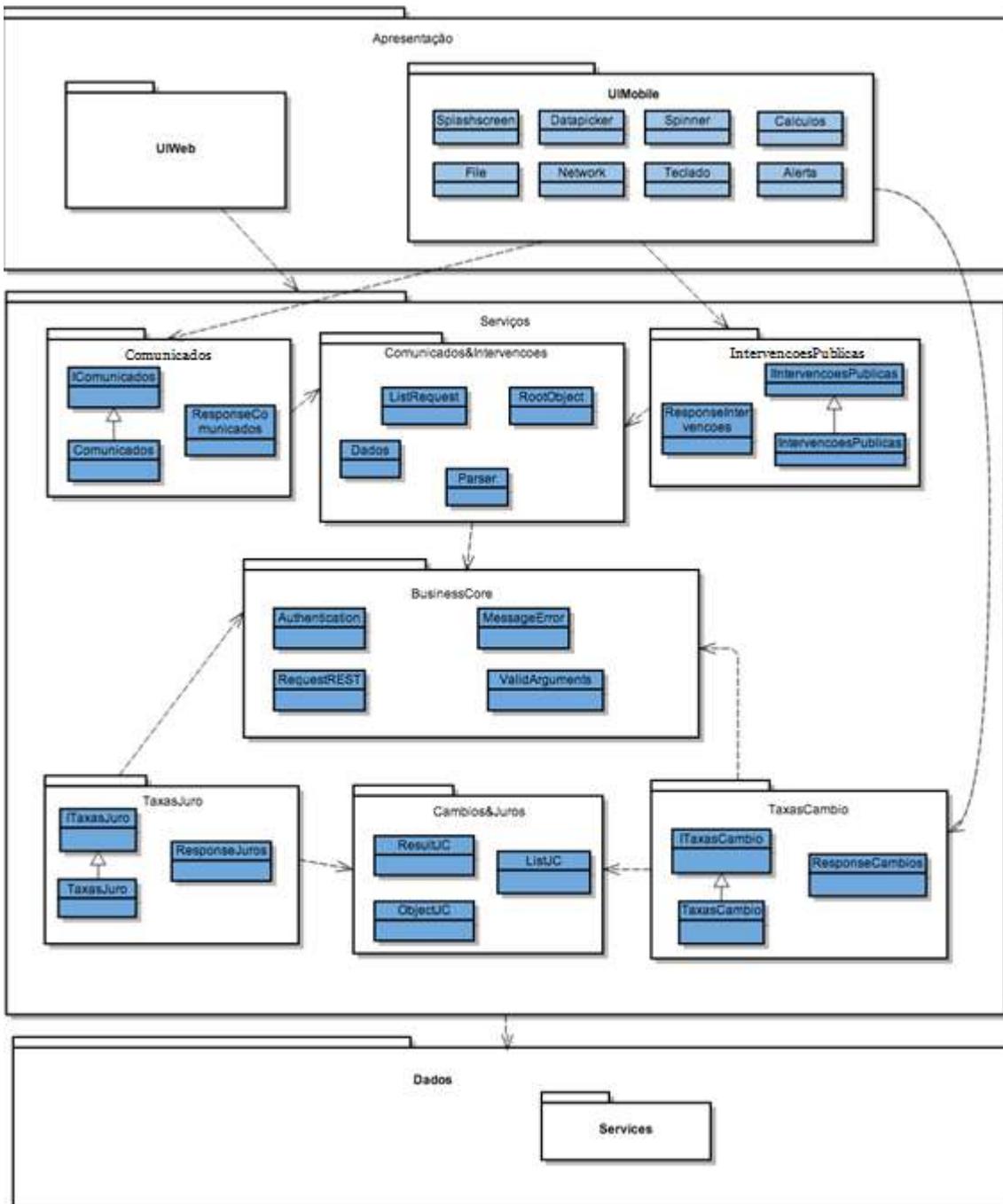


Figura 19 – Diagrama de módulos do sistema

Descrição de Módulos

UIWeb – Módulo da apresentação que está encarregue da comunicação entre o utilizador e o sistema. É responsável por apresentar os conteúdos ao utilizador, de forma otimizada, seja qual for o dispositivo de acesso. Este módulo representa a aplicação *web*.

UIMobile – Módulo que representa a aplicação *mobile*. Módulo da apresentação encarregue por disponibilizar uma interface *mobile* ao utilizador. É composto por oito *plugins* nativos em Java e Objective C, que permitem o uso de componentes nativos das plataformas Android e iOS.

IntervencoesPublicas – Módulo responsável pela disponibilização da lista de intervenções públicas.

Comunicados – Módulo responsável pela disponibilização da lista de comunicados.

Comunicados&Intervencoes – Módulo com as classes que auxiliam os pedidos aos serviços da camada de dados, e com as estruturas de dados das respostas.

TaxasJuro – Módulo responsável pela disponibilização da informação referente as taxas de juro.

TaxasCambio – Módulo responsável pela disponibilização da informação referente as taxas de câmbio.

Cambios&Juros – Módulo com as estruturas de dados que armazenam as respostas à invocação dos serviços da camada de dados.

BusinessCore – Módulo responsável pela autenticação dos serviços com a camada de dados e todas as classes que auxiliam os serviços.

Services – Módulo da camada de dados que permite obter os conteúdos. Será usado pela camada de serviços para obter toda a informação necessária as aplicações clientes.

5.2.2 Diagrama de classes

Uma forma mais detalhada de representar as entidades que compõem o sistema e todas as associações é através do modelo de classes que oferece uma visão mais profunda em relação ao diagrama de módulos.

Nesta fase foi criado um diagrama de classes da componente de serviços com todas as associações entre os diversos objetos e uma breve descrição de todas as classes. Este diagrama é ilustrado na Figura 20, e as descrições podem ser encontradas no Anexo B.

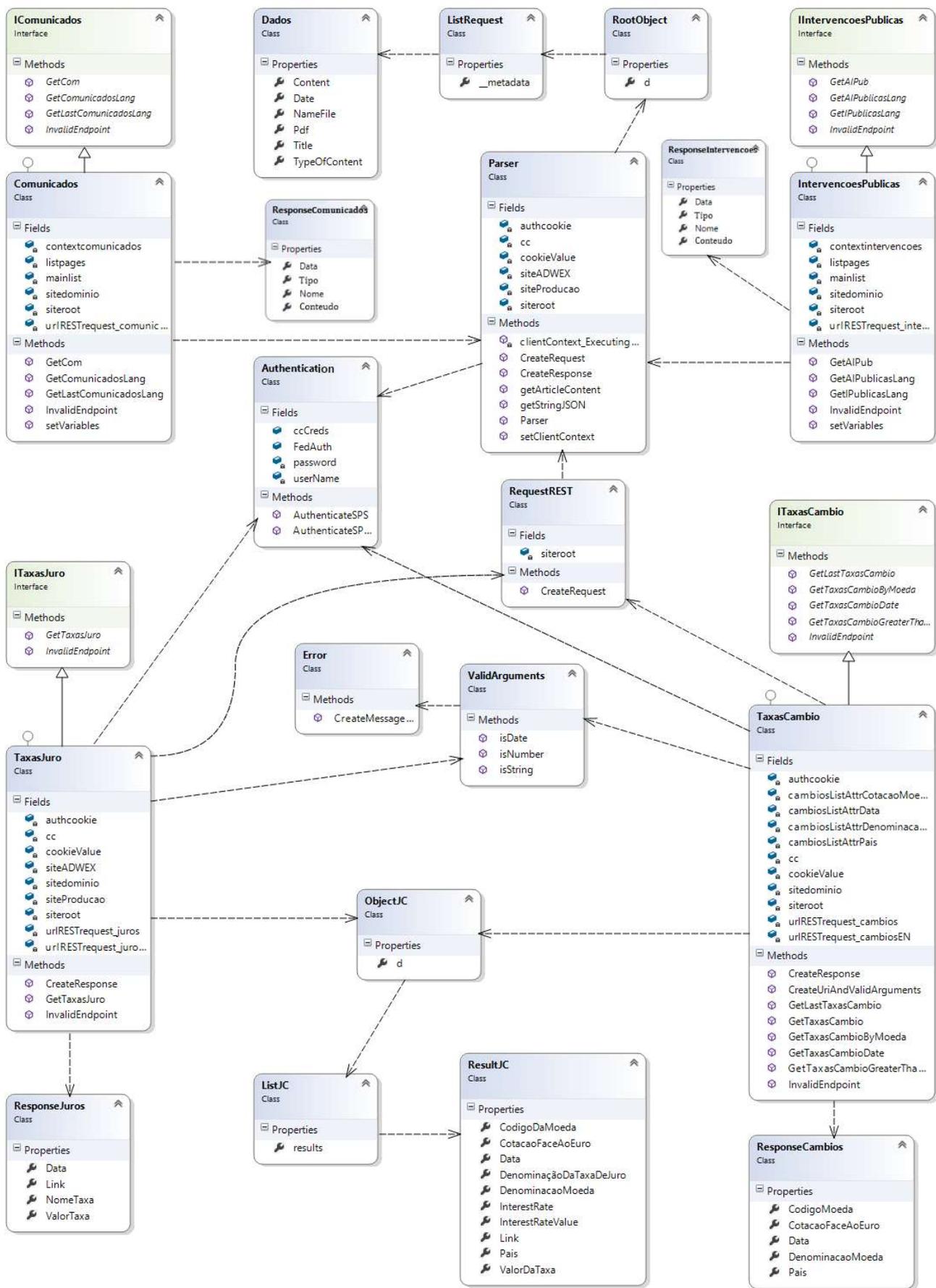


Figura 20 – Diagrama de classes da camada de serviços

5.2.3 Diagramas de sequência

Os diagramas de sequência visam mostrar a sequência de informação trocada entre os diversos objetos do sistema e descrevem como um grupo de objetos colabora na realização duma determinada ação (caso de uso) ao longo do tempo. São construídos a partir da descrição dos casos de uso, e dão ênfase à ordenação temporal em que as mensagens são trocadas entre os objetos do sistema.

Foi construído um diagrama de sequência por cada caso de uso, com o objetivo de clarificar e simplificar a implementação das aplicações cliente. De seguida nas Figuras 21, 22 e 23 são apresentados três exemplos destes diagramas, sendo que os restantes se encontram no Anexo B.

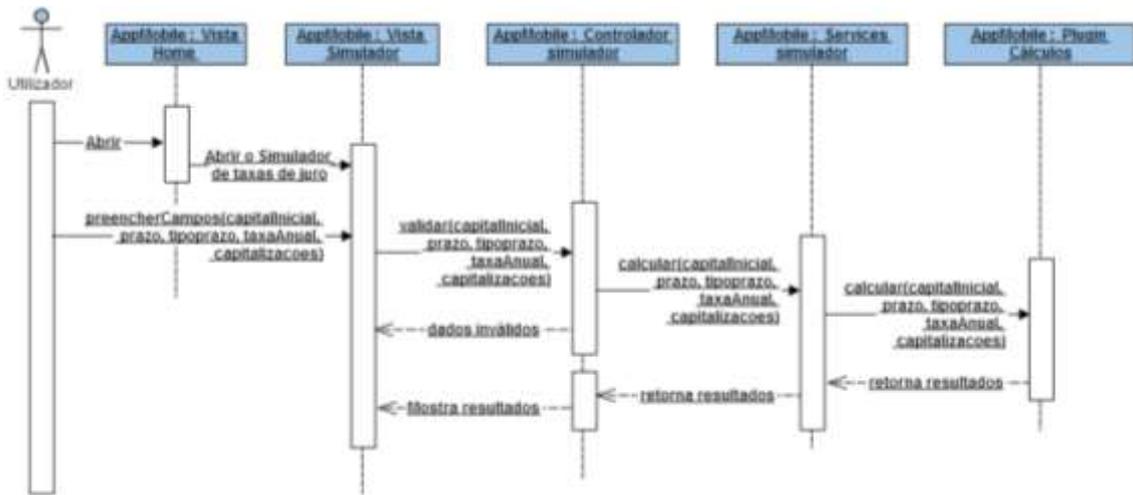


Figura 21 – Diagrama de sequência "Fazer simulação de taxas de juro"

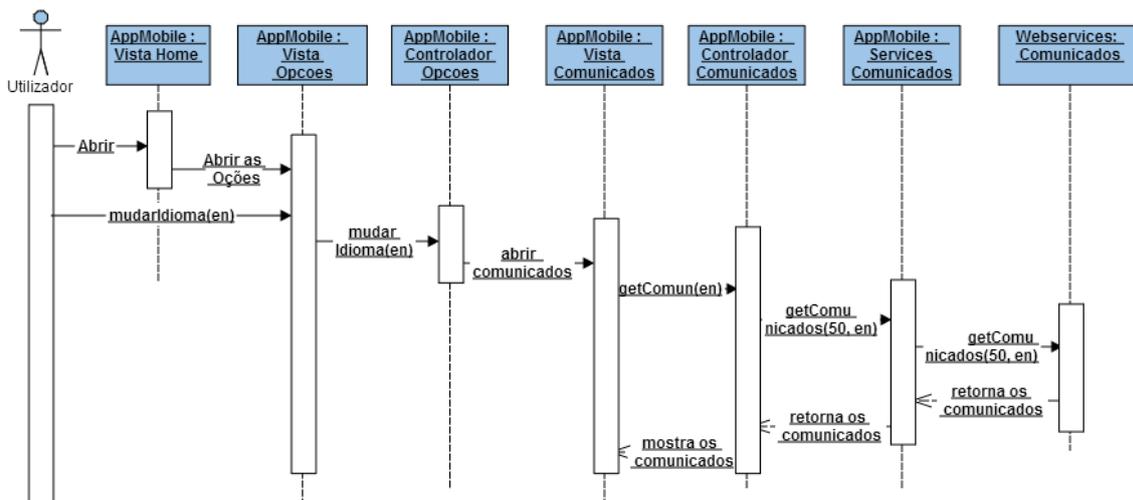


Figura 22 – Diagrama de sequência "Visualizar comunicados no idioma inglês"

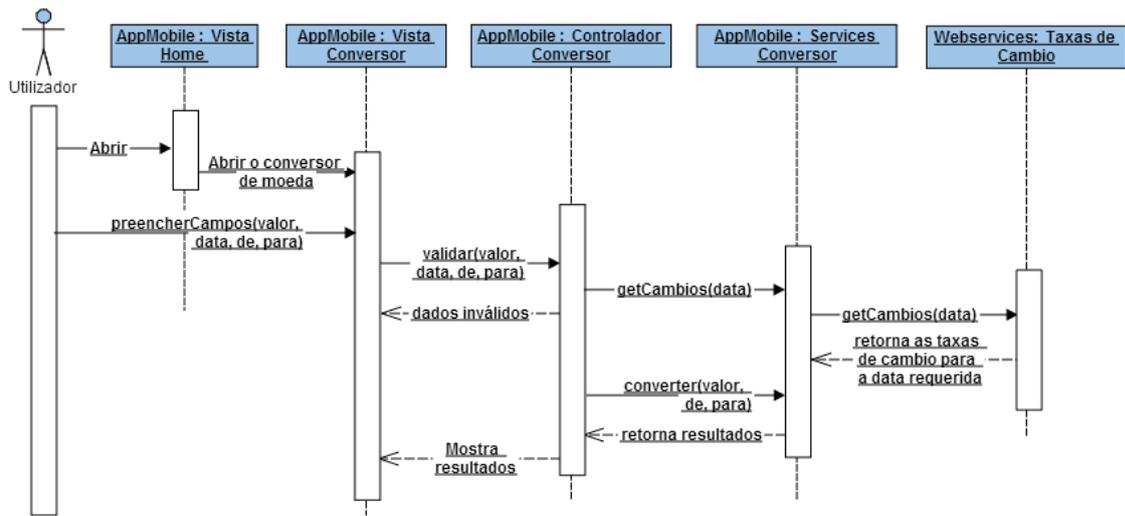


Figura 23 – Diagrama de seqüência "Fazer conversão de moeda"

Capítulo 6

Implementação

O sucesso de um projeto de software, requer uma implementação cuidada que cumpra com os requisitos definidos de acordo com a documentação gerada *a priori*.

Nesta fase é descrito todo o processo de desenvolvimento, as *frameworks* e metodologias usadas e como se procedeu o desenvolvimento das diferentes componentes que compõem o sistema.

6.1 *Web Services*

Os *web services* são componentes que possibilitam às aplicações comunicarem utilizando protocolos padrão da internet (XML ou JSON, por exemplo). São uma solução que procura resolver o problema de comunicação entre aplicações que empregam diferentes tecnologias.

Este projeto segue exatamente a mesma lógica, por um lado temos duas aplicações cliente que disponibilizam informação para os utilizadores e por outro o servidor no qual os dados estão armazenados. Para que a informação circule entre as aplicações cliente (camada de apresentação) e o servidor (camada de dados) foi necessário criar *web services*.

Foi criada uma API, que segue a arquitetura RESTful, onde cada URL define o serviço a utilizar e quais os parâmetros de entrada do serviço. Cada serviço é caracterizado por transações sem estado, onde todos os pedidos são geridos de forma independente. Para a criação desta API foi criado um projeto WCF *service application* 4.0 no Visual Studio 2010. WCF (*Windows Communication Foundation*) é um modelo de programação da Microsoft usado para a construção de aplicações orientadas a serviços. Este modelo permite aos programadores criar soluções seguras, confiáveis e de boa performance que se interligam com todas as plataformas de forma simplificada.

6.1.1 API RESTful

A API RESTful é composta por sete módulos e vinte e uma classes, no entanto, explicar todos os detalhes da implementação tornaria esta secção extensa e repetitiva, pois muitos dos processos da implementação são iguais entre os diferentes serviços. Embora cada serviço tenha as suas características, a lógica de implementação é igual, pelo que nesta secção apenas são descritos os processos considerados mais importantes: o processo de autenticação com a camada de dados, a criação dos pedidos e o *parsing* dos dados e resposta.

Foram implementados os serviços dos comunicados, intervenções públicas, taxas de câmbio e taxas de juro. Estes quatro serviços precisam ser parametrizáveis e dinâmicos de forma a dar resposta às diferentes necessidades das aplicações cliente. Assim, todos os métodos destes serviços permitem escolher o formato e a linguagem no qual devem ser trocadas as mensagens, ou seja, podem ser no formato (XML ou JSON) e na linguagem (Português ou Inglês). Abaixo na Figura 24 estão ilustrados os detalhes da implementação dos serviços acima referidos.

URL	Método HTTP	Classe	Parâmetros	Operação	Observações	Código de sucesso	Código de erro
taxasCambio/date	GET	TaxasCambio	date, format, language	Obter as taxas de câmbio para a data introduzida como parâmetro	O parâmetro date é de preenchimento obrigatório, no entanto os parâmetros format e date são opcionais.	101	102
taxasCambio/codigomoeda	GET	TaxasCambio	codigo, format	Obter todas as taxas de câmbio do código da moeda introduzido como parâmetro	O parâmetro codigo é de preenchimento obrigatório, no entanto o format é opcional.	111	112
taxasCambio/last	GET	TaxasCambio	format	Obter as taxas de câmbio válidas do dia mais recente	O url pode ser invocado sem parâmetros.	121	122
taxasCambio/after	GET	TaxasCambio	date, format	Obter as taxas de câmbio posteriores a data introduzida como parâmetro	O parâmetro date é de preenchimento obrigatório, no entanto o format é opcional.	131	132
comunicados/	GET	Comunicados	number, language, format	Obter a lista de comunicados	O url pode ser invocado sem parâmetros. No caso de ser invocado sem parâmetros será retornado apenas os últimos 30 comunicados na linguagem portuguesa.	201	201
taxasJuro/	GET	TaxasJuro	format	Obter as principais taxas de juro mais recentes	O url pode ser invocado sem parâmetros.	301	302
intervencoespublicas/	GET	Intervencoes Publicas	number, language, format	Obter a lista de intervenções Públicas	O url pode ser invocado sem parâmetros. No caso de ser invocado sem parâmetros será retornado apenas as últimas 30 intervenções na linguagem portuguesa.	401	402

Figura 24 – Tabela com os detalhes de cada operação dos serviços RESTful desenvolvidos

Autenticação

A camada de dados apenas pode ser acessada pelos *web services* da plataforma gestora de conteúdos (SharePoint), uma vez que não é permitido aceder e manipular diretamente os dados no lado do servidor, como já explicado na Secção 5.1.1 acima. No entanto, o uso dos *web services* da camada de dados só é possível com autenticação.

Foi criada a classe Authentication, que é responsável pela autenticação dos serviços com a camada de dados. Esta classe implementa um método com o modelo de autenticação FBA (*Forms Based Authentication*), pois a API RESTful é executada fora do contexto do SharePoint e este modelo de autenticação é o único que funciona na atual arquitetura do Banco de Portugal. O método encontra-se no Anexo G.1.

Primeiro é solicitado o serviço *web* Authentication.asmx que permite iniciar sessão no SharePoint, são passadas as credencias via POST e é retornado o *cookie* de sessão, que depois é usado em cada pedido à camada de dados. Quando um serviço efetua um pedido à camada de dados verifica se o *cookie* ainda não expirou, caso já tenha expirado é novamente invocado o método e é adicionado o *cookie* ao pedido. O cookie tem sempre a duração de uma hora, pelo que passado esse tempo expira sendo necessário criar um novo. No Anexo G.2 está ilustrado o excerto de código usado na criação de um pedido.

Criação de pedidos

Para a realização dos pedidos foi implementado um objeto para a criação de transações, de forma a uniformizar o processo entre todos os serviços e permitir a reutilização independentemente do objeto. Assim, foi implementado o objeto CreateRequest, que é responsável pela criação de todos os pedidos entre a camada de serviços e a camada de dados. O código referente a esta classe é ilustrado no Anexo G.3. Na criação de um pedido são passados todos os atributos necessários, nomeadamente, o URL, e os parâmetros que permitem criar o *cookie* de autenticação. Após a criação do pedido, a classe que implementa o serviço inicia o mesmo, e guarda a resposta numa *string*, que será desserializada para uma estrutura de dados (modelo com as classes do contrato de dados). Um exemplo desses contratos é o modelo de classes que permite guardar a resposta da camada de dados a um pedido do serviço TaxasJuro, ver Anexo G.4. Os dados são guardados nestas estruturas, e depois analisados e retornados ao utilizador.

Parsing dos dados e resposta

Um dos problemas inerentes de aceder indiretamente aos dados através do uso de *web services*, é o fato de não se ter controlo sobre a formatação e estruturação dos dados recebidos. Os dados são recebidos no formato JSON, que sendo um formato standard e facilmente interpretado pelo C# simplifica o processo, no entanto, as respostas trazem muita informação não relevante, pelo que é necessário extrair apenas o conteúdo útil para as aplicações cliente. Foi necessário criar mecanismos para extração da informação relevante, deste modo, todas as classes que implementam os serviços, nomeadamente as classes *TaxasJuro*, *TaxasCambio*, *Comunicados* e *IntervencoesPublicas*, após obterem a resposta da camada de dados, invocam um método próprio de cada serviço para esta extração. Um destes exemplos é o método *CreateResponse* da classe *TaxasJuro* apresentado o Anexo G.5. Deste modo, a estrutura de dados com a resposta é analisada e apenas o conteúdo relevante para o serviço é copiado para a nova estrutura de dados que será retornada por este.

6.2 Aplicação *mobile*

Para o desenvolvimento desta componente foi escolhida a *framework* Apache Cordova responsável pelo empacotamento da aplicação *web* como nativa, e a *framework* AngularJS que permite desenvolver a componente *web* da aplicação. A escolha do Cordova justifica-se com a necessidade de aceder a diversos recursos próprios de cada plataforma (*datapicker*, teclado, *network*, ...), pois além de ser *Open Source*, oferece mecanismos para fácil implementação dos mesmos. Por outro lado a escolha do AngularJs justifica-se por ser uma *framework* MVC muito rica na construção de aplicações HTML5, e por garantir aplicações com alto nível de desempenho.

No desenvolvimento de uma aplicação é muito importante que exista uma separação clara de responsabilidades entre as camadas da aplicação, permitindo que o projeto seja modular, flexível, testável e acima de tudo de fácil manutenção. Deste modo a aplicação foi desenvolvida segundo a estrutura ilustrada abaixo na Figura 25, onde podemos ver como cada componente se integra com os restantes.

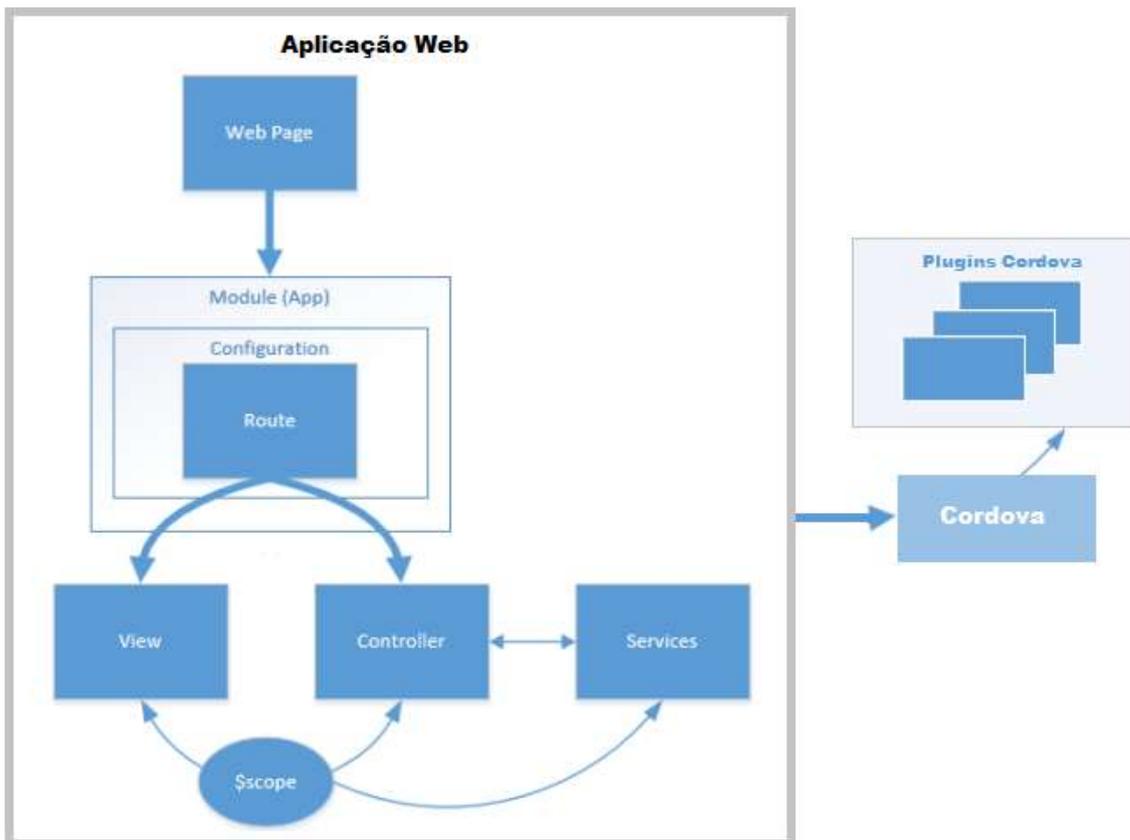


Figura 25 – Estrutura dos diferentes componentes da aplicação mobile

Nas secções seguintes são explicados os diferentes componentes da aplicação, como estas se relacionam e algumas decisões importantes tomadas durante o desenvolvimento.

6.2.1 Cordova

Para a criação do projeto foi usada a versão 3.4.0 da *framework* Apache Cordova. Foi implementada uma etapa preliminar do projeto, focada apenas na configuração dos ambientes de desenvolvimento e no entendimento das configurações da *framework*. Uma vez que a aplicação foi desenvolvida para as plataformas iOS e Android, foi necessário configurar dois ambientes de desenvolvimento diferentes.

Em primeiro lugar foi configurado um projeto Apache Cordova no IDE de desenvolvimento Eclipse, para a versão Android, e em seguida repetiu-se o processo no ambiente de desenvolvimento Xcode para a plataforma iOS. Abaixo nas Figuras 26 e 27 apresenta-se a estrutura do projeto nos dois ambientes de desenvolvimento referidos.



Figura 26 – Estrutura do projeto da aplicação *mobile* no IDE de desenvolvimento Eclipse

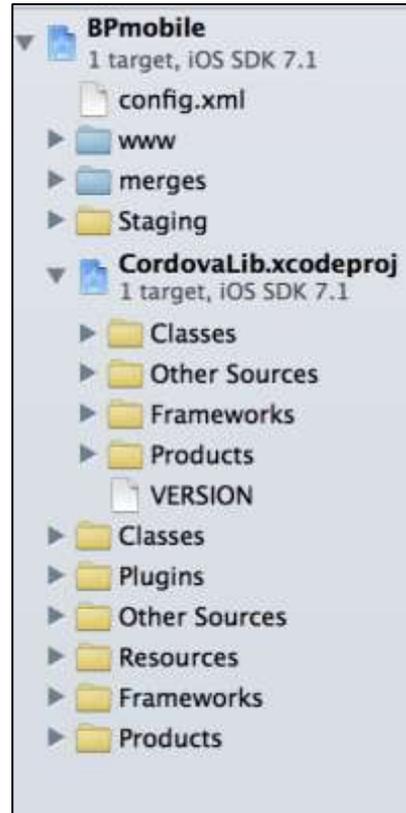


Figura 27 – Estrutura do projeto da aplicação *mobile* no IDE de desenvolvimento Xcode

Com a criação das estruturas de projeto ficaram criadas as condições para iniciar o desenvolvimento, da componente *web* da aplicação, que depois será empacotada pelo *framework* Apache Cordova para correr como aplicação nativa. Esta componente é exatamente a mesma para as duas plataformas (o código é igual), pelo que o desenvolvimento começou por ser feito em iOS no Xcode, e depois copiado para o eclipse.

6.2.1 Web Page

A aplicação é composta por uma única página HTML (*index.html*), que contém a chamada a todos os ficheiros JavaScript e CSS usados, e inicializa a aplicação através do uso da propriedade **ng-app**.

O atributo **ng-app** na tag `<html>` informa que além de HTML, o documento (DOM) é também um documento AngularJS. Esta propriedade instancia o módulo principal da aplicação onde são feitas as configurações iniciais. O módulo foi iniciado com o nome **BancoPortugal**.

Dentro da tag <body> foi adicionado a tag <ion-nav-view> que permite que toda a geração de código seja renderizada dentro desta.

6.2.2 Módulo de configuração da aplicação

Na *Web Page* foi inicializado o módulo principal da aplicação que permite fazer as configurações globais. Para iniciar estas configurações foi necessário criar um ficheiro JavaScript com o nome “app.js”, onde foi escrito o código com todas as configurações importantes. Este módulo é composto por um conjunto de blocos de configuração e execução que são aplicados à aplicação durante o processo de inicialização.

Foi criada uma função de execução de forma a configurar as variáveis de contexto global da aplicação, nomeadamente a definição da largura da janela, a desativação do menu do lado direito, e a definição de *tablet*, ou seja, a largura a partir do qual um dispositivo é considerado *tablet*.

Foi também criada uma função de configuração com as configurações do *provider* referente aos idiomas suportadas na aplicação, sendo estes, português e inglês. Deste modo todos os textos estáticos da aplicação estão em dois ficheiros JavaScript diferentes, nomeadamente “en.js” e “pt.js” para os textos em Inglês e Português respetivamente. Consoante a escolha do idioma por parte do utilizador, é carregado o valor das variáveis do ficheiro inglês ou português para as respetivas *views*, já que todas as variáveis têm exatamente o mesmo nome, mas com o conteúdo traduzido.

Por último, foi criada outra função de configuração, que permite configurar os diferentes estados da aplicação. Esta função carrega as *views (templates)* e *controllers* de acordo com o URI, ou seja, observa qual o url (`$stateProvider.url()`) e tenta mapear o caminho de acordo com uma rota pré-definida, assim consegue executar o *controller* e a *view* correspondente para cada url. A *view* correspondente a um determinado *controller* é sempre carregada dentro da tag <ion-nav-view> presente no ficheiro index.html. No Anexo G.6 está ilustrado um excerto do código do módulo principal da aplicação descrito acima.

6.2.3 Model View Controller (MVC)

O AngularJS incorpora os princípios básicos do padrão MVC [22] de como implementar o *client-side* de aplicações *web*. O *model* é um objeto “\$scope” que contém os dados que circulam entre os *controllers* e as *views*, ou seja as propriedades e comportamentos desse objeto representam o que pode ser acedido pela respetiva *view*.

Os *controllers* são responsáveis por determinar o estado inicial do *model* com os métodos e dados utilizados pela *view*. No entanto, o *controller* não tem responsabilidade de manter o estado da aplicação, nem interagir (controlar os pedidos/respostas) com os serviços remotos. Por último, as *views* correspondem ao código HTML resultante do pós-processamento do AngularJS que inclui as extensões de interação com o *model*.

Esta divisão cria uma fundação sólida para a arquitetura da aplicação, onde o *model* possui as referências dos dados, o *controller* define o comportamento, e a *view* lida com o *layout* e as interações com o *controller*, que responde de acordo com a interação. Na Figura 28 apresenta-se o diagrama de todas as *views*, *controllers* e *services* criados no desenvolvimento da aplicação, e em seguida uma descrição destes componentes.

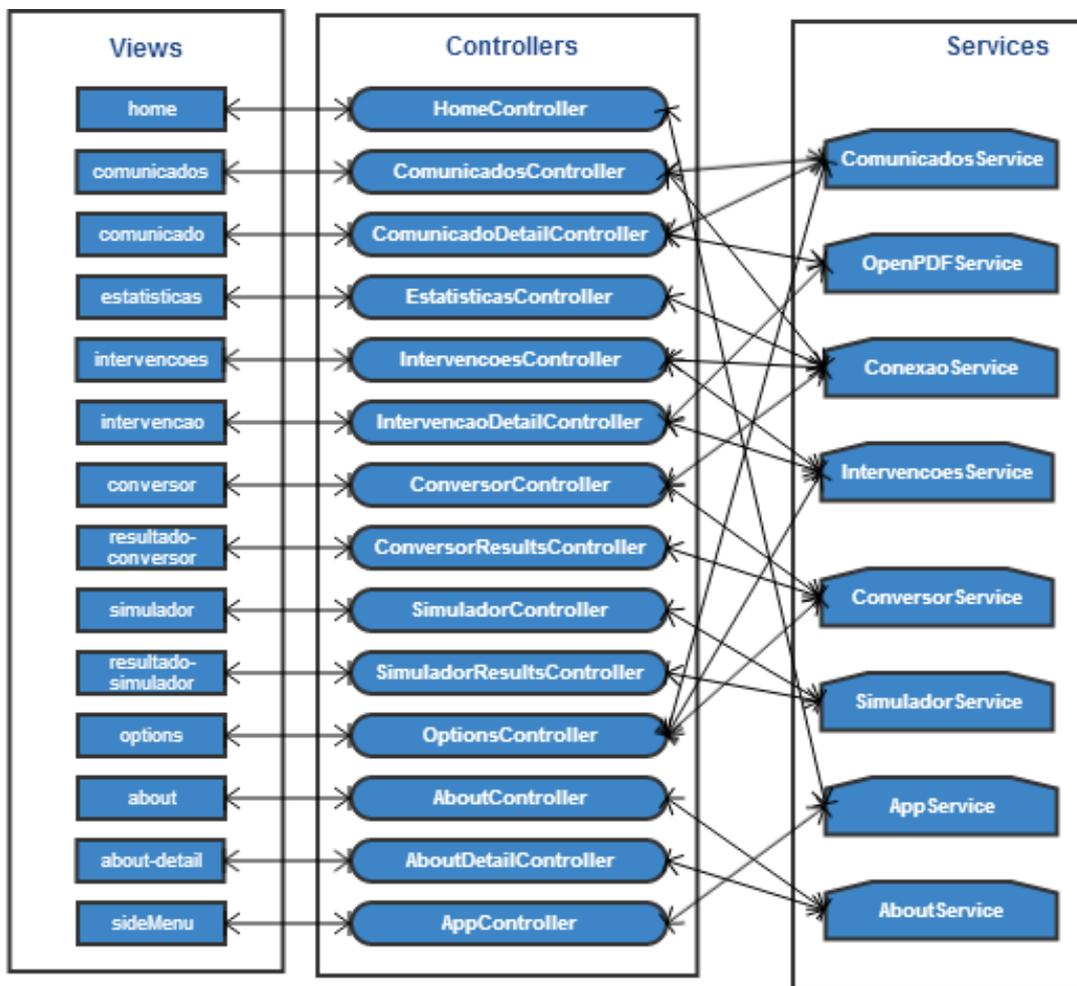


Figura 28 – Diagrama com todas as *Views*, *Controllers* e *Services* criados na aplicação.

Views

Foram criados catorze ficheiros HTML (*templates*) correspondentes às catorze *views* dos diferentes estados da aplicação. Cada *template* contém o código HTML referente às diferentes funcionalidades.

Controllers

Tal como nas *views* foram criados catorze *controllers* para os diferentes estados da aplicação, ou seja, um *controller* para cada *view*. O *controller* é responsável pelo comportamento por detrás do DOM, e é também responsável por inicializar e/ou adicionar comportamentos ao objeto \$scope (*model*), o qual permite a comunicação entre a *view* e o seu *controller*.

Em todas as *views* da aplicação foi adicionado no elemento <ion-header-bar> a diretiva ng-controller="AppController" para que aquela componente do *view* seja responsabilidade do *controller* AppController. Este *controller* tem a responsabilidade de adicionar o título e os botões de menu e de retroceder ao *header*, e ajustar a *view* de acordo com o tipo de dispositivo, ou seja, no caso de ser um *tablet* a *view* do menu é apresentada no lado esquerdo do ecrã de forma permanente (fica sempre aberto) e no restante espaço do ecrã é apresentado a *view* correspondente à ação que o utilizador escolheu. Nas Figuras 29 e 30 estão ilustradas duas imagens da aplicação quando acedida num *smartphone* e num *tablet*.



Figura 29 – Aplicação *mobile* quando acedida por um *tablet*.



Figura 30 – Aplicação *mobile* quando acedida por um *smartphone*.

Nas duas figuras acima, fica claro a lógica adicionada pelo *controller* *AppController* descrita anteriormente. Podemos ver que na versão *tablet* quando comparado com a versão *smartphone*, além do menu ser apresentado, o botão para o abrir o menu é ocultado. A lógica dos *controllers* pode ser simplificada através do uso de *services*. *Services* são objetos que realizam tarefas específicas comuns dentro da aplicação. Estes componentes são utilizados para compartilhar informações entre *controllers*, fazer a comunicação com o servidor e implementar a lógica de negócio, simplificando assim o trabalho do *controller*, que fica apenas com a responsabilidade de cuidar do fluxo de ações da aplicação. Todos os *controllers*, podem assim invocar métodos de mais do que um serviço consoante a necessidade do mesmo. Um exemplo de um *controller* que faz uso de *services* para implementar a lógica de negócio e fazer a comunicação com o servidor é o *controller* *ComunicadosController*. Este possui dependências de dois serviços, nomeadamente o *ComunicadosService* e o *ConexaoService* como ilustrado no diagrama da Figura 28.

Cordova Plugins

Os *plugins* do Apache Cordova são ferramentas importantes que visam melhorar os recursos e funcionalidades das aplicações móveis desenvolvidas em HTML5. Um *plugin* é um pacote de código que permite a uma aplicação HTML5 comunicar com a plataforma nativa na qual está a ser executada. Os *plugins* fornecem acesso a funcionalidades da plataforma que normalmente não estão disponíveis a aplicações baseadas em *browsers*, ou seja, fornecem uma forma de estender a API JavaScript da aplicação, resultando numa melhor integração com o software e hardware de cada dispositivo.

Foram usados oito *plugins* Cordova no desenvolvimento da aplicação, sendo que todos eles foram adicionados ao projeto diretamente do repositório Github sem necessidade de fazer nenhuma alteração, ou seja, o uso destes não exigiu nenhuma configuração ou programação adicional, mas apenas aprender a API JavaScript dos mesmos. Os *plugins* usados foram os seguintes:

- **Keyboard** – Permite fazer diversas operações com o teclado virtual. Usado para fazer aparecer e desaparecer o teclado sempre que necessário;
- **Datapicker** – Permite obter o *datapicker* nativo de cada dispositivo. Usado no conversor de moeda no campo de *input* “DATA”;
- **SpinnerDialog** – Usado sempre que o utilizador precisa de aguardar por uma resposta da aplicação;

- **Dialogs** – Usado para apresentar mensagens de alerta de acordo com cada dispositivo;
- **File** – Usado para fazer manipulação de ficheiros e o *download* dos ficheiros PDF;
- **Inappbrowser** – Fornece a capacidade de gerar uma instância do *browser* a partir da aplicação. Usado para dar acesso a aplicação BPstat *Mobile*;
- **Network-information** – Fornece informações do tipo de ligação a internet. Usado para detetar se o dispositivo tem ou não estabelecida uma ligação a internet;
- **SplashScreen** – Usado na apresentação da aplicação enquanto está a ser carregada. É uma imagem com o logótipo do Banco e uma mensagem de boas vindas.

Além dos *plugins* acima descritos foi necessário desenvolver um *plugin* específico para efetuar os cálculos do simulador de taxas de juro. Os cálculos em JavaScript não são precisos principalmente quando se trabalha com números extensos. A causa destes erros deve-se ao fato do JavaScript não ter tipos de dados específicos, pelo que quando se declara uma variável, não é definido a precisão de armazenamento.

Para resolver este problema foi desenvolvido um *plugin* nativo para as plataformas Android e iOS, responsável por todos os cálculos necessários numa simulação de taxas de juro. O *plugin* é composto por uma API JavaScript e por dois módulos de código nativo (um para cada plataforma). Estes módulos suportam a API JavaScript do *plugin*, ou seja, quando a aplicação chama a API JavaScript, ela é redirecionada para o módulo de código nativo que suporta e dá acesso a API nativa do dispositivo, permitindo assim que os cálculos sejam efetuados com maior precisão. No Anexo G.7 está ilustrado o código da API JavaScript do *plugin* desenvolvido e também do módulo nativo para iOS.

6.3 Protótipo do *site*

Para o desenvolvimento do protótipo do *site* foi escolhida a *framework* Twitter Bootstrap. A escolha deve-se ao facto de ser uma *framework* de *front-end* muito poderosa, elegante e intuitiva, que permite desenvolver aplicações *web* de forma fácil e rápida.

Como o objetivo era a construção de uma aplicação *web responsive*, a primeira tarefa passou por definir as larguras nos quais iam ser feitas as quebras de *layout*, ou seja, a partir de que dimensões o *layout* se adapta para *smartphones*, *tablets* e *desktops*. Foram então definidos os seguintes tamanhos para os diferentes dispositivos de acesso:

- Até 480px considerado um *smartphone*;
- De 480px até 640px considerado *tablet* mini;
- De 640px até 768px considerado *tablet* normal;
- Maior que 768px considerado *desktop*.

O próximo passo foi a criação do *layout*. Este foi estruturado usando a técnica conhecida como *layout* fluído, ou seja os tamanhos dos diferentes elementos do *site* foram criados com tamanhos relativos (%) ao invés de absolutos (px), permitindo assim que o *site* e todas as suas subpartes ocupem uma área relativa ao tamanho do ecrã dos dispositivos. Foram criadas as seguintes secções:

- **Header** – Zona de identificação do *site*, contém a pesquisa e o menu;
- **Secção de apresentação do *site*** – Zona onde é descrita uma breve história do Banco de Portugal e também um carrossel de imagens ilustrativas do mesmo;
- **Secção das taxas e publicações** – Secção composta por quatro colunas. Uma coluna com as principais taxas de juro, outra a apresentar o último artigo publicado pelo Banco de Portugal, outra com as principais taxas de câmbio de referência, e por último uma coluna com um conversor de moeda;
- **Secção de destaques e agenda** – Secção composta por duas colunas. Uma coluna para os destaques onde são apresentados a lista dos últimos destaques do Banco de Portugal, e outra com a agenda do Banco, onde é apresentada a lista dos principais eventos que irão acontecer;
- **Secção de atalhos** – Secção que dá acesso direto a outros portais do Banco de Portugal, nomeadamente o Museu, o Recrutamento, Portal do cliente Bancário e o BPstat;
- **Footer** – Zona onde se encontra o mapa do *site*, os termos e condições e as políticas de privacidade.

Após a criação do *layout*, definiram-se os estilos gerais da página, foram depois acrescentados os blocos iniciados por *media queries*, que são destinados a dispositivos com características específicas (neste caso apenas a largura e orientação do ecrã). As medidas usadas e descritas acima referem-se à largura da *viewport*, ou seja em dispositivos móveis a largura do ecrã e em *desktops* a largura da janela do navegador. As *media queries* permitem definir estilos específicos aos diferentes elementos que compõem o *site*, permitindo assim, que todo o conteúdo seja apresentado de forma otimizada independentemente da dimensão do ecrã do dispositivo de acesso.

Além da folha de estilos, foi também criado um ficheiro JavaScript responsável por toda a invocação a serviços e *parsing* de informação. Nas Figuras 31, 32 e 33 estão

ilustradas três imagens da aplicação *web responsive* desenvolvida quando acedida por um *desktop*, *tablet* e *smartphone* respetivamente.



Figura 31 – Protótipo do site desenvolvido quando acedido num desktop.



Figura 32 – Protótipo do site desenvolvido quando acedido num tablet



Figura 33 – Protótipo do site desenvolvido quando acedido num smartphone.

Capítulo 7

Testes

Os sistemas de software estão cada vez mais presentes nas nossas vidas, para que estes sejam disponibilizados com qualidade ao utilizador final, é necessário efetuar rigorosos testes ao longo dos vários estádios do desenvolvimento.

Um teste de software é o processo de execução de um produto que visa determinar se este cumpre as suas especificações e funciona corretamente no ambiente para o qual foi projetado. O objetivo desta fase é revelar todas as falhas do sistema, para que as mesmas possam ser identificadas e corrigidas antes da entrega final, ou seja da passagem a produção.

Neste capítulo serão explicados todos os testes realizados e a sua importância no processo de garantia de qualidade.

7.1 Testes de sistema

Após a integração de todos os módulos do sistema foi necessário efetuar um conjunto de testes para garantir que de facto o sistema funciona corretamente como um todo. O objetivo é interagir com o sistema sob o ponto de vista do utilizador final, percorrendo todas as funcionalidades na procura de alguma falha ou mau funcionamento relativamente aos objetivos esperados.

Para a realização destes testes foi necessário instalar os *web services* no ambiente de produção do Banco de Portugal, e configurar as aplicações cliente para que consumissem estes serviços. O ambiente de produção contém os conteúdos atualizados, e permite efetuar testes sem influências externas que possam comprometer os mesmos.

Foram realizados testes conforme estipulado na tabela de requisitos funcionais e de teste descrito na secção 4.3.2 . Para a sua realização foi criado um documento “Testes de sistema” presente no Anexo C, com diversas tarefas a realizar pelos *testers*, na procura de erros funcionais ou não funcionais da aplicação. O objetivo passa por

percorrer todos os serviços e funcionalidades da aplicação, na procura de anomalias ou pequenos erros, e identificá-los no documento para correção futura. Os erros identificados foram os seguintes:

- Arredondamentos do conversor de moeda diferentes do conversor do *site* institucional;
- Arredondamentos do simulador de taxas de juro diferentes do simulador de taxas de juro do portal do cliente bancário;
- Tempo de carregamento das intervenções públicas e comunicados demasiado longo, ocorrendo erro de ligação algumas vezes.

Os erros de arredondamento ocorriam porque o modelo usado nos cálculos era o arredondamento aritmético, quando o modelo correto deveria ser o “*Banker’s Rounding*”. Foram então alterados os dois serviços, de forma que o arredondamento usado passasse a ser o correto. Relativamente ao problema de tempo de carregamento, não foi considerado nesta fase, deixando para análise mais detalhada nos testes de carga.

7.2 Testes de carga

Os testes de carga são usados para avaliar e validar a aceitabilidade dos limites operacionais de um sistema de acordo com cargas de trabalho variáveis. O principal objetivo deste tipo de testes é encontrar o limite de capacidade da aplicação e identificar qual o limitante (codificação, hardware, ou tempo de resposta). Para a realização destes testes foi necessário a ajuda de um técnico do Núcleo de Engenharia de Software, o qual ajudou na preparação, execução e análise de resultados.

Para a realização destes testes foi necessário perceber junto do cliente quais os objetivos relativamente ao número de utilizadores simultâneos, e tempos de resposta para cada serviço, para que depois os mesmos fossem preparados e executados. Assim, os objetivos pretendidos pelo cliente foram os seguintes:

- **Número de utilizadores em simultâneo:** 50;
- **Utilizadores a fazer pedidos em inglês:** 25%;
- **Utilizadores a fazer pedidos em português:** 75%;
- **Tempo médio de resposta a um pedido:** 5 segundos.

Por outro lado e ainda no âmbito de preparação dos testes foi necessário definir quais os serviços a testar. Os serviços a testar são os seguintes:

- <http://services.bportugal.pt/bma/comunicados.svc/Comunicados/en-us/50>;
- <http://services.bportugal.pt/bma/comunicados.svc/Comunicados/pt-pt/50>;

- <http://services.bportugal.pt/bma/intervencoespublicas.svc/intervencoespublicas/pt-pt/50>
- <http://services.bportugal.pt/bma/intervencoespublicas.svc/intervencoespublicas/en-us/50>;
- <http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/pt-pt/date/2014-06-23>;
- <http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/en-us/date/2014-06-23>;

Para a realização destes testes foi criado um projeto “*Web Performance and Load Testing*” com o Visual Studio 2012, com diferentes condições de execução de forma a determinar a capacidade de resposta dos *web services* face às condições requeridas.

Nas secções seguintes, serão apresentados os testes realizados, os resultados obtidos e a resolução. Uma visão mais detalhada pode ser encontrada no documento “Testes de Carga” apresentados no Anexo D.

7.2.1 Teste 1 (*Web services sem implementação de cache*)

O primeiro teste foi configurado de acordo com as expectativas do cliente, e tinha como objetivo analisar a atual implementação face aos objetivos do mesmo. Assim, estes foram executados segundo as condições apresentadas na Tabela 7.

Condições de execução	
Utilizadores	50
Tempo de Execução	10 Minutos
Warm-up	30 Segundos
Think time	5 Segundos
Percentagem de testes em Inglês	25%
Percentagem de testes em Português	75%

Tabela 7 – Condições de execução do primeiro teste de carga

Os resultados obtidos estão ilustrados na Tabela 8, na qual estão representados o número de testes efetuados, indicando o número de sucessos e insucessos e o tempo médio para a execução de um teste. Na Tabela 9 são apresentados os tipos de erro.

Nome	Total de testes	Testes c/ sucesso (% do total)	Testes c/ erros (% do total)	Tempo médio de resposta (seg)
TestesEN	674	661 (98,07%)	13 (1,93%)	0,40 Segundos
TestesPT	2138	2121 (99,2%)	17 (0,80%)	0,27 Segundos

Tabela 8 – Resultados da execução do primeiro teste de carga

Erros			
Tipo	Subtipo	Numero	Mensagem de erro
HTTP Error	400 - BadRequest	20	400 - BadRequest
HTTP Error	403 - Forbidden	10	403 - Forbidden

Tabela 9 – Detalhes dos erros do primeiro teste de carga

Apesar dos testes serem executados apenas durante 10 minutos, ocorreram 30 erros não previstos, dos quais 20 do tipo “400 – *BadRequest*” e 10 do tipo “403 – *Forbidden*”. Os erros “403 – *Forbidden*” ocorreram porque os *web services* excederam o limite configurado de pedidos por minuto à camada de dados, enquanto os “400 - *BadRequest*” ocorreram porque o objeto que faz a leitura da *stream* de dados não era fechado corretamente “.close()”, o que provocava os erros quando havia concorrência de utilizadores.

Para resolver estes problemas, foi implementado um mecanismo de *cache* nos serviços, de forma a reduzir a carga de pedidos à camada de dados e corrigido o problema de leitura de dados. O sistema de *cache* implementado ficou com uma duração de 20 minutos, ou seja, após este tempo, os dados em *cache* são invalidados e só novos pedidos atualizam a *cache*.

7.2.2 Teste 2 (*Web services* com correções após o teste 1)

Após as alterações apresentadas no teste 1, foi feito outro teste, no entanto, com configurações diferentes como mostra a Tabela 10, com tempo de execução mais exigente.

Condições de execução	
Utilizadores	50
Tempo de Execução	20 Minutos
Warm-up	1 Minuto
Think time	5 Segundos
Percentagem de testes em Inglês	25%
Percentagem de testes em Português	75%

Tabela 10 – Condições de execução do segundo teste de carga

Este teste, contrariamente ao primeiro, não gerou erros e demonstrou uma melhoria significativa dos tempos de resposta, o que provou que as alterações efetuadas foram eficazes. Abaixo, na Tabela 11 estão ilustrados os resultados do teste 2, na qual podemos observar que os resultados obtidos asseguram os objetivos do cliente face ao número de utilizadores simultâneos e aos tempos de resposta na invocação de serviços.

Nome	Total de testes	Testes c/ sucesso (% do total)	Testes c/ erros (% do total)	Tempo médio de resposta (seg)
TestesEN	1692	1692(100%)	0 (0%)	0.024 Segundos
TestesPT	4701	4701(100%)	0 (0%)	0.077 Segundos

Tabela 11 – Resultados da execução do segundo teste de carga

7.2.3 Teste 3 (Quatro vezes mais carga relativamente ao teste 2)

O teste 3 serviu apenas para perceber se os serviços conseguiam aguentar uma carga quatro vezes superior à do teste 2. Na Tabela 12 estão representadas as condições de execução do teste 3.

Condições de execução	
Utilizadores	200
Tempo de Execução	20 Minutos
Warm-up	1 Minuto
Think time	5 Segundos
Percentagem de testes em Inglês	25%
Percentagem de testes em Português	75%

Tabela 12 – Condições de execução do terceiro teste de carga

O teste 3 foi configurado com as mesmas variáveis do teste 2, no entanto, aumenta o número para 200 utilizadores em simultâneo. Os resultados obtidos estão representados na Tabela 13, e superam largamente as expectativas. Num total de 24.304 testes, 100% dos testes foram realizados com sucesso com tempos médios de resposta muito abaixo do máximo pretendido pelo cliente.

Nome	Total de testes	Testes c/ sucesso (% do total)	Testes c/ erros (% do total)	Tempo médio de resposta (seg)
TestesEN	6046	6046 (100%)	0 (0%)	0,13 Segundos
TestesPT	18258	18258 (100%)	0 (0%)	0,014 Segundos

Tabela 13 – Resultados da execução do terceiro teste de carga

Em suma, os testes de carga foram extremamente importantes pois permitiram encontrar limitações no código que impediam a resposta adequada a 50 utilizadores em simultâneo, e corrigi-los antes da entrega final de maneira que a atual implementação permite uma carga quatro vezes superior à requerida inicialmente pelo cliente.

7.3 Testes de usabilidade

Os testes de usabilidade têm como objetivo verificar a facilidade que aplicação possui de ser claramente compreendida e manipulada pelo utilizador. Todos os testes efetuados são focados na avaliação da usabilidade da interface da aplicação *mobile*, nos diversos dispositivos do utilizador.

A metodologia utilizada para os testes foi a avaliação somativa. O propósito da avaliação somativa passa por avaliar quão bem um produto cumpre os seus objetivos após implementação e tendo em vista melhorias futuras. Para a realização destes testes foi elaborado um questionário de usabilidade, dividido em quatro partes distintas:

- **Dados Pessoais** – Identificação do utilizador e sua experiência com este tipo de aplicações;
- **Dados para o teste** – Informações do dispositivo e plataforma no qual será realizado o teste (útil para os testes de compatibilidade);
- **Tarefas de teste** – Tarefas a executar e avaliar pelo utilizador;
- **Avaliação de usabilidade (SUS)** – Questionário que avalia a concordância dos utilizadores com o sistema.

Ao todo foram quinze os utilizadores que realizaram o questionário de usabilidade, sendo que 100% destes usam regularmente dispositivos móveis, no entanto, apenas 60% está familiarizado com os serviços disponibilizados na aplicação. Relativamente aos dispositivos de teste 40% destes foram realizados em dispositivos iOS e 60% em dispositivos Android. A realização deste questionário pressupõe avaliar dois cenários diferentes: Sucesso na realização das tarefas e nível de experiencia de usabilidade. No entanto, foi também usado para recolher opiniões sobre como melhorar a usabilidade das tarefas realizadas.

A todos os utilizadores foi dado o questionário “Questionário de usabilidade” presente no Anexo E, e instalada a aplicação nos seus dispositivos móveis.

7.3.1 Sucesso na realização de tarefas

O sucesso na realização de uma tarefa prevê um teste com início e fim bem definidos. A métrica para analisar a realização de uma tarefa foi classificá-la com sucesso binário; conclui a tarefa (1) ou não conclui a tarefa (0). Todos os utilizadores conseguiram terminar as tarefas com sucesso, no entanto, nem todos tiveram a mesma facilidade na realização das mesmas. Para medir o nível de sucesso atribuiu-se um valor numérico para a maneira como o sucesso foi alcançado: muito fácil (1), fácil (2), pouco fácil (3), difícil (4), muito difícil (5).

7.3.2 Avaliação de usabilidade

Segundo Jakob Nielsen, no cenário de usabilidade global, o utilizador é o elemento chave [23]. Tudo o que este diz ou pensa sobre o sistema é a base para a realização do teste. Independentemente da forma como o utilizador interage com o sistema, o que importa é como este se sente no final (satisfeito ou insatisfeito). Neste caso, a métrica usada foi um questionário global da aplicação. O questionário usado foi o *System Usability Scale* (SUS), desenvolvido por Jonh Brooke [24] que é composto por 10 afirmações numa escala de 5 pontos onde os inquiridos avaliam o seu nível de concordância com a aplicação. Metade das questões está redigida de forma positiva e a outra metade está redigida de forma negativa, para obrigar o utilizador a estar atento quando estiver a responder ao questionário. Deste modo as questões com o índice ímpar são pontuadas da seguinte forma: discordo totalmente (0), discordo (1), neutro (2), concordo (3), concordo totalmente (4). Por outro lado as questões com o índice par são pontuadas da seguinte forma: discordo totalmente (4), discordo (3), neutro (2), concordo (1), concordo totalmente (0). A soma das contagens deve ser multiplicada 2,5 para

obter o valor global de usabilidade do sistema. As pontuações do SUS têm uma gama de 0 a 100. Pontuações abaixo de 60 pontos representam sistemas com experiências relativamente pobres e insatisfação do utilizador, e pontuações acima de 80 pontos representam experiências muito boas, com alto índice de satisfação dos utilizadores. Este questionário (avaliação de usabilidade) faz parte do “questionário de usabilidade” dado ao utilizador e presente no Anexo E.

7.3.3 Resultados

Os 15 inquiridos completaram as sete tarefas do questionário com êxito. Considerando apenas o sucesso binário, o aproveitamento foi de 100%, no entanto, o nível de sucesso na realização das tarefas foi um pouco diferente, como se pode ver pelo gráfico ilustrado na Figura 34.

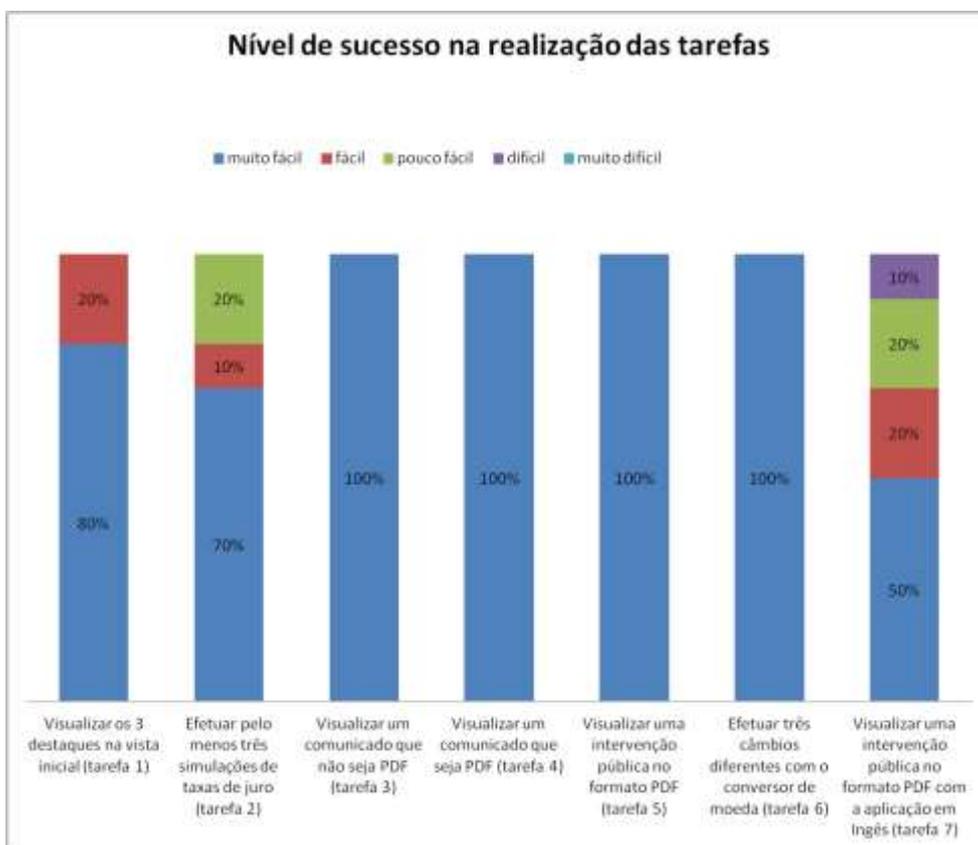


Figura 34 – Gráfico do nível de sucesso das diferentes tarefas baseada na dificuldade para completar uma tarefa

Em geral todos os utilizadores conseguiram identificar facilmente o menu, e perceber o objetivo de cada tarefa. As tarefas 3, 4, 5 e 6, referentes a visualização de comunicados e intervenções públicas foram as mais fáceis de realizar, não surgindo qualquer dúvida, o que prova que estas são claras e fáceis de aprender segundo as boas práticas de usabilidade. Na tarefa 1, 20% dos utilizadores respondeu que foi fácil

realizar a tarefa, pois acharam que a mesma pode ser melhorada através da inclusão de setas para a esquerda e direita para avançar ou retroceder para o destaque seguinte. Na tarefa 2, 10% dos utilizadores respondeu que foi fácil, e 20% respondeu que foi pouco fácil. As dificuldades encontradas nesta tarefa, prendem-se com o facto de o simulador não indicar as unidades nos campos de *input*, causando confusão nos valores a introduzir. Por fim na tarefa 7, 50% dos utilizadores responderam que foi muito fácil realizar a tarefa, 20% que foi fácil, 20% que foi pouco fácil e 10% respondeu que foi difícil. Foram apresentadas duas justificações para as dificuldades apresentadas na realização desta tarefa. A primeira deve-se à má compreensão da tarefa, alguns utilizadores pensavam que a lista de intervenções públicas tinha uma intervenção que estava no idioma inglês, só percebendo o objetivo da tarefa após ajuda, e a outra dificuldade foi devida à identificação no item de menu “opções”, que não é suficientemente claro que permite alterar o idioma da aplicação.

Todos os dados foram registados, e as tarefas que apresentavam dificuldades ao utilizador foram melhoradas, no entanto, por falta de tempo, a inclusão das unidades nos valores de *input* no simulador de taxas de juro, apenas deve ser considerada numa próxima versão da aplicação, uma vez que claramente melhorava a compreensão para os utilizadores.

Relativamente a avaliação de usabilidade (SUS) as pontuações atribuídas por cada utilizador estão ilustradas no gráfico da Figura 35.

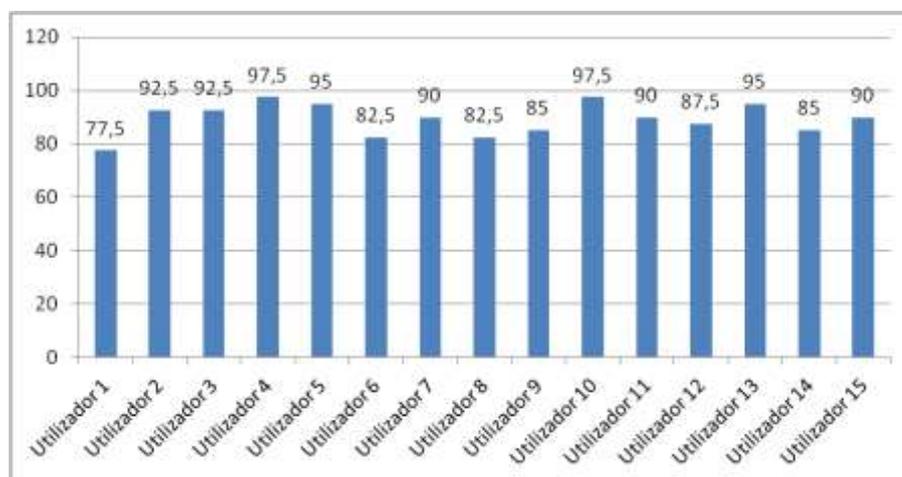


Figura 35 – Gráfico com as pontuações SUS de todos os utilizadores que responderam ao questionário.

Média das pontuações SUS: 89,33

A média das pontuações foi 89,33 o que prova que a experiência de utilização da aplicação foi positiva. Se olharmos para a classificação dos utilizadores, vemos que apenas um atribuiu pontuação abaixo de 80 pontos, e todos os outros pontuações acima, o que apenas é possível com interfaces muito boas [24].

Em suma, todos os utilizadores consideraram a aplicação com uma interface muito simples, agradável e fácil de aprender.

7.4 Testes de Compatibilidade

Os testes de compatibilidade visam assegurar a compatibilidade das aplicações entre diferentes ambientes e plataformas. O objetivo destes testes é garantir que a aplicação interage e comunica com diferentes versões de sistemas operativos e dispositivos, sem ocorrência de eventos indesejáveis como lentidão ou perda de dados, ou outro tipo de erros que impeça o uso normal da aplicação.

A escolha das *frameworks* de desenvolvimento restringiu a portabilidade da aplicação, pelo que esta primeira versão da aplicação apenas estará disponível para versões 6 ou superiores do iOS e 4 ou superiores do Android.

A aplicação foi testada nos seguintes dispositivos:

- iPhone 4;
- iPhone 4S;
- iPhone 5;
- iPad mini;
- iPad 64-bits;
- LG Nexus 5;
- Sony Ericsson Xperia;
- PadFone 2;
- Samsung galaxy S2.

Foi também testada em diversos simuladores iOS, tais como:

- iPhone;
- iPhone Retina;
- iPad;
- iPad Retina;
- iPad Retina 64-bits.

Em todos os dispositivos nos quais a aplicação foi testada, não foram detetados erros, no entanto, no Sony Ericsson Xperia com a versão 4.0 do Android a aplicação teve alguns problemas de desempenho, nomeadamente lentidão nas transições entre vistas da aplicação.

A aplicação foi igualmente testada nos seguintes sistemas operativos:

- iOS 6.1;
- iOS 7.0;
- iOS 7.1;
- Android 4.0;
- Android 4.1;
- Android 4.2;
- Android 4.3;
- Android 4.4.

Os testes nas diferentes versões dos sistemas operativos correram muito bem, e sem ocorrência de erros, com a exceção do Android 4.0 como já referido anteriormente.

Relativamente ao desempenho, a aplicação quando executada em dispositivos iOS tem um desempenho superior aos dispositivos Android, sendo que no Android 4.3 e 4.4 o desempenho é semelhante ao iOS.

Capítulo 8

Conclusões

Neste capítulo, são apresentadas as considerações finais, as contribuições deste projeto e por fim as perspectivas futuras.

8.1 Considerações Finais

Com base no desenvolvimento do projeto conducente a esta tese de mestrado, experiência adquirida e no estudo de soluções móveis, surgem assim as considerações finais.

A crescente evolução tecnológica na área *mobile* tem criado imensos desafios para os programadores. Cada vez mais existe uma maior fragmentação de dispositivos e plataformas, e desenvolver aplicações compatíveis com todas estas é uma tarefa que exige muito trabalho e dedicação.

Este trabalho apresenta os resultados de uma pesquisa na área do desenvolvimento de soluções para *mobile*, bem como do desenvolvimento de duas aplicações compatíveis com grande parte dos dispositivos atualmente existentes. Os objetivos propostos pelo Banco de Portugal foram cumpridos na íntegra, e ambas as aplicações cumprem todos os requisitos definidos. Os testes de software foram fundamentais na validação dos objetivos inicialmente propostos. Permitiram aumentar a capacidade de resposta das aplicações quanto ao número de utilizadores em simultâneo, corrigir falhas e problemas de compatibilidade em diversos dispositivos e simplificar a usabilidade das aplicações. No decorrer do projeto, ficou evidente a verdadeira utilidade destes testes, pois a quantidade de dispositivos é enorme e garantir que a aplicação funciona em todos, requer testar iterativamente a aplicação em diversos dispositivos à medida que esta vai ganhando novas funcionalidades e não apenas no final.

Desenvolver *sites* seguindo uma abordagem *responsive* é exigente, pois trabalhar com flexibilidade e adaptação de conteúdos é mais complexo do que com conteúdos

estáticos. Penso que esta abordagem se irá tornar rapidamente o *standard*, pois permite lidar com a proliferação de dispositivos de diferentes formatos e contextos, mantendo uma unificação de conteúdos.

Relativamente à estratégia de desenvolvimento de aplicações híbridas, embora ainda extremamente recente e com algumas limitações, o desenvolvimento destas aplicações tem um enorme potencial. No entanto, há ainda muito por fazer para que se torne uma estratégia sólida e confiável por grande parte da comunidade informática. A melhoria progressiva da capacidade dos dispositivos móveis, e a evolução do HTML5, são sem dúvida dois fatores que podem tornar esta estratégia mais popular junto da comunidade. Para finalizar, e de uma forma mais subjetiva, penso que as bases estão lançadas e no futuro esta será a estratégia principal no desenvolvimento de aplicações móveis.

8.2 Contribuições

As principais contribuições desta dissertação são:

- **Um protótipo do *site* Institucional do Banco de Portugal;**
Foi desenvolvido um protótipo do *site* Institucional do Banco de Portugal, focado na implementação dos serviços da página inicial. É uma aplicação *web responsive*, que adapta a exibição dos seus conteúdos de acordo com a dimensão e orientação do dispositivo de acesso.
- **Uma aplicação *mobile* disponível para os sistemas operativos Android e iOS nas versões *smartphone* e *tablet*;**
Foi desenvolvida uma aplicação *mobile* otimizada para as versões *tablet* e *smartphone* dos sistemas operativos Android e iOS. A aplicação foi submetida a rigorosos testes de software, e possibilita aos utilizadores acesso a serviços de forma otimizada em qualquer dispositivo.
- **Uma API REST que disponibiliza informação pública do Banco de Portugal;**
Foi desenvolvida uma API RESTful para dar suporte às aplicações cliente, que no futuro e se o Banco de Portugal assim o entender, poderá ser disponibilizada para o exterior para que outras aplicações tenham acesso a informação do Banco de Portugal, sem necessidade de obtê-la através da extração do HTML como acontece atualmente.
- **Um *plugin* Cordova para os sistemas operativos Android e iOS responsável por efetuar cálculos numéricos com maior precisão.**

Foi desenvolvido um *plugin* Cordova para efetuar os cálculos numéricos com código nativo, pois, uma das limitações do JavaScript é a falta de precisão em cálculos com maior complexidade. O *plugin* ainda não foi registado, pois precisa de algumas alterações para que fique mais genérico de modo a que outros utilizadores também o possam usar.

8.3 Perspetivas Futuras

Como perspetiva de trabalho futuro, existem cinco pontos na aplicação *mobile* que podem constituir meios evolutivos para a mesma, nomeadamente:

- Criação de um sistema de notificações para informar o utilizador sempre que um novo conteúdo (comunicado ou intervenção pública) é publicado pelo Banco de Portugal;
- Melhoria dos serviços do conversor de moeda e do simulador de taxas de juro, para que passem a apresentar as unidades dos valores de *input* pretendidos;
- Adição de outros serviços relevantes para o Banco de Portugal, como exemplo as publicações e a agenda;
- Disponibilização da aplicação no sistema operativo Windows Phone;
- Disponibilização do *plugin* responsável pelos cálculos, a toda a comunidade Cordova.

Estes são os pontos prioritários para o futuro da aplicação, no entanto, numa perspetiva mais a curto prazo o próximo passo passará por colocar a aplicação no Google Play e na App Store para que os utilizadores possam descarregá-la e usufruir da mesma. Relativamente ao protótipo do *site*, não deverá ser alterado no futuro, apenas servirá de linha orientadora para uma proposta futura do novo *site* Institucional do Banco de Portugal.

Bibliografia

1. IBM Corporation Software Group. *Native, web or hybrid mobile-app development*. IBM Corporation, 2012
2. LionBridge. *Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice*. LionBridge, 2012
3. Budiu, Raluca. *Mobile: Native Apps, Web Apps, and Hybrid App*. Nielsen Norman Group, 2013
4. Adobe Systems Inc. *Easily create apps using the web technologies you know and love: HTML, CSS and JavaScript; Phonegap* [Em linha]. US in San Francisco, 2013 [consultado 15 Outubro 2013]. Disponível na World Wide Web: <<http://phonegap.com/>>
5. Marcotte, Ethan. *Responsive web design*. Editions Eyrolles, 2011.
6. De Graeve, Katrien. *Responsive Web Design* [Em linha]. Novembro, 2011 [consultado 15 Outubro 2013]. Disponível na World Wide Web: <<http://msdn.microsoft.com/en-us/magazine/hh653584.aspx>>
7. Lopes, Sergio. *Design Responsivo por uma web única* [Em linha]. 15 de Abril, 2012 [consultado 24 Outubro 2013]. Disponível na World Wide Web: <<http://sergiolopes.org/responsive-web-design/#sthash.Gk5v2hug.dpuf>>
8. Google Developers. *Building Smartphone-Optimized Websites* [Em linha]. 16 de Abril, 2013 [consultado 25 Outubro 2013]. Disponível na World Wide Web: <<https://developers.google.com/webmasters/smartphone-sites/details>>
9. Omniupdate. *Adaptive Web Design*. Omniupdate resources, 2011.
10. Gustafson, Aaron. *Adaptive web design: Crafting Rich Experiences with Progressive Enhancement*. Easy Readers, LLC; 1st edition, 2011.
11. Warner, Janine. *What's the difference between adaptive and responsive web design* [Em linha]. DigitalFamily, 11 Abril, 2013 [consultado 28 Outubro 2013].

- Disponível na World Wide Web: <<http://www.digitalfamily.com/tutorials/css-article/whats-the-difference-between-adaptive-and-responsive-web-design/>>
12. Boudreaux, Ryan. *What is the difference between responsive vs adaptive web design* [Em linha]. Techrepublic, 11 Abril, 2013 [consultado 29 Outubro 2013]. Disponível na World Wide Web: <<http://www.techrepublic.com/blog/web-designer/what-is-the-difference-between-responsive-vs-adaptive-web-design/>>
 13. World Wide Web Consortium (W3C). *A vocabulary and associated APIs for HTML and XHTML* [Em linha]. W3C 31 Julho, 2014 [consultado 5 Novembro 2013]. Disponível na World Wide Web: <<http://www.w3.org/TR/html5/>>
 14. World Wide Web Consortium (W3C). *HTML5* [Em linha]. W3C, 2012 [consultado 5 Novembro 2013]. Disponível na World Wide Web: <<http://www.w3.org/html/logo/>>
 15. Sencha Touch. *HTML5 Mobile app development*. [Em linha]. Sencha Touch, 2013 [consultado 20 Novembro 2013]. Disponível na World Wide Web: <<http://www.sencha.com/products/touch/>>
 16. Bootstrap. *Bootstrap front-end framework*. [Em linha]. Bootstrap, 2013 [consultado 20 Novembro 2013]. Disponível na World Wide Web: <<http://getbootstrap.com>>
 17. JQuery Mobile. *JQuery Mobile*. [Em linha]. JQuery Mobile, 2013 [consultado 21 Novembro 2013]. Disponível na World Wide Web: <<http://jquerymobile.com/>>
 18. Modernizr. *Modernizr*. [Em linha]. Modernizr, 2013 [consultado 21 Novembro 2013]. Disponível na World Wide Web: <<http://modernizr.com/>>
 19. Banco de Portugal. *O Banco e o Eurosistema*. [Em linha]. Banco de Portugal, 2012 [consultado 2 Dezembro 2013] Disponível na World Wide Web: <<http://www.bportugal.pt/pt-PT/OBancoeoEurosistema/Paginas/default.aspx>>
 20. Banco de Portugal. *Organograma do Banco de Portugal*. [Em linha]. Banco de Portugal, 2012 [consultado 2 Dezembro 2013] Disponível na World Wide Web: <<http://www.bportugal.pt/ptPT/OBancoeoEurosistema/OrganizacaoEstrutura/Paginas/Organograma.aspx>>
 21. Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. The McGraw-Hill Companies, 2001.

22. AngularJs. *Superheroic JavaScript MVC Framework* [Em linha]. Google, 2014 [consultado 12 Fevereiro 2014] Disponível na World Wide Web: <<https://angularjs.org/>>
23. Nielsen, Jakob. *Usability 101: Introduction to usability*. Nielsen Norman Group, 2012
24. John, J. *SUS: A quick and dirty usability scale*. In Jordan, P. W. B. A.; Weerdmeester and McClelland, I. L. (Eds). *Usability Evaluation in Industry*, 1996, 189–94.

Capítulo 9

Anexos

Anexo A

A.1. Relatório de análise

Especificação de requisitos

Componente web

A componente *web* será apenas uma prova de conceito, ou seja, o objetivo é desenvolver um protótipo do *site* institucional focado apenas na página inicial, que mostre uma possível solução de como o novo *site* institucional deverá ser implementado. Esta componente não será colocada em produção. Apenas deve implementar as principais funcionalidades do ponto de vista do utilizador. Deverá ser capaz de adaptar-se automaticamente aos diferentes tamanhos de ecrã, proporcionando aos utilizadores uma ótima visualização e navegabilidade em qualquer dispositivo.

Uma vez que se trata de um protótipo, apenas terá um tipo de utilizador, que terá acesso total aos conteúdos.

Componente mobile

A componente *mobile* tem como objetivo a disponibilização de alguns dos principais serviços do Banco de Portugal num contexto *mobile*. Esta componente deverá ser colocada em produção aquando da sua conclusão. Os serviços a disponibilizar deverão ser identificados e analisados em cooperação com o GAB. A aplicação numa primeira fase deverá ser desenvolvida para as plataformas Android e iOS de forma otimizada para *smartphones* e *tablets*, no entanto deverá ser levado em conta a possibilidade de ser desenvolvida para Windows phone no futuro.

A aplicação apenas terá um tipo de utilizador que terá acesso total à aplicação, não sendo requerido nenhum login. A aplicação deverá ainda ser utilizável em modo *offline*.

API Web do Banco de Portugal

A API Web terá como objetivo a disponibilização dos serviços / conteúdos do Banco de Portugal. Será desenvolvida apenas para o uso nas aplicações acima descritas, no entanto poderá ser disponibilizada no futuro para programadores que queiram usar estes mesmos serviços. Deverá ser levado em conta a possibilidade de esta API crescer e disponibilizar muitos mais serviços no futuro.

Requisitos funcionais e de teste

Id	Componete	Nome	Descrição	Observações	Estado
RF01	Web	Visualizar as Taxas de Juro	Apresenta o valor das Taxas de Juro de referência na página inicial.	Os valores das taxas de juro devem ser os mais atuais. Não é necessário qualquer input por parte do utilizador.	
RF02	Web	Visualizar a última publicação	Apresentar informações referentes ao último artigo publicado na página inicial.	Deverá ser apresentado a capa e o título da última publicação.	
RF03	Web	Visualizar as Taxas de câmbio	Apresenta as cotações das principais moedas na página inicial.	Deverá apresentar as cotações das principais moedas do dia atual. As principais moedas são: Dólar dos Estados Unidos, Iene Japonês, Libra Esterlina, Franco Suíço, Real do Brasil.	
RF04	Web	Conversor de moeda	Permite efetuar uma conversão com base nas taxas do dia requerido.	Deverá efetuar o cálculo de conversão entre qualquer uma das moedas listadas na data requerida. Todos os campos são de preenchimento obrigatório.	
RF05	Web	Visualizar os comunicados mais recentes	Apresenta a lista dos comunicados mais recentes.	Deverá apresentar uma lista dos oito últimos comunicados ordenados por ordem cronológica. A data de publicação deverá ser exibida.	
RF06	Web	Visualizar as intervenções públicas mais recentes	Apresenta a lista das últimas intervenções públicas	Deverá apresentar uma lista das oito últimas intervenções públicas ordenadas por ordem cronológica. A data de publicação deverá ser exibida.	
RF07	Mobile	Simulador de Taxas de Juro	Permite efetuar simulações de Taxas de Juro	Deverá simular as Taxas de Juro, e apresentar os resultados do juro composto e do juro simples. Todos os campos são obrigatórios. Deverá ser possível fazer a simulação em modo offline.	
RF08	Mobile	Conversor de moeda	Permite efetuar uma conversão com base nas	Deverá efetuar o cálculo de conversão entre qualquer uma das	

			taxas do dia requerido.	moedas listadas na data requerida. Todos os campos são de preenchimento obrigatório.
RF09	<i>Mobile</i>	Visualizar os comunicados mais recentes.	Apresenta a lista dos comunicados mais recentes.	Deverá apresentar uma lista dos 50 últimos comunicados ordenados por ordem cronológica.
RF10	<i>Mobile</i>	Visualizar as principais intervenções do governador.	Apresenta a lista das últimas intervenções públicas	Deverá apresentar uma lista das 50 últimas intervenções públicas ordenadas por ordem cronológica.
RF11	<i>Mobile</i>	Visualizar os destaques específicos para a aplicação <i>mobile</i> .	Apresentar uma lista de 3 destaques na vista inicial da aplicação.	Deverá ser possível fazer “swipe” ao ecrã para visualizar o destaque seguinte.
RF12	<i>Mobile</i>	Aceder ao BPstat <i>mobile</i> na aplicação.	Incorpora o BPstat <i>mobile</i> numa <i>web view</i> na aplicação.	O BPstat <i>mobile</i> deverá manter a sua identidade gráfica.
RF13	<i>Mobile</i>	Aplicação bilingue	Permite escolher usar a aplicação em inglês ou português.	Todo o texto na aplicação deverá ser coerente com a linguagem selecionada.
RF12	<i>API Web</i>	Devolve os últimos comunicados ordenadas por data	Retorna a lista dos últimos 50 comunicados.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF13	<i>API Web</i>	Devolver as últimas Intervenções públicas ordenadas por data	Retorna a lista das últimas 50 intervenções na linguagem pretendida	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF14	<i>API Web</i>	Devolver o número especificado de intervenções públicas	Retorna a lista de intervenções, com o número de itens especificado como parâmetro.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF15	<i>API Web</i>	Devolver o número especificado de comunicados	Retorna a lista de comunicados, com o número de itens especificado como parâmetro.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF16	<i>API Web</i>	Devolver as taxas de juro	Devolve a lista das principais taxas de juro de referência.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF17	<i>API Web</i>	Devolver as taxas de câmbio do dia corrente	Devolve a lista das taxas de câmbio para a data introduzida.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF18	<i>API Web</i>	Devolver as últimas taxas de câmbio	Devolve a lista das últimas taxas de câmbio.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.
RF19	<i>API Web</i>	Devolver as taxas de câmbio de uma moeda	Devolve a lista das taxas de câmbio para o código introduzido.	Deverá ser possível pedir a resposta no formato xml ou json e na linguagem português ou inglês.

Requisitos não funcionais

Usabilidade – As três componentes deverão ser intuitivas e fáceis de usar.

Extensibilidade – A arquitetura do sistema deverá ser o mais flexível possível, de forma a permitir a adição de novos módulos e funcionalidades no futuro.

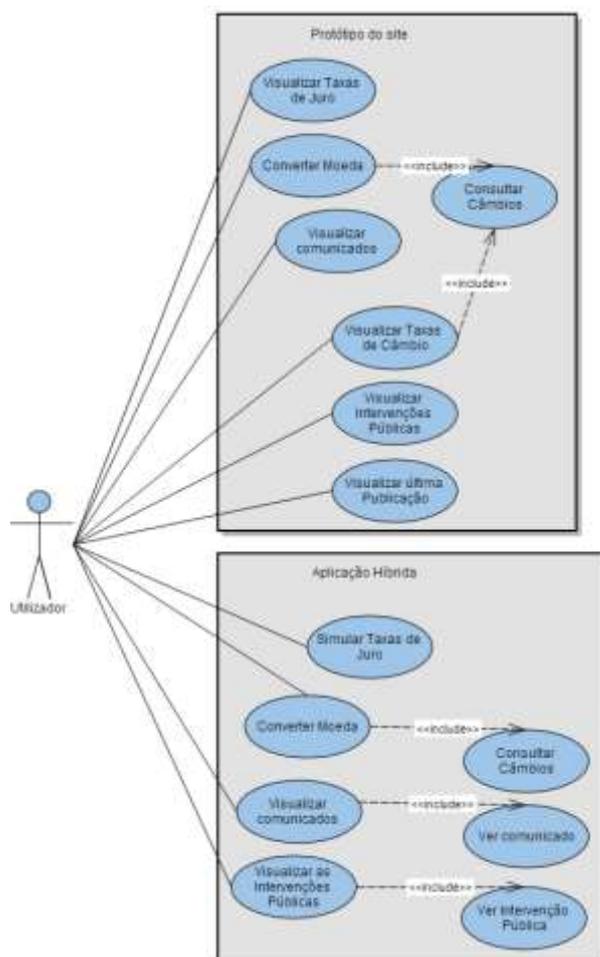
Modificabilidade – Deverá ser possível alterar os diferentes módulos do sistema sem comprometer-lo num todo.

Testabilidade – Deverá ser possível testar cada elemento em separado.

Confidencialidade – Não serão necessários quaisquer dados do utilizador para a utilização de qualquer componente do sistema.

Portabilidade – As componentes devem ser compatíveis com vários dispositivos.

Casos de uso



Caso de uso – Simular Taxas de Juro

Componente – *Mobile*

Objetivo - Efetuar uma simulação de Taxas de Juro.

Utilizador – Qualquer utilizador.

Fluxo de ações

6. Aceder a aplicação
7. Abrir o menu
8. Selecionar o item “Simulador de Taxas de Juro”
9. Preencher todos os campos:
 - a. Capital inicial – Número decimal positivo;
 - b. Prazo – Número inteiro positivo;
 - c. Tipo de prazo – Escolher Anos, Meses ou Dias;
 - d. Taxa anual – Número decimal positivo;
 - e. Capitalizações – Número inteiro positivo
10. Clicar no botão calcular

Resultado Expectável - É apresentada uma nova *vista* onde são apresentados os resultados dos juros e capitais simples/compostos no final do prazo.

Estado de erro - É apresentado um alerta com informação dos inputs inválidos.

Caso de uso – Conversor de moeda

Componente – *Mobile*

Objetivo – Fazer a conversão de 5 Euros para o Dólar Australiano no dia 12-12-2013.

Utilizador – Qualquer utilizador.

Fluxo de ações

6. Aceder a aplicação;
7. Abrir o menu;
8. Selecionar o item “Conversor de moeda”;
9. Preencher todos os campos;
 - a. Valor – Número decimal;
 - b. Data – O formato é apresentado;
 - c. De – Escolher na lista apresentada;
 - d. Para – Escolher na lista apresentada;
10. Clicar no botão calcular.

Resultado Expectável - É apresentada uma nova *vista* onde são apresentados os resultados da conversão, com os dados de input, e o resultado da conversão.

Estado de erro - É apresentado um alerta com informação dos inputs inválidos.

Caso de uso – Ver os comunicados em Inglês

Componente – *Mobile*

Objetivo - Mudar a linguagem da aplicação para Inglês e ver os comunicados mais recentes.

Utilizador – Qualquer utilizador.

Fluxo de ações

7. Aceder a aplicação;
8. Abrir o menu;
9. Selecionar o item “Opções”;

10. Mudar a linguagem para Inglês;
11. Voltar a abrir o menu;
12. Selecionar o item “Comunicados”;

Resultado Expectável - É apresentada uma nova *vista* com a listagem dos últimos 50 comunicados. Cada comunicado tem associado a data de publicação, um título, a informação a indicar se é um PDF e a referência para o conteúdo desse mesmo item.

Estado de erro – Sem conexão a internet.

Caso de uso – Ver as intervenções públicas

Componente – *Mobile*

Objetivo – Visualizar a lista das intervenções públicas

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder a aplicação;
2. Abrir o menu;
3. Selecionar o item “Intervenções públicas”;

Resultado Expectável - É apresentada uma nova *vista* com a listagem dos últimas 50 intervenções públicas. Cada intervenção tem associado a data de publicação, um título, a informação a indicar se é um PDF e a referência para o conteúdo desse mesmo item.

Estado de erro – Sem conexão a internet.

Caso de uso – Ver os destaques

Componente – *Mobile*

Objetivo – Visualizar a lista de destaques na vista inicial.

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder a aplicação;
2. Fazer “Swipe” para a esquerda e para a direita na vista inicial para ver os destaques.

Resultado Expectável – Ao fazer “swipe” para a esquerda mostra o destaque seguinte, e pelo contrário ao fazer “swipe” para a direita mostra o destaque anterior.

Estado de erro – Não existe.

Caso de uso – Aceder ao BPstat Mobile

Componente – *Mobile*

Objetivo – Aceder ao BPstat Mobile através da aplicação móvel.

Utilizador – Qualquer utilizador.

Fluxo de ações

1. Aceder a aplicação;
2. Abrir o menu;
3. Selecionar o item “Estatísticas”;

Resultado Expectável - É apresentada uma nova *vista* com a aplicação BPstat mobile dentro da aplicação móvel do Banco de Portugal.

Estado de erro – Sem conexão a internet.

Esboços da interface

Com base na informação reunida na fase de análise, foram criados esboços para as duas aplicações. Os esboços permitem uma visualização gráfica da interface das aplicações, o que permite perceber o comportamento destas nos diferentes ambientes e estados, e ainda ajuda a corrigir aspetos estéticos que não estejam de acordo com o que o cliente pretende.

Mockup da vista de menus da Aplicação Híbrida num smartphone



Mockup da vista do conversor de moeda da Aplicação Híbrida num smartphone

Este mockup mostra a interface de um conversor de moeda. No topo, há um cabeçalho com o título "Câmbios". Abaixo do cabeçalho, há um bloco de texto informativo: "Os cálculos são efectuados com base nas taxas de referência diárias publicadas pelo Banco de Portugal e pelo Banco Central Europeu" e "Aviso Os valores indicados são meramente informativos e não vinculam o Banco de Portugal." Abaixo do texto, há um formulário com os seguintes elementos: um campo de entrada para "Valor", um campo de entrada para a data "dd/mm/aa", dois campos de seleção para "De" e "Para" (cada um com um ícone de seta para baixo), um campo de entrada para o "Resultado" e um botão "Submit" no fundo.

Mockup da vista do simulador de taxas de juro da Aplicação Híbrida num smartphone

Simuladores

Simulador de Taxas de Juro

O simulador é um instrumento de auxílio ao cliente bancário. Os dados solicitados pelo programa são inseridos sob o controlo exclusivo do utilizador. .

Mockup da vista de comunicados da Aplicação Híbrida num smartphone

Comunicados

 **Comunicado g**
Comunicado sobre determinado assunto. sd ewide wedew fefe deved

 **Comunicado a**
Comunicado sobre determinado assunto. sd ewide wedew fefe deved

 **Comunicado b**
Comunicado sobre determinado assunto. sd ewide wedew fefe deved

 **Comunicado x**
Comunicado sobre determinado assunto. sd ewide wedew fefe deved

 **Comunicado w**
Comunicado sobre determinado assunto. sd ewide wedew fefe deved

 **Comunicado z**

Mockup da vista de comunicados da Aplicação Híbrida num tablet

Menu	Comunicados
Início	 Comunicado g Comunicado sobre determinado assunto. sd euie wadev fefe deved
Comunicados	 Comunicado a Comunicado sobre determinado assunto. sd euie wadev fefe deved
Estatísticas	 Comunicado b Comunicado sobre determinado assunto. sd euie wadev fefe deved
Intervenções públicas	 Comunicado x Comunicado sobre determinado assunto. sd euie wadev fefe deved
Conversor de moeda	 Comunicado w Comunicado sobre determinado assunto. sd euie wadev fefe deved
Simulador taxa de juro	 Comunicado z

Protótipo de baixa fidelidade da página inicial do site num desktop



Mockup da página inicial do site num desktop



Protótipo de baixa fidelidade da página inicial do site num tablet



Mockup da página inicial do site num tablet



Protótipo de baixa fidelidade da página inicial do site num smartphone



Mockup da página inicial do site num smartphone



Anexo B

B.1. Desenho e modelação do sistema

Diagrama de módulos

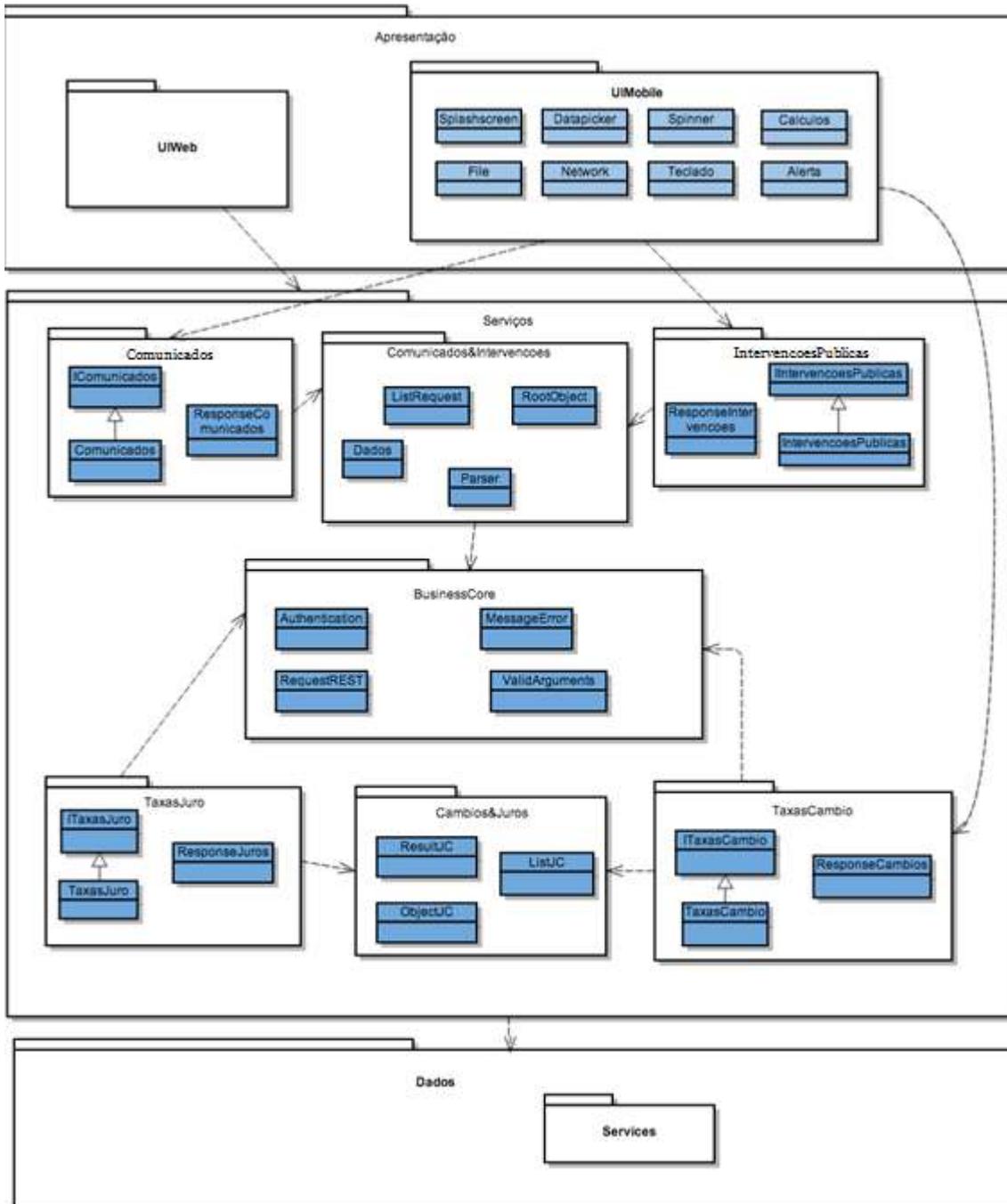


Diagrama de classes

Descrição das classes

IComunicados – Interface usada para definir o contrato do serviço Comunicados. Define o URL de cada operação e os parâmetros necessários para a sua invocação.

Comunicados – Classe que implementa a interface IComunicados.

ResponseComunicados – Classe que representa um objeto comunicado, indicando a data, o nome, tipo e conteúdo. É o objeto que será retornado aquando da invocação do serviço Comunicados.

Parser – Classe responsável por toda a lógica no tratamento de informação dos comunicados e intervenções públicas. Efetua o *parse* de informação entre os dados obtidos e os dados que são retornados.

RootObject – Classe que define a estrutura de dados que guarda os dados retornados aquando da invocação dos serviços SharePoint dos comunicados e intervenções públicas.

ListRequest – Objeto que pertence à estrutura de dados RootObject. Guarda a lista de objetos Dados.

Dados – Objeto pertence a estrutura de dados RootObject. Define um item das Comunicados e Intervenções públicas.

IIntervencoesPublicas – Interface usada para definir o contrato do serviço IntervencoesPublicas. Define o URL de cada operação e os parâmetros necessários para a sua invocação.

IntervencoesPublicas – Classe que implementa a interface IIntervencoesPublicas.

ResponseIntervencoes – Classe que representa um objeto intervenção, indicando a data, o nome, tipo e conteúdo. É o objeto que será retornado aquando da invocação do serviço IntervençõesPublicas.

Authentication – Classe responsável pela autenticação dos serviços com a camada de dados.

RequestREST – Classe responsável por criar os pedidos REST para invocar a camada de dados.

ValidArguments – Classe que valida os argumentos na invocação dos serviços. É responsável por verificar se os pedidos estão conforme os contratos.

Error – Define a estrutura de dados com as mensagens de erro, a retornar no caso da invocação incorreta de um serviço. Define o código do erro, e a descrição do mesmo.

ITaxasCambio – Interface usada para definir o contrato do serviço TaxasCambio. Define o URL de cada operação e os parâmetros necessários para a da invocação deste serviço.

TaxasCambio – Classe que implementa a interface ITaxasCambio.

ResponseCambios – Classe que representa um objeto câmbio. Define o nome da moeda, o valor, a data, o código da moeda e o país. É o objeto que será retornado aquando da invocação do serviço TaxasCambio.

ObjectJC – Classe que define a estrutura de dados que guarda os dados retornados aquando da invocação dos serviços SharePoint das taxas de juro e câmbio.

ListJC – Objeto que pertence à estrutura de dados ObjectJC. Guarda a lista de objetos ResultJC.

ResultJC – Objeto que pertence à estrutura de dados ObjectJC. Define um item das taxas de câmbio ou das taxas de juro.

ITaxasJuro – Interface usada para definir o contrato do serviço TaxasJuro. Define o URL de cada operação e os parâmetros necessários para a da invocação deste serviço.

TaxasJuro – Classe que implementa a interface ITaxasJuro.

ResponseJuros – Classe que representa um objeto TaxasJuro. Define o nome da Taxa de juro, o valor, a data e o gráfico associado. É o objeto que será retornado aquando da invocação do serviço TaxasJuro.

Diagramas de sequência

Diagrama de sequência "Ver destaques"

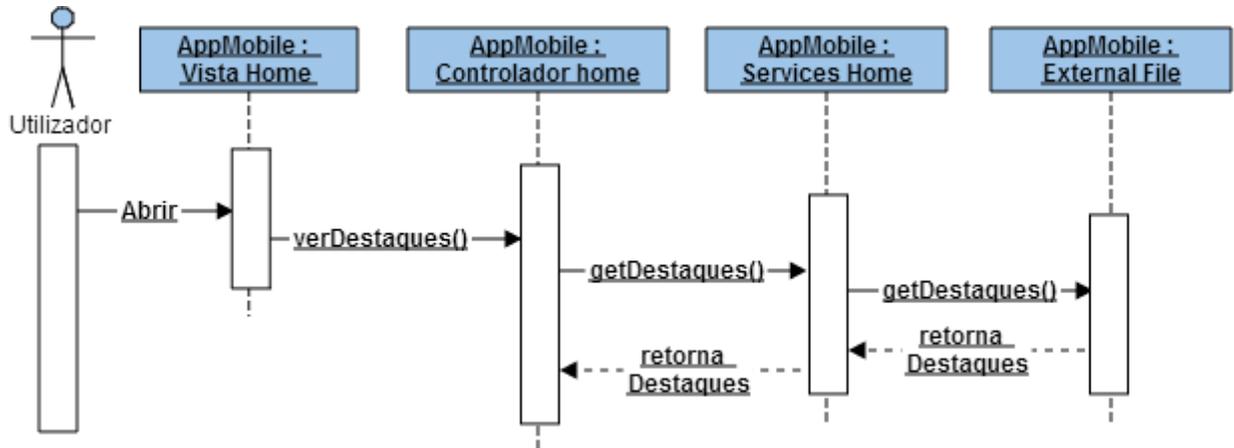


Diagrama de sequência "Abrir BPstat mobile"

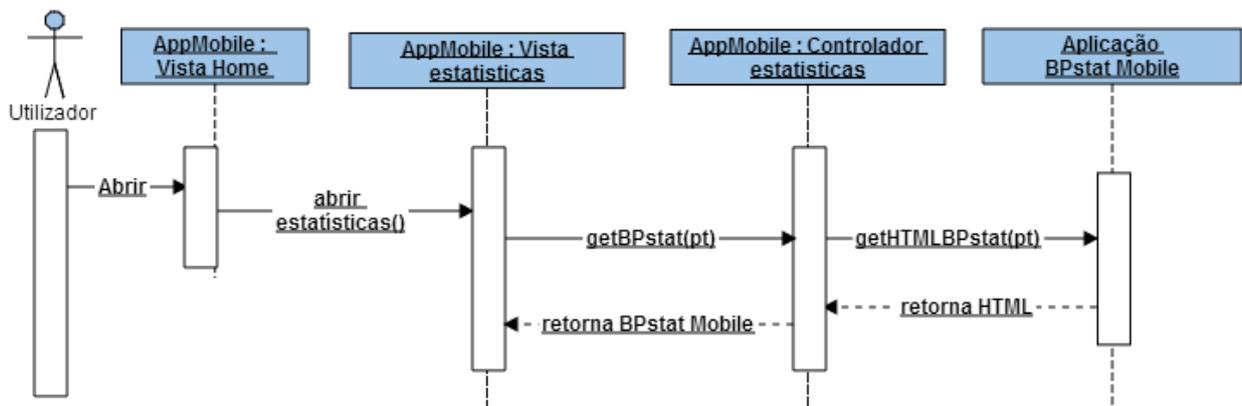


Diagrama de sequência "Ver intervenções públicas"

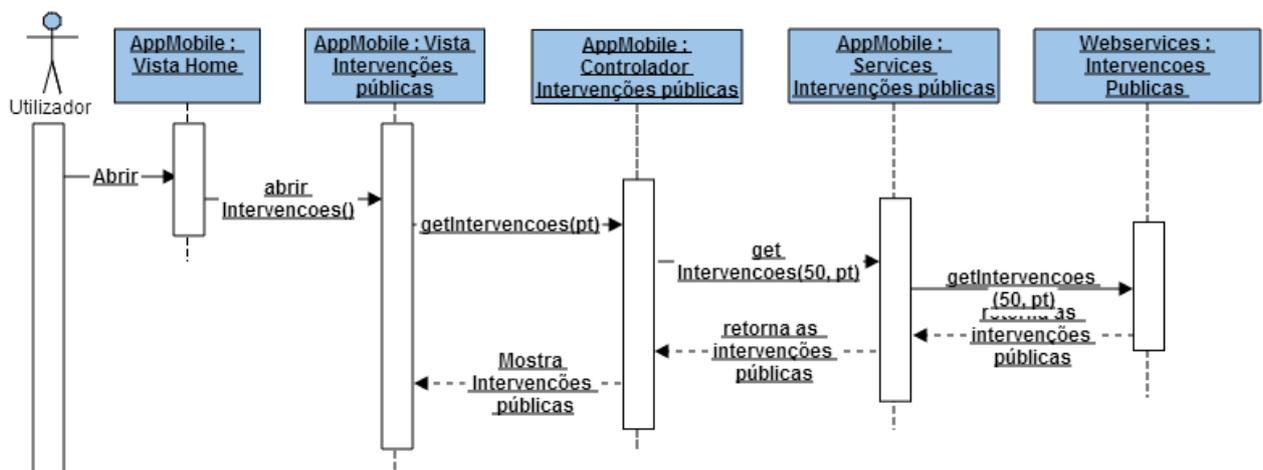


Diagrama de sequência "Visualizar comunicados no idioma inglês"

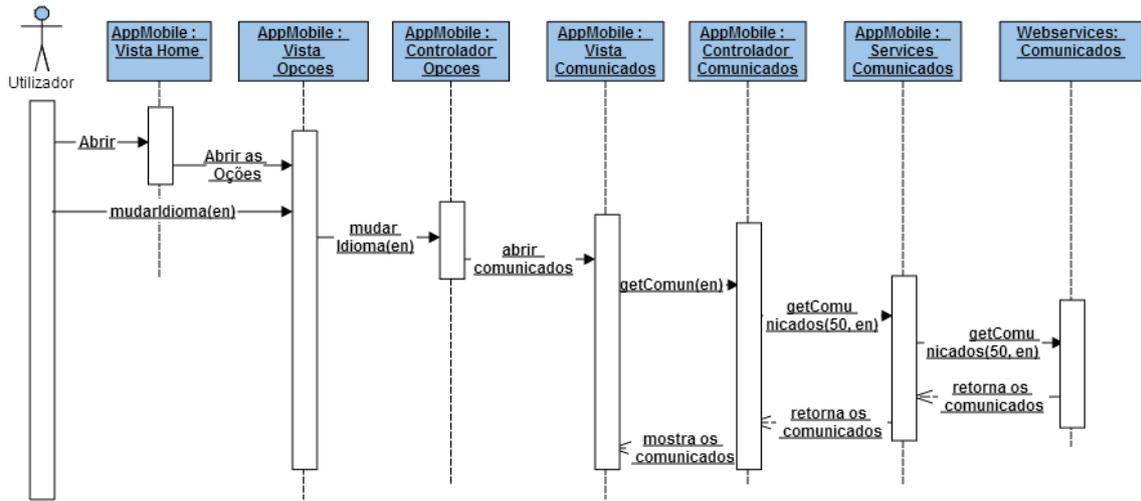


Diagrama de sequência "Fazer conversão de moeda"

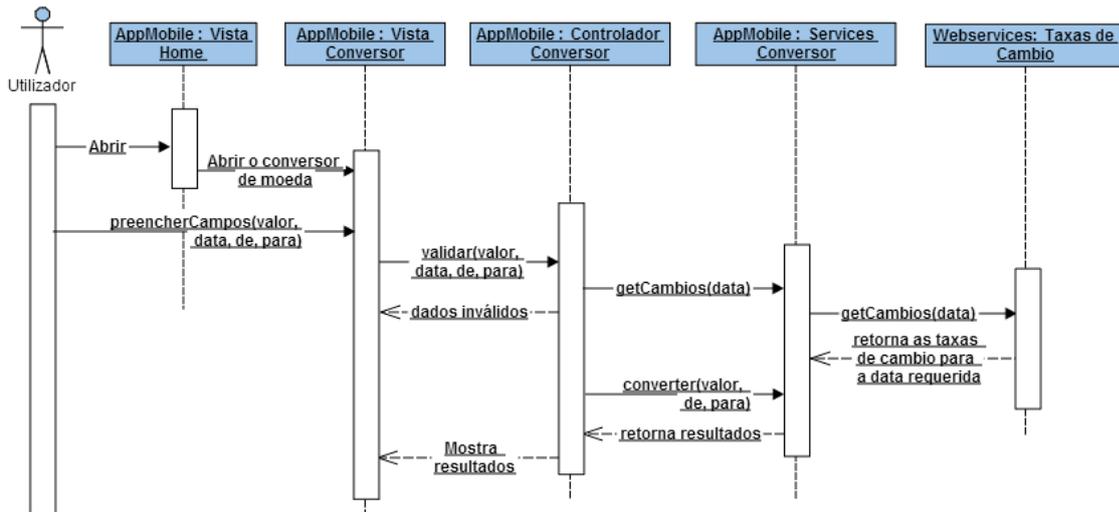
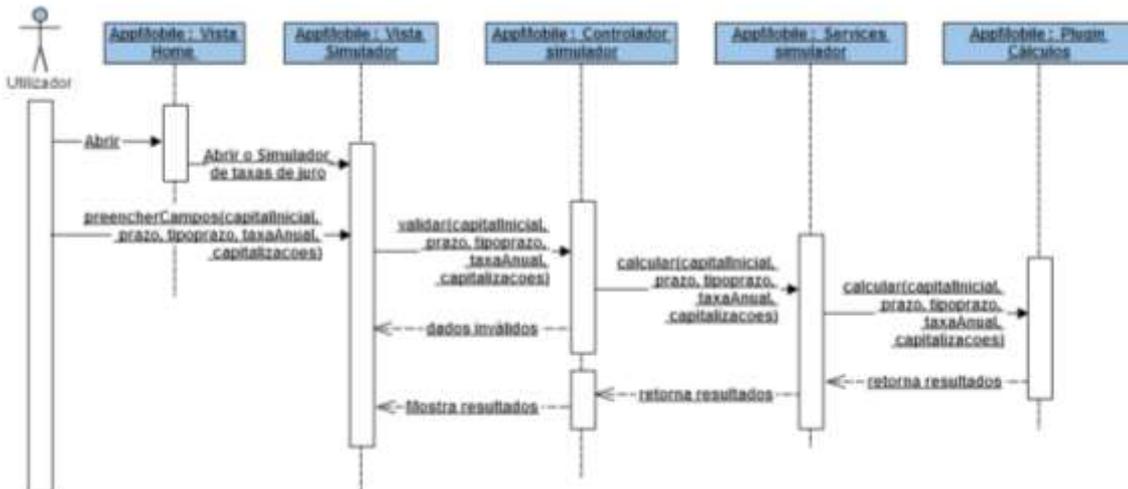


Diagrama de sequência "Fazer simulação de taxas de juro"



Anexo C

C.1. Testes de sistema

A tabela de testes abaixo deverá ser preenchida da seguinte forma:

1. Na coluna resultado identificar se a tarefa: Funciona(F); Funciona com algumas Restrições(FR); Não funciona(NF); Não foi testada(NT).
2. Na coluna Observações, caso o resultado seja (FR ou NF) apresentar a justificação do resultado.

Tarefa	Resultado Expectável	Resultado	Observações	Sugestões
1. Abrir a aplicação e visualizar a vista inicial.	1.1. Deverão ser exibidos os destaques.			
2. Abrir o menu lateral.	2.1. O menu lateral deverá ser posicionado a esquerda. 2.2. Deverá ser possível abrir/fechar o menu clicando no botão superior que identifica o menu, ou através do arrastamento do ecrã. 2.3. O menu apenas poderá ser aberto nas vistas de primeiro nível (todas as vistas acessíveis no próprio menu). 2.4. O menu deverá estar sempre visível (aberto) quando a aplicação for acedida via <i>Tablets</i> .			
3. Abrir o menu, e seleccionar o item comunicados.	3.1. Processo de carregamento. 3.1.1. Os comunicados são carregados apenas no primeiro acesso, ficando em sessão até o encerramento da aplicação. 3.1.2. Deverão ser listados os últimos 50 comunicados, publicados pelo Banco de Portugal. 3.1.3. Deverão estar ordenados de forma descendente. 3.1.4. Os comunicados no formato PDF deverão ser identificados como tal. 3.1.5. O clique em um item da lista de comunicados deverá permitir			

	<p>visualizar o mesmo em detalhe.</p> <p>3.1.6. A visualização de um comunicado deverá sempre utilizar a totalidade do ecrã quer em <i>Tablets</i> quer em <i>Smartphones</i>.</p> <p>3.2. Adicionar um comunicado à lista SharePoint.</p> <p>3.2.1. Se a aplicação estiver aberta e os comunicados já estiverem em sessão, o novo comunicado não será listado.</p> <p>3.2.2. Se a aplicação estiver aberta, mas os comunicados ainda não tenham sido visualizados, então o novo comunicado será listado.</p> <p>3.2.3. Se a aplicação for iniciada depois de adicionar o novo comunicado então este será listado.</p> <p>3.3. Remover um comunicado à lista SharePoint.</p> <p>3.3.1. Se a aplicação estiver aberta e os comunicados já estiverem em sessão, o comunicado removido continuará a ser listado, no entanto se este for um PDF, não será possível a visualização do mesmo.</p> <p>3.3.2. Se a aplicação estiver aberta, mas os comunicados ainda não tenham sido visualizados, então o comunicado removido deixará de ser listado.</p> <p>3.3.3. Se a aplicação for iniciada depois da remoção, o comunicado deixará de ser listado.</p>			
4. Abrir o menu e selecionar o item Estatísticas.	<p>4.1. Deverá exibir o portal BPstat mobile numa <i>webview</i> na própria aplicação.</p> <p>4.2. O funcionamento e layout do BPstat mobile deverá ser igual.</p>			
5. Abrir o menu e selecionar o item Intervenções Públicas.	<p>5.1. Processo de carregamento.</p> <p>5.1.1. As intervenções são carregadas apenas no primeiro acesso, ficando em sessão até o encerramento da aplicação.</p> <p>5.1.2. Deverão ser listadas às últimas 50 Intervenções, publicados pelo Banco de Portugal.</p> <p>5.1.3. Deverão estar ordenadas de forma descendente.</p>			

	<p>5.1.4. As Intervenções no formato PDF deverão ser identificadas como tal.</p> <p>5.1.5. O clique em um item da lista de Intervenções deverá permitir visualizar a mesma em detalhe.</p> <p>5.1.6. A visualização de uma Intervenção deverá sempre utilizar a totalidade do ecrã quer em <i>Tablets</i> quer em <i>Smartphones</i>.</p> <p>5.2. Adicionar uma Intervenção à lista SharePoint.</p> <p>5.2.1. Se a aplicação estiver aberta e as Intervenções já estiverem em sessão, a nova intervenção não será listada.</p> <p>5.2.2. Se a aplicação estiver aberta, mas as Intervenções ainda não tenham sido visualizadas, então o novo comunicado será listado.</p> <p>5.2.3. Se a aplicação for iniciada depois de adicionar o novo comunicado então este será listado.</p> <p>5.3. Remover uma Intervenção à lista SharePoint.</p> <p>5.3.1. Se a aplicação estiver aberta e as Intervenções já estiverem em sessão, a intervenção removida continuará a ser listada, no entanto, se for um PDF, não será possível a visualização do mesmo.</p> <p>5.3.2. Se a aplicação estiver aberta, mas as Intervenções ainda não tenham sido visualizadas, então a intervenção removida deixará de ser listada.</p> <p>5.3.3. Se a aplicação for iniciada depois da remoção, a Intervenção deixará de ser listada.</p>			
<p>6. Abrir o menu e selecionar o item Definições para mudar a linguagem da aplicação.</p>	<p>6.1. Deverá existir a possibilidade de escolher inglês ou português.</p> <p>6.2. A mudança de linguagem implica que os conteúdos exibidos estejam na linguagem escolhida.</p>			

<p>7. Abrir o menu e seleccionar o item Simulador de taxas de juro.</p>	<p>7.1. Todos os campos são de preenchimento obrigatório. 7.2. Os campos <i>capital inicial</i> e <i>taxa anual</i> aceitam valores decimais até 10 dígitos. 7.3. Os campos <i>prazo</i> e <i>capitalizações</i> apenas aceitam valores inteiros até 8 dígitos. 7.4. O campo <i>tipo de prazo</i> oferece três opções de escolha (Anos, Meses e Dias). 7.5. Todos os campos apenas aceitam valores positivos. 7.6. A simulação só será efetuada se todos os campos forem preenchidos corretamente. 7.7. Qualquer erro na introdução de um valor deverá ser sinalizado ao utilizador. 7.8. O resultado de uma simulação deverá ser exibido numa vista que ocupe a totalidade do ecrã, quer em <i>Tablets</i> ou <i>Smartphones</i>. 7.9. Os cálculos da simulação deverão ser iguais ao simulador do portal do cliente bancário</p>			
<p>8. Abrir o menu e seleccionar o item Conversor de moeda.</p>	<p>8.1. O campo <i>valor</i> aceita valores decimais até 12 dígitos. 8.2. O campo <i>valor</i> apenas aceita números positivos. 8.3. O campo <i>data</i> permite introduzir qualquer data. 8.4. Os campos <i>De</i> e <i>Para</i> deverão exibir todas as moedas atualmente em vigor. 8.5. Qualquer erro na introdução de um valor deverá ser sinalizado ao utilizador. 8.6. O resultado de uma simulação deverá ser exibido numa vista que ocupe a totalidade do ecrã, quer em <i>Tablets</i> ou <i>Smartphones</i>. 8.7. A mudança de data deverá atualizar o valor dos câmbios para o dia requerido. 8.8. Os cálculos da conversão deverão ser iguais ao conversor do <i>site</i> institucional</p>			

Anexo D

D.1. Testes de carga

Teste 1 - BMA sem cache implementada

Condições de execução

Utilizadores: 50

Tempo de Execução: 10 minutos

Tempo Warm-up: 30 segundos

Think time: Activo

Load Test Summary

Test Run Information

Load test name	Mobile_light
Description	
Start time	27-06-2014 14:28:01
End time	27-06-2014 14:38:01
Warm-up duration	00:00:30
Duration	00:10:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Overall Results

Max User Load	50
Tests/Sec	4,69
Tests Failed	30
Avg. Test Time (sec)	9,95
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	14,2
Avg. Page Time (sec)	0,31
Requests/Sec	14,2
Requests Failed	30
Requests Cached Percentage	0
Avg. Response Time (sec)	0,31
Avg. Content Length (bytes)	128.772

Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
http://services.bportugal.pt/bma/comunic...	0,042
http://services.bportugal.pt/bma/comunic...	0,031
http://services.bportugal.pt/bma/interve...	0,023
http://services.bportugal.pt/bma/interve...	0,014
http://services.bportugal.pt/bma/taxasca...	0,0070

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
Ingles	9,60
Portuques	9,07

Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
Ingles	Scenario1	674	13 (1,93)	10,3
Portuques	Scenario1	2.138	17 (0,80)	9,86

Page Results

URL (Link to More Details)	Scenario	Test	Avg. Page Time (sec)	Count
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/en-us/50	Scenario1	Ingles	0,69	682
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/pt-pt/50	Scenario1	Portugues	0,56	2.174
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/e...	Scenario1	Ingles	0,51	680
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/p...	Scenario1	Portugues	0,26	2.168
http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/pt-pt/Date/2014-05-...	Scenario1	Portugues	0,0044	2.165
http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/en-us/Date/2014-05-...	Scenario1	Ingles	0,0044	678

Transaction Results

Name	Scenario	Test	Response Time (sec)	Elapsed Time (sec)	Count
------	----------	------	---------------------	--------------------	-------

System Under Test Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
--------------	------------------	--

Controller and Agents Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
TESTESDECARGA2	5,70	6.422

Errors

Type	Subtype	Count	Last Message
Http Error	400 - BadRequest	20	400 - BadRequest
Http Error	403 - Forbidden	10	403 - Forbidden

Load Test Details

Pages														
Page	Scenario	Test	Network	Total	Avg.	Min	Median	90%	95%	99%	Max	Std Dev	Goal	% Meeting Goal
2014-05-27	Scenario1	Portugues	LAN	2.165	0,0044	0,0030	0,0040	0,0050	0,0070	0,017	0,035	0,0024	5,00	100
2014-05-27	Scenario1	Ingles	LAN	678	0,0044	0,0030	0,0040	0,0050	0,0070	0,013	0,063	0,0031	5,00	100
50	Scenario1	Portugues	LAN	2.168	0,26	0,0040	0,0050	0,0090	0,014	0,062	41,7	2,84	10,0	99,1
50	Scenario1	Ingles	LAN	680	0,51	0,0050	0,0070	0,019	0,023	34,4	39,2	4,18	10,0	98,5
50	Scenario1	Portugues	LAN	2.174	0,56	0,0050	0,0070	0,019	0,031	26,8	36,3	3,75	10,0	97,9
50	Scenario1	Ingles	LAN	682	0,69	0,0050	0,0060	0,018	0,042	25,8	29,0	3,83	10,0	97,2

Errors			
Type	Subtype	Count	Last Message
Total		30	
Http Error	400 - BadRequest	20	400 - BadRequest
Http Error	403 - Forbidden	10	403 - Forbidden

Requests						
Request	Scenario	Test	Total	Failed	Response Time	Content Length
2014-05-27	Scenario1	Ingles	678	0	0,0044	7.698
2014-05-27	Scenario1	Portugues	2.165	0	0,0044	7.777 (...)/TaxasCambio/en-us/Date/2014-
50	Scenario1	Ingles	682	13,0	0,69	173.722
50	Scenario1	Portugues	2.174	14,0	0,56	246.525 05-27
50	Scenario1	Ingles	680	0	0,51	277.456
50	Scenario1	Portugues	2.168	3,00	0,26	108.611 (...)/TaxasCambio/pt-pt/Date/2014-

Tests							
Test	Scenario	Total	Passed	Failed	Tests/Sec	Test Time	95% Test Time
Ingles	Scenario1	674	661	13,0	1,12	10,3	9,6
Portugues	Scenario1	2.138	2.121	17,0	3,56	9,86	9,07

Teste 2 - BMA com cache implementada

Condições de execução

Utilizadores: 50

Tempo de Execução: 20 minutos

Tempo Warm-up: 1 minuto

Think time: Activo

Load Test Summary

Test Run Information

Load test name	Mobile_light
Description	
Start time	01-07-2014 11:21:34
End time	01-07-2014 11:41:34
Warm-up duration	00:01:00
Duration	00:20:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Overall Results

Max User Load	50
Tests/Sec	5,33
Tests Failed	0
Avg. Test Time (sec)	9,20
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	16,0

Avg. Page Time (sec)	0,064
Requests/Sec	16,0
Requests Failed	0
Requests Cached Percentage	0
Avg. Response Time (sec)	0,064
Avg. Content Length (bytes)	122.860

Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
http://services.bportugal.pt/bma/taxasca...	0,032
http://services.bportugal.pt/bma/comunic...	0,023
http://services.bportugal.pt/bma/comunic...	0,023
http://services.bportugal.pt/bma/interve...	0,021
http://services.bportugal.pt/bma/interve...	0,016

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
Portugues	9,20
Ingles	9,06

Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
Portugues	Scenario1	4.701	0 (0)	9,26
Ingles	Scenario1	1.692	0 (0)	9,03

Page Results

URL (Link to More Details)	Scenario	Test	Avg. Page Time (sec)	Count
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/p...	Scenario1	Portugues	0,22	4.719
http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/pt-pt/Date/2014-05-...	Scenario1	Portugues	0,016	4.705
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/pt-pt/50	Scenario1	Portugues	0,011	4.732
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/e...	Scenario1	Ingles	0,0097	1.698
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/en-us/50	Scenario1	Ingles	0,0096	1.701

http://services.bportugal.pt/bma/taxascambio.svc/TexasCambio/en-us/Date/2014-05-...	Scenario1	Ingles	0,0046	1.692
---	-----------	--------	--------	-------

Transaction Results

Name	Scenario	Test	Response Time (sec)	Elapsed Time (sec)	Count
------	----------	------	---------------------	--------------------	-------

System Under Test Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
--------------	------------------	--

Controller and Agents Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
TESTESDECARGA2	6,89	4.190

Errors

Type	Subtype	Count	Last Message
------	---------	-------	--------------

Load Test Details

Pages														
Page	Scenario	Test	Network	Total	Avg.	Min	Median	90%	95%	99%	Max	Std Dev	Goal	% Meeting Goal
2014-05-27	Scenario1	Portugues	LAN	4.705	0,016	0,0030	0,0040	0,0070	0,032	0,26	0,65	0,052	5,00	100
2014-05-27	Scenario1	Ingles	LAN	1.692	0,0046	0,0030	0,0040	0,0050	0,0070	0,017	0,20	0,0060	5,00	100
50	Scenario1	Portugues	LAN	4.732	0,011	0,0050	0,0070	0,018	0,023	0,042	3,47	0,051	10,0	100
50	Scenario1	Ingles	LAN	1.698	0,0097	0,0050	0,0070	0,017	0,021	0,034	0,051	0,0056	10,0	100
50	Scenario1	Ingles	LAN	1.701	0,0096	0,0040	0,0070	0,017	0,023	0,041	0,096	0,0082	10,0	100
50	Scenario1	Portugues	LAN	4.719	0,22	0,0040	0,0050	0,011	0,016	0,035	52,9	3,13	10,0	99,5

Errors			
Type	Subtype	Count	Last Message
Total		0	

Requests						
Request	Scenario	Test	Total	Failed	Response Time	Content Length
2014-05-27	Scenario1	Portugues	4.705	0	0,016	7.777
2014-05-27	Scenario1	Ingles	1.692	0	0,0046	7.698
50	Scenario1	Portugues	4.732	0	0,011	217.569
50	Scenario1	Portugues	4.719	0	0,22	108.759
50	Scenario1	Ingles	1.701	0	0,0096	177.064
50	Scenario1	Ingles	1.698	0	0,0097	277.456

Tests							
Test	Scenario	Total	Passed	Failed	Tests/Sec	Test Time	95% Test Time
Ingles	Scenario1	1.692	1.692	0	1,41	9,03	9,06
Portugues	Scenario1	4.701	4.701	0	3,92	9,26	9,20

Teste 3 - BMA com cache implementada

Condições de execução

Utilizadores: 100

Tempo de Execução: 20 minutos

Tempo Warm-up: 1 minuto

Think time: Activo

Load Test Summary

Test Run Information

Load test name	Mobile_light
Description	
Start time	30-06-2014 17:11:53
End time	30-06-2014 17:31:53
Warm-up duration	00:01:00
Duration	00:20:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Overall Results

Max User Load	100
Tests/Sec	10,6
Tests Failed	0
Avg. Test Time (sec)	9,04
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	31,9
Avg. Page Time (sec)	0,010

Requests/Sec	31,9
Requests Failed	0
Requests Cached Percentage	0
Avg. Response Time (sec)	0,010
Avg. Content Length (bytes)	122.264

Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
http://services.bportugal.pt/bma/comunic...	0,034
http://services.bportugal.pt/bma/comunic...	0,032
http://services.bportugal.pt/bma/interve...	0,028
http://services.bportugal.pt/bma/taxasca...	0,024
http://services.bportugal.pt/bma/interve...	0,021

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
Ingles	9,13
Portuques	9,09

Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
Ingles	Scenario1	3.190	0 (0)	9,05
Portuques	Scenario1	9.526	0 (0)	9,04

Page Results

URL (Link to More Details)	Scenario	Test	Avg. Page Time (sec)	Count
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/pt-pt/50	Scenario1	Portuques	0,014	9.595
http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/en-us/Date/2014-05-...	Scenario1	Ingles	0,014	3.197
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/en-us/50	Scenario1	Ingles	0,013	3.211
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/e...	Scenario1	Ingles	0,012	3.205
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/p...	Scenario1	Portuques	0,0093	9.559
http://services.bportugal.pt/bma/taxascambio.svc/TaxasCambio/pt-pt/Date/2014-05-...	Scenario1	Portuques	0,0054	9.542

Transaction Results

Name	Scenario	Test	Response Time (sec)	Elapsed Time (sec)	Count
------	----------	------	---------------------	--------------------	-------

System Under Test Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
--------------	------------------	--

Controller and Agents Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
TESTESDECARGA2	10,9	5.432

Errors

Type	Subtype	Count	Last Message
------	---------	-------	--------------

Load Test Details

Pages														
Page	Scenario	Test	Network	Total	Avg.	Min	Median	90%	95%	99%	Max	Std Dev	Goal	% Meeting Goal
2014-05-27	Scenario1	Portugues	LAN	9.542	0,0054	0,0020	0,0040	0,0080	0,012	0,031	0,24	0,0062	5,00	100
2014-05-27	Scenario1	Ingles	LAN	3.197	0,014	0,0030	0,0040	0,0090	0,024	0,25	0,82	0,050	5,00	100
50	Scenario1	Portugues	LAN	9.559	0,0093	0,0040	0,0050	0,014	0,021	0,040	10,3	0,11	10,0	100,0
50	Scenario1	Portugues	LAN	9.595	0,014	0,0050	0,0080	0,026	0,034	0,056	5,02	0,066	10,0	100
50	Scenario1	Ingles	LAN	3.205	0,012	0,0050	0,0080	0,021	0,028	0,048	0,52	0,013	10,0	100
50	Scenario1	Ingles	LAN	3.211	0,013	0,0040	0,0080	0,024	0,032	0,052	3,55	0,064	10,0	100

Errors			
Type	Subtype	Count	Last Message
Total		0	

Requests						
Request	Scenario	Test	Total	Failed	Response Time	Content Length
2014-05-27	Scenario1	Ingles	3.197	0	0,014	7.698
2014-05-27	Scenario1	Portugues	9.542	0	0,0054	7.777
50	Scenario1	Ingles	3.211	0	0,013	177.064
50	Scenario1	Ingles	3.205	0	0,012	277.456
50	Scenario1	Portugues	9.595	0	0,014	217.569
50	Scenario1	Portugues	9.559	0	0,0093	108.759

Tests							
Test	Scenario	Total	Passed	Failed	Tests/Sec	Test Time	95% Test Time
Ingles	Scenario1	3.190	3.190	0	2,66	9,05	9,13
Portugues	Scenario1	9.526	9.526	0	7,94	9,04	9,09

Teste 4 - BMA com cache implementada

Condições de execução

Utilizadores: 200

Tempo de Execução: 20 minutos

Tempo Warm-up: 1 minuto

Think time: Activo

Load Test Summary

Test Run Information

Load test name	Mobile_light
Description	
Start time	01-07-2014 10:42:31
End time	01-07-2014 11:02:31
Warm-up duration	00:01:00
Duration	00:20:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Overall Results

Max User Load	200
Tests/Sec	20,3
Tests Failed	0
Avg. Test Time (sec)	9,05
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	61,1
Avg. Page Time (sec)	0,014

Requests/Sec	61,1
Requests Failed	0
Requests Cached Percentage	0
Avg. Response Time (sec)	0,014
Avg. Content Length (bytes)	122.164

Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
http://services.bportugal.pt/bma/comunic...	0,040
http://services.bportugal.pt/bma/taxasca...	0,037
http://services.bportugal.pt/bma/comunic...	0,034
http://services.bportugal.pt/bma/interve...	0,033
http://services.bportugal.pt/bma/interve...	0,024

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
Portugues	9,14
Ingles	9,10

Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
Portugues	Scenario1	18.258	0 (0)	9,05
Ingles	Scenario1	6.046	0 (0)	9,05

Page Results

URL (Link to More Details)	Scenario	Test	Avg. Page Time (sec)	Count
http://services.bportugal.pt/bma/taxascambio.svc/TexasCambio/pt-pt/Date/2014-05-...	Scenario1	Portugues	0,017	18.300
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/pt-pt/50	Scenario1	Portugues	0,017	18.406
http://services.bportugal.pt/bma/comunicados.svc/Comunicados/en-us/50	Scenario1	Ingles	0,013	6.085
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/e...	Scenario1	Ingles	0,012	6.071
http://services.bportugal.pt/bma/taxascambio.svc/TexasCambio/en-us/Date/2014-05-...	Scenario1	Ingles	0,011	6.058
http://services.bportugal.pt/bma/intervencoespublicas.svc/IntervencoesPublicas/p...	Scenario1	Portugues	0,0085	18.364

Transaction Results

Name	Scenario	Test	Response Time (sec)	Elapsed Time (sec)	Count
------	----------	------	---------------------	--------------------	-------

System Under Test Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
--------------	------------------	--

Controller and Agents Resources

Machine Name	% Processor Time	Available Memory at Test Completion (Mb)
TESTESDECARGA2	18,0	4.108

Errors

Type	Subtype	Count	Last Message
------	---------	-------	--------------

Load Test Details

Pages														
Page	Scenario	Test	Network	Total	Avg.	Min	Median	90%	95%	99%	Max	Std Dev	Goal	% Meeting Goal
2014-05-27	Scenario1	Portugues	LAN	18.300	0,017	0,0020	0,0040	0,013	0,037	0,31	1,09	0,064	5,00	100
2014-05-27	Scenario1	Ingles	LAN	6.058	0,011	0,0020	0,0040	0,0100	0,020	0,23	0,84	0,040	5,00	100
50	Scenario1	Portugues	LAN	18.364	0,0085	0,0040	0,0060	0,015	0,024	0,044	0,51	0,0093	10,0	100
50	Scenario1	Portugues	LAN	18.406	0,017	0,0040	0,012	0,029	0,040	0,068	5,58	0,071	10,0	100
50	Scenario1	Ingles	LAN	6.071	0,012	0,0050	0,0080	0,022	0,033	0,051	0,42	0,015	10,0	100
50	Scenario1	Ingles	LAN	6.085	0,013	0,0040	0,0080	0,023	0,034	0,055	0,54	0,019	10,0	100

Errors			
Type	Subtype	Count	Last Message
Total		0	

Requests						
Request	Scenario	Test	Total	Failed	Response Time	Content Length
2014-05-27	Scenario1	Portugues	18.300	0	0,017	7.777
2014-05-27	Scenario1	Ingles	6.058	0	0,011	7.698
50	Scenario1	Portugues	18.406	0	0,017	217.569
50	Scenario1	Portugues	18.364	0	0,0085	108.759
50	Scenario1	Ingles	6.085	0	0,013	177.064
50	Scenario1	Ingles	6.071	0	0,012	277.456

Tests							
Test	Scenario	Total	Passed	Failed	Tests/Sec	Test Time	95% Test Time
Ingles	Scenario1	6.046	6.046	0	5,04	9,05	9,10
Portugues	Scenario1	18.258	18.258	0	15,2	9,05	9,14

Anexo E

E.1. Testes de usabilidade

1 - Dados Pessoais																
Nome: _____																
Idade: _____																
Usa regularmente dispositivos móveis: Sim <input type="checkbox"/> Não <input type="checkbox"/>																
Se sim, prefere aceder as aplicações móveis ou às páginas <i>web</i> ? _____																
Porquê? _____																
Qual o sistema operativo com qual está mais familiarizado? iOS <input type="checkbox"/> Android <input type="checkbox"/> WPhone <input type="checkbox"/>																
Qual a sua familiaridade com os conceitos Taxas de câmbio e Taxas de juro?																
<table border="1"><thead><tr><th colspan="2">Pouco</th><th colspan="3">Muito</th></tr><tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>		Pouco		Muito			1	2	3	4	5	<input type="checkbox"/>				
Pouco		Muito														
1	2	3	4	5												
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
E com os Comunicados e intervenções Públicas disponibilizados pelo Banco de Portugal?																
<table border="1"><thead><tr><th colspan="2">Pouco</th><th colspan="3">Muito</th></tr><tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>		Pouco		Muito			1	2	3	4	5	<input type="checkbox"/>				
Pouco		Muito														
1	2	3	4	5												
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												

2 - Dados para o teste	
Smarphone <input type="checkbox"/> Tablet <input type="checkbox"/>	
Sistema operativo: _____	

3 - Tarefas de teste																
Efetuar pelo menos 3 simulações de taxas de juro diferentes.																
Concluiu a tarefa: Sim <input type="checkbox"/> Não <input type="checkbox"/>																
Classifique-a relativamente ao grau de dificuldade: <table border="1"><thead><tr><th colspan="2">Fácil</th><th colspan="3">Difícil</th></tr><tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>		Fácil		Difícil			1	2	3	4	5	<input type="checkbox"/>				
Fácil		Difícil														
1	2	3	4	5												
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
Cometeu algum erro na realização da mesma: Sim <input type="checkbox"/> Não <input type="checkbox"/>																
Se sim, qual e porquê? _____																

Acha que a tarefa pode ser melhorada: Sim <input type="checkbox"/> Não <input type="checkbox"/>																
Se sim, como? _____																
Visualizar um comunicado que não seja Pdf.																
Concluiu a tarefa: Sim <input type="checkbox"/> Não <input type="checkbox"/>																
Classifique-a relativamente ao grau de dificuldade: <table border="1"><thead><tr><th colspan="2">Fácil</th><th colspan="3">Difícil</th></tr><tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>		Fácil		Difícil			1	2	3	4	5	<input type="checkbox"/>				
Fácil		Difícil														
1	2	3	4	5												
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
Cometeu algum erro na realização da mesma: Sim <input type="checkbox"/> Não <input type="checkbox"/>																

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

Se sim, como? _____

Visualizar um comunicado que esteja no formato Pdf.

Concluiu a tarefa: Sim Não

Classifique-a relativamente ao grau de dificuldade:

Fácil					Difícil
1	2	3	4	5	

Cometeu algum erro na realização da mesma: Sim Não

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

Se sim, como? _____

Visualizar os 3 destaques na vista inicial.

Concluiu a tarefa: Sim Não

Classifique-a relativamente ao grau de dificuldade:

Fácil					Difícil
1	2	3	4	5	

Cometeu algum erro na realização da mesma: Sim Não

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

Se sim, como? _____

Visualizar uma intervenção pública que esteja no formato Pdf.

Concluiu a tarefa: Sim Não

Classifique-a relativamente ao grau de dificuldade:

Fácil					Difícil
1	2	3	4	5	

Cometeu algum erro na realização da mesma: Sim Não

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

Se sim, como? _____

Visualizar uma intervenção pública no formato Pdf com a aplicação na linguagem em Inglês.

Concluiu a tarefa: Sim Não

Classifique-a relativamente ao grau de dificuldade:

Fácil					Difícil
1	2	3	4	5	

Cometeu algum erro na realização da mesma: Sim Não

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

Se sim, como? _____

Utilizar o conversor de moeda para efetuar o câmbio em pelo menos três moedas diferentes.

Concluiu a tarefa: Sim Não

Classifique-a relativamente ao grau de dificuldade:

Fácil					Difícil
1	2	3	4	5	

Cometeu algum erro na realização da mesma: Sim Não

Se sim, qual e porquê? _____

Acha que a tarefa pode ser melhorada: Sim Não

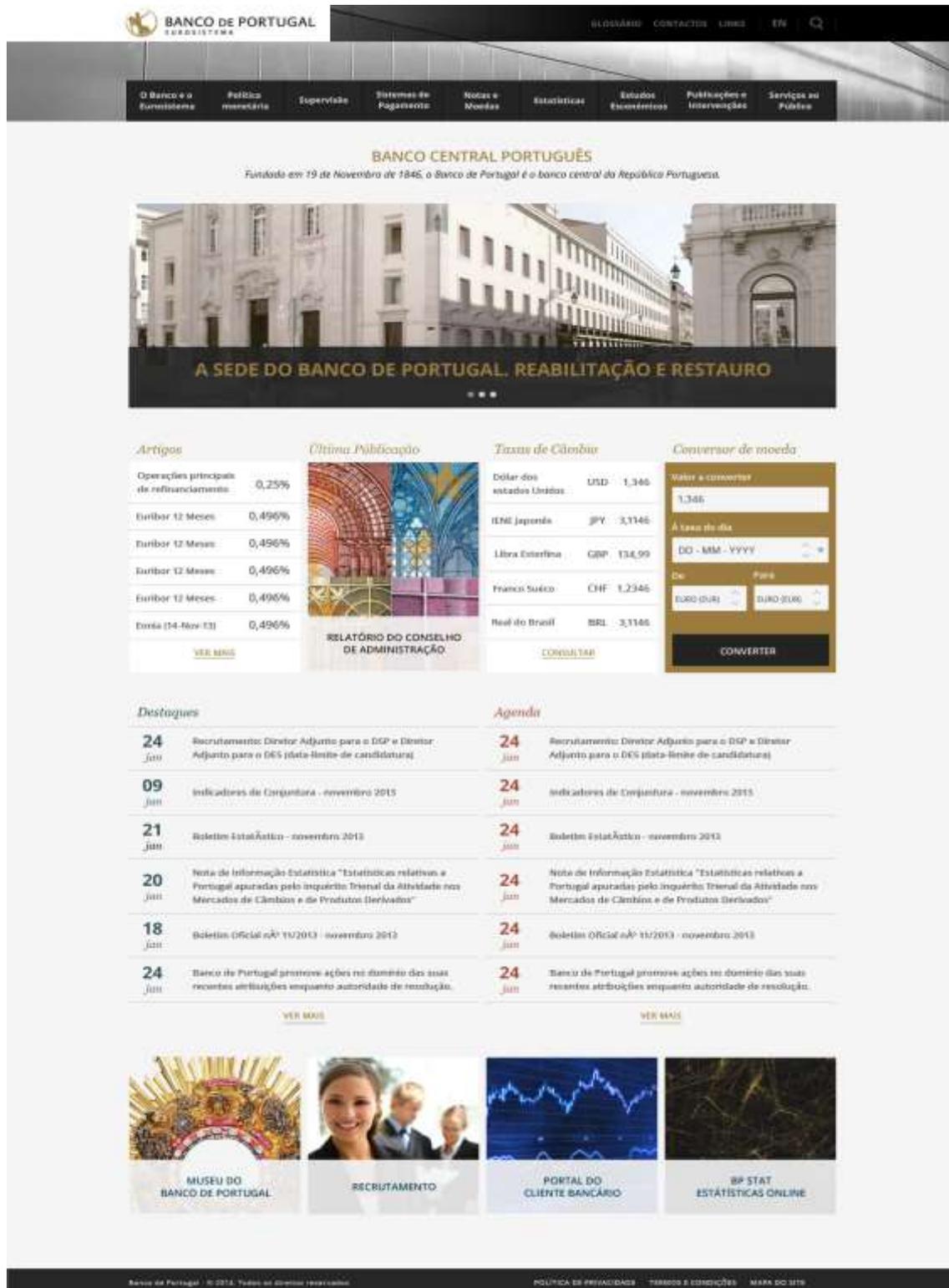
Se sim, como? _____

4 - Avaliação de usabilidade					
	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
1. Sinto que vou usar esta aplicação com frequência.					
2. Acho a aplicação desnecessariamente complexa.					
3. Achei a aplicação fácil de usar.					
4. Acho que vou precisar de ajuda para poder usar esta aplicação.					
5. As funcionalidades da aplicação estão bem integradas.					
6. Achei a aplicação muito inconsistente.					
7. Penso que a maioria das pessoas aprenderiam a usar a aplicação rapidamente.					
8. Achei a aplicação muito complicada de usar.					
9. Senti-me confiante a usar a aplicação.					
10. Preciso aprender muitas coisas para usar bem esta aplicação.					
11. Foi fácil encontrar a informação que procurava.					

Anexo F

F.1. Layouts WIZ

Mockup de alto nível da página inicial do site num desktop



Mockup de alto nível da vista inicial da aplicação híbrida num smartphone



Mockup de alto nível da vista de menus da aplicação híbrida num smartphone



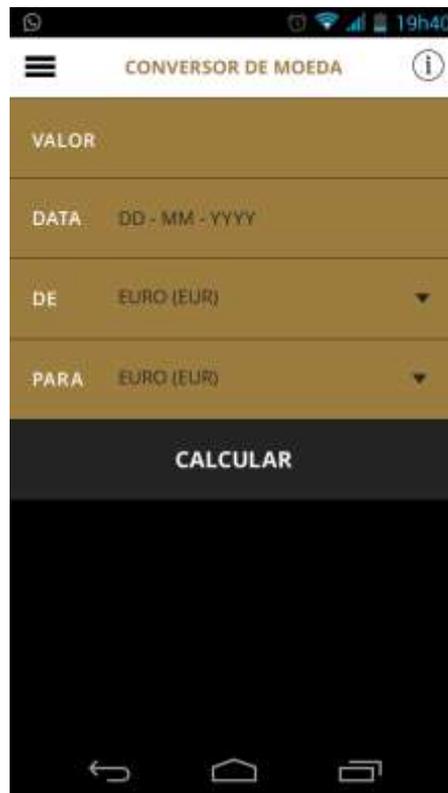
Mockup de alto nível da vista de comunicados da aplicação híbrida num smartphone



Mockup de alto nível da vista de um comunicado da aplicação híbrida num smartphone



Mockup de alto nível da vista do conversor de moeda da aplicação híbrida num smartphone



Mockup de alto nível da vista do simulador de taxas de juro da aplicação híbrida num smartphone



Mockup de alto nível da vista de uma intervenção pública da aplicação híbrida num smartphone



Mockup de alto nível da vista dos resultados de uma simulação na aplicação híbrida num smartphone



Anexo G

G.1. Método AuthenticateSPS com o modelo de autenticação FBA da classe Authenticate

```
public static string AuthenticateSPS()
{
    CookieContainer CookieJar = new CookieContainer();
    string siteroot =
    System.Configuration.ConfigurationManager.AppSettings["siteroot"];
    var url = siteroot + "/_vti_bin/authentication.asmx";
    Uri authServiceUri = new Uri(url);
    HttpRequest spAuthReq = HttpRequest.Create(authServiceUri) as
    HttpRequest;
    spAuthReq.CookieContainer = CookieJar;
    spAuthReq.Headers["SOAPAction"] =
    "http://schemas.microsoft.com/sharepoint/soap/Login";
    spAuthReq.ContentType = "text/xml; charset=utf-8";
    spAuthReq.Method = "POST";
    string envelope =
    "<?xml version='1.0\' encoding='utf-8\'?>" +
    "<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance\'"+
    "xmlns:xsd='http://www.w3.org/2001/XMLSchema\'"+
    "xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'><soap:Body>"+
    "<Login xmlns='http://schemas.microsoft.com/sharepoint/soap/'>"+
    "<username>{0}</username><password>{1}</password>"+
    "</Login></soap:Body></soap:Envelope>";
    envelope = string.Format(envelope, userName, password);
    StreamWriter streamWriter = new
    StreamWriter(spAuthReq.GetRequestStream());
    streamWriter.Write(envelope);
    streamWriter.Close();
    WebResponse response = spAuthReq.GetResponse();
    var returnValue = response.Headers[HttpResponseHeader.SetCookie];
    var posInit = returnValue.IndexOf("FedAuth=");
    var posEnd = returnValue.IndexOf(";", posInit);
    returnValue = returnValue.Substring(posInit + 8, posEnd - posInit - 8);
    response.Close();
    return returnValue;
}
```

G.2. Excerto do código usado na criação de um pedido dos *web services* com a camada de dados

```
if (cookieValue == null || authcookie == null || authcookie.Expired)
{
    cc = new CookieContainer();
    cookieValue = Authenticate.AuthenticateSPS();
    authcookie = new Cookie("FedAuth", cookieValue);
    authcookie.Expires = DateTime.Now.AddHours(1);
    authcookie.Path = "/";
    authcookie.HttpOnly = true;
    authcookie.Domain = request.RequestUri.Host;
    cc.Add(authcookie);
}
```

G.3. Código do objeto `CreateRequest` responsável pela criação de pedidos entre as camadas de serviços e dados

```
public HttpRequest CreateRequest(Uri UriRequest, ref CookieContainer cc, ref
Cookie authcookie, ref string cookieValue)
{
    HttpRequest request =
    (HttpRequest)HttpRequest.Create(UriRequest);
    request.Accept = "application/json";
    request.ContentType = "text/javascript";
    request.Method = RequestMethod.Http.Get;
    request.KeepAlive = true;
    char[] param = new Char[] { ';' };
    if (cookieValue == null || authcookie == null || authcookie.Expired)
    {
        cc = new CookieContainer();
        cookieValue = Authenticate.AuthenticateSPS();
        authcookie = new Cookie("FedAuth", cookieValue);
        authcookie.Expires = DateTime.Now.AddHours(1);
        authcookie.Path = "/";
        authcookie.HttpOnly = true;
        authcookie.Domain = request.RequestUri.Host;
        cc.Add(authcookie);
    }
    request.CookieContainer = cc;
    request.UseDefaultCredentials = true;
    NetworkCredential nc = new NetworkCredential("", "", "");
    CredentialCache ccCreds = new CredentialCache();
    ccCreds.Add(new Uri(siteroot), "NTLM", nc);
    request.Credentials = ccCreds;
    return request;
}
```

G.4. Código da estrutura de dados que permite guardar a resposta a um pedido do serviço TaxasJuro

```
using System.Runtime.Serialization;
using System.Runtime.Serialization.Json;
namespace BDP_RESTFul_Service
{
    [DataContract]
    public class ListJC
    {
        public List<ResultJuros> results { get; set; }
    }
    [DataContract]
    public class ObjectJC
    {
        public ListJC d { get; set; }
    }
    [DataContract]
    public class ResultJC
    {
        public DateTime Data { get; set; }
        public string DenominaçãoDaTaxaDeJuro { get; set; }
        public double ValorDaTaxa { get; set; }
        public string Link { get; set; }
        public string InterestRate { get; set; }
        public double InterestRateValue { get; set; }
        public double CotacaoFaceAoEuro { get; set; }
        public double CodigoDaMoeda { get; set; }
        public string DenominacaoMoeda { get; set; }
        public string Pais { get; set; }
    }
}
}
```

G.5. Método CreateResponse da classe TaxasJuro

```
public List<ResponseJuros> CreateResponse(ObjectJC md, string language)
{
    List<ResponseJuros> response = new List<ResponseJuros>();
    foreach (ResultJuros item in md.d.results)
    {
        ResponseJuros rj = new ResponseJuros();
        rj.Link = item.Link;
        rj.Data = item.Data;
        rj.NomeTaxa = item.DenominaçãoDaTaxaDeJuro;
        rj.ValorTaxa = item.ValorDaTaxa;
        response.Add(rj);
    }
    return response;
}
}
```

G.6. Excerto do código do módulo principal da aplicação híbrida

```
angular.module('BancoPortugal', ['ionic', 'ngTouch', 'pascalprecht.translate', 'fsCordova',  
'BancoPortugal.services', 'BancoPortugal.controllers'])
```

```
    .run(function($rootScope, $window){  
        $rootScope.windowWidth = $window.outerWidth;  
        $rootScope.appVersion = appConfig.version;  
        $rootScope.allowRightMenu = false;  
        if ($rootScope.windowWidth < 768) {  
            $rootScope.isTablet = false;  
        } else {  
            $rootScope.isTablet = true;  
        }  
        angular.element($window).bind('resize', function() {  
            $rootScope.windowWidth = $window.outerWidth;  
            $rootScope.$apply('windowWidth');  
        });  
    })  
  
    .config(function($translateProvider) {  
        $translateProvider.translations('en', translations_en);  
        $translateProvider.translations('pt', translations_pt);  
        $translateProvider.preferredLanguage('pt');  
    })  
  
    .config(function($stateProvider, $urlRouterProvider) {  
        .state('eventmenu', {  
            url: "/sidemenu",  
            abstract: true,  
            templateUrl: "views/sidemenu.html"  
        })  
        .state('eventmenu.home', {  
            url: '/home',  
            views: {  
                'menuContent' :{  
                    templateUrl: 'views/home.html',  
                    controller: 'HomeController'  
                }  
            }  
        })  
        ...  
        ...  
  
        $urlRouterProvider.when("", '/sidemenu/home');  
        $urlRouterProvider.otherwise('/sidemenu/home');  
    });
```

G.7. Código da API JavaScript do *plugin* desenvolvido e também do módulo nativo para iOS

API JavaScript

```
cordova.define("cordova/plugin/calculations",
function (require, exports, module) {
    var exec = cordova.require('cordova/exec');

    function simulation(name, win, fail) {
        return exec(win, fail, "Calculations", "simulation", [name]);
    }

    module.exports = {
        simulation: simulation
    };
});
```

Módulo nativo para iOS

Interface

```
#import <Cordova/CDVPlugin.h>
@interface HWPCalculations : CDVPlugin
- (void) simulation:(CDVInvokedUrlCommand*)command;
@end
```

Implementação da interface

```
#import "HWPCalculations.h"
@implementation HWPCalculations
- (void)simulation:(CDVInvokedUrlCommand*)command
{
    NSString* callbackId = [command callbackId];
    NSDictionary *values = [[command arguments] objectAtIndex:0];
    double capitalInicial = [[values objectForKey:@"capitalInicial"]
doubleValue];
    NSUInteger prazo = [[values objectForKey:@"prazo"] integerValue];
    NSUInteger tipoPrazo = [[values objectForKey:@"tipoPrazo"] integerValue];
    double taxaAnual = [[values objectForKey:@"taxaAnual"] doubleValue];
    NSUInteger capitalizacoes = [[values objectForKey:@"capitalizacoes"]
integerValue];
    double juroSimples = (capitalInicial * (taxaAnual/100) * prazo) /
tipoPrazo;
    double capitalSimples = juroSimples + capitalInicial;
    double juroComposto = (capitalInicial * pow((1.0+((taxaAnual/100.0)/
capitalizacoes)),floor((prazo*capitalizacoes)/tipoPrazo)*(1.0+
(((taxaAnual/100.0)/tipoPrazo)*(prazo - ((floor((prazo*capitalizacoes)
/tipoPrazo)*tipoPrazo)/capitalizacoes)))))) - capitalInicial;
    double capitalComposto =juroComposto + capitalInicial;
    NSMutableDictionary *results = [NSMutableDictionary dictionary];
    if(juroSimples == INFINITY){
        [results setValue:@"Infinito" forKey:@"juroSimples"];
    }else{
        NSNumber *juroS = [[NSNumber alloc] initWithDouble:juroSimples];
        [results setValue:juroS forKey:@"juroSimples"];
    }
    if(capitalSimples == INFINITY){
        [results setValue:@"Infinito" forKey:@"capitalSimples"];
    }else{
```

```

    NSNumber *capitalS = [[NSNumber alloc]
initWithDouble:capitalSimples];
[results setValue:capitalS forKey:@"capitalSimples"];
}
if(juroComposto == INFINITY){
[results setValue:@"Infinito" forKey:@"juroComposto"];
}else{
    NSNumber *juroC = [[NSNumber alloc] initWithDouble:juroComposto];
[results setValue:juroC forKey:@"juroComposto"];
}
if(capitalComposto == INFINITY){
[results setValue:@"Infinito" forKey:@"capitalComposto"];
}else{
    NSNumber *capitalC = [[NSNumber alloc]
initWithDouble:capitalComposto];
[results setValue:capitalC forKey:@"capitalComposto"];
}
CDVPluginResult* result = [CDVPluginResult
resultWithStatus:CDVCommandStatus_OK
messageAsDictionary:results];

[self success:result callbackId:callbackId];
}
@end

```

