

Mestrado em Engenharia Informática

Estágio

Relatório Final

Plataforma de apoio ao desenvolvimento de negócio

Plataforma de Contabilidade

Mónica Sofia Pais Graça

mgraca@student.dei.uc.pt

Orientador do DEI:

Carlos Fonseca

Orientador da Wit:

Raul Fonseca

Data: 12 de Julho de 2012



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Agradecimentos

Aos meus pais e irmãos, a quem eu devo tudo o que sou hoje.

Ao meu namorado André, que permaneceu a meu lado durante estes 6 anos e me ajudou em todas as fases complicadas.

Ao meu orientador Raul Fonseca, pela ajuda, paciência e esforço durante todo o estágio.

À Susana Órfão pela amizade, paciência, ajuda e tardes passadas na biblioteca da WIT a tentar fazer o nosso melhor.

À Raquel Marques e Sara Marques Gouveia pela amizade, ajuda e paciência na revisão de todos os documentos.

À WIT pela oportunidade de trabalhar num excelente ambiente e pela responsabilidade que me foi atribuída.

Ao meu orientador Carlos Fonseca, pelos conselhos e ajuda na revisão dos documentos.

Por último mas não menos importante, à minha família e amigos que me motivaram de todas as maneiras possíveis e que passaram meses a ouvir falar do estágio, contabilidade e afins, mas que nunca deixaram de estar presentes.

Obrigada a todos por tudo.

Resumo

Com o crescimento das empresas, poupar tempo em tarefas rotineiras para permitir que os recursos sejam utilizados noutras áreas passa a ser um ponto fundamental. Os processos de contabilidade, se não forem agilizados, podem fazer com que se demore horas a compilar dados que, de outra forma, poderiam ficar disponíveis em apenas um minuto.

Este Estágio de Mestrado em Engenharia Informática teve como objetivo o desenvolvimento de uma ferramenta que suportasse o processo de contabilidade da WTT como está definido e que permitisse integração, neste caso específico, com o PHC.

Desta forma foi desenvolvida uma Plataforma Web de Contabilidade que permite ao utilizador visualizar todos os dados de forma atrativa e intuitiva, deixando a compilação dos dados à distância de um clique.

No futuro, está prevista uma adaptação a possíveis alterações que sejam feitas na estrutura de contabilidade da WTT, seguida de um melhoramento da interface da plataforma com o utilizador.

Palavras-Chave

Contabilidade, Contabilidade Analítica, Contabilidade Geral, Relatórios, *Dashboards*, Notificações, Usabilidade, *Business Intelligence*, *Software Quality*, Recursos Humanos.

Índice

1	Introdução	1
1.1	Contextualização	1
1.2	Motivação.....	1
1.3	Objetivos	1
1.4	Estrutura do Documento.....	2
1.5	Noções Gerais	2
1.5.1	Contabilidade.....	2
1.5.2	Business Intelligence	3
2	Estado da Arte	5
2.1	Produtos de Contabilidade	5
2.1.1	PHC Contabilidade.....	5
2.1.2	Primavera Executive.....	5
2.1.3	CentralGest.....	6
2.1.4	Intuit – QuickBooks Enterprise	6
2.1.5	RCSOFT – xSoft ERP.....	6
2.1.6	SAP ERP.....	7
2.1.7	Gestware Administrativa	7
2.1.8	Comparação.....	8
2.2	Produtos de <i>Business Intelligence</i>	11
2.2.1	Zoho Reports	11
2.2.2	Pentaho.....	11
2.2.3	DMWorks – DMReports.....	12
2.2.4	LogiXML.....	12
2.2.5	Dundas	12
2.2.6	iDashboards	13
2.2.7	InetSoft Style Scope.....	13
2.2.8	InetSoft Visualization Free.....	13
2.2.9	Birst.....	14
2.2.10	Arcplan Enterprise.....	14
2.2.11	Quadbase.....	14
2.2.12	Comparação.....	15
2.3	Considerações Finais	18
3	Requisitos.....	19

3.1	Levantamento de Requisitos	19
3.2	Requisitos Externos	19
3.3	Requisitos Funcionais	20
3.4	Requisitos Não Funcionais	22
3.4.1	Performance	23
3.4.2	Escalabilidade	24
3.4.3	Segurança	26
3.4.4	Robustez.....	27
3.5	Mockups	27
4	Arquitetura.....	29
4.1	Arquitetura Geral	29
4.1.1	Plataforma de contabilidade/Back Office.....	30
4.1.2	Servidor de Gráficos.....	33
4.1.3	Web Service PHC	33
4.2	Tecnologias	34
4.2.1	Camada de Apresentação.....	34
4.2.2	Camada de Negócio.....	35
4.2.3	Camada de Persistência.....	35
4.2.4	API's Gráficas	35
4.2.5	Restantes Tecnologias	38
5	Implementação.....	39
5.1	Diagrama de <i>Packages</i>	39
5.1.1	AccountingPlatform	39
5.1.2	Web Service PHC	40
5.1.3	PHPService	41
5.2	Diagrama de Bibliotecas Externas.....	41
5.2.1	Accounting Platform.....	41
5.2.2	PHPService	42
5.3	Modelo da Base de Dados	43
5.4	Detalhes de Implementação	46
5.4.1	Contabilidade Analítica	46
5.4.2	Contabilidade Geral.....	47
5.4.3	Back Office	48
5.4.4	Otimizações	48

5.5	Produto Final.....	50
5.6	Testes	53
6	Plano de Trabalho e Implicações	55
6.1	Processo.....	55
6.2	Equipa.....	56
6.3	Planeamento Inicial.....	56
6.4	Planeamento Intermédio.....	57
6.5	Planeamento Final.....	58
7	Conclusões Finais e Trabalho Futuro.....	61
8	Referências.....	63

Lista de Tabelas

Tabela 1 – Duração de uma análise de dados de contabilidade da empresa antes e depois do desenvolvimento deste estágio.....	1
Tabela 2 – Comparação de soluções.....	9
Tabela 3 – Comparação de soluções.....	16
Tabela 4 – Requisitos do <i>web service</i>	20
Tabela 5 – Requisitos do <i>php service</i>	20
Tabela 6 – Requisitos de sessão.....	20
Tabela 7 – Requisitos de contabilidade analítica.	21
Tabela 8 – Requisitos de contabilidade geral.....	21
Tabela 9 – Requisitos do módulo de recursos humanos.	21
Tabela 10 – Requisitos do módulo de <i>software quality</i>	22
Tabela 11 – Requisitos do módulo <i>back office</i>	22
Tabela 12 – Listagem de requisitos não funcionais da aplicação.....	23
Tabela 13 – Classificação das regras.....	24
Tabela 14 – Lista de vantagens e desvantagens existentes em cada API.....	37
Tabela 15 – Packages criados na plataforma.	40
Tabela 16 – Packages usados no <i>web service</i>	41
Tabela 17 – Descrição detalhada de bibliotecas externas utilizadas na plataforma.	42
Tabela 18 – Descrição detalhada das bibliotecas externas utilizadas no serviço de gráficos..	42
Tabela 19 – Descrição detalhada do ER.	45

Lista de Figuras

Figura 1 – Comportamento da ferramenta utilizando o jMeter.....	24
Figura 2 – Análise de performance.	25
Figura 3 – Análise de CPU e memória.	25
Figura 4 – Arquitetura da plataforma	29
Figura 5 – Arquitetura da plataforma de contabilidade e as suas dependências.....	31
Figura 6 – Arquitetura do back office.....	32
Figura 7 – Dependência da plataforma com o serviço de gráficos	33
Figura 8 – Arquitetura do web service PHC.....	34
Figura 9 – Menu inicial da plataforma.	50
Figura 10 – Módulo da contabilidade analítica.	51
Figura 11 – Módulo de contabilidade geral.....	51
Figura 12 – Módulo de Recursos Humanos.	52
Figura 13 – <i>Back Office</i>	52
Figura 14 – Planejamento Inicial.	56
Figura 15 – Planejamento Intermédio.	57
Figura 16 – Planejamento Final.	58

Anexos

Anexo A – Estado da Arte

Anexo B – Requisitos não Funcionais

Anexo C – Especificação de Requisitos Funcionais

Acrónimos

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
CRM	<i>Customer Relationship Management</i>
ERM	<i>Employee Relationship Management</i>
ERP	<i>Enterprise Resource Planning</i>
FSE	<i>External Supplies and Services</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
KPI	<i>Key Performance Indicators</i>
MVC	<i>Model-View-Controller</i>
OLAP	<i>Online Analytical Processing</i>
RH	<i>Human Resources</i>
SNC	Sistema Nacional de Contas
SSO	<i>Single Sign-On</i>
SWQ	<i>Software Quality</i>
XML	<i>Extensible Markup Language</i>

Glossário

<i>Dashboard</i>	<i>Interface</i> que apresenta a informação mais importante para o utilizador de forma apelativa, tornando mais fácil a interação e respetiva leitura.
<i>Mockups</i>	<i>Design</i> que a plataforma deverá apresentar no fim do seu desenvolvimento.
<i>Tableau de Bord</i>	Tabela que apresenta indicadores financeiros e não financeiros.
Tesouraria	Tabela que apresenta os valores das contas bancárias da empresa.

1 Introdução

1.1 Contextualização

O projeto realizado neste estágio enquadra-se na área da contabilidade e de *Business Intelligence*. Todos os dados contabilísticos da empresa serão acedidos através de uma plataforma web, onde poderão ser visualizados através de gráficos e/ou tabelas, sendo ainda possível o *download* dos mesmos para visualização *offline*.

Business Intelligence enquadra-se no presente estágio devido à forma como todos os dados analisados são visualizados e estudados. A contabilidade da WIT encontra-se organizada em unidades, clientes e projetos, formando a conjugação dos três um centro de custo. A partir destes podemos consultar quaisquer custos, proveitos e resultados obtidos num caso em particular.

1.2 Motivação

A contabilidade é fundamental nas empresas para que estas consigam evoluir. Com o aumento de clientes e/ou projetos os processos de contabilidade tornam-se muito repetitivos e consomem muitos recursos. O desenvolvimento de uma plataforma de contabilidade que permita analisar e visualizar todos os dados detalhadamente, poupando tempo e recursos, vem facilitar todos esses processos, sendo este o principal objetivo deste trabalho.

1.3 Objetivos

Os objetivos deste trabalho concentram-se nos seguintes pontos:

- Criação de uma plataforma web de visualização de dados de contabilidade;
- Criação de um *back office* para introdução de dados e controlo de permissões.

Poupar tempo é um ponto fundamental para permitir que os recursos sejam utilizados noutras áreas. Desta forma, dados que anteriormente demoravam dias a ser compilados, ficam à distância de um único clique, aumentando assim a produtividade.

Antes	Depois
80h	1min

Tabela 1 – Duração de uma análise de dados de contabilidade da empresa antes e depois do desenvolvimento deste estágio.

Na tabela acima podemos observar a duração de uma tarefa de análise de dados contabilísticos da empresa. Os valores indicados na tabela foram obtidos através de questões feitas aos funcionários da empresa. Anteriormente, realizar um relatório contendo todos os dados de contabilidade de um ano demorava 80 horas. Hoje em dia demora apenas 1 minuto. Para isso basta consultar na plataforma os dados pretendidos e realizar o seu *download*. Um dos casos que prova a poupança de tempo é um acontecimento que se deu numa reunião da WIT, onde existe um pedido para se mudar um valor de orçamento introduzido no *back office*. Após esse valor ter sido introduzido, os valores que dependiam do mesmo foram automaticamente atualizados e consultados pelas pessoas presentes na reunião

no minuto seguinte. Com base neste exemplo, podemos concluir que o tempo que um funcionário gasta a realizar todo o tipo de relatórios de contabilidade diminuiu significativamente.

A criação de uma plataforma web de contabilidade vem permitir ao utilizador visualizar todos os dados de contabilidade da empresa através de gráficos e tabelas, sendo ainda possível obter relatórios dos mesmos.

Sendo os dados apresentados de forma gráfica, tornam-se mais apelativos permitindo uma interpretação mais eficaz, que consequentemente permite uma tomada de decisão mais bem informada por parte dos leitores.

1.4 Estrutura do Documento

O presente documento é composto por 8 capítulos:

1. Introdução;
2. Estado da Arte;
3. Requisitos;
4. Arquitetura;
5. Implementação;
6. Plano de Trabalho e Implicações;
7. Conclusão;
8. Referências.

O capítulo 2 - Estado da Arte, é dividido em dois tipos de análises: análise de produtos de contabilidade e de *Business Intelligence*. De seguida é feita uma análise comparativa entre eles. A análise de requisitos é apresentada no capítulo 3, contendo uma componente referente às tecnologias utilizadas. A arquitetura é descrita no capítulo 4, onde são apresentadas todas as componentes importantes da plataforma. No capítulo 5 encontra-se informação detalhada sobre a implementação. No capítulo 6 é apresentado o planeamento do estágio e a metodologia usada. Por fim, temos os capítulos 7 e 8, onde se encontram a conclusão e as referências, respetivamente.

1.5 Noções Gerais

1.5.1 Contabilidade

Contabilidade [1], [2] é o processo de classificar e gravar, geralmente em forma de quantias monetárias, eventos ou transações de carácter financeiro. Este processo tem também como objetivo facilitar a comunicação entre entidades/utilizadores de modo a que exista sempre uma noção do estado da empresa e de como esta se encontra no mercado.

A contabilidade divide-se em três tipos: orçamental, geral e analítica. No âmbito deste estágio apenas serão referenciadas a geral e a analítica, uma vez que a contabilidade orçamental não é relevante neste contexto.

A contabilidade geral é um sistema de informação económica e financeira que trata dos dados na sua generalidade e faz o acompanhamento global dos negócios de uma

empresa. Tem como objetivo a recolha exaustiva de dados financeiros (incluindo débitos e créditos), através de documentos, faturas, recibos, etc. Estes são guardados e documentados pelo ano fiscal, sendo obrigatório o seu arquivo por 10 anos. Como complemento da contabilidade geral podem existir os seguintes módulos:

- Tesouraria;
- “*Tableau de Bord*”;
- Recursos Humanos.

A contabilidade analítica é um exercício mais detalhado e permite acompanhar a evolução de cada projeto (ou centro de custo) em termos de custos, proveitos e resultados. Os principais objetivos da contabilidade analítica são os seguintes:

- Conhecer os custos, proveitos e resultados dos projetos e departamentos;
- Estabelecer previsões dos custos por departamento e por projeto;
- Controlar o orçamento anual atribuído a cada centro de custo e fixar responsabilidades aos diferentes gestores de cada uma das secções da empresa;
- Calcular os desvios orçamentais por departamento e por projeto;
- Avaliar a contribuição de cada projeto nos resultados da empresa;
- Determinar a rentabilidade dos projetos.

Com a contabilidade analítica, a empresa pode calcular o ponto crítico de cada projeto, ou seja, determinar a partir de que momento um projeto deixa de ter prejuízos para ser rentável e analisar quais os projetos mais lucrativos para a empresa.

1.5.2 Business Intelligence

Business Intelligence (BI) [8] consiste em técnicas que permitem identificar, extrair e analisar os dados existentes no negócio, como por exemplo os proveitos da venda de um determinado produto ou ainda os custos associados a um determinado centro de custo. Como objetivos de BI temos:

- Perceber os pontos fortes e fracos da empresa;
- Perceber as diferenças existentes entre dados de modo a tomar melhores decisões;
- Detetar possíveis oportunidades para inovar.

Através de ferramentas de BI os funcionários podem extrair e transformar todos os dados para posterior análise. Após isso é possível colocar as análises efetuadas em relatórios e *dashboards*. As ferramentas usam dados recolhidos de diversas fontes para juntar numa base-de-dados, após isso cada colaborador poderá aceder aos dados, criar gráficos, tabelas entre outros. Iremos abordar em mais pormenor algumas destas ferramentas no próximo capítulo.

2 Estado da Arte

Este capítulo divide-se em dois tipos de análise: produtos de contabilidade e de *Business Intelligence*, e tem como objetivo analisar os pontos fortes e fracos de cada um.

Para a escolha das ferramentas analisadas foram tidos em conta critérios como a produção de relatórios, visualização gráfica, detalhe das pesquisas, *dashboards* e usabilidade. Todos os dados indicados neste capítulo encontram-se bastante resumidos. Para ver o documento completo e em maior detalhe consultar o anexo A – Estado da Arte.

O capítulo divide-se em dois subcapítulos: um que analisa os produtos de contabilidade e faz uma comparação dos mesmos e outro que analisa os de BI e respetiva comparação.

2.1 Produtos de Contabilidade

Esta secção apresenta uma análise das principais soluções comerciais existentes na área da contabilidade geral e analítica, nomeadamente:

- PHC;
- Primavera Executive;
- Centralgest – ERP;
- Intuit – QuinckBooks Enterprise;
- RCSoft - xSoft ERP;
- SAP;
- Gestware Administrativa.

2.1.1 PHC Contabilidade



O módulo de contabilidade do PHC faz parte de uma das soluções do produto – solução financeira. Esta permite o controlo de todas as necessidades contabilísticas numa vertente geral, analítica e orçamental. O objetivo do PHC é agilizar todos os processos financeiros utilizando ferramentas próprias, que torna mais fácil a gestão financeira. Este é o produto de contabilidade usado pela WIT.

A plataforma permite a visualização de informação detalhada de movimentos gerais ou por centros de custo e contém funcionalidades como contabilidade analítica, geral e orçamental.

O licenciamento este é individual sendo as licenças atribuídas por máquina.

2.1.2 Primavera Executive



Primavera é uma empresa que se dedica ao desenvolvimento de *software* de gestão. A sua ferramenta permite o suporte e gestão do negócio empresarial através da monitorização do desempenho e dos resultados da organização.

De todos os módulos disponíveis apenas nos focaremos no módulo de contabilidade, pois é o mais relevante neste contexto. Este permite a introdução de movimentos, que podem ser associados/repartidos por centros de custos. Para cada um, estão disponíveis operações como análises resumidas/detalhadas e balancetes. Todos os valores observados pelo utilizador podem ser alterados e consultados em qualquer momento. Contém um *dashboard* denominado “Monitor de Negócio”, onde o utilizador pode analisar uma série de dados da empresa, como por exemplo, o volume de vendas realizadas ao longo do tempo.

Quanto ao licenciamento, este é comercial. O produto não disponibiliza uma versão *trial* por isso a análise foi feita com base nas funcionalidades anunciadas no *site*.

2.1.3 CentralGest centralgest

A CentralGest é uma empresa de desenvolvimento de *software* de gestão empresarial, sendo um dos seus produtos o CentralGest ERP. Este é uma solução a nível de contabilidade geral e analítica, recursos humanos e gestão de tesouraria.

O produto em estudo tem principal utilidade nas áreas de análise financeira e respetiva criação de relatórios. Como principais vantagens temos a gestão global de todos os conteúdos financeiros importantes para uma empresa, incluindo todos os dados de tesouraria e recursos humanos. Na gestão e análise de dados só é possível observar movimentos no global, não existe pormenorização a qualquer nível.

2.1.4 Intuit – QuickBooks Enterprise

QuickBooks organiza todo o tipo de contas conseguindo, em qualquer altura, indicar ao utilizador em que estado se encontra a empresa, do que esta precisa e como prosseguir. Disponibiliza uma produção de relatórios (tabelas/gráficos) quase automática permitindo assim uma visualização rápida de como se encontra a empresa. Os mesmos podem ser analisados numa funcionalidade denominada “*Company Snapshot*”. É possível consultar a evolução da empresa ao longo do ano, pois todo o tipo de pagamentos, vendas, entre outros, são categorizados e inseridos no produto, facilitando a sua pesquisa a qualquer momento.

Recorrendo a uma versão *trial* foi possível concluir que a ferramenta tem uma boa funcionalidade de relatórios. A mesma disponibiliza a criação automática de relatórios (tabelas/gráficos), o utilizador apenas tem de indicar qual a análise que deseja fazer (vendas, clientes, entre outros). Tem ainda uma funcionalidade bastante interessante: comparação entre o que realmente se gastou e o esperado gastar. Desta forma o utilizador pode ter noção do caminho a seguir pois tem uma previsão do que poderá vir a gastar.

Quanto ao licenciamento este é comercial e depende do número de utilizadores.

2.1.5 RSoft – xSoft ERP RSOFT

RSoft é uma empresa que foi criada em 1992, com o intuito de desenvolver soluções para o controlo de produção e comercialização de equipamentos informáticos. Só mais tarde iniciou o desenvolvimento em ERP. Ao analisar todas as áreas fundamentais do produto, foi

possível concluir que este tem uma boa base de ferramentas de contabilidade. Existe uma implementação de contabilidade geral e analítica, sendo possível dividir os dados por centros de custo, extraíndo destes balancetes e análises financeiras.

Existe uma falta de visualização gráfica dos dados. Podem ser consultadas tabelas (balancetes/extratos), mas não existe nenhuma funcionalidade que produza relatórios em que a informação se encontre detalhada conforme as escolhas do utilizador. Não existe qualquer tipo de informação a nível de licença do produto, tendo que existir contacto com a empresa para conseguir ter essa informação.

2.1.6 SAP ERP



SAP - "*Systems, Applications, and Products in Data Processing*" - é uma empresa multinacional líder de mercado em software de aplicação empresarial. Tem como objetivo facilitar a gestão das empresas, seja qual for a sua dimensão. A empresa dispõe de várias soluções a nível de:

- *Business Process* – utilização de estratégias que estimulem a produtividade, de modo a que exista uma redução de custos e uma melhoria de desempenho;
- *Business Analytics* – acesso em tempo real a toda a informação, de modo a permitir um acesso em qualquer momento aos dados processados, facilitando a tomada de decisões;
- *Technology* – Computação em tempo real para analisar grandes quantidades de dados. As tecnologias são utilizadas para permitir que essa análise seja feita com processos sincronizados e com a informação de todo o negócio, assegurando a integridade da mesma através de múltiplos *devices* e *cloud*.

A análise foi focada em duas ferramentas específicas: SAP ERP *Financials* e *Reporting and Analysis*, que pertencem ao *Business Process* e ao *Business Analytics*, respetivamente. No caso do SAP *Financials*, podemos agrupar os dados por centro de custo, projeto, produto ou rendimentos. Utilizando então a possível integração com *Reporting and Analysis*, essa informação passa a existir em relatórios para uma maior facilidade de análise.

Existem várias formas de adquirir licenças, nomeadamente com "*License Packages*" às quais é depois possível acrescentar utilizadores.

2.1.7 Gestware Administrativa



A Gestware é uma empresa nacional que se dedica exclusivamente ao desenvolvimento de *software* de gestão. A empresa criou dois produtos: Gestware Comercial e Gestware Administrativa.

Nesta análise focamo-nos na Gestware Administrativa, pois desta fazem parte as funcionalidades que mais têm a ver com a plataforma em desenvolvimento: Contabilidade Geral, Contabilidade Analítica, Pessoal e Balanço. Das funcionalidades existentes no produto, nenhuma delas é de visualização de dados. É possível agrupar os movimentos por centro de custo ou geral, mas existe falta de uma componente de pesquisa / visualização gráfica dos dados. Consequentemente, não existe a elaboração de relatórios, apenas sendo

disponibilizados balancetes ou extratos de movimentos. Existe uma versão *trial* para avaliação do produto.

2.1.8 Comparação

Apresentamos agora uma análise comparativa de todas as soluções referidas na secção anterior.

2.1.8.1. Vetores de Comparação

A comparação será realizada de acordo com os seguintes critérios:

- Visualização dos dados (gráficos e/ou tabelas):

Os dados contabilísticos são introduzidos pelos utilizadores dos produtos ou importados das bases de dados existentes, mas fazem falta sistemas de visualização nos mesmos que permitam ao utilizador observar de forma interativa a evolução da sua empresa. Na maioria das aplicações analisadas, as tabelas são disponibilizadas, faltando a parte gráfica.

- Produção de relatórios e respetivo download e/ou envio por email:

Em qualquer relatório que seja produzido é importante que existam opções como *download/email*, de modo a permitir consultas a documentos em qualquer local/hora.

- Sistema de notificações:

Um sistema de tem como objetivo avisar o utilizador de critérios que este esteja interessado e possivelmente tenha definido. Por exemplo, se um proveito se desviou muito do KPI definido, o utilizador poderá querer ser notificado dessa situação.

- Pesquisa elaborada por parâmetros:

Útil para conseguir criar gráficos/relatórios detalhados. A situação ideal seria o utilizador poder realizar pesquisas mais aprofundadas do que por centros de custo. Por exemplo, a WIT organiza a contabilidade por unidades, clientes e projetos, formando a junção dos três um centro de custo. Assim, deveria ser possível pesquisar apenas por unidades, clientes, projectos ou a combinação de dois ou três parâmetros.

2.1.8.2. Tabela de Comparação

A seguinte tabela resume os dados recolhidos no presente capítulo, indicando quais as soluções que cumprem os objetivos desejados segundo os critérios apresentados.

	Visualização	Produção de relatórios	Sistema de Notificações	Pesquisa elaborada
--	--------------	------------------------	-------------------------	--------------------

PHC	Tabelas	✗	✗	✗
Primavera	Gráficos/ Tabelas	✓	✗	✗
CentralGest	Tabelas	✓	✗	✗
Intuit	Gráficos/ Tabelas	✓	✗	✗
RCSOFT	Tabelas	✓	✗	✗
SAP	Gráficos/ Tabelas	✓	✓	✓
Gestware	Tabelas	✗	✗	✗

Tabela 2 – Comparação de soluções

✗ - Não contém este critério. ✓ - Contém este critério.

2.1.8.3. Análise Comparativa

A Tabela 2 indica as principais diferenças entre as plataformas analisadas.

O SAP cumpre um dos critérios cruciais na comparação entre as plataformas, de que as restantes carecem – a possibilidade de realizar uma pesquisa mais avançada, que permita obter resultados pormenorizados a nível de dados da empresa, para além de ter um sistema de notificações.

Através da combinação das funcionalidades de *Financials, Operations Analytics, Reporting and Analysis e Integrated Analytics and Reporting* o SAP permite conciliar todo o tipo de informação, sendo esta apresentada num relatório ou num *dashboard*. Os relatórios criados pelo produto são dos mais variados tipos, permitindo a visualização de todos os dados, graficamente ou não.

Para além do SAP, os dois produtos que mais funcionalidades têm são o Primavera e o Intuit. O Primavera tem análise de contabilidade e visualização dos dados através de um “Monitor de Negócios”, no caso do Intuit trata-se do “*Company Snapshot*”. Em ambos é possível observar uma serie de painéis já definidos. É neste ponto que se marca a diferença – a falta da possibilidade de combinar diferentes tipos de pesquisa faz com que a visualização dos dados por parte do utilizador seja limitada.

Tanto o RCSOFT como o CentralGest ficam aquém do Primavera e do Intuit por apenas permitem uma visualização de tabelas, não existindo qualquer tipo de componente gráfica.

Abordando agora os produtos que menos funcionalidades têm, de acordo com as definidas em 2.3.1, observamos que o PHC e o Gestware se encontram empatados na classificação. Tal como qualquer dos restantes produtos indicados, estes são ótimos produtos, mas apenas para o armazenamento de dados. Ambos permitem que os valores sejam guardados por centros de custo/gerais, sendo possível a sua visualização através de balancetes/extratos. No entanto, não permitem realizar gráficos pormenorizados, como por exemplo:

- Análise de custos ou proveitos de um cliente no âmbito de um determinado projeto;
- *Breakdown* de custos. Utilizando o caso da empresa cliente do atual desenvolvimento, tendo em conta que a contabilidade se encontra estruturada como: unidade, cliente e projeto para um único centro de custo, não seria possível, por exemplo.
- Contribuição de todos os projetos, clientes, unidades ou centros de custo para o total da empresa.

A combinação de pesquisas é também importante para a produção de relatórios e para a quantidade de dados que estes podem criar num único ficheiro. A maioria dos produtos em análise permite a produção de relatórios como balancetes ou extratos, podendo estes ser exportados para pdf ou excel.

2.2 Produtos de *Business Intelligence*

Esta secção apresenta uma análise das principais soluções comerciais existentes na área de *reporting* e *dashboards* usando as tecnologias de *Business Intelligence*.

- ZoHo Reports;
- Pentaho;
- DMWorks - DMReports;
- LogiXML;
- Dundas;
- iDashboard;
- InetSoft StyleScope;
- InetSoft Visualization Free;
- Birst;
- Arcplan Enterprise;
- Quadbase.

2.2.1 Zoho Reports

O Zoho é uma plataforma *online* que junta produtividade, negócio e colaboração de modo a permitir gerir todo o negócio de forma rápida e eficaz. Trata-se de uma ferramenta de BI que permite analisar facilmente os dados de pequenos e médios negócios, criando relatórios e *dashboards* de modo a melhorar a tomada de decisão de todos os utilizadores. Para conseguir alcançar os pontos referidos a informação a utilizar pode ser carregada de CSV, XLS, entre outros.

No decorrer do estudo do produto foram realizadas algumas experimentações de modo a adquirir algum conhecimento. Desta forma foi possível concluir que o produto permite ao utilizador inserir dados de tipos: CSV e XLS, entre outros, criar relatórios automaticamente e ainda interagir com os mesmos, sendo visíveis legendas e dados detalhados ao passar o rato sobre os gráficos. No que toca a usabilidade também é bastante bom pois o utilizador consegue automaticamente perceber de que informação se trata interagindo com os gráficos. No entanto, para grandes empresas, os dados a gerir seriam de grande dimensão, sendo assim mais complicado conseguir gerir a mesma.

Quanto à licença, o Zoho Reports tem vários tipos de planos, existe um gratuito mas no entanto é bastante limitado – a sua utilização é restrita a 2 utilizadores e o número de dados a analisar é limitado, existindo no máximo 5 *queries* à base de dados. Depois disto existem vários planos cujo preço vai aumentando consoante o número de funcionalidades disponíveis.

2.2.2 Pentaho



Pentaho é um produto de *Business Intelligence* a que se pode aceder na máquina pessoal do utilizador. O mesmo permite criar relatórios, *dashboards* e análises a partir de um conjunto de dados que se podem encontrar em ficheiros do tipo CSV ou TXT, numa base de dados

ou ainda em dados já armazenados no próprio programa. Existe um conjunto de aplicações *desktop* denominadas de “*Design Tools*” que permite ao utilizador criar relatórios com mais opções, nomeadamente, realizar o seu próprio *design*.

Ao ser efetuada uma análise e respetiva experimentação das funcionalidades do produto foi possível retirar algumas conclusões sobre o mesmo, nomeadamente a facilidade com que se pode produzir relatórios, análises ou *dashboards*. Qualquer uma das opções referidas é simples de usar, sendo toda a criação das funcionalidades realizada com um sistema de *drag and drop* de dados armazenados no produto, tornando muito rápida qualquer tarefa.

A nível de licença do produto, existem 3 soluções: *basic*, *professional* e *enterprise*. Qualquer uma delas tem uma subscrição anual. Existe, no entanto, uma versão *enterprise* gratuita por 30 dias para que se possa experimentar e conhecer o produto.

2.2.3 DMWorks – DMReports **DMWorks**

DMReports é uma plataforma de *Business Intelligence* que disponibiliza ao utilizador todas as funcionalidades necessárias para criar relatórios, tabelas interativas e *dashboards*, a partir de dados existentes numa base de dados.

Entre muitas outras funcionalidades existe a possibilidade de carregar dados de ficheiros excel, ligar a bases de dados já existentes e ainda integrar com sistemas ERP. Todos os relatórios criados podem ser partilhados e exportados para PDF. A opção de partilha pertence ao módulo web denominado de “*Reports Sharing*”, que foi usado para avaliar a qualidade dos relatórios criados. Existe ainda a possibilidade de personalizar os relatórios criando os próprios *templates* através do módulo: *DM Reports Designer*.

A licença do produto é paga e constituída por módulos.

2.2.4 LogiXML LogiXML™

LogiXML é um produto que permite criar aplicações web de BI de acordo com as características de cada empresa, nomeadamente o seu tamanho. Através da aplicação *desktop*: Logi Info Studio, podemos criar as nossas próprias aplicações com relatórios e *dashboards*, entre outros. Os dados a usar podem ser obtidos através de ligações a bases de dados ou ficheiros do tipo csv. Uma vez terminada a aplicação poderá ser acedida no servidor em que se encontra alojada.

No que toca à licença do produto, existe uma versão *trial* de 15 dias para que os utilizadores o possam experimentar. Após esse tempo o utilizador terá de recorrer a uma subscrição anual se quiser continuar com o produto.

2.2.5 Dundas Dundas

Dundas *dashboard* é um *software* de *Business Intelligence* que permite ao utilizador criar *dashboards*, alertas e ainda retirar conclusões dos dados que pode observar através de interfaces com boa usabilidade. O produto permite ligações a bases de dados como *MS SQL*

Server, *OLAP* entre outros, permitindo ainda o carregamento de dados através de *MS Excel* e *CSV*.

Existe um ponto que pode chamar à atenção de um utilizador frequente: a existência de *Mashup*, onde o utilizador pode facilmente ter os seus gráficos favoritos e acessíveis através de um único clique. Temos ainda a possibilidade de criar anotações sendo assim possível levantar “discussões” sobre assuntos de interesse comum e relacionados com as conclusões que se consiga retirar dos gráficos analisados. Contudo, o *software* foca-se apenas em *dashboards* e nas opções que podemos fazer com os mesmos. Não existe uma funcionalidade de criar relatórios, algo que é útil para funcionários que queiram dados específicos e impressos. Existe uma licença para o produto, não sendo especificada a forma como esta funciona (se é anual ou não).

2.2.6 iDashboards Dashboards

Driving Business Decisions

iDashboards é uma empresa com 5 produtos na área de BI: *Enterprise Edition*, *X Platform*, *In Cloud*, *Live LCD Edition* e por último, *Mobile*. Ao analisar resumidamente o que é cada um, optou-se por estudar a *X Platform*, pois qualquer uma das restantes ou tem parte das funcionalidades desta (como é o caso do *Enterprise Edition*) ou não se enquadra no presente estudo (como por exemplo o produto *Mobile*). *X Platform* é uma aplicação em *Flash* visualizada no navegador de cada utilizador que inclui alertas, *dashboards* e relatórios. Desta forma o utilizador pode ser notificado no momento em que um determinado limiar foi atingido, existindo assim análise em tempo real.

A ferramenta tem inúmeras funcionalidades nomeadamente: criação de *dashboards* e relatórios, filtragem dos dados visualizados, análises em tempo real de dados analíticos, importação/exportação de dados através de *spreadsheets*.

A licença é por número de utilizadores, não sendo fornecida a informação do preço.

2.2.7 InetSoft Style Scope

open standards innovation

Style Scope é uma aplicação WEB de BI gratuita que permite ao utilizador criar, partilhar e monitorizar *dashboards*. Pode ser ligada a bases de dados já existentes ou *spreadsheets*.

A versão gratuita tem algumas limitações, não podendo ser usada para efeitos comerciais, nem ter ligações a várias bases de dados em simultâneo ou acessos a vários tipos de dados como XML, *Web Services* e *OLAP*. Para isto o utilizador tem de pagar a licença do *Style Scope*. No entanto para uma aplicação gratuita contém bastantes funcionalidades de interesse do ponto de vista do utilizador. Pode criar *dashboards*, exportar os mesmos e ainda análise de dados, mas não produz qualquer tipo de relatórios.

2.2.8 InetSoft Visualization Free

open standards innovation

Visualization Free é uma aplicação web para visualização de dados não necessitando de instalação. Todos os dados existentes em *spreadsheets* podem ser carregados para a

aplicação e com a funcionalidade de *drag and drop* criar a visualização multidimensional desejada.

O utilizador pode criar o seu próprio *dashboard* sem ter de instalar qualquer ferramenta. Para isso basta aceder ao endereço, realizar o *login*, escolher dados já existentes ou carregar os seus próprios dados e, por último, arrastar os dados para criar os elementos desejados no *dashboard*, tornando assim a ferramenta prática e com a qual o utilizador não tem de ter qualquer preocupação. No entanto tem algumas limitações por ser gratuita.

2.2.9 Birst

Birst é uma nova abordagem de BI que se foca na análise dos dados contabilísticos de cada negócio através da visualização de dados de diferentes sistemas, de modo a ajudar o utilizador a tomar decisões. Para isso o Birst contém soluções como relatórios, *dashboards* e *ad-hoc queries*, tudo isto no mesmo local.

Ao analisar a lista de funcionalidades do produto podemos concluir que contém todas as funcionalidades importantes num *software* de BI: relatórios, *dashboards*, alertas e ainda o uso de soluções OLAP. No entanto falta um sistema de segurança como, por exemplo, o uso de *https*, pois apenas existe um controlo de permissões.

O produto tem licenças para *single* ou *multi developer*, não estando disponíveis os valores associados.

2.2.10 Arcplan Enterprise



Ferramenta de BI que permite ao utilizador criar relatórios, *dashboards*, análises e previsões de resultados a partir dos dados, juntando assim todos estes pontos numa única ferramenta.

Existe a possibilidade de integrar com produtos como SAP e criar *ad hoc queries* de modo a que o utilizador possa especificar o que necessita. O único ponto onde falha é na usabilidade: falta um sistema de atalhos ou criação de *dashboards*/relatórios favoritos como existe no Dundas, de modo a que o utilizador não necessite de estar sempre a fazer/criar determinadas pesquisas. Não foi possível obter qualquer tipo de informação a nível de licenciamento.

2.2.11 Quadbase

Quadbase tem um conjunto de 4 produtos: *EspressChart*, *EspressReport*, *EspressDashboard* e por último, *EspressReport ES*. Neste caso estudaremos o *EspressReport ES* pois é o que contém o conjunto de todas as funcionalidades principais: gráficos, relatórios e *dashboards*.

Existem inúmeras funcionalidades como por importação de dados das bases de dados, excel, XML ou até mesmo fazer as suas próprias *queries*. Exportar os relatórios resultantes para excel, pdf, entre outros e ainda um sistema de alertas que permite avisar o utilizador de certos critérios que este tenha configurado.

O único ponto onde a aplicação falha é na usabilidade. Um sistema que permitisse gerir *dashboards*/relatórios favoritos seria uma forma de o utilizador poder guardar as suas pesquisas.

Existem dois tipos de licenciamento: *Development License* e *Server/Deployment License*. O valor de cada uma não se encontra disponível.

2.2.12 Comparação

Apresentamos agora uma comparação das soluções comerciais de *business intelligence*.

2.2.12.1 Vetores de Comparação

O sistema de comparação será classificado de acordo com os seguintes critérios:

- Criação de relatórios:

A criação de relatórios para posteriores análises é um ponto muito importante para permitir ao utilizador retirar conclusões de todos os dados que considere relevantes assim como utilizar essa informação onde necessário.

- Criação de *dashboards*:

Os *dashboards* permitem visualizar os dados de uma forma dinâmica e atrativa, existindo a possibilidade de comparação de vários dados em simultâneo.

- Exportar relatórios e/ou *dashboards*:

Com a funcionalidade de exportar o utilizador pode criar documentos em qualquer altura e consultar os mesmos sem dependência de meios eletrónicos.

- Ligação a bases de dados/*software* já existentes:

A forma com que os dados passam a existir no produto é muito importante. É necessário que existam funcionalidades suficientemente gerais para que o utilizador não necessite de introduzir novamente os dados. Podemos referir como exemplo dois dos produtos mencionados anteriormente: o Dundas ou o Arcplan que permitem configurar ligações ao SAP. Desta forma as ligações a bases de dados/*software* já existentes são pontos muito importantes em qualquer produto de BI para facilitar o processo de importação de dados.

- Usabilidade:

Um bom *interface* não significa apenas que este seja atrativo, mas também intuitivo. Como exemplo temos um sistema de atalhos ou a possibilidade de criar relatórios/*dashboards* por defeito.

2.2.12.2. Tabela de Comparação

A seguinte tabela resume os dados recolhidos, indicando quais as soluções comerciais de BI que cumprem os objetivos desejados segundo os critérios apresentados.

	Relatórios	Dashboards	Exportar	Ligação a BD	Usabilidade
Zoho	✔	✔	✔	✗	✗
Pentaho	✔	✔	✔	✗	✗
DMWorks	✔	✔	✔	✔	✔
LogiXML	✔	✔	✔	✗	✗
iDashboards	✔	✔	✔	✗	✔
Style Scope	✗	✔	✔	✗	✗
Visualization Free	✗	✔	✔	✗	✗
Dundas	✔	✔	✔	✔	✔
ArcPlan	✔	✔	✔	✔	✗
Birst	✔	✔	✔	✗	✗
Quadbase	✔	✔	✔	✗	✗

Tabela 3 – Comparação de soluções

✗ - Não contém este critério. ✔ - Contém este critério. ✗✔ - Falha em alguns pontos.

2.2.12.3. Análise Comparativa

Analisando os critérios indicados e os produtos apresentados podemos concluir através da Tabela 3 que existem dois dos produtos que não apresentam relatórios e os restantes contêm pontos incompletos, ou seja, apenas têm algumas das possíveis funcionalidades. Isto verifica-se em dois pontos: ligações a BD/*software* existentes e usabilidade.

Começando pelo Zoho, podemos observar através das funcionalidades apresentadas que existe uma boa usabilidade, através de um *interface* rápido e perceptível. No entanto, faz falta um sistema de criação de relatórios/gráficos favoritos ou até mesmo um sistema de atalhos. No que toca às ligações a bases de dados, este pode ligar-se a algumas já existentes, mas falta a integração com produtos de ERP, como por exemplo o SAP. Para além disso temos a possibilidade de partilhar os dados com terceiros e de criar relatórios similares a algum já existente.

Analisando agora o Pentaho, este peca quer na usabilidade quer nas ligações/integrações com bases de dados/*software* existentes. Tem uma criação automática de relatórios/*dashboards* sendo todos eles criados automaticamente com um sistema de *drag and drop*. No entanto, faz falta um sistema de atalhos ou relatórios/gráficos favoritos para o utilizador não ter de realizar constantemente as mesmas pesquisas. Tal como o Zoho, seria bom para o utilizador um sistema de integração, por exemplo, com o SAP, isto porque muitos utilizadores o usam e facilitaria o tratamento de dados.

O DMWorks tem um ponto muito importante e interessante numa ferramenta destas, tendo em conta a área em que o estágio se insere. Permite a integração do produto com sistemas ERP, o que facilita a importação dos dados em que o utilizador se encontra interessado, tornando mais direta a criação de *dashboards*, entre outros. Para além disto, os relatórios criados podem ser parametrizados e os *templates* podem ser definidos pelo utilizador. Tal como acontece no Zoho, os dados podem ser partilhados através de uma plataforma web.

Por último temos o LogiXML, onde existe uma funcionalidade importante para empresas que preferem criar as suas próprias aplicações de BI, pois há a possibilidade de criar módulos de BI com todas as opções pretendidas pelo utilizador. Isto tudo é criado de forma simples e prática, é tudo à base de *drag and drop* de menus e especificação de cada *action* respetivamente. O acesso à plataforma disponibilizada pode ser efetuado em qualquer momento quer seja por computador ou por dispositivos móveis. No entanto, faz falta um sistema de gráficos/relatórios favoritos ou até mesmo um sistema de atalhos e ainda integração com produtos ERP.

Tanto o Style Scope como o Visualization Free são gratuitos e pertencem à mesma empresa: Inetsoft. Qualquer um destes se foca mais em *dashboards*, não existindo a funcionalidade de criação relatórios. O Style Scope corre na máquina de cada utilizador enquanto o Visualization Free é uma aplicação Web que não necessita de qualquer tipo de instalação. É apenas necessário carregar os dados e estes ficam disponíveis para criar o *dashboard* bastando o utilizador registar-se no produto.

O Quadbase e o Birst são produtos semelhantes. Ambos permitem criar *dashboards* e relatórios, contêm alertas e vários tipos de segurança. Tal como os produtos anteriormente indicados, têm dois pontos fracos: a integração com produtos ERP e a usabilidade. A usabilidade é um ponto muito importante para que um utilizador frequente não se canse de realizar constantemente as mesmas pesquisas e tenha gosto de usar o produto. Tal como estes produtos temos o Arcplan, mas neste caso é possível integrar com, por exemplo, o SAP. No entanto, tal como os restantes, falha na questão da usabilidade.

Quanto ao iDashboard tem apenas um ponto fraco – integração com produtos ERP. Apesar de ter ligações a bases de dados externas, falta a possibilidade de integrar com produtos específicos de modo a facilitar o tratamento de dados.

Por fim temos o Dundas, um produto que contém todos os vetores de comparação apresentados nesta análise. Permite criar relatórios e *dashboards*, tem atalhos: “Dundas Mashups” e permite ligações ao SAP.

2.3 Considerações Finais

A plataforma desenvolvida terá que satisfazer alguns requisitos fundamentais, nomeadamente: integração com o SSO – forma de controlar os múltiplos acessos existentes a diferentes, mas relacionadas, aplicações web – integração com o PHC e flexibilidade com o processo de contabilidade da WIT.

Integração com o SSO: a solução a apresentar necessita de ter integração com o SSO da WIT, o que não é suportado por grande parte das soluções analisadas.

Integração com o PHC: a solução a apresentar necessita de ter uma integração com o PHC, o que não é suportado pelos produtos analisados, pois apenas dão para integrar com ERP específicos (como por exemplo o SAP).

Flexibilidade com o processo de contabilidade da WIT: O processo de contabilidade da WIT é muito específico, existindo uma estrutura própria. Todas as funcionalidades desenvolvidas têm cálculos específicos, *queries* à base de dados específicas, contas próprias do SNC [15], entre outros. Todas estas particularidades levam a que fosse difícil utilizar qualquer um dos produtos analisados.

A plataforma a desenvolver não pretende ser tão completa como as soluções analisadas, mas apenas suportar o processo de contabilidade da WIT como está definido, e permitir integração com os sistemas existentes, como foi apresentado acima.

3 Requisitos

Este capítulo apresenta uma descrição de todos os requisitos da plataforma desenvolvida durante o estágio. Os requisitos indicados foram identificados com base nas necessidades do cliente e nas falhas existentes em cada aplicação analisada em 2.2.

Na primeira secção são apresentados os processos de levantamento e análise de requisitos, seguidos de uma lista de requisitos externos do *web service* e do *php service*. De seguida temos a secção com todos os requisitos funcionais, que se encontram divididos por módulos: sessão, contabilidade analítica, contabilidade geral, recursos humanos, *software quality* e por último, o *back office*. Por fim, temos os requisitos de *design*: os *mockups*.

3.1 Levantamento de Requisitos

O processo de levantamento de requisitos foi realizado na WTT em conjunto com os departamentos de contabilidade, recursos humanos e qualidade de *software* da empresa. Após sucessivas reuniões e análises de documentos criados por esses departamentos foi iniciada a elaboração do documento de requisitos. Uma vez concluído esse documento, foram marcadas reuniões com cada um dos departamentos de modo a perceber se os requisitos apresentados eram realmente os desejados. Consequentemente, alguns requisitos foram excluídos e outros acrescentados.

Ao longo de todo o processo de desenvolvimento foram sendo realizadas reuniões de modo a perceber se o projeto se encontrava a decorrer de acordo com o esperado. No decorrer dessas reuniões foram surgindo novos requisitos, que foram acrescentados ao documento.

3.2 Requisitos Externos

A plataforma necessita de dois projetos externos para correto funcionamento, são estes: *Web service PHC* e *PHP Service*. Cada um destes serviços tem os seus próprios requisitos, que são apresentados na Tabela 4 e na Tabela 5. As tabelas apresentadas nesta secção são resumos dos requisitos externos apresentados em detalhe no capítulo 2 do Anexo C – Especificação de requisitos.

Web Service

Na tabela abaixo são apresentados os requisitos externos do *web service* desenvolvido para importação dos dados. São apresentadas todas as mensagens que permitem a comunicação entre a plataforma de contabilidade e o *web service*, estas incluem as unidades, clientes, projetos, centros de custo, movimentos (gerais ou não), os valores dos recursos humanos e por último os da faturação.

ID	Nome	ID	Nome
RE-WS-01	Business unit request message	RE-WS-09	Movement request message
RE-WS-02	Business unit response message	RE-WS-10	Movement response message

RE-WS-03	Customer request message	RE-WS-11	General movement request message
RE-WS-04	Customer response message	RE-WS-12	General movement response message
RE-WS-05	Project request message	RE-WS-13	Human resource value request message
RE-WS-06	Project response message	RE-WS-14	Human resource value response message
RE-WS-07	Cost center request message	RE-WS-15	Invoice request message
RE-WS-08	Cost center response message	RE-WS-16	Invoice response message

Tabela 4 – Requisitos do *web service*.

PHP Service

Na tabela abaixo são apresentados os requisitos externos do *PHP Service* desenvolvido para criação de gráficos. São apresentados todos os gráficos que têm de ser realizados pelo mesmo para cobrir a funcionalidade de *download* de relatórios em PDF. Os mesmos encontram-se divididos em gráficos de barras (*accounting*), de barras *stacked*, de linhas (*RevenuesKPI*) e por fim circular (*contribution*).

ID	Nome
RE-PS-01	Accounting
RE-PS-02	Stacked
RE-PS-03	RevenuesKPI
RE-PS-04	Contribution

Tabela 5 – Requisitos do *php service*.

3.3 Requisitos Funcionais

Nesta secção são indicados os requisitos funcionais da plataforma em desenvolvimento. As tabelas apresentadas nesta secção são resumos dos requisitos externos apresentados em detalhe no capítulo 3 do Anexo C – Especificação de requisitos.

Sessão

O sistema deve permitir que o utilizador realize *login/logout* e que ocorra *timeout* da sessão. O utilizador que realize login tem de estar AUTORIZADO a tal, não sendo possível a entrada de utilizadores que não tenham permissões, as quais são configuráveis no *back office* tal como indicado nos requisitos funcionais do *back office*. A sessão deve ainda expirar ao fim do tempo definido (30 minutos).

ID	Nome
RF-SE-01	Iniciar Sessão/Login
RF-SE-02	Terminar sessão/Logout
RF-SE-03	Sessão <i>Timeout</i>

Tabela 6 – Requisitos de sessão.

Contabilidade Analítica

No módulo de contabilidade é possível visualizar gráficos de: proveitos custos e resultados, vendas, custos detalhados, contribuições de cada unidade/cliente/projeto,

pesquisas compostas com diferentes tipos de filtros e ainda a criação de relatórios através dos gráficos visualizados ou num menu á parte. Na tabela abaixo podemos analisar um breve resumo dos requisitos deste módulo.

ID	Nome	ID	Nome
RF-CA-01	Proveitos, custos e resultados	RF-CA-07	Pesquisa composta
RF-CA-02	Vendas: nacionais vs. Internacionais	RF-CA-08	Filtro de pesquisa
RF-CA-03	Proveitos vs. KPI	RF-CA-09	Módulo de criação de relatórios
RF-CA-04	Custos detalhados	RF-CA-10	Download de relatórios
RF-CA-05	Contribuição – Proveitos, custos e/ou resultados	RF-CA-11	Envio por <i>email</i> de relatórios
RF-CA-06	Contribuição detalhada de custos		

Tabela 7 – Requisitos de contabilidade analítica.

Contabilidade Geral

No módulo de contabilidade geral podemos visualizar tabelas de demonstração de resultados, balanço, fornecimentos e serviços externos, gastos com o pessoal, tesouraria e *tableau de bord*. Existe ainda um módulo denominado faturação que permite analisar a faturação detalhada por cliente e serviços. Na tabela abaixo é apresentado um breve resumo dos requisitos deste módulo.

ID	Nome	ID	Nome
RF-CG-01	Fornecimentos e serviços externos	RF-CG-06	Faturação – por serviços
RF-CG-02	Gastos com o Pessoal	RF-CG-07	Demonstração de resultados
RF-CG-03	Faturação	RF-CG-08	Balanço
RF-CG-04	Faturação – por clientes	RF-CG-09	Tesouraria
RF-CG-05	Faturação – top clientes	RF-CG-10	Tableau de Bord

Tabela 8 – Requisitos de contabilidade geral.

Recursos Humanos

O módulo de recursos humanos deve permitir ao utilizador visualizar tabelas (recursos humanos e QREN) pedidas pelo mesmo, estas são fornecidas pelo departamento de RH e que serão preenchidas com os dados inseridos no *back office* e os respetivos gráficos: número de trabalhadores e grau académico.

ID	Nome
RF-RH-01	Recursos Humanos
RF-RH-02	Número de trabalhadores
RF-RH-03	Grau académico
RF-RH-04	QREN
RF-RH-05	Reprovações

Tabela 9 – Requisitos do módulo de recursos humanos.

Software Quality

O módulo de *software quality* deve permitir ao utilizador visualizar a tabela pedida pelo mesmo, esta é fornecida pelo departamento de SWQ e que será preenchida com os dados inseridos no *back office* e os respetivos gráficos: pedidos de teste, *bugs* reportados e SWQ *Audits*.

ID	Nome
RF-SQ-01	<i>Software quality</i>
RF-SQ-02	Pedidos de Teste
RF-SQ-03	<i>Bugs</i> reportados vs. <i>Bugs</i> resolvidos
RF-SQ-04	SWQ <i>Audits</i>

Tabela 10 – Requisitos do módulo de software quality.

Back Office

O *back office* é responsável por gerir os produtos e permissões de cada utilizador. É aqui que são inseridos todos os dados que o PHC não disponibiliza, sendo estes: valores de KPI, orçamentos, tesouraria, recursos humanos e qualidade de *software*. Estes dados podem ser editados em qualquer altura, ou podem ser adicionados novos valores. A inserção é mensal, à exceção do KPI e dos orçamentos, que é anual. Existe ainda a componente de notificações que permite avisar o utilizador de componentes do seu interesse.

ID	Nome
RF-BO-01	Gestão de privilégios
RF-BO-02	Gestão de produtos
RF-BO-03	Carregar
RF-BO-04	Editar
RF-BO-05	Notificações – Criar
RF-BO-06	Notificações - Editar
RF-BO-07	Notificações - Apagar
RF-BO-08	Notificações - Listar

Tabela 11 – Requisitos do módulo back office.

3.4 Requisitos Não Funcionais

Requisitos não funcionais são todos os requisitos da aplicação relacionados com performance, escalabilidade, segurança, disponibilidade e usabilidade. Estes não são necessariamente pedidos pelo cliente. A informação apresentada de seguida encontra-se resumida. Todos os testes, resultados e otimizações feitas podem ser analisados detalhadamente no Anexo B – Requisitos não funcionais.

ID	Nome	Requisito
RNF-01	Integração com <i>Single-Sign-On</i>	Integração com o serviço SSO para o login do utilizador.
RNF-02	Integração com PHC	Integração com um <i>web service</i> de modo a realizar a importação dos dados necessários para a plataforma.
RNF-03	Encriptação	Utilização da encriptação: <i>Username Authentication with Symmetric key</i> , para importação de dados.

RNF-04	Atualização de Dados	O sistema deverá importar os dados do PHC com um intervalo de tempo de 1 dia.
RNF-05	Capacidade	O sistema deve suportar no mínimo 10 utilizadores em simultâneo.
RNF-06	Tempo de resposta	Todas as respostas do sistema devem ocorrer no MÁXIMO até aos 10 segundos, pois é um tempo razoável para o utilizador esperar.
RNF-07	Disponibilidade	A plataforma deverá manter-se acessível ao utilizador 99% do tempo.
RNF-08	Segurança	O sistema deve validar os parâmetros do URL de forma a verificar que não se trata de nenhum ataque de terceiros.
RNF-09	Segurança - HTTPS	O sistema TEM de usar HTTPS para proteger a transmissão de dados entre cliente-servidor.

Tabela 12 – Listagem de requisitos não funcionais da aplicação.

A integração com o SSO é necessária para questões de login do utilizador, de modo a que este tenha que realizar *login* uma única vez, seja qual for a plataforma da empresa em uso.

A integração com o PHC é necessária, pois todos os dados de contabilidade analítica se encontram armazenados nesse sistema, tendo assim que existir um *Web service* para aceder à base de dados do produto. O PHC é o produto de contabilidade usado na WIT.

A encriptação é necessária para evitar que alguém não autorizado aceda aos dados a partir da rede, o que não pode acontecer pois os dados são confidenciais. Ainda a nível de segurança, temos o uso de HTTPS para proteger os dados na camada cliente/servidor e a validação dos parâmetros do URL que segue para o servidor. Para isso tem de existir um filtro que protege a plataforma de ataques de *cross site scripting*.

A atualização dos dados é feita todas as semanas. Todas as segundas feiras, uma *thread* responsável por verificar o dia e a hora inicia a importação dos dados caso seja a altura correta.

De modo a tentar garantir que a plataforma se encontra protegida e sem qualquer tipo de problemas foi realizado um estudo de *benchmark* dos seguintes requisitos não funcionais: performance, escalabilidade, segurança e robustez. De seguida apresentamos um breve resumo da informação recolhida, todos os testes, resultados e otimizações feitas podem ser analisados detalhadamente no Anexo B – requisitos não funcionais.

3.4.1 Performance

Começando com a performance, esta foi estuda de três formas: primeiro foi analisada, otimizada e por fim foi feito o *benchmark* da mesma. Na análise de performance foi corrida a ferramenta *ySlow* [11] para obter a classificação da plataforma segundo as regras de *Steve Souders* [12]. As mesmas são classificadas segundo uma classificação de A a F, sendo A que satisfaz as regras e F que não satisfaz. Após correr a ferramenta obteve-se uma classificação de 74% contendo as seguintes regras mal classificadas:

Número de pedidos HTTP	Expires	GZip	Minificar JS e CSS	CSS no topo
F	F	F	C	F

De seguida, procedeu-se a uma análise das possíveis melhorias e do grau de otimização que podia ser alcançado. Desta forma concluiu-se que havia ainda muito espaço para otimizações. Posteriormente fizeram-se todas as otimizações possíveis obtendo-se no final uma classificação de 96%.

O *benchmark* foi feito utilizando duas ferramentas: *Httpperf* e *jMeter* [9], estas permitem realizar testes de sobrecarga a plataformas web utilizando parâmetros como: número de pedidos, número de ligações simultâneas e *timeout*. A diferença entre ambas é que o *Httpperf* quando ocorre um erro de *timeout* fecha a ligação, não sendo realizados mais pedidos. O *jMeter* já não tem este processo e realiza todos os pedidos até ao fim. Organizou-se os testes em dois grupos, sendo a diferença entre os mesmos as configurações usadas no *tomcat* e na base de dados. Por motivos de espaço apenas são colocados aqui os resultados mais relevantes e onde é mais evidente o comportamento da plataforma. Encontra-se na Figura 1 o resultado do último teste realizado utilizando o *jMeter*, do qual podemos concluir que a plataforma se encontra a corresponder aos requisitos definidos: suportar 10 utilizadores em simultâneo e ter um tempo de resposta abaixo dos 10 segundos. Neste caso o tempo de resposta atinge os 8 segundos com 200 ligações simultâneas e 60% de erros.

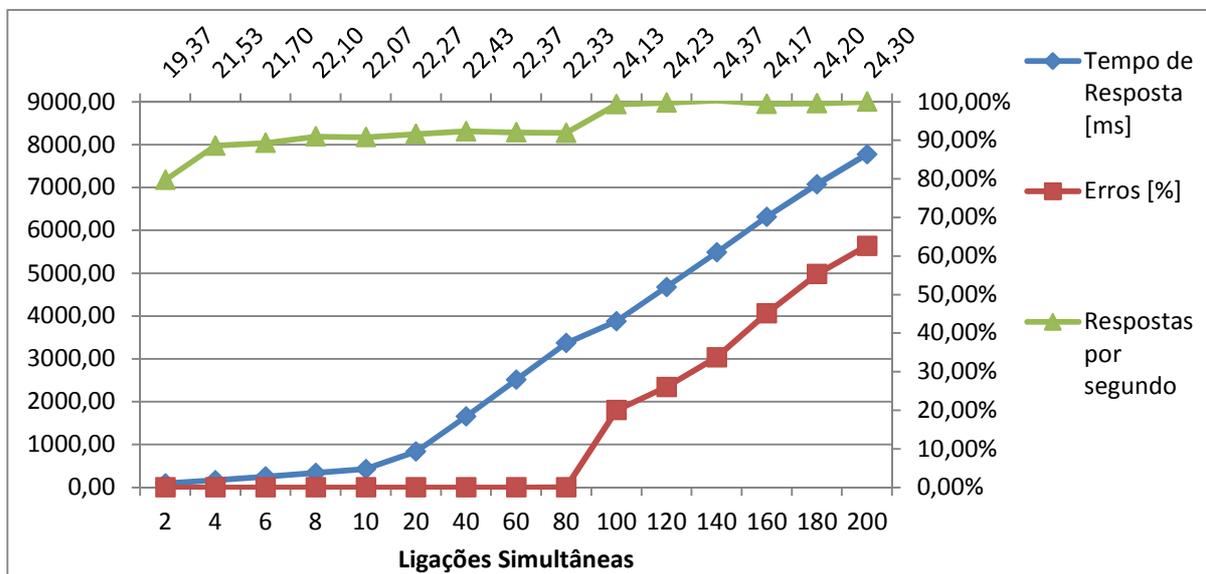


Figura 1 – Comportamento da ferramenta utilizando o *jMeter*.

3.4.2 Escalabilidade

A escalabilidade foi analisada recorrendo ao mesmo conjunto de testes que a performance. No entanto, estes foram feitos em duas páginas diferentes: uma mais leve que não deve sobrecarregar o servidor e outra que tem muitos acessos à base de dados e muito processamento de informação. Desta forma podemos analisar a sobrecarga do servidor consoante o aumento de número de ligações simultâneas acrescentando a informação do

CPU e da memória. Durante o decorrer dos testes foi observado que a utilização de uma página mais leve estava a sobrecarregar o CPU fazendo com que este atingisse rapidamente os 100%. Foram então analisadas as causas do mesmo através de ferramentas de *profile* [10] e realizadas otimizações. O mesmo processo foi seguido para a página com mais processamento do lado do servidor. No entanto, as otimizações realizadas serão explicadas mais à frente no capítulo de implementação e mais aprofundado no anexo. Nas figuras seguintes encontram-se os resultados dos testes para a página que mais sobrecarregava o servidor depois de todas as otimizações terem sido realizadas. É de salientar a existência de informação adicional disponível no anexo B.

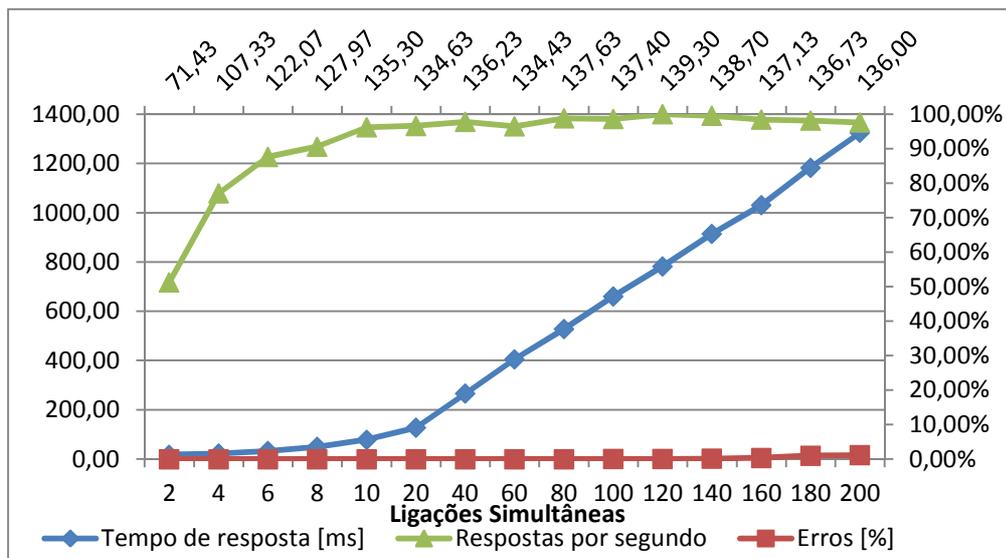


Figura 2 – Análise de performance.

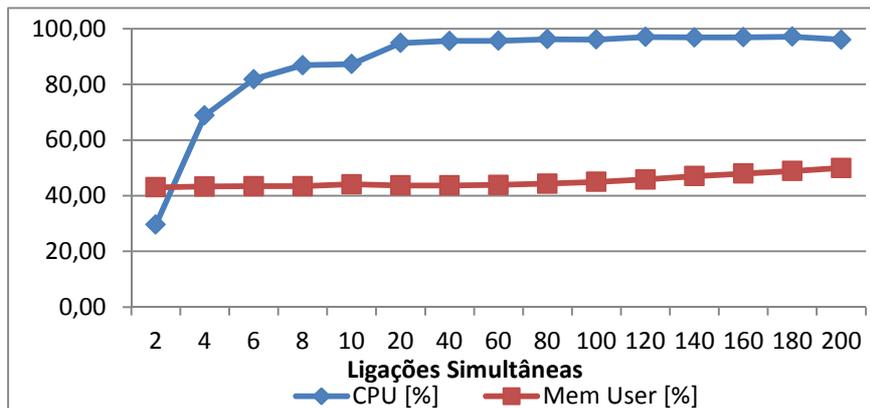


Figura 3 – Análise de CPU e memória.

Analisando os gráficos acima, na Figura 2 encontra-se o gráfico com a análise de performance, no eixo do lado esquerdo encontra-se o tempo de resposta em ms, no do lado direito a percentagem de erros. No eixo horizontal inferior temos o número de ligações simultâneas e no superior o número de respostas por segundo. No gráfico da Figura 3 temos a análise de CPU e memória, encontrando-se no eixo horizontal o número de ligações simultâneas e no vertical a percentagem de CPU e memória usada.

Podemos concluir que quer a nível de performance, quer de CPU e memória a plataforma se encontra a corresponder as expectativas e que melhorou muito em comparação com o gráfico da Figura 1.

3.4.3 Segurança

As questões de segurança são muito importantes na plataforma, pois esta contém dados confidenciais, sendo necessário proteger o acesso à mesma e encriptar os dados em trânsito. Serão tidas questões de segurança nos seguintes pontos: encriptação dos dados nas comunicações via *web service*, controlo de acesso à aplicação Web, intranet, autenticação utilizando SSO, segurança no acesso à base de dados, utilização de https, proteção contra ataques vindos exterior.

Encriptação

A importação de dados pelo *Web Service* é realizada usando a encriptação *Username Authentication with Symmetric Keys*. Esta é um mecanismo que protege a confidencialidade dos dados. A criptografia da *Symmetric Key*, depende de uma única chave partilhada que é usada para encriptar a mensagem. O cliente não necessita de nenhum certificado/chave, no entanto envia o *username* e *password* para autenticação, compartilhando previamente uma chave secreta com o servidor. Para poder ser criado este mecanismo, foi necessário a configuração dos *web services* para suportarem encriptação e criação de uma chave usando o *keytool* do java para configuração da chaves de acesso.

Controlo de Acesso

Na aplicação web apenas têm acesso os utilizadores que têm privilégios. Estes são configurados no *back office* existindo ainda diversos tipos de acessos. Tendo em conta que a WIT contém uma intranet com todas as suas plataformas, a aplicação encontra-se disponível apenas para a empresa, estando fechada para acessos do exterior. A única forma de aceder do exterior é através de uma ligação VPN, sendo para isso necessárias as credenciais do funcionário.

Autenticação utilizando SSO

Single Sign-On, é uma forma de controlar os múltiplos acessos existentes a diferentes, mas relacionadas, aplicações web. Utilizando esta propriedade, o utilizador realiza o *login* uma única vez, pelo que após essa etapa passa a ter acesso a todos os sistemas sem necessidade de uma nova autenticação. O mesmo acontece no *logout* da aplicação, utilizando *Single Sign-Off*, o utilizador ao sair da aplicação, realiza *logout* em todas as aplicações do sistema. Sendo assim, a aplicação utiliza a página de autenticação da intranet da WIT, para realizar o *login* na aplicação em desenvolvimento. Para que fosse possível colocar o SSO na plataforma, foi necessário proceder a configurações no Apache TomCat utilizado.

Segurança no acesso à Base de Dados

A base de dados da aplicação está protegida com *username* e *password*. Essa configuração encontra-se num ficheiro de configuração da aplicação em desenvolvimento.

Por motivos de segurança, a base de dados e a plataforma encontram-se na mesma máquina, evitando assim que os dados circulem na rede da empresa.

Utilização de HTTPS

Protocolo SSL que permite comunicação de forma “privada” evitando ataques como *Eavesdropping* [13]. Para confirmar que o SSL é realmente utilizado e que se encontra corretamente configurado foi utilizado a ferramenta *Wireshark* [14] que permite analisar os pacotes existentes entre cliente e servidor.

Foram ainda realizados testes para simular ataques à plataforma. Os resultados dos mesmos encontram-se no anexo B juntamente com um maior detalhe de todas as preocupações existentes a nível de segurança na plataforma. De salientar que uma das prevenções foi a criação de um filtro que permite filtrar o URL enviado para o servidor, com o objetivo de evitar ataques de *cross site scripting*.

3.4.4 Robustez

Um sistema robusto é um sistema que consegue lidar com erros sem que exista uma falha. Isto inclui dados errados, defeitos do próprio *software* e acontecimentos inesperados.

O *benchmark* de uma aplicação não é apenas à base de performance. Nos dias de hoje um sistema tem de garantir uma disponibilidade e segurança de 100%. Para isso é necessário focarmo-nos não só na performance mas também na segurança/confiança. Sendo assim o *dependability benchmark* analisa ambos, *performance* e *dependability*, o que inclui atributos como: disponibilidade, segurança, confiança, confidencialidade e manutenção. Tudo isto gira à volta de injeção de falhas de modo a simular um ambiente real onde um sistema pode por exemplo ir a baixo por falha de energia e a máquina onde o servidor está alojado desligar-se.

Foi elaborado um conjunto de testes para analisar a resposta do servidor a vários tipos de falhas. Desta forma podemos aferir a dependência existente a nível de intervenção humana para recuperação do sistema. Após a realização dos testes podemos retirar as seguintes conclusões: o sistema suporta um certo nível de falhas, nomeadamente falhas na base de dados - uma das otimizações realizadas é o uso de cache, o que permite que o sistema possa falhar sem ter qualquer interferência nos resultados no lado do cliente. Por outro lado temos falhas como por exemplo a máquina onde o servidor se encontra alojado desligar-se ou o servidor ir a baixo. Neste caso é necessária intervenção humana para recuperar o sistema. A única preocupação que se poderia ter seria arranjar uma forma de arrancar o servidor assim que a máquina inicia. Tudo isto é analisado com maior detalhe no anexo B – Requisitos não funcionais.

3.5 Mockups

Os *mockups* realizados podem ser consultados no Anexo C – Especificação de requisitos, capítulo 4.

4 Arquitetura

Este capítulo descreve a arquitetura da aplicação desenvolvida – Plataforma de contabilidade. Esta é uma aplicação pertencente ao departamento de contabilidade da WIT e permite visualizar todos os dados desse departamento através de *web services*, que interagem com a base de dados do PHC – produto de contabilidade utilizado pela empresa.

4.1 Arquitetura Geral

Nesta secção descrevemos a arquitetura da plataforma desenvolvida durante o estágio.

Optou-se por dividir a arquitetura em 3 componentes:

- Plataforma de contabilidade/*Back office*;
- Serviço de Gráficos;
- *Web Service* PHC.

Para cada componente, são indicadas quais as dependências existentes e os respetivos diagramas de arquitetura. Na figura seguinte, é apresentado um diagrama com todos os componentes, tornando mais fácil a compreensão da arquitetura da plataforma.

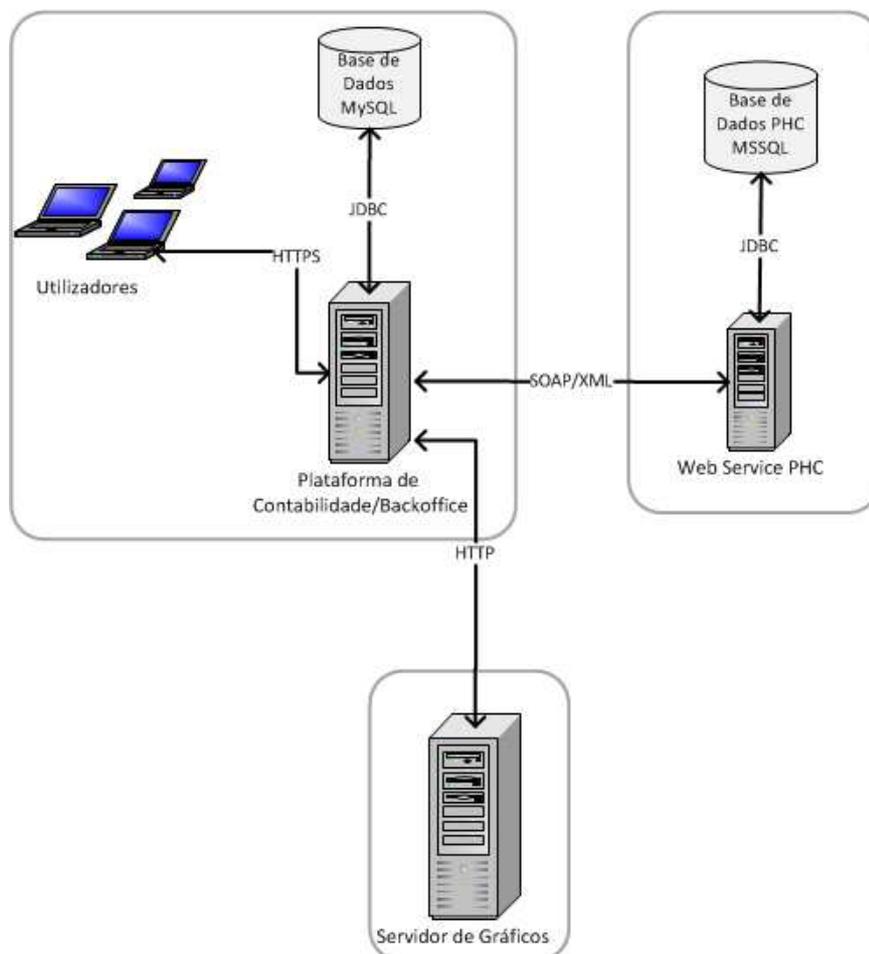


Figura 4 – Arquitetura da plataforma

O web service PHC é responsável por importar os dados necessários da base de dados do PHC, sendo assim obrigatória a existencia dessa ligação. Após a importação dos dados, estes são enviados para o servidor utilizando XML, onde grande parte dos dados necessários para o correto funcionamento dos requisitos apresentados são guardados. Os valores são constantemente atualizados no PHC, e todos os dias, à meia noite (hora programada) o servidor inicia a importação dos dados, acedendo ao *web service* e guardando-os na base de dados.

No canto superior esquerdo do diagrama temos o componente principal: Plataforma de contabilidade. Nesta encontra-se centralizado todo o projecto e as suas principais funcionalidades. Esta plataforma é responsável por gerar os gráficos, os relatórios de contabilidade geral/analítica, os módulos de recursos humanos e qualidade de *software*. Por outro lado, também contém o *back office*, que faz a gestão de todos os dados que têm de ser inseridos por terceiros, por não se encontrarem no PHC, e o respetivo envio para a base de dados. A plataforma depende de 2 serviços: do *web service* e do servidor de gráficos. Para o tratamento da informação, é ainda necessário a ligação à base de dados (mysql).

Por último, temos o servidor de gráficos em php, responsável por gerar todos os gráficos necessários para os relatórios em PDF. Para isso, necessita que o servidor lhe envie um pedido http, contendo os respetivos valores utilizados. Os gráficos são então gerados e enviados para o servidor na resposta.

Mais à frente encontra-se uma descrição detalhada de cada componente indicado no diagrama.

4.1.1 Plataforma de contabilidade/Back Office

O servidor é responsável por disponibilizar ao utilizador as seguintes funcionalidades gerais:

- Visualização de dados;
- Criação de Relatórios;
- Inserção de dados;
- Gestão de privilégios;
- Gestão de produtos.

Para implementar estas funcionalidades, é necessário invocar sub-funcionalidades do servidor de gráficos e do *web service* do PHC, tal como ilustrado na Figura 5.

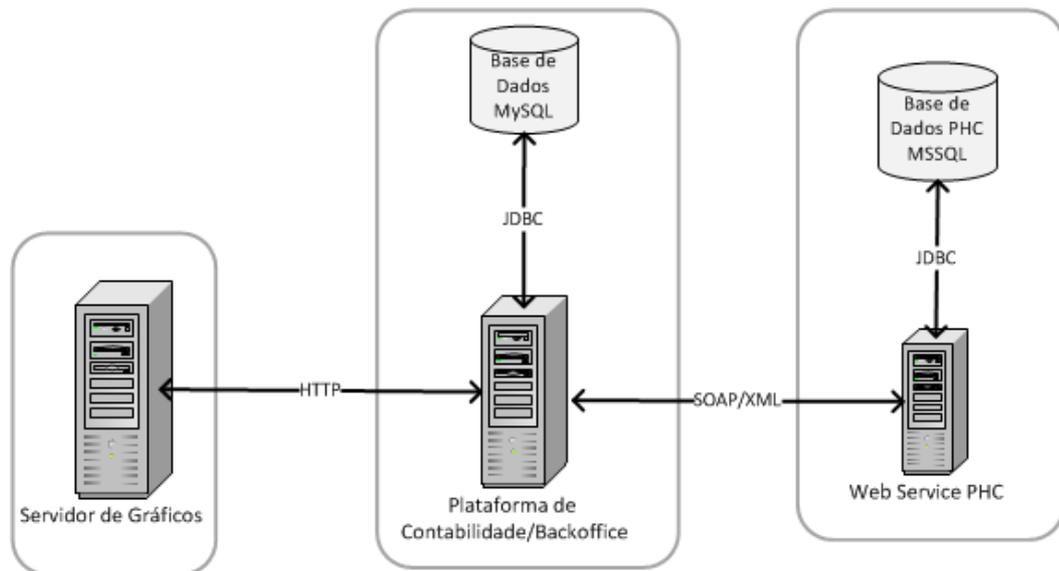


Figura 5 – Arquitetura da plataforma de contabilidade e as suas dependências.

Para a implementação da funcionalidade de visualização dos dados, a plataforma necessita de realizar uma série de cálculos com os valores existentes na base de dados. Para isso é necessário importar dados do PHC via *web service*. Esta importação ocorre todos os dias pela meia noite devido às constantes mudanças nos valores do PHC que interferem com os valores da plataforma. Existe uma *thread* responsável por analisar a hora atual que, caso seja a configurada, realiza um pedido ao *web service*, obtendo como resposta um pacote SOAP/XML com todos os dados relativos a unidades, clientes, projetos, centros de custo, faturação e movimentos existentes. Os dados não são confirmados nem são validadas as diferenças com os inseridos no dia anterior. Entre a data da importação e um ano atrás, todos os dados são eliminados e voltados a inserir, isto porque os dados podem sofrer alterações no PHC. Não há nada que impeça os utilizadores do mesmo de alterar uma despesa de um funcionário de há uns meses atrás por exemplo. Sendo assim, definiu-se, em conjunto com o departamento de contabilidade da WTI, o espaço de um ano para eliminação dos dados para garantir que os mesmos ficam atualizados.

Para a criação de relatórios é necessária a dependência do serviço PHP. Quando o utilizador clica na opção de criar relatório, o servidor envia um pedido http contendo todos os dados necessários para a criação dos gráficos. Este por sua vez, utilizando a tecnologia pchart (indicada em 4.2.4.3), procede à criação dos gráficos enviando a imagem dos mesmos como resposta ao pedido http.

A visualização de dados tal como depende do PHC, depende também da inserção dos dados no *back office*. Neste caso o *back office* encontra-se incluído na plataforma, tendo apenas acesso a este quem tem permissões para tal. O *Back office* é um módulo da plataforma de contabilidade responsável pela introdução de parte dos dados utilizados e das configurações necessárias para gerir os produtos e privilégios de utilizadores. A importação utilizando o PHC não contém todos os dados necessários para o desenvolvimento dos requisitos mencionados em 3.3. Sendo assim, foi necessário a intervenção do utilizador na inserção de dados, pois não existe nenhuma plataforma para gestão dos mesmos. Assim, usasse o *back office* para esse efeito.

Existe um módulo para cada conjunto de dados introduzidos/configurações nas seguintes áreas:

- Tesouraria;
- Orçamentos;
- KPI's;
- Qualidade de *software*;
- Recursos humanos;
- Gestão de produtos;
- Gestão de privilégios.

Cada módulo dispõe de um formulário para inserção dos dados. Quando este está preenchido e o utilizador efetua, a atualização, é realizada uma série de validações para confirmar se os dados se encontram corretamente formatados e se não contêm caracteres incorretos.

Para isto, existem na base de dados duas tabelas responsáveis por todo o tipo de dados e respetivos *templates* a utilizar no *back office*: *configuration_value* e *template*. O objetivo é as tabelas serem gerais o suficiente para que sejam usadas em todo o tipo de inserção de dados e respetiva leitura dos mesmos. Dois dos campos usados na tabela *template* indicam qual a expressão regular do valor a inserir e, caso isso não aconteça, qual a mensagem a mostrar ao utilizador. Desta forma, é feito tudo de forma homogénea para qualquer módulo: inserção, confirmação por *java script* se o valor se encontra correto e, caso contrário, surge o alerta a avisar como se deve preencher. Depois, os dados serão então todos inseridos na base de dados da plataforma.

A configuração de visualização dos dados a que cada utilizador tem acesso é realizada no *back office* por um utilizador mestre, este é o único que detém permissões para alterar os privilégios. Existe ainda a componente de gerir produtos. Como esta informação não se encontra em qualquer outro local tem de ser inserida e gerida no *back office*.

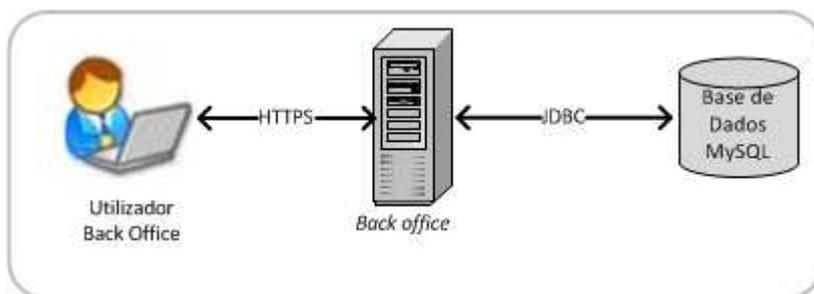


Figura 6 – Arquitetura do back office

A Figura 6 ilustra o fragmento da arquitetura do sistema relativa ao *back office*. Como podemos observar, existe o utilizador do *back office* – todos os membros da empresa que têm privilégios para a inserção dos dados necessários usarão este perfil de utilizador.

É de referir que no relatório apresentado na fase intermédia, o *back office* era uma componente à parte. No entanto, quando foi iniciado o desenvolvimento foi analisado em conjunto com o orientador da empresa qual a melhor abordagem a seguir, realizar uma aplicação à parte ou juntar à plataforma de contabilidade. Visto que o *back office* se relaciona com a plataforma e que não foi encontrada qualquer vantagem em fazer uma separação, foi decidido integrar o *back office* na plataforma de contabilidade, sendo assim mais um módulo.

4.1.2 Servidor de Gráficos

O servidor de gráficos é responsável pela criação de todos os gráficos utilizados na criação de relatórios. Este contém um conjunto de 4 ficheiros php, cada um responsável por gráficos de barras, empilhados, linhas e circulares. Desta forma não é necessário um ficheiro por cada gráfico gerado, mas sim um ficheiro por cada tipo, sendo apenas necessário filtrar os parâmetros na entrada para saber o que colocar em cada eixo. Em seguida, foi necessário fazer uso das funções disponíveis pela tecnologia *phpchart*, de modo a criar o gráfico e retornar o output para a plataforma. Uma vez que os ficheiros se encontram num servidor PHP denominado de *xampp* basta realizar um pedido para o URL onde o mesmo se encontra para gerar o gráfico e ter o retorno no output. O servidor encontra-se num processo independente pois corre *xampp*, enquanto a plataforma de contabilidade corre no *tomcat*.

Tal como já foi indicado, a troca de informação entre ambos os serviços é feita com pedidos http.

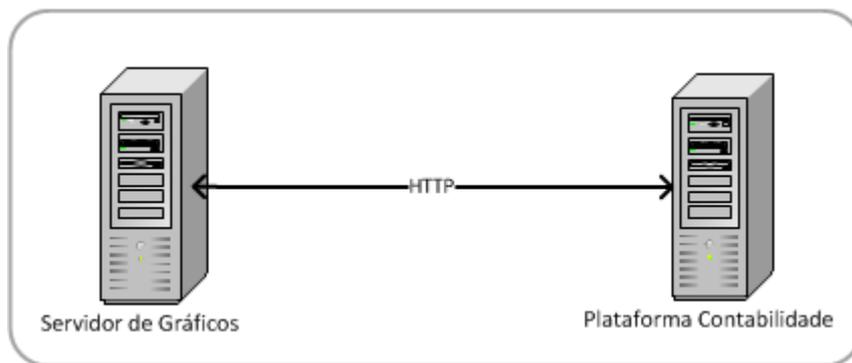


Figura 7 – Dependência da plataforma com o serviço de gráficos

É possível observar pela Figura 7 a troca de pedidos entre a plataforma e o serviço. Mais uma vez, é de notar que é a plataforma que realiza os pedidos.

4.1.3 Web Service PHC

Este serviço é responsável por toda a importação de dados da contabilidade analítica. Na figura a baixo, encontra-se o diagrama de arquitetura.

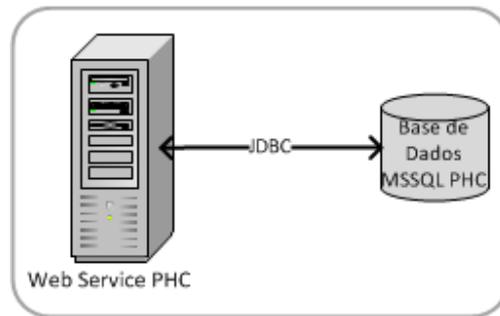


Figura 8 – Arquitetura do web service PHC

O Web service acede à base de dados do PHC, devolvendo todos os dados necessários que depois são enviados para o servidor utilizando XML. Analisando em detalhe como funciona o *web service*, verifica-se que é composto por objetos representantes dos dados a passar para o servidor, métodos de acesso à base de dados responsáveis por aceder à base de dados do PHC e por último temos a classe responsável por chamar os métodos da base de dados e retornar os resultados para o XML.

4.2 Tecnologias

Esta secção resume as tecnologias utilizadas durante o desenvolvimento da plataforma.

Como tecnologias de *api's* gráficas estudadas temos:

- Google Charts API;
- JFreeCharts;
- pChart;
- Charts4j.

Restantes tecnologias:

- Struts;
- MySQL;
- jExcel;
- iText.

4.2.1 Camada de Apresentação

Apache Struts [5] é uma ferramenta *open-source* web que facilita a criação de aplicações web em java, sendo uma das ferramentas permitidas na WIT para desenvolvimento Web. Uma maneira simples de conseguir criar páginas web dinâmicas é utilizando a arquitetura MVC. O *Model* representa o código da base de dados, a *View* representa o código do *design* da página e o *Controller* trata da lógica da plataforma.

4.2.2 Camada de Negócio

Nesta camada foi adotada a linguagem Java uma vez que é a linguagem usada pela empresa e é suportada pela *Web Framework* utilizada: *Struts*. Como servidor foi utilizado o *Apache Tomcat* [4], o mesmo é utilizado em grande parte dos projetos da empresa.

4.2.3 Camada de Persistência

A tecnologia usada para o armazenamento de dados foi o MySQL, isto porque é o usado nos projetos da empresa, gratuito e um ótimo motor de bases de dados.

4.2.4 API's Gráficas

4.2.4.1. Google Charts API

Website: <http://code.google.com/intl/pt-PT/apis/chart/>

Ferramenta que serve para criar vários tipos de gráficos em *frontend* utilizando código *java script*. Esta necessita de ligação à internet, é gratuita e simples de usar. Existem dois tipos de ferramentas disponíveis na construção de gráficos: *Image Charts* e *Google Chart API*.

O *Image Charts* envia os dados utilizados na construção dos gráficos para o Google. Como são dados confidenciais essa opção foi excluída. Sendo assim o *front-end* faz uso da segunda opção, a mesma mantém os dados localmente.

Como funcionalidades temos:

- Gráficos circulares (2D e 3D);
- Gráficos de área, podendo os mesmos ser *stacked*;
- Gráficos de barras, podendo os mesmos ser *stacked*;
- Gráficos de linhas;
- Criação de tabelas;
- Possibilidade de adicionar eventos e animações;
- Existência de legendas;
- Possibilidade de personalizar os gráficos.

Serão utilizados os seguintes gráficos:

- Barras (incluindo *stacked*);
- Circular (3D);
- Linhas;
- Tabelas.

4.2.4.2. JFreeCharts

Website: <http://www.jfree.org/index.html>

É uma ferramenta java *open-source*, gratuita, simples de perceber e bem documentada, que possibilita a criação de gráficos e a respetiva imagem.

Como funcionalidades gráficas temos:

- Gráficos de barras, podendo os mesmos ser *stacked*;
- Gráficos circulares (2D e 3D);
- Os gráficos podem ser legendados;
- Gráficos de área, podendo os mesmos ser *stacked*;
- Customização dos gráficos.

4.2.4.3. PChart

Website: <http://www.pchart.net/>

pChart é uma biblioteca PHP gratuita, que disponibiliza ao utilizador a criação de imagens diretamente no web *server*, sendo depois possível a sua utilização no lado do cliente.

Como funcionalidades gráficas temos:

- Gráficos de barras, podendo os mesmos ser *stacked*;
- Gráficos radar;
- Gráficos circulares (2D e 3D);
- Os gráficos podem ser legendados;
- Gráficos de área, podendo os mesmos ser *stacked*;
- Efeitos visuais;
- Possibilidade de colocar *labels* nos gráficos.

No caso da plataforma, o *pChart* é utilizado para a criação de gráficos em PHP. Existe um serviço http que permite aceder aos gráficos necessários, sendo devolvida a imagem destes, de modo a que possa ser utilizada na construção de relatórios.

4.2.4.4. Charts4j

Website: <http://code.google.com/p/charts4j/>

Charts4j é uma *api* gratuita de java que permite a criação de gráficos no lado do servidor. Como funcionalidades temos:

- A existência de todos os gráficos/tabelas do *Google charts tool*;
- Pode ser incorporado com qualquer aplicação Web;
- Código 100% java, não necessitando de bibliotecas de gráficos externas.

Esta tecnologia foi descartada, pois faz uso do *Image Charts* do *Google Chart tool*. Tal como referido em 3, os dados são enviados para o Google para construção dos gráficos, não sendo permitido o uso desta ferramenta.

4.2.4.5. Análise

Nesta secção são analisadas as principais vantagens e desvantagens de cada tecnologia estudada, indicando-se no fim quais as utilizadas e as razões dessa escolha.

API	Categoria	Desvantagens	Vantagens
Google Charts API	<i>Client Side</i>	<ul style="list-style-type: none"> Faz uso da internet para construção dos gráficos. 	<ul style="list-style-type: none"> Bem documentada; Permite ao utilizador interagir com o gráfico; Gráficos apelativos; É gratuita Vários tipos de gráficos disponíveis.
JFreeCharts	<i>Server Side</i>	<ul style="list-style-type: none"> Gráficos pouco apelativos. 	<ul style="list-style-type: none"> Vários tipos de gráficos disponíveis; Bem documentada; É gratuita; Não necessita de ligação à internet.
PChart	<i>Server Side / Client side</i>		<ul style="list-style-type: none"> Vários tipos de gráficos disponíveis; Bem documentada; Gráficos apelativos; É gratuita Não necessita de ligação à internet.
Charts4j	<i>Server Side</i>	<ul style="list-style-type: none"> Faz uso do Image Charts para construção dos gráficos; Pouco documentada; Necessita de ligação à internet. 	<ul style="list-style-type: none"> Gráficos apelativos; É gratuita; Vários tipos de gráficos disponíveis.

Tabela 14 – Lista de vantagens e desvantagens existentes em cada API.

Na Tabela 14 são apresentadas as principais desvantagens e vantagens de cada tecnologia. Da análise das tecnologias indicadas, a nível de *cliente side* foi escolhida como ferramenta a usar o *Google Charts*. O *ImageCharts* foi descartado pois envia os dados para a Google. O *Google charts* permite que os gráficos sejam bastante apelativos, existindo ainda interação do utilizador com os mesmos, sendo assim uma ótima escolha, principalmente por questões de usabilidade. O *pChart* não foi considerado, pois os gráficos não permitem interação com o utilizador, sendo assim o escolhido para a criação de relatórios - *server side*.

Por último temos a escolha da ferramenta a usar em *server side*. Existem três possibilidades: *pChart*, *jFreeCharts* e *chart4j*. Comparando as funcionalidades, vantagens e desvantagens de cada uma, é possível concluir que a mais adequada é o *pCharts*. O *jFreeCharts*, tem gráficos muito pouco apelativos, tornando assim a análise mais pobre. O *charts4j*, tal como já foi indicado, usa o *Image Charts*, excluindo assim a sua escolha.

A necessidade de haver tecnologias do lado do servidor e do lado do cliente deve-se ao facto da ferramenta escolhida para o lado do cliente não permitir usar os gráficos do lado

do servidor. No requisito em que são produzidos relatórios através do módulo de criação de relatórios (requisito com o ID RF-CA-09), não é necessária a visualização dos gráficos para produzir os relatórios. Caso existam 20 gráficos no relatório, por exemplo, não iríamos querer visualizar todos os gráficos no *front end* (péssimo para a usabilidade e performance da página), que seria o que teria de ser feito caso utilizássemos a mesma ferramenta para ambos. Sendo assim, juntamente com o orientador da empresa chegou-se à conclusão que seria preferível ter duas ferramentas distintas: *Google Charts* para o lado do cliente e *pChart* para o lado do servidor.

4.2.5 Restantes Tecnologias

IText

Biblioteca que permite a criação e manipulação de documentos, sendo possível criar, ler e realizar atualizações a documentos PDF.

jExcel

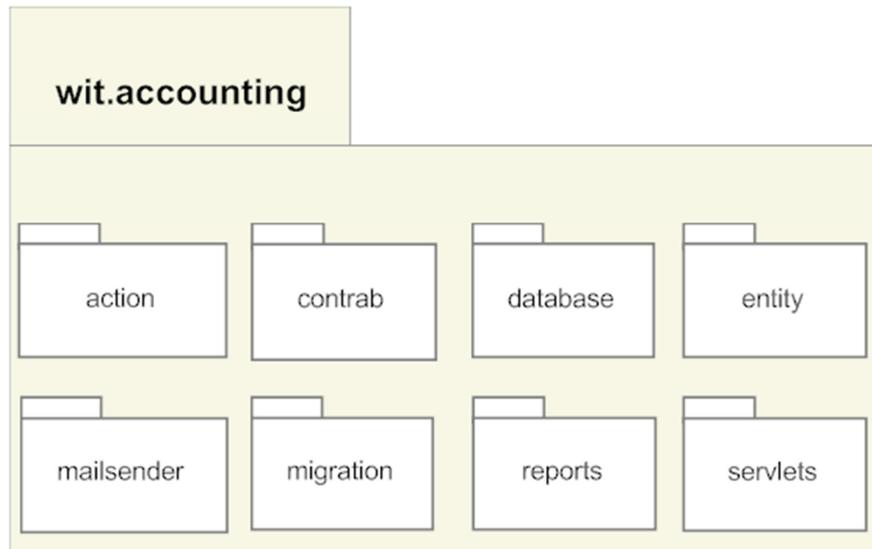
Biblioteca java *open source* que permite ler e escrever ficheiro *excel*. A mesma é utilizada para realizar os *downloads* dos relatórios.

5 Implementação

Partindo agora para um grau de detalhe mais aprofundado, neste capítulo analisamos todos os *packages* utilizados no projeto e as suas dependências.

5.1 Diagrama de *Packages*

5.1.1 AccountingPlatform



Podemos observar no diagrama acima que existem no total 9 *packages* organizados de forma hierárquica.

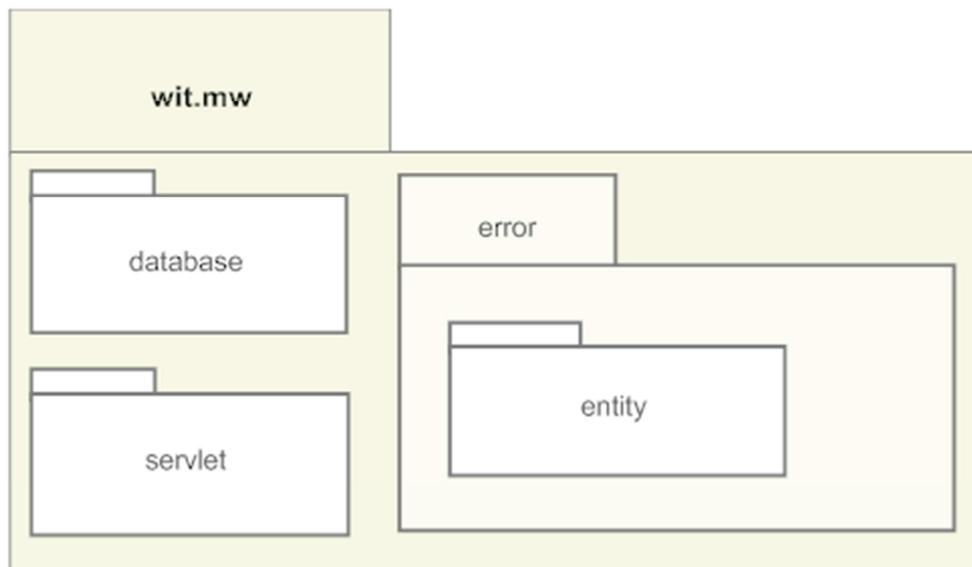
Packages	Descrição
wit.accounting	<i>Package</i> principal responsável por iniciar e terminar todos os recursos, <i>logs</i> do servidor, ficheiros de propriedades e todos os módulos.
action	Contém todas as <i>Actions</i> criadas na implementação da <i>pattern</i> MVC devido ao uso de <i>framework struts</i> . Desta forma encontram-se aqui todas as relações View-Controller encontradas no projeto.
contrab	<i>Package</i> responsável por inicializar e terminar a <i>thread</i> MigrationPHC que, tal como o nome indica, é a que trata da importação dos dados vindos do PHC. É verificada a hora e inicializada a importação à meia-noite de cada dia.
Database	Módulo responsável por todos os acessos à base de dados. É composto por 3 classes: DatabaseModule – interface que contém todos os métodos utilizados no acesso à base de dados, AbstractDatabase – classe que implementa o DatabaseModule e que contém todos os métodos globais a usar e, por fim, temos o SQLiteDatabase que estende AbstractDatabase e que contém todos os acessos à base de dados.
Entity	Neste <i>package</i> encontram-se todas as entidades utilizadas no projeto.

mailsender	Módulo que implementa a funcionalidade de envio de relatórios por <i>email</i> . O mesmo utiliza o Java Mail API.
migration	Módulo responsável por fazer a ligação entre o <i>web service</i> e o servidor. Recebe os dados e coloca-os na base de dados da plataforma.
reports	Módulo responsável por ter todas as classes necessárias para criação de relatórios e/ou gráficos. É aqui que se utilizam todas as tecnologias <i>server side</i> para criação de relatórios em PDF e Excel, bem como as ligações ao servidor de gráficos PHP.
servlets	Aqui encontra-se a <i>InitServlet</i> , que é responsável por inicializar a aplicação. Existem ainda os filtros para UTF-8 e <i>Cross Site Scripting</i> .

Tabela 15 – Packages criados na plataforma.

5.1.2 Web Service PHC

No diagrama abaixo é apresentada a estrutura do *web service* criado para importar os dados do PHC. Este é de menor dimensão, sendo composto por 5 *packages*.



Packages	Descrição
wit.mw	Package principal responsável por iniciar todos os recursos, logs do servidor, ficheiros de propriedades, todos os módulos e terminar os mesmos.
database	Módulo responsável por todos os acessos à base de dados. É composto por 3 classes: <i>DatabaseModule</i> – interface que contém todos os métodos utilizados no acesso à base de dados, <i>AbstractDatabase</i> – classe que implementa o <i>DatabaseModule</i> e que contém todos os métodos globais a usar. Por fim temos o <i>MSSQLDatabase</i> que herda <i>AbstractDatabase</i> e que contém todos os acessos à base de dados.
error.entity	Neste package encontram-se todas as entidades utilizadas no web service.

Servlets

Aqui encontra-se a *InitServlet*, que é responsável por iniciar a aplicação. Existe ainda o filtro para UTF-8.

Tabela 16 – Packages usados no web service.

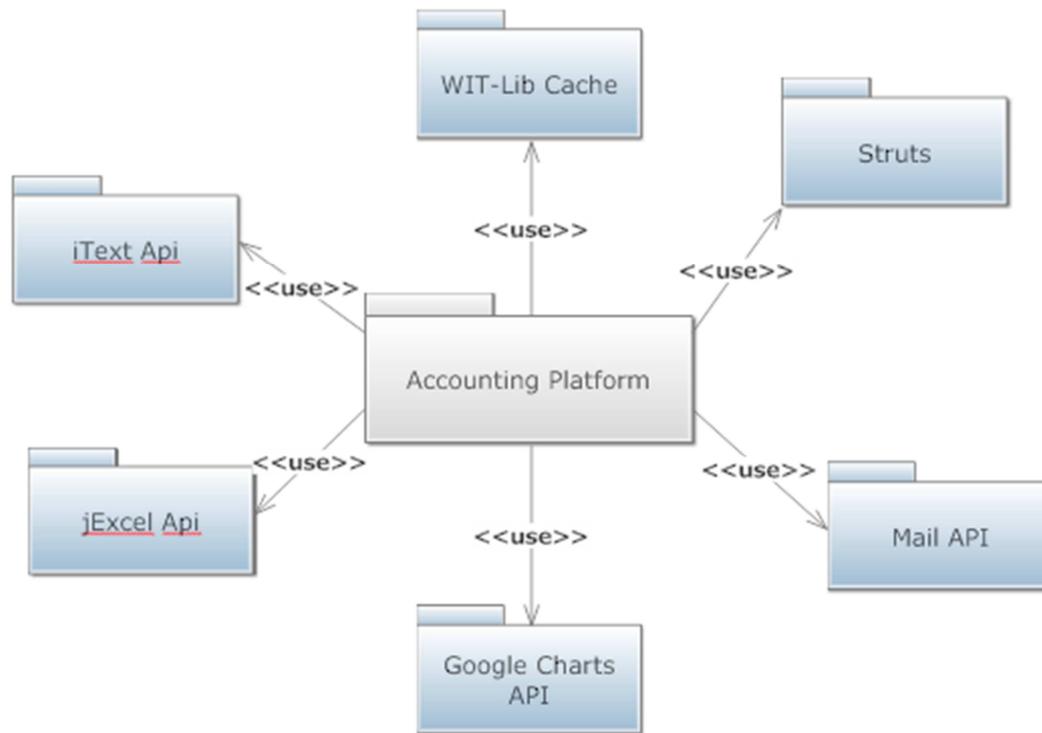
5.1.3 PHPService

Neste caso a estrutura do projeto é diferente, existindo apenas um ficheiro para cada tipo de gráfico, o que faz um total de 4 ficheiros.

5.2 Diagrama de Bibliotecas Externas

Nesta secção podemos visualizar o diagrama com as bibliotecas externas utilizadas pela plataforma de contabilidade: *Accounting Platform*. No caso do *web service* não é feita qualquer tipo de uso de bibliotecas externas.

5.2.1 Accounting Platform



Biblioteca	Descrição
WIT-Lib Cache	A Wit-Lib Cache é utilizada para implementar o mecanismo de cache criado na otimização de escalabilidade. Desta forma existe uma <i>Hashtable</i> com os dados que guardados na cache.
Struts	<i>Struts</i> é a ferramenta web utilizada. A mesma implementa o modelo MVC – <i>Model View Controller</i> . O <i>Model</i> representa o código da base de dados, a <i>View</i> representa o código do <i>design</i> da página e o <i>Control</i> trata da lógica da plataforma.
Mail API	É utilizada o <i>JavaMail API</i> para implementar a funcionalidade de envio de relatórios por <i>email</i> para os utilizadores.

Google Charts API	O <i>Google Charts API</i> é simples de usar, gratuito e oferece uma boa interatividade entre o utilizador e os gráficos. É utilizado para criar todos os gráficos de <i>frontend</i> .
JExcel API	<i>JExcel API</i> é uma biblioteca java <i>open source</i> que permite ler e escrever ficheiro excel. É utilizada para gerar os relatórios em formato <i>excel</i> , para <i>download</i> .
iTextAPI	<i>IText</i> é uma biblioteca que permite a criação e manipulação de documentos pdf. É usada para gerar os relatórios em formato PDF, para <i>download</i> .

Tabela 17 – Descrição detalhada de bibliotecas externas utilizadas na plataforma.

5.2.2 PHPService

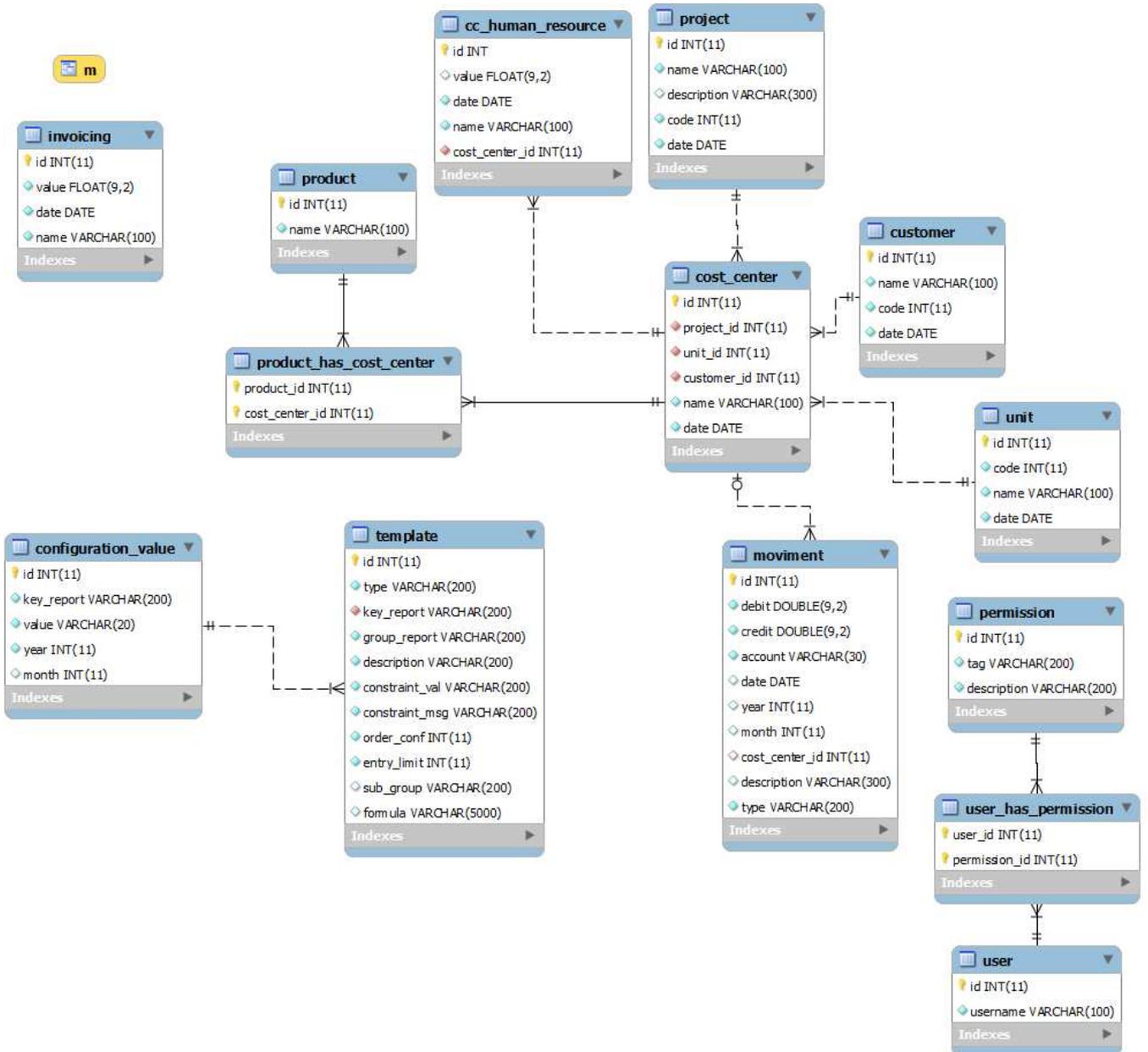


Bibliotecas	Descrição
pCharts	<i>pChart</i> é uma biblioteca PHP que permite gerar imagens no lado do servidor. É utilizada para criar os gráficos a utilizar nos relatórios exportados para PDF.

Tabela 18 – Descrição detalhada das bibliotecas externas utilizadas no serviço de gráficos.

5.3 Modelo da Base de Dados

Abaixo é apresentado o diagrama ER da base de dados criada, que é composta por uma *view* (m) e 14 tabelas.



Nome	Tipo	Descrição
cost_center	Tabela	Tabela onde são armazenados todos os dados detalhados dos centros de custo que existem no PHC incluindo id, nome, id da unidade, id do cliente e id do projeto.
unit	Tabela	Lista de todas as unidades existentes na contabilidade da empresa. Os detalhes correspondem ao id, nome e código da unidade.
customer	Tabela	Lista de todos os clientes existentes na empresa. Os detalhes correspondem ao id, nome e código do cliente.
project	Tabela	Lista de todos os projetos existentes na empresa. Os detalhes correspondem ao id, nome, descrição e código do projeto.
moviment	Tabela	A tabela moviment serve para guardar todos os dados contabilísticos da empresa, o que inclui: id, débito, crédito, a conta em causa, a data, o ano, o mês, a descrição, o id do centro de custo correspondente e o tipo. O crédito e débito dizem respeito aos gastos da empresa, estando associados a uma determinada conta SNC [15] representada por “account”. Depois disto, temos a data. No início optou-se por ter um campo do tipo <i>date</i> , mas quando se implementou a contabilidade geral verificou-se existem os meses 13, 14 e 15, tendo assim que existir alterações a este nível que permitissem inserir meses posteriores a 12. Daí a criação dos campos <i>year</i> e <i>month</i> . Existe, ainda, uma descrição associada ao movimento de um centro de custo. Por fim, temos o <i>type</i> que indica se o movimento associado é da contabilidade analítica ou geral.
cc_human_resource	Tabela	Neste caso, temos todos os custos dos funcionários da empresa, que estão SEMPRE relacionados com um centro de custo. Existe assim um <i>name</i> que indica o nome do funcionário, um <i>id</i> , um <i>cost_center_id</i> e por último, o valor associado ao funcionário representado por <i>value</i> .
product	Tabela	A tabela product contém a lista de todos os produtos criados no <i>back office</i> . Estes são representados por um <i>name</i> e por um <i>id</i> .
product_has_cost_center	Tabela – entidade fraca	Como os produtos podem ter vários centros de custo associados existe uma entidade fraca que relaciona ambas as tabelas: product-cost_center.
template	Tabela	A tabela template contém os templates de todas as tabelas que necessitam de inserção de dados. Desta forma existe um id, um <i>type</i> que indica o tipo de dados que estamos a inserir (texto, números, etc.), um <i>group_report</i> que indica a que grupo pertence a tabela (<i>file</i> , <i>employee</i> , etc.), um <i>key_report</i> que define uma <i>key</i> para cada entrada e que é única. Temos a descrição que permite preencher as tabelas do <i>back office</i> com os títulos corretos nas entradas. Para isso tem que existir o campo <i>order_conf</i> , de modo a sabermos qual a ordem de

		visualização. Para permitir que a inserção de dados seja realizada corretamente existem os campos: <i>constraint_val</i> e <i>constraint_msg</i> que contêm, respectivamente, a expressão regular que deve ser validada pelo <i>input</i> do utilizador, e a mensagem deve ser apresentada quando a expressão regular não é verificada. O campo <i>entry_limit</i> indica o tamanho máximo do campo a inserir pelo utilizador na tabela. O <i>sub_group</i> é necessário para o caso de existirem grupos nas tabelas. Por último, a <i>formula</i> permite fazer cálculos que sejam necessários, nomeadamente totais.
configuration_value	Tabela	Neste caso temos um id, a <i>key</i> : <i>key_report</i> , que corresponde à entrada na tabela <i>template</i> , o valor inserido pelo utilizador e, por fim, o ano e o mês.
invoicing	Tabela	Tabela onde são guardados os dados de faturação dos clientes sendo os mesmos representados por um id, pelo nome do cliente: <i>name</i> , pelo valor: <i>value</i> e por último pela data da faturação: <i>date</i> .
user	Tabela	A tabela user é responsável por ter uma lista de todos os utilizadores da aplicação. Os mesmos são representados por um <i>username</i> e por um id.
permission	Tabela	A tabela permission contém uma lista de todos os tipos de permissões da aplicação. As mesmas são representadas por um id, uma descrição e uma <i>tag</i> que indica o tipo de permissão.
user_has_permission	Tabela – entidade fraca	Como os utilizadores podem ter várias permissões associadas, existe uma entidade fraca que relaciona ambas as tabelas: user-permission.
m	Vista	Vista responsável por ter uma lista de números de 1 a 12 de modo a que o resultados das <i>queries</i> realizadas contenham obrigatoriamente movimentos nos 12 meses.

Tabela 19 – Descrição detalhada do ER.

5.4 Detalhes de Implementação

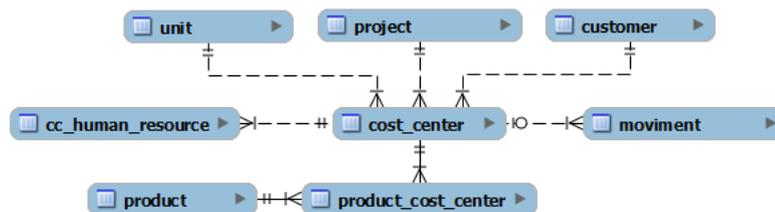
As páginas *web* foram implementadas usando HTML, *javascript*, *struts* e *css*. De um modo geral, a implementação pode ser dividida em módulos: contabilidade analítica, geral e *back office*. Cada um destes componentes contém detalhes específicos, generalizando-se cada caso de modo a não se repetir código.

5.4.1 Contabilidade Analítica

Toda a implementação por trás gira à volta de bases de dados, cálculos e criação de gráficos. Seguindo a lógica do modelo MVC, temos a *view*, o ficheiro jsp principal: *Analytical.jsp*. Depois disso é usado *iframes* para permitir a visualização lateral do menu e ao mesmo tempo ter o gráfico com os resultados, não sendo necessário atualizar a página inteira e perder a pesquisa realizada.

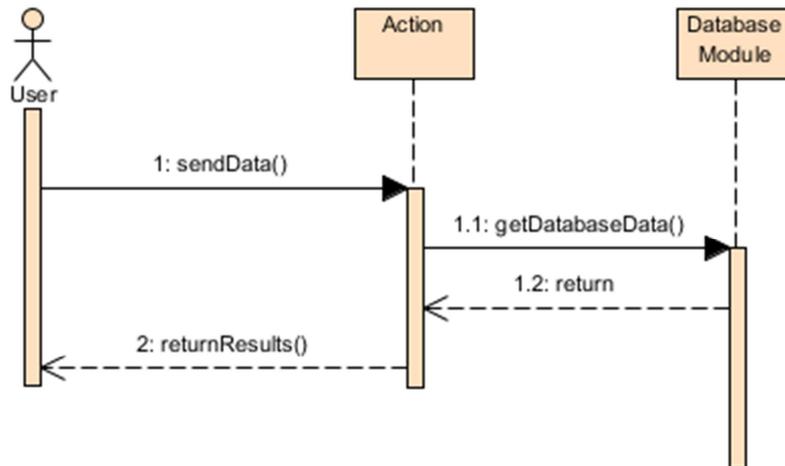
No que toca ao menu usado na pesquisa da contabilidade analítica, a implementação foi elaborada com o auxílio da biblioteca: *jQueryUI* [6], que já disponibiliza *css* para a criação do menu, não sendo necessário qualquer tipo de implementação extra. Após o utilizador selecionar os dados a pesquisar e premir o botão para criar o gráfico, os dados são validados e enviados para a *Action* responsável pelo gráfico. Depois disto, são chamados os acessos há base de dados que vão lidar com todos os valores a apresentar sendo retornada uma lista com os dados do objeto a visualizar.

A contabilidade analítica gira à volta de duas tabelas da base de dados: “*moviment*” e “*cc_human_resource*”. Os dados a visualizar nas pesquisas são distribuídos por 5 tabelas: “*unit*”, “*customer*”, “*project*”, “*cost_center*” e por último “*product*”. Tudo isto pode ser visualizado no diagrama abaixo. É de salientar que um centro de custo é composto por uma unidade, um cliente e um projeto, daí as dependências na base de dados.



A ligação entre cada tabela e o centro de custo é feita pelos respetivos id's: *unit.id*, *customer.id* e *project.id*. A informação dos produtos não se encontra disponível no PHC, daí ter sido acrescentada uma funcionalidade de gestão de produtos no *back office*.

Qualquer um dos gráficos apresentados na analítica necessita de processamentos específicos por parte do servidor. Cada um deles tem contas próprias do SNC [15] e cada uma tem de ser processada da sua maneira (em alguns casos conta apenas o débito, noutras apenas o crédito ou ambos), além de que as normas podem ser alteradas anualmente. Desta forma criamos uma *Action* para cada tipo de gráfico onde são acedidos os métodos da base de dados que permitem tratar de custos e proveitos.



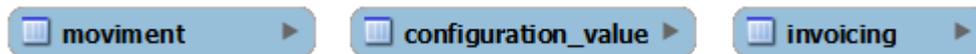
O mesmo processo é feito em todos os módulos da plataforma: temos o utilizador que envia os dados para a *action* e esta acede à base de dados devolvendo de seguida os resultados. Podemos observar no diagrama de sequência acima, de um modo muito geral, a forma como são tratados os dados.

5.4.2 Contabilidade Geral

A contabilidade geral divide-se em 4 relatórios e 1 pequeno módulo: demonstração de resultados, balanço, fornecimentos e serviços externos, gastos com o pessoal, e faturação, respetivamente. Os 4 relatórios são tratados de maneiras diferentes: os dois primeiros seguem ambos o mesmo tipo de tratamento, mas são visualizados campos diferentes; os dois últimos são tratados da mesma forma e encontram-se generalizados o suficiente para não terem de ser feitas *queries* demasiado específicas. Por último, temos o módulo da faturação, que segue o mesmo esquema da contabilidade analítica: menu lateral com a possibilidade de pesquisa e uma *iframe* para visualização dos resultados.

Os *templates* das tabelas apresentadas encontram-se num ficheiro *excel* que é acedido pela plataforma no momento do primeiro pedido que o utilizador realiza ao respetivo relatório. A informação é lida e colocada num *array* de objetos onde temos toda a informação necessária: estrutura da tabela, descrição de cada entrada da mesma, uma chave que é única para cada entrada e contas do PHC que é necessário aceder com as respetivas fórmulas para obter um resultado.

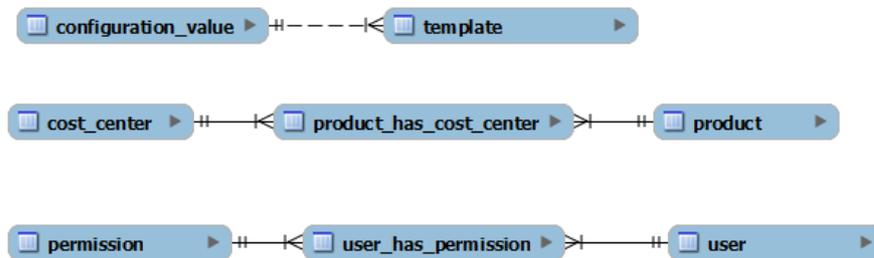
Quanto às tabelas da base de dados relevantes para este caso temos as seguintes:



Na tabela *moviment* temos os dados das contas do PHC que permitem realizar os cálculos pretendidos. É de se notar que na contabilidade geral os movimentos não estão associados a centros de custo, daí não estar a relação na tabela. Sendo assim na tabela *moviment* temos os dados das 4 tabelas, na *configuration_value* temos os valores que foram inseridos no *back office* e que é necessário visualizar e, por último, temos a tabela *invoicing* que contém os dados do módulo faturação.

5.4.3 Back Office

O *back office* tem como funcionalidades gerir os produtos, os utilizadores e inserir os dados que não se encontram no PHC. Desta forma o mesmo envolve 8 tabelas representadas no esquema abaixo.



Tal como já foi mencionado na explicação do diagrama ER as tabelas *configuration_value* e *template* encontram-se relacionadas para permitir que as tabelas que são preenchidas no *back office* contenham valores associados na tabela *configuration_value*. Quanto aos utilizadores e aos produtos, o raciocínio é o mesmo: o utilizador adiciona um produto relacionando o mesmo com um centro de custo, gerando assim a entidade fraca: *product_has_cost_center*. Os utilizadores têm permissões específicas relacionadas através da tabela: *user_has_permission*.

Existem 3 módulos e 3 relatórios que dependem dos valores do *back office*. São eles contabilidade analítica, recursos humanos, *software quality*, fornecimentos e serviços externos, gastos com o pessoal e tesouraria, respetivamente. A contabilidade analítica necessita dos valores de KPI para gerar o gráfico Revenues vs KPI. Os recursos humanos e o *software quality* precisam de toda a informação inserida e todos os requisitos dos mesmos dependem dos dados. Por fim temos os relatórios: os dois primeiros dependem dos valores inseridos no orçamento e a tesouraria depende por completo dos dados do *back office*, pois é lá que o utilizador a preenche.

5.4.4 Otimizações

Ao realizar o *profiling* das páginas usando a ferramenta *jprofiler* [10] foi possível observar que a aplicação se encontrava a perder muito tempo em todos os acessos à base de dados e com o respetivo tratamento de resultados, principalmente na contabilidade geral. Desta forma procedemos a algumas otimizações a nível do número de acessos à base de dados, da forma como se tratam os dados, modo de ler o *template* que se encontra no ficheiro *excel*, parâmetros de configuração de base de dados e, por fim, cache.

As otimizações foram feitas em dois módulos: contabilidade geral e analítica. Todas as otimizações se encontram detalhadas no Anexo B – Requisitos não funcionais. Aqui apenas se encontra um resumo das mesmas.

Contabilidade Analítica

Foram realizadas otimizações em dois pontos: na base de dados e na cache. Optou-se por visualizar os registos da aplicação e recorrer ao comando: “*explain extended query*”, de

modo a tentar perceber se existia alguma otimização a ser feita e quais as *queries* que se encontravam a demorar mais tempo, chegando assim à conclusão que duas delas tinham um tempo de execução maior que as outras. Ao rever as *queries* chegou-se à conclusão de que existia espaço para reduzir o número de acessos à base de dados.

Quando foi gerado o script para a criação da base de dados foram criados desde início os índices para melhoria da velocidade das pesquisas. Desta forma com o comando *explain extended* do *mysql* conclui-se que todas as *queries* se encontravam a usar os índices quando é indicado, ou seja, quando existem muitas entradas.

Por fim procedeu-se a uma segunda otimização: utilização de cache, pois os dados das tabelas utilizadas apenas mudam uma vez por dia. Sendo assim utilizou-se uma cache desenvolvida pela WTT, onde se define o tempo em milissegundos em que os dados vão permanecer em cache, sendo neste caso esse tempo de 1 dia. Passámos então a colocar em cache o resultado das *queries*.

Contabilidade Geral

Primeiro que tudo foi feita uma análise de todo o código para ver onde poderíamos começar as otimizações, concluindo assim que havia espaço para otimizar o número de acessos há bases de dados. Cada vez que o utilizador realizava login na aplicação e fazia um pedido para visualizar os relatórios do módulo, a *action* responsável pelo mesmo ia ao *excel* ler o *template* (um pedido igual a um acesso ao *excel*), realizava *parse* das fórmulas a utilizar e de seguida por cada conta do SNC ia-se buscar o valor associado à mesma e efetuam-se os cálculos necessários. Por cada acesso à base de dados trazia o valor de uma conta, resultando em sobrecarga no processamento dos dados visto que cada entrada da tabela pode ter várias contas SNC associadas para fazer com os cálculos.

Desta forma analisou-se detalhadamente o código e efetuam-se algumas otimizações. O carregamento do *template* do *excel* passou a ser feito apenas uma vez por sessão. Para isso utilizou-se uma *Hashtable* se guarda o *array* numa *key*. Passa então a existir uma leitura do *excel* por sessão.

De seguida passámos a analisar o método que trata de enviar os pedidos para a base de dados e de efetuar os devidos cálculos. Optámos então por otimizar o número de acessos à base de dados. À medida que vamos analisando as contas a aceder na base de dados vamos criando a *query* de acesso à mesma trazendo assim todas as contas de um certo cálculo de uma só vez.

Ao analisar os métodos de acesso à base de dados no *jprofiler* observou-se que existia algum tempo de espera por parte dos mesmos. Optou-se assim por alterar a forma como as ligações à base de dados eram iniciadas. Anteriormente as ligações eram iniciadas apenas à medida que era necessário. No produto final são iniciadas 20 ligações durante o arranque do servidor, não sendo necessário esperar pelo estabelecimento das mesmas. Para chegar a este valor foram-se realizando testes para confirmar qual o valor ideal a utilizar. A partir do 20 não existe qualquer tipo de alteração no tempo de espera.

Por último, procedeu-se com a colocação da cache. O utilizador apenas acede à base de dados para retirar os valores uma vez por dia. Desta forma, todos os valores se encontram previamente calculados e não é necessário qualquer tipo de acesso a bases de dados ou realizar os cálculos durante os restantes acessos. Visto que a cache traz bastantes vantagens, optou-se por utilizar a cache em todos os relatórios que apenas dependem dos valores do PHC e não de terceiros como é o caso dos valores inseridos no *back office*.

5.5 Produto Final

A presente secção visa apresentar o produto final, tentando abranger um pouco de cada módulo. Apenas será fornecido um exemplo de cada um, seguido da descrição do mesmo. Todos os dados presentes nas figuras foram alterados por motivos de confidencialidade.

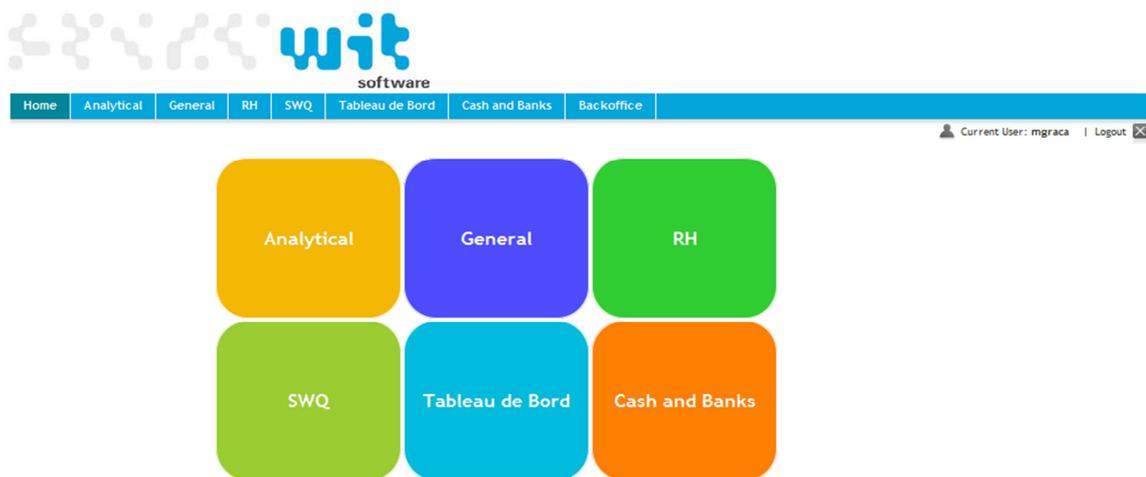


Figura 9 – Menu inicial da plataforma.

Na figura acima podemos observar o menu inicial da plataforma, onde temos acesso a todos os módulos da mesma. É ainda visível que o utilizador fez *login* com o *username* mgraca, o que significa que na página da *intranet* onde o SSO faz *login* o utilizador inseriu esse mesmo *username*.

Daqui para a frente vamos apresentar os módulos da aplicação, mas apenas será apresentada uma figura por módulo.

Na figura abaixo podemos analisar os gráficos da contabilidade analítica, sendo que o utilizador selecionou uma unidade e a opção de contribuição. É de salientar que os resultados visualizados se dividem em 3 gráficos circulares: a contribuição de cada cliente nos custos, proveitos e resultados da unidade selecionada. As restantes funcionalidades da contabilidade analítica seguem o mesmo *interface*, sendo as únicas diferenças o tipo de gráfico e a tabela resumo (que não existe nos gráficos de contribuição).

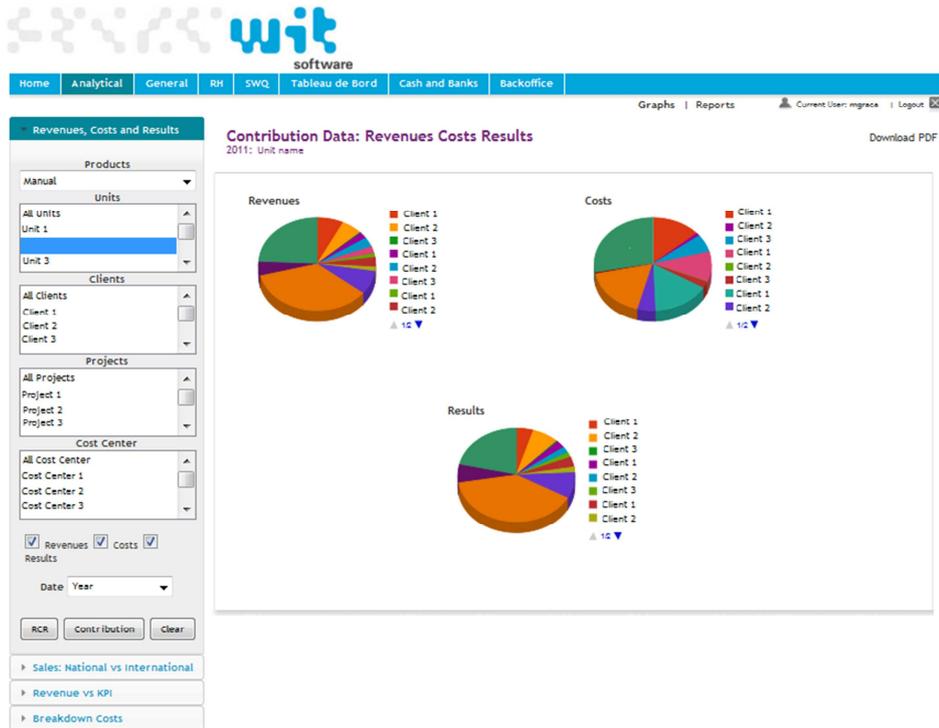


Figura 10 – Módulo da contabilidade analítica.

Na figura abaixo é apresentada uma tabela da contabilidade geral, sendo que o utilizador selecionou a opção *Employees Costs*. É de salientar que o *interface* das restantes opções é semelhante, apenas mudam os campos a apresentar nas tabelas. A única diferença é no *Invoicing*, que é semelhante à contabilidade analítica (contém o menu lateral com as opções de gráficos).

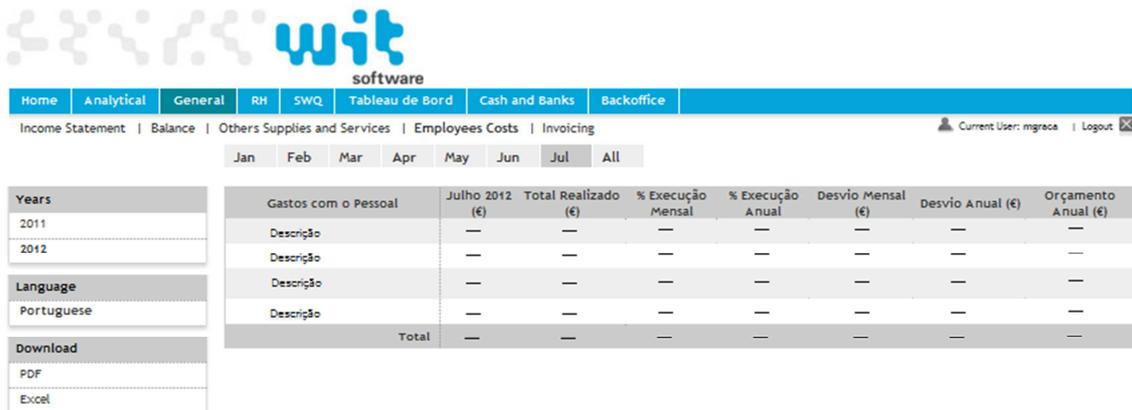


Figura 11 – Módulo de contabilidade geral.

O módulo de *Cash and Banks* (tesouraria) não é ilustrado nesta secção pois é muito semelhante ao apresentado na figura anterior.

Na figura abaixo podemos observar o menu de recursos humanos, onde temos as opções indicadas nos requisitos: criar o relatório de RH, a tabela QREN e os gráficos. Como se pode constatar o menu é muito semelhante ao da contabilidade analítica.

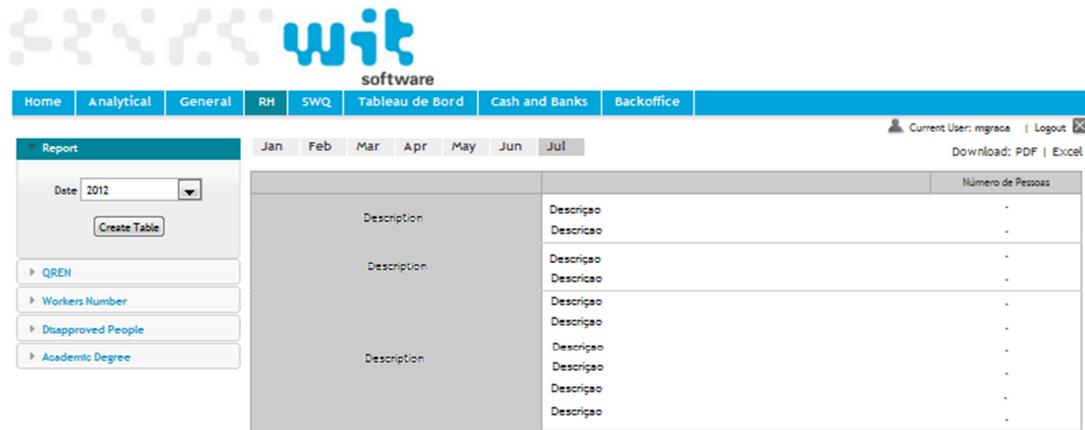


Figura 12 – Módulo de Recursos Humanos.

O módulo de *software quality* também é bastante semelhante, temos as opções indicadas nos requisitos: criar o relatório de SWQ e os gráficos. Mais uma vez, os menus são semelhantes por motivos de *user experience*.

Por fim, temos o *back office* – módulo responsável pela inserção de todos os dados que não se encontram no PHC e por gerir os utilizadores e os produtos. Como se pode observar na figura abaixo existem os diferentes menus: RH, *Software Quality*, *Cash and Banks*, *Budget*, *KPI*, *Permissions* e por último *Products*. Neste caso temos o exemplo do *Budget*, com a opção *Edit* no fim de cada coluna. Uma vez premido, o botão transforma-se em *Save*.

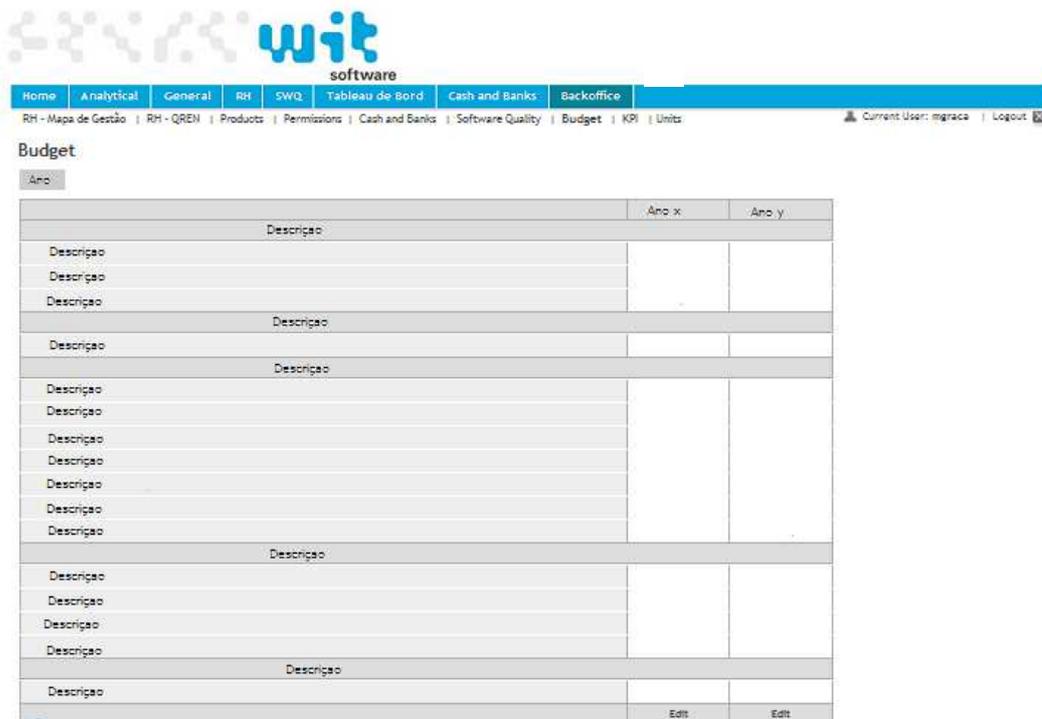


Figura 13 – Back Office

5.6 Testes

Durante o desenvolvimento da plataforma foi realizada uma série de testes de acordo com o plano previsto. Estes consistiram na confirmação de todos os valores que passam pela plataforma, onde não existe margem para erro, pois são dados contabilísticos usados no dia-a-dia da empresa.

Nenhum dos testes foi realizado pela equipa de qualidade de *software* da empresa, uma vez que esta não tem conhecimentos de contabilidade para confirmar os valores que se encontram na plataforma. Além disso, todos os dados são confidenciais e não podiam ser partilhados. Como foi necessário recorrer ao departamento de contabilidade para realizar os testes, não se podia exigir do mesmo uma especificação de testes.

6 Plano de Trabalho e Implicações

Neste capítulo são abordados os seguintes pontos: processo utilizado, planeamento inicial, planeamento intermédio e planeamento final.

6.1 Processo

Este projeto seguiu uma metodologia ágil. O manifesto [3] para esta metodologia refere:

“Ao desenvolver e ao ajudar outros a desenvolver software, temos vindo a descobrir melhores formas de o fazer. Através deste processo começámos a valorizar:

Indivíduos e interações mais do que processos e ferramentas

Software funcional mais do que documentação abrangente

Colaboração com o cliente mais do que negociação contratual

Responder à mudança mais do que seguir um plano

Ou seja, apesar de reconhecermos valor nos itens à direita, valorizamos mais os itens à esquerda.”

Existem 12 princípios no desenvolvimento ágil [3]:

- Satisfazer o cliente através da entrega rápida e contínua de *software*;
- Alterações nos requisitos são sempre recebidas, mesmo em fases tardias do desenvolvimento;
- Entregar frequentemente *software* funcional;
- O cliente e a equipa de desenvolvimento devem trabalhar juntos, diariamente, durante o decorrer do projeto;
- Os projetos são realizados por pessoas motivadas, existindo um bom ambiente e apoio;
- A comunicação deve ser feita pessoalmente e de forma direta, pois é a melhor forma de passar informação;
- A principal medida de progresso é a entrega de *software* funcional;
- Manter um ritmo constante;
- A atenção permanente à excelência técnica e um bom desenho da solução aumentam a agilidade;
- As melhores arquiteturas, requisitos e desenhos surgem de equipas auto-organizadas;
- Simplicidade;
- Adaptação regular a mudanças no projeto.

Durante o projeto o *software* em desenvolvimento foi sendo fornecido ao cliente, através de atualizações regulares.

Os requisitos foram alterados com alguma frequência, tendo sido pedida a adição de mais funcionalidades ao produto.

Foram realizadas reuniões regulares, onde foram verificados todos os aspetos já realizados, foi fornecido *feedback* sobre os mesmos, e foram enumerados os pontos importantes a alterar/acrescentar.

Existiu ainda um constante apoio por parte do orientador na empresa, em todos os aspetos: sugestões e questões técnicas, entre outros. O facto de ser uma metodologia ágil, não significa que não exista um processo. Existiam objetivos bem definidos, existia um plano e os requisitos podiam sofrer ajustes mas não ser alterados por completo.

6.2 Equipa

A equipa foi composta pelo orientador: Raul Fonseca e pela estagiária: Mónica Graça. Existiram ainda reuniões regulares com a cliente/diretora de contabilidade da WTT, Susana Órfão.

6.3 Planeamento Inicial

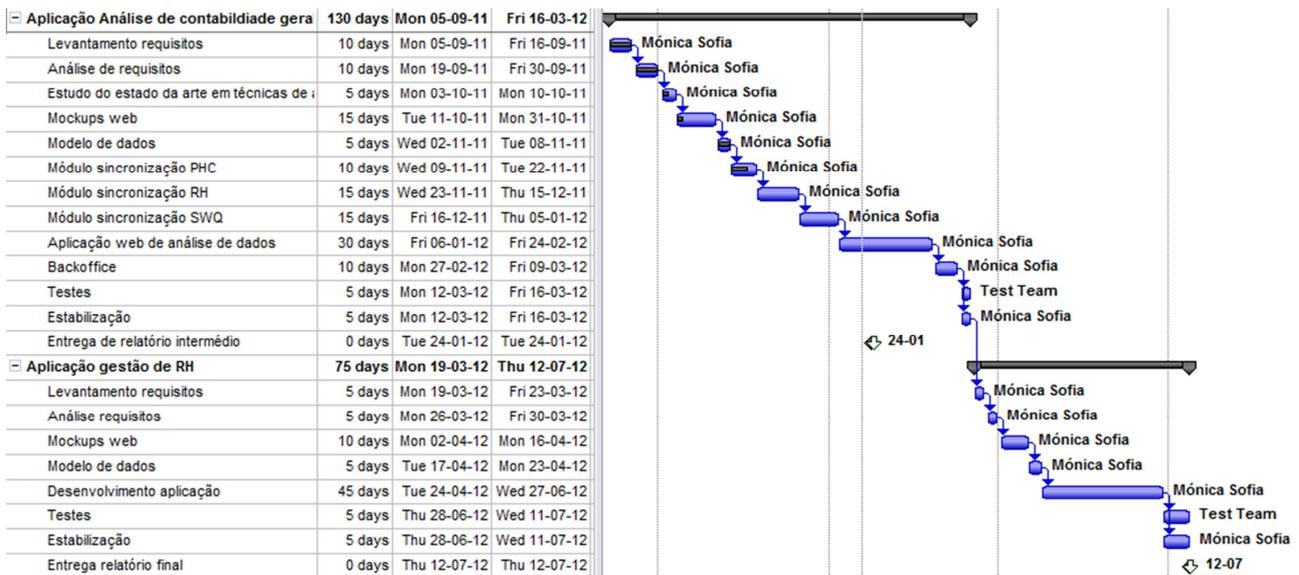


Figura 14 – Planeamento Inicial.

A Figura 14 ilustra o plano definido inicialmente para este estágio.

O planeamento inicial do estágio seria desenvolver uma plataforma de contabilidade, terminando a mesma a 16 de Março com uma duração de 130 dias. Começava com um levantamento de requisitos, seguido de uma análise do estado da arte e realização de *mockups web*.

Após todos os pontos de preparação para o desenvolvimento estarem realizados, seguiu-se a criação da base de dados. Inúmeras iterações seriam realizadas até se chegar ao modelo de dados correto.

Finalmente, o desenvolvimento poderia começar, existindo assim três módulos:

- PHC – criação do *web service* que acede à base de dados do PHC, e sincronização com o mesmo para importação dos dados;
- RH – criação do módulo de recursos humanos no *back office*, de modo a ser possível carregar os dados dos recursos humanos;
- SWQ – módulo de carregamentos dos dados de *software quality* no *back office*.

Uma vez todos os dados recolhidos, proceder-se-ia há criação da aplicação para a visualização dos dados – plataforma de contabilidade. Por fim, seria criado o *back office* juntando todos os módulos criados. De seguida seriam realizados todos os testes necessários. Após a criação da plataforma, seria feita uma aplicação de gestão de recursos humanos. Todos os passos da aplicação anterior seriam repetidos, adaptando a este caso em específico. Esteve previsto o desenvolvimento desta aplicação a partir do dia 19 de Março, tendo uma duração de 75 dias.

6.4 Planeamento Intermédio

Na figura abaixo é apresentado um planeamento do estágio.

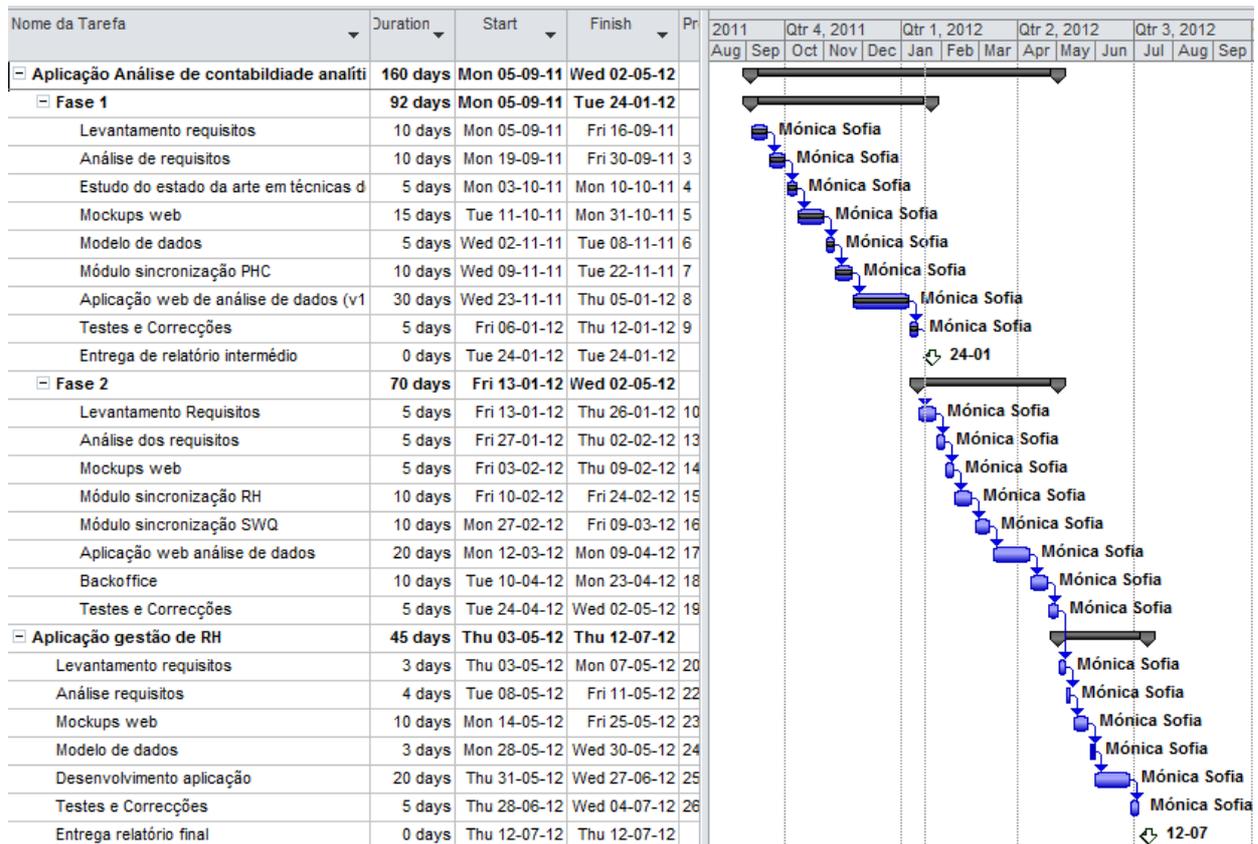


Figura 15 – Planeamento Intermédio.

Passaram a existir duas fases na implementação da plataforma inicial, devido ao facto de o cliente ter colocado prioridade máxima na contabilidade analítica, tendo sido associada esta à fase 1 como ilustrado na Figura 15. A mesma foi terminada no dia 12 de janeiro.

A segunda fase foi prevista para total dedicação aos restantes pontos:

- Recursos Humanos;

- *Software Quality;*
- *Tesouraria;*
- *Tableau de Bord.*

O tempo dedicado à plataforma de gestão de recursos humanos diminuiu, passando este a ter uma duração de 45 dias.

6.5 Planeamento Final

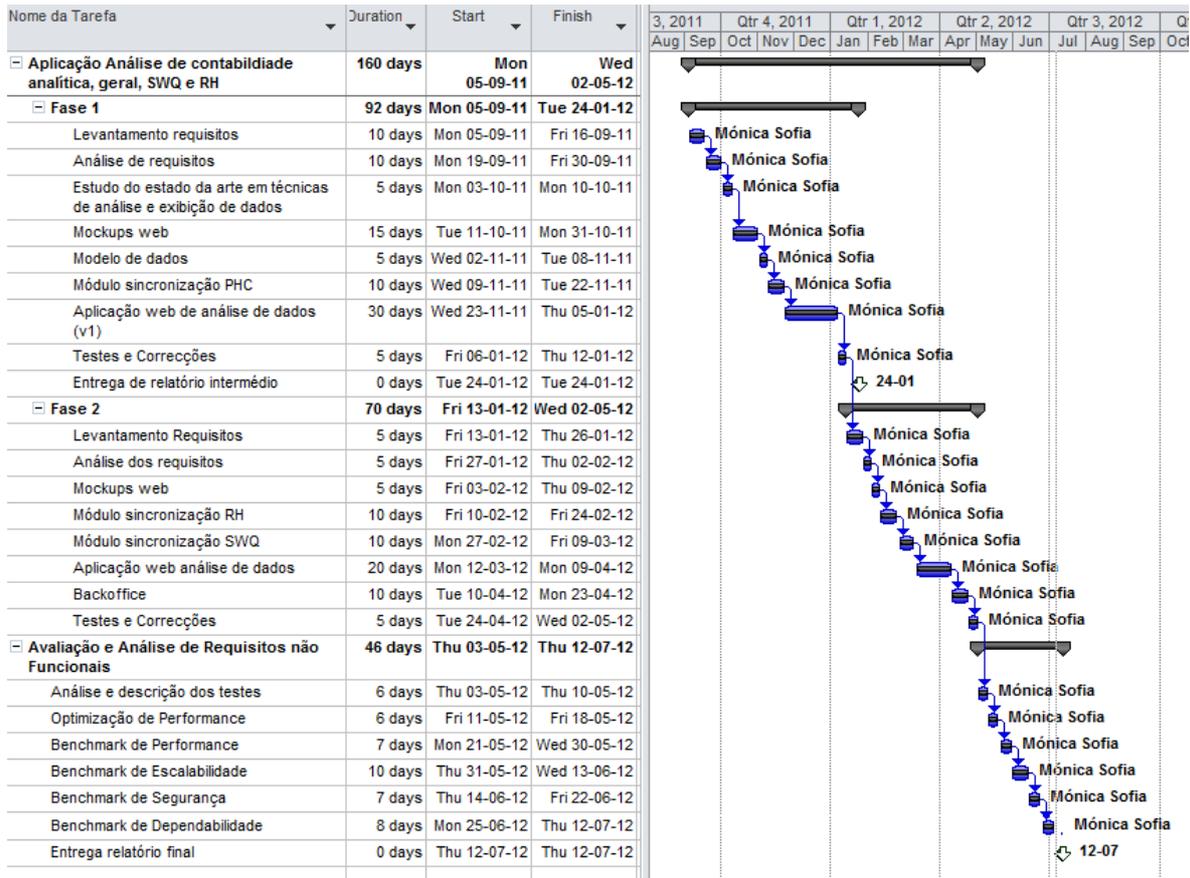


Figura 16 – Planeamento Final.

Nos últimos dois meses o planeamento sofreu uma nova alteração. A aplicação de gestão de recursos humanos deixou de existir e passou a ser feita uma análise e avaliação de requisitos não funcionais. Isto aconteceu por duas razões:

- A aplicação de gestão de recursos humanos passou a ser um estágio no ano 2012/2013, pois o tempo disponível para realizar a mesma no presente ano não permitiria fazer uma aplicação tão completa quanto o desejável.
- Numa das reuniões gerais da empresa o CEO esteve presente e a analisar a plataforma, colocando algumas questões sobre possíveis funcionalidades a acrescentar. Foi quando se analisaram questões como: “Quais os custos do departamento de SWQ?” ou então “Qual o custo gasto na equipa de desenvolvimento?”, que se constatou que não poderiam ser respondidas com a atual estrutura de contabilidade da empresa, bem como, todas as questões que detalhem a informação a um nível departamental. A estrutura atual apenas permite responder às

questões indicadas nos requisitos, como por exemplo: “Qual o custo da unidade x?” ou então “Qual a contribuição de cada cliente na unidade y?”, entre outras.

Como resultado da reunião o CEO afirmou que o desenvolvimento da plataforma deveria ser suspenso, o ser retomado quando a estrutura da WIT conseguir responder às questões mencionadas. Isto passará por reestruturar os centros de custo, existindo a possibilidade de se acrescentar mais um nível para permitir um maior detalhe, deixando de existir apenas a noção de unidade_cliente_projeto, onde se focava parte das funcionalidades. Desta forma o desenvolvimento foi interrompido no início de Maio, e devendo ser retomado no mês de Agosto altura em que será terminada a implementação do sistema de notificações e do relatório “*Tableau de Bord*”.

Serão feitas as alterações necessárias para que a plataforma fique de acordo com o novo sistema contabilístico, sendo necessário alterar a estrutura da base de dados, os métodos que acedem à mesma e ainda algumas questões de *interface*. Para além disto, será necessário alterar a forma como o PHC identifica os centros de custo (unidade_cliente_projeto) e ainda fazer alterações no sistema de submissão relatórios semanais da WIT, pois existem dados no PHC que dependem dele. Neste momento, a plataforma encontra-se acessível para uso do departamento de contabilidade.

7 Conclusões Finais e Trabalho Futuro

A nível pessoal, posso assegurar que o estágio foi, sem dúvida, o maior desafio que tive em mãos até hoje. Desde a adaptação a um ambiente empresarial a problemas que foram surgindo no dia-a-dia e ainda a relação com clientes, não foi de todo um ano fácil. No entanto é com orgulho que atingi o objetivo pretendido – melhorar os processos de contabilidade da empresa poupando tempo e aumentando produtividade ao departamento de contabilidade da mesma.

A nível de competências, o estágio contribuiu para o aumento dos meus conhecimentos técnicos e para um melhor desenvolvimento de *software*. Concluindo, fez de mim uma melhor Engenheira de *Software*, sem dúvida alguma.

No futuro, como já foi indicado na secção do planeamento, será realizada uma adaptação da plataforma à nova estrutura de contabilidade. Serão feitas alterações a nível de: estrutura da base de dados, métodos de acesso à mesma e ainda algumas questões de *user interface*. Tudo isto apenas poderá ser feito após as alterações do PHC e do processo de submissão de relatórios semanais. Serão ainda implementadas as funcionalidades em falta: *Tableau de Bord* e o sistema de notificações. Prevê-se que sejam necessários 15 dias para realizar todas as alterações e implementações.

8 Referências

1. “Accountancy”, Wikipedia [Online], <http://en.wikipedia.org/wiki/Accountancy>
2. “Conceitos de Contabilidade”, Nota Positiva [Online], http://www.notapositiva.com/trab_professores/textos_apoio/contabilidade/01concontabilidade.htm
3. Manifesto for Agile Software Development [Online], <http://agilemanifesto.org/>
4. “Apache Tomcat”, Apache Software Foundation [Online], <http://tomcat.apache.org/>
5. “Struts”, Apache Software Foundation [Online], <http://struts.apache.org/>
6. jQuery UI [Online], <http://jqueryui.com/>
7. jQuery BlockUI [Online], <http://jquery.malsup.com/block/>
8. Business Dictionary [Online], <http://www.businessdictionary.com/definition/business-intelligence-BI.html>
9. “JMeter.”, Apache Software Foundation [Online], <http://jmeter.apache.org/index.html>.
10. JProfiler. [Online], <http://www.ej-technologies.com/products/jprofiler/overview.html>.
11. YSlow [Online], <http://yslow.org/>.
12. S. Souders, *High Performance Web Sites*, O’Reilly Media, 2007.
13. A. Kalinovsky, *convert Java: Techniques For Decompiling, Patching, And Reverse Engineering*, Sams, 2004.
14. Wireshark [Online], <http://www.wireshark.org/>.
15. SNC Código de Contas [Online], http://www.cnc.min-financas.pt/SNC_projecto/SNC_CodigoContas_codigo.pdf