



Universidade do Minho

Universidade do Minho

Escola de Engenharia

Tiago Domingos Barbosa Araújo

**Exploração de Dados para Soluções de Retalho em
Business Intelligence**

Dissertação de Mestrado

Mestrado Integrado em Engenharia de Gestão e
Sistemas de Informação

Trabalho efetuado sob a orientação de:

Prof. Dr. Luís Paulo Reis (orientador)

Outubro de 2015

Agradecimentos

Este trabalho não seria possível de realizar sem o apoio, disponibilidade e paciência de inúmeras pessoas que me acompanharam nos últimos 5 anos.

Um agradecimento ao meu orientador Professor Luís Paulo Reis pelo acompanhamento, motivação e constante desafio na obtenção dos melhores resultados.

Aos meus diretores de curso Luís Paulo Reis e Luís Mota Pereira que não me fecharam porta do gabinete quando precisei de umas sessões de *coaching profissional*.

Aos meus professores que durante o percurso académico não agradei da forma que mereciam, que apesar de ser obrigação, ficavam privados da vida familiar para dar aulas ao pós-laboral.

À minha família pela disponibilidade, confiança incondicional, educação e valores que hoje preservo e que me ajudaram a chegar até aqui.

Aos meus amigos que me deram como desaparecido mas que perceberam e me apoiaram nestes primeiros passos da minha carreira.

Aos meus colegas de curso e do DSI, que hoje são amigos, e só eles sentiram comigo as dores das noites em branco, os feriados e domingos, mesmo quando nos privaram o acesso à universidade.

Aos meus colegas da Inovretail pela paciência e investimento na ajuda que disponibilizaram, principalmente aos meus superiores que confiaram em mim.

Abstract

Never before have organizations invest in both Business Intelligence technologies, mainly in the retail market. Business knowledge is increasingly recognized as a major competitive advantage point in an increasingly saturated market. With e-commerce becomes more and more difficult physical stores manage to attract public and convince him to make the purchase respective.

The Inovretail, a company focused on improving shopping environments to make them more profitable, has been making efforts in this direction. Seeks to give its customers tools to make the most attractive areas in order to influence the purchase. It does so not only implemented BI solutions that give the current state of the business, but also to study environmental variables projects, layout of sales outlets and forecasting metrics. Thus, any manager can make a decision based on reliable data of the past, present and future of your business to make a strategic action. The Customer Experience services are essentially supported by one of the products of Inovretail: the SEEPlus, trying to meet the need for specialized BI products for retail.

This project, motivated by this lack identified by multiple clients, passes by studying the SEEPlus, identify points of access to data, create cubes from the existing DW and create a plugin for Excel that allows users to create their own dashboards and share them. The choice of Excel is motivated by the desire known typical manager to see everything in the spreadsheet, but this being linked to SEEPlus at any time can update the tables and graphics with new information.

With the integration of SEEPlus in Excel expected to increase the number of users who connect to the system. While the perception of users when using a dashboard is that the refresh time may be higher than they want, in Excel, the new less experienced users may not have the same perception of what is an integrated DW in an Excel plugin. Thus it is challenging the current approach to SEEPlus architecture to address some performance issues in data access.

This document reflects the observation of all the variables that directly or indirectly influence the implementation of the project, including the study of technologies and methodologies used, state of the art and related work. It also presents the timing of proposed activities for the project, ending with future work for new features and improvements.

Resumo

Nunca como hoje as organizações investem tanto em tecnologias de Business Intelligence (BI), principalmente no mercado do retalho. O conhecimento de negócio é cada vez mais reconhecido como o maior ponto de vantagem competitiva num mercado cada vez mais saturado. Com o *e-commerce* torna-se cada vez mais difícil as lojas físicas conseguirem atrair público e convence-lo a realizar a respetiva compra.

A Inovretail, empresa focada na melhoria dos ambientes de lojas para torna-las mais rentáveis, tem vindo a fazer esforços nesse sentido. Procura dar aos seus clientes ferramentas que permitam tornar os espaços mais atrativos de modo a influenciar a compra. Fá-lo não só implementado soluções de BI, que dão o estado atual do negócio, mas também com projetos de estudo de variáveis ambientais, *layout* dos pontos de venda e previsão de métricas. Assim, qualquer gestor pode tomar uma decisão com base em dados fidedignos do passado, presente e futuro do seu negócio para tomar uma ação estratégicas.

Os serviços de Customer Experience são essencialmente suportados por um dos produtos da Inovretail: o SEEPlus. Este sistema consegue integrar qualquer fonte de dados dos clientes (*flat files*, *web services* ou base de dados) de modo a disponibilizar toda a informação relacionada com o retalho, vendas, *footfall*, temperatura, humidade, níveis de CO₂, consumo energético, ruído, *stocks*, entre outros. Apesar de lidar com vários processos da organização, o seu objetivo é funcionar como um DM para o retalho.

Completando o Data Warehouse (DW) estão várias interfaces de acesso aos dados, GUI, API, *web services*, não havendo forma de acesso direto aos dados em bruto. O GUI é um *web site* em *silverlight* com vários módulos (Dashboards, Operational Reports, Maps, Business Monitor, Alarms, User Management, Dashboard Management, Web Upload, Replenishment Attributes, entre outros). Alguns estão disponíveis na mobilidade no SEEMoblie para *Android* e *iPhone* e *Windows Phone*. A interface programática são *web services* que todos os módulos usam para realizar operações de CRUD (*create*, *read*, *update* e *delete*) no DW. Neste momento não existe nenhuma forma de aceder aos dados por uma conexão OLAP a um cubo ou direto ao DW, o que limita a utilização à lista de *dashboards* do produto.

Este projeto foi motivado por essa falta que foi identificada por vários clientes. A solução passou por estudar o SEEPlus, identificar pontos de acesso aos dados, criar

cubos a partir do DW existente e criar um *plugin* para o *Excel* que permita aos utilizadores criarem os seus próprios *dashboards* e partilhá-los. A escolha do *Excel* é motivada pelo conhecido típico desejo do gestor de ver tudo na folha de cálculo, mas estando esta ligada ao SEEPlus a qualquer momento pode atualizar as tabelas e gráficos com nova informação.

Com a integração do SEEPlus no Excel espera-se aumentar o número de utilizadores que se ligam ao sistema. Enquanto a perceção dos utilizadores quando usam um *dashboard* é que os tempos de refrescamento podem ser superiores ao que desejam, no Excel, os novos utilizadores menos experientes podem não ter a mesma perceção do que é um DW integrado num *plugin* do Excel. Portanto foi questionada a abordagem da atual da arquitetura do SEEPlus para colmatar alguns problemas de performance no acesso aos dados.

Este documento reflete a observação de todas as variáveis que direta ou indiretamente influenciam a implementação do projeto. Deste modo, contém o estudo de tecnologias e metodologias a utilizar, do estado da arte e trabalho relacionado na área e da própria filosofia da Inovretail. Apresenta ainda a calendarização das atividades propostas para o projeto, finalizando com os trabalhos futuros para novas funcionalidades e melhorias.

Palavras chave: Business intelligence, Data warehouse, OLAP, Excel plugin, Cloud, Azure

*A experiência é uma lanterna dependurada nas costas
que apenas ilumina o caminho já percorrido.*

Confúcio

Índice de Conteúdos

AGRADECIMENTOS	II
ABSTRACT	IV
RESUMO	V
ÍNDICE DE CONTEÚDOS.....	IX
ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABELAS.....	XVI
SIGLAS E ACRÓNIMOS.....	XVII
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	2
1.2 OBJETIVOS DE INVESTIGAÇÃO E DESENVOLVIMENTO PROFISSIONAL.....	3
1.3 ESTRUTURA DO TRABALHO	3
2 ABORDAGEM METODOLÓGICA.....	5
2.1 ESTRATÉGIA DE PESQUISA BIBLIOGRÁFICA	5
2.2 METODOLOGIA DE INVESTIGAÇÃO	6
2.3 QUESTÕES ÉTICAS E GESTÃO DE DADOS	7
2.4 ANÁLISE DE RISCOS.....	7
2.4.1 <i>Resultados</i>	8
3 ENQUADRAMENTO EMPRESARIAL	12
3.1 INOVRETAIL.....	12
3.2 SEEPLUS.....	13
3.2.1 <i>Arquitetura</i>	14
3.2.2 <i>Módulos</i>	16
3.2.3 <i>Dashboards</i>	18
3.3 SMARTMEASURE	20
3.4 CUSTOMER EXPERIENCE LAB.....	20
3.5 PROJETOS E SERVIÇOS	21
3.6 DATA EXPLORER	21
3.7 CONCLUSÃO.....	21
4 ESTADO DA ARTE	23
4.1 BUSINESS INTELLIGENCE	23
4.2 DATA WAREHOUSE	24
4.2.1 <i>Metodologias</i>	26

4.3	TECNOLOGIAS OLAP	28
4.3.1	<i>Cubos OLAP</i>	29
4.3.2	<i>Operações de Consulta</i>	29
4.3.3	<i>Arquitetura OLAP</i>	30
4.4	CLOUD.....	33
4.4.1	<i>Microsoft Azure</i>	33
4.5	PADRÕES DE ARQUITETURA PARA GUI	36
4.5.1	<i>MVVM (Model-View-ViewModel)</i>	36
4.5.2	<i>MVP (Model-View-Presenter)</i>	37
4.6	TRABALHO RELACIONADO.....	38
4.7	CONCLUSÃO.....	39
5	DATA EXPLORER	41
5.1	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	41
5.1.1	<i>Login</i>	41
5.1.2	<i>Repositório de Relatórios</i>	41
5.1.3	<i>Exploração de Dados em Cubos</i>	42
5.1.4	<i>Visualização de Relatórios</i>	42
5.1.5	<i>Template</i>	42
5.1.6	<i>Gestão de Relatórios</i>	42
5.1.7	<i>Partilha de Relatórios</i>	42
5.1.8	<i>Requisitos Não Funcionais</i>	43
5.2	FUNCIONALIDADES	43
5.2.1	<i>Menu de Navegação</i>	43
5.2.2	<i>Funcionalidades Chave</i>	44
5.2.3	<i>Exploração de Dados</i>	45
5.2.4	<i>Relatórios</i>	46
5.2.5	<i>Pivot table</i>	47
5.2.6	<i>Gestão de Relatórios</i>	49
5.2.7	<i>Formulários de Perguntas e Informações</i>	51
5.3	ARQUITETURA	52
5.4	DESENVOLVIMENTO.....	55
5.4.1	<i>Modelação dos Cubos OLAP</i>	55
5.4.2	<i>Integração e Processamento de Cubos</i>	63
5.4.3	<i>Plugin Excel em Windows Forms</i>	65
5.4.4	<i>Modelo de Dados</i>	69
5.4.5	<i>Assistente de Instalação</i>	72
5.4.6	<i>Conclusão</i>	78

6	CONCLUSÕES E PERSPETIVAS DE DESENVOLVIMENTO.....	80
6.1	CONCLUSÕES SUMÁRIAS	80
6.2	LIMITAÇÕES DO ESTUDO	81
6.3	TRABALHOS FUTUROS	81
6.3.1	<i>Refreshamento em Background.....</i>	82
6.3.2	<i>Identificar relatório Data Explorer</i>	82
6.3.3	<i>Trabalhar em offline</i>	83
6.3.4	<i>Relatórios com várias Folhas de Cálculo.....</i>	83
6.3.5	<i>Integração com plataformas documentais.....</i>	83
6.3.6	<i>Secções de relatório.....</i>	83
	REFERÊNCIAS BIBLIOGRÁFICAS	85
ANEXO A	SCRIPT WIX	89
ANEXO B	PLANO DE ATIVIDADES.....	90
EXECUÇÃO	ERRO! MARCADOR NÃO DEFINIDO.	

Índice de Figuras

Figura 1 - Metodologia de Investigação, Design science figura adaptada, (Vaishnavi & Kuechler, 2004)	7
Figura 2 - Produtos e Serviços da Inovretail	12
Figura 3 - Ilustração dos módulos do SEEPlus	14
Figura 4 - Camadas Principais de suporte à GUI	14
Figura 5 - Representação da Arquitetura do SEEPlus	15
Figura 6 - Representação da Arquitetura dos Serviços.....	16
Figura 7 - Exemplo de um dashboard do SEEPlus.....	18
Figura 8 - Exemplo de um dashboard das variáveis Atmosféricas do SEEPlus.....	19
Figura 9 - Exemplo de um report web sobre as vendas.....	19
Figura 10 - Arquitetura comum num sistema de BI	24
Figura 11 - Arquitetura de um sistema de BI	26
Figura 12 - Data warehouse organizacional	27
Figura 13 - DW organizacional alimentado por DM segmentados	28
Figura 14 - Exemplo de um esquema multidimensional	29
Figura 15 - Componentes gerais de um servidor ROLAP	31
Figura 16 - Componentes gerais de um servidor MOLAP.....	31
Figura 17 - Componentes gerais de um servidor HOLAP.....	32
Figura 18 - Cloud Computing Benefits (Win-Pro Consultancy, 2012).....	33
Figura 19 - MVVM Pattern	36
Figura 20 - Objetos e Relações no <i>pattern</i> MVP	38
Figura 21 - Data Explorer - Barra de Navegação	43
Figura 22 - Data Explorer - Ecrã de Login.....	44
Figura 23 - Data Explorer - Menu de Logout.....	44
Figura 24 - Data Explorer - Áreas de Trabalho e Menus	45
Figura 25 - Data Explorer - Lista dos factos disponíveis por categoria	45
Figura 26 - Data Explorer - <i>Pivot table</i> e <i>Interface</i> para criar pesquisa.....	46
Figura 27 - Data Explorer - Carregamento de Relatório	47
Figura 28 - Data Explorer - Refrescamento dos dados.....	47
Figura 29 - Data Explorer - Formatação de <i>Pivot table</i>	48
Figura 30 - Data Explorer - Menu Autofit columns e Preserve cell formating.....	48
Figura 31 - Data Explorer - Guardar Relatório.....	49

Figura 32 - Data Explorer - Formulário para guardar relatório	49
Figura 33 - Data Explorer - Lista de relatórios existentes	50
Figura 34 - Data Explorer - Editar relatório existente	51
Figura 35 - Data Explorer - Mensagem de confirmação	51
Figura 36 - Data Explorer - Mensagem de sucesso	52
Figura 37 - Nova arquitetura do SEEPlus com o Data Explorer	52
Figura 38 - Estrelas com factos em várias granularidades diferentes.....	53
Figura 39 - Arquitetura SQL Server com SSAS (Denny Lee, 2007)	54
Figura 40 - Assistente de Criação de Máquina Virtual Windows Server.....	55
Figura 41 - Criação de nova base de dados no SSAS.....	56
Figura 42 - Componentes do projeto do SASS.....	57
Figura 43 - Estrutura do Cubo de Store Analytics	60
Figura 44 - Estrutura do Cubo de Employee Analytics.....	61
Figura 45 - Estrutura do Cubo de Operation Analytics	62
Figura 46 - Hierarquia da dimensão Estrutura Organizacional	62
Figura 47 - Alguns conectores do SEEPlus em destaque o ProcessOLAPConnector ...	64
Figura 48 - Algoritmo de processamento dos cubos	64
Figura 49 - Novo projeto Excel 2013 Addin no Visual Studio 2015	66
Figura 50 - Objeto principal de inicialização do plugin do Excel	66
Figura 51 - Módulos do Data Explorer.....	67
Figura 52 - Objetos View do Pattern MVP	68
Figura 53 - Objetos Presenter e Model Data do Pattern MVP	69
Figura 54 - Modelo de dados para o plugin Data Explorer	70
Figura 55 - Modelo de dados completo com as tabelas de sistema	72
Figura 56 - Equação utilizada para a seleção da ferramenta de assistente de instalação	73
Figura 57 - Instalação do Assistente WiX.....	75
Figura 58 - Criação de um novo projeto WiX.....	75
Figura 59 - Estrutura de pastas e ficheiros Xml do WiX.....	76
Figura 60 – Assistente de Instalação a) Página Inicial b) Página de seleção da pasta de instalação	77
Figura 61 - Assistente de Instalação – a) Página de confirmação da instalação, b) Página de estado	78
Figura 62 - Assistente de Instalação – a) Página de instalação com sucesso, b) Página de desinstalação	78

Figura 63 - Calendarização da Tese de dissertação	90
Figura 64 - Calendarização das tarefas executadas na Tese Dissertação	91

Índice de Tabelas

Tabela 1 - Riscos associados a implementação do Data Explorer	9
Tabela 2 - Riscos decorridos ao longo da execução do plano	11
Tabela 3 - Módulos e funcionalidades do <i>SEEPplus</i>	17
Tabela 4 - Siglas que contituem as métricas do cubo.....	59
Tabela 5 - Descrição dos elementos do modelo de dados	71
Tabela 6 - Comparativo entre Ferramentas de Assistente de Instalação	73

Siglas e Acrónimos

API	Application Programming Interface
BI	Business Intelligence
DaaS	Database as a Service
DM	Data Mart
DTU	Database Throughput Unit (SQL Azure)
DW	Data Warehouse
ETL	Extract, Transform and Load
GUI	Graphical User Interface
HOLAP	Híbrido OnLine Analytic Processing
IaaS	Infrastructure as a Service
IIS	Internet Information Services
KPI	Key Performance Indicators
MOLAP	Multidimensional OnLine Analytic Processing
MVP	Model-View-Presenter
MVVM	Model-View-ViewModel
OLAP	OnLine Analytic Processing
OLTP	OnLine Transaction Processing
PaaS	Plataform as a Service
ROLAP	Relational OnLine Analytic Processing
SaaS	Software as a Service
SQL	Structured Query Language
SSAS	SQL Server Analysis Services
SSIS	SQL Server Integration Services
VM	Virtual Machine
XML	eXtensible Markup Language

1 Introdução

Num mercado cada vez mais competitivo, existem muitos fatores que diferenciam as organizações. Estratégias de marketing e posicionamento no mercado, cultura, fatores influenciam a participação relativa e crescimento no contexto de mercado em que estão inseridos (Porter, 1999). Todos estes fatores podem ser bem conseguidos com a experiência de um gestor brilhante. Contudo, quando a dimensão de negócio atinge um nível em que capacidade humana não consegue atingir, passa a ser imprescindível o uso de tecnologia para recolha, tratamento e análise de indicadores de negócio.

Num exemplo do mercado de retalho, estrategicamente o negócio poderá interagir com todo o ciclo logístico, desde, a fonte na compra da matéria-prima, transformação, abastecimento, armazenamento e venda ao público para o cliente final. Neste contexto as relações entre entidade de negócio aumentam diariamente influenciando naturalmente a complexidade da hierarquia organizacional, dos processos e da memória organizacional de dados.

No passado pode-se assistir ao crescente do volume de dados, de cada vez mais fontes e com tipos de dados heterogéneos, e a tendência é que todas estas dimensões cresçam de forma exponencial. Com tais características fica muito difícil para uma organização ter um sistema centralizado e normalizado que possibilite uma visão global e interligada de todos os processos de negócio. Os DW e outros sistemas de BI são a solução mais comum e a que tem apresentado mais resultados positivos.

Hoje em dia já não é suficiente ter o estado do negócio em tempo real para suporte das decisões do momento seguinte, é necessário estar um passo mais à frente. Os negócios mais inteligentes contam com assistentes de suporte à decisão que preveem o impacto de um ou mais acontecimentos numa determinada variável e que sugerem ou tomam medidas de mitigação ou prevenção que antecipadamente previnem o risco de negócio.

Com a falta de especialização em soluções de BI produtivizadas para o retalho, a Inovretail, empresa de consultoria e desenvolvimento de *software*, oferece uma suite empresarial de monitorização e avaliação do estado atual do negócio em tempo real, com foco no mercado de retalho alimentar e não alimentar. A sua missão é melhorar os ambientes das lojas, alinhando estratégias de negócio, marketing, atmosféricas e sonoras, para as tornar mais agradáveis para os clientes finais, aliciando-os na compra. Esta *suite* denominada SEEPlus, é um produto comercializado num modelo de SaaS.

Os 3 canais (web, mobile e desktop) disponíveis no SEEPlus, encontra-se funcionalidades como *dashboards*, indicadores e relatórios de negócio, alarmes de negócio, canais de auditorias de loja, sistema de inquéritos internos, previsão de métricas de negócio (vendas, número clientes, stocks), otimização de *stock* e monitorização de ambientes atmosféricos. Na componente web estão presentes as principais funcionalidades do SEEPlus, detalhadas no decorrer deste documento, já no mobile apenas alguns *dashboards* e outros módulos estão acessíveis ao utilizador. Por fim, o canal desktop apenas se cinge a pequenos *dashboards* para indicadores relacionados com os locais físicos de venda do retalho.

Com a diversidade de dados (vendas, produtos, clientes, stocks, temperatura e humidade dos locais, eventos atmosféricos) e fontes, sensores atmosféricos, contadores, POS (*point of sale*), ERPs (*Enterprise Resource Planning*), torna-se difícil estabilizar o número de *dashboards* que cubram todas as necessidades de negócio de todos os clientes da Inovretail. Por muito que o produto seja direcionado para o retalho, cada cliente tem as suas especificidades de negócio, e torna-se difícil apenas com um produto cobrir todas as possibilidades.

1.1 Motivação

Inspirada na intenção de responder a uma necessidade de negócio, é necessário um ato de engenharia para projetar e implementar uma solução de BI dinâmica de exploração de dados e criação de relatórios, para vários tipos de utilizadores (gestores, consultores, administradores, *power user* e outros).

Este trabalho afigura-se um desafio aliciante tem termos profissionais e alavanca para um início de carreira, exigindo não só a conceção, mas o desenvolvimento de uma solução a partir do zero. Por a base ser um produto e não um projeto de *software*, incumbirá a preocupação de encontrar uma arquitetura suficientemente genérica para todo o tipo de clientes do SEEPlus.

O facto de o projeto incluir todo o ciclo de desenvolvimento de uma solução de BI, desde a colheita, tratamento, carregamento e apresentação dos dados, requer a revisitação de muitos conhecimentos adquiridos ao longo de todo o mestrado, paradigmas da programação, boas práticas, base de dados, *big data*, sistemas inteligentes e até marketing para potenciar a comercialização de um produto.

Finalmente a circunstância de se encontrar num ambiente empresarial que obrigará a que o produto final tenha de fazer dinheiro, isto ajudará a focar nas necessidades do cliente para o produto final não falhe a expectativa.

1.2 Objetivos de investigação e desenvolvimento profissional

O trabalho de investigação que se desenrola tem como objetivo desenvolver um pacote de *software* para exploração de dados provenientes de DW e cubos OLAP. Tendo em conta o desenvolvimento do artefacto de *software*, foram definidos os seguintes objetivos de investigação:

1. Analisar a arquitetura do SEEPlus e propor melhorias por forma melhorar a performance e compatibilidade com outros módulos internos ou sistemas externos;
2. Comprovar que é possível tornar o *Excel* mais usável com *plugins* específicos para uma determinada função;
3. Implementar um *plugin* para o *Excel* que explore dados de um DW e que tenha uma boa usabilidade para o utilizador.

Compreende-se, portanto, que o atual trabalho é acima de tudo um trabalho de desenvolvimento e estudo da arquitetura do SEEPlus, que propõe melhorias, e desenvolve um novo módulo de exploração de dados, com o objetivo de satisfazer os objetivos. Adicionalmente pretende-se realizar um projeto *end-to-end* desde o planeamento, conceção, desenvolvimento e implementação num cliente real, com toda a responsabilidade de um projeto desta natureza acarreta, mais difícil de realizar durante o Mestrado.

1.3 Estrutura do trabalho

O documento está organizado na sequência das atividades realizadas no âmbito da presente dissertação de mestrado.

O capítulo 1 dizendo respeito à introdução, reflete sobre o tema abordado, com os subcapítulos: motivação - que deu inspiração à escolha do problema, objetivos de investigação – como o nome indica são os objetivos da investigação preliminar, abordagem de investigação - que expõe o âmbito sobre as técnicas e métodos utilizados nas atividades a realizar, e finalmente, o presente subcapítulo que descreve a estrutura geral do documento.

O capítulo seguinte aborda a estratégia de pesquisa bibliográfica selecionada, bem como a estratégia de investigação que é seguida, baseado no modelo *Design Science*, seguindo-se as questões éticas do projeto e a respetiva lista de riscos.

O terceiro capítulo faz o enquadramento deste projeto no ambiente empresarial que se encontra, descrevendo a empresa o seu mercado, ambiente, produtos e serviços. Termina com uma descrição do SEEPlus, produto no qual este projeto terá interação, bem com a arquitetura, módulos e sistemas externos.

O quarto capítulo apresenta-se como estado da arte dos principais temas abordados na investigação. São ainda analisadas algumas tecnologias que são à cabeça restritas pelos requisitos do cliente.

O quinto capítulo é dedicado às principais atividades do desenvolvimento do projeto, apresentando os requisitos funcionais e não funcionais seguido detalhe e ilustração das funcionalidades, a arquitetura com os pontos mais importantes e por fim todo o processo de implementação com os cubos OLAP, o *plugin* Excel, a integração com o motor do SEEPlus e o assistente de instalação.

O sexto capítulo expõe-se um pouco das conclusões e perspectivas de desenvolvimento sobre todas as etapas deste projeto, desde a investigação da pré-dissertação, dissertação, os trabalhos futuros e melhorias e ambiente empresarial vivido.

Por último, um capítulo com as referências bibliográficas que foram meio de conhecimento para a elaboração do presente documento.

2 Abordagem Metodológica

Para alcançar os objetivos anteriormente apresentados, será seguida uma abordagem de investigação baseada, principalmente, no contato direto com o SEEPlus e com os colaboradores da Inovretail que têm o conhecimento desta plataforma, usando técnicas de levantamento de funcionalidades e modelos de documentação de arquiteturas.

Numa fase inicial após a revisão de literatura, começou-se por documentar e identificar os módulos do SEEPlus candidatos a se ligarem com o Data Explorer, no sentido de se perceber como este dependerá deles e onde se encaixa dentro da arquitetura existente. É fundamental focar na autonomia do Data Explorer face aos outros módulos para que na evolução natural do SEEPlus não seja necessário sempre novos desenvolvimentos para garantir retro compatibilidade entre o Data Explorer e as novas versões do SEEPlus. Este será um bom ponto de partida para a análise da arquitetura do SEEPlus e das possíveis melhorias a serem introduzidas.

Numa fase posterior, após a análise do esforço real para desenvolver o Data Explorer, serão realizadas comparações entre os requisitos do Data Explorer e especificações de soluções idênticas. Pretende-se identificar funcionalidades transversais, boas práticas de usabilidade e funcionalidades não cobertas pelos requisitos anteriormente identificados que possam enriquecer o Data Explorer.

Antes do desenvolvimento propriamente dito, será documentada uma arquitetura candidata que será apresentada ao gestor de produto do SEEPlus com o objetivo de alinhar os requisitos identificados para o Data Explorer para o que é o *road map* do SEEPlus.

2.1 Estratégia de Pesquisa Bibliográfica

Segundo Webster e Watson, uma revisão de literatura de qualidade dever-se-ia centrar palavras-chave identificadas a partir de conceitos da revisão de literatura (Webster & Watson, 2002). A mesma deve ter como base as seguintes tarefas:

1. Identificar artigos relevantes das palavras-chave em questão;
2. Procurar referências relevantes citadas nos artigos identificados na tarefa 1 (*Go backward*);
3. Procurar referências relevantes que cite os artigos identificados na tarefa 1 (*Go forward*).

As palavras-chaves usadas foram: *Business intelligence, Data warehouse, OLAP, Excel plugin, Cloud, Azure.*

Contudo, apesar de conhecer as recomendações do autor e do aconselhamento do orientado da dissertação, não foi possível seguir a seleção dos artigos seguindo a tarefa 2 e 3. Neste sentido a pesquisa foi realizada com seleção dos artigos mais referenciados, segundo as palavras-chave da revisão de literatura, e cuja análise preliminar do *abstract* fazia sentido para a pesquisa em curso. Complementar a esta estratégia, para temas mais práticos, onde a informação pode estar mais suscetível de confidencialidade, por exemplo: manuais, especificações de ferramentas, casos de estudo, usou-se o motor de busca da Google, fazendo sempre referência à palavra-chave/ferramentas + *references*. Para aumentar a probabilidade de encontrar *links* de caris mais científico, anexou-se a palavra-chave pdf no motor de busca.

Os repositórios consultados foram: Scopus, Web of Knowledge / ISI Web of Science e o Google Scholar. Complementar, para encontrar bons exemplos de dissertações e encontrar inspiração, usou-se os repositórios da Universidade do Minho (RepositóriUM) e Universidade do Porto (Repositório-Aberto).

2.2 Metodologia de Investigação

Vaishnavi e Kuechler, definem que inventar é trazer à existência. Sendo este projeto de natureza de inovação e conceção, em que não existe conhecimento para a maioria das tarefas, caso contrário seria um projeto rotineiro, será indispensável a realização de pesquisas científicas que preencham essas lacunas. Contudo o trabalho que se realizará da implementação gerará novas necessidades de conhecimento que produziram novas ações de implementação, tornando este num ciclo vicioso. Por esta razão o plano do projeto e outras ideias podem sofrer alterações. A metodologia proposta pelos autores, *Design Science Research Methodology Process Model*, ajuda a olhar para as várias fases da investigação de forma pragmática, questionando sempre o salto para a fase seguinte ou a revisão das fases anteriores, motivado por um problema que afete o seguimento normal do projeto.

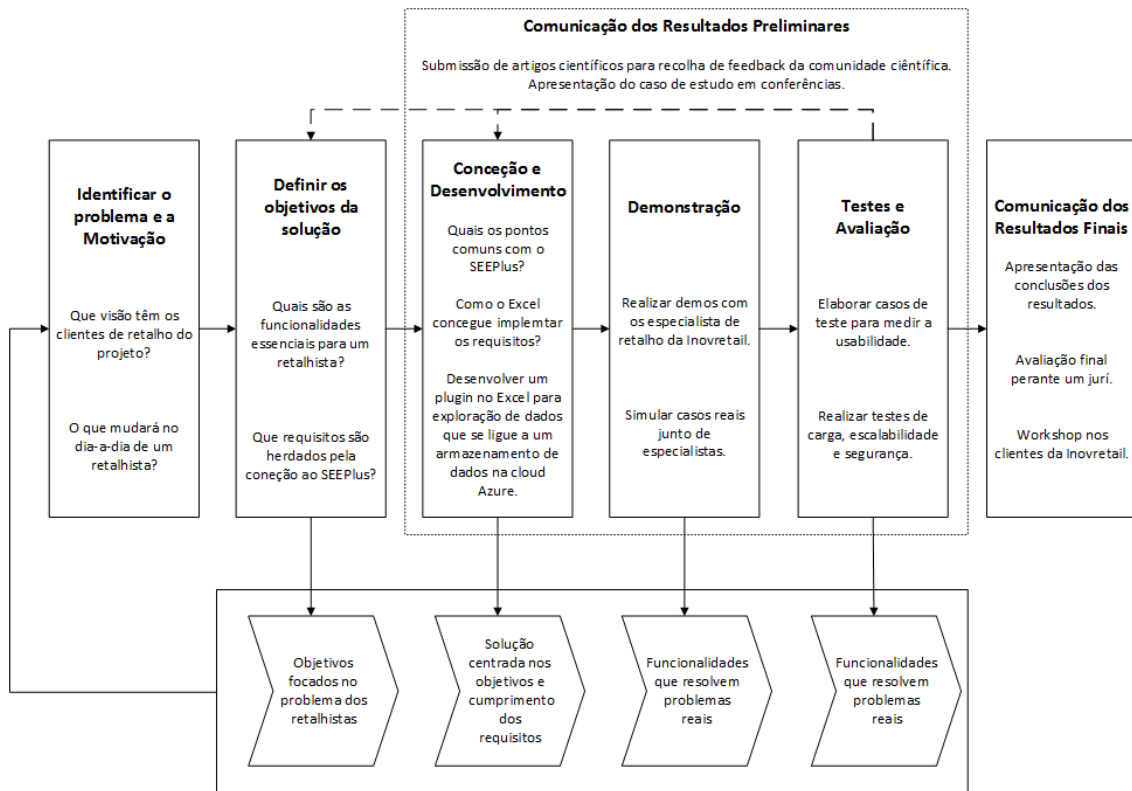


Figura 1 - Metodologia de Investigação, Design science figura adaptada, (Vaishnavi & Kuechler, 2004)

2.3 Questões Éticas e Gestão de Dados

Pela natureza empresarial do projeto, a confidencialidade incorre não só nos dados que serão disponibilizados para desenvolvimento, testes e validação da solução, como também informação sobre os processos internos, informações cedidas nas reuniões, e principalmente o código desenvolvido.

Os dados usados para desenvolvimento, testes e validação, serão reais de clientes atuais, que a Inovretail mantém um contrato de confidencialidade. Por essa razão os dados de clientes não podem ser acedidos para além das pessoas envolvidas no projeto, no caso o aluno que desenvolve o atual projeto e o seu orientador.

Apesar a sensibilidade dos dados, a Inovretail prefere o uso dos dados reais pela dimensão e heterogeneidade que eles tomam, por ser impraticável gerar tantos dados fictícios como aqueles que existem nos seus clientes.

2.4 Análise de Riscos

Para além dos riscos comuns a qualquer projeto e às pessoas envolvidas como doença, indisponibilidade, falta de tempo ou de natureza da gestão do projeto, foram

identificados os riscos mais específicos sobre a tecnologia, alteração de requisitos ou a confidencialidade de alguns termos. Estes riscos podem ser analisado na Tabela 1.

Posteriormente são apresentados as ocorrências que causaram alterações ao plano inicial no que toca ao tempo previsto para cada atividade ou situações que resultaram em novas tarefas para colmatar falhas no plano inicial. Este resultado pode ser analisado na Tabela 2.

2.4.1 Resultados

Dos riscos que se vieram a verificar, foram registadas as principais ocorrências na Tabela 2.

Na primeira ocorrência, verificou-se a inflexibilidade dos componentes da Microsoft para o desenvolvimento da GUI. A alternativa dos componentes da Telerik foi potenciadas pela parceria com a Inovretail, que permitiu diminuir a probabilidade e a seriedade do risco em 50%.

Com o entusiasmo dos gestores da Inovretail à medida que iam vendo o progresso no trabalho, espletou uma série de ideias novas que resultaram em novos requisitos. Isto obrigou à utilização de novas tecnologias. Isto obrigou a um esforço adicional compensado em horas de trabalho, não planeadas inicialmente. Isto resultou no sucesso da implementação dessas funcionalidades mas não mitigou a recorrência do risco.

Era do conhecimento que a documentação era escassa e que isso poderia atrasar a análise de alguns componentes do SEEPlus. Não sendo possível diminuir significativamente a probabilidade de ocorrência, apenas se optou por outras vias como o conhecimento dos programadores e outros envolvidos no SEEPlus. No entanto podemos dizer que o impacto diminuiu, consequência do conhecimento já adquirido.

O sucesso da utilização de frameworks externas depende da documentação ou formação que é disponibilizada. A Telerik também tem algumas lacuna essa área, o que já era do conhecimento da Inovretail. No entanto aquela era a única alternativa, pela experiência interna e a parceria já existente. O risco registado foi mitigado por acções de estudo mas que não conseguiram diminuir impacto ou a probabilidade. Apenas a experiência adquirida ao longo do projeto conseguiu evitar caminhos que ressaltassem em novas ocorrências.

No geral os problemas registados foram todos resolvidos da forma desejada, mas nem sempre foi possível mitigar na totalidade ou diminuir a probabilidade de recorrências.

Tabela 1 - Riscos associados a implementação do Data Explorer

ID.	Descrição	Probab. [1-5] (P)	Impacto [1-5] (I)	Seriedade [1-25] (S=P*I)	Ação Atenuante	Probab. após AA (P')	Impacto após AA (I')	Seriedade após AA (S'=P'*I')
1	Incompatibilidade ou não presença de alguma funcionalidade do <i>Excel</i> face aos requisitos.	3	5	15	Como não é solução mudar de ferramenta, será necessário deixar de lado o requisito que não pode ser implementado, isto se não for possível resolver a incompatibilidade.	2	5	10
2	Alteração dos requisitos e consequente incompatibilidade com o experimento científico corrente	1	5	5	Negociar um novo plano de trabalhos e tarefas, desde que não alterem o tema e conceitos iniciais da investigação	1	5	5
3	Dificuldades de desenvolvimento das tecnologias desconhecidas	2	3	6	Realizar a devida formação	1	3	3
4	Fuga de informação	2	4	8	Usar informação confidencial apenas nas máquinas de desenvolvimento usadas e em caso de transporte num armazenamento externo, usar um sistema de ficheiros encriptado.	1	2	2
5	Fuga de informação e consequente rescisão de contrato entre a Inovretail e o seu cliente.	2	5	10	Assumir a responsabilidade perante o cliente da Inovretail,			
6	Rescisão de contrato entre o aluno e a Inovretail	1	5	5	Negociar a continuidade do projeto fora da Inovretail para dar finalidade à conclusão da tese de mestrado.	1	2	2
7	Carência de informação disponibilizada pela Inovretail	3	4	12	Marcar presença em reuniões de contexto do SEEPlus que o aluno não é obrigado a frequentar e realizar reuniões extras semanal do ponto da situação.	2	3	6

Abordagem Metodológica

ID.	Descrição	Probab.	Impacto	Seriedade	Ação Atenuante	Probab.	Impacto	Seriedade
		[1-5] (P)	[1-5] (I)	[1-25] (S=P*I)		após AA (P')	após AA (I')	após AA (S'=P'*I')
6	Dificuldade de acesso à infraestrutura do SEEPlus	4	4	16	Pedir mais permissões do SEEPlus.	2	3	6
7	Alteração dos requisitos e consequente incompatibilidade com o experimento científico corrente	1	5	5	Implementar os requisitos que rejeitados mais os novos que substituíram os rejeitados.	1	4	4

Tabela 2 - Riscos decorridos ao longo da execução do plano

ID Risco Associado	Causa	Consequência	Ação Atenuante Aplicada	Probab. após AA (P')	Impacto após AA (I')	Seriedade após AA (S'=P'*I')
1	Nem o Excel nem a <i>framework</i> da Microsoft disponibiliza componentes necessariamente flexíveis para a conceção de GUI amigáveis para o utilizador.	O utilizador terá de fazer bastantes cliques para executar tarefas simples porque os componentes gráficos não permitem muitas personalizações. Alguns ecrãs teriam mais botões e por isso mais informação não útil.	Utilizar componentes desenvolvidos por entidades externas que não os componente nativos da Microsoft. Os componentes utilizador foram os <i>Windows Forms</i> da Telerik para o qual a Inovretail já tinha adquirido em licenças anteriores. Isto permitiu personalizar componentes e até alterar o <i>layout</i> de alguns ecrãs.	1	5	5 (-50%)
2	Por força de novos requisitos foi necessário usar tecnologias como SFTP e <i>wizards</i> de Instalação e integração com o Motor ETL do SEEPlus. Estes novos requisitos tem como força o foco do projeto em ser comercializado.	Estas tarefas extras necessitaram de mais tempo de estudo das tecnologias que até aquele momento não tinham sido previstas. Teve um efeito derrapante nos tempos de execução.	Para combater o atraso face ao inicialmente planeado houve um esforço adicional de horas de trabalho durante as semanas em que o plano sofreu alterações. O plano manteve-se, até porque o detalhe inicial não especificava tecnologias nem micro-tarefas.	1	5	5 (0%)
3	O aparecimento de novos requisitos obrigou ao estudo do Motor ETL do SEEPlus.	Com a pouca documentação existente tornou-se extenso olhar apenas para código.	Pedir ajuda às pessoas competentes que desenvolveram o Motor ETL e fazer reuniões para juntar as partes envolvidas no projeto.	1	2	2 (-33%)
7	Inexistência ou não disponibilização da informação de alguns módulos do SEEPlus. Algumas <i>frameworks</i> usadas como a Telerik tinham documentação um pouco pobre para iniciantes.	Obrigatoriedade de fazer tentativas erros no uso de SDK e API externas até conseguir descodificar o que a documentação pretendia explicar mas não o conseguia com sucesso.	Tentou-se obter alguma documentação a pedido exclusivo mas sem sucesso, por isso não possível executar uma ação atenuante eficaz. Usar API com documentação mais rica nunca pode ser uma alternativa.	2	3	6 (0%)

3 Enquadramento Empresarial

Dado o contexto empresarial desta tese de mestrado, será dada uma introdução mais aprofundada à empresa em questão e a sua oferta de mercado. Será dado foco no principal produto, relevante para esta tese.

3.1 Inovretail

A Inovretail é uma *startup* inserida no parque tecnológico de Ciência e Tecnologia da Universidade do Porto (UPTEC), fundada em 2010 por 3 especialistas da área de retalho (alimentar e não-alimentar) (Inovretail, 2014; Visão, 2012).

Assume-se como uma empresa de inovação tecnológica, totalmente focada na pesquisa e desenvolvimento de soluções destinadas a melhorar a experiência do cliente das lojas de retalhistas.

A missão da empresa é encontrar soluções que ajudam os retalhistas a criar ambientes de lojas mais atraentes, dinâmicos e orientados para o cliente, melhorando assim a satisfação do cliente e potencial retorno. Para isso, procura novos paradigmas na forma como os clientes interagem com o espaço de retalho e seu ecossistema sensorial, proporcionando ferramentas exclusivas para ajudar nas decisões de negócio baseadas em fatos sobre a estratégia de ambiente.

Desde o início que começou a trabalhar com grandes *players* do retalho (Salsa e Levis), o que alavancou o crescimento da empresa financeiramente e intelectualmente, muito devido aos gestores da empresa, pela sua experiência na área do retalho, gestão e produto.

Conta já com um leque de produtos e serviços que não cobre apenas alguns nichos de distribuição ou vestuário, mas também segmentos de massas.

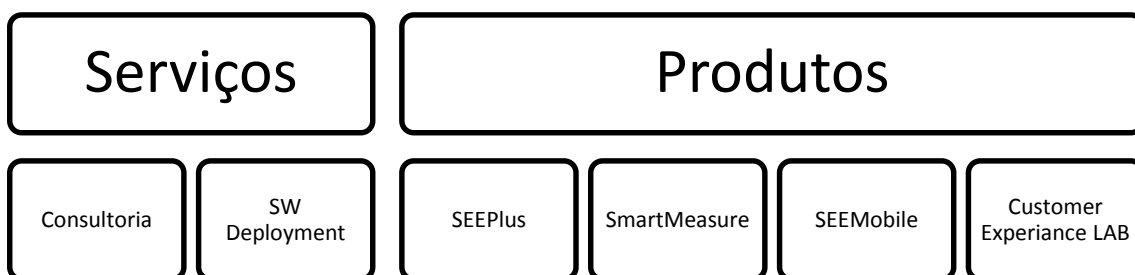


Figura 2 - Produtos e Serviços da Inovretail

3.2 SEEPLUS

SEEPlus, produto core, é uma solução completa de *customer experience management*, que oferece aos retalhistas uma plataforma única com a capacidade de implementar e gerir, com eficácia, ambientes de venda e estratégias atmosféricas de negócios, com o objetivo de melhorar a notoriedade da marca, satisfação do cliente, o desempenho de vendas e custos operacionais.

Não é um sistema operacional, pelo que todos os dados recebidos dos pontos de vendas são tratados e transformados para serem integrados nas bases de dados, que posteriormente os utilizadores podem consultar e analisar os próprios *dashboards*. Apesar de se apresentar como um DW, algumas funcionalidades geram e consomem dados que claramente são dados operacionais, como é o caso dos questionários

As principais funcionalidades do produto core são:

- Medir os impactos atmosféricos da loja sobre o comportamento do cliente o respetivo impacto na receita;
- Definir e monitorar ativamente a execução da estratégia atmosférica;
- Compreender a eficácia da oferta de produtos de valor para o cliente;
- Definir e medir iniciativas do cliente face às campanhas de Marketing;
- Analisar, prever e o impacto das campanhas de Marketing nas vendas;
- Medir e compreender a configuração da loja e a eficiente colocação de produtos;
- Comparar as diferentes configurações das lojas;

O produto tem uma arquitetura modular, onde cada módulo responde a um conjunto de funcionalidades paralelas que se ajustam às necessidades do cliente. O conjunto de módulos é abaixo ilustrado e posteriormente descrito. Cada um dos módulos contem funcionalidades que vêm complementar objetivo principal do SEEPlus.



Figura 3 - Ilustração dos módulos do SEEPlus

3.2.1 Arquitetura

O SEEPlus tem uma arquitetura em camadas simples: Interface, Negócio e Dados, fortemente orientada a *web services* e assente numa infraestrutura em *cloud*, Figura 4 e Figura 5. Esta é uma visão geral da solução pois tem outros módulos que surgiram em contextos e com tecnologias distintas, que por força da necessidade herdaram vários padrões de desenvolvimento e vários pressupostos arquiteturais.



Figura 4 - Camadas Principais de suporte à GUI

O SEEPlus não cinge apenas às camadas representadas na figura anterior, até porque tem mais do que um órgão mecânico de sistema, entenda-se órgão mecânico um órgão que realiza trabalho não por vontade própria mas a comando de alguma pessoa, no caso máquinas ou computadores. Para perceber melhor pode-se verificar na Figura 5, da direita para a esquerda, os órgãos mecânicos pelo qual os dados passam desde as recolha nas fontes de dados até às aplicações cliente.

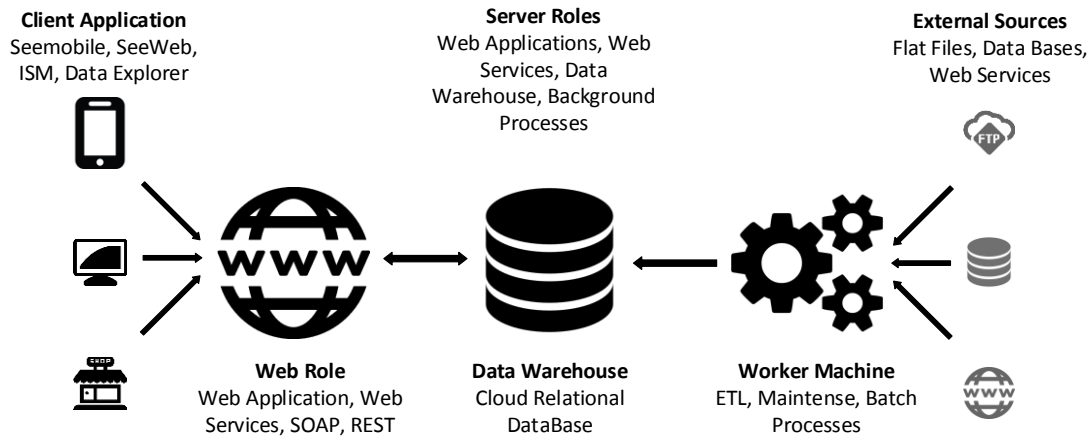


Figura 5 - Representação da Arquitetura do SEEPlus

Os dados podem chegar através de *Flat Files*, FTP, base de dados de vários formatos ou até mesmo *web services* desde que seja garantido a interface do SEEPlus. Existem interfaces para *tickets* de vendas, leituras de variáveis atmosféricas, registos de entrada e saída de cliente (*footfall*), dados de referência como lojas, empregados, produtos, gama de produtos entre outros.

Os dados externos chegam ao sistema por meio da máquina responsável por tratar, integrar e consolida-los, designa-se de *worker machine*. Esta não é a sua única responsabilidade, outros processos de background como emitir alertas de negócio, enviar notificações, *emails* ou até mesmo executar tarefas de manutenção são executados neste órgão mecânico. Todos estes processos são executados na base de dados do sistema.

O DW, apesar de não seguir as boas práticas, é bem mais do que isso porque inclui todo um conjunto de processos de negócio que estão longe de respeitar qualquer regra ou boa prática de *data warehousing*. Pode-se encontrar no modelo de dados implementações na 3 e 5 forma normal ou até estrelas completamente desnormalizadas, muito pelo volume de dados que para conseguir velocidade nos dados, sem ter de recorrer a *big data*, tem de ter os mesmos dados em vários formatos e granularidades. Também percebeu-se que o volume seria um obstáculo à implementação de cubos, e que o preço deste tipo de tecnologias foi uma variável tida em conta.

Acima dos dados está uma série de camadas de negócio e de apresentação de dados, que estão à responsabilidade de outro órgão mecânico do sistema, a Web Role. Esta máquina disponibiliza os dados aos utilizadores sobe a forma de *web services* e *web applications*. Arquitecturalmente aqui existem alguns módulos que nitidamente as fronteiras e âmbito não foram bem definidos, porque existem ligações entre objetos que

se estes estivessem bem definidos não existiriam. Ora vejamos, na Figura 6, estão representados alguns módulos do SEEPlus. Cada módulo tem a sua própria *sub-role* de serviços e apresentação e finalmente existem uma *sub-role* comum com alguns objetos transversais. Acontece que algumas *web applications* estão a consumir serviços como se o código fosse parte da aplicação e ainda encontramos serviços duplicados entre o *iCommon* e outros módulos.

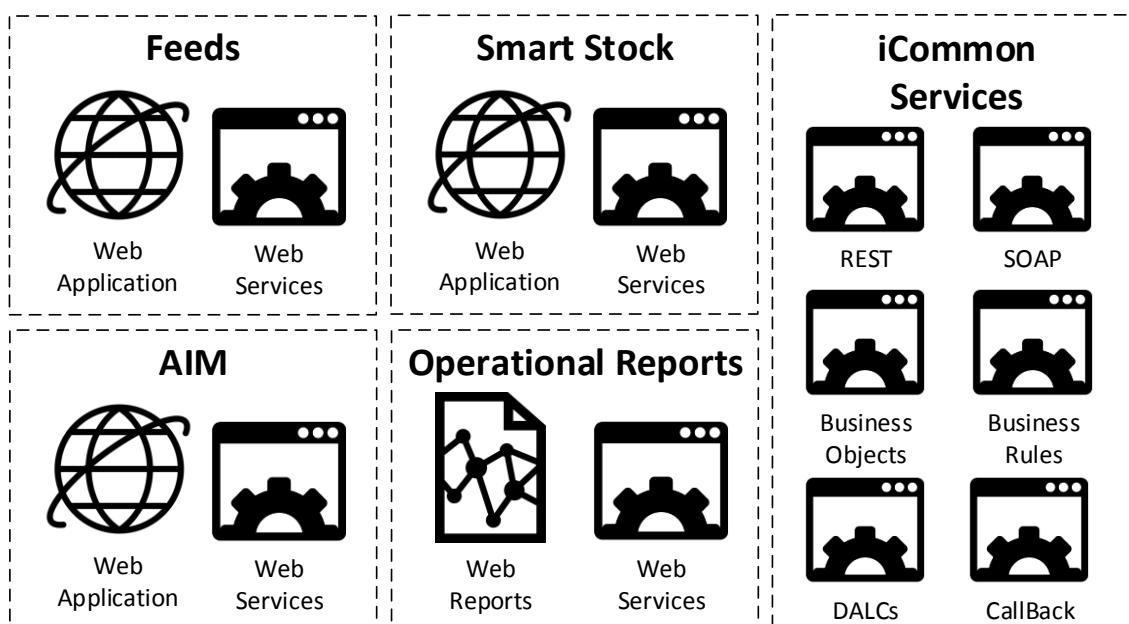


Figura 6 - Representação da Arquitetura dos Serviços

Apesar das incoerências na implementação de serviços, com algum esforço facilmente se conseguiria melhorias significativas. Outras sugestões serão dadas no capítulo para esse efeito.

A última linha é a apresentação em aplicações cliente, internas ou externas do SEEPlus. Espera-se que todas as aplicações cliente estejam a usar estes *web services* e não ligações diretas à base de dados. Não foi possível analisar a estrutura, documentação ou até mesmo código destas aplicações.

3.2.2 Módulos

A Tabela 3 apresenta, para cada módulo uma enumeração do seu conjunto de funcionalidades.

Tabela 3 - Módulos e funcionalidades do SEEPlus

Módulo	Funcionalidades
Aim	Advanced Business Dashboards; Customizable Business Reporting Platform; Real Time Monitoring; Business Alarmistic and Smart Auditing; GeoReporting (Map Navigation);
Explorer	Monitoring Atmospheric environments; KPIs statistics and benchmarking; Store performance benchmarking; Smart Finder Engine; Analytical Templates;
Command Center	Atmospheric Strategy; Store Auditing; In Store Monitor; IntelliStore;
Feeds	Create Survey Templates for multiple uses; Multiple Deploy Channels available (Web, Mobile, In Store); Location Specific Surveys; Survey Scheduler; Flexible Questions builder; Define your own Scales; Create and deploy surveys on the fly; Create your own Survey validation criteria; Integrated Analytics; Data exporting capabilities;
Spaces	Store Macro Space Designer; Store Layout Analytics (Sales, Footfall, Space); Advanced Visitor Forecasting;
Operations	Service Requests Manager; Service Template Creator; Multiple Deployment Channels; Tickets managements; Multiple platforms Ready (iPhone, Android); Low TCO and high flexibility

Em termos de imagem o SEEPlus tem muitos módulos onde alguns se relacionam e respondem a um conjunto de problemas, outros tem uma grande dependência face a outros.

É um produto baseado nas teorias mais populares de gestão e marketing, apesar de algumas funcionalidades ainda não estarem na versão final. É crucial, no curto prazo, mostrar evidências de sucesso para que o mercado tenha confiança no produto.

Contêm *dashboards* específicos e parametrizáveis pelos utilizadores. Para melhor avaliar o interface da solução e interações entre o produto e o utilizador, vai-se mostrar alguns ecrãs.

Não foram analisadas e ilustrar todas as funcionalidades, porque são usados vários paradigmas de interação com o utilizador como *emails*, *sms*, notificações no *smartphone*, monitores nas lojas, *web services* de integração e outros, mas dá para ficar com uma ideia geral dos ambientes que o utilizador tem.

3.2.3 Dashboards

A figura abaixo mostra um *dashboard* de vendas com várias perspetivas temporais como, *flash*, semanal, mensal, anual e *performance*, onde esta última compara o nível de vendas alcançado face ao objetivo definido pelo sistema. Este objetivo é definido por algoritmos que têm em conta a estratégia do cliente (aumentar ou simplesmente manter) com base em dados de anos transatos.

Pode-se observar no canto inferior direito o *widget* de *Operations* que resume o número de *issues* reportados, resolvidos e em aberto. Estes *issues* são espoletados pelos funcionários dos ambientes de lojas quando detetam alguma anomalia na infraestrutura e depois, consoante a sua natureza, são alocados ao respetivo diretor responsável, seguindo o ciclo de vida normal de um *issue*.



Figura 7 - Exemplo de um dashboard do SEEPlus

Na figura a baixo o *dashboard Atmospheric* mostra o estado ambiental das lojas. Duma forma geral ele pretende dar duas perspetivas ao utilizador, uma de auditoria (lado direito) onde é possível fazer uma análise de acontecimentos passados e uma perspetiva de *status* (lado esquerdo).

De notar no *widget ATM Startegy Execution Control* que segundo a estratégia definida pelo cliente o que está *below*, *on*, *abobe* ou *critical* e no *widget Alarmistics &*

Audit um top 5 das lojas com mais *Alarms* e bottom 2 na lista de todas as lojas do cliente.

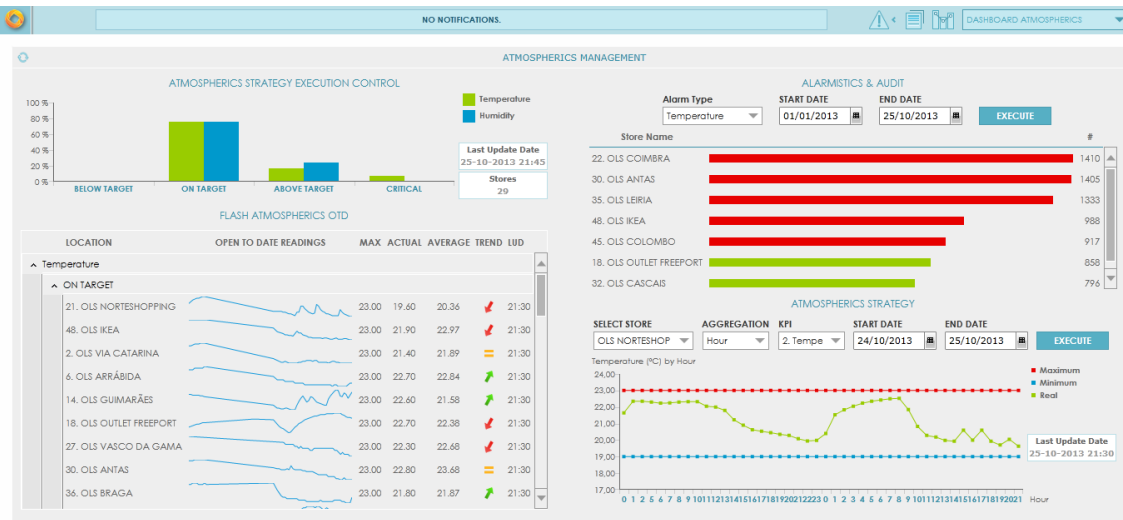


Figura 8 - Exemplo de um dashboard das variáveis Atmosféricas do SEEPlus

Por fim, a figura a baixo é um exemplo de um relatório de vendas por produto. Existem muitos outros como vendas por mês, ano, produto, categoria, objetivo ou comparativo de vários fatores.

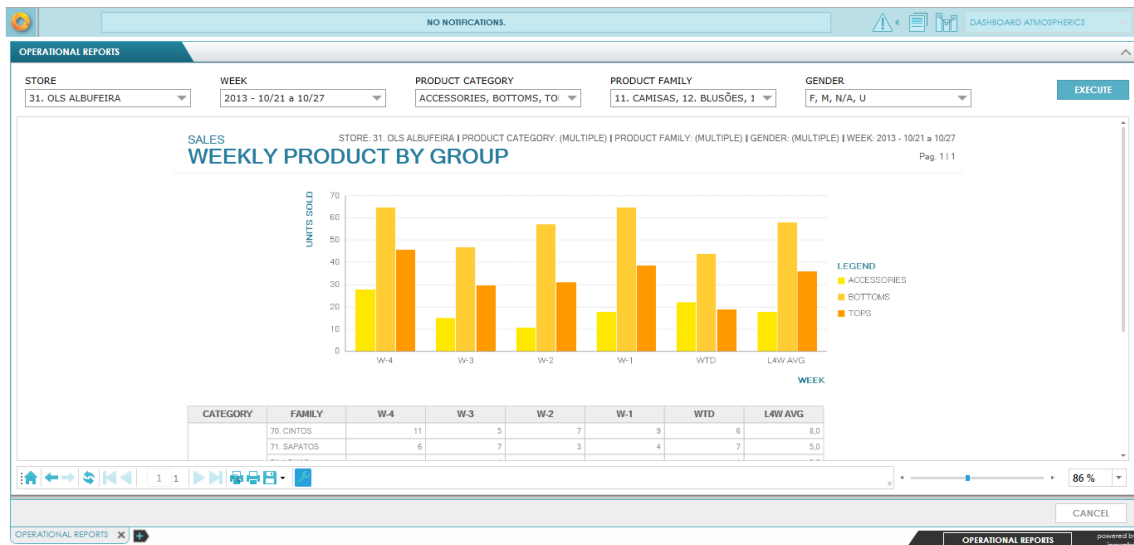


Figura 9 - Exemplo de um report web sobre as vendas

No geral é uma aplicação bastante *user friendly*, claro que esta apreciação carece de uma utilização mais intensiva do sistema, e testes de validação de usabilidade, mas pelas janelas fornecidas existe organização e divisão clara dos objetivos dos *dashboards* e *widjets*. Seria interessante analisar com mais detalhe alguns pontos.

3.3 SmartMeasure

O SmartMeasure é uma solução de previsão e análise de vendas relacionado com campanhas de marketing. Faculta uma visão de negócio e a evolução das vendas com valores mensuráveis, e objetivos claros, que até então os retalhistas não conseguiam obter.

As principais funcionalidades são:

- Previsão do valor das vendas com definição de objetivos diários;
- Alertas de fraco desempenho das vendas;
- Previsão de períodos fracos de vendas (sazonalidade);
- Previsão do impacto de campanhas de marketing;
- Previsão do melhor preço para a campanha;
- Mensuração do impacto do canibalismo consequente das promoções;

O produto ainda está em desenvolvimento pelo que algumas previsões ainda não estão completamente afinadas ao que seria desejável.

Essencialmente o SmartMeasure gera métricas e disponibiliza *web services* para alimentar outras soluções de BI, como por exemplo o SEEPlus, daí não ter ambiente gráfico para o utilizador.

O SmartMeasure é um produto que podia ser integrado no SEEPlus seguindo a estratégia de marketing da Inovretail de oferecer apenas uma grande solução para o retalho.

3.4 Customer Experience LAB

Este pretende ser um produto de primeiro contacto com o cliente, para uma análise preliminar do ambiente das lojas. Passa por disponibilizar pequenos componentes, folhas de cálculo e relatórios que são realizados por consultores em conjunto com o cliente.

É um misto de produtos e serviços muito personalizado às necessidades do cliente, ou seja é um serviço estratégico onde a organização adquire conhecimento dos seus clientes e daí nascem novas ideias e oportunidades de negócio. Muitos produtos existentes (SEEPlus, SmartMeasure) nasceram de serviços que foram produtivizados

3.5 Projetos e Serviços

São essencialmente projetos com uma linha de serviços de análise, pesquisa e desenvolvimento de ambientes de loja, vendas, disposição de espaços e montras. Por vezes um serviço que envolve muito de investigação na procura de algoritmos que melhor tirem partido dos dados dos clientes e das tecnologias. Os projetos são interessantes na abordagem de extrair *know-how* dos clientes e usar esse conhecimento noutros projetos.

O desenvolvimento não passa apenas pela oferta de produtos como SEEPlus e SmartMeasure configurados nos clientes. Existem um conjunto de projetos oportuniíssimos fora do foco principal da empresa que tentam colmatar as alturas de menor venda, típico da sazonalidade.

3.6 Data Explorer

Este novo módulo, em parte, já existe mas numa forma que não é desejada atualmente pela Inovretail. Atualmente o Monitor módulo que será descontinuado é um explorador de métricas pelas diversas hierarquias organizacionais (produtos, locais e recursos humanos). O Data Explorer, módulo aqui desenvolvido, será um *plugin* para o *Excel* de exploração de dados provenientes do DW e cubos OLAP do SEEPlus. Atualmente o Monitor é um módulo em *silverlight* que está integrado na interface web em *silverlight* que se liga diretamente ao DW.

3.7 Conclusão

A Inovretail, empresa em claro crescimento, poder-se-á dizer que está a deixar o estatuto de *startup*, começando a demonstrar solidez e confiança no mercado. Apesar de não ser referido neste capítulo, vimos na introdução que desde muito cedo tem no seu caderno de clientes, conta com marcas de renome nacional e internacional. Na página oficial podemos ver: Salsa, Levis, G-Star, Worten, Nike, Continente, Macro entre outros, que demonstra o reconhecimento de qualidade das maiores potências do retalho.

O SEEPlus, produto core da empresa, um dos responsáveis pelo sucesso de vendas, tem uma arquitetura modular em *cloud*, mais propriamente na plataforma Azure da Microsoft. Ele pretende agregar o máximo de variáveis de um ambiente, como: vendas, clientes, *footfall*, temperatura, humidade, CO₂, *stocks* e outros. Com estas medidas consegue dar em tempo real um panorama geral da organização e

individualmente de cada ponto de venda. O principal objetivo é coletar esta informação para fazer previsão de vendas, *stocks* e número de clientes, para poder no futuro fazer planeamentos de encomendas, escalonamento do horário dos funcionários ou melhorar o layout das lojas, fazendo assim diminuir os custos os seus clientes.

O Data Explorer, módulo do SEEPlus desenvolvido no corrente projeto de dissertação, será uma alternativa aos *dashboards* e relatórios estáticos do produto, dando assim ao utilizador uma dinâmica de análise superior à atualmente no SEEPlus.

A Inovretail junto com o SEEPlus, fornece serviços de Customer Experience que enriquecem a experiência do SEEPlus com projeto de consultoria focados sempre em ajudar o cliente a conhecer melhor o seu negócio e consequentemente aumentar o seu lucro, aumentando as vendas ou diminuindo custos.

4 Estado da Arte

4.1 Business Intelligence

As tecnologias, análogo à teoria da transformação das espécies segundo a de Charles Darwin, também sofreram muitas transformações devido às alterações das regras de negócio ao longo dos anos, sendo globalização talvez o fator que mais influenciou estas transformações (Vesna Luzar-Stiffler, 2008).

Nonaka e Takeuchi referem que o conhecimento é a única vantagem competitiva sustentável. A decisão futura diante das várias opções que a globalização proporciona, deve passar por um processo de análise apoiado pelo conhecimento inerente dos negócios da organização, (Ikujiro Nonaka, 2003).

Desde então, as organizações perceberam a riqueza dos dados existentes nos seus sistemas, folhas de cálculo e bases de dados dispersas, e começaram a investir em sistemas analíticos, nomeadamente em BI, (Power, 2007).

A primeira proposta oficial do termo *Business Intelligence* foi em 1989, por Howard Dresner, analista da Gartner Group, que descreveu como "conceitos e métodos para melhorar a tomada de decisões empresariais suportado em sistemas de apoio" (Power, 2007). Mais recentemente aponta-se como um conjunto de arquiteturas, ferramentas, bases de dados, aplicações e metodologias, com o objetivo de possibilitar um acesso interativo aos dados e manipulação destes, dando aos gestores e analistas a possibilidade de realizar análises adequadas, (Raisinghani, 2003). No entanto a referência vem desde as primeiras tecnologias de informação em 1958 por Luhn, citando "a capacidade de apreender as inter-relações dos factos apresentados de forma a orientar as ações em direção a um objetivo" (Luhn, 1958).

BI é um conjunto de metodologias, técnicas, ferramentas, tecnologias e atos de engenharia que influenciam o desenvolvimento de soluções para auxílio à tomada de decisão.

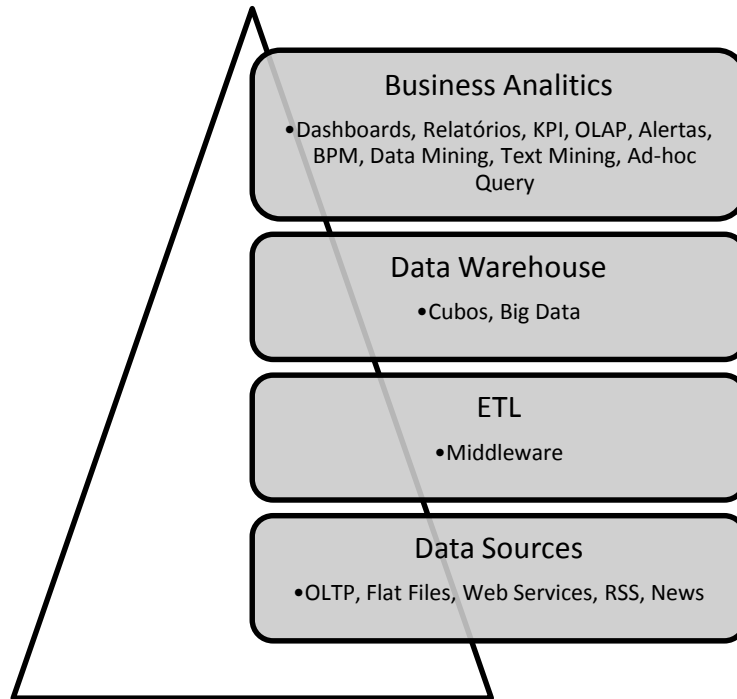


Figura 10 - Arquitetura comum num sistema de BI

Genericamente, BI é um processo evolutivo de conhecimento dos dados desde o formato original, passando pela transformação em informação, e finalmente resultando em conhecimento. Sumarizando a figura anterior, os dados representam a matéria-prima, são transformados por técnicas de consolidação (ETL) para serem armazenados em repositórios de dados (DW, cubos), prontos para os diversos interessados em descobrir conhecimento mais alargado dos dados, usando técnicas de análise e ferramentas adequadas.

4.2 Data Warehouse

A maioria das soluções de BI têm como base um DW como forma de guardar toda a informação das várias áreas de negócio da empresa servidos por vários sistemas dispersos tornando difícil a visualização unificada de todos dados.

Os precursores a empregar o termo DW, cuja definição se mantém até o hoje, foram: Inmon e Kimball (Inmon, 1991; Kimball, The Data Warehouse Lifecycle Toolkit, 1998). Inmon, estimado como o pai do DW, define-o como “uma coleção de dados orientada ao assunto, integrada, não-volátil, e variante no tempo em suporte às decisões organizacionais”. Com outra abordagem, Kimball é mais prático, simplista e

direto, afirmando que DW não é mais que uma cópia de dados de transações estruturadas para a pesquisa e análise.

As arquiteturas de DW surgiram o objetivo de diminuir os problemas de volume e diversidade de dados organizando-os numa nova estrutura. Esta estrutura de apoio à decisão aperfeiçoa os resultados já alcançados com a antiga estrutura, além de sanar diversos outros problemas.

DW várias vezes descrito como um repositório de dados orientados ao assunto, com o objetivo de suportar o processo de decisão (Inmon, 1991). De outra forma, um DW é uma base de dados que compreende as seguintes características:

- Orientado por assunto, demonstra a organização que descreve a estrutura dos dados, com o objetivo de facilitar a análise do desempenho do negócio, em detrimento da organização dada pela estrutura operacional que está orientada para o processo do negócio;
- Integrado, pretende estabelecer a concentração e a organização numa fonte única de dados;
- Histórico, reconhece a importância da análise temporal da evolução da informação;
- Não volátil, os dados quando inseridos na base de dados, não devem ser objeto de atualização de conteúdo, ao contrário das bases de dados operacionais que são sujeitas a alterações sistemáticas.

DW orientado ao assunto não é apenas uma característica basilar, mas também uma metodologia. A construção de um DW deve estar voltada em torno dos principais assuntos de negócio da organização. Os decisores procuram uma visão geral da organização de forma holística, contudo apesar de todo o esforço, é uma atividade muito complexa tratar essas informações ao mesmo tempo. Assim, para uma melhor tomada de decisão, as informações devem estar organizadas por assuntos. Desta forma no desenvolvimento de um DW deve-se ter a preocupação de identificar desses assuntos como forma de facilitar o uso futuro do sistema.

A Figura 11 representa a arquitetura de um sistema de BI, em que se pode visualizar os elementos principais que fazem parte deste, nomeadamente:

- Fontes de dados: Repositórios do qual informação será carregada para o DW;

- ETL: Ferramentas de extração, transformação e carregamento de dados para o DW.
- DW: Repositório onde é armazenada a informação tratada sem erros ou inconsistências.
- Ferramentas de análise: Ferramentas cujo objetivo é possibilitar a exploração, por parte do utilizador ou outros sistemas inteligentes, da informação no DW.

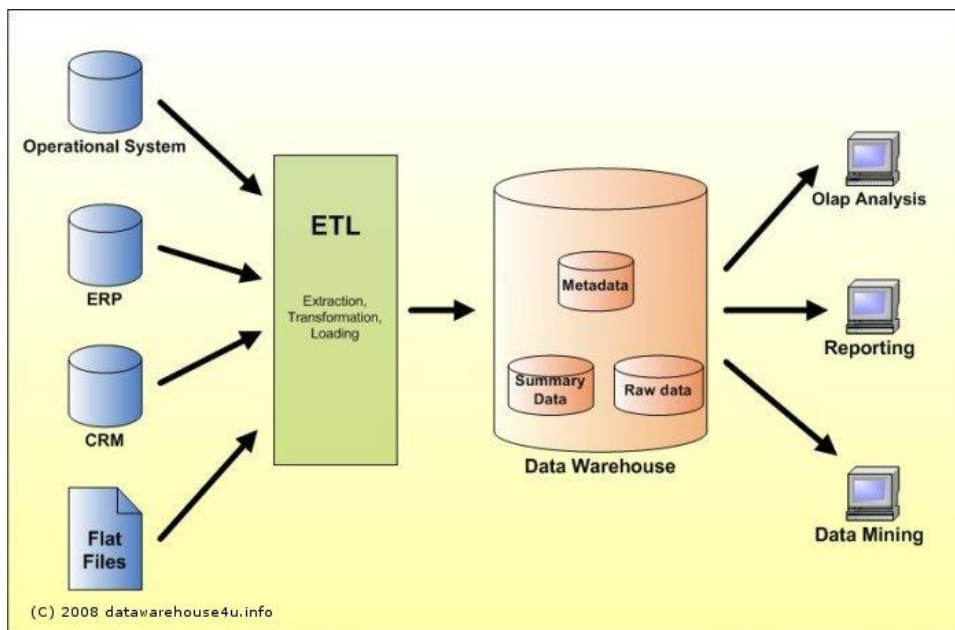


Figura 11 - Arquitetura de um sistema de BI

4.2.1 Metodologias

Existem duas grandes metodologias de implementação de DW, porém antes será necessário definir o que é um DM. Segundo Moody e Kortink, DM são “postos de venda” de um DW, que disponibilizam informação aos analistas de dados apenas de uma área de negócio da organização, enquanto o DW disponibiliza uma visão unificada da informação de toda a organização (Moody & Kortink, 2000). Isto traz a vantagem de segmentar os dados de forma distribuída e consegue direcionar apenas a informação relevante para um determinado tipo de utilizadores.

Posto isto as duas metodologias que se seguem, Inmon e Kimball, defendem de forma diferente a abordagem DW e DM.

4.2.1.1 Inmon

A metodologia de Inmon, também conhecida como *Top Down Approach*, defende mais a raízes de DW, sendo que deve existir sempre um DW organizacional com toda a informação dos processos de negócio da organização (Breslin, 2004).

Inmon defende que um DW como um ponto único de acesso aos dados, com toda a informação de todos os processos de negócio, da mesma forma todas as ferramentas de visualização e análise devem aceder apenas a este ponto único de armazenamento. Isto significa que o DW organizacional é a única fonte de informação fidedigna que os utilizadores devem usar, (Inmon, 1991). É por isso uma metodologia que obriga a pensar no problema de implementação de um sistema de BI de forma holística, como referido por Mary Breslin, “entre outras coisas, um DW e as bases de dados operacionais fazem parte de um todo maior” (Breslin, 2004).

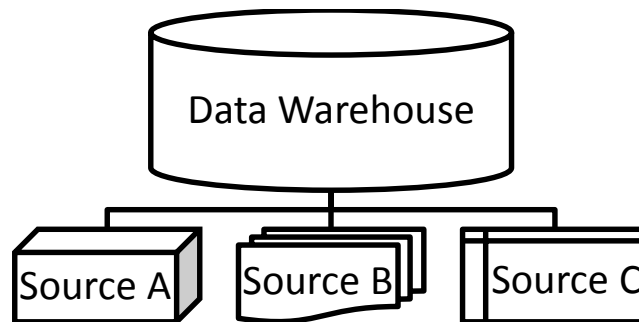


Figura 12 - Data warehouse organizacional

4.2.1.2 Kimball

Kimball é menos radical na abordagem ao DW. A sua metodologia também é conhecida por *Bottom Up Approach*, porque a implementação deve começar com DM de cada processo de negócio que depois esses alimentam um DW organizacional (Kimball & Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 2002).

Este processo apresenta menos riscos pois o desenvolvimento de um DM revela-se muito menos complexo individualmente em cada DM do que um DW. No entanto não havendo uma visão global pode ocorrer em incoerências entre DM. Cada DM terá as suas dimensões e medidas, contudo muitas delas serão comuns entre cada DM, mas fisicamente estarão separadas. A metodologia de Kimball vem ajudar na identificação de objetos comuns entre DM.

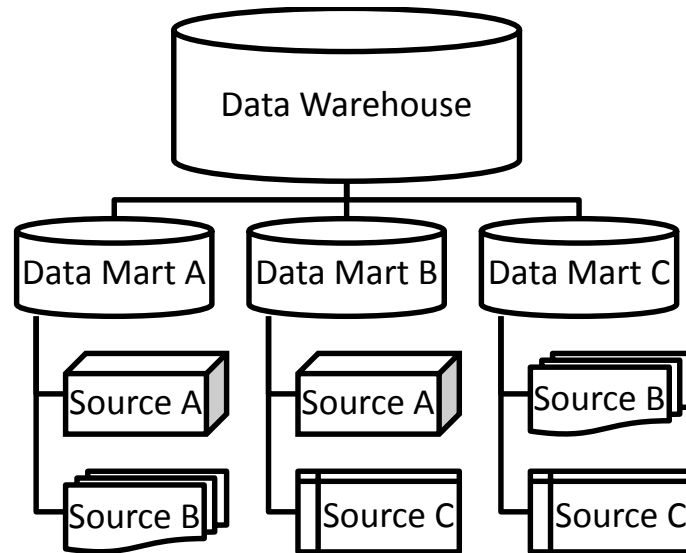


Figura 13 – DW organizacional alimentado por DM segmentados

4.3 Tecnologias OLAP

As ferramentas OLAP, devido às suas propriedades, são ferramentas acessíveis para realizar consultas e análises sobre os dados armazenados em sistemas de DW, dado que são ferramentas dimensionadas para tirar partido de armazenamento de dados modelado numa forma multidimensional, (Oracle, Oracle OLAP, 2014).

O facto de estes sistemas estarem organizados em tabelas de facto, com métricas bem definidas e dimensões estruturadas segundo uma hierarquia, estão presentes propriedades simples e bastante flexíveis que dificilmente se encontram em outros sistemas de dados como OLTP.

Confrontando com os sistemas de *reporting*, em que os dados não estão organizados por assunto, dificilmente se consegue gerar um relatório sem ter conhecimento profundo da tecnologia e do modelo de dados transacional. As ferramentas apenas com estes elementos (factos, métricas, dimensões e hierarquias), juntamente com algumas operações de pesquisa, consegue ser muito mais flexível. Na atualidade a Microsoft já permite no Office realizar questões numa linguagem quase natural para obter *reports*, isto não seria conseguido sem a organização dos sistemas OLAP, (Microsoft, Introduction to Microsoft Power Query for Excel, 2014).

A versatilidade destas consultas permitem a um decisor possuir uma visão mais flexível, e por vezes mais ampla sobre o negócio que se encontra a gerir.

Nos sistemas OLAP pode-se encontrar várias arquiteturas. Entre elas as mais implementadas são ROLAP, MOLAP e HOLAP. Os critérios necessários para a escolha da arquitetura, bem como as características de cada um, ficará de fora deste tema.

4.3.1 Cubos OLAP

Não será de todo incorreto dizer-se que a modelação dimensional é uma técnica composta pela elaboração de modelos lógicos (chamados de esquemas dimensionais), a serem armazenados fisicamente numa base de dados, com o principal objetivo de aumentar a *performance* no acesso aos dados (Kimball, The Data Warehouse Lifecycle Toolkit, 1998).

Todos os modelos multidimensionais são compostos por tabelas de factos, que são nada mais do que tabelas que possuem uma chave primária, composta pelas chaves estrangeiras das dimensões a que se relaciona, e as medidas que cada entrada regista. Estas medidas de cada entrada serão obrigatoriamente valores numéricos, pois de outra forma não será possível realizar operações matemáticas. Ao conjunto de dimensões e medidas numa entrada pode-se chamar de facto.

Na Figura 14 está representado um esquema multidimensional, onde se define uma tabela de factos com 3 dimensões (tempo, produto e cliente) e uma medida (lucro).

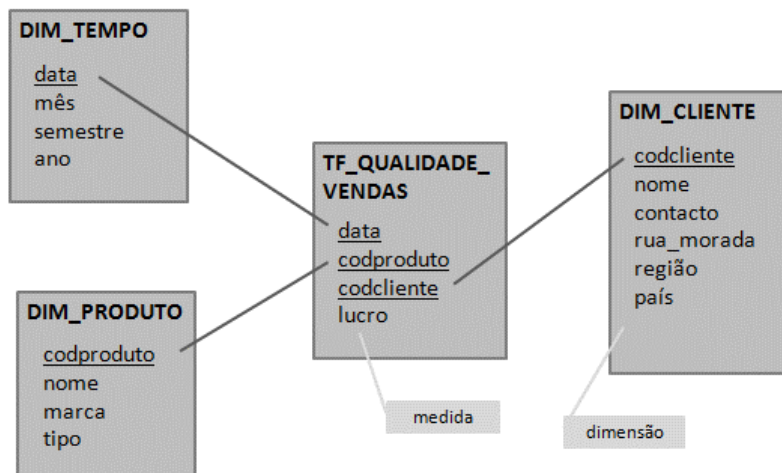


Figura 14 - Exemplo de um esquema multidimensional

4.3.2 Operações de Consulta

Uma *query* a um sistema OLAP constitui na exploração de um cubo, em uma ou mais dimensões, com a aplicação de sucessivas operações que produzem o processo de análise dos factos armazenados no cubo. O resultado será uma estrutura multidimensional, dependendo do número de dimensões, mas que visualmente será mostrado como uma tabela com vários cabeçalhos, cores e indicadores, realçando os resultados (Kimball, The Data Warehouse Lifecycle Toolkit, 1998). Se não tomasse o

aspecto de uma tabela seria quase impossível um humano conseguir interpretar os dados, (Golfarelli & Rizzi, 2009).

Cada operação efetuada no cubo altera a composição original do mesmo, sendo os mais comuns: *roll-up*, *drill-down*, *slice and dice*, *pivot*, *drill-across* e *drill-through* (Golfarelli & Rizzi, 2009):

- **Roll up** - Reduz o nível de detalhe que os dados estão sujeitos. Este nível de detalhe é definido pela hierarquia da dimensão. Por exemplo numa dimensão Tempo, uma hierarquia possível será (Ano – Semestre – Trimestre – Mês – Dia). Desta forma será possível visualizar os dados ao mês e fazer *roll-up* para o Ano e analisar os dados anuais.
- **Drill-down** - Contrariamente ao *roll-up* aumenta o nível de detalhe que os dados são sujeitos. No exemplo dado no *roll-up* quando os dados estão agregado ao nível hierárquico do ano, é possível fazer um *drill-down* para o mês ou dia.
- **Pivot** - Transforma a apresentação dos dados com uma perspetiva, trocando linhas por colunas, ou seja, as dimensões que definem as colunas passam para as linhas e vice-versa.
- **Slice-and-dice** - Como o nome sugere, aplica corte no cubo mostrando apenas parte do mesmo. Na linguagem SQL mais comum nos DW, é equivalente a uma condição na cláusula *where*.
- **Drill-through** - Menos comum, permite efetuar a ligação entre os dados armazenados no cubo presentes no servidor OLAP e os correspondentes dados armazenados nos sistemas de origem, ou seja no DW.
- **Drill-across** - Estabelece a comparação entre cubos relacionados, comparando os seus dados, estabelecendo a correspondência entre os factos e as dimensões a serem representados.

4.3.3 Arquitetura OLAP

4.3.3.1 ROLAP

Os sistema em arquitetura ROLAP (*Relational OLAP*) o servidor faz de *middleware* entre o sistema de base de dados, tipicamente um DW, e o sistema de apresentação de dados. Desta forma os servidores ROLAP exploram as potencialidades de um motor de base de dados relacionais de fácil ligação, integração e seleção de conteúdos. Eles exploram as consultas multidimensionais normalmente otimizadas para este tipo de sistemas relacionais, (Connolly & Begg, 2001).

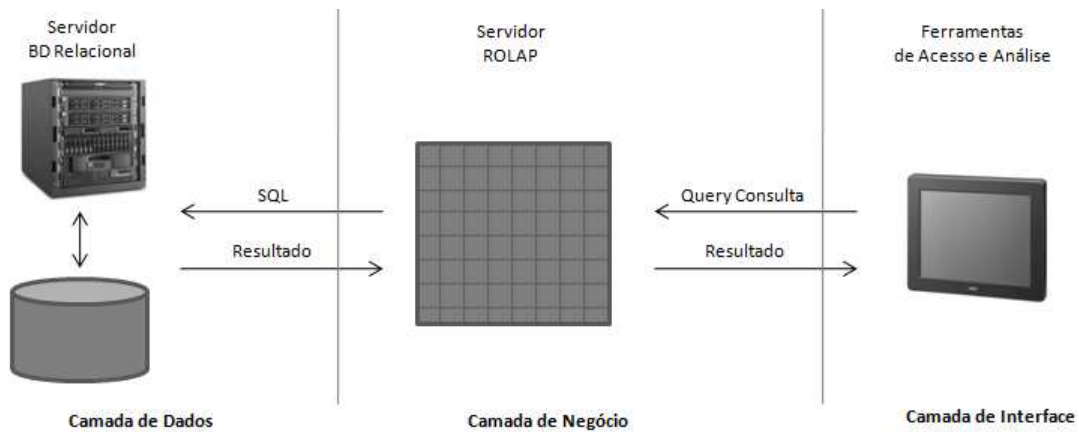


Figura 15 - Componentes gerais de um servidor ROLAP

Esta abordagem tem vantagens e desvantagens. Vantagem porque não necessita de um repositório para as matrizes multidimensionais dos cubos OLAP, e assim menos dispendioso. No entanto tem a desvantagem de necessitar de converter constantemente os dados relacionais em estruturas de dados multidimensionais analíticas dos sistemas OLAP, o que piora o desempenho deste sistema em relação a outras arquiteturas OLAP, (Connolly & Begg, 2001).

4.3.3.2 MOLAP

O MOLAP (Multidimensional OLAP) caracteriza-se essencialmente por não os dados não tem um formato relacional mas sim multidimensional. Isso torna as pesquisas muito mais rápidas por o mapeamento ser direto entre a *queries* do cliente e a estrutura armazenada nestes sistemas. Isto é auxiliado pelas propriedades de indexação própria que estes sistemas têm e os tornam tão rápidos (Connolly & Begg, 2001).

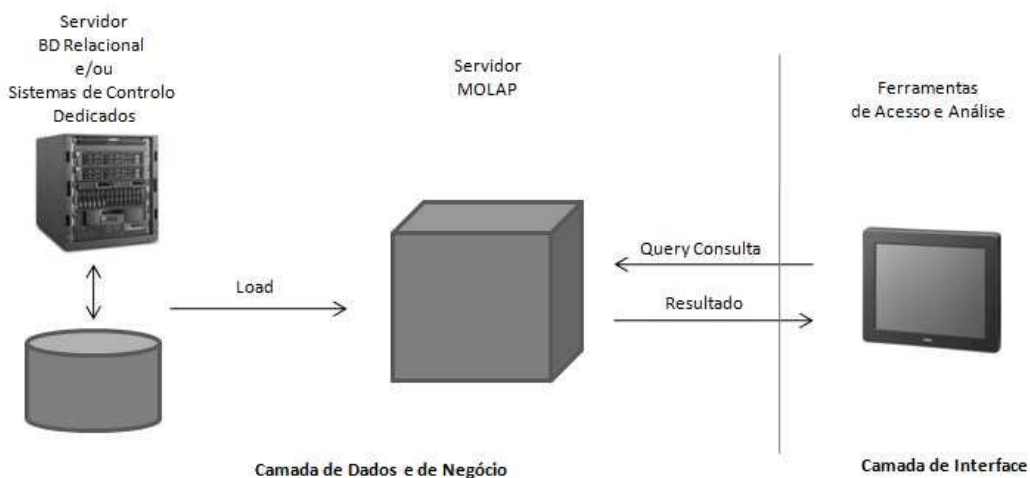


Figura 16 - Componentes gerais de um servidor MOLAP.

Uma desvantagem é que a manutenção requer mais ações para que o servidor OLAP esteja sempre atualizado com os últimos dados do DW. Isto também requer outro tipo de conhecimento de análise e ferramentas por parte dos administradores de base de dados. Outra desvantagem prende-se com a escalabilidade muito inferior ou quase nula em relação a outros sistemas de armazenamento. O carregamento necessário entre os sistemas operacionais e o servidor MOLAP pode ser um entrave quando são necessárias pesquisas em tempo real.

4.3.3.3 HOLAP

O problema da pesquisa a dados em tempo real pode ser resolvido com um sistema híbrido de OLAP (HOLAP), combinando ROLAP e MOLAP. Uma arquitetura deste tipo pode conter dados em tempo real do passado recente, normalmente menos volume, e dados de histórico, grandes volumes que não conseguem escalar num MOLAP, com os dados do passado (médio prazo, 2-3 anos dependendo do problema).

O processo de seleção de vistas a materializar para um MOLAP, a indexação dos conteúdos armazenados e os processos de integração sobre os cubos são similares aos utilizados nos sistemas ROLAP e MOLAP.

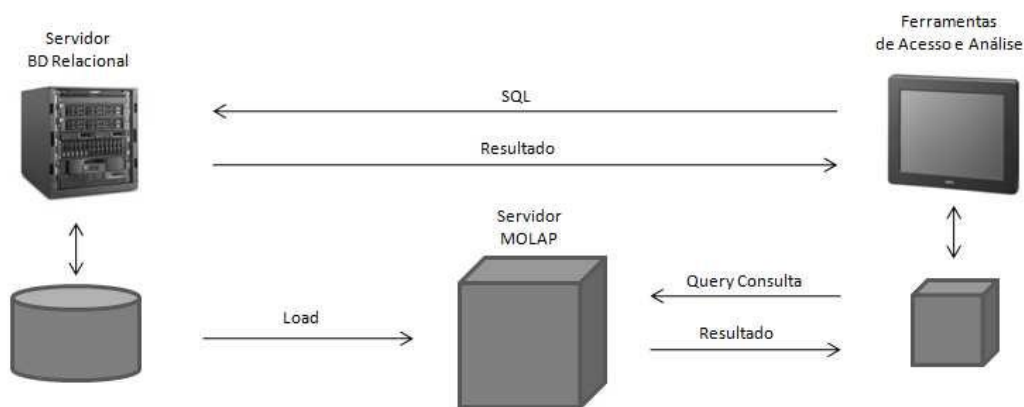


Figura 17 - Componentes gerais de um servidor HOLAP.

A principal desvantagem é a redundância pois podem existir varias fontes para a mesma pesquisa e pode comprometer a integridade dos dados quando o sistema é usado por vários utilizadores ao mesmo tempo. Ainda apenas um pequeno grupo de dados estará eficientemente controlado.

4.4 Cloud

A filosofia da SEEPlus e da Inovretail é *cloud-oriented*. Desta forma sempre que possível qualquer solução da Inovretail deve ser implementada na *cloud*. Como também a Inovretail é *microsoft-oriented* os serviços de *cloud* serão sempre a plataforma Azure. Assim serão analisadas algumas características do Azure.

Implementar um projeto na *cloud*, trás grandes vantagens, principalmente para o início de um projeto:

- Baixo investimento inicial;
- Baixa manutenção;
- Completamente escalável;
- Menos preocupações de manutenção;
- Protocolos de segurança já implementadas.

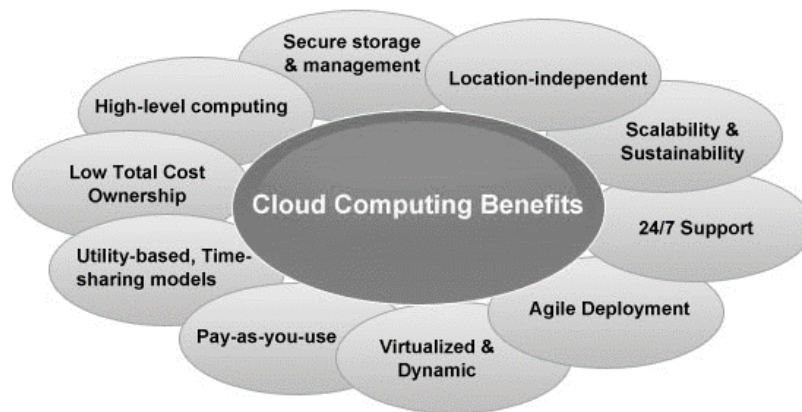


Figura 18 - Cloud Computing Benefits (Win-Pro Consultancy, 2012)

Não é relevante analisar outras plataformas ou serviços de *cloud* porque a implementação em Azure é uma restrição deste projeto. Existem muitos outros fornecedores, fazendo referência aos maiores como Amazon e Google.

4.4.1 Microsoft Azure

“Em poucas palavras, é uma plataforma na nuvem da Microsoft: uma coleção em expansão de serviços integrados — computação, armazenamento, dados, redes e aplicações — que o ajudam a mover-se rapidamente, a fazer mais e a poupar dinheiro” (Microsoft, O que é o Microsoft Azure?, 2015).

O Azure fornece infraestrutura como serviço (IaaS) e plataforma como serviço (PaaS). Esta poderosa combinação de serviços geridos e não geridos permite criar, implementar e gerir aplicações como desejar com mais produtividade.

Suporta qualquer sistema operativo, linguagem, ferramenta e estrutura, desde Windows a Linux, SQL Server a Oracle, C# a Java. Algumas destas tecnologias estão disponíveis apenas no modelo IaaS, já as tecnologias Microsoft quase todas podem ser usadas como PaaS.

Oferece um SLA (*service level agreement*) de disponibilidade de 99,95%, suporte técnico 24x7 e monitorização do estado de funcionamento do serviço permanente.

Os recursos disponíveis num ambiente podem ser aumentado ou diminuído facilmente para corresponder às exigências, seja por custos ou carga de processamento. A Microsoft usa um sistema de faturação que cobra apenas aquilo que está a ser utilizado no momento, podendo ser por memória usada, operações por minuto do processador, largura de banda utilizada.

Para além disso garantem redundância nos seus *backups* no mínimo em 2 pontos do globo, mas a rede conta com 19 regiões com *data centers*. O Azure é o primeiro fornecedor de serviços em nuvem multinacional na China continental e está a prosseguir a expansão para novas regiões do mundo (Microsoft, O que é o Microsoft Azure?, 2015).

Entre todos serviços do Azure os chave para este projeto são:

- Windows Azure
- SQL Azure
- Azure .NET Cloud Services

4.4.1.1 Windows Azure

O Windows Azure é um serviço de *virtual machine* que se consegue ligar de forma segura através de VPN ou com regras de acesso em rede. O balanceamento de carga destas máquinas pode ser gerido automaticamente pelo Azure ou definir regras próprias. É possível montar VMs para testes ou como máquinas de produção, apenas com alguns cliques, já com ferramentas como SQL Server, IIS, Dynamics CRM, desde que estas estejam na lista das licenças disponíveis da subscrição ativa no Azure, e no final dos testes apagar a mesma VM. Isto dá uma flexibilidade impensável com infraestruturas próprias e com custos bastante reduzidos.

4.4.1.2 SQL Azure

O SQL Azure é uma versão *cloud* do SQL Server 2008, que tem sido migrado aos poucos para o SQL Server 2012, apesar de a Microsoft não fazer referência no site oficial. O serviço é disponibilizado num modelo de DaaS (Microsoft, Azure SQL

Database, 2015). É um pouco mais limitado do que o SQL Server por razões óbvias da *cloud*, é uma das desvantagens da *cloud*. No entanto a criação de uma nova base de dados, os recursos disponíveis, o agendamento de *backups* é feito no Portal do Azure com poucos cliques (Microsoft, Management Portal for SQL Database, 2015).

4.4.1.3 Azure .NET Cloud Services

Este serviço permite criar ambiente para criar aplicações web e API. As principais características são:

- Plataforma segura para aplicações confiáveis;
- Dados seguros e hospedagem com suporte 24/7;
- O acesso à gama de ferramentas e tecnologias da Microsoft;
- Soluções dinâmicas, elásticas e escaláveis;
- Total controlo de custos.

Este tipo de soluções está restrito a tecnologias Microsoft pelos serviços correrem no servidor web da Microsoft o IIS, e por isso o desenvolvimento também fica restrito ao uso do IDE Visual Studio. As soluções depois de testadas num IIS e Visual Studio local, podem ser encapsuladas num pacote de *software* e fazer-se o *upload* para a *cloud* (Microsoft, Cloud Services, 2014).

Compute oferece três tipos de funções diferentes que são projetados para atender às necessidades específicas de determinados tipos de aplicações. Todos esses papéis executado como VMs hospedados em *data centers* Azure. (Tejada, 2012).

- *Web Role* – Com o objetivo de executar aplicações do servidor web, fornece um suporte *out-of-the-box* Microsoft IIS 7 e ASP.NET. A *Web Role* também pode ser usada para disponibilizar serviços web e APIs como SOAP (*Simple Object Access Protocol*), REST (*Representational State Transfer*) ou WCF (*Windows Communication Foundation*).
- *Worker Role* – Para o desenvolvimento mais comum nas aplicações é a máquina executa processos de *batch*, podendo processar imagens carregadas na Web Role, processar vídeo ou outro tipo de dados.
- *VM Role* - Permite fazer o *upload* de uma imagem de uma VM que contém os componentes que a implementar numa infraestrutura Azure.

4.5 Padrões de Arquitetura para GUI

Os padrões de desenvolvimento são muitas vezes a solução recorrente para problemas de conceção de GUI e outros de mais baixo nível. Isto pode trazer menos ou mais atenção no projeto dependendo do contexto e da aplicabilidade destes padrões no desenvolvimento de aplicações. No que interessa analisar, estes padrões devem dar linhas orientadoras da hierarquia de objetos de interface e dados para minimizar a dependência entre eles, recorrendo a protocolos de implementação, que consequentemente pode aumentar o número de ligações entre eles. A principal vantagem de usar padrões de conceção é a forma como potenciam a reutilização de código (Granlund, Lafrenière, & Carr, 2001).

4.5.1 MVVM (Model-View-ViewModel)

MVVM é um *pattern* de desenvolvimento de *software*, introduzido por Martin Fowler (Fowler, Presentation Model, 2004) e usado principalmente em tecnologias Microsoft desde 2006/2007 com o aparecimento do WPF e Silverlight.

Este *pattern* facilita a separação clara entre a interface do utilizador (*View*), controlo (*ViewModel*) e camada de negócio e os dados (*Model*). Estes componentes ligam-se por uma forte ligação de eventos e chamadas assíncronas o que os torna mais dependentes mas com menos regras de ligação. O *ViewModel* recebe e despoleta eventos da *View* e do *Model* quando alguma coisa é alterada numa das partes (Wikipédia, 2014).

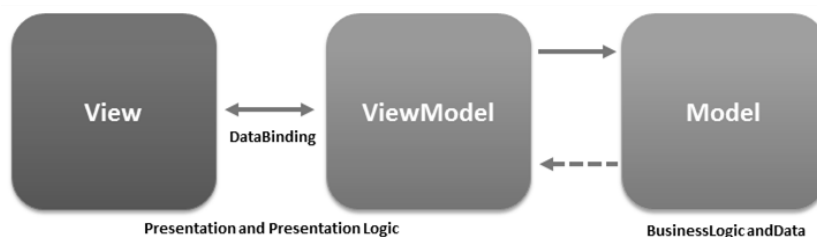


Figura 19 - MVVM Pattern

Pode-se ter no limite mais do que uma *View* ligada ao mesmo *ViewModel*, e quando um valor é alterado todas as *Views* vão ser notificadas da alteração e todas elas vão mostrar o novo valor. De outra forma é possível ter uma *View* com dois *ViewModel* em que cada um terá uma responsabilidade bem definida no controlo dos dados (Wikipédia, 2014).

Resumidamente cada componente tem a obrigação:

- *Model*: Aceder e controlar dados no seu estado bruto, com todas as regras de negócio associadas, muitas vezes implementado por uma camada de serviços de negócio;
- *View*: São todos os componentes de interface apresentados ao utilizador como botões, listas ou caixas de texto, não tem qualquer tipo de lógica a não ser a lógica de apresentação.
- *ViewModel*: É o controlador dos dados que converte-os do *Model* para a *View* e vice-versa. Ele contém, para além dos dados, todos os comandos e propriedades da *View*, assim ele é responsável pelo comportamento da *View*, mas a responsabilidade de apresentação é da *View*. Este é o elemento chave, e mais complexo, do *pattern*.

Um dos maiores responsáveis pelo desenvolvimento deste *pattern* aponta algumas desvantagens. Para projetos menores este pode ser um parão muito declarativo e burocrático, já para projetos maiores o *ViewModel* pode ficar muito complexo. Criar e ler o código e documentação pode ser mais difícil com todos os eventos que têm de ser representados. Por vezes o número de ligações entre objetos são mais do que os próprios objetos o que para o cérebro humano alguns diagramas podem ficar confusos (Gossman, 2006).

4.5.2 MVP (Model-View-Presenter)

Este *pattern* é derivado de outros como o MVC, mas mais orientado para aplicações Desktop como Windows Forms do .net da Microsoft ou Swing do Java, sendo que o MVC normalmente está mais associado a aplicações Web (Fowler, MSDN Dot Net Blogs, 2008).

Ele é composto por 3 tipo de objetos com responsabilidades distintas que se completam:

- *Model*: É uma única classe que contém todas as propriedades de uma entidade de dados. Ele guarda a informação mas não de forma persistente. As suas propriedades por sua vez podem ser objetos complexos com múltiplas propriedades ou até mesmo listas.
- *View*: É o objeto que transforma consegue transformar todos os objetos em formas visuais, mas é um componente que pouco faz. Contém as propriedades que refletem os dados do objeto *Model*. Dispara os eventos da GUI que são enviados *triggers* para o *Presenter*.

- *Presenter*: Faz a ligação entre os dados e a GUI. Recebe *inputs* da *View* e reflete as mesmas ações em alterações de dados no objeto *Model*. No limite ainda pode receber *triggers* de alterações de dados na base de dados e refletir essas alterações na *View*. É o objeto que realiza mais trabalho e contem as regras de negócio da aplicação.

Este é um *pattern* muito simples na parte de implementação dando a liberdade de usar outros padrões de conceção menos abstratos como *creation*, *structural* e *behavior patterns* dos Gof (Gang of Four) (Gamma, Helm, Johnson, & Vlissides, 1994).

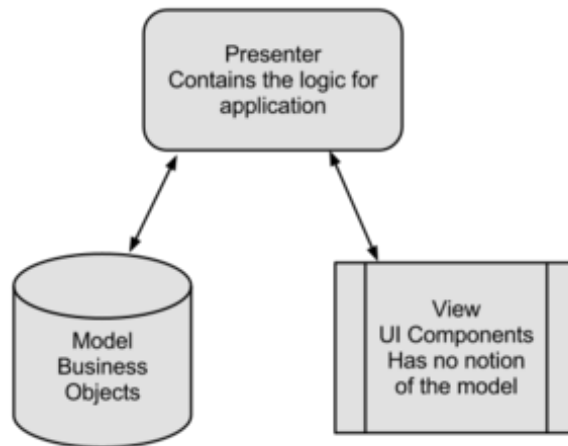


Figura 20 - Objetos e Relações no *pattern* MVP

4.6 Trabalho Relacionado

Não existe um número oficial de soluções de BI. Contudo de todas as soluções existentes, o que se procura são soluções que se liguem ao *Excel* por meio de um *plugin* que exige um registo ou autenticação para aceder a um repositório OLAP.

Um concorrente direto é o PowePivot e Power BI da Microsoft, mas que tem propósitos e requisitos distintos. No entanto é possível usar estes dois *plugins* para criar tabelas dinâmicas e gráficos que ficam ligados às fontes de dados e que a qualquer momento o utilizador pode refrescar os dados. Harinath, Pihlgren, Guang-Yeu Lee [2010] referem que com *PowerPoint* é possível criar aplicações *end-to-end* com captura dos dados, ligação entre eles, e visualização, sendo ainda possível publicar uma folha de *Excel* na *web* mais numa perspetiva de *self-service* BI. Tal como os anteriores autores, Barry Ralston [2011] também refere as potencialidades da linguagem DAX ou SQL, para criar *queries* que conseguem obter tabelas complexas que uma interface não conseguiria apenas com ações de *drag and drop*. Por fim referem que é possível ligar

essas folhas de cálculo com o SharePoint da Microsoft, mas que não se enquadra na investigação que decorre (Harinath, Pihlgren, & Guang-Yeu Lee, 2010; Ralston, 2011).

Um segundo concorrente é a Oracle com o produto Hyperion Essbase. Varsha Bishnoi nos 2 artigos publicados (Bishnoi, An introduction to Hyperion EssBase, 2012; Bishnoi, Implementation of EssBase Application using Oracle Hyperion, 2012) refere que o Hyperion Essbase é mais do que um cliente de análise de dados. “É um software de base de dados multidimensional otimizado para o planeamento, análise e *reporting* de gestão. Combina de forma única um *design* técnico inovador, com uma arquitetura cliente-servidor aberto”. A conexão é feita por meio de um *plugin* do *Excel* que permite ligar ao servidor OLAP multidimensional, e aplicar os operadores comuns. Mais alguns detalhes são mostrados no manual oficial do produto disponibilizado pela Oracle (Oracle, SpreadSheet Add-in User's Guide for Excel, 2007).

De facto não se poderá dizer que os dois produtos semelhantes substituem totalmente o Data Explorer. Por um lado no PowerPivot falta a componente de autenticação ao servidor do SEEPlus que impõe aos utilizadores uma *role* que limita o número de factos que este pode ver, e o produto da Oracle, mais semelhante, é uma solução completa que faz mais concorrência ao SEEPlus do que propriamente ao Data Explorer.

4.7 Conclusão

Verificou-se que o conceito de BI bem a ser investigado há mais anos do que se pensava, apesar disso só mais recentemente as organizações começam a estar mais atentas a este tipo de tecnologia. Os DW e tecnologias OLAP são as que potenciam as análises mais ricas, com tempos de resposta a pedidos mais eficientes, no entanto estas tecnologias trazem alguns custos devido à redundância necessária para alcançar a devida eficiência. Seria bom conseguir aprofundar mais o conceito de DW com análise de arquiteturas e casos de estudo com o objetivo de no futuro ter conhecimento das boas práticas aplicadas na indústria, para além das duas metodologias de Inmon e Kimball, e assim propor melhorias à arquitetura do SEEPlus. No que toca à tecnologia OLAP procurar tecnologias OLAP na *cloud*, isto porque a Inovretail pretende usar o SSAS da Microsoft mas seria positivo propor alternativas.

A *cloud* Azure da Microsoft, também é uma tecnologia usada pela Inovretail, será o ambiente de desenvolvimento, testes e produção da solução do Data Explorer.

Na tentativa de encontrar trabalhos semelhantes, existem desenvolvimentos em *Excel* com PowerPivot e PowerBI mas não é com esse tipo de soluções que se vai conseguir resolver o problema inicialmente identificado. O produto Essbase da Oracle, é mais um concorrente do SEEPlus do que o Data Explorer, que pretende ser uma solução parametrizável para qualquer tipo de negócio. O SEEPlus e o Data Explorer ambicionam ser um produto chave na mão para os retalhistas com menos parametrizações, devido à sua especificidade do mercado em que insere.

Com os conceitos analisados, que serão usados para implementar uma ferramenta de análise de dados no *Excel*, pretende-se dar ao utilizador toda a liberdade de criar os relatórios e *dashboards* que desejar, sem ficar preso aos *dashboards* estáticos disponíveis no SEEPlus. O uso de um *plugin* no *Excel* justifica-se pela forma que consegue bloquear os utilizadores de aceder à fonte de dados, exigindo a autenticação para identificação das permissões que lhe são devidas. Para além disso outras funcionalidades podem ser implementadas, tais como atalhos para KPIs ou visualização e partilha de *templates* de outros utilizadores.

5 Data Explorer

Neste capítulo serão aprofundadas todas as matérias relativas ao desenvolvimento do Data Explorer, como requisitos, detalhes das funcionalidades, a arquitetura os artefactos de *software* desenvolvidos.

5.1 Requisitos Funcionais e Não Funcionais

Todo o desenvolvimento de engenharia de *software* para que tenha sucesso no cumprimento dos seus objetivos, necessitam que estejam definidos à cabeça para que possam ser medidos os desvios durante e no final do projeto e assim determinar o sucesso ou insucesso do projeto. A Inovretail também tem o seu processo de negócio para projetos de desenvolvimento, como tal, no início do projeto foram definidos os requisitos funcionais num FRD (*Functional Requirement Definition*). Este documento por questões de confidencialidade não estará disponível em anexo e as partes resumidas mais abaixo não poderão ser detalhadas. Contudo será possível disponibilizar os tópicos resumidos deste FRD.

Segue-se por tópicos as funcionalidades estruturais do projeto.

5.1.1 Login

O Data Explorer tem de permitir aos utilizadores fazerem *login* no SEEPlus com as mesmas credenciais que já usam em outros módulos do SEEPlus.

Para além disso deve permitir ao utilizador aceder a vários ambientes com perspetiva de cumprir o requisito de alguns clientes da Inovretail que querem ter acesso a um ambiente de testes para avaliar versões *beta*.

A não autenticação no Data Explorer deve bloquear todas as restantes funcionalidades dos requisitos seguintes.

5.1.2 Repositório de Relatórios

Os relatórios gerados segundo os requisitos seguintes devem ser guardados num repositório do SEEPlus com o objetivo do cliente poder mudar de computador e abrir o mesmo relatório. Isto é essencial para a partilha entre utilizadores.

Se o utilizador pretender guardar um relatório no seu computador o Data Explorer não tem obrigatoriedade de controlar funcionalmente o mesmo.

5.1.3 Exploração de Dados em Cubos

O Data Explorer deve facilitar o acesso aos cubos do SEEPlus e dar toda a liberdade de gerar relatórios dinâmicos com personalizações ao gosto do cliente. Estes relatórios tem de ser atualizáveis com os dados mais recentes do cubo.

Da lista de todos os cubos disponíveis no SEEPlus, um cliente não deve ter acesso a todos eles, a menos que tenha sido partilhado por outro utilizador.

Como acontece no SEEPlus que os clientes estão limitados ao número de factos e registos por tabela de factos, o mesmo não tem de acontecer nos cubos, ou seja, um utilizador quando tem acesso a um Cubo pode visualizar os dados do mesmo.

5.1.4 Visualização de Relatórios

Para os relatórios guardados em sistema o utilizador deve conseguir ver a qualquer momento mesmo quando tem um Excel novo vazio. A lista de relatórios deve estar disponível num *menu* do Excel.

Para restrição deste projeto um relatório ficará restrito a uma única folha de dados do Excel, pelo que quando um relatório é gerado ele deve ser independente de outras folhas de dados, ou então as ligações pode ser perdidas.

5.1.5 Template

Tem de ser possível ao utilizador criar *templates* para relatórios com o objetivos de se conseguir reutilizar o mesmo. O *template* tem de ser composto por uma secção de cabeçalho, outra para o corpo do relatório e finalmente uma secção para o rodapé.

Tal como os relatórios os *templates* também podem ser partilhados com outros utilizadores, sendo que um utilizador com acesso ao mesmo pode alterare-lo quando bem entender.

5.1.6 Gestão de Relatórios

Qualquer relatório ou *template* poderá ser editado, eliminado ou partilhado, desde que o relatório esteja a ser gerido no Data Explorer e não a partir do disco do utilizador.

5.1.7 Partilha de Relatórios

Deve ser possível partilhar relatórios não só por intermédio do SEEPlus, mas também por outros canais como *email* ou impressão física.

5.1.8 Requisitos Não Funcionais

Uso de cubos OLAP como avança de marketing. Não sendo obrigatório recomenda-se a utilização da plataforma SSAS.

Implementar o Data Explorer numa arquitetura baseada em *web services*.

Disponibilidade de serviço 24/7 apenas com quebras de serviço para *upgrade* para novas versões.

5.2 Funcionalidades

Antes de se analisar um pouco da arquitetura e dos meios usados para implementar os requisitos definidos, será exposto um resumo das funcionalidades para contextualizar a necessidade de alguns objetos arquitetónicos analisados posterior a este capítulo.

5.2.1 Menu de Navegação

Sendo o Data Explorer um *plugin* para o Excel, ele alega-se como uma expansão das funcionalidades já existentes. Desta forma apresenta novos menus na barra superior, numa *tab* denominada de “SEE-EXPLORER”, ver Figura 21. See-Explorer é o nome comercial dado pela equipa *marketing* da Inovretail. O nome inicial de Data-Explorer e See-Monitor foram outras opções, contudo o nome do projeto mantém-se Data Explorer.

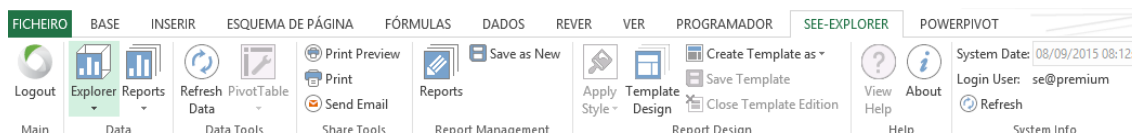


Figura 21 - Data Explorer - Barra de Navegação

O *login* será a primeira operação que o utilizador tem de efetuar. Na Figura 22 é possível observar o formulário para inserir o *username* e *password*. O *server url* corresponde ao ambiente SEEPlus que o Data Explorer se irá ligar. Um utilizador pode ter acesso a mais do que um ambiente. Esta funcionalidade está pensada para a comercialização do projeto como um produto onde um retalhista pode aceder a vários ambientes. Esta situação pode acontecer se por exemplo houver várias insígnias ou marcas onde cada uma tem o seu OLAP ou ambiente SEEPlus. Só com o login efetuado com sucesso o utilizador pode refrescar dados, descarregar ou criar novos *templates*.

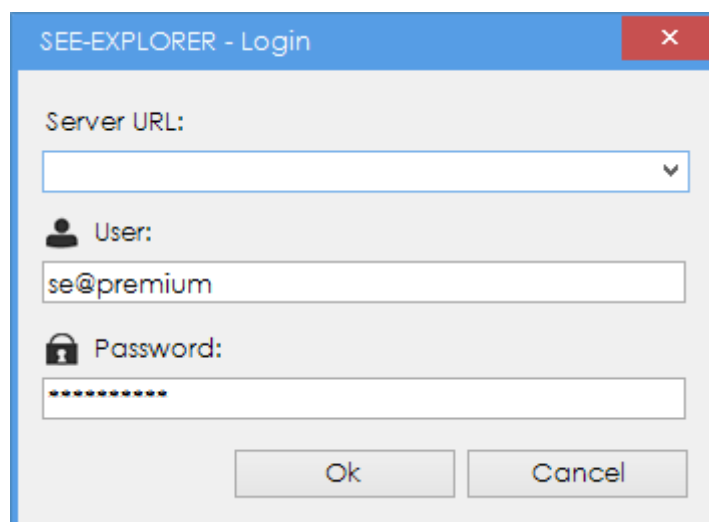


Figura 22 - Data Explorer - Ecrã de Login

Após o *login* efetuado com sucesso os *menus*, que podem ser observado na Figura 21, podem ficar ativos ou inativos consoante as permissões do utilizador.

O *logout* não é mais do que desconectar o Excel dos cubos OLAP, privando os utilizadores não *logados* de explorarem certas funcionalidades e dados.

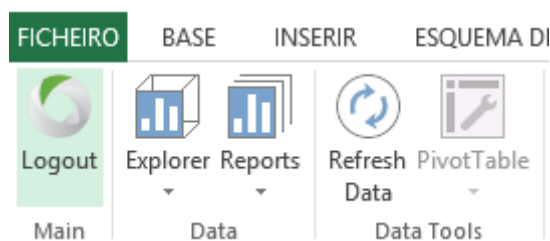


Figura 23 - Data Explorer - Menu de Logout

5.2.2 Funcionalidades Chave

O Data Explorer tem 8 áreas de trabalhos principais com menus chave que respondem a necessidades do utilizador em diferentes senários:

1. Main – Menus de sistema, *login* e *logout*;
2. Data – Acesso a dados, cubos ou relatório já carregados;
3. Data Tools – Ferramentas de manipulação e atualização de dados;
4. Share Tools – Partilha de relatórios, por exemplo *email* ou impressão em papel;
5. Report Management – Carregamento e pesquisa de relatórios;
6. Report Design – Ferramentas de conceção de relatórios;
7. Help – Menus de informação e ajuda do Data Explorer;
8. System Info – Informações de Sistema e sessão do utilizador.

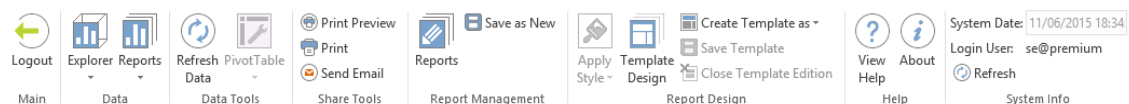


Figura 24 - Data Explorer - Áreas de Trabalho e Menus

5.2.3 Exploração de Dados

O Data Explorer faz uso dos cubos OLAP para facilmente fornecer aos utilizadores ferramentas fáceis e dinâmicas com recurso a *pivot tables* que com poucos cliques consiga fazer uma questão complexa ao sistema de dados.

O menu de *Data > Explorer* lista, por categorias, os vários cubos ou tabelas de factos, que existem no OLAP, ver Figura 25. Neste momento existem 3 categorias, indicadores para lojas, funcionários ou operações.

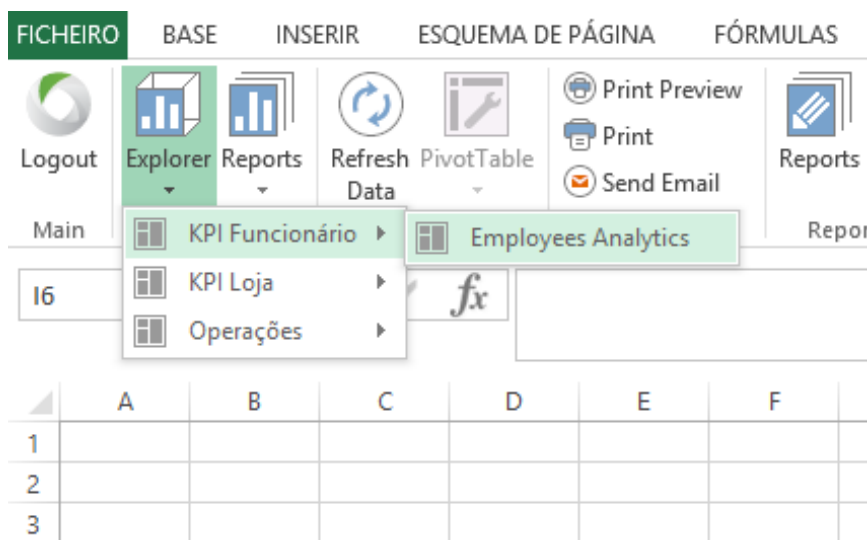


Figura 25 - Data Explorer - Lista dos factos disponíveis por categoria

Quando seleccionamos um cubo numa das categorias, imediatamente é criada uma *pivot table* ligada aos dados, onde no menu do lado esquerdo do Excel aparece uma lista com as métricas do cubo e as suas dimensões, ver Figura 26.

	A	B
1	Rótulos de Linha	MRG NET VALUE
2	S01-2013	159 340,84 €
3	S02-2013	154 100,95 €
4	S03-2013	105 261,87 €
5	S04-2013	109 378,11 €
6	S05-2013	109 944,48 €
7	S06-2013	107 227,86 €
8	S07-2013	116 856,51 €
9	S08-2013	98 535,83 €
10	S09-2013	91 693,65 €
11	S10-2013	100950,2356
12	S11-2013	100022,3294
13	S12-2013	125038,2567
14	S13-2013	165995,7243
15	S14-2013	120103,1456
16	S15-2013	92150,6906
17	S16-2013	99225,0785
18	S17-2013	130772,5416
19	S18-2013	128071,6261
20	S19-2013	122909,1816
21	S20-2013	121656 7719

Campos da Tabela ...

Escolha campos para adicionar ao relatório:

TIK QTY

TIK QTY ADJ

TIK QTY HOM

▶ Calcs

▶ Calendário

▶ Hierarquia Calendário

▶ Hierarquia Semanas

Semana

Arrastar campos entre as áreas abaixo:

FILTROS	COLUNAS

LINHAS

VALORES

Hierarquia Se... MRG NET VAL...

Figura 26 - Data Explorer - Pivot table e Interface para criar pesquisa

Por razões de âmbito, não vai ser explicada aqui a construção de uma *query* através da interface do Excel, assume-se que o leitor tem a base necessária para perceber as diferenças entre *filters*, *columns*, *rows*, *values*. Aos interessados podem consultar o livro *Pivot Table Data Crunching* (Jelen & Alexander, 2011) ou a patente da Microsoft (Patente N° US6626959 B1, 1999) para obter informação nesta matéria.

5.2.4 Relatórios

O *menu Reports* lista os relatórios por categorias. Estes relatórios e categorias são definidos pelo utilizador, aquando a construção e posteriormente geridos na área de *Report Management*. Estes mistos de línguas com os *menus* a Inglês e Português acontece porque alguns *menus* são controlados pelo utilizador, e foram definidos dessa forma.

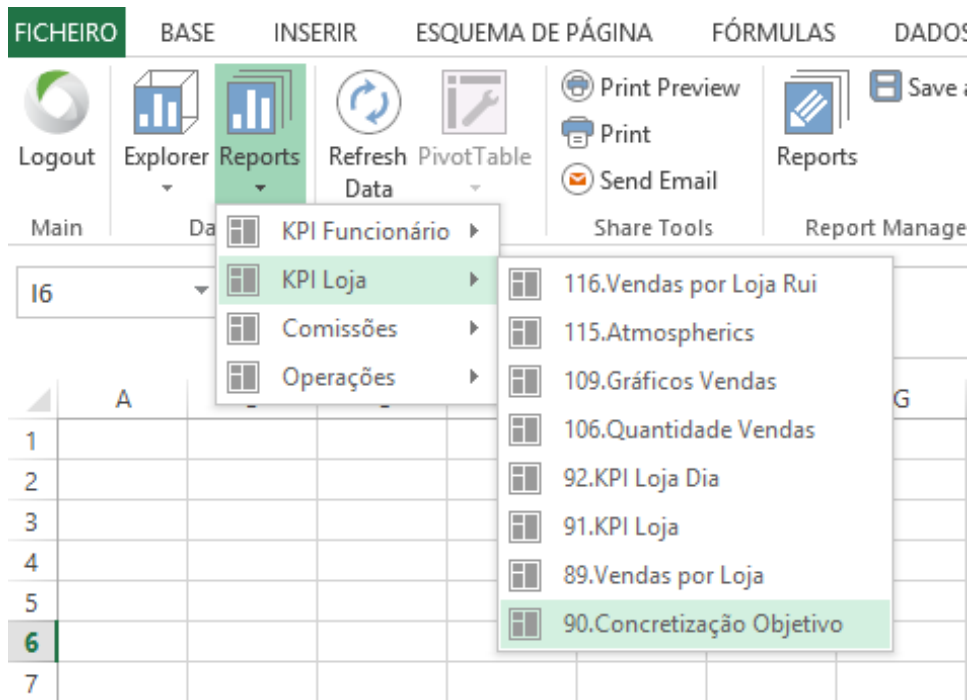


Figura 27 - Data Explorer - Carregamento de Relatório

Quando o utilizador seleccionar um relatório da lista, na Figura 27, é criada uma nova folha de cálculo com o relatório guardado no sistema.

A riqueza destes relatórios não está só na apresentação e forma dos dados mas principalmente no conteúdo e conhecimento que nos pode trazer. No entanto para isso eles tem de mostrar uma imagem da realidade, caso contrário podem mostrar falsas verdades. Para mostrar a última imagem que o sistema conhece, a Figura 28 mostra o menu *Refresh* que atualiza todos os *pivot tables* que estão ligados aos cubos do Data Explorer.

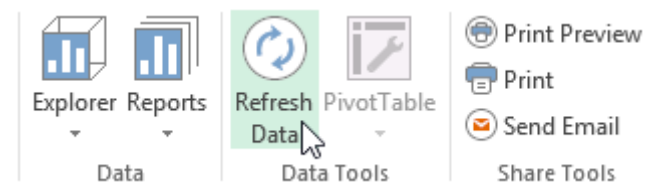


Figura 28 - Data Explorer - Refrescamento dos dados

5.2.5 Pivot table

Muitas vezes quando refrescamos os dados as folhas de dados têm tendência ficarem desformadas como formatações base cor ou tipo ou tamanho de letra ou a largura das colunas.

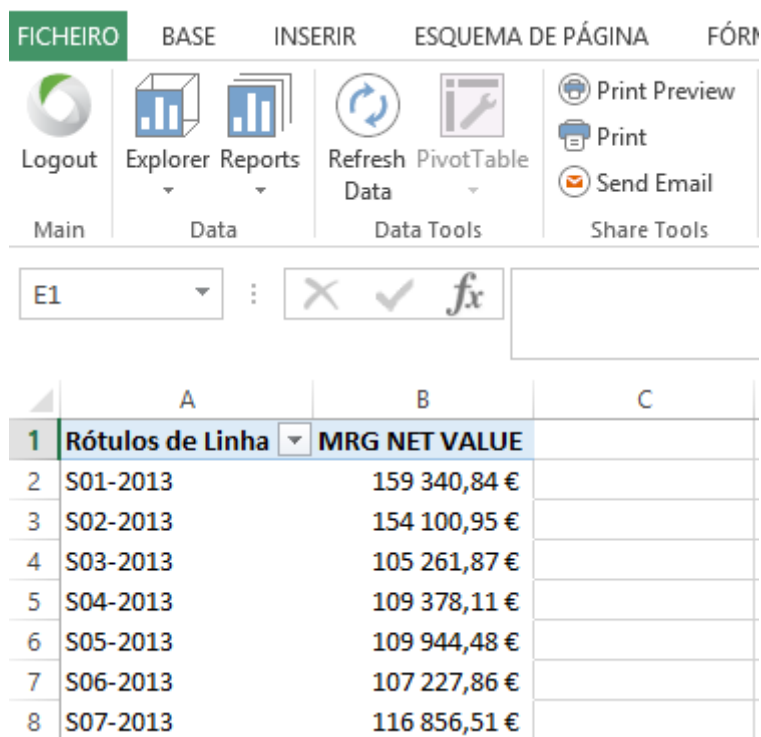


Figura 29 - Data Explorer - Formatação de *Pivot table*

Este conjunto de *menus* vem resolver este problema, para que o utilizador defina quais são as células onde ele deseja que preservada a formatação original e quais as quais as colunas que devem ajustar ao tamanho do texto.

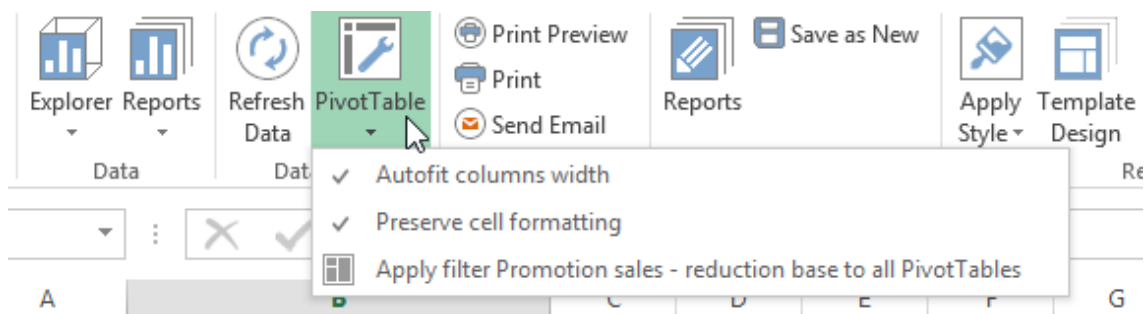


Figura 30 - Data Explorer - Menu Autofit columns e Preserve cell formatting

Para da formatação de células das *pivot tables*, que podemos ver na Figura 30, a lista de menus seguinte pretende ser um facilitador de filtros para *pivot tables*. Quando estamos numa folha de calculo que pode ter vários *pivot tables* e esses *pivot table* têm dimensões em comum, o utilizador pode querer aplicar um filtro igual nas duas tabelas. Aqui podem ser geridos os filtros e aplicados a todas as *pivot tables* com apenas um clique.

5.2.6 Gestão de Relatórios

Esta funcionalidade permite aos utilizadores gerir todos os relatórios existentes em sistema (criar, apagar ou editar).

Todas as ações têm em conta a folha de cálculo que está selecionada no momento. A formatação do *report* nada tem a ver com as funcionalidades do Data Explorer, com exceção às *pivot tables* ligadas aos cubos. Depois de contruído basta clicar em “Save as new”, como na Figura 31, para que o mesmo seja guardado.

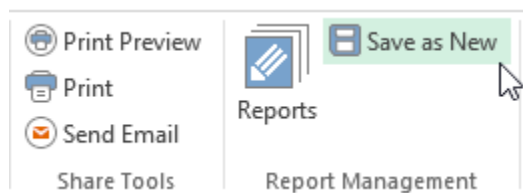


Figura 31 - Data Explorer - Guardar Relatório

Depois disso uma nova janela aparece, ver Figura 32, onde é necessário preencher alguns campos como: nome do relatório, opções de partilha, categoria entre outros.

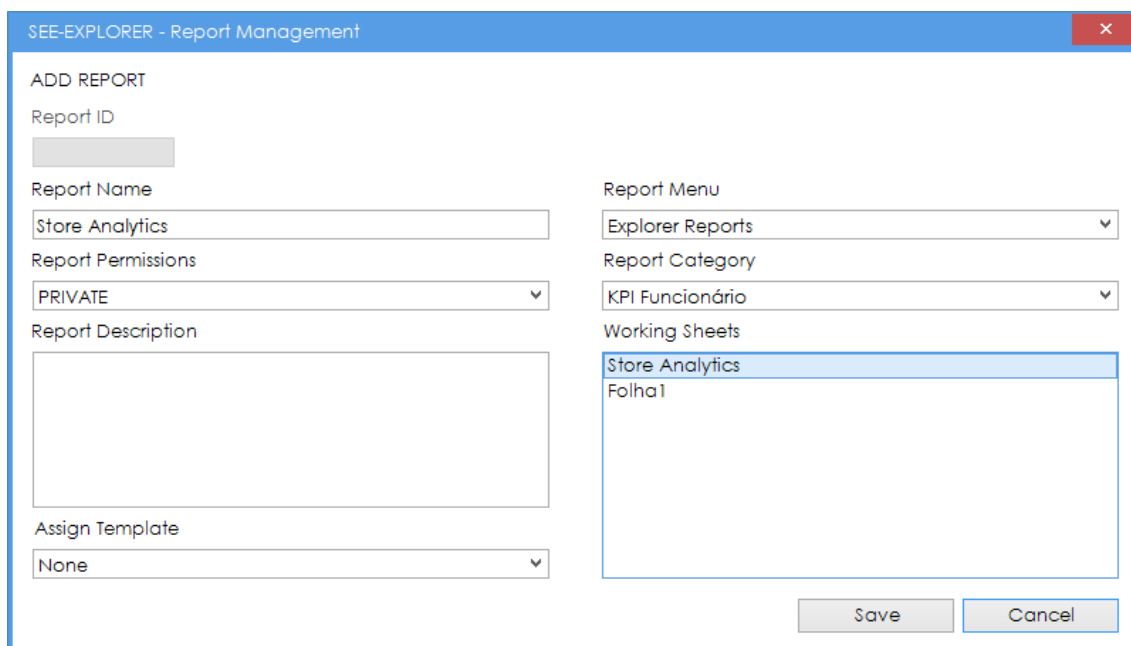
A screenshot of the 'SEE-EXPLORER - Report Management' dialog box. The title bar reads 'SEE-EXPLORER - Report Management'. The main area is titled 'ADD REPORT'. It contains several input fields: 'Report ID' (a text box), 'Report Name' (a text box containing 'Store Analytics'), 'Report Permissions' (a dropdown menu set to 'PRIVATE'), 'Report Description' (a large text area), 'Assign Template' (a dropdown menu set to 'None'), 'Report Menu' (a dropdown menu set to 'Explorer Reports'), 'Report Category' (a dropdown menu set to 'KPI Funcionário'), and 'Working Sheets' (a list box containing 'Store Analytics' and 'Folha1', with 'Store Analytics' selected). At the bottom right, there are 'Save' and 'Cancel' buttons.

Figura 32 - Data Explorer - Formulário para guardar relatório

Finalizado o preenchimento dos campos e clicando no botão “Save”, o mesmo é guardado em sistema e dependendo das permissões de partilha, os restantes utilizadores podem abrir e editar o relatório.

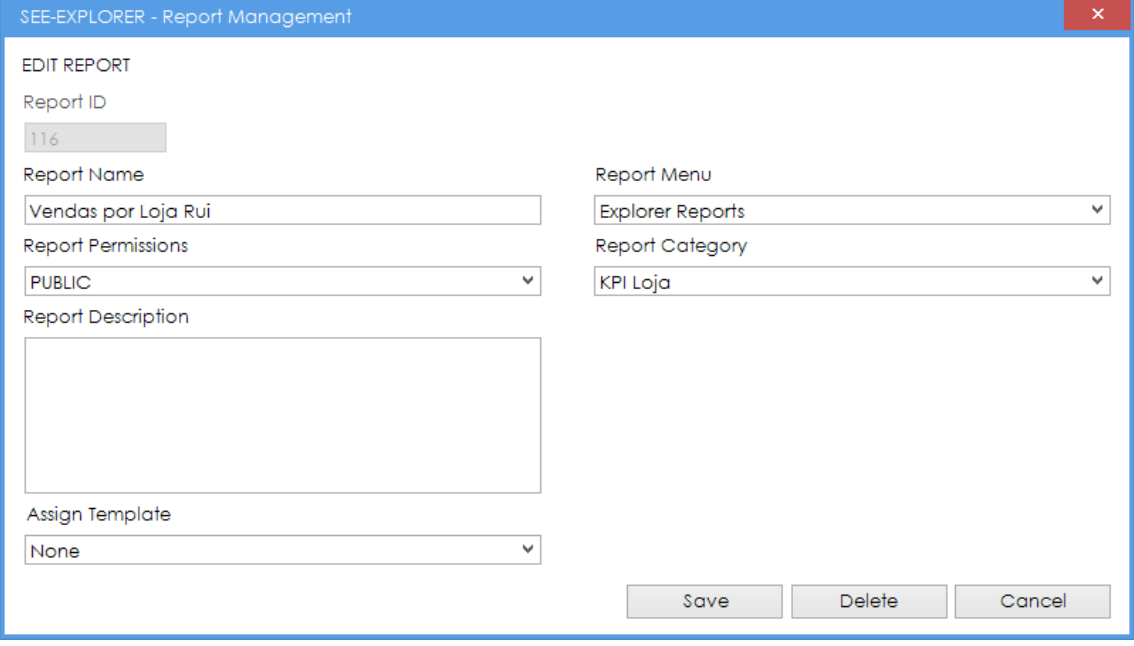
É possível pesquisar relatórios de outros utilizadores e eventualmente editar, bastando para isso usar o menu “Reports” da Figura 31 que abrirá uma janela como a que podemos ver na Figura 33.

<input type="checkbox"/>	ID	Name	Menu	Permissions	Action
<input type="checkbox"/>	124	Simulação Comissões	Explorer Reports	PUBLIC	
<input type="checkbox"/>	123	Comissões Colaboradores	Explorer Reports	PUBLIC	
<input type="checkbox"/>	122	Comissões Gerentes	Explorer Reports	PUBLIC	
<input type="checkbox"/>	118	Analise Pessoal Levis	Explorer Reports	PUBLIC	
<input type="checkbox"/>	117	Vendas por Loja - Store Specific	Explorer Reports	PUBLIC	
<input type="checkbox"/>	116	Vendas por Loja Rui	Explorer Reports	PUBLIC	
<input type="checkbox"/>	115	Atmospherics	Explorer Reports	PUBLIC	
<input type="checkbox"/>	109	Gráficos Vendas	Explorer Reports	PUBLIC	
<input type="checkbox"/>	107	Exploração de Incidentes	Explorer Reports	PUBLIC	
<input type="checkbox"/>	106	Quantidade Vendas	Explorer Reports	PUBLIC	
<input type="checkbox"/>	95	KPI Colaborador Dia	Explorer Reports	PUBLIC	
<input type="checkbox"/>	93	KPI TOP Colaboradores	Explorer Reports	PUBLIC	

Figura 33 - Data Explorer - Lista de relatórios existentes

Observamos na lista os botões para se conseguir editar ou eliminar um relatório e a coluna *Permissions* para se perceber se o relatório é Público ou Privado do utilizador. Ainda é possível observar uma seta com o texto “Search All” que mostra uma pequena *popup* com alguns filtros que se conseguem aplicar à lista de relatórios.

O botão de editar não serve para editar o relatório em si mas as informações dele como se pode observar na figura abaixo.



SEE-EXPLORER - Report Management

EDIT REPORT

Report ID
116

Report Name
Vendas por Loja Rui

Report Permissions
PUBLIC

Report Description

Report Menu
Explorer Reports

Report Category
KPI Loja

Assign Template
None

Save Delete Cancel

Figura 34 - Data Explorer - Editar relatório existente

5.2.7 Formulários de Perguntas e Informações

Dado que futuramente este projeto terá interesses comerciais, foi necessário ter atenção a alguns pormenores como *icons*, tipos de letra e pormenores em alguns ecrãs. As figuras seguintes mostram as *popup* de informação e questões que normalmente ver nas aplicações desktop.

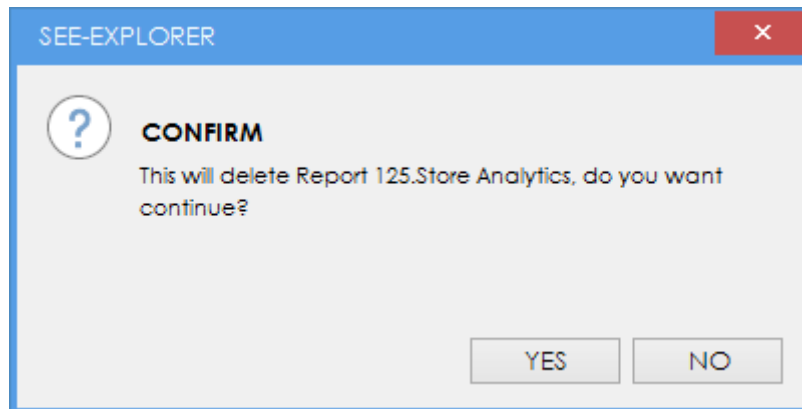


Figura 35 - Data Explorer - Mensagem de confirmação

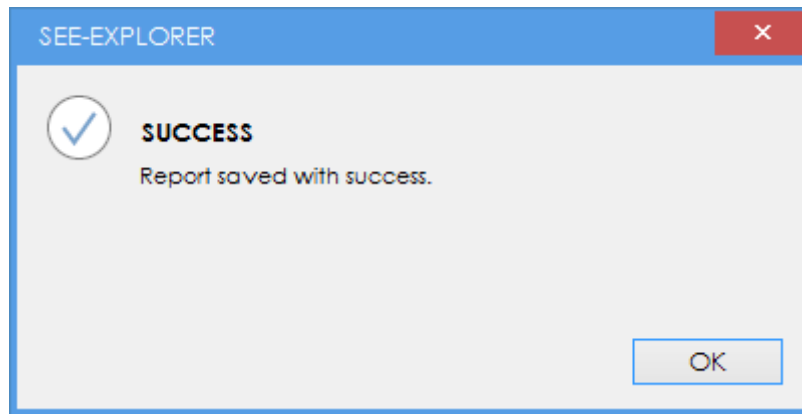


Figura 36 - Data Explorer - Mensagem de sucesso

5.3 Arquitetura

Como já analisado no capítulo 3.2.1, a arquitetura do SEEPlus conta com vários órgãos mecânicos, isto é elementos ativos que realizam trabalho por comando de ações de utilizadores ou tarefas agendadas. O Data Explorer veio trazer um novo órgão mecânico com a necessidade de interligar cubos OLAP com o Excel. Estes cubos analíticos vieram trazer velocidade aos dados e facilidade de exploração que num DW seria mais difícil explorar e as pesquisas mais lentas.

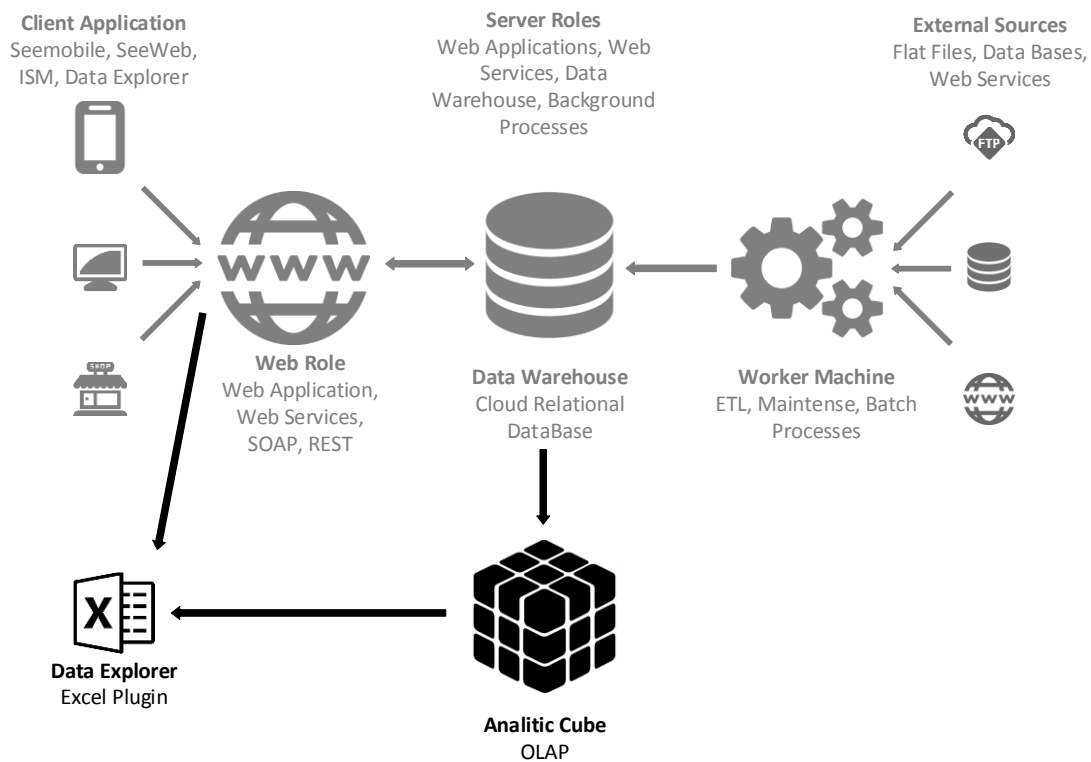


Figura 37 - Nova arquitetura do SEEPlus com o Data Explorer

Quando tentava perceber quais as tecnologias que poderiam ajudar no desenvolvimento deste *plugin*, as primeiras a analisar foram as tecnologias já existente, pela simples razão de poupar custos à solução e por outro lado reutilizar o que já existe.

No que toca a tecnologia existente, o DW do SEEPlus é um SGBD Microsoft Azure na versão 2008. Os primeiros testes tiveram maus resultados em termos de tempos de resposta por restrição de algumas ferramentas de indexação que tornariam os pedidos mais rápidos. Um facto que prova isso é que no DW tem factos repetidos em várias granularidades para que dependendo da questão que precisemos dos dados é necessário consultar esse facto numa estrela onde o facto está à granularidade desejada, como podemos ver na figura seguinte.

METRICS_HOUR	METRICS_DAY	METRICS_WEEK	METRICS_MONTH
TIME_ID	TIME_ID	TIME_ID	TIME_ID
STORE_ID	STORE_ID	STORE_ID	STORE_ID
SALES_VALUE	SALES_VALUE	SALES_VALUE	SALES_VALUE
SALES_QTY	SALES_QTY	SALES_QTY	SALES_QTY

Figura 38 - Estrelas com factos em várias granularidades diferentes

Esta estrutura redundante não é simples para o utilizador. No entanto torna-se bastante versátil e rápido num contexto de *data warehouse*.

A solução seria usar um sistema analítico como o OLAP, onde estivessem apenas os dados necessários para contruir relatórios deste *plugin*. Para além de que o uso de cubos OLAP é um dos requisitos não funcionais, no entanto como prova de conceito outras alternativas foram validadas.

Depois da decisão, questionou porque o OLAP não tinha sido uma alternativa até então, onde se obteve a resposta clara e aceitável quando se deparou com os custos e manutenção necessária para um sistema deste. Em todo o caso é um custo que tem de ser assumido para o sucesso comercial deste projeto.

Arquitectualmente foram tidas em atenção algumas recomendações para maximizar a performance dos SSAS aplicando boas práticas sugeridas pela Microsoft. É factual que o SSAS não é escalável porque não está preparado para máquinas com dezenas de processador e gigabytes de memória (Lee & Unkroth, 2010). Para juntar à limitação de escalabilidade, este projeto também esteve sempre limitado a um único

servidor onde estaria alojado o serviço do SSAS. A melhor prática seria usar uma arquitetura idêntica à da Figura 39.

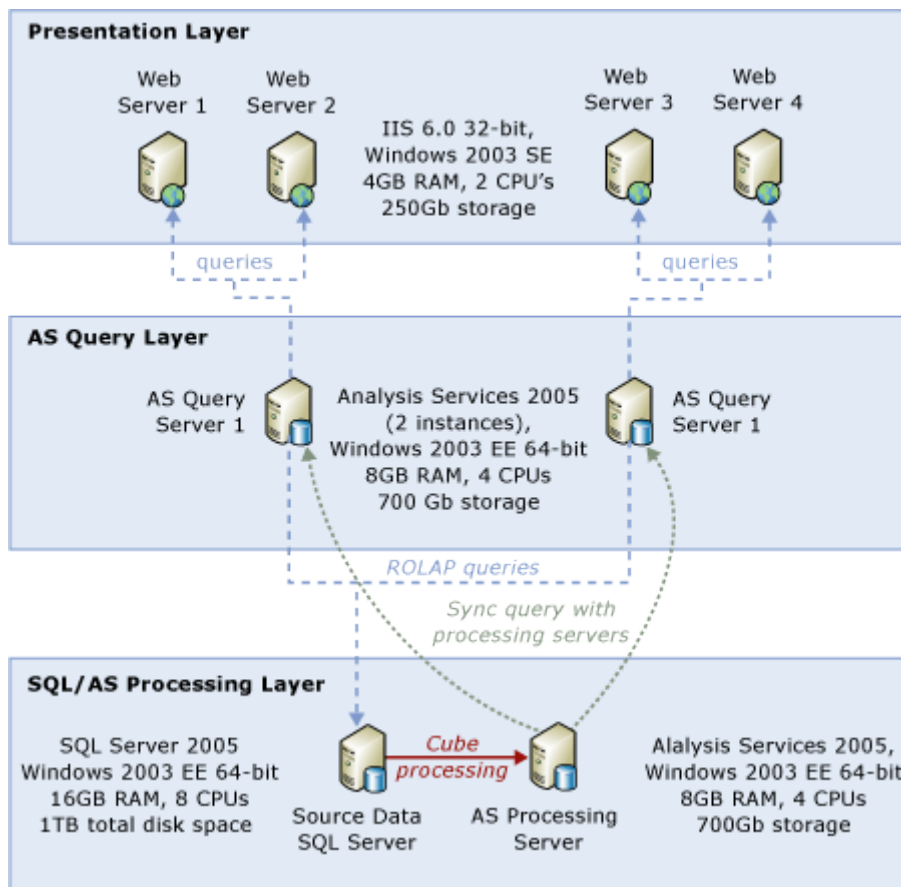


Figura 39 - Arquitetura SQL Server com SSAS (Denny Lee, 2007)

Nas camadas de dados e apresentação a escalabilidade e sobrecarga dos servidores está à responsabilidade da plataforma Azure em *cloud* que garante que são alocados mais recursos sempre que necessário. Desta forma sempre que temos vários utilizadores a aceder a um servidor IIS será replicado de imediato para novas instâncias. O mesmo acontece com a base de dados, que muitas vezes consegue estar a 240% dos seus recursos. Nestas situações a Microsoft não especifica quantas instâncias estão a correr para suportar 240% dos recursos disponíveis. Teoricamente é impossível usar 240% dos recursos no entanto estas percentagens são referentes aos recursos disponíveis quando nenhum recurso é usado. Ou seja, uma máquina SQL Azure com 50 DTU, quando são usados 240% dos recursos significa que estão usados 120 DTU. DTU é uma medida que a Microsoft não especifica bem o que é. Número de processadores, quantidade de memória principal e taxa de transferência de dados são fatores que influenciam os DTU consumidos mas a fórmula a Microsoft nunca clarificou. Todas as ações de replicação de

instâncias na base de dados, mas que também da web role, apoiam o requisito da disponibilidade 24/7.

Como já foi referido a escalabilidade do servidor OLAP poderia ser conseguido com a replicação de instâncias. No entanto no contexto de *cloud* a Microsoft não disponibiliza nenhum serviço escalável para esta tecnologia, ao contrário ao que já faz com o SQL Server, Hive ou serviços *web*.

5.4 Desenvolvimento

5.4.1 Modelação dos Cubos OLAP

A modelação desenvolvida em SSAS na versão 2008 R2 já foi justificada anteriormente quando a seleção das tecnologias tendo em conta algumas restrições do projeto. O servidor utilizado é uma VM com o Windows Server Data Center que está alojada na *cloud* Azure.

A criação da máquina é suportada por um assistente do Azure com *templates* onde algumas configurações são personalizáveis, Figura 40. No caso a máquina é uma A2 [2 core, 3.5GB RAM], (Microsoft, Virtual Machines Pricing, 2015).

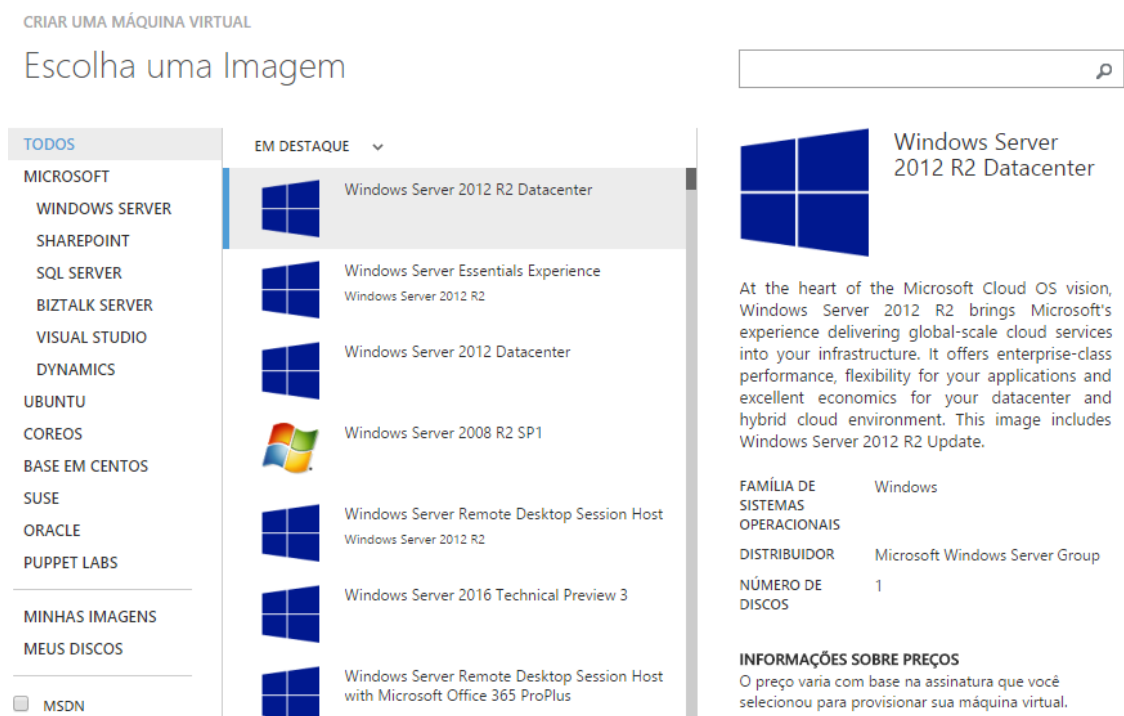


Figura 40 - Assistente de Criação de Máquina Virtual Windows Server

Após configurado todo o ambiente foi criada uma base de dados no SSAS para Data Explorer, Figura 40.

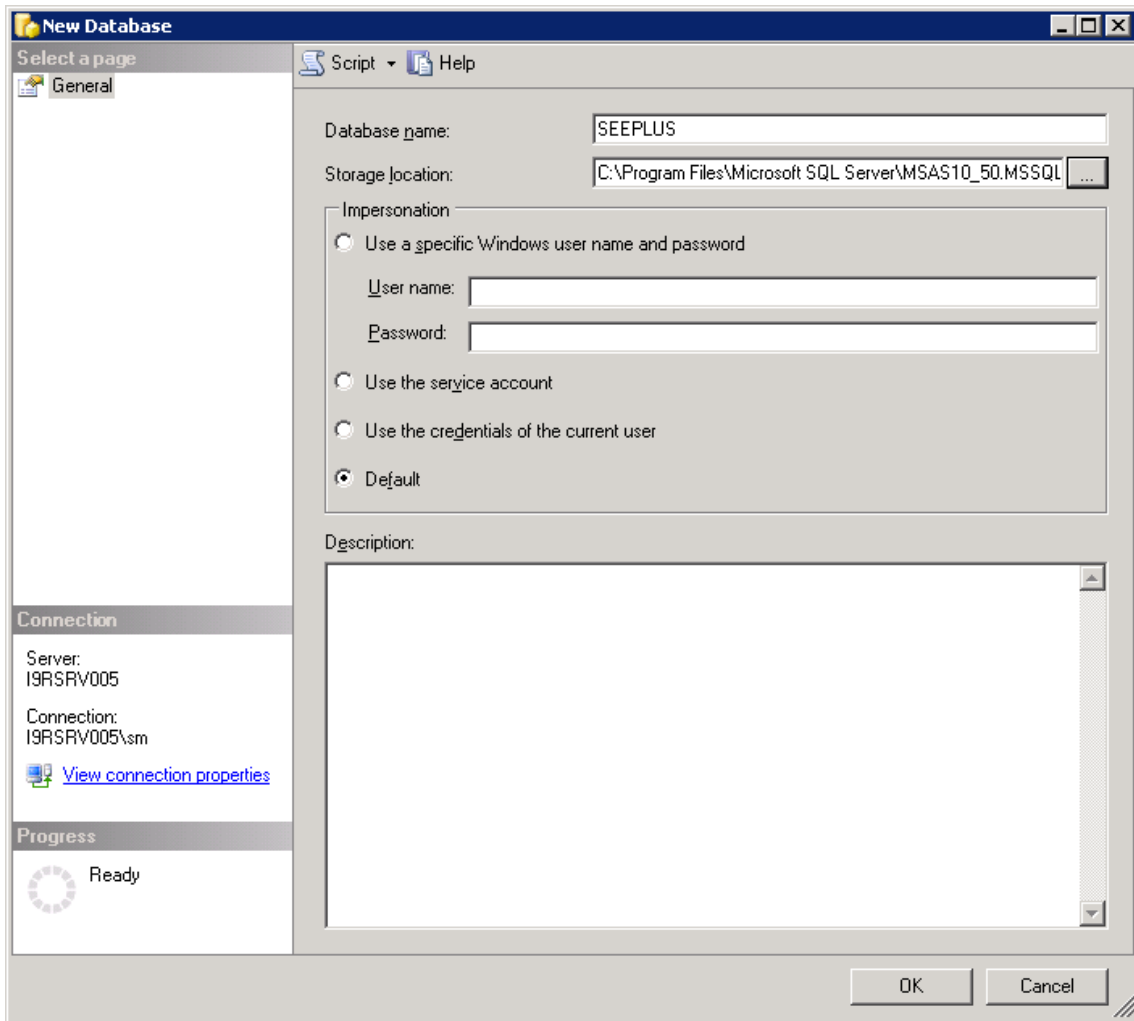


Figura 41 - Criação de nova base de dados no SSAS

Foram criados 3 cubos, como podemos ver na Figura 42. Cada um foi desenhado para formar uma estrela, com uma tabela de factos e as respetivas dimensões. No total existem 12 dimensões.

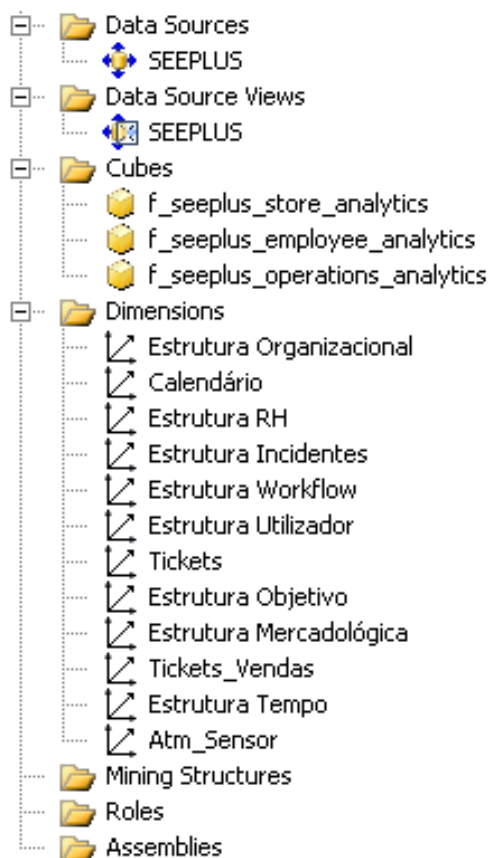


Figura 42 - Componentes do projeto do SASS

Para suportar o carregamento de dados para o Cubo foram criadas 15 *views* sobre a base de dados do SEEPlus. A opção pelas *views* justifica-se pela facilidade de manutenção em caso da alteração do modelo de dados original. Assim apenas as *views* têm de ser novamente compiladas tendo em conta o novo modelo de dado. Isto cria uma interface entre os dois sistemas para que nenhum tenha conhecer a estrutura do outro.

As principais *views* são:

- `ext_out.vw_f_atm`: Contêm factos atmosféricos (temperatura, humidade, níveis CO2, ruído sonoro) por hora/dia/localização;
- `ext_out.vw_f_objective`: Contêm os factos dos objetivos de uma loja. Estes podem ser objetivo do total de vendas, orçamento das vendas, objetivo de vendas por categoria de produtos por dia/localização;
- `ext_out.vw_f_sls_tkt`: Contêm os factos referentes a *tickets* de venda e devoluções à granularidade hora/dia/localização;
- `ext_out.vw_sls_tkt`: Tal como a *view* anterior, esta contém os factos das vendas de *tickets* de vendas e devoluções mas ao nível da transação. Com

esta *view* é possível calcular indicadores que com a *view* agregada não é possível por causa de algumas métricas não aditivas;

- `ext_out.vw_sls_loc`: Contêm também factos de vendas à granularidade da loja/dia que contêm outro tipo de métricas que não podem existir na granularidade das *views* de vendas anteriores;
- `ext.out.vw_stk_day`: Contêm factos dos níveis de *stock* passados ao dia/localização;
- `ext_out.vw_store_operation`: Contêm factos de operações de loja. Estas operações são pedidos de trabalho gerados pelo ISM, um dos módulos do SEEPlus. Pedido de trabalho é por exemplo uma requisição de reposição de *stock*, manutenção de loja, limpeza, informações entre outros.

Nas dimensões encontram-se, para além das que já se falaram na granularidade dos factos, as seguintes:

- `ext_out.vw_dim_calendar`: Atributos do tempo, hora/dia;
- `ext_out.vw_dim_location`: Atributos de localização, loja, *head office*, armazém;
- `ext_out.vw_dim_employee`: Atributos de empregado;
- `ext_out.dim_objective_type`: Atributos para um tipo de objetivo;
- `ext_out.vw_dim_product`: Atributos de um produto vendido ou em *stock*;
- `ext_out.vw_dim_user`: Atributos de utilizador para uma operação de loja;
- `ext_out.vw_dim_workflow`: Atributo do estado de uma operação de loja.

Estas *views*, fonte de dados para dimensões e factos, modelaram-se os 3 cubos: *store analytics*, *employee analytics* e *operations analytics*.

Os cubos aqui analisados fazem referência a assuntos relacionados com retalho de vários departamentos: Financeiro, Distribuição, Compras, Recursos Humanos ou Marketing. Nos exemplos clássicos pode-se analisar apenas vendas para um departamento da organização, aqui trata-se os assuntos aos objetos de negócio: Lojas, Empregados Loja, Recursos Humanos, Processos de Negócio.

Os nomes das métricas são compostos por uma série de siglas que caracterizam as métricas. Estas siglas depois de assimiladas facilitam a leitura em alternativa a descrições extensas. A tabela seguinte descreve cada uma.

Tabela 4 - Siglas que contituem as métricas do cubo

Sigla	Descrição
SLS	Métrica de vendas (SaLeS).
VALUE	Valor das métricas para diferenciar da quantidade.
QTY	Quantidade da métrica para diferenciar do Valor: Exemplo: valor e quantidade de vendas (QuanTitY).
GROSS	O valor da métrica antes de retirados os impostos.
NET	Valor da métrica está limpa de impostos.
HOM	Valor homólogo face à dimensão calendário do registo atual, segundo a regra dia do ano N é igual ao dia do ano N-1, Exemplo 2015-05-05 > 2014-05-05.
HOM ADJ	Valor homólogo, tal como a sigla HOM, mas com uma regra diferente definida pelo retalhista. A regra preferencial dos retalhistas é o cruzamento de datas festivas homólogas e dia da semana com dia da semana homólogo. Por exemplo: Cruzar métricas do fim-de-semana homólogo, Natal, Páscoa, Carnaval ou outros eventos Anuais cujo dia e o mês podem variar.
OBJ	Objetivo de vendas (OBJctive Target).
BDG	Orçamento de vendas (BuDGet).
MRG	Margem de Vendas (MaRGin).
CUS	Métricas de contagem de Clientes (CUStomer Footfall)
TIK	Métricas sobre Tickets, granularidade mais baixa de uma operação de venda (TicKet).
STK	Métrica de <i>stock</i> de artigos.
HUM	Humidade relativa do ar.
TEMP	Temperatura do ar.

Com esta composição de siglas, por exemplo métrica [SLS NET VALUE HOM] representa o valor (VALUE) de vendas (SLS) limpo de impostos relativo (NET) à data homóloga (HOM). Este e outro exemplo podem ser analisados na exploração dos cubos nos capítulos seguintes.

5.4.1.1 Cubo *Store Analytics*

Este cubo diz respeito a todas as métricas para uma loja de retalho. Como ilustrado na Figura 43 vêem-se algumas métricas e o *data source* correspondente, à granularidade das dimensões calendário e estrutura organizacional.

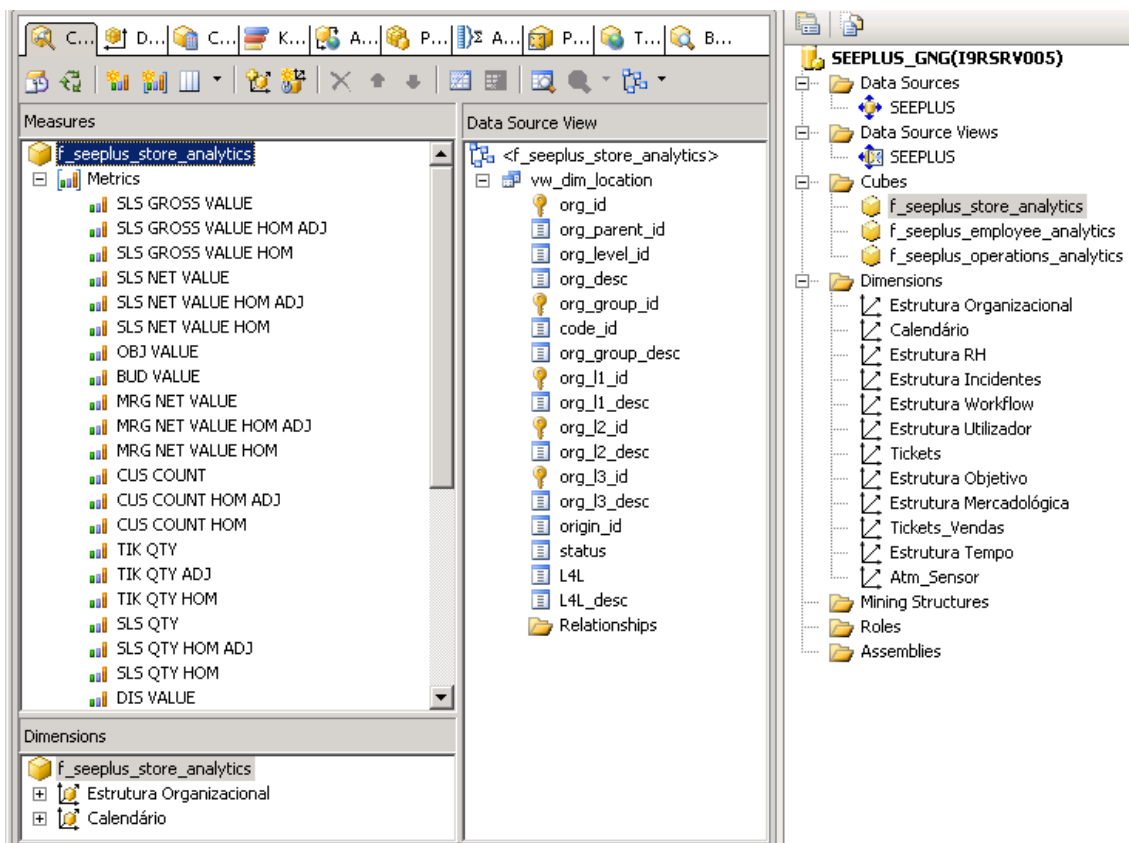


Figura 43 - Estrutura do Cubo de Store Analytics

Nele podem ser consultadas métricas de vendas, objetivos, orçamentos, margem de lucro, contagem de clientes, contagem de *tickets*, *stocks* e dados atmosféricos.

5.4.1.2 Cubo *Employee Analytics*

Este cubo diz respeito a todas as métricas para uma loja/empregado. Como ilustrado na figura seguinte vêem-se algumas métricas e o *data source* correspondente, à granularidade das dimensões calendário e estrutura organizacional, estrutura mercadológica, estrutura de objetivos, *tickets* e sensores de atmosféricos. Em retalho é muito comum aparecerem estruturas de entidades por facilitar o agrupamento por diversos níveis, mais ou menos refinados consoante a pesquisa necessária.

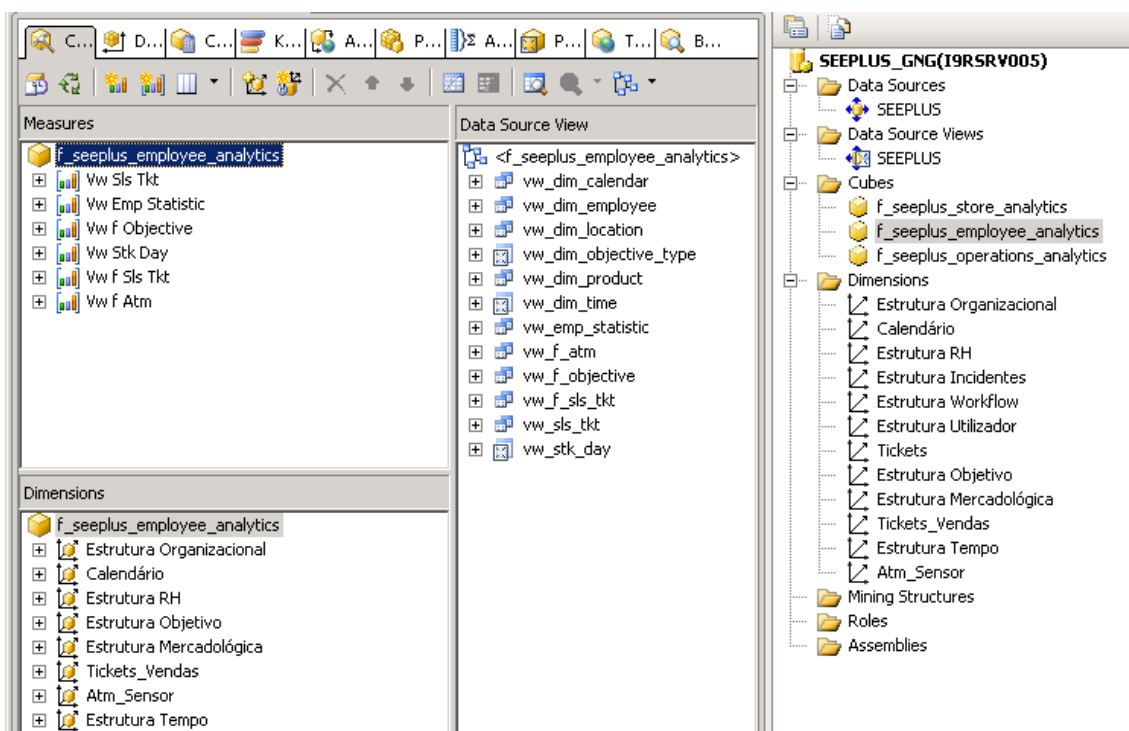


Figura 44 - Estrutura do Cubo de Employee Analytics

Aqui as métricas estão divididas por grupos para uma divisão clara dos assuntos que se tratam neste cubo, pois na verdade pode-se consultar vendas a vários níveis organizacionais. Nele podem ser consultadas métricas de vendas, objetivos, orçamentos, margem de lucro, contagem de clientes, contagem de *tickets*, *stocks* e dados atmosféricos.

5.4.1.3 Cubo *Operation Analytics*

Este cubo diz respeito a todas as métricas de operações em loja. Como ilustrado na figura seguinte vêem-se algumas métricas e o *data source* correspondente, à granularidade das dimensões calendário e estrutura organizacional, estrutura workflow, e estrutura de utilizadores.

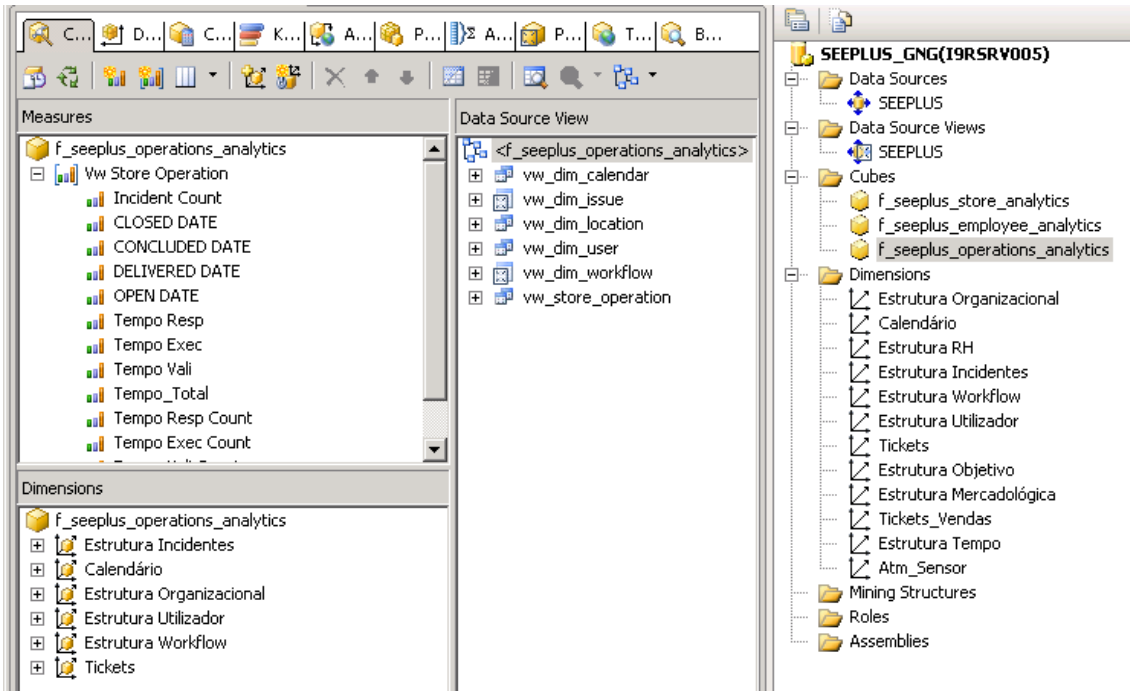


Figura 45 - Estrutura do Cubo de Operation Analytics

Neste cubo é possível analisar as execuções de tarefas a pedido pelo ISM, um dos módulos do SEEPlus, onde empregados ou chefias podem alocar tarefas a colegas ou fazer pedidos de informações. O *workflow* de cada tarefa permite analisar tempos de execução, tempos de resposta ou outras análises que ficam à descrição do utilizador.

5.4.1.4 Estruturas hierárquicas das dimensões

As dimensões já referidas na modelação dos cubos estão fortemente ligadas a estruturas hierárquicas organizacionais. Estas estruturas permitem navegar nos dados a vários níveis de granularidade dentro desta dimensão.

A dimensão estrutura organizacional tem uma hierarquia. Esta divide as localizações organizacionais (lojas, *head office*, armazéns) em 3 níveis, *holding*, insígnia e loja ou local físico. Os primeiros 2 níveis são estruturas lógicas de negócio que dividem as lojas por sociedade gestora e marca/insígnia.

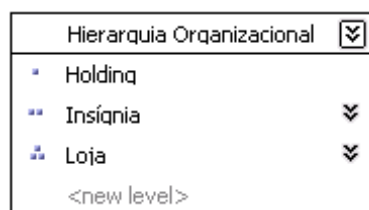


Figura 46 - Hierarquia da dimensão Estrutura Organizacional

Este é só um exemplo, outras dimensões têm hierarquia, mas pela extensão de imagens ou explicações repetitivas, podem ser analisados em anexo.

5.4.2 Integração e Processamento de Cubos

Após a criação das dimensões e cubos no SSAS, os mesmos foram processado para obter os dados mais recentes do DW do SEEPlus. Esta tarefa de refrescamento deve ocorrer periodicamente para que os cubos não fiquem desatualizados.

A Microsoft recomenda o uso do SQL Agent para agendar tarefas no SSAS, no entanto o SEEPlus não usa SQL Server, mas sim o SQL Azure que não contem este SQL Agent (Microsoft, SSAS, 2009).

Para este tipo de problemas o SEEPlus conta com um motor de processos de integração e consolidação para agendar tarefas quer no sistema do SEEPlus como noutros sistemas externos. Neste caso, o servidor SSAS do Data Explorer será tratado com um sistema externo.

Este motor de *batch* assenta numa arquitetura *message broker*, onde tudo o que é informação é representado por uma mensagem numa *queue*. As mensagens são processadas e enviadas para outra *queue*, até chegar à *queue* final. A fonte de mensagens pode ser um ficheiro, *web service* ou uma base de dados que independentemente do formato será uma mensagem com uma determinada *syntax*. O conjunto de fluxos de mensagens entre *queues* formam um processo de integração. Entre cada *queue* existe um programa responsável por tratar e transformar mensagens para serem enviadas a um output do *middleware* de *message broker*. O destino final destas mensagens será um sistema de dados, por exemplo o DW do SEEPlus ou outro sistema externo.

Após a análise do que era este motor de integração do SEEPlus, foi necessário desenvolver um conector para processar os cubos. Este conector teria de herdar uma série de objetos da API do motor e gerar uma mensagem que respeite a *syntax* do motor. Assim todas as mensagens fluem ao longo de todo o processo, sendo perceptíveis por esta *framework*. Este conector terá de tomar uma ação espoletada por uma mensagem do *message broker*, no caso o RabbitMQ (Pivotal, 2015), ver Figura 47.

Para que tudo inicialize o SEEPlus tem uma API para listar e criar agendamento de tarefas, onde as mesmas serão geradas segundo um calendário recorrente. Isto substitui a ausência do SQL Agent do SQL Server. Esta API apenas gera mensagens e

injeta-as no RabbitMQ que ao serem processadas vão chegar de alguma forma ao conector ProcessOLAPConnector.

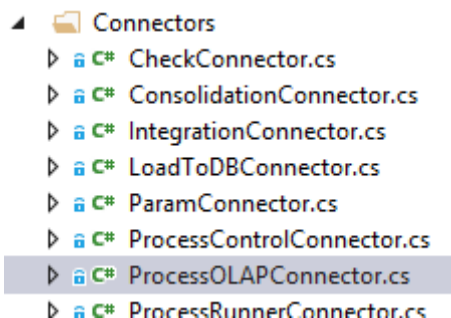


Figura 47 - Alguns conectores do SEEPlus em destaque o ProcessOLAPConnector

Este conector desenvolvido para o Data Explorer, quando é inicializado recebe alguns argumentos que foram definidos no agendamento da tarefa. Assim o mesmo conector pode correr com várias configurações. Estes argumentos serão o nome do servidor a instância do SSAS a processar. A partir do momento que o conector recebe a mensagem o seu objetivo é ligar-se ao SSAS e processar as dimensões e os cubos.

Para tal o conector desenvolvido em C# teve de fazer referência a uma DLL da Microsoft especifica para se conectar a este tipo de servidores (Microsoft, Connect to Analysis Services, 2015).

Por questões de confidencialidade não será possível mostrar um exemplo do código, no entanto a figura seguinte mostra o algoritmo em pseudocódigo ao estilo C#.

```
void ProcessOLAPThread() {  
  
    //Wait for message of RabbitMQ  
    var message = Job.RabbitMQ.WaitMessage();  
  
    //Create authentication and connecto to SSAS  
    var principal = new Impersonator(message.username, message.domainName,  
message.password)  
    var server = new Microsoft.AnalysisServices.Server(principal);  
  
    //Find data base  
    var database = server.Databases.FindByName(message.databaseName);  
  
    //Process all Dimensions  
    foreach (Dimension dim in database.Dimensions)  
        dim.Process(processTypeDim);  
  
    // Process all Patitions of MeasureGroups in all Cubes  
    foreach (Cube cube in database.Cubes)  
        foreach (MeasureGroup mg in cube.MeasureGroups)  
            foreach (Partition partition in mg.Partitions)  
                partition.Process(processTypeMG);  
  
}
```

Figura 48 - Algoritmo de processamento dos cubos

Este algoritmo é bastante simples e facilmente convertível em código compilável em C#. A maior dificuldade foi, mais uma vez, compreender a arquitetura por trás deste módulo do SEEPlus. Com a escassa documentação o código e a disponibilidade das pessoas são a única ajuda possível.

5.4.3 Plugin Excel em Windows Forms

Para estender as funcionalidades do Excel existem 3 possibilidades, *template*, *workbook* e *addin*. O *template* e *workbook* são mais para a finalidade de adicionar *macros* que correm em *background* num documento Excel ou usar essas mesmas *macros* para criar *templates* de documentos. Em ambos os casos está restrito o acesso a alguns eventos e propriedades de um documento Excel. Um *addin* acrescenta funcionalidades ao Excel e é independente do documento. Ele recebe eventos gerados pelos *inputs* do utilizador e daí executa uma ação. Em todo caso existem sempre limitações na manipulação de documentos Excel em qualquer um dos tipos. Não havendo documentação oficial para comparar prós e contras entre os tipos, foi possível analisar alguns pontos de vista de bloggers e daí tirar algumas conclusões.

Este projeto pretende com poucos conhecimentos que um utilizador consiga usar o Excel para consultar dados e extrair conhecimento do seu negócio. O *addin* permite criar menus e janelas personalizadas e ao mesmo tempo aceder a todo o conteúdo de um documento, podendo ser manipulado programaticamente. Com esta desagregação dos documentos o *addin* seria a melhor opção.

Para o desenvolvimento de um *addin* Excel a Microsoft dispõe de *templates* de projeto para o Visual Studio, Figura 49.

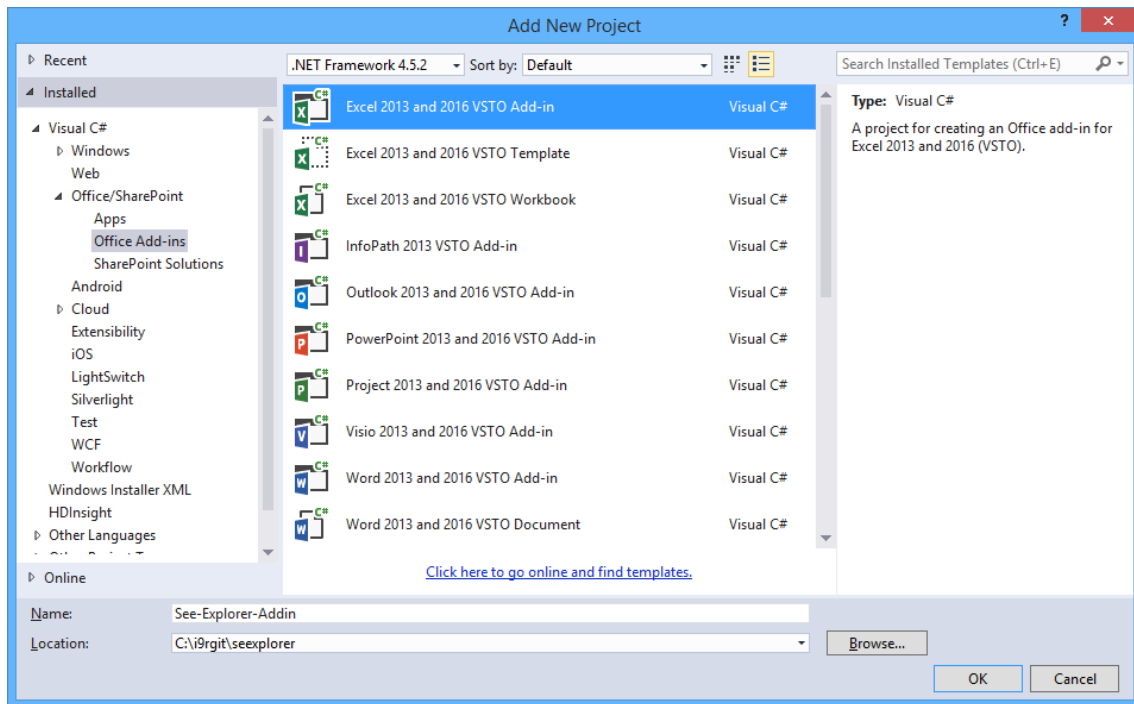


Figura 49 - Novo projeto Excel 2013 Addin no Visual Studio 2015

Este projeto do tipo *class library*, idêntico a outros, gera uma DLL (Dynamic Link Library) com uma particularidade para funcionar em Excel. Ele contém um objeto que herda de uma biblioteca do Microsoft Office (Microsoft.Office.Tools.AddInBase, Figura 50) que é responsável por inicializar o AddIn e *renderisar* os menus que aparecem no ecrã inicial do Excel, como na Figura 21.

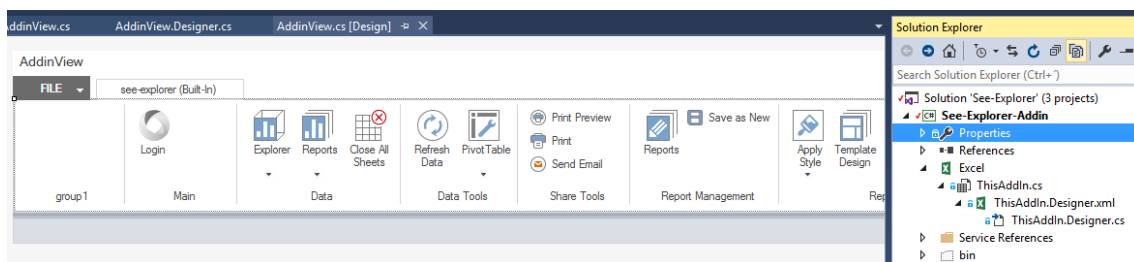


Figura 50 - Objeto principal de inicialização do plugin do Excel

Na figura anterior ainda é possível observar o *designer* que o Visual Studio disponibiliza para desenhar os botões ao estilo “*drag and drop*”. A cada elemento da interface existem muitos eventos, como *click*, *double-click*, *focus*, *mouse-hover*, que podem ser subscritos para capturar os *inputs* do utilizador e executar as ações da funcionalidade correspondente.

5.4.3.1 Arquitetura e Metodologia de Desenvolvimento

Todo o projeto de desenvolvimento minimamente complexo, sem fazer juízos de valor sobre projetos complexos, necessita de uma metodologia que determine orientações a tomar durante o processo de desenvolvimento. Ao mesmo tempo deve ajudar a desbloquear dúvidas estruturais da arquitetura do artefacto de *software*.

Para a escolher do *pattern* de desenvolvimento foram analisadas os MVVM e MVP, capítulos 4.5.1 e 4.5.2 respetivamente. Pelas conclusões tiradas da análise, o MVP é aquele que se melhor adapta a realidade deste projeto pela componente orientada aos objetos e interface em *Windows Forms*.

Seguir uma metodologia de desenvolvimento iria trazer sempre prós e contras. A favor a abstração da implementação, a divisão de problemas menores e mais fáceis de validar e facilidade de revisão da implementação face a alterações na conceção do software. Contra o facto de a abstração dificultar a leitura do código, onde uma camada de abstração pode ter outras camadas de abstração por trás, e ainda o risco de uma implementação do *pattern* com defeitos que poderá criar uma situação de *anti-pattern*.

Após a reflexão de alguns dos pontos anteriores e elaborados alguns diagramas, surgiu a representação dos módulos do *plugin* na Figura 51, organizados em camadas com a classificação de cada elemento no *pattern* MVP.

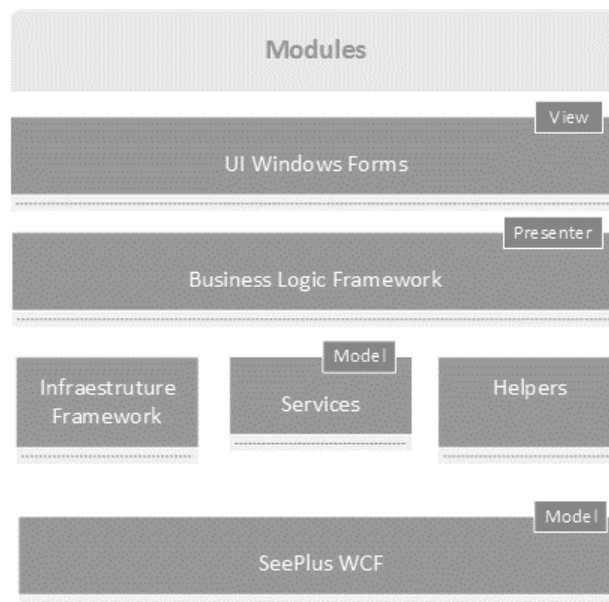


Figura 51 - Módulos do Data Explorer

Na primeira camada de apresentação da GUI, foram implementadas as *views* do *plugin*. Esta apenas tem os ecrãs sem qualquer tipo de lógica de negócio Para uma *view* ser inicializada necessita de um *presenter* que manipule todos os eventos na *view* e

reflita a ação num *model*. Esta camada de lógica de negócio é uma abstração de objetos que se herdaram para criar uma *framework* de processamento e apresentação de dados. Finalmente na componente *model*, os serviços consumidos do SEEPlus são sempre traduzidos em objetos de negócio para serem entregues à camada de *presenter*. Os módulos *Infraestructure* e *Helpers* são utilitários de código que facilitam tarefas de manipulação e conversão de objetos. O *Infraestructure* para tratamento de objetos com regras de negócio e os *Helpers* para manipulação tipo de dados como genéricos como *strings*, listas, datas, *XML* entre outros.

5.4.3.2 Implementação do pattern MVP

Numa implementação correta do *pattern* as várias camadas devem ser implementadas por objetos diferentes. Esta diferenciação de objetos originou a criação das *views* na Figura 52 e dos *presenters* e *models* na Figura 53.

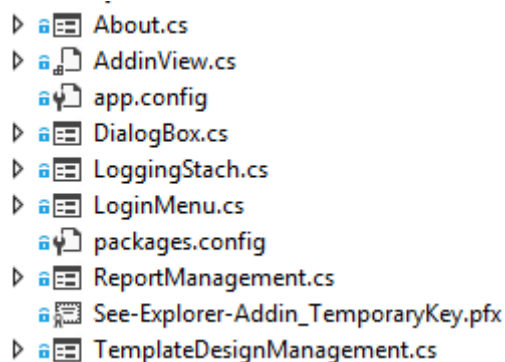


Figura 52 - Objetos View do Pattern MVP

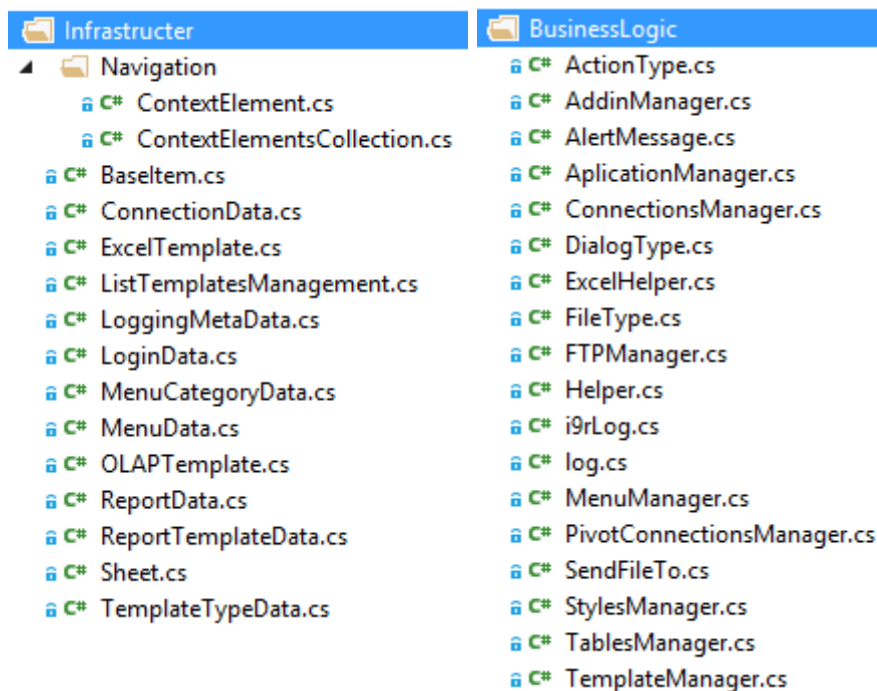


Figura 53 - Objetos Presenter e Model Data do Pattern MVP

5.4.4 Modelo de Dados

O modelo de dados da Figura 54 representa as tabelas e as relações criadas para suportar os menus e janelas que aparecem nas funcionalidades já ilustradas no capítulo 5.2. Outras tabelas de sistema, representadas no modelo de dados da Figura 55, estão a ser usadas mas não foram desenhadas no âmbito deste projeto. Por exemplo parametrizações de sistema (*olap connection*, *FTP server* ou *default values*) é armazenado numa tabela para o efeito. Outras funcionalidades como *login* e permissões são geridas pelo SEEPlus que o Data Explorer apenas consome esses serviços. No entanto algumas configurações foram necessárias para aceder a esse modelo de dados.

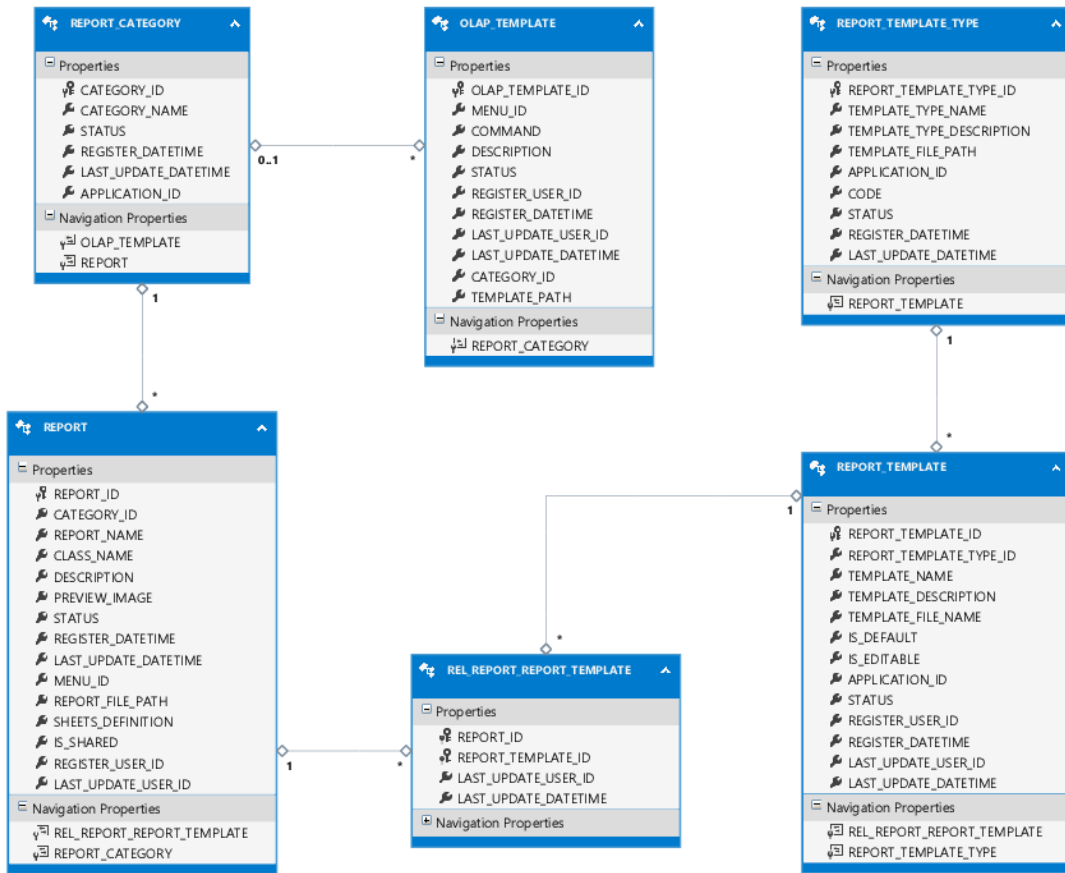


Figura 54 - Modelo de dados para o plugin Data Explorer

A Tabela 5 descreve o que cada elemento do modelo de dados armazena.

Tabela 5 - Descrição dos elementos do modelo de dados

Tabela	Descrição
OLAP_TEMPLATE	Um registo desta tabela define um cubo existente no SSAS, para que nos menus sejam listados os cubos disponíveis.
REPORT	Armazena a lista de relatórios disponíveis no SEEPlus por categoria. Juntamente guarda o nome do ficheiro e local no FTP.
REPORT_CATEGORY	Categorias de relatórios que depois serão usadas para a lista de menus dos <i>reports</i> e <i>templates</i> .
REPORT_TEMPLATE	Armazenamento de <i>templates</i> de relatórios.
REPORT_TEMPLATE_TYPE	Esta tabela de sistema, não pode ser editada pelo utilizador, contém os tipos de <i>templates</i> neste momento só existe o tipo <i>Header</i> .
REL_REPORT_REPORT_TEMPLATE	Relação entre um relatório e os <i>templates</i> nele incorporado. Ainda que os <i>templates</i> estejam limitados a um <i>header</i> , no futuro será possível definir um <i>footer</i> e assim um relatório poderá ter mais do que um <i>template</i> .

O modelo de dados completo da Figura 55, contém as tabelas de sistema, onde as tabelas a cor verde representam a informação do utilizador e da sessão ativa, encarnado as tabelas com a definição de permissões por utilizador e *role* e a azul as tabelas do Data Explorer que também estão no modelo da Figura 54. Ainda existe uma tabela a negro que define as aplicações/módulos do SEEPlus.

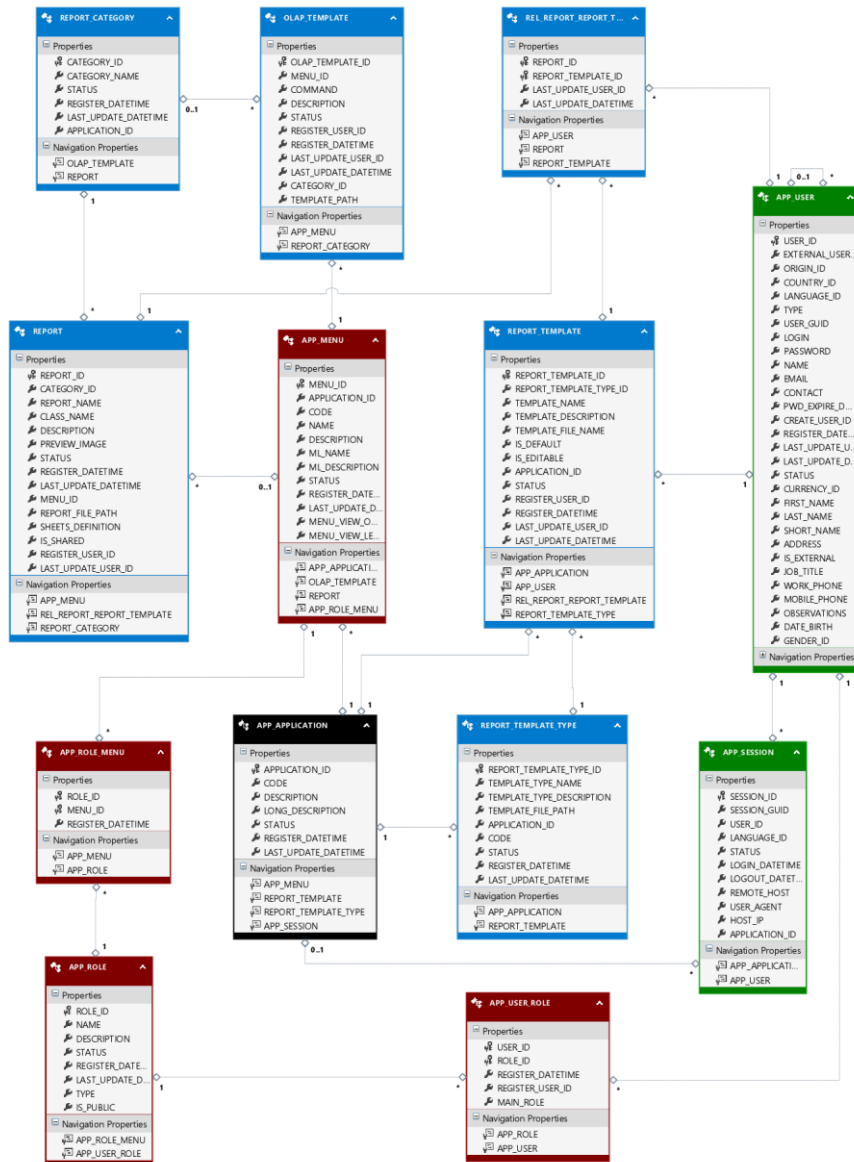


Figura 55 - Modelo de dados completo com as tabela de sistema

5.4.5 Assistente de Instalação

Com o cariz empresarial que este projeto herdou, a necessidade de um instalador foi uma necessidade forçosa, apesar que não estar na lista dos requisitos.

A Microsoft teve a preocupação de integrar o Visual Studio com algumas *frameworks* para assistência personalizada de instalação de *software*, entre elas Visual Studio Setup and Deployment Projects, InstallShield 2010 Limited Edition, Windows Installer XML Toolset entre outras. As últimas ferramentas enumeradas são as que foi possível reunir documentação.

A seleção uma ferramenta, gerou a necessidade de uma escala para medir as várias funcionalidades, onde um fator de ponderação daria mais importância a ferramentas que cumprissem com a essas funcionalidades. A coluna fator de ponderação

da Tabela 6 tem uma escala simples de 1 a 3, que classifica a importância da funcionalidade para este projeto. Na escala, 3 significa que a funcionalidade é importante e 1 pouco importante. Depois fez-se um cruzamento entre as funcionalidades e as *frameworks* onde 1 significa que cumpre com a funcionalidade e 0 não cumpre com a funcionalidade. Finalmente fez-se a soma do produto destas colunas saber qual das 3 ferramentas cumpre melhor para os objetivos do assistente que se pretende.

$$\sum_{i=1}^{i=1} \text{Funcionalidades} \quad \text{FatorPonderação}_i * \text{Ferramenta}_i$$

Figura 56 - Equação utilizada para a seleção da ferramenta de assistente de instalação

Tabela 6 - Comparativo entre Ferramentas de Assistente de Instalação

Funcionalidade (i)	Fator Ponderação [1-3]	VS Setup and Deployment Projects	InstallShield 2010 Limited Edition	Windows Installer XML Toolset
1. Integração com Visual Studio	3	1	1	1
2. <i>Designers</i> para personalizar o instalador	2	1	1	0
3. Gerar ficheiro MSI	3	1	1	1
4. Várias versões do .NET Framework	3	1	1	1
5. Configurar serviços do Windows	1	1	1	1
6. Configurar grupos e contas de utilizadores	1	0	0	1
7. Configurar propriedades para o IIS	1	1	1	1
8. Criar novas <i>web applications</i>	1	0	0	1
9. Instalar e configurar certificados do IIS	1	0	0	1
10. Instalar e configurar bases de dados SQL	1	0	0	1
11. Modificar ficheiros XML	2	0	0	1
12. Instalar pré-requisitos	3	1	1	0*
13. Personalizar a interface do <i>setup</i>	3	1	1	1

14. Integração com o <i>MSBuild</i>	2	0	1	1
15. Gestão de ações personalizadas	3	1	1	1
16. Suporte para grandes atualizações	3	1	1	1
17. Suporte para pequenas atualizações e <i>patches</i>	2	0	0	1
18. Migração de projetos de <i>setup</i>	1	0	1	1
Pontuação Global	36	25	28	31
Pontuação Relativa	100%	69%	78%	86%

Os resultados finais não deixam qualquer tipo de dúvida com a tendência clara para o WiX (Windows Installer XML Toolset). Em análise mais pormenorizada o WiX falha uma funcionalidade importante na instalação de pré-requisitos. Numa pesquisa rápida percebeu-se que era fácil arranjar um *workaround* para a situação adicionando scripts de *Pre* ou *Post Build*. As dependências não são geridas pela *framework* da ferramenta mas poder-se-á considerar como uma funcionalidade presente no WiX. Assim estamos numa situação em que o WiX teria 34 pontos em 36 possíveis.

A versão profissional do InstallShield seria também uma boa opção. A razão pelo qual não foi analisada reside no preço. Um dos requisitos para esta ferramenta de assistente de instalação é que não poderia ter custos. Desta forma o InstallShield Pro nunca seria opção.

Após algum desenvolvimento do assistente em WiX não houve grandes barreiras. Ainda assim foi necessário criar um instalador para a arquitetura x86 e outro individual para x64.

5.4.5.1 Instalação do WiX no Visual Studio 2015

A instalação é simples basta aceder à página oficial da WiX Toolset e descarregar a última versão. Para este projeto foi utilizada a versão 3.9 e migrado mais tarde para a versão 3.10. A migração aconteceu também porque a meio do projeto a solução foi migrada do Visual Studio 2010 para o Visual Studio 2015 e assim manteve-se também a *framework* WiX atualizada (WiX, WiX Toolset, 2015; WiX, WiX Toolset, 2015; soundarmoorthy, 2015).

Quando procedemos à instalação podemos ver uma janela na Figura 57.



Figura 57 - Instalação do Assistente WiX

5.4.5.2 Criação do novo Projeto WiX

Quando finalizado podemos criar um projeto novo no Visual Studio como podemos ver na Figura 58.

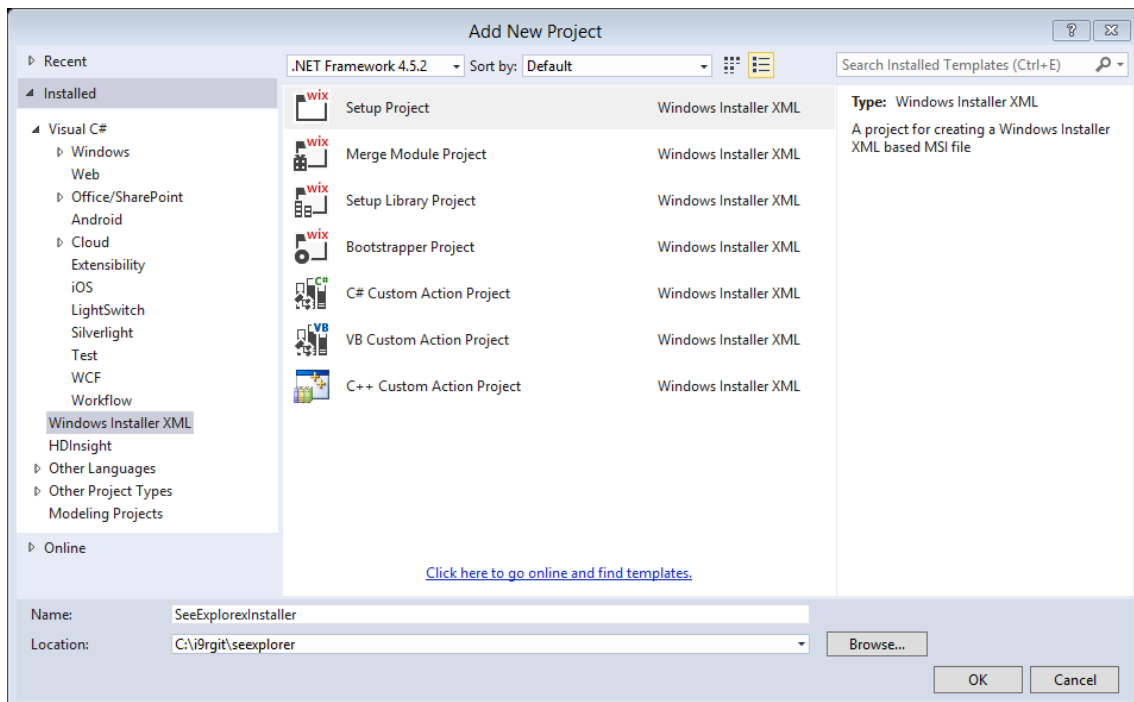


Figura 58 - Criação de um novo projeto WiX

Após a criação do projeto foi necessário ler a documentação do WiX, na versão 3. A criação deste projeto foi baseada em *templates* e exemplos encontrado pela internet que posteriormente foram ajustados para a realidade da SEEPlus e do Data Explorer. Na

Figura 59 pode-se observar que foram criadas várias pastas cada uma com o seu objetivo.

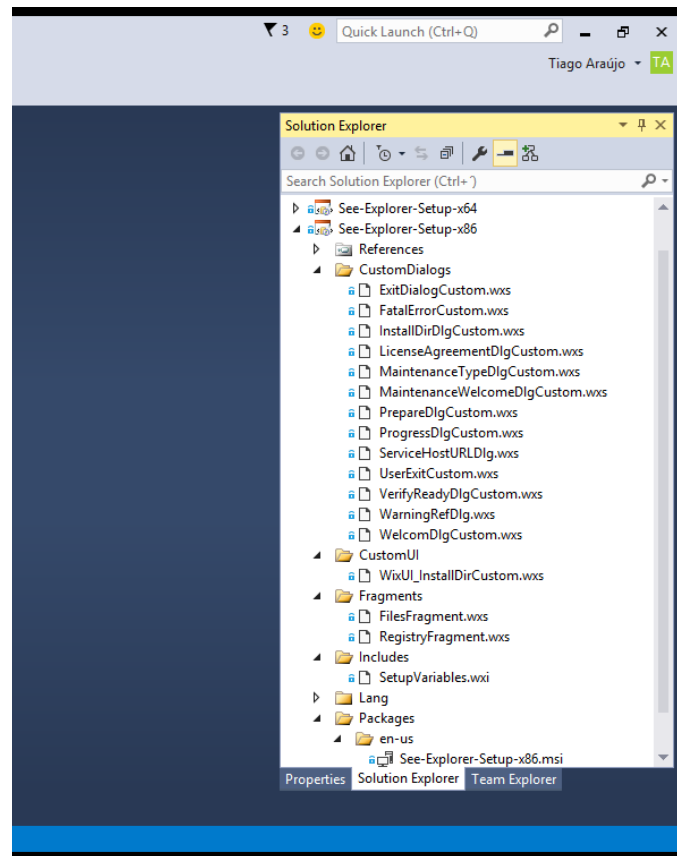


Figura 59 - Estrutura de pastas e ficheiros Xml do WiX

Todos os ficheiros são objetos definidos em XML, como o nome da *framework* sugere. Resumidamente cada pasta tem a função de:

- CustomDialogs: Cada ficheiro representa uma janela distinta que poderá aparecer durante a instalação do Data Explorer, com referência a outras janelas, dependências ou ações dos utilizadores durante a instalação;
- CustomUI: Aqui estão definidos alguns componentes onde é definido o aspeto visual, tipo de letra, o tipo de ações (botões) que depois pode ser herdado em outros componentes e janelas. A definição do aspeto visual pode ser por herança, ou seja os componentes herdam classes visuais, ou então pode ser por atribuição onde o ficheiro XML está a definir o aspeto específico a um componente sem que este tenha qualquer referência.
- Fragments: São um conjunto de objetos que depois de compilados não podem ser mutáveis, por exemplo uma lista de componentes. Estes objetos depois podem ser referenciados por outros objetos. Por exemplo *registry*

do Windows, os ficheiros que compõem o instalador. Para se perceber melhor, objetos como as janelas são objetos compilados mas que vão tomar um estado e comportamento diferente ao longo da sua execução, já com o tipo de objetos *fragments* isso não acontece.

- Includes: Apenas um ficheiro porque aqui apenas se define algumas constantes usadas noutros objetos XML. Estas constantes são de uso e definição livre. Estão aqui definidas por exemplo a versão *Major*, *Minor* e *Build*, Nome do Produto, *ReleaseFolder*, entre outros.
- Lang: Está definidos as *labels* com as traduções para a língua inglesa. É possível criar traduções para outras línguas criando outros objetos *WixLocalization*.
- Packages: Contêm todos os packages, neste casos os ficheiros MSI (Microsoft Installer, mais recentemente denominado de Windows Installer), a serem instalados quando a execução do assistente de instalação. Neste momento só contêm o MSI para instalar o See Explorer.

5.4.5.3 Instalação do Data Explorer

As imagens seguintes ilustram os passos que podem aparecer ao utilizador quando a instalação do Data Explorer. O departamento comercial da Inovretail decidiu dar outro nome ao projeto: o see-explorer, apenas por questões comerciais e de imagem como em outros produtos como o SEEMobile, SEEMonitor e o SEEWall. No entanto o nome interno do projeto continua a ser o Data Explorer.

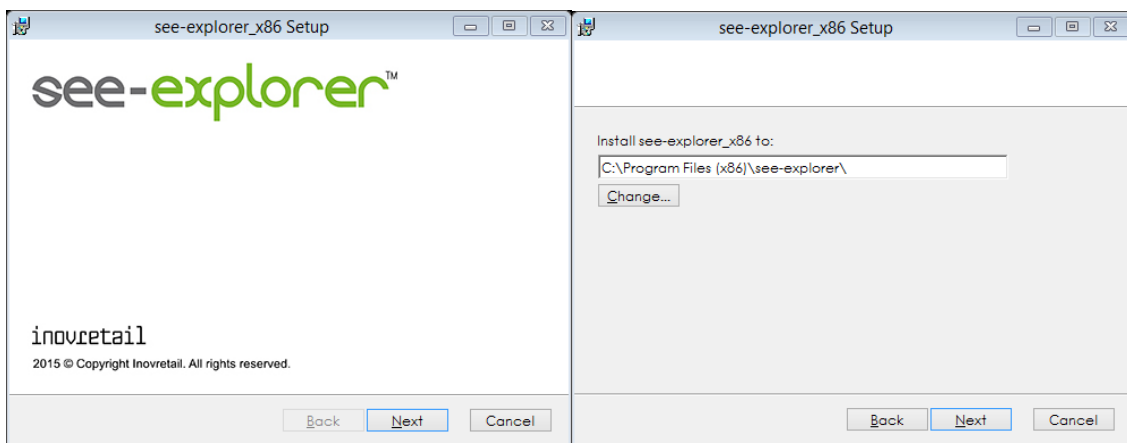


Figura 60 – Assistente de Instalação a) Página Inicial b) Página de seleção da pasta de instalação

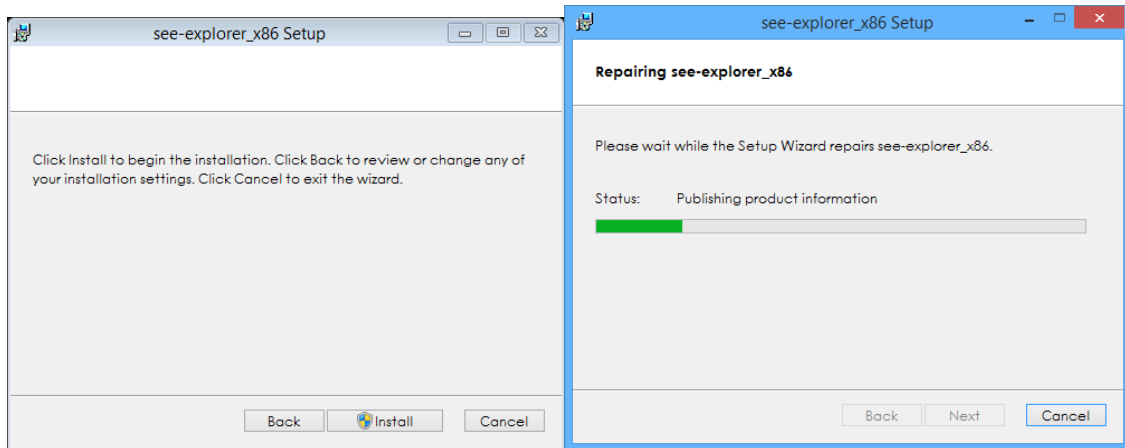


Figura 61 - Assistente de Instalação – a) Página de confirmação da instalação, b) Página de estado

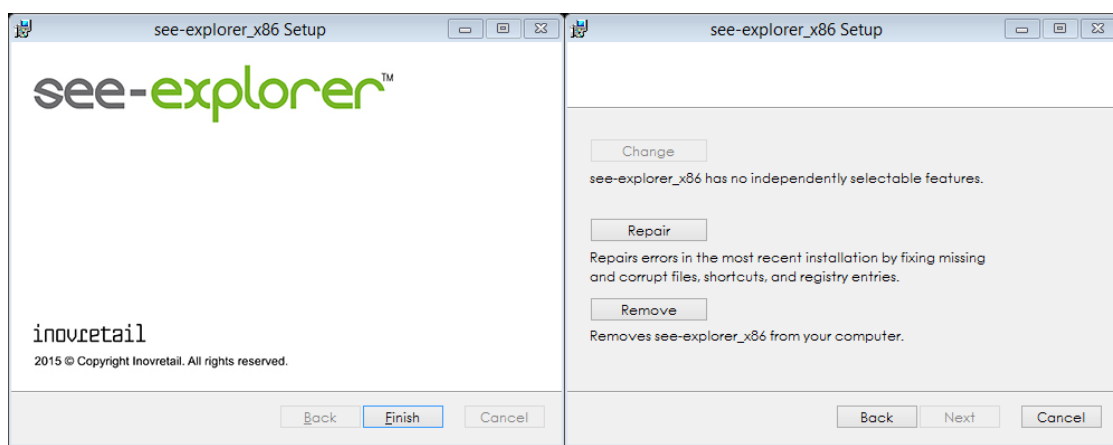


Figura 62 - Assistente de Instalação – a) Página de instalação com sucesso, b) Página de desinstalação

5.4.6 Conclusão

Pelos capítulos anteriores verifica-se que os desenvolvimentos abordaram diversificadas tecnologias e linguagens de programação, para servidor e cliente, em áreas distintas dentro das Tecnologias de Informação como: Administração de base de dados, Business Intelligence, Web/Desktop Developer e até Arquitetura de *Software*.

O desenvolvimento nem sempre foi concluído por fases, começando pelos componentes de servidor passando para os componentes de cliente que consomem os serviços do servidor, algumas vezes foram revistos os desenvolvidos anteriores para que os novos desenvolvimentos fossem concluídos o sucesso desejado, aprendendo boas e más práticas.

De todos os desenvolvimentos o *plugin* Excel foi aquele que deu mais trabalho e mais situações de bloqueio aconteceram. Nem sempre foi possível resolver os problemas da forma ideal mas sempre foram resolvidos com alguns *workaround*. Por se tratarem de problemas muito técnicos não parece relevante entrar em detalhes. Os problemas iniciais da criação da GUI foram mitigados pela utilização da *framework* da

Telerik que trouxe um visual mais fluido e componentes mais versáteis para o utilizador. O modelo de dados para a GUI foi uma coisa natural que foi aparecendo com a definição dos requisitos e das entidades de dados.

Os cubos não trouxeram grandes problemas, mas a criação das *views* para carregamento dos dados obrigou a analisar muitos sub-modelos de dados do SEEPlus. Estes modelos nem sempre respeitavam as mesmas normas como, nomenclatura, tipos de dados ou ligações de chave estrangeira, o que foi um obstáculo.

O componente de refrescamento dos cubos integrado no motor ETL do SEEPlus, apesar de menos importante para este projeto, foi das melhores experiências de desenvolvimento do projeto. A análise numa arquitetura baseada em mensagens, a flexibilidade e mutação dos processos, e velocidade com que os dados conseguem ser processados, foi uma alternativa acertada ao Integration Services (SSIS).

6 Conclusões e perspectivas de desenvolvimento

Este capítulo contém um resumo do trabalho realizado na dissertação, de modo a sumariar os esforços da elaboração deste trabalho assim como as contribuições e desafios que podem vir a ser despoletados.

6.1 Conclusões sumárias

Este projeto, motivado por uma necessidade de negócio, teve como objetivo acrescentar um novo módulo a um produto de BI existente no mercado: o SEEPlus. A somar a gama de produtos da Inovretail, pretende ajudar nas tarefas de um decisor da área de retalho.

Da análise, fruto das informações passadas nas reuniões com o gestor de produto da Inovretail, justificou-se a inclusão do estudo da arquitetura do SEEPlus para ser possível dar sugestões de melhoria e apresentar modelos de dados alternativos que poderiam ter sido implementadas no decorrer desde projeto. Contudo, derivado ao impacto da mudança no SEEPlus, não foi possível passar das sugestões para desenvolvimentos na arquitetura. Uma das dificuldades foi o esforço necessário analisar toda a documentação, código fonte, os módulos e no final ainda desenvolver.

Com o novo módulo do Data Explorer, o principal *output*, constatou-se juntos dos responsáveis do produto, que apesar de não finalizado, os requisitos e funcionalidades estão claras para a equipa do SEEPlus e que as propostas de trabalhos futuros são um bom ponto de partida para as novas versões do Data Explorer. Os próprios gestores de produto da Inovretail ainda não sabem como isto vai ser absorvido pelo mercado e só com feedbacks dos comerciais será possível validar algumas das funcionalidades desenvolvidas bem como ver a viabilidade os trabalhos futuros já propostos. Era interessante a demonstração a um cliente durante os trabalhos da tese, mas tal não foi possível.

A calendarização sofreu algumas alterações, como tarefas novas e outras alteradas exigindo abordagens diferentes para concluí-las. Da lista de ricos inicial, veio-se a verificar alguns deles e pensa-se que as ações de mitigação tomadas tiveram impactos positivos. Um dos problemas identificados no motor ETL do SEEPlus na pré-dissertação veio se a confirmar e o Data Explorer também sofreu com as alterações deste motor. Esta nova abordagem de ETP por mensagens está descrita no capítulo 5.4.2.

Olhando para os objetivos iniciais, conseguiu-se reunir opiniões que, a criação de um *plugin* Excel para exploração de dados, foi um objetivo cumprido. Já na parte do estudo da arquitetura do SEEPlus pode não ser claro, mas verifica-se houve muito contacto com componentes do sistema. Os *web services*, *web applications* e o motor de ETL foram mencionados várias vezes o que demonstra a interação com estes objetos do sistema. A presença na migração das plataformas durante a tese de, Visual Studio 2010 para 2015, SQL Azure V8 para V12, Azure SDK 1.6 para 2.7.1 e WiX 9 para 10, trouxe desafios que permitiram aprender com os maus exemplos de *software* que a migração veio a melhorar. No final todas estas ações de alguma forma obrigaram a algumas alterações no plano, mas foi positivo a passagem por este processo.

6.2 Limitações do estudo

Pela natureza empresarial do projeto, este estudo apresenta à cabeça algumas limitações porque o objetivo será sempre obter lucro da ação de investigação, o que num ambiente científico estará focado na geração e difusão de conhecimento para comunidade. Assim muitos pressupostos tem de ser assumidos por imposição do proponente. Estas limitações tornaram os objetivos mais claros e menos suscetíveis a desvios. Pelo contrário o mercado externo que está em contante mudança e influencia mais o mundo empresarial, tal como se verificou nas migrações, referidas anteriormente, onde a causa foram as alterações do SDK Azure.

A primeira limitação atribuir-se pela Inovretail ser parceiro Microsoft e consequentemente *Microsoft-oriented*. Logo tecnologias de DW, OLAP e *reporting* serão sempre tecnologias Microsoft e de preferência compatível com a *cloud* Azure. Esta limitação foi uma vantagem pois o encurtou o leque de soluções, o que para o curto espaço de tempo de execução, reduziu o tempo de decisão e facilitou a escolha de alternativas, havendo mais entrega no estudo e desenvolvimento.

Estas limitações, as alterações de mercado, foco e qualidade na entrega para a faturação, trouxe uma visão diferente daquela que normalmente é exigida, impondo um sentido de responsabilidade extra que nos faz crescer.

6.3 Trabalhos Futuros

Um produto como pretende ser o Data Explorer diferencia-se de um projeto na medida em que ele deve conseguir servir vários clientes com a mesma versão de

software. Já um projeto normalmente ocorre num espaço de tempo com um determinado âmbito que não tem necessariamente requisitos transversais de um produto.

Com este propósito de produto, algumas funcionalidades poderão ser desenvolvidas no futuro, tal como já foi comunicado com o Gestor de Projeto. Nesse sentido alguns componentes desenvolvidos já foram pensados para deixar espaço para elas.

6.3.1 Refrescamento em Background

Uma primeira funcionalidade seria a de os relatórios serem refrescados automaticamente, não durante a utilização, mas quando são abertos no Excel. A necessidade vem dos utilizadores que consultam alguns relatórios semanalmente ou mensalmente e o refrescamento dos *pivot tables* ligados aos cubos podem demorar desde uns segundos até vários minutos, dependendo do histórico de dados que está a ser consultado. Para resolver este problema sugere-se que haja algum processo agendado no motor *batch* que refresque os relatórios periodicamente. Este agendamento poderia ser definido pelo utilizador e até teria a possibilidade de ser notificado quando o trabalho tivesse concluído.

Outra grande aplicabilidade seria na geração dos relatórios diários, mensais ou até semestrais que quase todas as empresas têm que podiam ser atualizados, exportados para algum formato (*pdf*, imagem ou *csv*) e enviado por algum canal (*email*, *ftp*, *onedrive* ou *dropbox*). Assim aqueles relatórios nunca deixariam de ser entregues por esquecimento ou por férias de um funcionário.

6.3.2 Identificar relatório Data Explorer

Ainda é muito difícil identificar que um relatório tenha sido gerado no *plugin* do Data Explorer. É possível consultar alguns *meta* dados e daí tirar um grau de probabilidade, mas ter 100% a certeza que é um relatório do Data Explorer ainda não é possível. Daí os relatórios estarem a ser guardados estritamente no SEEPlus, não só pela partilha com outros utilizadores, mas assim ao descarregar um relatório do Excel o *plugin* tem a certeza que é um relatório do Data Explorer e que pode refrescar as conexões e os dados dos cubos.

No futuro naturalmente que seria importante o utilizador guardar no próprio computador sem a necessidade de partilhar com o sistema.

6.3.3 Trabalhar em *offline*

Como sugere no título trabalhar em *offline* será o utilizador editar relatório sem ter acesso à internet. Não teria acesso a novos dados sem ligação à internet, mas poderia formatar e carregar assim que haja uma ligação ativa.

Neste momento existe um *workaround* para este problema, bastando apagar o relatório antigo e carregar o formatado com o mesmo nome. Assim outros utilizadores não darão pela falta do relatório que foi carregado com o nome do antigo.

Com a funcionalidade implementada do capítulo anterior (6.3.2), seria fácil fazer a correspondência entre o relatório *offline* e o *online* e sugerir ao utilizador se deseja sobrepor o relatório *offline* com o *online* existente no sistema.

6.3.4 Relatórios com várias Folhas de Cálculo

Neste momento um relatório para o Data Explorer apenas corresponde a uma folha de cálculo. Na necessidade de um relatório com várias folhas de dados terá de ser guardado e carregado em separado. Mas claro que isto trás dependências que outros utilizadores podem não ter conhecimento e na utilização gerará problemas.

6.3.5 Integração com plataformas documentais

A integração com plataformas documentais poderá ser um requisito para clientes que tenham necessidade para certificação de qualidade de processos ou restrições de confidencialidade. Nesta situação os relatórios não seriam guardados no repositório do SEEPlus mas numa plataforma empresarial como o SharePoint, Moodle ou outro *Content Management*.

Isto poderia abrir outra a possibilidade de criar um produto novo completamente desconexo do SEEPlus. Os cubos seriam substituídos por outro DW externo, o *login* seria um serviço LDAP ou *Active Directory* e o repositório o SharePoint. Ficaria sempre a dependência de uma pequena base de dados para alguns dados de arranque do *plugin*. No entanto esta possibilidade será sempre uma questão comercial da Inovretail.

6.3.6 Secções de relatório

Em muitas ferramentas de *reporting* é possível definir secções como: *header*, *body*, *detail* e *footer*. O Data Explorer ainda está limitado ao tipo de *template* do *header*. Uma boa forma de criar relatórios mais dinâmicos será a expansão dos tipos de *templates* para *body* e *footer*. Assim uma organização que tenha um *header* e *footer* com

algumas informações organizacionais, tais como logotipo ou contactos, terá de alterar meia dúzia de *templates* a invés de todos os relatórios já existentes.

Referências Bibliográficas

- Bishnoi, V. (2012). An introduction to Hyperion EssBase. *Journal of Global Research in Computer Science*.
- Bishnoi, V. (2012). Implementation of EssBase Application using Oracle Hyperion. *Journal of Global Research in Computer Science*.
- Breslin, M. (2004). Data Warehousing Battle of the Giants. *Comparing the Basics of the Kimball and Inmon Models*. Obtido em 01 de 2015, de <http://www.olap.it/Articoli/Battle%20of%20the%20giants%20-%20comparing%20Kimball%20and%20Inmon.pdf>
- Connolly, T., & Begg, C. (2001). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Wesley.
- Denny Lee, N. D. (06 de 2007). *SQL Server Best Practices Article*. Obtido de SQL Server: <https://technet.microsoft.com/en-us/library/cc966449.aspx>
- Fowler, M. (07 de 2004). *Presentation Model*. Obtido em 12 de 2014, de <http://martinfowler.com/eaaDev/PresentationModel.html>
- Fowler, M. (09 de 2008). *MSDN Dot Net Blogs*. Obtido de Using the Model-View-Presenter (MVP) Design Pattern to enable Presentational Interoperability and Increased Testability: <http://blogs.msdn.com/b/jowardel/archive/2008/09/09/using-the-model-view-presenter-mvp-design-pattern-to-enable-presentational-interoperability-and-increased-testability.aspx>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns*. Obtido de Elements of Reusable Object Oriented Software: <http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>
- Golfarelli, & Rizzi. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill Osborne Media.
- Gossman, J. (4 de 3 de 2006). *Advantages and disadvantages of M-V-VM*. Obtido em 1 de 2015, de <http://blogs.msdn.com/b/johngossman/archive/2006/03/04/543695.aspx>
- Granlund, Å., Lafrenière, D., & Carr, D. (2001). A Pattern-Supported Approach to the User Interface Design Process. Obtido de <http://www.sm.luth.se/csee/csn/publications/HCIInt2001Final.pdf>

- Harinath, S., Pihlgren, R., & Guang-Yeu Lee, D. (2010). *Professional Microsoft PowerPivot para Excel e SharePoint*. Birmingham, Reino Unido: Wrox.
- Ikujiro Nonaka, H. T. (2003). *Criação de Conhecimento Na Empresa*. Obtido de http://books.google.pt/books/about/Cria%C3%A7ao_de_Conhecimento_Na_Empresa.html?id=FN_LCwX0s-oC&redir_esc=y
- Inmon, W. (1991). *Building the Data Warehouse*. WILEY.
- Inovretail. (2014). *Inovretail*. Obtido em 01 de 2015, de <http://www.inovretail.com/>
- Jelen, B., & Alexander, M. (2011). *Pivot Table Data Crunching: Microsoft Excel 2010*. Pearson Education, Inc. Obtido de <https://books.google.pt/books?id=xeh0OR-xEWQC>
- Kimball, R. (1998). *The Data Warehouse Lifecycle Toolkit*. Wiley.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley. Obtido em 01 de 2015, de <http://it-ebooks.info/book/2637/>
- Lee, D., & Unkroth, K. (06 de 2010). Scale-Out Querying for Analysis Services with Read-Only Databases. Obtido em 05 de 2015, de [https://technet.microsoft.com/en-us/library/ff795582\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/ff795582(v=sql.100).aspx)
- Luhn, H. (1958). A Business Intelligence System. *IBM Journal*. Obtido em 02 de 2015, de <http://domino.research.ibm.com/tchjr/journalindex.nsf/%209fe6a820aae67ad785256547004d8af0/fc097c29158e395f85256bfa00683d4c!OpenDocument>
- Microsoft. (2009). SSAS. Obtido de Designing Operational Scripts for a Multidimensional Database Solution: [https://msdn.microsoft.com/en-us/library/ms174502\(v=sql.105\).aspx](https://msdn.microsoft.com/en-us/library/ms174502(v=sql.105).aspx)
- Microsoft. (2014). *Cloud Services*. Obtido em 01 de 2015, de Microsoft Azure: <http://azure.microsoft.com/en-us/services/cloud-services/>
- Microsoft. (2014). Introduction to Microsoft Power Query for Excel. Obtido de <https://support.office.com/en-us/article/Introduction-to-Microsoft-Power-Query-for-Excel-6e92e2f4-2079-4e1f-bad5-89f6269cd605?CorrelationId=15347639-cef9-4331-9cdc-9449c89098ce&ui=en-US&rs=en-US&ad=US>
- Microsoft. (01 de 2015). *Azure SQL Database*. Obtido em 02 de 2015, de Microsoft Azure: <http://azure.microsoft.com/en-us/marketplace/application-services/>
- Microsoft. (2015). *Connect to Analysis Services*. Obtido de Connect to Analysis Services: [https://msdn.microsoft.com/en-us/library/ms174926\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms174926(v=sql.110).aspx)

- Microsoft. (01 de 2015). *Management Portal for SQL Database*. Obtido em 02 de 2015, de Microsoft Azure: <https://msdn.microsoft.com/en-us/library/azure/dn771027.aspx>
- Microsoft. (2015). *O que é o Microsoft Azure?* Obtido de Microsoft Azure: <http://azure.microsoft.com/pt-pt/overview/what-is-azure/>
- Microsoft. (2015). *Virtual Machines Pricing*. Obtido de Microsoft Azure: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/>
- Moise, W., Conlon, T., & Thompson, M. (1999). *Patente N° US6626959 B1*. Obtido de <https://www.google.com/patents/US6626959>
- Moody, D., & Kortink, M. (2000). From Enterprise Models to Dimensional Models. *A Methodology for Data Warehouse and Data Mart Design*, p. 12. Obtido em 01 de 2015, de <http://ceur-ws.org/Vol-28/paper5.pdf>
- Oracle. (2007). *SpreadSheet Add-in User's Guide for Excel*. Obtido em 01 de 2015, de http://docs.oracle.com/cd/E10530_01/doc/epm.931/essexcel.pdf
- Oracle. (2014). Oracle OLAP.
- Pivotal. (2015). *RabbitMQ*. Obtido de RabbitMQ: <https://www.rabbitmq.com/>
- Porter, M. E. (1999). *Estratégias competitivas essenciais*. CAMPUS. Obtido de <https://www.google.pt/books?id=SMfDDZCuCIEC&lpg=PA7&ots=SF1aRZUZRs&dq=porter%20Estrat%C3%A9gia%20empresarial%3A%20tend%C3%Aancias%20e%20desafios%201999&lr&pg=PP1#v=onepage&q&f=false>
- Power, D. J. (03 de 2007). *A Brief History of Decision Support Systems*. Obtido em 02 de 2015, de DSS Resources: <http://dssresources.com/history/dsshistory.html>
- Raisinghani, M. (2003). *Business Intelligence in the Digital Economy: Opportunities, Limitations and Risks*. Idea Group Inc.
- Ralston, B. (2011). *Getting Started with PowerPivot for Excel*. Apress. doi:10.1007/978-1-4302-3381-7_1
- soundarmoorthy. (27 de 01 de 2015). *WiX Toolset Downloads*. Obtido de WiX Toolset: <https://wix.codeplex.com/releases>
- Tejada, Z. (20 de 03 de 2012). *Dev Pro Connections*. Obtido em 01 de 2015, de Windows Azure Development: An Architectural Overview: <http://devproconnections.com/windows-azure-development/windows-azure-development-architectural-overview>
- Vaishnavi, V., & Kuechler, B. (2004). *Design Science Research in Information Systems*.

- Vesna Luzar-Stiffler, V. H. (2008). *Proceedings of the ITI 2008 30th International Conference on INFORMATION TECHNOLOGY INTERFACES*. Obtido de <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4588364>
- Visão. (2012). *Economia Investimento*. Obtido de http://www.pofc.qren.pt/ResourcesUser/2011_Documentos/Clipping/20110721_BA_Visao.pdf
- Webster, J., & Watson, R. (2002). Analyzing the Past to Prepare the Future: Writing a Literature Review. Obtido em 12 de 2015, de <https://people.creighton.edu/~lch50201/summer2004/article1.pdf>
- Wikipédia. (2014). *Model View ViewModel*. Obtido em 12 de 2014, de http://en.wikipedia.org/wiki/Model_View_ViewModel
- Win-Pro Consultancy, L. (09 de 2012). *Benefits of Cloud Computing*. Obtido em 02 de 2014, de IT Support Singapore Company for Outsourcing IT Services: <http://www.it-support-singapore.com/it-services/cloud-virtualization-microsoft-hyper-v-vmware-vsphere-citrix-xenserver/benefits-of-cloud-computing/>
- WiX. (21 de 03 de 2015). *WiX Toolset*. Obtido de WiX Toolset: <http://wixtoolset.org/>
- WiX. (20 de 02 de 2015). *WiX Toolset*. Obtido de WiX Toolset Manual Table of Contents: <http://wixtoolset.org/documentation/manual/v3/>

Anexo A Script WiX

Exemplo do código criado para a página de Welcome na instalação.

```
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">
  <?include $(sys.CURRENTDIR)Includes\SetupVariables.wxi ?>
  <Fragment>
    <UI>
      <Dialog Id="WelcomeDlgCustom" Width="370" Height="270"
Title="!(loc.WelcomeDlg_Title)">
        <Control Id="Next" Type="PushButton" X="236" Y="243" Width="56"
Height="17" Default="yes" Text="!(loc.WixUINext)" >
          <Publish Property="WixUI_InstallMode" Value="Update">Installed AND
PATCH</Publish>
        </Control>
        <Control Id="Cancel" Type="PushButton" X="304" Y="243" Width="56"
Height="17" Cancel="yes" Text="!(loc.WixUICancel)">
          <Publish Event="SpawnDialog" Value="CancelDlg">1</Publish>
        </Control>
        <Control Id="Bitmap" Type="Bitmap" X="0" Y="0" Width="370" Height="234"
TabSkip="no" Text="!(loc.WelcomeDlgBitmap)" />
        <Control Id="Back" Type="PushButton" X="180" Y="243" Width="56"
Height="17" Disabled="yes" Text="!(loc.WixUIBack)" />
        <Control Id="BottomLine" Type="Line" X="0" Y="234" Width="375" Height="0"
/>
        <Control Id="Description" Type="Text" X="15" Y="105" Width="340"
Height="60" Transparent="yes" NoPrefix="yes" Text="!(loc.WelcomeDlgDescription)"
>
          <Condition Action="show">NOT Installed OR NOT PATCH</Condition>
          <Condition Action="hide">Installed AND PATCH</Condition>
        </Control>
        <Control Id="PatchDescription" Type="Text" X="15" Y="95" Width="340"
Height="60" Transparent="yes" NoPrefix="yes"
Text="!(loc.WelcomeUpdateDlgDescriptionUpdate)" >
          <Condition Action="show">Installed AND PATCH</Condition>
          <Condition Action="hide">NOT Installed OR NOT PATCH</Condition>
        </Control>
        <Control Id="Title" Type="Text" X="15" Y="75" Width="340" Height="40"
Transparent="yes" NoPrefix="yes" Text="!(loc.WelcomeDlgTitle)" />
        <Control Id="Version" Type="Text" X="15" Y="165" Width="340" Height="40"
Transparent="yes" NoPrefix="yes"
Text="!(loc.WelcomeDlgVersion)$(var.ProductVersion)" />
      </Dialog>

      <InstallUISequence>
        <Show Dialog="WelcomeDlgCustom" Before="ProgressDlg"
Overridable="yes">NOT Installed OR PATCH</Show>
      </InstallUISequence>
    </UI>
  </Fragment>
</Wix>
```

Anexo B Plano de Atividades

Neste capítulo será apresentada a lista de trabalhos executados durante a dissertação.

A calendarização da Figura 63 engloba o planeamento das tarefas realizadas na tese de dissertação, incluindo a data de início e fim e duração. O planeamento da dissertação sofreu alterações consequência de imprevistos durante o desenvolvimento. A lista de riscos da Tabela 1 incorpora alguns desses imprevistos que podem ser analisados no capítulo 2.4, juntamente com a Tabela 2 com algumas das ocorrências.

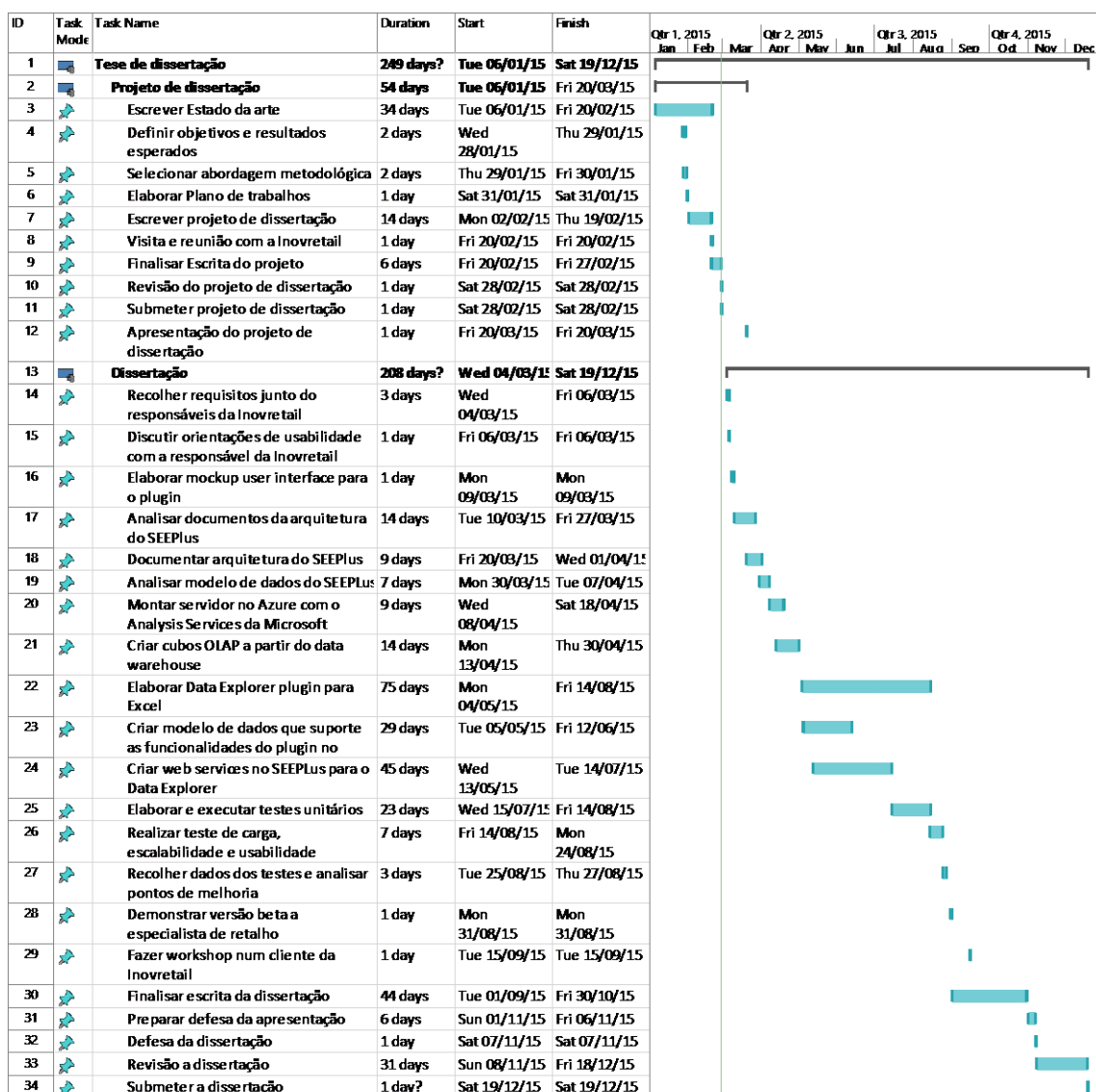


Figura 63 - Calendarização da Tese de dissertação

No decorrer das atividades planeadas com as contantes mudanças do ambiente externo ao projeto, o mesmo sofreu algumas alterações. Na Figura 64 podemos ver a tarefas

executadas onde as tarefas com diferenças face ao plano estão sinalizadas com um asterisco (*).



Figura 64 - Calendarização das tarefas executadas na Tese Dissertação