



UNIVERSIDADE
LUSÓFONA

Trabalho Final de Curso

MoneyBox

Alírio Correia Tavares Mendes

Nº 21300110

Orientador: Prof. Rui Ribeiro

Trabalho Final de Curso | LEI | Abril 2017

Direitos de cópia

MoneyBox Copyright de Alírio Correia Tavares Mendes, ULHT.

A MoneyBox tem o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Índice Geral

Índice Geral.....	III
Índice de Imagens.....	IV
Resumo	V
Abstract	V
1. Introdução	6
2. Enquadramento teórico	6
2.1 Tecnologias	6
2.1.1 Appcelerator Studio.....	6
2.1.2 Xcode	7
2.1.3 Swift.....	7
2.1.4 Postman.....	8
2.1.5 MongoDB.....	8
3. Método.....	9
3.1 Arquitetura.....	9
3.1.1 Camada de apresentação.....	9
3.1.2 Camada de negócio	9
3.1.3 Camada de Dados.....	10
3.2 Desenvolvimento.....	11
3.2.1 Processo de desenvolvimento	11
3.2.2 Informações Técnicas.....	12
3.2.2.1 Integração com a Dropbox	12
3.2.2.2 Notificações remotas (Push Notification)	12
3.2.2.3 Gráficos	12
3.2.2.4 Configuração da folha Excel	13
3.2.2.5 Processamento movimentos.....	14
3.2.2.6 Outros desenvolvimentos	14
4. Resultados.....	15
4.1 Registo e Autenticação.....	15
4.2 Ecrã principal e Navegação	16
4.3 Outros ecrãs relevantes.....	18
4.3.1 Dashboard	18
4.3.2 Movements	19
4.3.3 Entities.....	19
4.3.4 Notifications	21
5. Conclusão e trabalho futuro	22
Bibliografia	24
Glossário	24
Anexos.....	25
Anexo 1 - Requisitos Funcionais.....	26
Anexo 2 - XML exemplo.....	27

Índice de Imagens

Figura 1 - Titanium Appcelerator	6
Figura 2 - Xcode IDE	7
Figura 3 - Swift Language	7
Figura 4 - Postman	8
Figura 5 - Base de dados MongoDB	8
Figura 6 - Arquitetura da MoneyBox	9
Figura 7 - Modelo de dados simplificado.....	10
Figura 8 - Diagrama UML	13
Figura 9 - Webservice em Java	14
Figura 10 - Ecrã de registo	15
Figura 11 - Ecrã de login.....	16
Figura 12 - Dashboard.....	17
Figura 13 - Filtro dos dados do Dashboard	18
Figura 14 - Ecrã dos movimentos.....	19
Figura 15 - Ecrã das entidades	20
Figura 16 - Ecrã de configuração dos Excel.....	20
Figura 17 - Ecrã de notificações	21

Resumo

O objetivo primordial deste Trabalho Final de Curso é criar o *FrontEnd* de uma plataforma de gestão financeira pessoal. A plataforma, MoneyBox (<http://moneyboxplanet.com/>), permite automatizar o processo de leitura de documentos financeiros, como extratos e faturas, nos mais diversos formatos eletrónicos, e apresentar informações estruturadas sobre a saúde financeira do utilizador.

Seguramente já existem algumas plataformas com a mesma ambição, mas a proposta da MoneyBox difere destas no sentido em que foge da integração com a banca, o que tem como vantagens a independência das APIs da banca, e também das legislações no que toca ao sigilo bancário em diferentes países, e também, outras aplicações no mercado dependem também da introdução manual dos movimentos por parte dos utilizadores, o que conta como maior fator da resistência do público em utilizar estas aplicações.

Palavras-chave: Finanças Pessoais, Swift, iPad, Web Services, MongoDB, Desenvolvimento Mobile.

Abstract

The primary objective of this Final Work Project is to create the *FrontEnd* of a personal financial management platform. The platform, Moneybox(<http://moneyboxplanet.com/>), lets you automate the process of reading financial documents, such as statements and invoices, in several electronic formats, and provide structured information on the financial health of the user.

There are already some platforms with the same ambition, however Moneybox's proposal differs from those through avoiding an integration with the bank. This has as advantages the independence of banking APIs and confidentiality laws regarding banking in different countries. Additionally, other applications on the market are dependent on manual entry of a users' movements, which is the main reason why not more people use financial management apps. Which is exactly what Moneybox is solving.

Keywords: Personal Finance, Swift, iPad, Web Services, MongoDB, Mobile Development.

1. Introdução

Este trabalho é desenvolvido no âmbito da disciplina de Trabalho Final de Curso e consiste na realização, em conjunto como uma start-up, de um *FrontEnd* mobile para uma plataforma de gestão pessoal. MoneyBox é uma aplicação financeira que lhe permite analisar as suas finanças pessoais. Trata-se de um serviço na “cloud” capaz de oferecer ao seu utilizador informação crítica sobre as suas finanças pessoais, entender a sua estrutura financeira familiar e construir relatórios de análise com elevado rigor e detalhe.

2. Enquadramento teórico

Neste capítulo, são apresentadas as diversas tecnologias utilizadas no projeto para desenvolver a aplicação.

2.1 Tecnologias

2.1.1 Appcelerator Studio



Figura 1 - Titanium Appcelerator

Inicialmente chamada Titanium Studio, este é o ambiente de desenvolvimento que permite a criação de aplicações moveis iOS, Android e Windows Phone, com um único código JavaScript. A Framework permite aceder a componentes UI nativos como barras, menus e caixas de diálogo, assim como tem funcionalidades como aceder ao sistema de ficheiros, geolocalização, acelerómetro e mapas. Todas o código fonte é enviado para o dispositivo, e depois é interpretado por utilizando um JavaScript “Engine”, que difere de plataforma para plataforma. As aplicações geradas a partir desta tecnologia tendem a um pouco mais a carregar.

Inicialmente a aplicação foi desenvolvida com a ferramenta Appcelerator, essa decisão foi tomada para facilitar a chegada ao mercado em todos as plataformas mobile. Considerou-se o uso do Xamarin, que na altura ainda não tinha sido adequerida pela Microsoft, e por essa razão financeiramente não compensava, mas traria uma melhor performance do que a solução Appcelerator.

No decorrer do projeto chegou-se a conclusão que uma solução nativa (usando o Xcode como IDE e Swift como linguagem) traria mais benefícios em termos de performance.

2.1.2 Xcode



Figura 2 - Xcode IDE

Xcode é o IDE da Apple, para gerenciamento de projetos relacionados com o sistema operacional Mac OS X. Xcode possui ferramentas que permite criar e melhorar seus aplicativos. É um software poderoso e mais simples de utilizar para o desenvolvimento de aplicativos grandes. Antigamente chamava-se Project Builder. O Xcode pode ser obtido no website de developers da Apple (<http://developer.apple.com>).

2.1.3 Swift



Figura 3 - Swift Language

Swift é uma linguagem de programação multiparadigma criada pela Apple para desenvolvimento de programas para iOS e OS X. Apresentada na Apple's 2014 Worldwide Developers Conference, Swift foi projetada para trabalhar com os frameworks da Apple, Cocoa e Cocoa Touch, e com códigos em Objective-C.

Swift oferece maior segurança e performance. É uma linguagem fácil de aprender, o que faz com muitos iniciantes optem por ela para entrar no mundo de programação da Apple.

2.1.4 Postman



Figura 4 - Postman

Postman é uma ferramenta que permite aos programadores testar, partilhar e documentar web APIs. A ferramenta foi muito útil neste trabalho pois permitiu garantir que os serviços estavam a responder devidamente, e a resolver casos de *bugs* que de outra forma seriam défices de encontrar.

2.1.5 MongoDB



Figura 5 - Base de dados MongoDB

MongoDB é uma base de dados *Open Source* noSQL, que usa um modelo de dados orientado a documentos. A contrastar com o modelo relacional que usa tabelas e linhas, um base de dados mongo é constituída por seleções e documentos. No qual os documentos estão compostos por *sets* de pares *key-value* e as coleções são *sets* de documentos. As coleções e documentos são tipicamente representados por documentos JSON. A startup escolheu o mongo porque permite uma maior escalabilidade e flexibilidade na gestão dos dados.

3. Método

3.1 Arquitetura

A plataforma está dividida num modelo de três camadas, de forma que o software executado em cada camada possa ser substituído sem prejuízo para o sistema. De modo que atualizações e correções de bugs possam ser feitas sem prejudicar as demais camadas. Por exemplo: alterações de interface gráfica podem ser realizadas sem o comprometimento das informações contidas na base de dados.

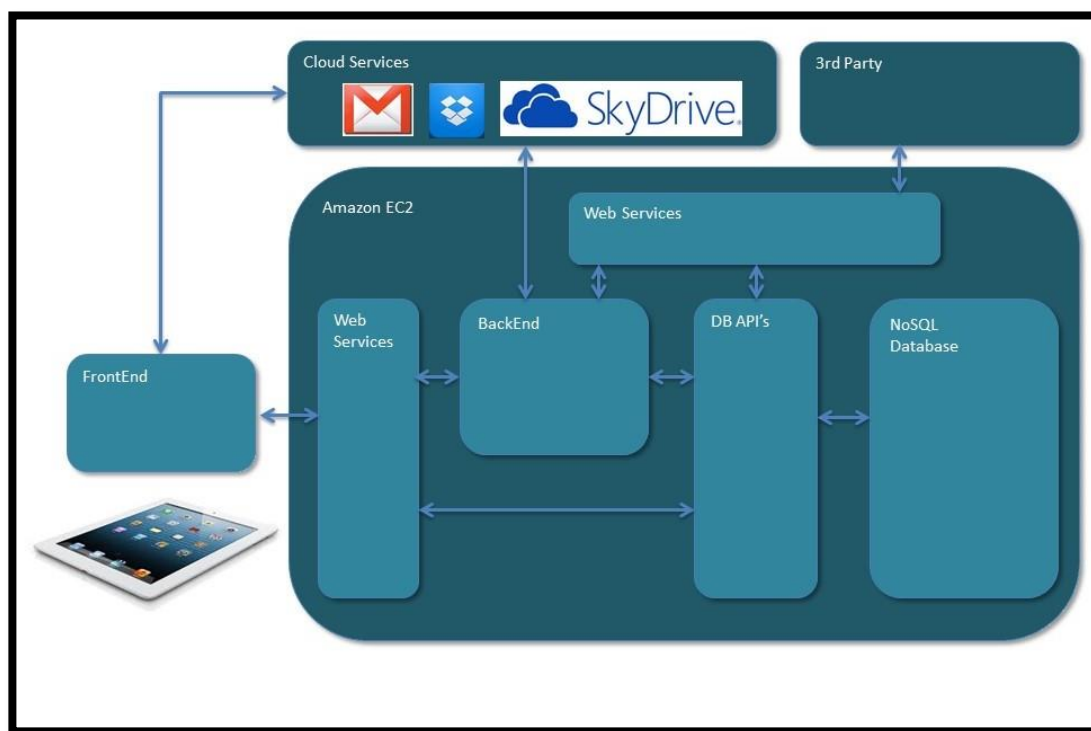


Figura 6 - Arquitetura da MoneyBox

3.1.1 Camada de apresentação

Esta é a camada que os utilizadores irão interagir, pode ser uma aplicação móvel ou *desktop*, um *website*, etc. Nesta camada o utilizador final poderá navegar nos dados e alterá-los. Esta camada tem como responsabilidade de comunicar com a camada de negócio, no intuito de obter a informação mais atualizada e pedir qualquer alteração nos dados.

3.1.2 Camada de negócio

Também chamada de Lógica empresarial, Regras de negócio ou Funcionalidade. É nela que ficam as funções e regras de todo o negócio. Não existe uma interface para o usuário e seus dados são voláteis, ou seja, para que algum dado seja mantido deve ser utilizada a camada de dados. Uma das responsabilidades desta camada no caso desta plataforma é fazer download dos ficheiros (Excel e/ou PDF), ler os dados e enviar o que for necessário para a camada de dados.

3.1.3 Camada de Dados

A terceira camada é definida como o repositório das informações e as classes que a manipulam. Esta camada recebe os pedidos da camada de negócios e seus métodos executam esses pedidos numa base de dados. Uma alteração na base de dados alteraria apenas as classes da camada de dados, mas o restante das camadas não seria afetado por essa alteração.

A camada de dados é servida de uma NoSql base de dados MongoDB, a startup escolheu o mongo porque permite uma maior escalabilidade e flexibilidade na gestão dos dados.

Modelos representados em JSON.

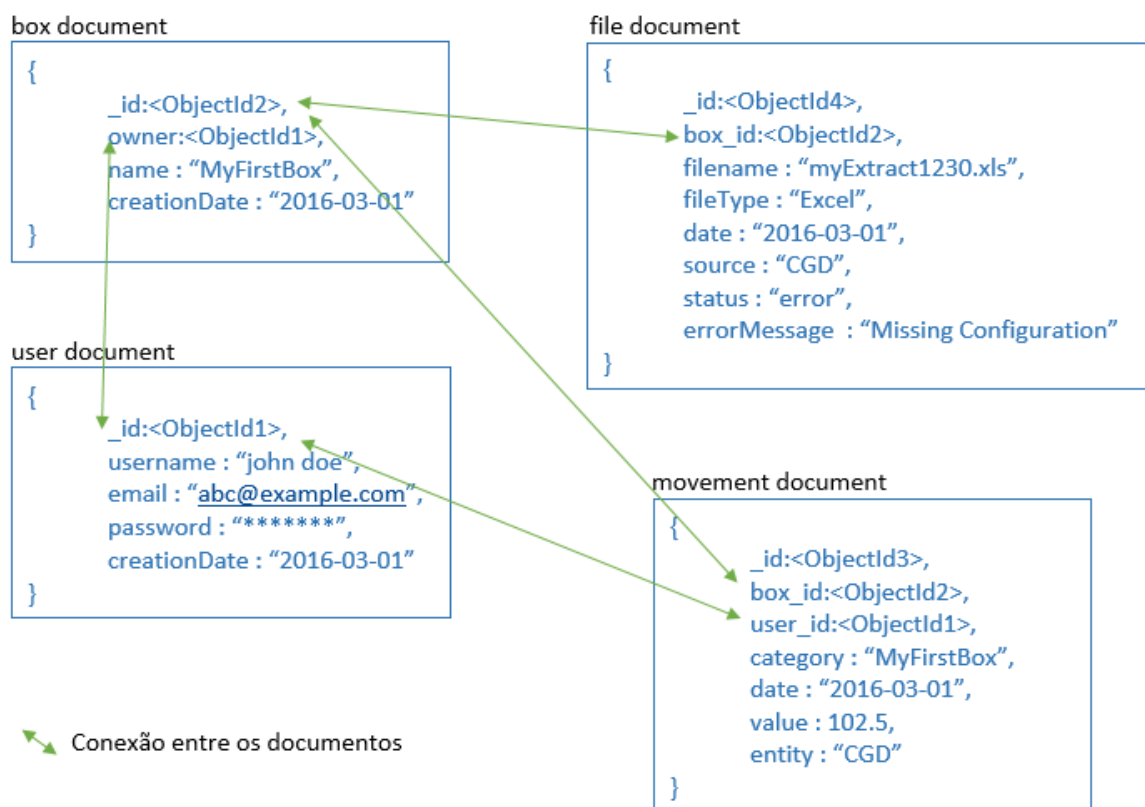


Figura 7 - Modelo de dados simplificado

Na figura 7 é apresentado um modelo simplificado, apenas com os documentos chave do modelo de dados.

3.2 Desenvolvimento

3.2.1 Processo de desenvolvimento

O projeto já ia com alguns meses de desenvolvimento quando me juntei, eles já tinham os requisitos bem definidos. O objetivo inicial era desenvolver utilizando a ferramenta Appcelerator (baseado em JavaScript) com o objetivo de ser multi-plataforma (iOS, Android e Windows Phone).

O objetivo inicial da minha participação era desenvolver apenas alguns requisitos (por exemplo o ecrã do Dashboard e o ecrã de configuração das colunas do Excel) em cima de um protótipo que a empresa já tinha.

Trabalhei nesta versão durante alguns meses, chegando à conclusão que tinha problemas de performance, e também a documentação era fraca. Nesta altura propus aos diretores da Start-up a passagem para uma solução nativa focada no iOS, tendo ele aceitado. Esta alteração fez com que eu tivesse que refazer toda a aplicação, aumentando o âmbito do projeto.

Apesar do âmbito do TFC ser apenas o desenvolvimento da aplicação móvel eu estive envolvido nos processos de tomada de decisão, participando em reuniões semanas onde se debateram questões de negócio, mercado, segurança e muito mais.

Ainda para além do frontend desenvolvi algumas funcionalidades no backend, como serviços REST e envio de Push Notification quando o backend tinha informação nova para os utilizadores.

3.2.2 Informações Técnicas

Nesta secção vão ser apresentadas informações de âmbito técnico sobre vários aspetos do projeto.

3.2.2.1 Integração com a Dropbox

Uma das componentes mais importantes é a integração com os serviços de armazenamento na cloud, como tal nesta fase um dos requisitos implementados foi a integração com a API da Dropbox utilizando uma biblioteca que Swift fornecida pela Dropbox (<https://github.com/dropbox/SwiftyDropbox>), no processo de registo o utilizador tem o como opção a configuração do deste serviço, o utilizar é redirecionado para uma pagina web onde pode efetuar a autenticação no Dropbox e permitir o acesso a nossa plataforma e retorna um token de acesso a nossa aplicação, esse token é enviado ao backend e guardado para acessos futuros. Quando o serviço é configurado o *backend* cria uma pasta onde o utilizador deverá meter os documentos que correspondem as transações financeiras.

3.2.2.2 Notificações remotas (Push Notification)

Um dos requisitos da aplicação é ser possível atualizar os dados sem ser preciso a interação do utilizador. Para tal foram implementadas notificações remotas, em que o servidor ao detetar uma alteração nos ficheiros, vai ler os ficheiros alterados e envia uma notificação para o dispositivo do utilizador e este atualiza os dados sem necessidade de qualquer intervenção por parte do utilizador.

3.2.2.3 Gráficos

O uso de gráficos nesta aplicação é inevitável porque facilita a compreensão e o análise dos dados por parte do utilizador, adicionalmente facilita a comparação das transações num período específico.

Foram implementados vários gráficos:

Gráfico	Requisito	Implementação
Barras vertical	Comparação com períodos homólogos	Implementado por mim
Barras horizontal	Top 5 categorias de despesas	Implementado por mim
<i>Pie Chart</i>	Balanço atual	Biblioteca <i>Charts</i>
Gráfico em linhas	Despesas e receitas no tempo	Biblioteca <i>Charts</i>

3.2.2.4 Configuração da folha Excel

Para processar os dados é preciso saber em que colunas da folha Excel esta cada informação relevante, como por exemplo o montante, data e descrição do movimento.

Para isso foi necessário criar uma *view* para permitir ao utilizador a especificação da sua folha Excel.

Fazendo uma chamada ao *backend* obtenho uma descrição em XML do ficheiro (ver exemplo no anexo 2).

Utilizei uma biblioteca SWXMLHash (<https://github.com/drmohundro/SWXMLHash>) do iOS para fazer parse do XML e com base no XML construí uma tabela tipo Excel no dispositivo, que é representada por objetos das classes do modelo ExcelData.

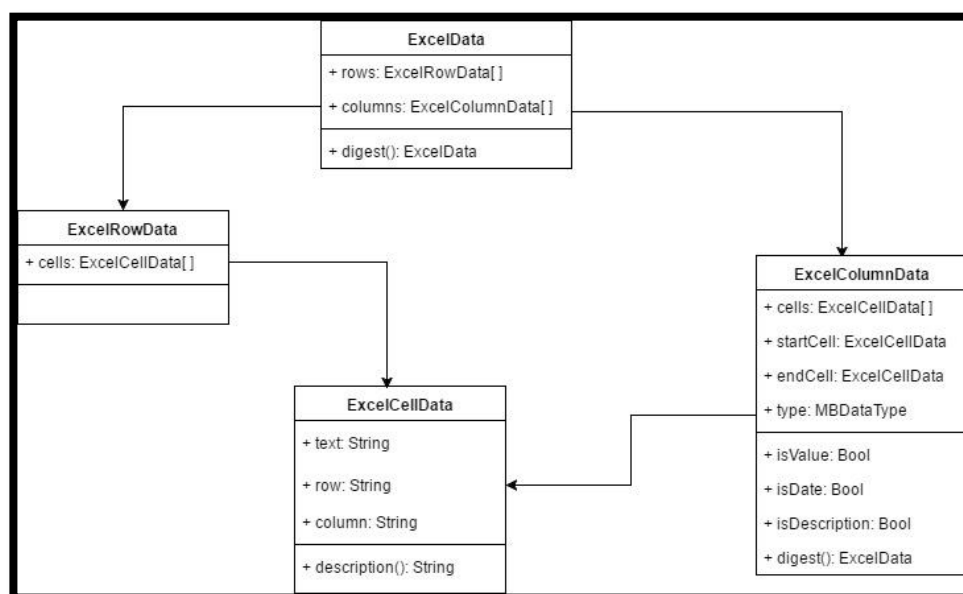


Figura 8 - Diagrama UML

Construí um pequeno motor que percorria os dados do Excel e sugeria ao utilizador o tipo qual tipo de coluna se tratava, facilitando a vida ao utilizador.

Na Figura 8 apresenta-se o diagrama UML do modelo que suporta esse motor. A classe **ExcelData** representa o Excel por inteiro, constituído por linhas e colunas. As **ExcelRowData** são as linhas, sendo **ExcelColumnData** as colunas, ambas preenchidas por células. As células têm uma string que é o seu valor e têm também a informação do seu posicionamento no Excel. As colunas têm também métodos que percorrem todas as células e utilizando *regex* próprios identificadas (com um certo grau de credibilidade) o tipo da coluna.

3.2.2.5 Processamento movimentos

Após o utilizador ter configurado os campos do Excel, o servidor aplicacional irá fazer *download* dos ficheiros e ler linha a linha os documentos e guardar os detalhes das transações na camada de dados usando o modelo apresentado na secção “Camada de Dados”. A aplicação faz pedidos REST e recebe os movimentos e/ou a agregação destes na forma de JSON. Do lado da aplicação esses dados são somente de leitura.

3.2.2.6 Outros desenvolvimentos

No desenvolvimento dos serviços, todas em Java, tive de aprender a lidar com uma base de dados MongoDB a fazer *queries* complexas de agregação de dados, nesse processo aprendi ainda a configurar o servidor aplicacional Glassfish da Oracle e a fazer *deploy* das aplicações para esse servidor.

Um dos serviços implementados foi o de guardar sessão de um utilizador, o serviço recebe um Id de utilizador, a data atual e um objeto que identifica uma sessão, esse objeto tem um token do dispositivo que depois será utilizado para enviar notificações ao utilizador, contem também informações sobre o tipo de dispositivo (IP, EMEI, Tipo, OS, etc.).

```
@POST
@Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
@Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
@Path("/users/{user}/createUserSession")
public Response createUserSession( UserSession userSession,
    @PathParam("user") String userId,
    @QueryParam("timestamp") Long timestamp )
{
    logObjects(userId);
    MoneyResponse<UserSession> mResponse = null;
    userSession.setUser_id(userId);
    if ( (userId == null) || userId.isEmpty() )
    {
        MoneyResponseHeader header = new MoneyResponseHeader(ResponseType.ERROR, "Bad Param. Missing parameters", "63400", null);
        throw new WebApplicationException(Response.status(Status.BAD_REQUEST).entity(header).build());
    }

    try {
        moneyService.createUserSession(userSession);
    }
    catch (MoneyManagerException moneyEx) {
        LOGGER.severe("Caught MoneyServiceException. Something went awfully wrong. (" + moneyEx.getMessage() + ")");
        MoneyResponseHeader header = new MoneyResponseHeader(ResponseType.ERROR, "Could create UserSession", "50500", moneyEx.getMessage());
        mResponse = new MoneyResponse<>(header);

        Response response = Response.serverError().entity(mResponse).build();
        return response;
    }

    MoneyResponseHeader header = new MoneyResponseHeader(ResponseType.SUCCESS, "UserSession Created", "50201", null);
    mResponse = new MoneyResponse<>(header, userSession);

    LOGGER.info(mResponse.toString());
    Response response = Response.created(null).entity(mResponse).build();
    return response;
}
```

Figura 9 - Webservice em Java

Este pedido depois comunica com a camada de negócio, que por sua vez vai fazer um pedido a camada de dados para guardar a sessão na base de dados.

Também foi necessário implementar serviços para fazer uso desta sessão, como pedir os tokens do utilizador, apagar sessões obsoletas, etc.

4. Resultados

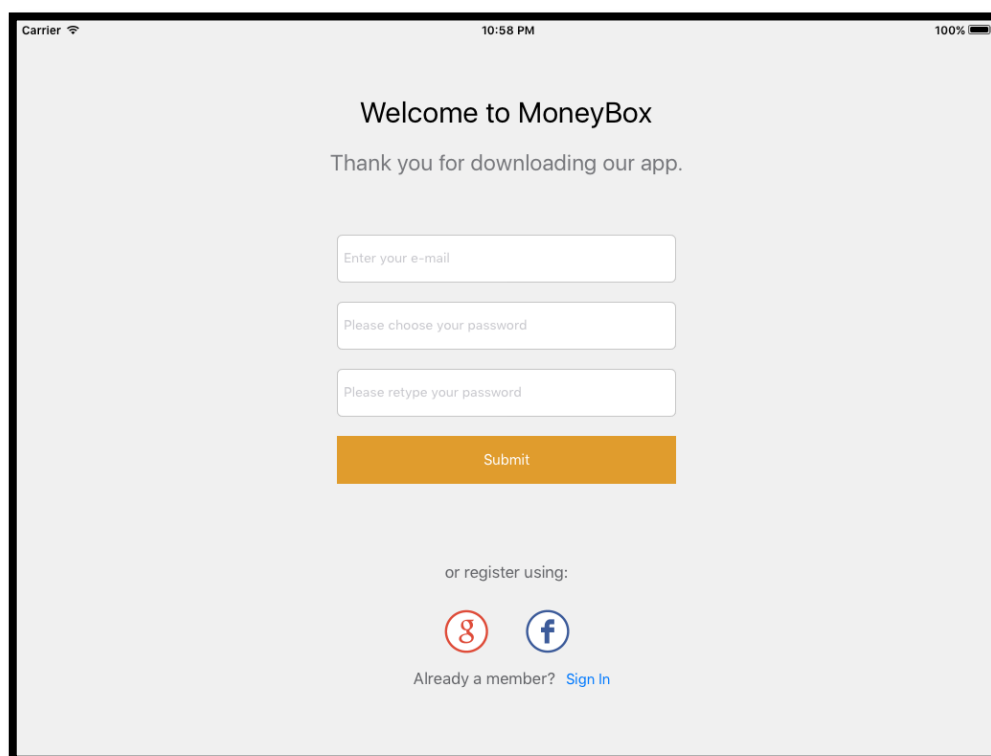
4.1 Registo e Autenticação

Como todos os dados que aplicação utiliza são guardados no *backend*, a atualização da aplicação requer autenticação.

Na primeira utilização o utilizador terá que se registar. O registo é feito a partir do ecrã abaixo, e são necessários apenas o email pessoal e uma palavra passe, nós que queríamos que o registo fosse o mais simples possível para evitar inercia do utilizador em usar a aplicação.

Também planeado, mais não implementado ainda, o utilizador poderá registar-se com uma conta do Gmail ou Facebook.

Ainda na parte do registo o utilizador poderá opcionalmente conectar a conta a uma conta na *cloud* da Dropbox. Este passo é opcional pois o utilizador poderá fazê-lo também dentro da aplicação.



Carrier 10:58 PM 100%

Welcome to MoneyBox

Thank you for downloading our app.



Enter your e-mail

Please choose your password

Please retype your password

Submit

or register using:

Already a member? [Sign In](#)

Figura 10 - Ecrã de registo

Após completar o registo, o utilizador será dirigido para o ecrã principal da aplicação.

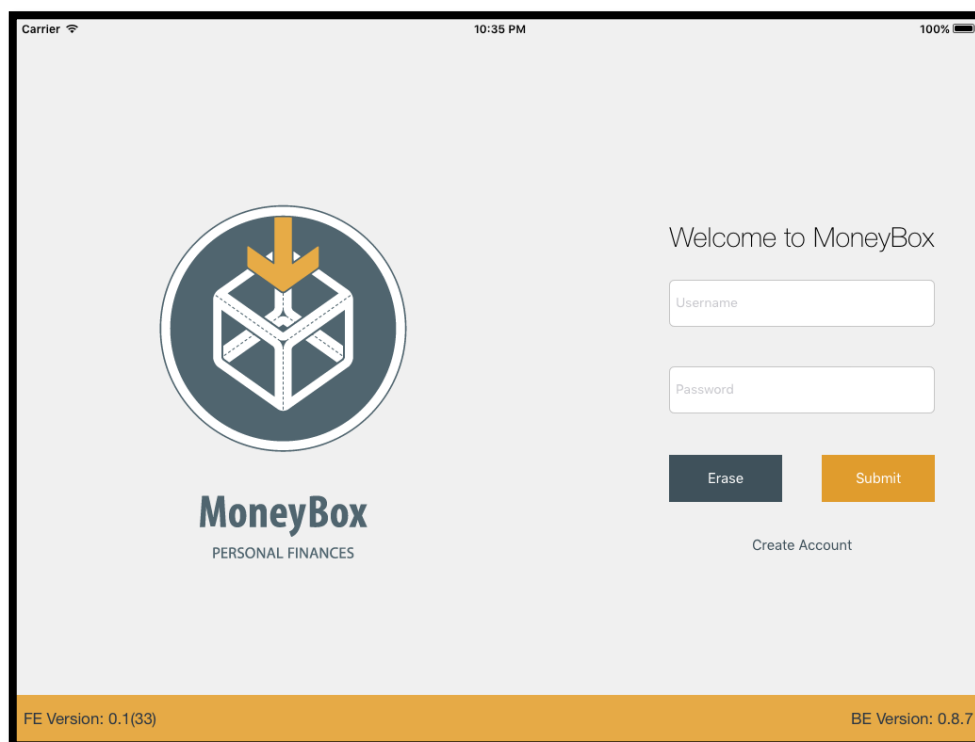


Figura 11 - Ecrã de login

Caso o utilizador faça *logout* ou a sessão termine, para voltar a ter acesso aos a sua conta o utilizador terá que fazer *login* através deste ecrã.

4.2 Ecrã principal e Navegação

Após devida autenticação, o utilizador terá acesso ao ecrã principal da aplicação.

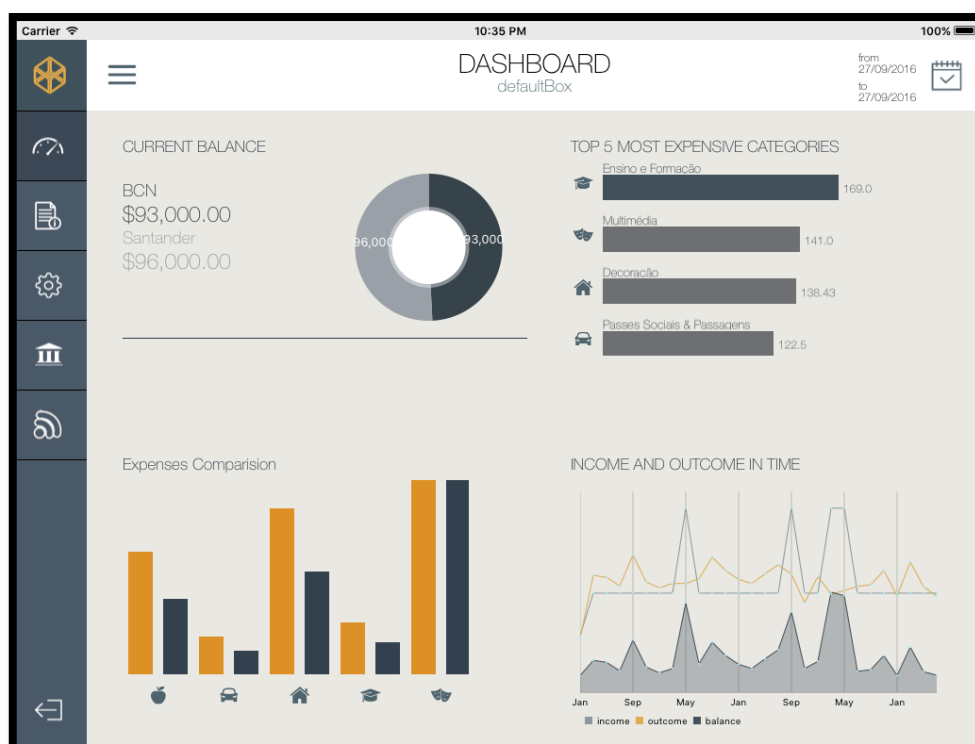


Figura 12 - Dashboard

O ecrã principal está dividido em duas partes, num padrão *master-detail*, do lado esquerdo temos o menu onde podemos alterar o que é apresentado do lado direito ou fazer *logout* da aplicação.

4.3 Outros ecrãs relevantes

4.3.1 Dashboard

No dashboard podemos encontrar representação gráfica dos dados financeiros do utilizador.

Esses dados podem ser salário, compras, venda de ações, etc.

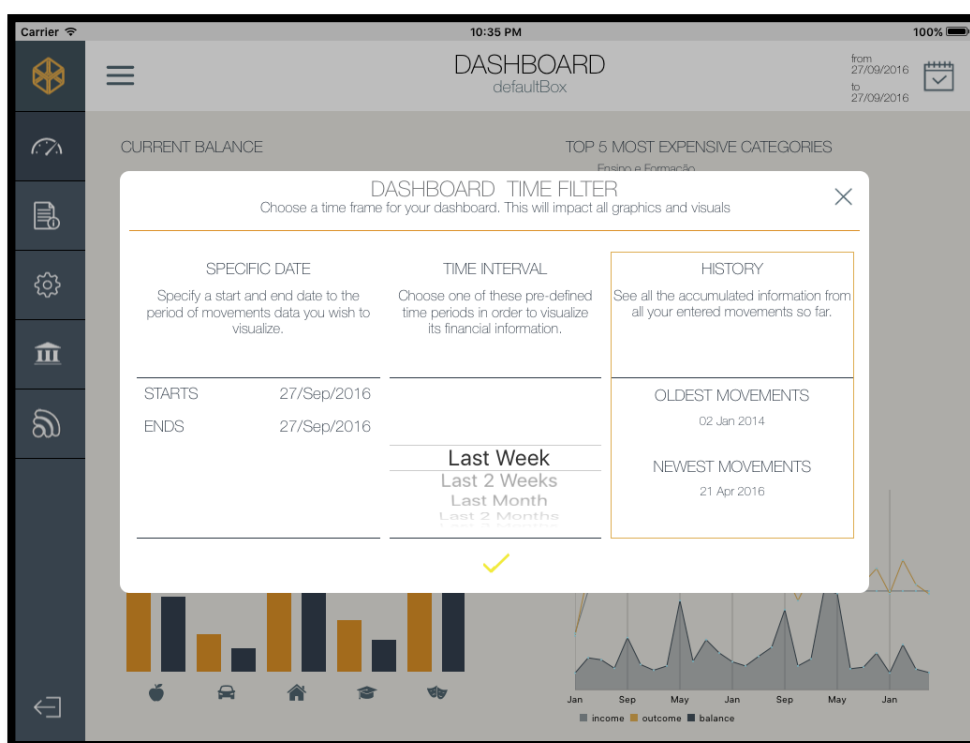
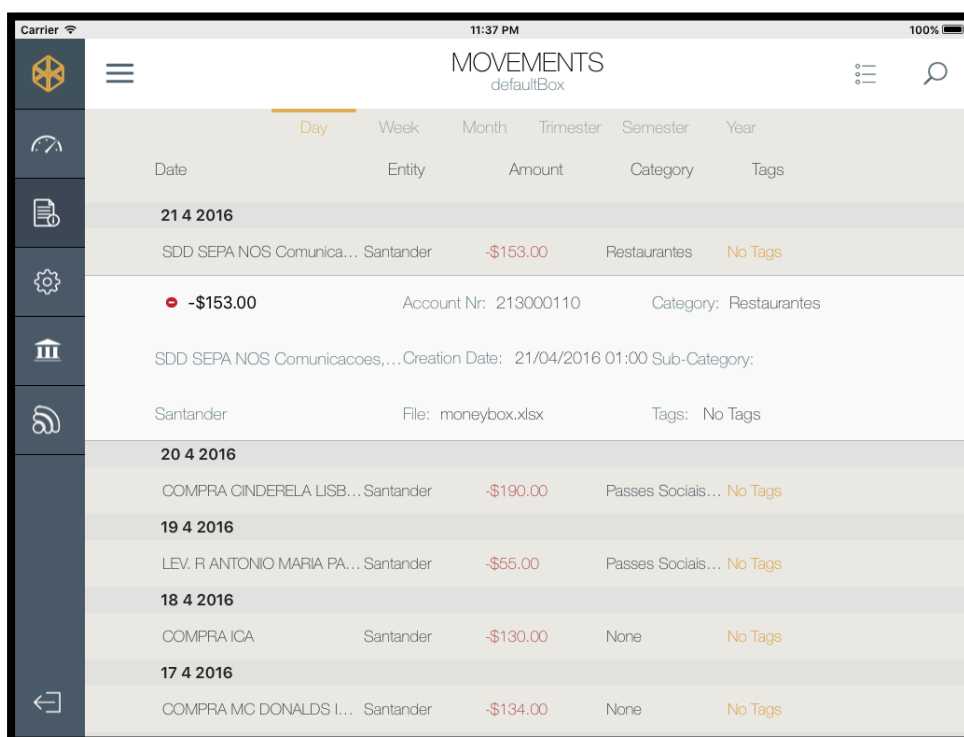


Figura 13 - Filtro dos dados do Dashboard

4.3.2 Movements

No ecrã dos movimentos podemos encontrar os movimentos individuais, assim com eles estão no documento original (Excel, etc...), também poderemos aceder aos ecrãs de agregação, onde os valores dos movimentos se encontram agregadas por semana, mês, trimestre, semestre e ano, assim como gráficos de comparação de períodos homólogos.



MOVEMENTS				
defaultBox				
Date	Entity	Amount	Category	Tags
21 4 2016				
SDD SEPA NOS Comunica...	Santander	-\$153.00	Restaurantes	No Tags
- \$153.00 Account Nr: 213000110 Category: Restaurantes				
SDD SEPA NOS Comunicacoes,... Creation Date: 21/04/2016 01:00 Sub-Category:				
Santander	File: moneybox.xlsx		Tags: No Tags	
20 4 2016				
COMPRA CINDELELA LISB...	Santander	-\$190.00	Passes Sociais...	No Tags
19 4 2016				
LEV. R ANTONIO MARIA PA...	Santander	-\$55.00	Passes Sociais...	No Tags
18 4 2016				
COMPRA ICA	Santander	-\$130.00	None	No Tags
17 4 2016				
COMPRA MC DONALDS I...	Santander	-\$134.00	None	No Tags

Figura 14 - Ecrã dos movimentos

4.3.3 Entities

Neste ecrã poderemos configurar toda a informação sobre as entidades (CGD, Santander, Millennium, etc...) com que trabalhamos, podemos fazer todas operações CRUD sobre elas, podemos ver informação individual de cada entidade como nome, data de criação, a pasta da *cloud* em que se encontram os documentos, o numero de documentos lidos, com sucesso ou não.

Podemos entrar em cada entidade e ver informação de cada documento e editar a configuração de leitura dos documentos desta mesma.

Entity Name	Type of document	Last Execution
BCN	Type	11/04/2016 06:18
<p>These entities need your attention. It might be that some upload were not read correctly</p>		
FILES	27 SUCCESS	LOCATION
27	0 ERROR	/MoneyBox/BCN
LOADINGS		Date
		11/04/2016 06:18
Santander	Type	11/04/2016 06:36

Figura 15 - Ecrã das entidades

Please tap a column, in order to identify the type of value it presents. The gray labels mean MoneyBox is trying to help you. Based on this information all your future extracts from this Source will be read and processed automatically.

DATE OF TRANSFER(?)	ACCOUNT NAME	ACCOUNT NAME	MONEY TRANSFERED	MONEY TRANSFERED(?)
HISTÓRICO DE CONTA NÚMERO 262286449				
Moeda:	EUR			
Tipo:	Todos			
Data de:	2014-10-07			
Data até:	2015-04-07			
Data Lanc.	Data Valor	Descrição	Valor	Saldo
2014-10-07	2014-10-07	TRF P/ UNISEGUROS	-414.04	19.729,68
2014-10-09	2014-10-09	ELE 918627 PETROFAST - POSTOS D	-80.9	19.648,78
2014-10-13	2014-10-13	ELE 918627 EL CORTE INGLES-C OP	-204.8	19.443,98
2014-10-14	2014-10-14	PAG.SERVICOS ENT:21218 REF:231500485	-10.12	19.433,86
2014-10-14	2014-10-14	PAG.SERVICOS ENT:21218 REF:231500486	-13.82	19.420,04
2014-10-14	2014-10-14	PAG.SERVICOS ENT:10147 REF:464597323	-54.27	19.365,77
2014-10-14	2014-10-14	ELE 918627 A S AJUDA-POS 11484	-99.02	19.266,75
2014-10-15	2014-10-15	TRF P/ Condominio G5	-710.55	18.556,20
2014-10-15	2014-10-15	ELE 918627 PAG BX VAL-PORT/ TEL MOV 15	-27.1	18.529,10

Value Date Description

Figura 16 - Ecrã de configuração dos Excel

4.3.4 Notifications

Nas notificações o utilizador receberá as mais variadas informações, desde problemas com a sua com promoções, informações sobre a sua subscrição, etc... Este ecrã funcionará como caixa de email.

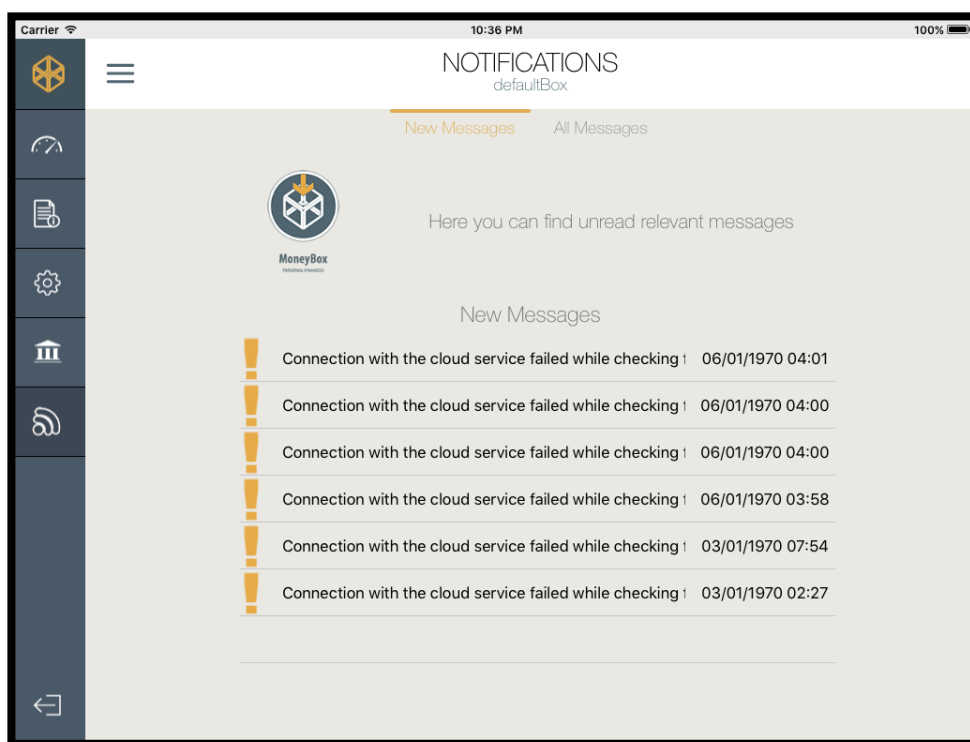


Figura 17 - Ecrã de notificações

5. Conclusão e trabalho futuro

Os meus objetivos pessoais ao começar com MoneyBox eram aprender novas tecnologias e trabalhar em um projecto com ligação ao mercado.

A maior dificuldade que tivemos foi o desempenho do aplicativo inicial porque queríamos uma funcionalidade que iria ler um PDF e ter um quadrado para marcar o campo relevante no ficheiro de fatura. A plataforma Titanium Appcelerator, no entanto, não oferece desempenho nativo como. Como mencionado anteriormente fomos para uma solução nativa, que corrigiu o problema de desempenho, mas que atrasou o projeto por várias semanas.

O que resta a fazer é o saldo da conta em tempo real, que é provavelmente um dos pontos mais importante, porque até agora MoneyBox não está integrado com as instituições financeiras. Isso significa que não temos acesso aos saldos de contas de nossos clientes. Conhecemos apenas as transações individuais e daí podemos agregar e analisar as receitas e despesas do usuário. Isso significa que não sabemos o que era antes das transações mais antigas conhecidas pelo MoneyBox. O saldo atual daria ao utilizador uma visão mais completa de sua situação financeira.

Além disso, existem alguns requisitos menores para implementar no aplicativo como algumas medidas de segurança, comunicação de HTTPS e criptografia de dados.

Até agora MoneyBox só tem uma integração com Dropbox, como uma fonte de dados, mas para o futuro, o objetivo é integrar com todos os outros serviços disponíveis de armazenamento na *cloud*.

O principal objetivo para o aplicativo era dar ao usuário uma maneira fácil de interagir com os dados financeiros usando ecrãs simplistas e feedback proativa. A primeira parte da meta que realizamos, no entanto, o feedback pró-ativo ainda está para ser implementado.

Como foi mencionada antes plataforma é suposto ter integração com todos os serviços de armazenamento na *cloud* mais populares, mas nesta primeira fase nós focamos na dropbox,

deixando assim para uma fase posterior serviços populares como a GoogleDrive e a OneDrive da Microsoft.

Fez-se um protótipo de integração do PDF, mas por dificuldades técnicas não ficou concluído. Como foi dito atrás, nós começamos a desenvolver numa plataforma híbrida e um pouco limitada, por essa razão nos deixamos a parte de leitura de PDF para uma segunda fase.

Ao iniciar o projeto, o objetivo era juntar-se à equipe como um desenvolvedor móvel e usar o MoneyBox como um TFC, mas acabou comigo sendo convidado para me tornar parte dos fundadores.

O mais importante para mim foi ter tido a oportunidade de aprender com pessoas que já que tinham largos anos de experiencia, ter conseguido sair com uma visão da plataforma no seu todo e contribuído em todas as partes do projeto.

Bibliografia

Curso de iOS e Swift

<https://www.udemy.com/complete-ios-8-and-xcode-6-course-make-iphone-ipad-apps/learn/v4/overview>

The Swift Programming Language, 2014 Apple Inc.

Documentação SWXMLHash

<https://github.com/drmohundro/SWXMLHash>

Documentação Glassfish

<https://glassfish.java.net/documentation.html>

Glossário

CRUD → acrónimo de *Create, read, update and delete*.

Backend → Camada da plataforma que corre nos servidores, inclui a camada de negocio e a de dados

Frontend → Camada da plataforma responsável pela apresentação de dados

JSON → JavaScript Object Notation

REST → Representational state transfer

API → Application programming interface

UML → Unified Modeling Language

Master-detail → Padrão de desenvolvimento de UI que divide-se o ecrã em 2, deixando do lado esquerdo a navegação (Master) e do lado direito o conteúdo (Detail)

Push Notifications → Tecnologia utilizada para envio de mensagens para os equipamentos cliente. Neste cenário é o servidor que toma a iniciativa do envio da mensagem.

Regex → Do inglês “*regular expression*”, regex é um padrão que define uma sequência de caracteres.

HTTPS → Hyper Text Transfer Protocol Secure - protocolo de transferência de hipertexto seguro, é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS.

Anexos

Anexo 1 - Requisitos Funcionais

Id	Título	Descrição
1.	Registo de utilizador	O utilizador deve conseguir registar-se na plataforma a partir da aplicação através da introdução de: Email, Password
2.	Login de utilizador	O utilizador de conseguir fazer login na aplicação
3.	Conectar a <i>cloud</i>	Deve ser possível conectar a conta de um utilizador a um serviço de armazenamento na <i>cloud</i> (p.e. Dropbox)
4.	Tutorial inicial	Na primeira utilização da aplicação deve ser exibido um tutorial ao utilizador
5.	Comparação dos gastos em períodos homólogos	Deve ser apresentado um gráfico de barras vertical com a comparação das despesas do utilizador em períodos homólogos (p.e. mensal)
6.	Mostrar ao utilizador em que categorias ele(a) gasta mais	Deve ser apresentado um gráfico de barras horizontal com as categorias mais dispendiosas do utilizador
7.	Mostrar saldo das transações	Deve ser apresentado um gráfico do tipo “Pie chart” com o saldo das transações de cada entidade
8.	Mostrar histórico anual dos movimentos	Deve ser apresentado um gráfico em linhas com os gastos, despesas e balanço mensal ao longo do ano
9.	<i>Dashboard</i> com gráficos	Após o registo e/ou login o primeiro ecrã deve ser um Dashboard com gráficos dos requisitos 5 a 8
10.	Ecrã de transações unitária	Apresentar uma tabela que contenha todas as transações e o detalhe delas
11.	Ecrã de transações agregadas	Deve ser apresentado uma tabela com informações agregadas (p.e. semanais, mensais, trimestrais, etc.)
12.	Ecrã de gestão das entidades	Apresentar um ecrã que permita gerir (Criar, Apagar, Listar) as entidades com que o utilizador trabalha (p.e. Santander, BBVA, etc.)
13.	Ecrã de configuração do Excel	Deve ser apresentado um ecrã onde o utilizador poderá indicar os campos necessários á leitura dos documentos
14.	Ecrã de notificações	Deve ser apresentado uma tabela com informações pertinentes ao estado da conta do utilizador

Anexo 2 - XML exemplo

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document>
<sheet name="2016-03">
<row id="1">
<column id="1">Caixa Agricula</column>
<column id="2"/>
<column id="3"/>
<column id="4"/>
<column id="5"/>
</row>
<row id="2">
<column id="1">Extrato de:</column>
<column id="2">2016-03-25</column>
</row>
<row id="3">
<column id="1">Conta:</column>
<column id="2">1.23456789E8</column>
</row>
<row id="4">
<column id="1">Data Valor</column>
<column id="2">Data Efeito</column>
<column id="3">Descritivo</column>
<column id="4">Montante</column>
<column id="5">Saldo</column>
</row>
<row id="5">
<column id="1">2016-02-26</column>
<column id="2">2016-02-27</column>
<column id="3">COMPRA PINGO DOCE</column>
<column id="4">-158.0</column>
<column id="5">3058.0</column>
</row>
<row id="6">
<column id="1">2016-02-27</column>
<column id="2">2016-02-28</column>
<column id="3">COMPRA ENERGETICA COMB LDA CASAL A</column>
<column id="4">-139.0</column>
<column id="5">2919.0</column>
</row>
<row id="7">
<column id="1">2016-02-28</column>
<column id="2">2016-02-29</column>
<column id="3">LEV. R PROF GUILHERME ASSUNCAOMAFR</column>
<column id="4">-127.0</column>
<column id="5">2792.0</column>
</row>
<row id="8">
<column id="1">2016-02-29</column>
```

```

<column id="2">2016-03-01</column>
<column id="3">COMPRA ICA</column>
<column id="4">-59.0</column>
<column id="5">2733.0</column>
</row>
<row id="9">
<column id="1">2016-03-01</column>
<column id="2">2016-03-02</column>
<column id="3">COMPRA ICA</column>
<column id="4">-126.0</column>
<column id="5">2607.0</column>
</row>
<row id="10">
<column id="1">2016-03-02</column>
<column id="2">2016-03-03</column>
<column id="3">COMPRA LIDL</column>
<column id="4">-183.0</column>
<column id="5">2424.0</column>
</row>
<row id="11">
<column id="1">2016-03-03</column>
<column id="2">2016-03-04</column>
<column id="3">COMPRA ENERGETICA COMB LDA CASAL A</column>
<column id="4">-89.0</column>
<column id="5">2335.0</column>
</row>
<row id="12">
<column id="1">2016-03-04</column>
<column id="2">2016-03-05</column>
<column id="3">TRF.EX-011833883 DE NOS TECHNOLOGY</column>
<column id="4">3000.0</column>
<column id="5">5335.0</column>
</row>
<row id="13">
<column id="1">2016-03-05</column>
<column id="2">2016-03-06</column>
<column id="3">LEV. R PROF GUILHERME ASSUNCAOMAFR</column>
<column id="4">-134.0</column>
<column id="5">5201.0</column>
</row>
<row id="14">
<column id="1">2016-03-06</column>
<column id="2">2016-03-07</column>
<column id="3">SDD SEPA PR BE WATER,SA</column>
<column id="4">-177.0</column>
<column id="5">5024.0</column>
</row>
<row id="15">
<column id="1">2016-03-07</column>
<column id="2">2016-03-08</column>

```

```

<column id="3">COMPRA TELEFERICO DE GAIA VILA NO</column>
<column id="4">-199.0</column>
<column id="5">4825.0</column>
</row>
<row id="16">
<column id="1">2016-03-08</column>
<column id="2">2016-03-09</column>
<column id="3">RECIBO DE EMPRESTIMO D089-20019</column>
<column id="4">-450.0</column>
<column id="5">4375.0</column>
</row>
<row id="17">
<column id="1">2016-03-09</column>
<column id="2">2016-03-10</column>
<column id="3">COMPRA LIVRARIA BERTRAND 1990-09</column>
<column id="4">-145.0</column>
<column id="5">4230.0</column>
</row>
<row id="18">
<column id="1">2016-03-10</column>
<column id="2">2016-03-11</column>
<column id="3">COMPRA H3 EXPO LISBOA</column>
<column id="4">-92.0</column>
<column id="5">4138.0</column>
</row>
<row id="19">
<column id="1">2016-03-11</column>
<column id="2">2016-03-12</column>
<column id="3">SDD SEPA EDP</column>
<column id="4">-126.0</column>
<column id="5">4012.0</column>
</row>
<row id="20">
<column id="1">2016-03-12</column>
<column id="2">2016-03-13</column>
<column id="3">COMPRA ENERGETICA COMB LDA CASAL A</column>
<column id="4">-50.0</column>
<column id="5">3962.0</column>
</row>
<row id="21">
<column id="1">2016-03-13</column>
<column id="2">2016-03-14</column>
<column id="3">COMPRA FARMACIA OCIDENTAL 2560-20</column>
<column id="4">-57.0</column>
<column id="5">3905.0</column>
</row>
<row id="22">
<column id="1">2016-03-14</column>
<column id="2">2016-03-15</column>
<column id="3">LEV. R PROF GUILHERME ASSUNCAOMAFR</column>

```

```
<column id="4">-90.0</column>
<column id="5">3815.0</column>
</row>
<row id="23">
<column id="1">2016-03-15</column>
<column id="2">2016-03-16</column>
<column id="3">COMPRA OCEANPROFIT LDA 1990-20</column>
<column id="4">-115.0</column>
<column id="5">3700.0</column>
</row>
<row id="24">
<column id="1">2016-03-16</column>
<column id="2">2016-03-17</column>
<column id="3">COMPRA SUSHITIME</column>
<column id="4">-66.0</column>
<column id="5">3634.0</column>
</row>
<row id="25">
<column id="1">2016-03-17</column>
<column id="2">2016-03-18</column>
<column id="3">COMPRA ENERGETICA COMB LDA CASAL A</column>
<column id="4">-67.0</column>
<column id="5">3567.0</column>
</row>
<row id="26">
<column id="1">2016-03-18</column>
<column id="2">2016-03-19</column>
<column id="3">COMPRA MC DONALDS IMPERIAL PORTO</column>
<column id="4">-91.0</column>
<column id="5">3476.0</column>
</row>
<row id="27">
<column id="1">2016-03-19</column>
<column id="2">2016-03-20</column>
<column id="3">COMPRA ICA</column>
<column id="4">-27.0</column>
<column id="5">3449.0</column>
</row>
</sheet>
</document>
```