

Dados de Reconhecimento Facial *em Business Intelligence*

Rui Manuel do Carmo Pinho Fumega Fontinha

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Dr. Paulo Oliveira

Júri:

Presidente:

Doutor José António Reis Tavares

Vogais:

Doutora Maria de Fátima Coutinho Rodrigues

Doutor Paulo Jorge Machado Oliveira

Resumo

Vivemos cada vez mais numa era de crescentes avanços tecnológicos em diversas áreas. O que há uns anos atrás era considerado como praticamente impossível, em muitos dos casos, já se tornou realidade. Todos usamos tecnologias como, por exemplo, a Internet, *Smartphones* e GPSs de uma forma natural. Esta proliferação da tecnologia permitiu tanto ao cidadão comum como a organizações a sua utilização de uma forma cada vez mais criativa e simples de utilizar. Além disso, a cada dia que passa surgem novos negócios e *startups*, o que demonstra o dinamismo que este crescimento veio trazer para a indústria.

A presente dissertação incide sobre duas áreas em forte crescimento: Reconhecimento Facial e *Business Intelligence* (BI), assim como a respetiva combinação das duas com o objetivo de ser criado um novo módulo para um produto já existente.

Tratando-se de duas áreas distintas, é primeiramente feito um estudo sobre cada uma delas. A área de *Business Intelligence* é vocacionada para organizações e trata da recolha de informação sobre o negócio de determinada empresa, seguindo-se de uma posterior análise. A grande finalidade da área de *Business Intelligence* é servir como forma de apoio ao processo de tomada de decisão por parte dos analistas e gestores destas organizações.

O Reconhecimento Facial, por sua vez, encontra-se mais presente na sociedade. Tendo surgido no passado através da ficção científica, cada vez mais empresas implementam esta tecnologia que tem evoluído ao longo dos anos, chegando mesmo a ser usada pelo consumidor final, como por exemplo em *Smartphones*. As suas aplicações são, portanto, bastante diversas, desde soluções de segurança até simples entretenimento.

Para estas duas áreas será assim feito um estudo com base numa pesquisa de publicações de autores da respetiva área. Desde os cenários de utilização, até aspetos mais específicos de cada uma destas áreas, será assim transmitido este conhecimento para o leitor, o que permitirá uma maior compreensão por parte deste nos aspetos relativos ao desenvolvimento da solução.

Com o estudo destas duas áreas efetuado, é então feita uma contextualização do problema em relação à área de atuação da empresa e quais as abordagens possíveis. É também descrito todo o processo de análise e conceção, assim como o próprio desenvolvimento numa vertente mais técnica da solução implementada.

Por fim, são apresentados alguns exemplos de resultados obtidos já após a implementação da solução.

Palavras-chave: *Business Intelligence*, Reconhecimento Facial, Armazém de Dados

Abstract

We live in an era of increasing technological advancements in several areas. What was considered as virtually impossible a few years ago has, in many cases, become a reality. We all use technologies such as the Internet, the Smartphone and GPS in a natural way. The ease with which these technologies can be used has allowed both the average person and large organizations to utilize technology in an increasingly more creative and simple way. Moreover, with each passing day new businesses and start-ups are created, demonstrating the dynamism that this growth has brought to the industry.

This thesis focuses on two areas with robust growth: Face Recognition and Business Intelligence (BI), and the combining of the two with the objective of creating a new software module for an already existing product.

Given that these are two distinct areas, a study of each is first conducted. The area of Business Intelligence is dedicated to organizations and focuses on the gathering of business-related information for a company, with subsequent analysis. The ultimate objective of the area of Business Intelligence is to support the decision-making process by an organization's analysts and managers.

Differently, Face Recognition is more present in society. Having stemmed from science fiction, Face Recognition is now commonly used by a vast number of companies. This technology, however, has evolved so much throughout the years that today it is used by the final consumer, an example of which are smartphones. Therefore, the applications of Face Recognition are varied, from security solutions to simple entertainment.

For both Business Intelligence and Face Recognition a study based on the research of authors in each area will be conducted. The information will range from each technology's fundamental use to specific aspects of each area. This knowledge will be passed on to the reader, thus allowing for a more comprehensive understanding of aspects related to the development of the software solution.

After the study of these two areas, a contextualization of the problem in relation to the company's field of action will be conducted, as well as possible approaches to the problem. The entire process of conception and analysis will also be detailed, in addition to the development of the implemented solution in a more technical manner.

Lastly, several examples of achieved results after the implementation of the software solution are presented.

Keywords: Business Intelligence, Face Recognition, Data Warehouse

Agradecimentos

Pretendo expressar os meus agradecimentos a todos os que contribuíram para a realização desta dissertação.

Começo por agradecer a todos os colegas da InovRetail que se encontraram sempre disponíveis para ajudar, especialmente ao Hugo Lopes, José Araújo e Liane Ferraz.

Obrigado também ao meu orientador, Dr. Paulo Oliveira, cuja ajuda foi determinante na entrega da dissertação.

A todos os meus amigos que proporcionaram momentos de distração nas alturas em que mais precisava.

Quero também agradecer a toda a minha família, especialmente aos meus pais, que me apoiaram nos bons e maus momentos ao longo do desenvolvimento da tese, assim como do próprio Mestrado em si.

Last but not least, obrigado também ao meu primo Pedro e à Carrie que, mesmo a milhares de quilómetros de distância, me conseguiram ajudar.

Índice

1	Introdução	1
1.1	InovRetail	1
1.2	Motivação	3
1.3	Áreas de estudo	3
1.4	Estrutura do documento	4
2	Estado da Arte	7
2.1	Business Intelligence	7
2.1.1	Armazém de Dados	8
2.1.2	Modelo Dimensional	12
2.1.3	Carregamento no Armazém de Dados	15
2.1.4	Desempenho	17
2.2	People Counters	20
2.2.1	LASE PeCo	23
2.2.2	Experian FootFall	26
2.3	Reconhecimento Facial	29
2.3.1	Áreas de utilização	30
2.3.2	Algoritmos de reconhecimento	36
2.3.3	Produtos Existentes	38
3	Análise	45
3.1	Objetivos	45
3.2	Abordagens	46
3.2.1	Abordagem 1	46
3.2.2	Abordagem 2	46
3.3	Desenho da Solução	48
3.3.1	Arquitetura	48
3.3.2	Componente Lojas	48
3.3.3	Componente Armazém de Dados	53
3.3.4	Componente Utilizador	55
4	Desenvolvimento	57
4.1	Componente Loja	57
4.1.1	Intel AIM	58
4.1.2	Processos de Carregamento na BD Local	58
4.1.3	Processo de Carregamento no Armazém de Dados	63
4.2	Componente Armazém de Dados	67
4.2.1	Fluxo	67
4.2.2	Criação de Cubo de Dados	73
4.3	Componente Utilizador	87

5	Resultados obtidos	89
5.1	Distribuição por Sexo	89
5.2	Distribuição por Faixa Etária	92
5.3	Vendas.....	94
5.4	Impacto dos fatores demográficos nas vendas	96
6	Conclusão	99
6.1	Melhorias / Trabalho Futuro.....	100

Lista de Figuras

Figura 1 – Fatores relacionados com a experiência do cliente em loja	2
Figura 2 – Arquitetura de um <i>Armazém de Dados</i>	9
Figura 3 – Topologia em Estrela	14
Figura 4 – Topologia em Floco de Neve	14
Figura 5 - Ferramenta ETL (SQL Server Integration Services)	17
Figura 6 – OLTP vs OLAP, frequência de acesso.....	18
Figura 7 – Tipos de particionamento	19
Figura 8 – Taxas de conversão por área de retalho	22
Figura 9 – Imagem Heat Map numa loja.....	24
Figura 10 – PeCo SC / SC High	25
Figura 11 – PeCo TC.....	25
Figura 12 – PeCo TC - Área de deteção	26
Figura 13 – Exemplo FootFall Indices – Mapa.....	27
Figura 14 – Exemplo FootFall Indices - Semanas	28
Figura 15 – Exemplo de aplicação de <i>Reconhecimento Facial</i>	29
Figura 16 – Sistema CCTV (dispositivos de captação de imagens).....	30
Figura 17 – Sistema CCTV (central de controlo).....	31
Figura 18 – Sistema biométrico de reconhecimento facial.....	32
Figura 19 – Exemplo de reconhecimento facial aplicada à fotografia	33
Figura 20 – Reconhecimento Facial no Facebook.....	34
Figura 21 – Samsung Galaxy S4 Smart Pause.....	34
Figura 22 – Kinect 2 - Autenticação	35
Figura 23 – Skype com Kinect – <i>Skeleton Tracking</i>	36
Figura 24 – Utilização do algoritmo “Haar-based Cascade Classifier”	37
Figura 25 – InsideVisions Faceclick	41
Figura 26 – InsideVisions Targetads.....	42
Figura 27 – EXVA Adnamic	42
Figura 28 – Diagrama de arquitetura.....	48
Figura 29 – Modelo de dados da Base de Dados Local	50
Figura 30 – Conteúdo da tabela na Base de Dados Local, D_GENDER	51
Figura 31 - Conteúdo da tabela na Base de Dados Local, D_AGE_GROUP	51
Figura 32 - Conteúdo da tabela na Base de Dados Local, STAG_ISP_READS	52
Figura 33 – Modelo de dados do Armazém de Dados	55
Figura 34 – Parametrização dos processos no <i>Windows Task Scheduler</i>	59
Figura 35 – Tarefa responsável pelo <i>Reboot</i> da máquina.....	59
Figura 36 – Configuração das variáveis de ambiente DEVICE_ID e STORE_ID.....	62
Figura 37 – Referência <i>Web Service</i>	65
Figura 38 – Processo de carregamento de dados no Armazém de Dados.....	68
Figura 39 – <i>Web Services</i> disponibilizados pelo SeePlus	69
Figura 40 – Tabela DW - STAG_SMARTSCAN_READS	73

Figura 41 – Diagrama dados do cubo multidimensional.....	74
Figura 42 – Resultado <i>view</i> vw_dim_calendar.....	75
Figura 43 – Hierarquias da dimensão “Data”.....	76
Figura 44 – Navegação dimensão “Data”, hierarquia “Date Hierarchy”.....	76
Figura 45 – Resultado <i>view</i> vw_dim_time.....	77
Figura 46 – Navegação dimensão “Tempo”.....	77
Figura 47 – Resultado <i>view</i> vw_dim_age.....	78
Figura 48 – Navegação dimensão “Faixa Etária”.....	78
Figura 49 – Resultado <i>view</i> vw_dim_gender.....	79
Figura 50 – Navegação dimensão “Sexo”.....	79
Figura 51 – Resultado <i>view</i> vw_dim_location.....	80
Figura 52 - Hierarquias da dimensão “Loja”.....	80
Figura 53 – Navegação dimensão “Loja”.....	81
Figura 54 – <i>Measure Groups</i> criados.....	81
Figura 55 – Resultado <i>view</i> vw_sls_tkt.....	82
Figura 56 – <i>Measure Group</i> vw_sls_tkt.....	83
Figura 57 – Resultado <i>view</i> vw_vst_demographics.....	84
Figura 58 – Associação <i>Measure Group</i> a Dimensões.....	85
Figura 59 – Processamento do cubo.....	86
Figura 60 – Estabelecimento ligação ao cubo através do Excel.....	87
Figura 61 – Seleção do cubo.....	88
Figura 62 – Pivot Table com base no cubo.....	88
Figura 63 - Visitantes por Sexo na Loja 1.....	90
Figura 64 - Visitantes por Sexo na Loja 2.....	90
Figura 65 - Visitantes por Sexo e Hora – Loja 1.....	91
Figura 66 - Visitantes por Sexo e Hora – Loja 2.....	91
Figura 67 - Visitantes por faixa etária – Loja 1.....	92
Figura 68 - Visitantes por faixa etária – Loja 2.....	92
Figura 69 – <i>Drill-Down</i> dimensão Data.....	93
Figura 70 - Visitantes por faixa etária e dia da semana – Loja 1.....	93
Figura 71 - Visitantes por faixa etária e dia da semana – Loja 2.....	94
Figura 72 - Venda líquida na Semana 34.....	95
Figura 73 - Taxa de Conversão Loja 1 na Semana 34.....	95
Figura 74 - <i>Dashboard</i> em Excel (Semana 36).....	96
Figura 75 – <i>Dashboard</i> em Excel (Semana 35).....	97

Lista de Tabelas

Tabela 1 - Diferenças entre metodologias	12
Tabela 2 – Exemplos de factos por domínio	12
Tabela 3 - Exemplo tabela diária	20
Tabela 4 - Exemplo tabela agregada mensalmente	20
Tabela 5 - Vantagens/Desvantagens reconhecimento facial como identificação biométrica...	32
Tabela 6 – Atributos retornados pela API do Intel AIM	40
Tabela 7 – Descrição das métricas do cubo	83

Lista de Excertos de Código

Código 1 – Pedido efetuado ao <i>Socket</i> do Intel AIM	60
Código 2 – Resposta recebida do <i>Socket</i> do Intel AIM	61
Código 3 – Inserção na Base de Dados Local dos dados provenientes do Intel AIM.....	61
Código 4 – Processo de purga da tabela central na Base de Dados local.....	63
Código 5 – Criação de string XML.....	65
Código 6 – XML resultado	65
Código 7 – Envio da <i>string</i> XML para o <i>Web Service</i>	66
Código 8 – Colocação valor “C” no INTEGRATION_STATUS.....	66
Código 9 – Declaração <i>Web Service</i>	69
Código 10 – Definição <i>Web Service</i>	70
Código 11 – <i>Web Service</i> , garantia de sucesso na execução (componente lojas).....	71
Código 12 – Procedimento responsável por inserir dados no Armazém de Dados.....	72
Código 13 – View vw_dim_calendar	75
Código 14 – View vw_dim_time.....	77
Código 15 – View vw_dim_age	78
Código 16 – View vw_dim_gender.....	79
Código 17 – View vw_dim_location.....	79
Código 18 – View vw_sls_tkt.....	82
Código 19 – View vw_vst_demographics.....	84

Glossário

Ad-hoc	Necessidade específica de determinado momento.
Agile Modeling	Metodologia de implementação de Software.
AIM	Intel AIM (Audience Impression Metrics).
Armazém de Dados (AD)	Diz respeito ao armazenamento da informação relativa a todas as áreas de negócio.
Atmospherics	Fatores ambientais de uma loja como temperatura e volume da música.
Azure	Tecnologia de Cloud Computing da Microsoft.
BD	Base de Dados.
Business Intelligence	(BI) Conjunto de processos e ferramentas cuja finalidade é ajudar o utilizador a tomar decisões relativas ao seu negócio.
C#	Linguagem de programação orientada a objetos
CCTV	Closed-Circuit Television (Circuito fechado de televisão).
CIF	Corporate Information Factory. Arquitetura pertencente à metodologia de desenvolvimento de armazéns de dados da autoria de Inmon.
Cloud	Cloud Computing. Disponibilização de aplicações de hardware e software na Internet.
Constellation	Esquema de modelação de dados usado em BI.
CRUD	Create Read Update Delete. Operações base em tabelas de uma BD.
Cubo	Estrutura de dados que contém parte da informação do Armazém de Dados de forma a tornar as consultas rápidas e intuitivas.
Dashboard	É um “painel” que tem o objetivo de mostrar a informação do desempenho de uma organização de uma maneira direta e apelativa.
Data Mart	Consiste numa parte da informação contida num Armazém de Dados, relativa a determinada área do negócio.

DirectShow	API do DirectX proprietária da Microsoft, utilizada em aplicações multimédia.
Drill-down	Capacidade de expandir uma estrutura de dados.
Estrela	Esquema de modelação de dados utilizado em BI.
ETL	Extract, Transform, Load; é um processo que permite a extração de dados de uma origem para um destino, utilizando-se normalmente bases de dados.
FireWire	Interface de ligação de dispositivos de áudio e vídeo.
Floco de Neve	Esquema de modelação de dados usado em BI.
Hardware	Componente física de uma solução informática.
Heat Mapping (retalho)	Representação gráfica, no <i>layout</i> de uma loja, da frequência de passagem dos seus visitantes.
HTTP	Hypertext Transfer Protocol. Protocolo de apresentação na camada de aplicação do modelo OSI.
InovRetail	Empresa onde foi desenvolvida a presente dissertação.
IP	Internet Protocol. Protocolo da camada de rede do modelo OSI.
ISEP	Instituto Superior de Engenharia do Porto.
Laser Tracking	Rastreamento com base na tecnologia laser.
Measure Group	Conjunto de métricas, normalmente de uma determinada área de negócio.
MEI	Mestrado em Engenharia Informática.
ODBC	API de acesso a Bases de Dados-
OLE-DB	É uma API desenvolvida pela Microsoft que permite o acesso a bases de dados.
Open Source	Software com código aberto.
PC	Personal Computer.
Pixel	Ponto de uma imagem, com determinadas coordenadas e cor.
People Counter	Sistema que permite efetuar a contagem de visitantes de uma loja.
Query	É uma forma de interrogar uma base de dados, sendo a mais

	utilizada a linguagem SQL.
RAID	Redundant Array of Independent Disks. Técnica de particionamento de discos com vista a melhorar o desempenho e a tolerância a falhas.
Roll-up	Capacidade de colapsar uma estrutura de dados.
SeeMobile	Produto proprietário da InovRetail, acessível através de Smartphones e Tablets.
SeePlus	Produto proprietário da InovRetail.
Skeleton Tracking	Rastreamento do corpo efetuado pelo Kinect 2.
Smartphones	Telemóvel com funcionalidades avançadas, com a possibilidade de serem estendidas através da instalação de aplicações.
Socket	Interface de disponibilização de informação que circula pela rede.
Software	Parte não física de uma solução informática, composta pelo conjunto de instruções que dão origem a programas.
SQL	Linguagem de interrogação de bases de dados bastante utilizada.
SQL Server	Sistema de gestão de bases de dados proprietário da Microsoft.
SSAS	SQL Server Analysis Services. Ferramenta de exploração de informação numa Base de Dados.
SSD	Solid State Drive. Discos de armazenamento com velocidades superiores aos convencionais, devido ao facto de não possuírem componentes mecânicos.
Staging Area	Local de armazenamento temporário num Armazém de Dados, onde se efetua transformações aos dados a carregar.
Surrogate Keys	Identificadores únicos de cada registo de uma tabela de dimensão.
Tablets	Dispositivos que possuem tanto algumas funcionalidades de computadores pessoais como de Smartphones.
USB	Universal Serial Bus. Interface de ligação de dispositivos.
Web-Based	Sistemas baseados na Internet.
Web Browser	Software utilizado para navegar na Internet.
Web Service	Método de comunicação entre aplicações Web através da rede.

Webcam

Câmara de vídeo de baixo custo, normalmente utilizada para videoconferências através de Internet.

XML

eXtensible Markup Language. Formato de ficheiro que, através da utilização de *tags*, identifica estruturas de informação nele presentes.

1 Introdução

O presente trabalho de dissertação incide sobre duas áreas de estudo distintas, sendo elas Reconhecimento Facial e Business Intelligence. É abordada a sua respetiva integração numa solução final, sendo esta parte integrante do produto da InovRetail, empresa onde foi desenvolvida.

De forma a contextualizar o leitor, segue-se uma descrição da empresa, assim como do produto onde a solução concebida será integrada.

1.1 *InovRetail*

A InovRetail, onde foi desenvolvida a presente dissertação, foi fundada em 2010 e trata-se de uma empresa na área de investigação e desenvolvimento de soluções tecnológicas com foco na área do retalho, cuja missão passa por melhorar a experiência do cliente em loja.

O retalho trata-se de um meio muito exigente do ponto de vista tecnológico, uma vez que lida cada vez mais com imensas quantidades de dados diariamente. Um dos maiores desafios é, por isso, realizar a gestão de todos estes dados, sendo que o objetivo final passa pela sua análise de uma forma simples e estruturada.

A InovRetail disponibiliza assim aos seus clientes uma plataforma de gestão (SeePlus) com especial foco em três grandes áreas:

- Análise da performance de cada loja, através de métricas relacionadas com Vendas, Orçamentos, etc.
- Impacto de fatores relacionados com *atmospherics*¹ na performance de cada loja.
- Monitorização e melhoramento da eficácia do *layout* de cada loja.

¹ Fatores tais como temperatura na loja, volume da música, fragrância, etc.

Tratando-se de uma plataforma de gestão da experiência do cliente em retalho, que permite aos retalhistas obterem informação mais detalhada sobre o ambiente das suas lojas, assim como o impacto que este tem no seu desempenho, o SeePlus permite então ao cliente:

- A medição do impacto que os *atmospherics* têm nos seus visitantes e, conseqüentemente, no desempenho em termos de vendas das lojas;
- Definir e executar várias estratégias relacionadas com o *feedback* dos clientes, utilizando-se para isso o preenchimento de questionários;
- Comparar diferentes formatos de loja, com base no rastreamento dos seus visitantes;
- Criar e partilhar análises de negócio dinâmicas, através de ferramentas de fácil utilização;
- Reduzir consumo energético em loja, derivado de uma maior perceção de onde se concentram os desperdícios;
- Visualizar informação em *Smartphones/Tablets*, através do SeeMobile.

Do ponto de vista do utilizador final, isto resulta numa aplicação *web-based* acessível através do *Web Browser*, sendo esta composta por várias secções respeitantes a diferentes módulos do SeePlus.

O desempenho de uma loja baseia-se naquilo que são os fatores com impacto direto ou indireto sobre a experiência do cliente em loja. São eles os presentes na seguinte imagem:



Figura 1 – Fatores relacionados com a experiência do cliente em loja

Um dos pontos de foco da InovRetail diz respeito, tal como referido anteriormente, aos *atmospherics*. Estes tratam-se de variáveis que, através de sensores colocados na loja, podem então ser medidas, assim como o respetivo impacto que estas têm no seu desempenho. São extremamente importantes, uma vez que influenciam a atração de novos clientes, assim como a sua retenção. Por exemplo:

- Música ou temperatura elevados podem levar ao abandono da loja por parte dos clientes.
- Fragrâncias agradáveis levam a uma retenção dos clientes dentro da loja, o que pode resultar num aumento da taxa de conversão² e, conseqüentemente, do lucro.

1.2 Motivação

Atualmente, um dos fatores em falta diz respeito à disponibilização de dados demográficos dos visitantes de cada loja. Isto torna-se relevante, uma vez que, sem esta informação, não é possível aos negócios que possuem um público-alvo específico analisar de uma forma mensurável se este está a ser atingido com sucesso.

Isto é especialmente relevante na área do retalho especializado, uma vez que cada negócio é muito mais direcionado para um público específico. Além disso, há todo um conjunto de fatores externos e dependentes da região, tais como fatores culturais, económicos, religiosos, etc., que têm também a sua influência.

Um dos pontos há muito presente no *roadmap* da InovRetail era o de adicionar à *framework* do SeePlus a informação demográfica dos visitantes das lojas de retalho. Assim, o objetivo passa por implementar todo um fluxo, desde a aquisição desta informação (utilizando para isso tecnologia de reconhecimento facial adquirida a fornecedores externos) até à disponibilização de ferramentas de exploração de dados.

Desta forma, torna-se possível por parte do negócio criar relatórios que incluam esta informação adicional, inexistente até ao momento.

1.3 Áreas de estudo

De forma a introduzir os temas desenvolvidos no âmbito desta dissertação, é primeiro necessário realizar um estudo sobre as áreas abordadas, sendo elas as seguintes:

- *Business Intelligence*
- Reconhecimento Facial

² Métrica de desempenho de uma loja (abordado no Capítulo 2)

A primeira grande área, *Business Intelligence* (BI), trata-se de algo fundamental em empresas de retalho, onde atua a InovRetail. Podemos definir BI como um conjunto de metodologias cujo objetivo final é proporcionar às organizações um melhor conhecimento do desempenho das várias áreas de negócio ao longo do tempo. Com este conhecimento, torna-se muito mais fácil por parte do negócio tomar decisões, tendo como objetivo o sucesso da organização. A base desta informação concentra-se no que é chamado de Armazém de Dados (AD), sendo este o sistema responsável por armazenar dados com origem em várias fontes, assim como disponibilizá-los aos utilizadores finais.

A outra grande área de estudo é algo que cada vez mais começa a entrar no nosso quotidiano: Reconhecimento Facial. Ao contrário de BI, esta área é facilmente reconhecida, e até mesmo utilizada, por um maior número de pessoas. Isto deve-se ao facto de estar presente em vários produtos que fazem parte do nosso quotidiano, como por exemplo em *Smartphones* e câmaras fotográficas digitais. Reconhecimento Facial diz respeito à tecnologia que permite identificar rostos através de uma simples *webcam*, utilizando para isso algoritmos que analisam a imagem à procura de padrões.

Assim, juntando estas duas áreas e aplicando-as à indústria do retalho, é possível obter-se soluções que nos permitam, por exemplo:

- Verificar quantas pessoas do sexo feminino entram numa loja em determinada semana e comparar com o período homólogo.
- Obter o tempo médio que os clientes gastam a olhar para determinada prateleira de exposição.

1.4 Estrutura do documento

O presente documento é composto por seis capítulos.

O primeiro diz respeito à introdução, onde é feita uma contextualização da empresa onde esta dissertação foi desenvolvida, a sua área de negócio de atuação e a forma como esta será afetada pela solução desenvolvida. É também feita uma breve descrição das áreas de estudo abrangidas por este desenvolvimento.

O segundo capítulo é referente ao Estado da Arte. É aqui feita uma análise detalhada dos pontos principais de cada uma das áreas de estudo referidas anteriormente. Com base em autores específicos de cada área, são aqui detalhados os seus principais componentes, assim como descritas as respetivas áreas de aplicação, sendo dados alguns exemplos. É aqui incluída uma secção (*People Counters*) que, apesar de não se tratar de uma área de estudo propriamente dita tão relevante como as referidas anteriormente, é também importante que o leitor tenha conhecimento da sua existência.

O terceiro capítulo corresponde a uma análise do problema. É descrito o objetivo a cumprir, assim como as possíveis abordagens para o problema. É também feito um trabalho de

especificação da solução a ser desenvolvida, nomeadamente um desenho conceptual da arquitetura a implementar.

O quarto capítulo diz respeito a uma descrição mais detalhada da empresa e do produto, assim como do próprio desenvolvimento em si. Tendo por base o desenho conceptual definido no capítulo três, são descritos todos os passos realizados durante o próprio desenvolvimento, sendo que estes se encontram organizados pelas três secções da arquitetura: Loja, Armazém de Dados e Utilizador.

O quinto capítulo é referente aos resultados obtidos. Uma vez que será o cliente final a usufruir da solução implementada e a elaborar ele mesmo os relatórios que considere necessários, neste capítulo são apresentados alguns exemplos destes mesmos relatórios que se considerem relevantes e mostrem as capacidades da solução desenvolvida.

Por fim, no sexto capítulo é feita uma conclusão sobre o trabalho realizado. São também descritas as dificuldades sentidas durante a realização do trabalho, assim como pontos de melhoria futuros.

2 Estado da Arte

Pretende-se com este capítulo introduzir os conceitos das várias áreas de estudo exploradas nesta dissertação, de forma a ajudar o leitor na sua compreensão.

Este capítulo encontra-se dividido em três secções: *Business Intelligence*, Reconhecimento Facial e *People Counters*. Como referido no capítulo da Introdução, as duas primeiras secções dizem respeito às grandes áreas em estudo, enquanto que a secção de *People Counters*, apesar de não ser tão relevante, ajuda o leitor na compreensão de conceitos relacionados com a contagem de visitantes de uma loja.

2.1 *Business Intelligence*

A área do retalho, onde atua a InovRetail, lida diariamente com grandes quantidades de dados de várias áreas de negócio distintas presentes numa organização, como por exemplo vendas, logística e aprovisionamento. Utilizando os sistemas operacionais existentes, torna-se, por isso, complexa a gestão de todos estes dados, que muitas das vezes está espalhada por vários sistemas distintos e sem ligação entre eles. Na grande maioria das vezes não é também guardado qualquer tipo de histórico, o que por si só inviabiliza análises ao desempenho da organização ao longo do tempo.

Com a recente evolução tecnológica, a área do retalho tem à sua disposição novos sistemas de apoio à decisão. Este é um ponto fundamental no sucesso de uma empresa, pois qualquer má decisão pode ter consequências desastrosas e que podem até, em casos extremos, levar ao seu encerramento. É, por isso, bastante importante a existência de ferramentas que apoiem os colaboradores das organizações responsáveis por tomarem decisões.

Atualmente, já existem várias soluções que permitem efetuar análises destes dados de uma forma simples e, em muitos casos, de uma forma *ad-hoc*. Estas soluções têm por base o

desempenho das várias áreas da empresa ao longo do tempo. Desta forma, existem factos que sustentam as decisões tomadas.

Business Intelligence diz então respeito à utilização de metodologias próprias, utilizando-se como apoio estas ferramentas, com vista a serem tomadas decisões.

A primeira vez que o termo *Business Intelligence* foi utilizado já na era das tecnologias da informação, data de 1958:

“a capacidade de apreender as inter-relações dos factos apresentados de forma a orientar as ações em direção a um objetivo” (Luhn, 1958)

De seguida apresenta-se uma descrição de cada um dos principais conceitos de *Business Intelligence*.

2.1.1 Armazém de Dados

Grande parte das soluções de BI utilizam o conceito de Armazém de Dados (AD) como forma de guardar toda a informação das várias áreas de negócio da empresa. Esta informação poderá estar dispersa por vários sistemas, sendo que centralizando-a num único local, torna-se assim mais fácil visualizá-la através de relatórios, assim como proceder à sua própria gestão.

(Inmon, 2002) define que um Armazém de Dados é um conjunto de dados orientados por assuntos, integrados, variáveis no tempo e não voláteis, cujo objetivo passa por dar suporte nos processos de tomada de decisão.

Por outras palavras, um Armazém de Dados é, na sua essência, um conjunto de ferramentas com base num repositório de dados, cujo objetivo passa por permitir a exploração de informação e ajudar o utilizador final na tomada de decisão (Chaudhuri & Dayal, 1997). Tal como referido anteriormente, os sistemas operacionais, também denominados OLTP (*On-Line Transaction Processing*), não foram projetados para armazenar grandes quantidades de informação relativa a dados históricos nem permitir a sua exploração em tempo real. Os Armazéns de Dados, por sua vez, permitem esta exploração de uma forma mais eficiente.

O objetivo de um AD é, por isso, centralizar a informação dos vários sistemas operacionais num único local, ao longo do tempo, e possibilitar a realização de análises relativas às várias áreas de negócio da companhia, utilizando-se para isso diversas ferramentas de exploração. Esta capacidade de explorar quantidades massivas de dados é denominada OLAP (*On-Line Analytical Processing*).

Existem outras diferenças entre os sistemas operacionais e um Armazém de Dados:

- Devido à forma como a informação está modelada num AD (modelo dimensional), a sua consulta através deste é mais fácil e intuitiva do que no sistema operacional

correspondente, sendo rapidamente compreendida por pessoas que trabalhem na área do negócio e que não tenham conhecimentos técnicos.

- Um Armazém de Dados tem por base um esquema simples que permite facilmente adicionar novos atributos às dimensões e às tabelas de factos.
- Também devido à estrutura da informação e à existência de poucas relações entre tabelas, a consulta num AD é mais eficiente do que nos sistemas operacionais (Moody & Kortink, 2000), mesmo tendo várias vezes a quantidade de informação destes.
- O próprio carregamento da informação no Armazém de Dados com origem em várias fontes proporciona uma maior homogeneidade no seu acesso. Isto permite, por exemplo, que não seja necessário os utilizadores configurarem acessos através de vários programas, a várias fontes, pois passa a haver apenas uma: o Armazém de Dados.

A seguinte imagem representa a arquitetura de um AD, em que se pode visualizar os elementos principais que fazem parte deste, nomeadamente:

- **Sistemas fonte de informação:** Repositórios cuja informação será carregada no Armazém de Dados.
- **ETL (*Extract, Transform, Load*):** Ferramentas de importação de dados destes sistemas para o Armazém de Dados.
- **Armazém de Dados:** Local onde é armazenada a informação propriamente dita.
- **Ferramentas de exploração:** Conjunto das várias ferramentas cujo objetivo é permitir a exploração, por parte do utilizador, da informação contida no AD.

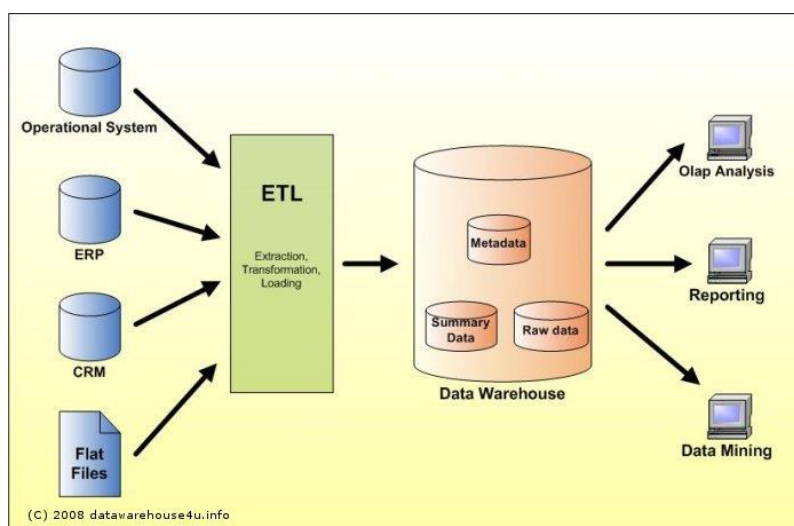


Figura 2 – Arquitetura de um Armazém de Dados³

³ Retirado de (Datawarehouse4u.info, 2009)

2.1.1.1 Metodologias de Kimball e Inmon

Existem duas grandes metodologias de criação de um Armazém de Dados. Porém, antes da descrição de cada uma, é importante definir o conceito de *Data Mart*.

(Moody & Kortink, 2000) afirmam que os *Data Marts* representam os “pontos de venda” de um AD, que por sua vez disponibilizam informação possível de ser analisada por utilizadores finais analistas.

Associado ao conceito de Armazém de Dados, isto significa que um *Data Mart* é um subconjunto de informação do AD, sendo que normalmente diz respeito a apenas uma área de negócio da organização.

Quando estamos a falar de organizações com informação respeitante a várias áreas (por exemplo, vendas, clientes, transferências de stock, etc.), este conceito é extremamente relevante, uma vez que:

- Segmenta os dados fisicamente
- Permite que apenas um determinado conjunto de utilizadores tenha acesso a determinada informação.

Existem, portanto, duas metodologias quanto ao processo de criação de raiz de um AD: a metodologia de Inmon e a metodologia de Kimball, que serão descritas de seguida. Será também descrita, não tão detalhadamente, uma terceira metodologia denominada de *Agile Modeling*, que tem sido cada vez mais utilizada devido ao facto de permitir o desenvolvimento de um Armazém de Dados de uma forma ainda mais ágil do que as restantes.

2.1.1.1.1 Metodologia de Inmon

A metodologia de Inmon, também designada de *Top Down Approach*, tem por base a arquitetura *Corporate Information Factory* (CIF). (Breslin, 2004)

Esta arquitetura define o Armazém de Dados como o seu ponto central, onde toda a informação proveniente dos sistemas operacionais deve convergir. Da mesma maneira, todas as ferramentas de exploração devem ler esta informação diretamente ao AD.

Isto significa que, na área de apoio à decisão de uma organização, o Armazém de Dados é a única fonte de dados “fidedigna” (Inmon, 2010), uma vez que a informação já foi tratada aquando do seu carregamento, e se encontra por isso, uniformizada.

Os outros componentes da *Corporate Information Factory* dizem respeito às ferramentas de ETL (*Extract, Transform, Load*), aos *Data Marts* e às bases de dados operacionais existentes na empresa.

É, portanto, uma metodologia que leva a uma reestruturação da companhia, sendo a seguinte frase uma boa ilustração da sua abrangência:

“Isto significa que, entre outras coisas, um Armazém de Dados e as bases de dados operacionais fazem parte de um todo maior” (Breslin, 2004)

2.1.1.1.2 Metodologia de Kimball

Por sua vez, a metodologia de Kimball é o oposto da de Inmon. Muitas vezes designada por *Bottom Up Approach*, a sua maior diferença diz respeito ao facto de, na de Kimball, o Armazém de Dados ser a união de todos os *Data Marts* que o compõe. (Kimball & Ross, 2002)

Isto significa que se trata de um processo bastante mais ágil, uma vez que vão sendo construídos *Data Marts* à medida que é feita a sua disponibilização para os utilizadores finais.

Uma vez que este modelo de desenvolvimento é ágil, torna-se necessário garantir a coerência de dados, ou seja, todos estes *Data Marts* devem possuir informação em comum. Por outras palavras, os dados provenientes das várias áreas de negócio carregados no AD (em *Data Marts*), devem seguir um formato homogéneo de forma a facilitar a sua exploração. Isto é atingido com a utilização do modelo dimensional (a ser definido na secção seguinte), em que, por exemplo, a estrutura de lojas de uma organização, no que diz respeito aos níveis da sua hierarquia por área geográfica, é exatamente a mesma, independentemente do *Data Mart* a ser analisado.

2.1.1.1.3 Metodologia Agile Modeling

Uma das metodologias que tem sido cada vez mais adotada é a denominada de *Agile Modeling*.

Agile baseia-se no princípio da simplicidade e modelação *light-weight*, de forma a permitir que o foco se centralize no próprio desenvolvimento. (Couceiro, 2012)

Com esta metodologia, o desenvolvimento da solução é iniciado praticamente na mesma altura em que tem lugar o processo de desenho e modelação do Armazém de Dados, traduzindo-se assim em custos iniciais baixos. Além disso, torna-se mais fácil reagir a problemas que possam surgir, nomeadamente a alterações de requisitos, uma vez que o desenvolvimento e as reuniões com o cliente ocorrem normalmente em paralelo.

Principalmente motivado por estes dois fatores, pode-se assim concluir que a sua adoção em empresas *start-up* seja cada vez maior, como é o caso específico da InovRetail., empresa onde foi desenvolvida a presente dissertação.

Analisando estas metodologias, é possível enumerar algumas das diferenças entre cada uma:

Tabela 1 - Diferenças entre metodologias

	<i>Inmon</i>	<i>Kimball</i>	<i>Agile</i>
<i>Custos iniciais</i>	<i>Altos</i>	<i>Baixos</i>	<i>Muito Baixos</i>
<i>Tempo de desenvolvimento</i>	<i>Alto</i>	<i>Baixo</i>	<i>Baixo</i>
<i>Normalização tabelas</i>	<i>3ª Forma normal</i>	<i>Desnormalizadas</i>	<i>Desnormalizadas</i>
<i>Complexidade</i>	<i>Alta</i>	<i>Baixa</i>	<i>Média</i>
<i>Necessidade</i>	<i>Médio/Longo prazo</i>	<i>Curto prazo</i>	<i>Imediata</i>
<i>Risco (na implementação)</i>	<i>Alto</i>	<i>Baixo</i>	<i>Baixo</i>
<i>Envolvimento com o cliente/negócio</i>	<i>Médio</i>	<i>Alto</i>	<i>Muito Alto</i>
<i>Dimensão do projeto</i>	<i>Grande</i>	<i>Pequeno/Médio/Grande</i>	<i>Pequeno/Médio/Grande</i>

2.1.2 Modelo Dimensional

A estrutura da informação de um Armazém de Dados segue as regras do modelo dimensional, ou seja, a separação entre medidas (factos) e “entidades” (dimensões). Além de melhorar o desempenho das análises efetuadas, este modelo tem a grande vantagem de ser de fácil compreensão tanto para utilizadores técnicos, como para utilizadores que façam parte do negócio. (Moody & Kortink, 2000)

Os factos dizem respeito a algo que se pretende medir. Foram capturados em determinado período do tempo e, por isso mesmo, mantêm o seu valor indefinidamente. Seguem-se exemplos de alguns dos factos e domínios mais comuns no sector do retalho:

Tabela 2 – Exemplos de factos por domínio

<i>Facto</i>	<i>Domínio</i>
<i>Valor líquido</i>	<i>Vendas</i>
<i>Quantidade vendida</i>	<i>Vendas</i>
<i>Quantidade stock</i>	<i>Stocks</i>
<i>Valor da compra</i>	<i>Compras</i>
<i>Valor transferido</i>	<i>Transferências</i>

Por sua vez, as dimensões dizem respeito às “entidades” que servem de contexto para estes factos. “Num modelo multidimensional, estes factos apenas são tidos em conta quando vistos num espaço rodeado por dimensões”. (Chaudhuri & Dayal, 1997)

Utilizando como exemplo a Tabela 2, o facto “Valor Líquido” por si só não tem significado. No entanto, se se adicionar algumas dimensões, como por exemplo Loja, Produto e Data, é possível responder ao seguinte:

- Quantos artigos existiam em *stock* na loja 1 no último domingo?
- Quantos artigos da categoria A foram vendidos em 2013?
- Qual o valor líquido das vendas na loja 1, de artigos da Categoria A, no mês de Março de 2014, face ao mês do período homólogo?

Estes são alguns exemplos das muitas análises possíveis apenas com estas 3 dimensões. Em todos eles, como se pode constatar, está presente a dimensão Data. Isto deve-se ao facto de se tratar da dimensão mais importante, pois permite visualizar tendências, assim como fazer previsões. (Chaudhuri & Dayal, 1997)

Juntando-se assim os factos com as dimensões, obtém-se uma ferramenta de exploração de dados poderosa, que os utilizadores finais compreendem facilmente, mesmo que não possuam formação técnica.

A seguinte frase ilustra de uma forma abrangente a utilidade do modelo dimensional:

“A possibilidade de visualizar algo tão abstrato como um conjunto de dados de uma forma concreta e tangível é o segredo da compreensão”. (Kimball & Ross, 2013)

2.1.2.1 Topologias em Estrela e em Floco de Neve

Como foi descrito anteriormente, existem dois tipos de tabelas: factos e dimensões. Uma vez que as dimensões servem de contexto aos factos, estes são o elemento central do modelo de dados. Possuem ligações para as várias dimensões que utilizem através de *surrogate keys* (identificadores únicos de cada registo).

Há essencialmente duas topologias utilizadas: Estrela e Floco de Neve (Chaudhuri & Dayal, 1997). A diferença está no facto de, na topologia Floco de Neve, as tabelas de dimensão possuírem ligações para outras tabelas que não as de factos, enquanto na topologia em Estrela isto não acontece. Seguem-se duas imagens com um exemplo de cada:

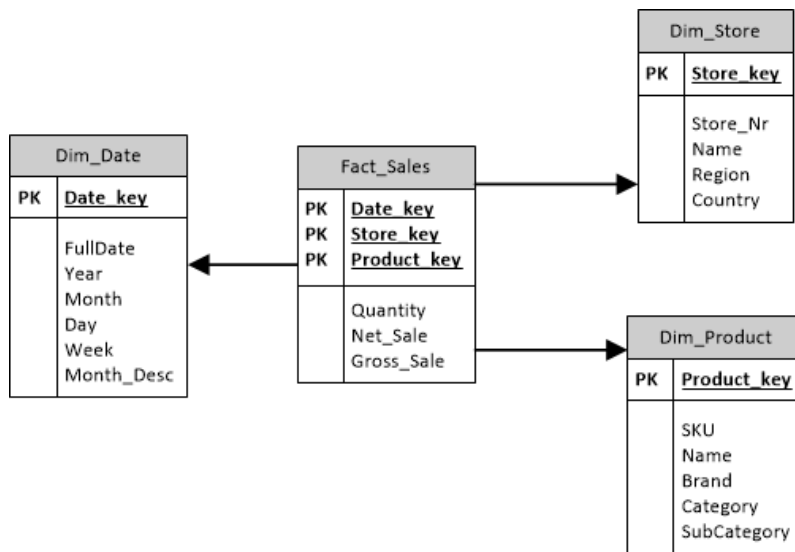


Figura 3 – Topologia em Estrela

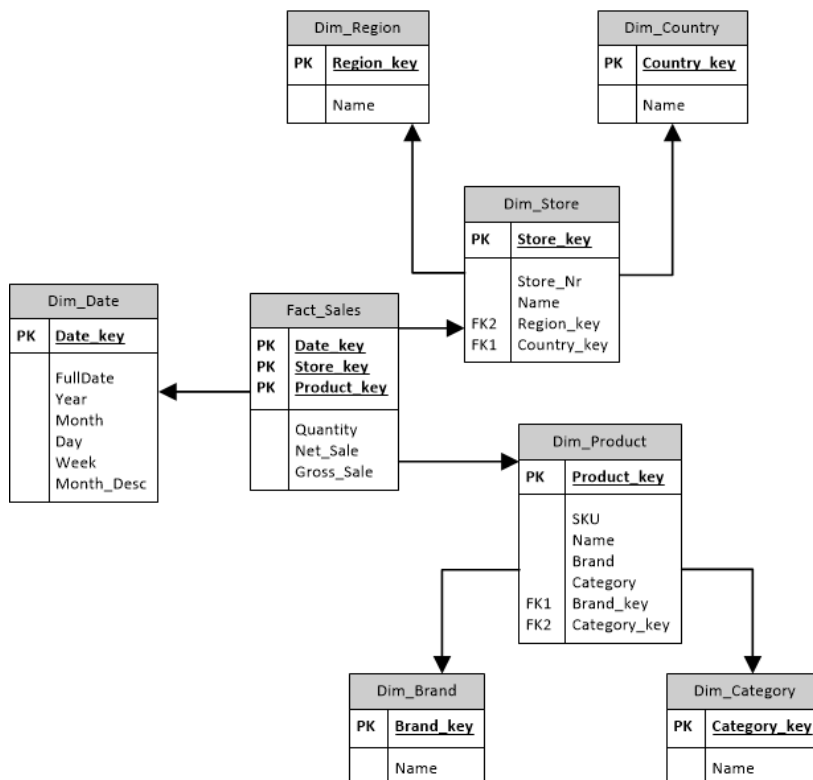


Figura 4 – Topologia em Floco de Neve

Pode-se ver nas imagens que o modelo de dados da topologia Floco de Neve se encontra normalizado, ao contrário da topologia em Estrela. Isto proporciona maior flexibilidade na gestão das dimensões. No entanto, esta normalização pode piorar o desempenho das consultas, uma vez que é necessário efetuar o cruzamento entre mais tabelas, o que acaba por impactar no plano de execução das *queries* lançadas pelo utilizador.

Existe ainda uma terceira topologia chamada Constelação de Factos, que nada mais é do que múltiplas tabelas de factos que partilham dimensões. (Chaudhuri & Dayal, 1997)

2.1.3 Carregamento no Armazém de Dados

Um dos componentes mais importantes num Armazém de Dados é o responsável pelo carregamento de informação neste. Estes dados são provenientes, na sua grande maioria, dos sistemas operacionais já existentes na companhia, sendo que alguns fazem parte de sistemas antigos, denominados *legacy*.

Uma vez que estamos a falar de vários sistemas, cada um com as suas próprias características tanto técnicas como ao nível da respetiva modelação de dados, o processo de extração e carregamento de dados no AD pode ser bastante complexo. (Mayer, 2014)

Face a este problema, foram desenvolvidas ferramentas que ajudam neste processo. Seguem-se alguns exemplos:

- Microsoft SQL Server Integration Services (<http://www.microsoft.com>) (Microsoft, 2014)
- Oracle Data Integrator (<http://www.oracle.com>) (Oracle, 2014)
- IBM InfoSphere DataStage (<http://www.ibm.com>) (IBM, 2014)
- SAS Data Integration Studio (http://www.sas.com/en_us/home.html) (SAS, 2014)
- SAP – Business Objects Data Integrator (<http://www.sap.com>) (SAP, 2014)

Estas ferramentas têm o nome de ETL (*Extract, Transform, Load*). Tal como o próprio nome indica, estas dividem-se em 3 fases:

- *Extract*: A primeira fase de um processo de carregamento de dados é a extração propriamente dita de um sistema fonte.
Este processo está dependente do sistema fonte, uma vez que este deve possuir um método que permita a ligação por parte de sistemas externos. Alguns exemplos de interfaces que permitem esta interconectividade:
 - ODBC
 - OLEDB
 - Oracle Open Connect

Na eventualidade de o sistema não possuir nenhuma interface de ligação ao exterior (comum em sistemas *legacy*), podem ser utilizados *flat files* (ficheiros de texto) como forma de importação no AD.

- *Transform*: Após a extração destes dados para uma área do Armazém de Dados denominada *Staging Area*⁴, a fase de transformação trata-se assim de um passo fundamental na grande maioria dos casos, no que diz respeito à “limpeza” dos dados. É este o processo mais trabalhoso do ETL, no entanto, a utilização de ferramentas como as de *profiling*, ajudam na sua execução.
É nesta fase que surgem grande parte dos erros e anomalias na informação proveniente dos sistemas fontes, uma vez que se tratam de grandes quantidades de informação com origem em múltiplos sistemas. (Chaudhuri & Dayal, 1997)
Isto verifica-se especialmente quando os sistemas fonte envolvidos não possuem validação de alguns campos e/ou utilizam diferentes tipos de dados para representar o mesmo. Por exemplo, o sexo de um indivíduo pode ser representado por M/F num determinado sistema, e por 0/1 noutra. Isto leva a que seja necessário definir um *standard* a ser implementado no Armazém de Dados, de forma a uniformizar estas duas formas de representação do mesmo atributo.
É por isso uma fase crucial, especialmente quando estamos a falar de sistemas *legacy*, devido às suas limitações técnicas.
- *Load*: Encontrando-se a informação já uniformizada e pronta para carregamento no Armazém de Dados, segue-se a fase deste mesmo carregamento.
Durante esta fase, efetua-se uma ligação ao AD, utilizando-se uma interface que este disponibilize (tal como na fase de extração), e inserem-se os vários registos.
Um ponto importante neste passo diz respeito à eventualidade de ocorrerem falhas ao inserir. Deve ser permitido voltar a inserir a informação sem que isto origine inconsistência de dados.
Todo este processo de ETL tem impacto no desempenho da Base de Dados (e pode levar, em alguns casos, a que esta fique indisponível), pelo que é de grande importância que este processo seja transparente para os utilizadores finais. A solução passa por escolher uma janela temporal adequada, normalmente durante a madrugada, em que a utilização da plataforma pelos utilizadores é mínima ou até mesmo nula. (Oracle, 2012)
Na Figura 5 é possível visualizar um fluxo de ETL na ferramenta SQL Server Integration Services.

⁴ - Local de armazenamento temporário num Armazém de Dados, onde se efetua transformações aos dados a carregar.

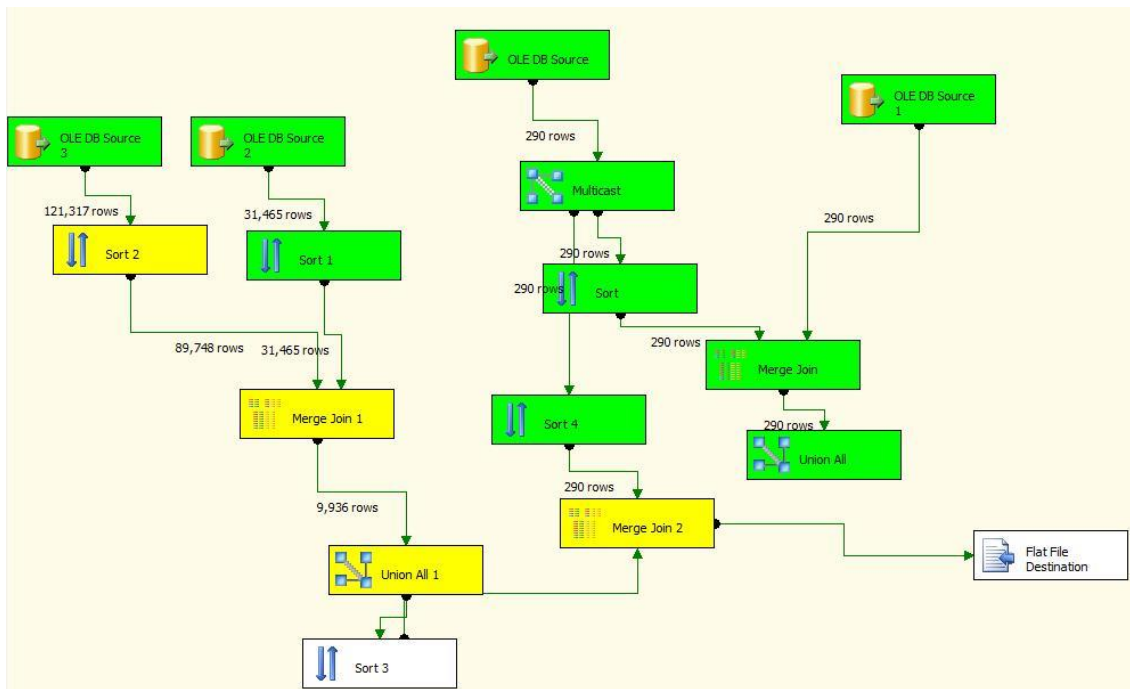


Figura 5 - Ferramenta ETL (SQL Server Integration Services)⁵

2.1.4 Desempenho

Um dos aspetos principais da existência dos Armazéns de Dados diz respeito ao desempenho com que são executadas consultas e nos impactos que estas têm a um nível organizacional.

“À medida que um *Armazém de Dados* cresce, garantir o desempenho de resposta a queries *ad-hoc* de cada vez mais utilizadores torna-se um grande desafio.” (DeWitt, et al., s.d.)

Num sistema operacional, que está constantemente a ser acedido para transações pontuais, não é aceitável sobrecarregar a base de dados (BD) com extrações massivas de dados, pois corre-se o risco de tornar o sistema mais lento, ou até mesmo de o bloquear. Pelo contrário, um Armazém de Dados é concebido para permitir realizar este tipo de extrações, sem que estas tenham impacto para os restantes utilizadores. Uma vez que o carregamento dos dados num AD ocorre normalmente durante a madrugada, no dia seguinte este apenas é acedido para tarefas de consulta de informação.

Temos assim gráficos distintos que representam a frequência de acesso a sistemas OLTP e a sistemas OLAP:

⁵ Retirado de http://consultingblogs.emc.com/jamiethomson/archive/2007/03/08/SSIS_3A00_-_OnPipelineRowsSent.aspx

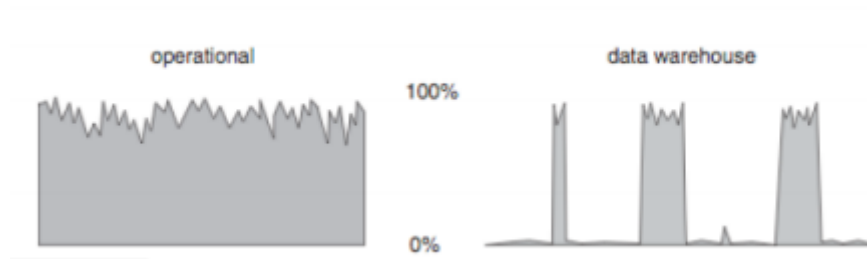


Figura 6 – OLTP vs OLAP, frequência de acesso⁶

Pode-se concluir que, com a existência de um Armazém de Dados numa organização, o sistema operacional não é afetado, no que diz respeito ao seu desempenho e disponibilidade.

Uma vez que um AD retém informação histórica, as análises efetuadas sobre este abrangem em grande parte dos casos, dados respeitantes a vários dias, meses ou até anos. É assim necessário ter em consideração o tempo de resposta a estas análises, assim como o impacto que isto tem no sistema.

Existem várias técnicas e fatores, tanto relacionados com *hardware* como com *software*, que influenciam este desempenho, sendo os principais os seguintes:

- **Hardware utilizado:** o *hardware* continua a ser um dos fatores mais relevantes. De nada serve implementar várias técnicas de melhoramento de desempenho se o *hardware* for obsoleto.
- **Técnicas de particionamento dos discos:** Uma vez que apenas recentemente surgiram os discos SSD⁷, a utilização de técnicas de RAID⁸ em discos com componentes mecânicos (eg: SAS⁹) permite dividir a carga entre os discos, sem que isto tenha um grande impacto nos custos, de forma a que estes acompanhem o aumento de desempenho dos processadores e memórias. (Patterson, et al., 1988)
No entanto, cada vez menos estes discos mecânicos conseguem acompanhar o restante *hardware*, sendo que, eventualmente, a solução passa pela aquisição de discos SSD.
- **Índices:** A utilização de índices (em tabelas) melhora o desempenho das consultas. No entanto, é necessário analisar sobre que atributos devem ser criados. (Chaudhuri & Dayal, 1997)

Existem vários tipos de índices, nomeadamente:

- Bitmap – Utiliza operações de lógica sobre vetores binários e é implementado em atributos com baixa cardinalidade.
- Hash – Utiliza um algoritmo de *Hash* como forma de representar o endereço de cada registo.

⁶ Retirado de (Oliveira, 2013)

⁷ *Solid State Drive* - Discos de armazenamento com velocidades muito superiores aos convencionais.

⁸ *Redundant Array of Independent Disks* - Técnica de particionamento de discos.

⁹ *Serial Attached SCSI*

- B-Tree – Normalmente aplicado sobre vários atributos, adequa-se a situações em que estes possuam uma elevada cardinalidade.
- **Tabelas particionadas:** Num Armazém de Dados, as tabelas de factos contêm informação histórica, pelo que efetuar consultas sobre estas é uma operação extremamente pesada para os suportes de armazenamento atuais. Uma forma de agilizar este processo consiste em particionar estas tabelas. Se estivermos a falar de uma tabela que armazene uma *snapshot* diária de informação de vendas, por exemplo dos últimos 2 anos, com detalhe ao artigo e à loja, rapidamente chegamos a milhões ou mesmo biliões de registos nessa mesma tabela. Ao efetuarmos uma consulta cujo resultado diga respeito, por exemplo, a 2 dias, e se a tabela se encontrar particionada ao dia, é possível aceder diretamente a apenas duas partições, o que reduz o tempo de resposta drasticamente.

Atualmente, na versão 11g Release 2 da Oracle Database, existe uma ferramenta cujo objetivo passa por auxiliar na estratégia de particionamento. A Figura 7 demonstra quais os tipos de particionamento existentes nessa versão da Oracle Database.

Partitioning Strategy	Data Distribution	Sample Business Case
Range Partitioning	Based on consecutive ranges of values.	Orders table range partitioned by order_date
List Partitioning	Based on unordered lists of values.	Orders table list partitioned by country
Hash Partitioning	Based on a hash algorithm.	Orders table hash partitioned by customer_id
Composite Partitioning <ul style="list-style-type: none"> • Range-Range • Range-List • Range-Hash • List-List • List-Range • List-Hash • Hash-Hash 	Based on a combination of two of the above-mentioned basic techniques of Range, List, Hash, and Interval Partitioning	Orders table is range partitioned by order_date and sub-partitioned by hash on customer_id Orders table is range partitioned by order_date and sub-partitioned by range on shipment_date

Figura 7 – Tipos de particionamento ¹⁰

- **Agregações:** Muitas das vezes o pretendido pelos utilizadores é visualizar informação a um nível que não o mais atómico. Utilizando o exemplo das vendas, se o pretendido for consultar o valor total de vendas do mês de Março de 2014, e a tabela com os dados guardar informação ao nível do dia, ao invés de se consultar esta tabela, existe a possibilidade de se consultar uma outra com informação agregada ao mês. Segue-se um exemplo.

Da tabela diária seguinte:

¹⁰ Retirado de (Baer, 2010)

Tabela 3 - Exemplo tabela diária

<i>Time_key</i>	<i>Valor</i>
20140301	3243
20140302	2954
20140303	3134
20140304	5432
20140305	4321
20140306	5543
...	...
20140531	3454

Resulta a seguinte tabela mensal agregada:

Tabela 4 - Exemplo tabela agregada mensalmente

<i>Time_key</i>	<i>Valor</i>
201403	124358
201404	102432
201405	97442

Desta forma, consulta-se apenas uma linha, ao contrário de 31 (uma para cada dia), cujos valores de vendas se somariam.

Neste exemplo em concreto, este conceito foi aplicado sobre a dimensão Data. No entanto é possível agregar tabelas ao nível de outras dimensões. Uma dimensão muito utilizada também é a dimensão Produto, em que podem surgir agregações ao nível da Categoria ou de qualquer outro elemento que faça parte da sua estrutura mercadológica.

2.2 People Counters

No retalho, cujo grande foco diz respeito à área das vendas, a contagem dos clientes é um aspeto importante no que diz respeito ao desempenho do negócio.

É possível, possuindo uma solução de *Business Intelligence* com base num Armazém de Dados, efetuar análises, por exemplo, ao valor bruto ou quantidades de venda no último ano, de forma a se obter uma perspetiva da tendência durante esse período. Com isto, é possível agir com base nos resultados obtidos.

No entanto, um fator determinante a ter em conta é o de número de visitantes que entram na loja. Isto é possível colocando-se dispositivos à entrada da loja que detetam entradas e saídas e permite análises ainda mais precisas. Servindo a área de retalho vestuário como exemplo, o número de pessoas que entra na loja ao longo do dia é fundamental para uma análise do seu desempenho.

Isto deve-se ao facto de nesta área de retalho ser muito comum um potencial cliente entrar na loja e sair sem ter comprado nenhum artigo. Existem diversos fatores que levam a que isto aconteça, tais como não ter encontrado o que pretendia, o ambiente da loja incomodá-lo (música muito alta, temperatura elevada...), ter decidido entrar só para ver os artigos que a loja tem, nomeadamente de novas coleções, etc. Se existirem muitos visitantes com este perfil, pode ser um sinal de que a capacidade da loja em cativar o público a comprar é fraca, sendo que talvez seja necessário tomar medidas quanto a este problema.

Pode-se assim concluir que “O desempenho de uma loja está relacionado com a sua capacidade de atrair visitantes e “persuadi-los” a adquirirem artigos, de forma a serem considerados como sendo geradores de lucro”. (Perdikaki, et al., 2012)

Esta capacidade de conversão dos visitantes de uma loja a comprarem tem o nome de “Taxa de Conversão”, e tem a seguinte fórmula:

$$TC = \frac{\text{Número de Visitantes que compraram}}{\text{Número de Visitantes totais}} * 100$$

Atualmente, pode ser reforçada de várias maneiras, como por exemplo com:

- Promoções
- Lugares de destaque para artigos selecionados
- Eventos
- Ambiente na loja agradável

Esta taxa de conversão é altamente dependente do negócio em questão, e tanto se pode situar perto dos 0% como dos 100%. Um Web Site pode converter apenas 2% dos seus visitantes, enquanto que um supermercado converte com um valor muito próximo dos 100%, pelo menos em algumas das secções. (Conroy & Bearse, 2006)

A seguinte imagem ilustra esta discrepância entre várias áreas do retalho.

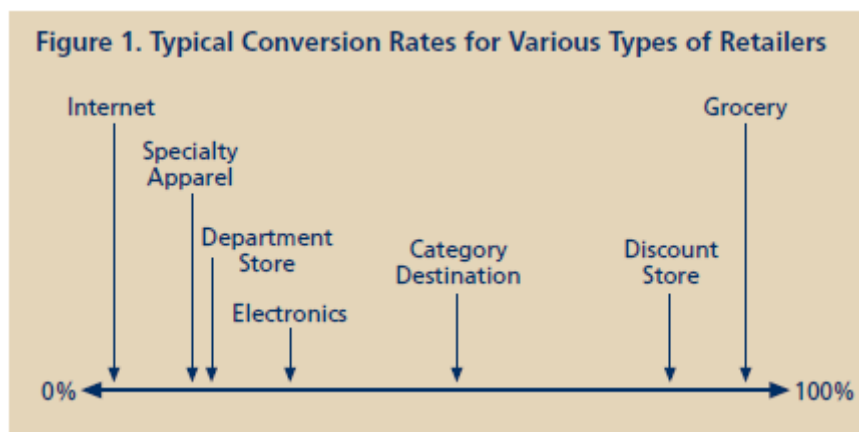


Figura 8 – Taxas de conversão por área de retalho¹¹

A (Conroy & Bearse, 2006) realizou vários estudos tendo como principal foco esta taxa de conversão e quais os fatores que a influenciam. Um caso interessante é o facto de lojas disponíveis por múltiplos canais, *online* e físicas, converterem facilmente os clientes nas suas lojas físicas. Isto deve-se ao facto de o cliente já ter pesquisado no *Web Site* o que pretendia, e possuir assim uma forma fácil e rápida de adquirir o produto na loja física. Esta taxa de conversão está também relacionada com o facto de estas lojas múltiplo-canal transmitirem normalmente uma imagem de confiança.

Além da vantagem que a contagem de clientes tem ao permitir o cálculo da taxa de conversão, traz também outras vantagens, tais como ser possível aferir quais os períodos de maior afluência de clientes. Se a informação do número de clientes for armazenada ao longo do dia, como por exemplo, de hora a hora, torna-se possível saber quais são estes períodos de maior afluência. Com os resultados desta análise consegue-se, entre outros, ajustar o número de funcionários consoante a necessidade, o que leva a uma otimização de recursos e a uma consequente redução nos custos. (Perdikaki, et al., 2012)

Existem outras alternativas para se obter o número de clientes sem se utilizarem os dispositivos concebidos para o efeito, nomeadamente:

- Contagem dos talões
- Contagem dos cartões de cliente utilizados

Estas alternativas, no entanto, não permitem saber o número de clientes que saem sem realizar qualquer compra, o que impossibilita a realização das análises mencionadas anteriormente. A obtenção destes dados através de sistemas próprios é, por isso, algo que se justifica na maioria dos casos face às alternativas apresentadas anteriormente, uma vez que estes fornecem uma contagem exata de pessoas que entraram na loja.

¹¹ Retirado de (Conroy & Bearse, 2006)

Alguns exemplos de sistemas de contadores:

- LASE PeCo (<http://www.peoplecounter.de/en/>)
- TRAF-SYS (<http://www.trafsys.com/people-counting/>)
- SensMax (<http://sensmax.eu/>)
- Irisys People Counting (<http://www.irisys.co.uk/people-counting/>)
- Experian FootFall (<http://www.footfall.com/>)
- SenSource (<http://www.sensourceinc.com/peoplecounters.htm>)
- TrueView People Counter (<http://www.cognimatics.com>)
- IPSOS Retail Performance (<http://www.ipsos-retailperformance.com/>)
- SMS StoreTraffic (<http://www.storetraffic.com/>)
- Headcount Systems (<http://www.headcountsystems.com/index.html>)

Segue-se uma descrição mais detalhada de dois sistemas muito utilizados de contadores: LASE PeCo e Experian FootFall.

2.2.1 LASE PeCo

Um dos sistemas existentes atualmente é o LASE PeCo¹², proprietário da LASE PeCo Systemtechnik GmbH.

Fundada em 2008, trata-se de uma empresa derivada da LASE Systemtechnik GmbH, sendo esta já especialista em *laser scanners*. Assim, a LASE PeCo utiliza esta tecnologia e respetivo *know-how* na área de contagem de clientes.

Normalmente associada à área do retalho, os equipamentos que efetuam a contagem de clientes disponibilizados pela LASE podem ser aplicados tanto na entrada das lojas como em locais estratégicos dentro destas, o que permite saber quais as áreas com maior afluência de clientes.

No entanto, este sistema pode ser aplicado noutras áreas além do retalho. Seguem-se alguns exemplos existentes no *website* da LASE PeCo:

- **Cidades:** Instalando sensores em zonas pedestres é possível, por exemplo, verificar as alterações devido a efeitos climatéricos ou sazonais, devido ao surgimento de novas superfícies comerciais ou outros fatores.
- **Delimitações de perímetro:** tudo o que seja uma zona privada e de acesso restrito beneficia de soluções baseadas em *laser tracking*. Locais como prisões, propriedades privadas, edifícios administrativos, etc., aumentam os seus níveis de segurança com a aplicação destes sistemas.

¹² <http://www.peoplecounter.de/en/>

- **Eventos:** Locais onde se realizem eventos podem beneficiar deste tipo de sistemas, uma vez que proporcionam uma contagem de visitantes automática. A LASE PeCo disponibiliza, neste caso, serviços de aluguer.

Uma das soluções disponibilizadas pela LASE PeCo diz respeito a um conceito denominado *Heat Mapping*. Muito utilizado na área do retalho, isto permite visualizar quais as áreas da loja com maior afluência de clientes. Tal é possível com a colocação dos equipamentos dentro da própria loja e em locais estratégicos, tal como foi referido anteriormente.

Com a utilização destas tecnologias e a respetiva representação através de *Heat Map*, obtém-se uma maior compreensão do comportamento dos clientes em loja, assim como também é possível avaliar se as promoções e produtos em destaque estão a ter resultados práticos. (LASE PeCo, 2014)

Assim, é possível obter-se imagens como a seguinte:



Figura 9 – Imagem Heat Map numa loja¹³

No que diz respeito aos equipamentos que a LASE PeCo disponibiliza para ambientes indoor, existem em comercialização os seguintes:

2.2.1.1 PeCo SC

Este sensor funciona através de câmaras 3D e a largura da sua área de deteção é de até 3.5 metros. Tem uma eficácia de 98% e é ideal para ser aplicada em locais onde a passagem de pessoas esteja dispersa, como por exemplo em entradas de centros comerciais.

Existe a variante PeCo SC High, em que a largura da área de deteção aumenta para 4 metros em relação à versão normal, assim como a altura máxima de montagem, que passa também de 4 para 6 metros nesta versão.

¹³ Retirado de <http://www.retailrescue.com/retail-security-and-people-counting.htm>

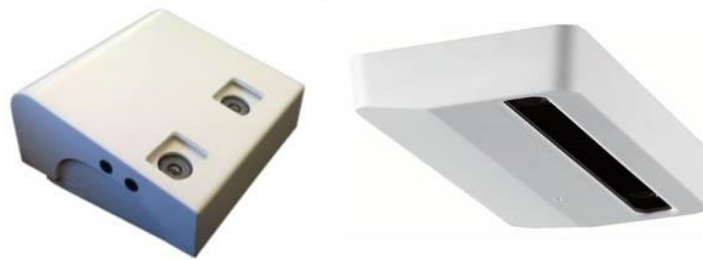


Figura 10 – PeCo SC / SC High¹⁴

2.2.1.2 PeCo TC

Com base em tecnologia de deteção por Infravermelhos, este sensor é ideal para ser aplicado em locais onde a zona de deteção seja reduzida, como por exemplo em entradas de pequenas lojas.



Figura 11 – PeCo TC¹⁵

Tem a vantagem de poder ser instalado a alturas inferiores a 4 metros, no entanto a área de deteção diminui à medida que a altura também diminui. Isto deve-se ao facto de a zona de deteção poder ser representada pela forma de um cone, como se pode verificar na Figura 12.

¹⁴ Retirado de <http://www.peoplecounter.de/en/>

¹⁵ Retirado de <http://www.peoplecounter.de/en/>

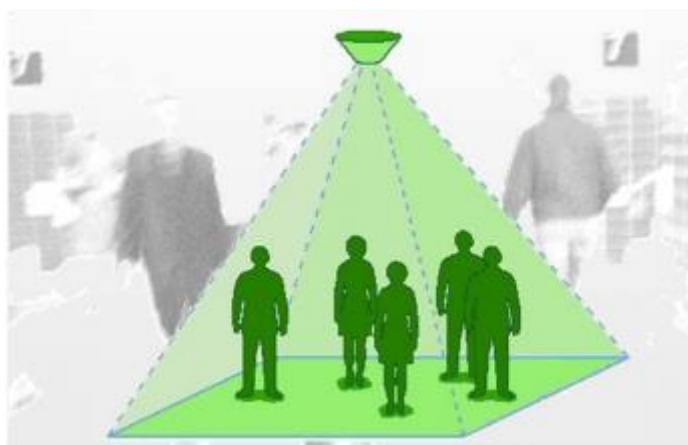


Figura 12 – PeCo TC - Área de detecção ¹⁶

2.2.2 Experian FootFall

O Experian FootFall¹⁷ é outro dos grandes concorrentes na área de People Counting. Inicialmente fundada em 1991 sob o nome “FootFall Ltd”, esta empresa sediada em Inglaterra foi adquirida em 2005 pela Experian, uma consultora multinacional na área de gestão de informação.

2.2.2.1 FootFall Indices

Um dos fatores de diferenciação da FootFall diz respeito aos “FootFall Indices”. Estes permitem aos retalhistas a monitorização do impacto que certos eventos têm no comportamento dos clientes. Estes eventos podem estar relacionados com a meteorologia, férias, questões económicas e financeiras, eventos desportivos ou outros eventos nacionais ou internacionais. (Experian Footfall, 2014)

Trata-se da disponibilização aos seus clientes, de relatórios com informação de contadores calculados pela própria FootFall com base em lojas dispersas por várias zonas do país. Isto permite ter uma perceção do impacto que estes fatores têm no negócio e comparar os resultados obtidos internamente com o panorama nacional fornecido pelos *Indices*.

Atualmente, estão disponíveis os seguintes países/regiões:

- República da Irlanda
- Alemanha
- França
- Hong Kong
- Itália

¹⁶ Retirado de <http://www.peoplecounter.de/en/>

¹⁷ <http://www.footfall.com/>

- Polónia
- Portugal
- Espanha
- Suíça
- Reino Unido e Irlanda

Segue-se um exemplo de um relatório disponibilizado no Web Site da FootFall como forma de demonstração. Diz respeito ao Reino Unido e, para cada região, o valor em percentagem indica se o número de visitantes médios em lojas aumentou ou diminuiu em relação ao período homólogo:

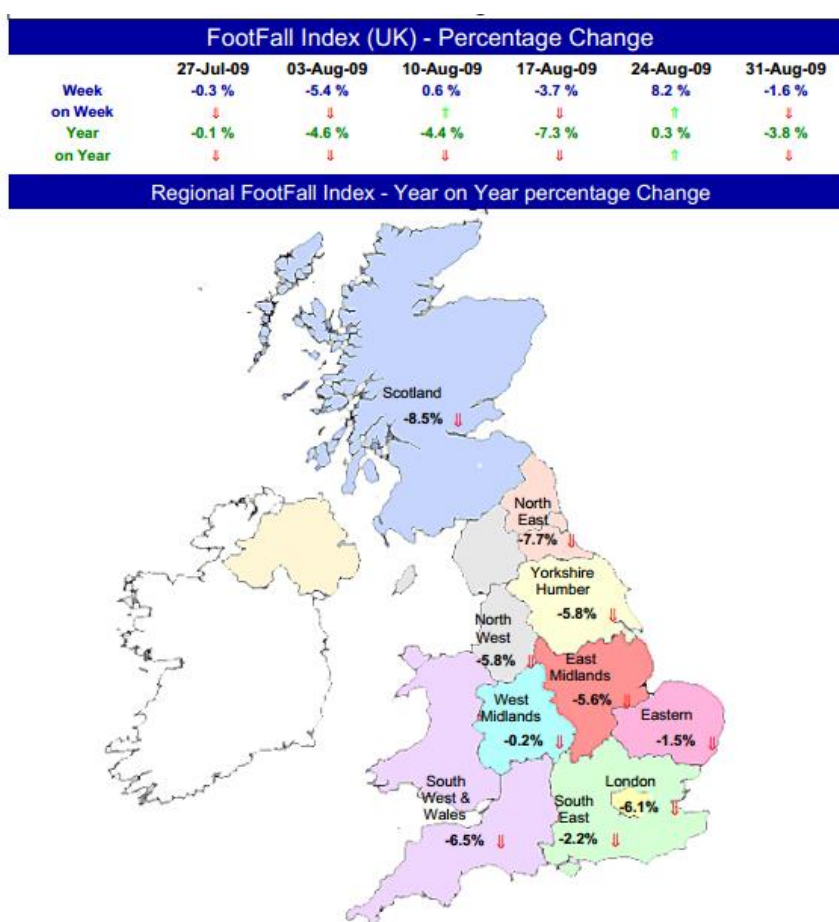


Figura 13 – Exemplo FootFall Indices – Mapa¹⁸

Outra análise bastante útil também disponibilizada pelos Índices permite ter uma perceção por semana da evolução do número de clientes, e comparação com períodos homólogos, como se pode comprovar na Figura 14:

¹⁸ Retirado de <http://www.footfall.com/>



Department Store Index - w/c 31/08/09

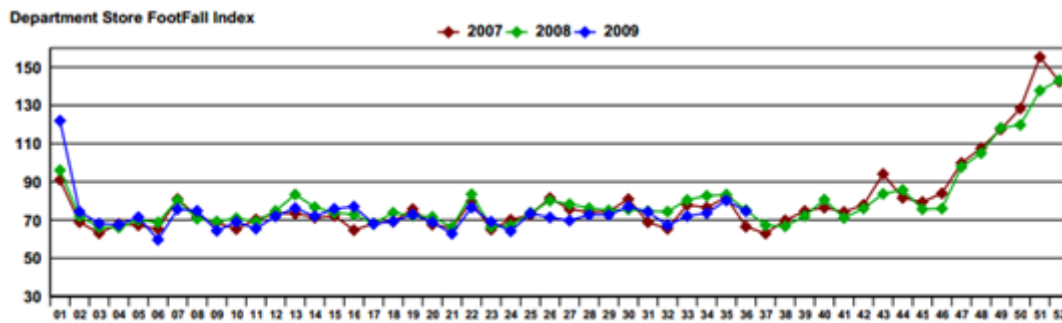


Figura 14 – Exemplo FootFall Indices - Semanas¹⁹

2.2.2.2 Tecnologia

No que diz respeito à tecnologia utilizada pela Experian, esta baseia-se no seguinte:

- Câmaras térmicas
- Câmaras de vídeo estereoscópicas
- Laser
- Raios Infravermelhos
- Contadores de automóveis

No entanto, cada tecnologia possui determinada finalidade, adaptada a um meio ambiente específico. Por exemplo, sensores de infravermelhos não são adequados para locais com grande afluência de tráfego. (Experian Footfall, 2014)

É assim necessário realizar uma análise cuidada do meio ambiente onde serão instalados os sensores, de forma a determinar a tecnologia que melhor se adapta à situação.

¹⁹ Retirado de <http://www.footfall.com/>

2.3 Reconhecimento Facial

Uma área que muito tem evoluído nos últimos anos é a do reconhecimento facial. Isto deve-se ao facto de trazer vantagens em inúmeras áreas de aplicação. (Introna & Nissenbaum, 2009)

Esta tecnologia permite analisar desde imagens até vídeos em tempo real e, através de algoritmos desenvolvidos especificamente com esta finalidade, distinguir rostos de pessoas que se encontrem enquadradas na imagem. Além desta simples deteção, é também possível proceder ao reconhecimento de uma face. Ou seja, dependendo da tecnologia utilizada e da sua aplicação, é possível, após ter sido feita a respetiva deteção, comparar o resultado obtido com registos numa base de dados. Assim, consegue-se não apenas detetar rostos, mas também associá-los a resultados obtidos previamente. Isto é particularmente útil na área forense, como se poderá ver na secção referente às áreas de aplicação desta tecnologia.

No entanto, independentemente da tecnologia utilizada e da área onde esta é aplicada, o reconhecimento facial não é 100% eficaz, pelo que poderão surgir erros de deteção, principalmente no que diz respeito a vídeos em tempo real. Isto deve-se ao facto de ser uma tecnologia altamente dependente da qualidade de imagem e das suas características. Convém, por isso, ter em conta que, apesar de hoje em dia se obterem resultados muito satisfatórios, esta é uma tecnologia em constante desenvolvimento.



Figura 15 – Exemplo de aplicação de *Reconhecimento Facial*²⁰

²⁰ Retirado de <http://www.digitalcamerareview.com>

2.3.1 Áreas de utilização

Esta tecnologia tem um grande potencial em várias áreas, com utilizações que vão desde o simples entretenimento até complexos sistemas de segurança. Destacam-se os seguintes:

2.3.1.1 Vigilância / CCTV

De acordo com (Introna & Nissenbaum, 2009), a aplicação de Reconhecimento Facial em soluções de vigilância trata-se de “um dos cenários mais ambiciosos, dada o estado atual da tecnologia.”

Através da utilização de circuitos fechados de televisão (CCTV), atualmente é possível cobrir diversas áreas de um determinado local, sejam estas *indoor* ou *outdoor*. Estes sistemas funcionam com a colocação de câmaras de vigilância em locais estratégicos, que transmitem as imagens capturadas para uma central de controlo, responsável pela gestão destas transmissões.

Possuindo tanto de ambicioso como de controverso, trata-se assim de uma aplicação bastante útil para a sociedade.

Seguem-se imagens de um sistema CCTV, tanto dos dispositivos responsáveis pela captação de imagens, como de uma central de controlo:



Figura 16 – Sistema CCTV (dispositivos de captação de imagens)²¹

²¹ Retirado de <http://upload.wikimedia.org/>



Figura 17 – Sistema CCTV (central de controlo)²²

Estes sistemas podem ser utilizados pelas autoridades em locais públicos de forma a garantir a segurança dos cidadãos. A aplicação de reconhecimento facial nestes casos, aliada à capacidade de comparação com imagens armazenadas em bases de dados, aumentaria ainda mais a segurança nestes locais, pois permitiria uma resposta por parte das autoridades mais rápida. (Introna & Nissenbaum, 2009)

2.3.1.2 Sistemas de autenticação

Hoje em dia os diferentes métodos de autenticação existentes são cada vez mais importantes. Desde simples *software* até áreas restritas de certas organizações, como por exemplo governamentais, a autenticação desempenha um papel fundamental. Perante esta necessidade de restringir acessos, é a autenticação que determina se o utilizador realmente é quem ele diz ser.

Isto pode ser executado de diversas maneiras, nomeadamente com informação biométrica. Esta diz respeito a dados biológicos do utilizador que o distinguem dos restantes. Alguns exemplos:

- Retina
- Voz
- Impressão digital

É importante referir que, apesar de alguns métodos de aquisição desta informação transmitirem robustez na sua aplicação (por exemplo, impressões digitais), “é impossível

²² Retirado de <http://www.techprosecurity.com>

afirmar que exista um método que seja o melhor em todas as aplicações, tecnologias e políticas de administração” (Wayman, et al., 2005)

Sendo assim, a utilização do rosto para motivos de autenticação é normalmente conjugada com outros métodos, uma vez que este por si só é falível. Um exemplo disto é a junção desta tecnologia com a autenticação por impressão digital.

Seguem-se vantagens e desvantagens da utilização da tecnologia de reconhecimento facial aplicada em sistemas de autenticação:

Tabela 5 - Vantagens/Desvantagens reconhecimento facial como identificação biométrica

<i>Vantagens</i>	<i>Desvantagens</i>
<i>-Método principal de identificação utilizado pelos humanos</i>	<i>-Eficácia depende de diversos fatores, tais como qualidade e ângulo da foto</i>
<i>-Rápida aquisição de amostra</i>	<i>-Facilmente contornável</i>
<i>-Equipamentos baratos</i>	<i>-Baixa eficácia como sistema de autenticação</i>

No entanto, quando associado a tecnologia 3D, a eficácia do reconhecimento facial aumenta drasticamente, uma vez que é produzido um “molde” da face do utilizador, o que se torna muito menos propício a erros aquando da comparação com a base de dados.

Esta tecnologia 3D foi testada num aeroporto de Moscovo, com eficácia de 100%. (Homeland Security News Wire, 2011)

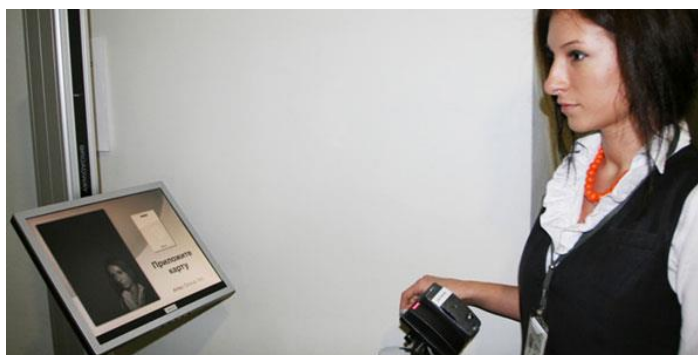


Figura 18 – Sistema biométrico de reconhecimento facial²³

²³ Retirado de (Future Travel Experience, 2011)

2.3.1.3 Smartphones / Câmaras digitais

Uma das áreas onde a tecnologia de reconhecimento facial se encontra disponível para o grande público é a das câmaras digitais e *Smartphones*. Cada vez com mais funções que dão uso a esta tecnologia, destacam-se assim algumas das já existentes.

2.3.1.3.1 Funções de fotografia

Presente tanto em câmaras digitais como em *Smartphones*, algumas das funções existentes que utilizam a tecnologia de reconhecimento facial com foco em funções aplicadas à fotografia, são as seguintes:

- Focagem automática nos rostos
- Obturador acionado assim que seja detetado um sorriso
- *Zoom* automático na face após ter sido tirada a foto



Figura 19 – Exemplo de reconhecimento facial aplicada à fotografia²⁴

2.3.1.3.2 Funções sociais

Com o crescimento das redes sociais, esta tecnologia tem vindo a ser utilizada como forma de identificação automática das pessoas presentes em fotografias.

É exemplo disso o projeto *DeepFace*²⁵, desenvolvido pelo Facebook, em que após se efetuar o *upload* de várias fotografias, as respetivas pessoas são detetadas e agrupadas de acordo com a sua semelhança, de forma ao utilizador identificar os amigos correspondentes. Isto é conseguido através de algoritmos que simulam redes neuronais, e tem uma eficácia de 97.25%, muito próxima da dos humanos, que se situa em 97.53%. (Anthony, 2014)

²⁴ Retirado de <http://www.digitalcamerareview.com>

²⁵ <https://www.facebook.com/publications/546316888800776/>

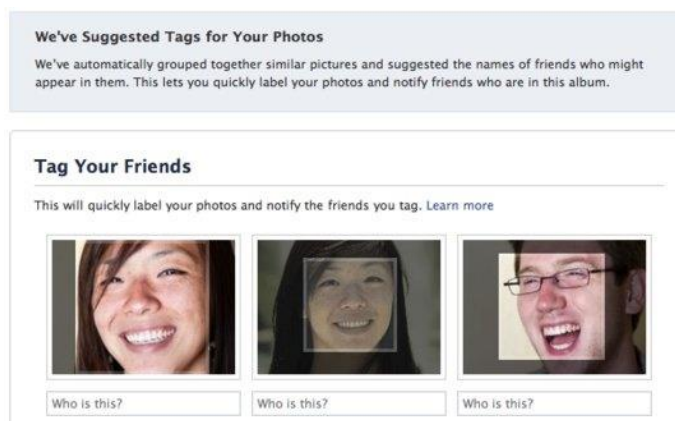


Figura 20 – Reconhecimento Facial no Facebook²⁶

2.3.1.3.3 Funções do Sistema Operativo

Sendo esta tecnologia cada vez mais acessível, a sua integração em produtos do dia-a-dia já existentes torna-se cada vez mais comum. É exemplo disto o destaque para o reconhecimento facial aquando do lançamento do Samsung Galaxy S4.

A junção desta tecnologia com a de *eye tracking* permite ao utilizador realizar várias ações em alguns locais do sistema operativo Android.

Em certas aplicações pré-instaladas é possível pausar e resumir um vídeo que se esteja a visualizar (Smart Pause), assim como efetuar *scroll* em certos locais, usando-se para isso apenas o olhar. (Shah, 2013)

Apresenta-se de seguida uma imagem capturada do próprio SO Android, com uma breve descrição do Smart Pause:

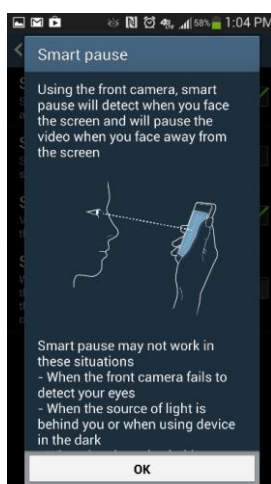


Figura 21 – Samsung Galaxy S4 Smart Pause²⁷

²⁶ Retirada de ("buy-facebook-fans", 2014)

2.3.1.4 Entretenimento

Outra das áreas disponíveis para o grande público que implementa a tecnologia de reconhecimento facial é a do entretenimento. Existe atualmente o Kinect 2 e a PlayStation 4 Camera, que fazem parte integrante das respetivas consolas (Xbox One e PlayStation 4). Nas secções seguintes são descritas duas aplicações desta tecnologia nesta área.

2.3.1.4.1 Autenticação

Tanto o Kinect 2 como a Playstation 4 Camera permitem efetuar login no sistema através do reconhecimento da face do utilizador. No entanto, e devido à natureza da utilização destes sistemas de entretenimento, esta autenticação tem apenas o objetivo de tornar o processo mais cómodo, uma vez que não garante os níveis de fiabilidade de um sistema profissional.

O Kinect 2 é, no entanto, ligeiramente superior neste processo de autenticação. Isto deve-se ao facto de utilizar tecnologia de Infravermelhos de forma a “disparar” fotões, o que lhe permite construir com grande precisão uma versão digital em 3D do que está a captar, de forma semelhante ao funcionamento de um sonar (FinalVerdict, 2013). Esta captura em 3D, como foi referido anteriormente, permite uma maior eficácia no processo de reconhecimento facial.

A PlayStation 4 Camera, por sua vez, utiliza duas câmaras que, funcionando em conjunto, realizam a triangulação do utilizador. Isto faz com que os resultados não sejam tão precisos como os do Kinect 2.

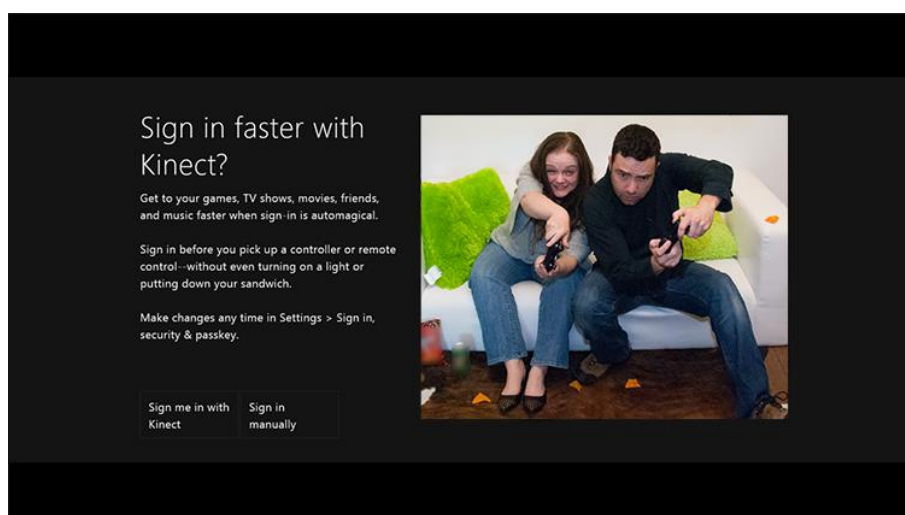


Figura 22 – Kinect 2 - Autenticação²⁸

²⁷ Retirado de <http://www.androidtapp.com/samsung-galaxy-s4-a-definitive-guide/how-to-use-smart-pause-for-video-on-samsung-galaxy-s4/>

²⁸ Retirado de (Xbox.com, 2014)

2.3.1.4.2 Skype

O seguinte exemplo não se baseia em reconhecimento facial, mas sim em *skeleton tracking*. No entanto, uma vez que são tecnologias com algumas semelhanças entre si, torna-se interessante realizar mesmo assim uma descrição.

Na Xbox One é possível realizar chamadas de vídeo através do Skype, tal como em qualquer outro dispositivo. Com o Kinect 2, no entanto, é possível efetuar o *tracking* de quem se encontrar enquadrado na imagem. Quando a pessoa presente na imagem se levanta e circula pela sala, a tecnologia de *skeleton tracking* que o Kinect 2 utiliza, permite que este faça *zoom* e o siga no seu percurso. (Bright, 2013)

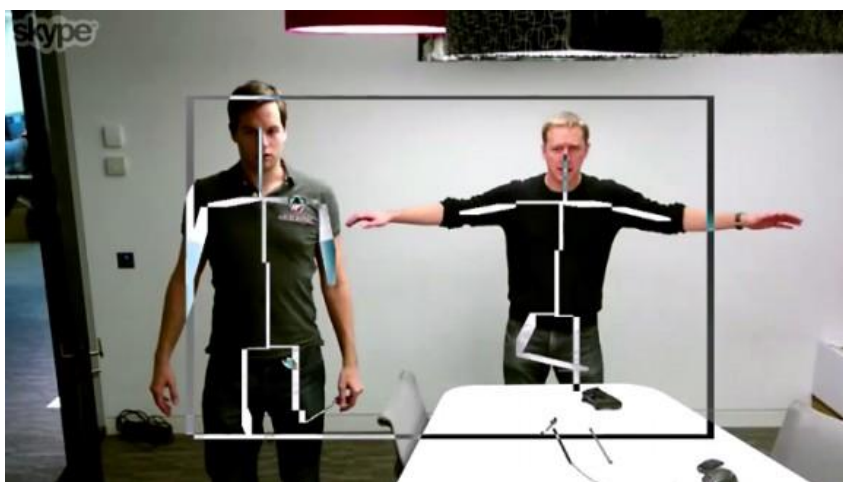


Figura 23 – Skype com Kinect – *Skeleton Tracking*²⁹

2.3.2 Algoritmos de reconhecimento

A tecnologia de reconhecimento facial tem o objetivo de simular o comportamento dos humanos nesta área. No entanto, uma máquina apenas trabalha com informação binária, pelo que, para esta, uma foto tem exatamente o mesmo formato que o texto, áudio, vídeo, etc.

Tendo isto em conta, os vários sistemas cujo objetivo passe pela deteção ou reconhecimento para posterior comparação de rostos implementam algoritmos que lhes permitem realizar este processo.

Antes de se apresentar alguns destes algoritmos, é importante referir que esta é uma área que depende de diversos fatores e características físicas do cenário, como por exemplo:

- Iluminação
- Ângulo
- Acessórios/Maquilhagem utilizada

²⁹ Retirado de (Langley, 2013)

Estes fatores podem influenciar drasticamente a identificação por parte do sistema, sendo que alguns deles afetam até o reconhecimento por parte dos humanos.

Existem, portanto, algoritmos tanto para detecção como para reconhecimento e posterior comparação de faces. Segue-se a descrição de alguns deles.

2.3.2.1 Haar-based Cascade Classifier (algoritmo de detecção)

Desenvolvido por Paul Viola e Michael Jones, este algoritmo está presente no OpenCV, uma livraria *open source* com algoritmos na área da imagem digital.

“O algoritmo utiliza um classificador *Ada-Boost* que “disseca” de uma forma agressiva a imagem, de forma a localizar rostos.” (Bolme, et al., 2007)

O seu objetivo passa por detetar faces, sendo que para o efeito é necessário um treino prévio com imagens que contenham caras, assim como imagens que não contenham nenhuma.

Este algoritmo tem também a vantagem de poder ser usado não apenas para detetar faces, mas sim qualquer objeto. Tal como referido anteriormente, realizando-se um treino prévio desta vez com o objeto em questão, torna-se possível esta detecção.

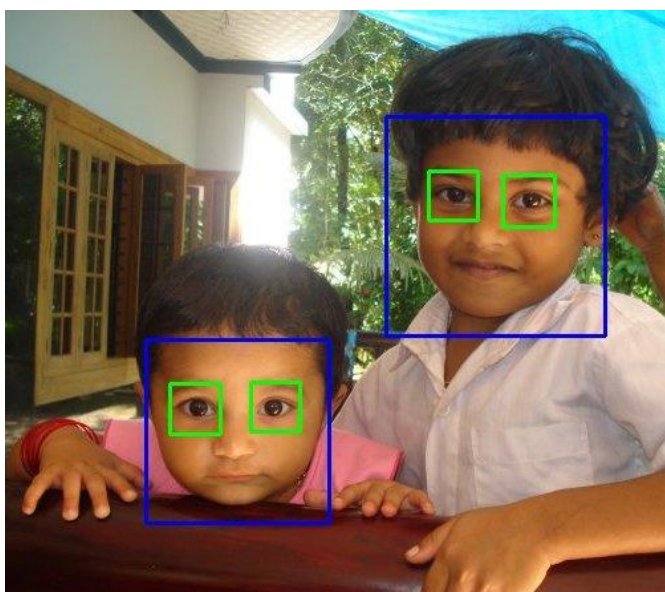


Figura 24 – Utilização do algoritmo “Haar-based Cascade Classifier”³⁰

2.3.2.2 CSU Principal Components Analysis (algoritmo de reconhecimento/comparação)

Algumas das aplicações da tecnologia de reconhecimento facial exigem deste uma comparação da face detetada no momento com uma base de dados já existente. Um exemplo

³⁰ Retirado de (OpenCV, 2014)

disto é a sua implementação em sistemas de segurança, como forma de autenticação. Isto requer uma captura prévia da face de cada utilizador, às quais se vai comparar a imagem captada no momento, de forma a ser efetuada uma busca por uma correspondência.

Um algoritmo que efetua este reconhecimento é o CSU PCA (*Principal Components Analysis*).

Tendo por base a representação da face sob a forma de vetores, um para cada *pixel*, este algoritmo efetua a comparação entre estes vetores, executando para isso operações de álgebra. (Bolme, et al., 2007)

2.3.2.3 CSU Elastic Bunch Graph Matching (algoritmo de reconhecimento/comparação)

Tal como o CSU PCA, o CSU EBGM tem o objetivo de identificar similaridades entre rostos.

Como foi dito anteriormente, muitas das vezes as capturas faciais não se encontram no formato ideal, seja devido à diferença de luz, ao ângulo a que a cara se encontra, etc. Isto traduz-se numa dificuldade acrescida ou até mesmo na impossibilidade por parte dos algoritmos lineares em processar a imagem.

O algoritmo *Elastic Bunch Graph Matching* (EBGM) tem estas variações em conta. Realizando operações com base nos pixels da imagem, e utilizando para isso os conceitos de “*Gabor filters*” e “*jets*”, o EBGM consegue detetar pontos específicos da cara, como por exemplo os olhos e a boca, independentemente das condições anteriormente referidas serem variáveis, sendo que num algoritmo linear não seriam detetados. (Introna & Nissenbaum, 2009)

2.3.3 Produtos Existentes

Atualmente existem várias soluções no mercado relativas à captura de informação através de reconhecimento facial. Seguem-se alguns exemplos:

- Cognitec (<http://www.cognitec.com/>)
- Aurora (<http://www.facerec.com/>)
- Intel AIM (<https://aimsuite.intel.com/>)
- NEC (<http://www.nec.com/en/global/solutions/security/products/index.html?>)
- EXVA Technologies (<http://www.exva.pt/>)
- Animetrics (<http://animetrics.com/forensicagps/>)
- InsideVisions (<http://insidevisions.com/>)

Uma vez que o pretendido no desenvolvimento da presente dissertação passa pela aquisição de informação demográfica dos visitantes de cada loja, destacam-se assim três destas soluções, sendo elas o Intel AIM, InsideVisions e EXVA Technologies. Na secção seguinte apresenta-se uma descrição mais detalhada de cada uma.

2.3.3.1 Intel AIM

O AIM (*Audience Impression Metrics*)³¹ é uma aplicação proprietária da Intel que, através da utilização de câmaras e da tecnologia de Reconhecimento Facial, deteta pessoas no seu campo de visão.

Isto é conseguido através da utilização de um algoritmo também proprietário da Intel (AIM *Suite Audience Counter*) que, além de detetar pessoas, fornece várias métricas tais como tempo de visualização e informação demográfica (faixa etária e sexo). Todos estes dados são recolhidos de uma forma anónima, uma vez que não é feito reconhecimento facial, ou seja, o algoritmo apenas deteta faces e não associa nenhuma delas a uma pessoa em específico. A taxa de precisão de deteção de fatores demográficos desta ferramenta encontra-se à volta dos 80%, contribuindo para este número fatores como a iluminação do local, assim como o posicionamento da câmara. (Solution Providers For Retail, 2013)

Atualmente, o AIM pode ser instalado tanto em equipamentos com Windows como em equipamentos com Linux, mais especificamente Ubuntu e Red Hat Enterprise. Neste caso em concreto, foi utilizado Windows.

2.3.3.1.1 Requisitos de Hardware

A utilização do AIM está dependente de dois aspetos importantes no que diz respeito ao *hardware* utilizado: o computador propriamente dito e a câmara.

No que diz respeito à câmara, esta pode ser de vários tipos:

- USB ou FireWire com suporte a DirectShow, em sistemas Windows
- USB UVC com suporte a Video 4, em sistemas Linux
- Axis IP Camera com suporte ao protocolo VAPIX
- Câmaras IP com suporte ao protocolo Panasonic

Um aspeto importante é o facto de qualquer câmara que respeite estes protocolos ser compatível com o sistema, o que permite a aquisição de câmaras bastante baratas.

Por sua vez, em termos de *hardware* do próprio computador, os requisitos mínimos passam por:

- CPU Intel Core i3 2.4GHz
- 2GB RAM para Windows XP, Ubuntu e Red Hard; 4GB RAM para Windows Vista / 7
- 20GB Livres em disco

Como se pode verificar, tratam-se de requisitos não muito exigentes, sendo apenas necessário dar especial destaque ao processador utilizado. Isto deve-se ao facto de o AIM se encontrar em constante monitorização e processamento, o que exige mais deste componente.

³¹ <https://aimsuite.intel.com/>

2.3.3.1.2 Disponibilização de informação

Uma das vantagens da utilização do AIM é o facto de este disponibilizar a informação capturada para aplicações de terceiros a consultarem. Na Tabela 6 encontra-se uma descrição destes atributos disponibilizados.

Tabela 6 – Atributos retornados pela API do Intel AIM

Parâmetro	Descrição
Magic Word	Código de validação
Version	Versão da API em utilização
Type	Tipo de informação retornada pela API
Payloadsize	Tamanho da mensagem
StartStop	Indica se um visitante entrou ou saiu do campo de deteção
Gender	Sexo
Vtime	Tempo de visualização
Age	Faixa etária
X	Coordenada X do canto superior esquerdo da face do visitante detetado
Y	Coordenada Y do canto superior esquerdo da face do visitante detetado
FW	Largura da face do visitante detetado
FH	Altura da face do visitante detetado

Esta disponibilização para aplicações de terceiros é feita através de duas maneiras possíveis:

- API HTTP
- API Socket TCP/IP

Com esta informação relativa aos visitantes, torna-se assim possível integrá-la em aplicações desenvolvidas por terceiros, o que vai de encontro ao pretendido no desenvolvimento a ser descrito nos capítulos seguintes.

2.3.3.2 InsideVisions

A InsideVisions³² é uma empresa fundada em 2009 em Coimbra, Portugal.

Atualmente, a InsideVisions possui dois produtos, ambos relacionados com a tecnologia de reconhecimento facial e vocacionados para a área do retalho. São eles os descritos de seguida.

2.3.3.2.1 Faceclick

O Faceclick permite, tal como o Intel AIM, realizar a captura de informação relativa aos visitantes das lojas através de reconhecimento facial. São retornadas as seguintes variáveis:

- Faixa etária
- Sexo
- Número de pessoas detetadas

Com uma eficácia de 95%, o Faceclick disponibiliza aos clientes finais uma *Dashboard* onde é possível visualizar esta informação de uma forma apelativa.



Figura 25 – InsideVisions Faceclick³³

³² <http://insidevisions.com/>

³³ Retirada de <http://insidevisions.com/site/faceclick>

2.3.3.2 Targetads

Desenvolvido pela InsideVisions, também o Targetads utiliza a tecnologia de reconhecimento facial, sendo que neste caso o utilizador final é a própria pessoa que se encontra a ser detetada pelo sistema em determinado momento.

Com o objetivo de personalizar a publicidade para cada grupo demográfico, este produto baseia-se em câmaras instaladas em painéis publicitários. Após ser efetuado o reconhecimento facial de quem se encontra em frente ao painel, sendo que neste caso é detetado grupo etário, sexo, tempo de visualização e expressão facial, é então exibido um anúncio adequado ao grupo demográfico em questão.



Figura 26 – InsideVisions Targetads³⁴

2.3.3.3 EXVA Technologies

A EXVA³⁵ é uma empresa com origem na Universidade do Minho, fundada em 2008, sendo a sua área de atuação o processamento e análise de vídeo, utilizando para isso várias técnicas e algoritmos avançados.

O produto mais relevante para a presente dissertação é o Adnamic. Trata-se de uma plataforma de obtenção de vários dados relativos a visitantes, tais como Idade, Sexo ou Emoções.



Figura 27 – EXVA Adnamic

³⁴ Retirada de <http://insidevisions.com/site/targetads>

³⁵ <http://www.exva.pt/>

De acordo com a própria empresa, esta tecnologia pode ser aplicada, por exemplo, em:

- Shoppings
- Hipermercados
- Aeroportos
- Lojas de Retalho
- Museus e Exposições
- Empresas de Estudo de Mercado

3 Análise

Este capítulo diz respeito a toda a componente de análise, conceção e desenho da solução a ser desenvolvida.

Uma vez feito o estudo do Estado da Arte, começa-se assim por descrever em detalhe o problema presente. Será primeiramente definido qual o objetivo a atingir, enquadrando-o na necessidade da empresa no momento em que foi realizada a presente dissertação, e quais os benefícios que a sua realização trará, tanto para o cliente final como para a InovRetail.

Serão também analisadas as várias abordagens possíveis de resolução do problema, sendo apresentadas tanto as vantagens como as desvantagens de cada uma, assim como as causas que levaram à escolha de uma destas abordagens em específico.

Por fim, será descrito todo o trabalho de conceção, como por exemplo a arquitetura da solução e os vários componentes que esta contém.

Pretende-se assim definir os passos que o desenvolvimento em si deve seguir, de forma a ser obtida uma solução estruturada e que possa ser facilmente reutilizada no futuro.

3.1 *Objetivos*

A solução implementada diz respeito ao desenvolvimento de um novo módulo do SeePlus, produto da InovRetail, com o nome de SmartScan, sendo que este fez parte da realização de um piloto implementado num dos clientes da InovRetail.

Grande parte das lojas dos retalhistas de diversas áreas possuem atualmente sistemas de contadores de visitantes. No entanto, estes contadores apenas disponibilizam o número de pessoas que entram na loja. Com a implementação do SmartScan, pretende-se assim

complementar estas contagens com outro tipo de informação mais detalhada, procurando-se assim responder, por exemplo, às seguintes questões:

- Qual a percentagem de mulheres que visitam a loja 1 aos fins-de-semana?
- Qual o grupo demográfico com maior predominância durante a manhã? E durante a tarde? Qual o seu impacto na taxa de conversão?

O objetivo passou então pela instalação de novo equipamento nas lojas do cliente, paralelamente aos sistemas de contadores já existentes, de forma a ser possível obter dados demográficos dos visitantes das suas lojas. Posteriormente, através da instalação do Intel AIM (programa proprietário da Intel descrito no capítulo anterior), foi possível obter esta nova informação, sendo esta de seguida carregada no Armazém de Dados actualmente existente.

Uma vez que o AIM disponibiliza dados demográficos sobre os visitantes das lojas do cliente, isto permite a realização de relatórios com esta informação, não existente até ao momento. Após a realização do piloto, esta solução fará parte integrante do SeePlus, ficando assim a InovRetail com mais um módulo que poderá ser adaptado e comercializado a outros potenciais clientes.

3.2 Abordagens

De forma a desenvolver uma solução para o problema, foram analisadas duas abordagens possíveis.

3.2.1 Abordagem 1

A primeira abordagem diz respeito ao desenvolvimento de todos os processos envolvidos desde a captura de imagem até à disponibilização de ferramentas de exploração de dados. Com esta abordagem, seria portanto necessário criar algoritmos de deteção facial, algo que, após uma breve análise, revelou ser bastante complexo, implicando um custo inicial elevado, tanto em termos monetários como no que diz respeito ao cumprimento de prazos.

A grande vantagem passaria por uma maior autonomia. Uma vez que esta tecnologia de reconhecimento facial seria proprietária, isto traduzir-se-ia numa redução de custos a longo prazo. Além disso, e devido a esta autonomia, seria possível alterar o algoritmo de deteção consoante as necessidades do cliente.

3.2.2 Abordagem 2

Por sua vez, a segunda abordagem diz apenas respeito ao desenvolvimento dos componentes responsáveis por gerir a informação.

Isto implica a aquisição a fornecedores externos de ferramentas de deteção facial, aliviando assim o esforço necessário nesta área por parte da InovRetail. Apenas seria necessário que estas ferramentas proporcionassem uma forma de interação, com o objetivo de fornecerem a informação por elas capturada.

Esta abordagem tem a vantagem de utilizar tecnologia proprietária já existente de empresas cuja especialidade é, de facto, o reconhecimento facial, proporcionando assim uma maior robustez na sua eficácia.

No entanto, isto leva a uma perda de autonomia, uma vez que se fica dependente de um fornecedor, e a um maior custo a longo prazo.

Uma vez que a área de experiência da InovRetail diz respeito à disponibilização de plataformas de apoio na área de gestão de informação, não fazia sentido investir no desenvolvimento de raiz de uma tecnologia de reconhecimento facial, especialmente quando o objetivo a curto prazo passa pela realização de um piloto. Por isso mesmo, optou-se pela segunda abordagem, ou seja, pela aquisição de uma solução que efetue o reconhecimento e apenas forneça a informação daí resultante, sendo os algoritmos do reconhecimento facial em si transparentes para a InovRetail.

Como referido na secção 2.3.3, encontram-se atualmente em mercado vários produtos que implementam estes algoritmos de reconhecimento facial. Neste caso em específico, foram postos como hipótese de utilização neste piloto as seguintes tecnologias de reconhecimento facial proprietárias:

- Intel AIM
- EXVA Technologies

Após pormenorizada análise de ambas as soluções, a decisão passou pelo recurso ao Intel AIM dadas as características *plug-and-play* do mesmo, o que permite um reduzido período de desenvolvimento. Esta solução conta ainda com suporte por parte da Intel, o que é uma mais-valia. Além disso, o facto de praticamente qualquer câmara poder ser utilizada é outro fator a favor deste *software*, uma vez que permite a aquisição de equipamentos com baixo custo.

No decorrer do desenvolvimento surgiu no entanto o conhecimento de outro produto disponível no mercado, denominado Faceclick, proprietário da InsideVisions. Esta solução será equacionada em futuras implementações do módulo SmartScan.

3.3 Desenho da Solução

3.3.1 Arquitetura

Optando-se então pelo Intel AIM, pretende-se nesta secção apresentar a arquitetura do sistema.

A solução implementada é composta por uma infraestrutura baseada na *cloud*, assim como nos pontos de venda físicos do cliente. Isto permite uma maior flexibilidade tanto para o utilizador como para a própria implementação do sistema.

O objetivo passa por capturar dados através do Intel AIM e armazená-los num Armazém de Dados (AD) já existente. Desta forma, torna-se possível por parte dos utilizadores a utilização de ferramentas que permitam aceder a estes dados e efetuar análises de negócio.

Apresenta-se de seguida um diagrama da arquitetura, ao qual se segue uma descrição dos diversos componentes e das suas interligações.

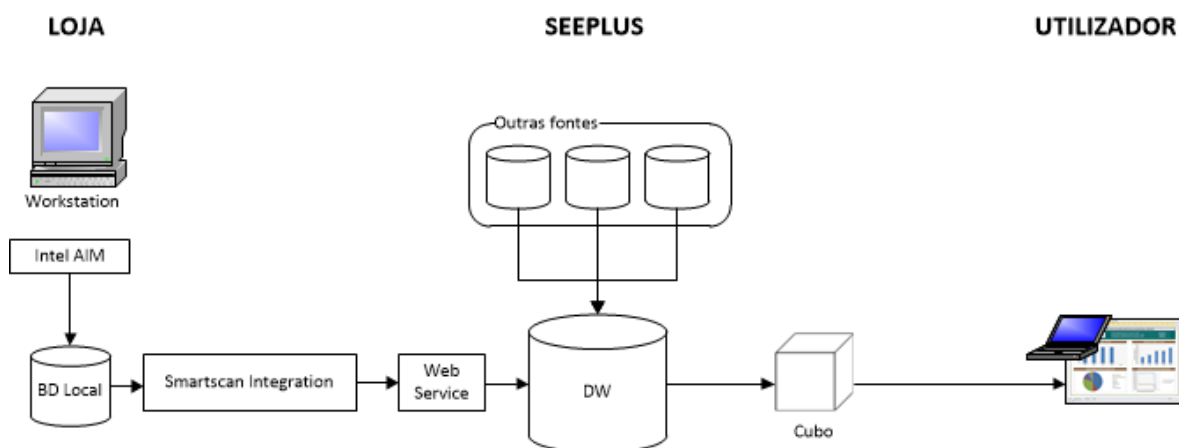


Figura 28 – Diagrama de arquitetura

Como se pode ver pelo diagrama, a infraestrutura é composta por três grandes áreas, descritas de seguida.

3.3.2 Componente Lojas

A área das lojas diz respeito às lojas do cliente e contém equipamentos com o Intel AIM instalado, cuja função passa por detetar informação demográfica sobre os seus visitantes e inseri-la numa BD local. Com estes dados armazenados, é possível enviá-los para uma estrutura central, o Armazém de Dados (parte integrante do SeePlus). Uma vez que o cliente

pode ter lojas em qualquer parte do mundo, esta trata-se da área com maior abrangência geográfica de toda a arquitetura. Se se tratar de uma multinacional com atuação em diversos países, esta componente espalhar-se-á eventualmente a uma escala global.

Estes equipamentos são colocados um em cada loja e a sua função é, além de efetuar a detecção dos visitantes, realizar o envio desta informação para o AD. São compostos por PCs convencionais com ligação à Internet e equipados com uma *Webcam* direcionada para a entrada da loja.

Através do Intel AIM, são adquiridos dados em tempo real dos visitantes que se encontram na área de captação da camera, cuja informação demográfica é detetada com cerca de 80% de eficácia (Solution Providers For Retail, 2013) e, através do Socket TCP/IP disponibilizado por este programa, é possível capturar esta informação, que será então inserida numa Base de Dados (BD) ainda no equipamento presente na loja.

Uma vez que este processo é referente a cada um dos equipamentos, ou seja, está presente em cada uma das lojas do cliente, a informação daí resultante pode não ter relevância de uma forma isolada e fora de contexto, pelo que é necessário proceder à sua centralização numa plataforma única que cruze vários domínios de dados, o Armazém de Dados.

3.3.2.1 Base de Dados local

De forma a armazenar os dados disponibilizados pela API do Intel AIM, foi modelada uma base de dados em SQL Server a ser instalada localmente em cada loja (denominada no presente documento de “Base de Dados Local”), idealmente na mesma máquina que possui instalado o AIM, por questões de desempenho.

A Figura 28 apresenta o respetivo modelo de dados.

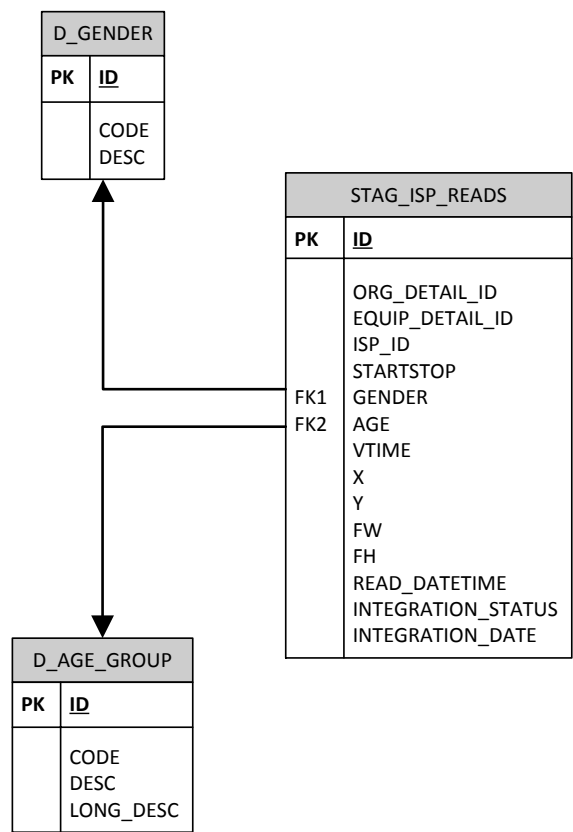


Figura 29 – Modelo de dados da Base de Dados Local

Além destas tabelas, o modelo da Base de Dados Local possui outras adicionais cuja finalidade diz respeito a outros módulos. Assim, apenas as descritas acima serão objeto de análise.

Como se pode verificar, o modelo contém uma tabela central e duas ligadas a esta. Segue-se uma breve descrição de cada uma, assim como dos atributos considerados mais relevantes.

3.3.2.1.1 D_GENDER

Diz respeito ao sexo de quem se encontra no raio de captação da câmara. Existem 3 valores possíveis retornados pela API do Intel AIM, sendo eles:

- 0 - Desconhecido
- 1 - Masculino
- 2 - Feminino

Assim, a tabela é preenchida com estes valores, de forma a ter registos em comum com a tabela central quando esta é preenchida com informação proveniente da API.

ID	CODE	DESC
0	0	UNKNOWN
1	1	MALE
2	2	FEMALE

Figura 30 – Conteúdo da tabela na Base de Dados Local, D_GENDER

3.3.2.1.2 D_AGE_GROUP

Contém informação relativa à faixa etária, organizada por intervalos. A API retorna os seguintes valores:

- 0 – AGE_UNKNOWN
- 1 – Child
- 2 – Teenager
- 3 – Young Adult
- 4 – Older Adult
- 5 – Senior

O valor “2 – Teenager”, diz respeito a um intervalo de idades que, de acordo com a documentação do Intel AIM, atualmente não é disponibilizado pela API. Foi, no entanto, inserido na tabela, para o caso de no futuro vir a ser disponibilizado.

Tal como na tabela relativa ao sexo (D_GENDER), esta também é preenchida com estes mesmos valores de forma a ter registos em comum no momento do preenchimento da tabela central.

ID	CODE	DESC	LONG_DESC
0	0	AGE_UNKNOWN	AGE_UNKNOWN
1	1	CHILD	AGE_UNDER_16
2	2	TEENAGER	AGE_BETWEEN_13_AND_19
3	3	YOUNG_ADULT	AGE_BETWEEN_16_AND_34
4	4	OLDER_ADULT	AGE_BETWEEN_35_AND_64
5	5	SENIOR	AGE_UPPER_65

Figura 31 - Conteúdo da tabela na Base de Dados Local, D_AGE_GROUP

3.3.2.1.3 STAG_ISP_READS

É nesta tabela central do modelo onde vai ser inserida a informação proveniente do Intel AIM e que diz respeito aos dados a serem analisados posteriormente. Isto é possível uma vez que, no momento em que é feito este armazenamento, é inserida informação relativa ao período em que foi capturada, na forma de *Timestamp* (coluna READ_DATETIME).

Uma vez que se trata da tabela mais relevante, sem a qual não existiria qualquer métrica ou mesmo solução final, segue-se uma descrição de todos os seus atributos:

- ID: Id único do registo, gerado automaticamente
- ORG_DETAIL_ID: Id da loja. Cada BD Local (pertencente a cada loja) terá apenas um valor possível
- EQUIP_DETAIL_ID: Id do equipamento físico.
- ISP_ID: Id retornado pela API da Intel
- STARTSTOP: Indica se alguém começou a ser detetado no raio de ação da câmara (valor = 0) ou se alguém deixou de ser detetado (valor = 2)
- GENDER: Sexo do visitante detetado. Possui um dos 3 valores acima descritos e, tal como referido anteriormente, desta forma é possível efetuar a ligação entre esta tabela e a D_GENDER.
- AGE: Faixa etária do visitante detetado. Possui um dos 6 valores acima descritos e, tal como referido anteriormente, desta forma é possível efetuar a ligação entre esta tabela e a D_AGE_GROUP
- VTIME: Tempo de visualização
- X: coordenada X do canto superior esquerdo da face do visitante detetado
- Y: coordenada Y do canto superior esquerdo da face do visitante detetado
- FW: Largura da face do visitante detetado
- FH: Altura da face do visitante detetado
- READ_DATETIME: Data e Hora da leitura. Este atributo é fundamental para o armazenamento de informação histórica. Existe um valor *default* configurado com a data e hora corrente.
- INTEGRATION_STATUS: Atributo utilizado como forma de controlo. Serve para identificar os registos que já se encontram carregados no Armazém de Dados (descrito num capítulo seguinte). Possui o valor *default* 'R'.
- INTEGRATION_DATE: Atributo apenas utilizado de forma a registar a hora em que o registo foi inserido na BD.

Na figura 32, é possível visualizar um exemplo de alguns registos presentes nesta tabela.

ID	ORG_DETAIL_ID	EQUIP_DETAIL_ID	ISP_ID	STARTSTOP	GENDER	AGE	VTIME	X	Y	FW	FH	READ_DATETIME	INTEGRATION_STATUS	INTEGRATION_DATE
77300	58	13	1	0	2	3	112	164	249	254	169	2014-09-20 07:19:26.750	C	2014-09-20 07:19:26.750
77301	58	13	1	2	2	4	23	358	153	254	169	2014-09-20 07:19:35.903	C	2014-09-20 07:19:35.903
77302	58	13	2	0	2	5	128	214	286	254	169	2014-09-20 07:19:51.340	C	2014-09-20 07:19:51.340
77303	58	13	2	2	2	5	240	203	231	254	169	2014-09-20 07:19:53.480	C	2014-09-20 07:19:53.480

Figura 32 - Conteúdo da tabela na Base de Dados Local, STAG_ISP_READS

Uma vez que o número de registos nesta tabela aumenta rapidamente ao longo do dia, foi necessário criar um processo de purga. Todos os dias ao início da manhã, antes de a loja abrir,

é feita a limpeza dos dados que já foram carregados para o Armazém de Dados (secção seguinte) e, portanto, já são considerados redundantes nesta BD Local.

3.3.3 Componente Armazém de Dados

O SeePlus, como foi referido anteriormente, é um dos produtos da InovRetail que permite a exploração de informação de negócio armazenada em sistema.

Sendo que tem por base a utilização de um Armazém de Dados, é neste que deverão estar presentes os dados a serem analisados. Estes podem ter origem em vários sistemas, sendo a informação mais relevante para esta dissertação apenas a proveniente do Intel AIM, ou seja, os dados demográficos dos visitantes. No entanto, é possível cruzar estes dados com outras áreas de negócio, como por exemplo Vendas, de forma a criar relatórios que analisem o seu impacto na performance da loja.

O AD atualmente existente tem por base Azure³⁶, uma tecnologia de *Cloud Computing* da Microsoft. A utilização desta tecnologia tem várias vantagens, sendo que a mais relevante para a realização deste projeto é o próprio facto de a base de dados se encontrar acessível a um nível global sem grande esforço na configuração de acessos.

Isto permite que, no caso de clientes com lojas em vários locais no mundo, cada uma delas apenas necessite de uma ligação à Internet de forma a enviar os dados capturados pelas câmaras para o Armazém de Dados.

Uma vez consolidada a informação de todas as lojas no SeePlus, é então possível ao utilizador final efetuar as análises pretendidas.

De especial importância para a presente dissertação, existem dois aspetos a serem abordados:

- Definição de tabelas a serem criadas no AD
- Modelo de dados do AD composto por estas novas tabelas, assim como por outras já existentes e que venham a ser utilizadas pelo SmartScan.

3.3.3.1 Tabelas criadas

No total foram necessárias três novas tabelas no Armazém de Dados, relativas ao desenvolvimento do SmartScan:

- D_GENDER
- D_AGE_GROUP
- STAG_SMARTSCAN_READS

As duas primeiras são relativas às dimensões Sexo e Faixa Etária e são em tudo semelhantes às descritas na secção anterior, relativa à BD Local.

³⁶ <http://azure.microsoft.com/pt-pt/>

A STAG_SMARTSCAN_READS trata-se da tabela de factos e, apesar de partilhar muitas semelhanças com a tabela presente na Base de Dados Local, existem ligeiras diferenças. Situa-se atualmente numa *Staging Area* do Armazém de Dados, passando por trabalho futuro/melhoria a criação de um fluxo com destino numa tabela final. Segue-se uma descrição dos seus atributos:

- ID: Id automático atribuído a cada novo registo
- NODE_ID: Id do equipamento da loja
- ISP_ID: Id retornado pela API da Intel
- STARTSTOP: Indica se alguém começou a ser detetado no raio de ação da câmara (valor = 0) ou se alguém deixou de ser detetado (valor = 2)
- GENDER: Sexo do visitante
- AGE: Faixa etária do visitante
- VTIME: Tempo de visualização
- X: coordenada X do canto superior esquerdo da face do visitante detetado
- Y: coordenada Y do canto superior esquerdo da face do visitante detetado
- FW: largura da face do visitante detetado
- FH: altura da face do visitante detetado
- READ_DATETIME: Data e Hora da leitura. Existe um valor *default* configurado com a data e hora corrente. No entanto, este apenas será usado em último caso, ou seja, se a informação proveniente da loja vier a Null.
- INTEGRATION_STATUS: Atributo de controlo em relação à consolidação de cada registo. Tal como na Base de Dados Local, também aqui existe este atributo, mas numa vertente de uma consolidação posterior referenciada como ponto de melhoria. Valor *default* = "R"
- INTEGRATION_DATE: Atributo apenas utilizado de forma a registar a hora em que o registo foi inserido na Armazém de Dados

3.3.3.2 Modelo de Dados Armazém de Dados

No Capítulo 4, na secção relativa à construção da ferramenta de exploração da informação, serão referidas várias *views* com base em tabelas existentes no Armazém de Dados. De forma a ajudar à sua compreensão, torna-se assim necessário apresentá-las. Uma vez que o Armazém de Dados atualmente possui uma grande quantidade de tabelas, encontram-se representadas na imagem seguinte apenas as relevantes, ou seja, as que são utilizadas por estas *views*, sendo que já se encontram aqui representadas as novas tabelas definidas na secção anterior.

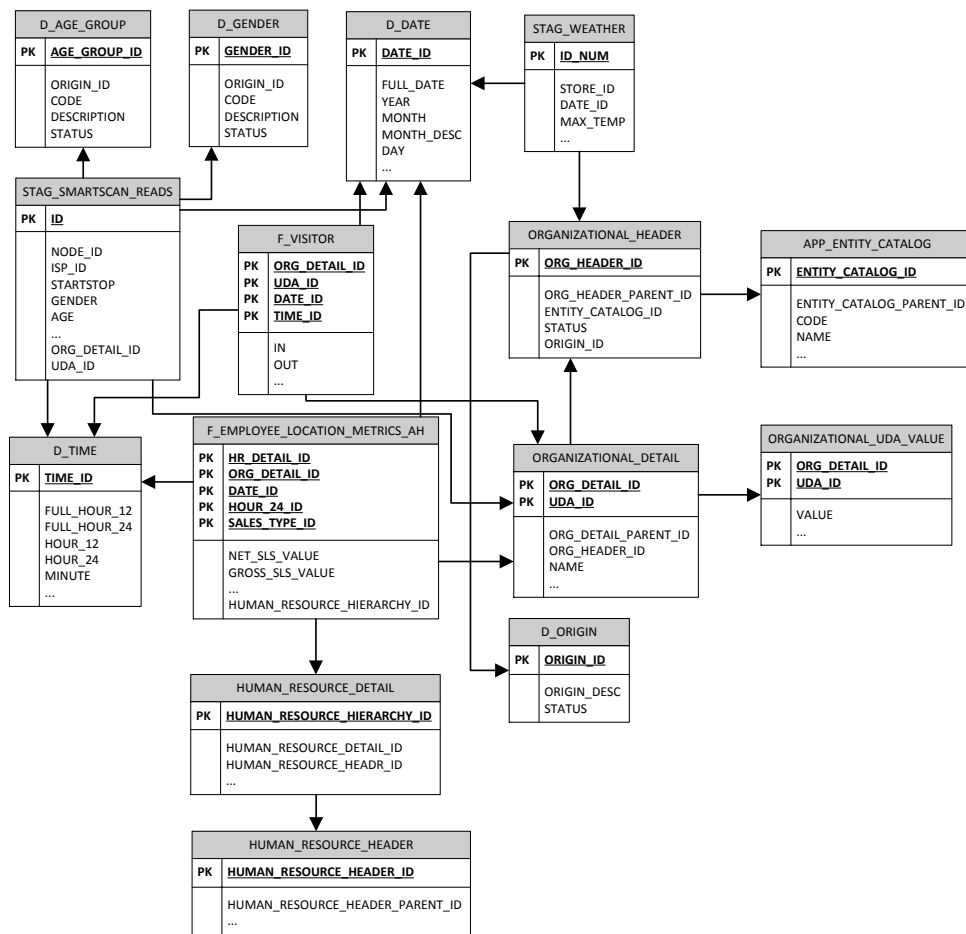


Figura 33 – Modelo de dados do Armazém de Dados

Como se pode visualizar, estão incluídas tanto tabelas de factos como tabelas de dimensões. Um aspeto importante é o facto de as dimensões relativas às lojas e aos colaboradores serem compostas por várias tabelas. Isto deve-se ao facto de ser assim possível criar hierarquias customizáveis para cada cliente, o que permite, por exemplo, que a estrutura organizacional de determinado cliente seja composta por quatro níveis, ao passo que a de outro seja composta por apenas dois.

3.3.4 Componente Utilizador

O objetivo final da solução implementada resume-se à disponibilização de informação sobre o negócio, com garantias de qualidade, assim como ferramentas que permitam que esta seja explorada por parte do utilizador.

Tendo isto em consideração, foi criado um cubo de exploração de dados ao qual o utilizador pode aceder e realizar análises *ad-hoc*. Este cubo tem por base a tecnologia Analysis Services da Microsoft e permite um fácil manuseamento, em grande parte devido ao facto de a forma

de acesso mais frequente passar pela utilização do Excel, programa muito utilizado por analistas. Desta forma, é também possível incluir estas análises em ficheiros já existentes e cruzar informação.

Com isto, os utilizadores conseguem criar relatórios que cruzem vários domínios de dados, tais como vendas e orçamentos. Assim, têm a possibilidade de visualizar o desempenho da organização ao longo do tempo, utilizando também estes novos atributos relativos aos dados demográficos dos seus visitantes.

Fortemente ligada aos resultados obtidos após o desenvolvimento do SmartScan, esta componente relativa ao utilizador final não tem propriamente uma vertente técnica na área da Engenharia Informática. Por isso mesmo, não é objeto de análise, sendo mais abordada no Capítulo 5 relativo aos resultados obtidos, onde são dados alguns exemplos de relatórios possíveis de serem criados.

4 Desenvolvimento

Após ter sido feita uma análise e desenho conceptual da solução a ser implementada, este quarto capítulo aborda o desenvolvimento em si.

É efetuada uma divisão pelas três grandes componentes da solução:

- Loja
- Armazém de Dados
- Utilizador

Desta forma, torna-se assim mais fácil para o leitor efetuar o acompanhamento de todo o fluxo de desenvolvimento.

Para cada componente, serão descritos os vários aspetos técnicos deste, tais como todo o código desenvolvido, tanto relativo a projetos em C# como a *views* e procedimentos em SQL, sendo demonstrados, quando aplicável, os resultados após estes serem executados.

Desde a componente da loja, onde é iniciado o processo de captura de imagem, até à componente do utilizador, será assim descrita a solução ao longo de todo o seu fluxo.

4.1 Componente Loja

O processo de aquisição de dados tem início na componente de loja. Tal como referido anteriormente, cada loja possui equipamento composto por um computador convencional equipado com uma câmara e ligação à Internet.

Os principais componentes de *software* aqui presentes são os seguintes:

- Intel AIM

- Base de Dados local (SQL Server)
- Ficheiros executáveis - interface entre o Intel AIM e a BD Local
- Windows Task Scheduler para os ficheiros executáveis.

Segue-se uma descrição detalhada de cada um destes componentes.

4.1.1 Intel AIM

Tal como descrito no capítulo referente ao Estado da Arte, o Intel AIM trata-se do *software* proprietário da Intel adquirido pela InovRetail, responsável pela deteção dos dados demográficos dos visitantes, utilizando para isso tecnologia de reconhecimento facial.

De um ponto de vista técnico, trata-se de um programa instalado e executado após o arranque da máquina. A respetiva instalação encontra-se fora de âmbito da presente dissertação, sendo esta mais focada no processo de aquisição dos dados.

O AIM fica assim em constante monitorização através da câmara e, caso detete um visitante na sua área de captação, disponibiliza os atributos descritos na secção 2.3.3.1.2. Estes são capturados através da *Socket-based API*, que permite a sua aquisição de uma forma simples e eficaz.

Uma vez que estes dados são disponibilizados em tempo real, torna-se necessário armazená-los na Base de Dados Local descrita no Capítulo 3, de forma a ser possível trabalhar com eles. No entanto, é necessário construir uma interface que sirva de interligação entre estes dois componentes, sendo esta descrita na secção seguinte.

4.1.2 Processos de Carregamento na BD Local

Tendo sido feita a descrição dos dados que a API do Intel AIM disponibiliza através do *Socket*, assim como da Base de Dados Local (nos capítulos 2 e 3 respetivamente), resta descrever os processos responsáveis por este mesmo fluxo de informação, ou seja, pela captura do que o *Socket* disponibiliza e consequente inserção na Base de Dados Local.

Uma vez que este *Socket* disponibilizado pelo AIM fornece dados em tempo real, foi necessário criar pequenos aplicativos que são executados na mesma máquina, cuja finalidade é servir de interface, ou seja, capturar estes dados e inseri-los na BD Local. Têm também outras funções de controlo, como por exemplo o processo de purga da tabela central³⁷ referido anteriormente.

Estes aplicativos foram desenvolvidos em .NET e não possuem qualquer tipo de interface com o utilizador. São processos em background, invocados através do Windows Task Scheduler, de forma a terem o mínimo de impacto no sistema onde se encontram instalados.

³⁷ Tabela do modelo da Base de Dados Local que armazena a informação proveniente do Intel AIM.

Segue-se uma lista dos processos já parametrizados no *Task Scheduler* de uma máquina pertencente a uma loja, em que se pode visualizar o estado actual e a regra de accionamento de cada um:

SmartScanAIM	Running	At system startup
SmartScanClean	Ready	At 7:00 every day
SmartScanIntegration	Queued	At system startup - After triggered, repeat every 5 minutes indefinitely.
SmartScanMonitor	Queued	At system startup - After triggered, repeat every 10 minutes indefinitely.
SmartScanReboot	Ready	At 6:00 every day

Figura 34 – Parametrização dos processos no *Windows Task Scheduler*

No que diz respeito à integração da informação proveniente do AIM na Base de Dados Local, são então dois destes processos os responsáveis por este fluxo:

- SmartScanAIM
- SmartScanClean

Todo este fluxo tem início na tarefa SmartScanReboot, que nada mais é do que a execução de uma instrução na linha de comandos com o objetivo de reiniciar a máquina. Isto é executado ao início da manhã, num horário em que a loja ainda se encontre fechada. Na loja do exemplo anterior, está configurado para as 6:00.

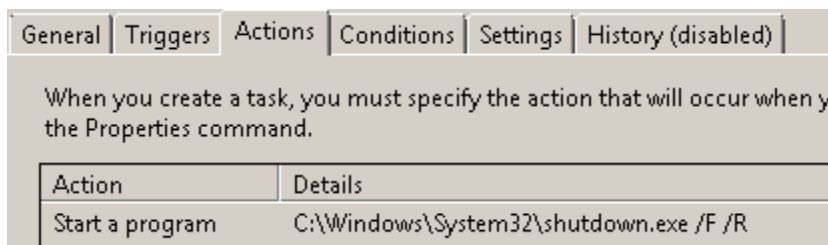


Figura 35 – Tarefa responsável pelo *Reboot* da máquina

Após ser efetuado o *reboot* da máquina, entra então em execução o Intel AIM. Ao mesmo tempo, e como se pode verificar pelo *Task Scheduler*, entra em funcionamento o processo SmartScanAIM, que será descrito de seguida.

4.1.2.1 SmartScanAIM

O SmartScanAIM trata-se do processo responsável pela captura de informação do *Socket* disponibilizado pelo AIM e pelo respetivo carregamento na Base de Dados Local.

Faz parte da solução desenvolvida em .NET e é composta por 2 blocos principais de código: captura e inserção de dados.

O processo começa por estabelecer uma ligação ao *Socket* do AIM:

```
private static string SocketSendReceive(string server, int port)
{
    byte[] request = new byte[6];
    Byte[] response = new Byte[256];
    bool reading = true;

    request[0] = (byte)0xFA; // Magic word (first byte)
    request[1] = (byte)0xCE; // Magic word (second byte)
    request[2] = (byte)0x01; // Version
    request[3] = (byte)MESSAGE_GET_EVENT_VIEWER; // Command
    request[4] = (byte)0x01; // Payload size (0 for apiGetAudienceStatus)
    request[5] = (byte)0x01; // Payload size (0 for apiGetAudienceStatus)

    Byte[] bytesReceived = new Byte[256];

    // Create a socket connection with the specified server and port.
    Socket s = ConnectSocket(server, port);

    if (s == null)
        return ("Connection failed");

    // Send request to the server.
    s.Send(request, request.Length, 0);
}
```

Código 1 – Pedido efetuado ao *Socket* do Intel AIM

A variável `MESSAGE_GET_EVENT_VIEWER` encontra-se definida com o valor 5. Isto indica ao AIM que se pretende receber informação em tempo real, ao invés de apenas uma *snapshot* do momento.

Segue-se então a captura dos dados propriamente ditos. Para tal, é executado um ciclo infinito, de forma a que a aplicação se encontre sempre à escuta. Caso a API retorne informação, esta é guardada em vários atributos de um objeto, para de seguida ser inserida na BD Local.

```
while (reading)
{
    try
    {
        if (s.Receive(response, response.Length, 0) > 0)
        {
            isprecord record = new isprecord();

            record.magicword = response[0].ToString() + response[1].ToString();
            record.version = response[2].ToString();
            record.type = response[3].ToString();
            record.payloadsize = response[4].ToString();
        }
    }
}
```

```

        record.startstop = response[5];
        record.personid = response[6] + response[7] + response[8] + response[9];
        record.gender = response[10];
        record.age = response[11];
        record.vtime = response[14] + response[15] + response[16] +
response[17];
        record.x = response[18] + response[19];
        record.y = response[20] + response[21];
        record.fw = response[22] + response[23];
        record.fh = response[24] + response[25];

        saverecord(record);
    }
}

```

Código 2 – Resposta recebida do *Socket* do Intel AIM

A função responsável por inserir o conteúdo deste objeto na Base de Dados Local é descrita de seguida. A tabela onde esta informação é inserida trata-se da STAG_ISP_READS, tabela central do modelo descrita anteriormente. Nota para o facto de, uma vez que os atributos READ_DATETIME, INTEGRATION_STATUS e INTEGRATION_DATE possuem valores *default*, esta informação não ser incluída no *Insert* propriamente dito.

```

Int16 store_id = Convert.ToInt16(System.Environment.GetEnvironmentVariable("STORE_ID",
EnvironmentVariableTarget.Machine));
Int16 device_id =
Convert.ToInt16(System.Environment.GetEnvironmentVariable("DEVICE_ID",
EnvironmentVariableTarget.Machine));

using (SqlConnection conn = new SqlConnection(cs))
{
    conn.Open();
    using (SqlCommand cmd = new SqlCommand("INSERT INTO dbo.STAG_ISP_READS
([ORG_DETAIL_ID],[EQUIP_DETAIL_ID],[ISP_ID],[STARTSTOP],[GENDER],[AGE],[VTIME],[X],[Y
],[FW],[FH])
VALUES(@org_detail_id,@equip_detail_id,@personid,@startstop,@gender,@age,@vtime,@x,@y
,@fw,@fh)", conn))
    {
        cmd.Parameters.AddWithValue("@org_detail_id", store_id);
        cmd.Parameters.AddWithValue("@equip_detail_id", device_id);
        cmd.Parameters.AddWithValue("@personid", i.personid);
        cmd.Parameters.AddWithValue("@startstop", i.startstop);
        cmd.Parameters.AddWithValue("@gender", i.gender);
        cmd.Parameters.AddWithValue("@age", i.age);
        cmd.Parameters.AddWithValue("@vtime", i.vtime);
        cmd.Parameters.AddWithValue("@x", i.x);
        cmd.Parameters.AddWithValue("@y", i.y);
        cmd.Parameters.AddWithValue("@fw", i.fw);
        cmd.Parameters.AddWithValue("@fh", i.fh);

        int rows = cmd.ExecuteNonQuery();
        conn.Close();
    }
}

```

Código 3 – Inserção na Base de Dados Local dos dados provenientes do Intel AIM

Como se pode verificar pelo código descrito acima, tanto o `store_id` como o `device_id` são atributos cujos valores são lidos de variáveis de ambiente da máquina onde esteja a ser executado. Isto deve-se ao facto de os seus valores serem variáveis de loja para loja, pelo que basta assim efetuar esta configuração (exemplificado na Figura 36). Desta forma, a distribuição desta componente do SmartScan pelas lojas torna-se um processo muito mais célere.

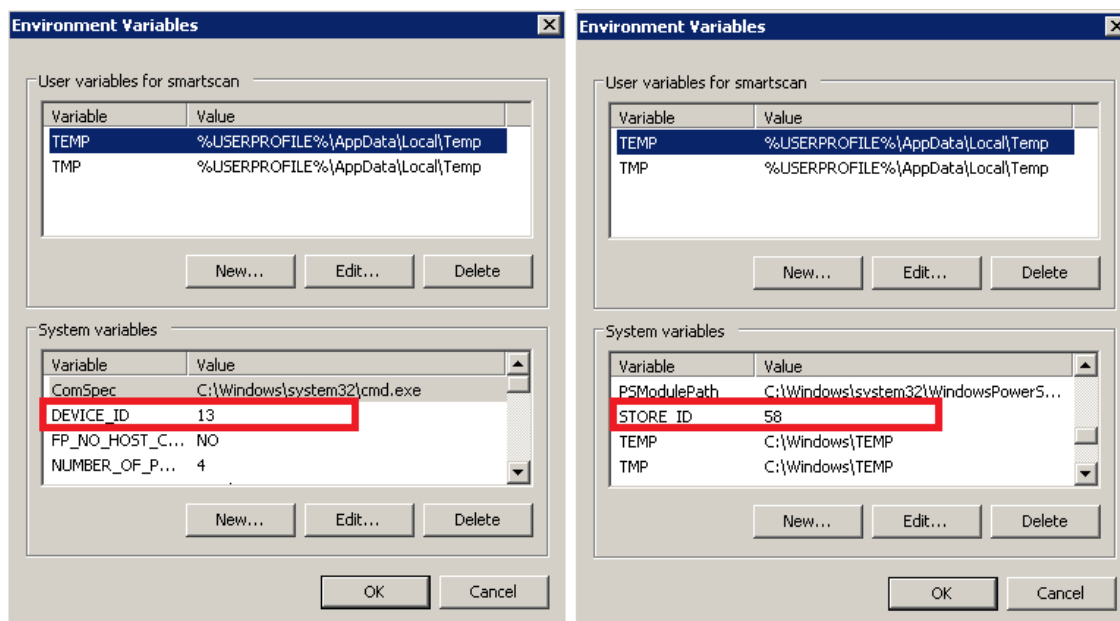


Figura 36 – Configuração das variáveis de ambiente `DEVICE_ID` e `STORE_ID`

Realizado este processo, o resultado final é um novo registo na tabela central da Base de Dados Local. Uma vez que se trata de um ciclo infinito, o processo repete-se cada vez que o `Socket` disponibiliza nova informação.

4.1.2.2 SmartScanClean

Tal como referido anteriormente, existe um processo de purga da tabela central da Base de Dados Local. Isto deve-se ao facto de rapidamente esta tomar grandes proporções.

Uma vez que a informação existente nesta tabela de factos vai sendo periodicamente carregada no Armazém de Dados (secção seguinte), torna-se desnecessário manter um histórico na BD Local.

Todos os dias, antes de a loja abrir, é executado o processo `SmartScanClean`. Assim, garante-se que no início de cada dia a tabela central se encontra vazia.

O processo em si faz parte da mesma solução em .NET descrita anteriormente e limita-se a executar uma operação de Delete diretamente sobre a tabela, como se pode ver no seguinte excerto de código:

```
static void cleanDB(SqlConnection conn)
{
    using (SqlCommand cmd = new SqlCommand("DELETE FROM dbo.STAG_ISP_READS WHERE
INTEGRATION_STATUS = 'C' ", conn))
    {
        using (SqlDataReader reader = cmd.ExecuteReader())
        {}
    }
}
```

Código 4 – Processo de purga da tabela central na Base de Dados local

A cláusula “Where Integration_Status = ‘C’” diz respeito ao carregamento no Armazém de Dados. Este campo possui dois valores possíveis (C ou R), sendo que o valor *default* é R. Quando se encontra com o valor R, significa que o registo apenas está disponível localmente, ou seja, ainda não foi carregado no Armazém de Dados. Quando se encontra com o valor C, significa que já foi carregado.

Tendo isto em conta, com esta condição garante-se que não são apagados registos que ainda não tenham sido carregados no AD.

Após o processo ser executado, obtém-se a tabela STAG_ISP_READS vazia, o que, aliando ao *reboot* programado do equipamento, garante uma performance linear ao longo dos dias. Esta estabilidade no desempenho de cada equipamento é essencial, uma vez que existem outros processos vitais para o negócio a serem executados paralelamente.

4.1.3 Processo de Carregamento no Armazém de Dados

Este subcapítulo aborda o tema do carregamento no Armazém de Dados da informação localizada na BD Local de cada loja. É composto por componentes pertencentes tanto ao lado da loja, como ao lado do Armazém de Dados.

De forma a manter a coerência, será aqui apenas descrita a parte relativa à loja. A componente do AD será explicada no capítulo seguinte.

Encontrando-se já a Base de Dados Local preenchida com informação relativa aos visitantes da loja, o próximo passo passa por carregar estes dados numa estrutura central, o Armazém de Dados. Este vai armazenar informação de todas as Bases de Dados Locais de cada loja, de forma a permitir aos utilizadores do sistema realizarem análises que englobem a totalidade da estrutura organizacional. Tal como em todos os processos até agora descritos, é importante

realçar que também a seguinte descrição do processo relativo ao carregamento no AD, é “replicada” em cada loja.

No *Task Scheduler* da máquina de cada loja, encontra-se a tarefa denominada “SmartScanIntegration”. Esta encontra-se parametrizada para ser executada logo após o arranque do sistema operativo e repete-se de 5 em 5 minutos.

A parte aqui descrita, relativa à componente da loja, faz parte da mesma solução em .NET onde se encontram os restantes processos. É responsável por extrair informação da BD Local e criar uma *string* em formato XML, seguindo-se do seu envio para o Armazém de Dados, comunicando com um *Web Service* que este disponibiliza.

O primeiro passo diz respeito à extração dos últimos 50 registos presentes na tabela central da Base de Dados Local que ainda não tenham sido carregados. Para isso, é utilizado o atributo *INTEGRATION_STATUS*, que deverá possuir o valor R. O motivo para se seleccionar apenas os primeiros 50 registos é apenas de forma a prevenir que o tamanho da mensagem HTTP a ser enviada através da rede não ultrapasse o limite. Isto não constitui nenhum problema, uma vez que os registos que não sejam seleccionados em determinada altura, serão posteriormente, já que o processo de carregamento é executado de 5 em 5 minutos.

Logo de seguida é criada uma *string* em XML com o valor de cada atributo, para cada registo retornado.

Todo este processo está presente no seguinte excerto de código:

```
static string readdata(SqlConnection conn)
{
    string data;
    data = "<READS>";

    using (SqlCommand cmd = new SqlCommand("SELECT TOP(50) * FROM
dbo.STAG_ISP_READS WHERE INTEGRATION_STATUS = 'R' ", conn))
    {
        using (SqlDataReader reader = cmd.ExecuteReader())
        {
            int i = 0;
            while (reader.Read())
            {
                string timeread = reader.GetDateTime(12).ToString("yyyy-
MM-dd HH:mm:ss");

                data += "<READ NODE_ID=\" " + reader.GetInt32(2) + "\"
ISP_ID=\" " + reader.GetInt32(3) + "\" STARTSTOP=\" " +
reader.GetInt32(4) + "\" GENDER=\" " + reader.GetDouble(5)
+ "\" AGE=\" " + reader.GetDouble(6) + "\" VTIME=\" " +
reader.GetInt32(7) + "\" X=\" " + reader.GetInt32(8) +
 "\" Y=\" " + reader.GetInt32(9) + "\" FW=\" " +
reader.GetInt32(10) + "\" FH=\" " + reader.GetInt32(11) +
 "\" READ_DATETIME=\" " + timeread + "\" />";

                ids[i] = reader.GetInt32(0);
                i++;
            }
        }
    }
}
```

```

    }
    total = i;
    data += "</READS>";
    return data;
}
}
data += "</READS>";
return data;
}

```

Código 5 – Criação de string XML

Após este processamento, a *string* em XML tem o seguinte aspeto:

```

<READS>
  <READ NODE_ID="1" ISP_ID="1" STARTSTOP="1" GENDER="1.0" AGE="1.0" VTIME="1" X="1"
  Y="1" FW="1" FH="1" READ_DATETIME="2014-10-07 14:00:00.000" />
  <READ NODE_ID="2" ISP_ID="2" STARTSTOP="2" GENDER="2.0" AGE="3.0" VTIME="2" X="1"
  Y="1" FW="1" FH="1" READ_DATETIME="2014-10-07 14:01:00.000" />
</READS>

```

Código 6 – XML resultado

Nota para o facto de o atributo NODE_ID do XML dizer respeito ao campo EQUIP_DETAIL_ID da tabela, ou seja, o ID do equipamento em questão. Posteriormente, será feito o mapeamento do equipamento à loja correspondente.

Esta *string* já se encontra então no formato correto para ser enviada para o Armazém de Dados. Isto é feito através da rede, utilizando-se como destino um *Web Service* presente no AD. Este, após receber estes dados, será encarregue de inserir a informação no próprio AD.

É importante referir que previamente foi adicionada uma referência do serviço ao projeto em .NET. Segue-se uma imagem desta configuração:

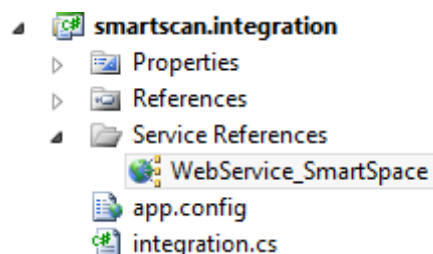


Figura 37 – Referência *Web Service*

O método responsável por realizar então o envio utilizando este *Web Service*, encontra-se descrito de seguida:

```
static int savewebservice( string data )
{
    try
    {
        using (WebService_SmartSpace.SmartSpaceEnvClient service = new
        WebService_SmartSpace.SmartSpaceEnvClient())
        {
            int resul = service.savesmartscan(data);
            Console.WriteLine("WebService result: {0}",resul);
            return resul;
        }
    }
    catch (Exception exp)
    {
        System.Console.WriteLine("Error 2: {0}", exp);
        return -1;
    }
}
```

Código 7 – Envio da *string* XML para o *Web Service*

Por fim, e caso este envio tenha sido concluído com sucesso, apenas resta colocar o atributo `INTEGRATION_STATUS` dos registos enviados, com o valor C, que significa que já foram carregados no Armazém de Dados:

```
static void changestatus( SqlConnection conn )
{
    for (int i = 0; i < total; i++)
    {
        using (SqlCommand cmd = new SqlCommand("UPDATE dbo.STAG_ISP_READS SET
        INTEGRATION_STATUS = 'C' WHERE ID = " + ids[i] + "", conn))
        {
            cmd.ExecuteNonQuery();
        }
    }
}
```

Código 8 – Colocação valor "C" no `INTEGRATION_STATUS`

Assim, e uma vez que esta informação já foi carregada, quando o processo de purga for executado, estas linhas serão apagadas, libertando espaço na Base de Dados Local.

Falta, no entanto, descrever a parte referente à receção desta *string* em XML e consequente inserção no Armazém de Dados. Uma vez que, do ponto de vista da arquitetura, este componente se encontra no lado do AD, será assim descrito na secção seguinte.

4.2 Componente Armazém de Dados

Este subcapítulo aborda a componente central de toda a arquitetura, ou seja, onde se situa o Armazém de Dados.

A informação demográfica capturada em cada loja não tem qualquer significado relevante se for apresentada de uma forma isolada. Assim, é necessário proceder a uma centralização de toda esta informação num único local. Tal como se pode verificar no estado da arte, este trata-se de um dos objetivos da existência de um Armazém de Dados. Por conseguinte, esta componente é focada no carregamento dos dados enviados pelas lojas no AD do SeePlus.

Tendo já sido descritas as tabelas do AD consideradas relevantes, será aqui descrito o processo de carregamento. Por fim, será detalhada a construção de um cubo de exploração de dados, que será acedido pelos utilizadores finais. Este acesso, no entanto, faz parte da última secção da arquitetura do sistema, pelo que neste subcapítulo apenas será abordada a construção do cubo propriamente dito.

4.2.1 Fluxo

Tal como referido na secção relativa à Componente Loja, o processo de carregamento no Armazém de Dados divide-se em duas partes. A parte responsável pelo envio da informação em cada loja já foi descrita, sendo que falta então proceder à análise dos vários processos existentes na componente onde se encontra o AD, ou seja, os responsáveis pela receção destes dados.

Esta componente envolve duas áreas técnicas distintas:

- *Web Service*
- Base de Dados

O *Web Service* tem a função de receber a informação de cada loja e passá-la para o Armazém de Dados. Este, por sua vez, possui tanto as tabelas destino desta informação, como procedimentos responsáveis por tudo o que sejam operações CRUD³⁸.

Segue-se um diagrama mais detalhado sobre esta parte da arquitetura da solução:

³⁸ Create, Read, Update, Delete

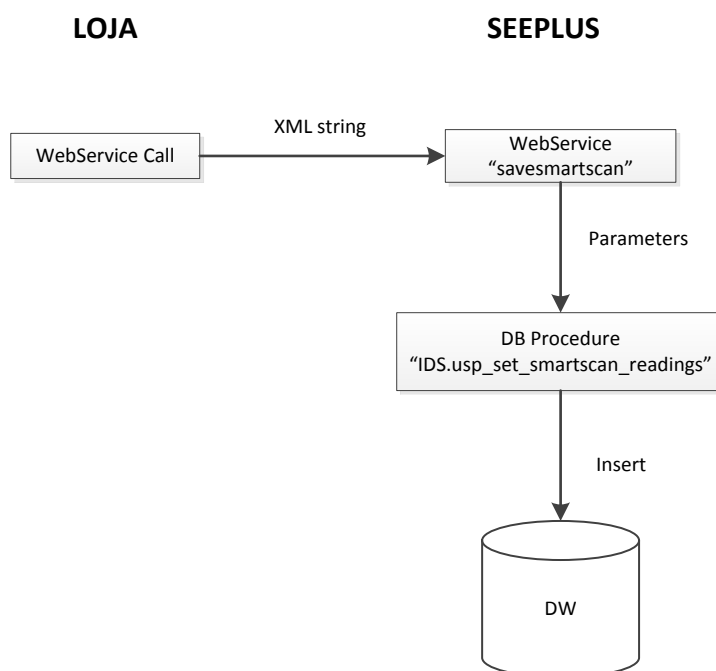


Figura 38 – Processo de carregamento de dados no Armazém de Dados

Uma vez que o primeiro componente, ou seja, o responsável pela chamada ao *Web Service*, já foi descrito anteriormente, será aqui feita uma análise de todo o fluxo imediatamente a seguir a esse ponto.

4.2.1.1 Web Service

O *Web Service* tem a finalidade de receber os dados em formato XML provenientes de cada loja, e de os passar para a Base de Dados do AD propriamente dita.

Ao invés de cada loja inserir diretamente a informação na BD, utiliza-se este *Web Service* como forma de pré-processamento, adicionando-se assim mais esta camada. Isto garante uma maior segurança, além de permitir a manipulação dos dados a serem inseridos, caso venha a ser necessário no futuro. Nestes casos, se não existisse o *Web Service*, seria necessário alterar os processos em todas as lojas, processo esse bastante dispendioso, uma vez que o cliente pode ter lojas a uma escala global.

O projeto em .NET referente a este *Web Service* encontra-se na solução do SeePlus, na secção relativa aos *Web Services* que este disponibiliza. Os métodos lá existentes além dos descritos neste documento não são relevantes, uma vez que fazem parte de outro módulo do SeePlus, o SmartTrack³⁹.

³⁹ Módulo que permite obter informação do percurso dos visitantes dentro de uma loja. Utilizando a tecnologia de Bluetooth e Wifi presente nos telemóveis, efetua assim a triangulação de cada visitante.

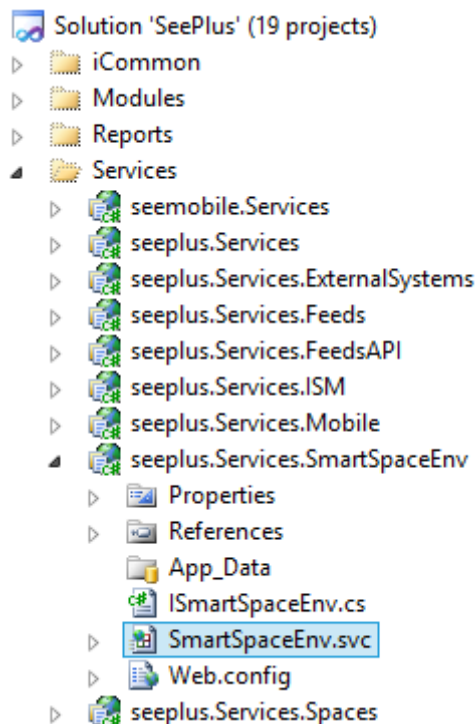


Figura 39 – Web Services disponibilizados pelo SeePlus

A função criada para o efeito recebe então a *string* em XML e invoca um *Stored Procedure* no Armazém de Dados que, este sim, se encarrega de inserir dados na própria BD.

Segue-se o código relativo à declaração do *Web Service*, assim como a definição da referida função.

```
[OperationContract]
    [WebInvoke(
        BodyStyle = WebMessageBodyStyle.Wrapped,
        ResponseFormat = WebMessageFormat.Json,
        RequestFormat = WebMessageFormat.Json)]
    //int savescaning(string store_id, string node_id, string mac_id, string
    read_datetime, decimal read_value_rssi, string type);
    int savesmartscan(string filter);
```

Código 9 – Declaração *Web Service*

```
public int savesmartscan(string filter)
{
    try
    {
```

```

        var connection = ConnectionHelper.CreateConnection();
        var command = ConnectionHelper.CreateCommand(connection,
"[IDS].[usp_set_smartscan_readings]");

        var param = command.Parameters.Add("@p_smartscan_data",
SqlDbType.NText);
        param.Value = filter;

        param = command.Parameters.Add("@oExecResult", SqlDbType.Int);
        param.Direction = ParameterDirection.Output;

        param = command.Parameters.Add("@oExecMessage", SqlDbType.NChar, 999);
        param.Direction = ParameterDirection.Output;
        connection.Open();

        command.ExecuteNonQuery();
        int r = (int)command.Parameters["@oExecResult"].Value;
        connection.Close();

        return r;
    }
    catch
    {
        return -1;
    }
}

```

Código 10 – Definição *Web Service*

Tal como se pode comprovar, são criados vários parâmetros de output. Estes parâmetros serão utilizados pelo *Stored Procedure*, sendo mais detalhados de seguida.

O *Web Service* tem assim a finalidade de servir como uma interface entre as lojas e o Armazém de Dados. Caso o processo tenha sido finalizado com sucesso, é retornado um valor diferente de -1 de volta para a loja que invocou o *Web Service*.

Desta forma, é possível que o processo de cada loja responsável por invocar o *Web Service* verifique se este foi executado com sucesso e, em caso negativo, não colocar o *status* dos registos com o valor "C" (já carregado no AD). É assim garantido que todos os registos sejam carregados com sucesso no Armazém de Dados.

```

if (savewebservice(data) == 1)
{
    Console.WriteLine("Data sent");
    /*Change Integration Status of data submitted*/
    changestatus(conn);
}
else
{
    Console.WriteLine("Error in Webservice!!");
    Environment.Exit(-1);
}

```

```
}
```

Código 11 – *Web Service*, garantia de sucesso na execução (componente lojas)

4.2.1.2 Stored Procedure

Recebendo a informação em XML de cada loja, o *Web Service* não efetua o *Insert* de cada registo diretamente no Armazém de Dados. Em vez disso, reenvia este XML para um *Stored Procedure* existente na Base de Dados.

Este *Stored Procedure*, por sua vez, recebe parâmetros do *Web Service*, sendo que um deles é o XML com a informação a ser inserida, e executa então o *Insert* de cada registo na tabela IDS.STAG_SMARTSCAN_READS.

Segue-se o código do procedimento, incluindo o cabeçalho onde são declarados os parâmetros de *input* e *output*:

```
CREATE PROCEDURE [IDS].[usp_set_smartscan_readings]
(
    @p_smartscan_data          XML
    ,@oExecResult              INT OUTPUT
    ,@oExecMessage             NVARCHAR(MAX) OUTPUT
)
WITH RECOMPILE
AS
BEGIN
    SET NOCOUNT ON

    BEGIN TRY

        INSERT INTO [IDS].[STAG_SMARTSCAN_READS]
            ([NODE_ID]
            , [ISP_ID]
            , [STARTSTOP]
            , [GENDER]
            , [AGE]
            , [VTIME]
            , [X]
            , [Y]
            , [FW]
            , [FH]
            , [READ_DATETIME])
        SELECT
            C.value('(@NODE_ID)[1]', 'INT')
            , C.value('(@ISP_ID)[1]', 'INT')
            , C.value('(@STARTSTOP)[1]', 'INT')
            , C.value('(@GENDER)[1]', 'FLOAT')
            , C.value('(@AGE)[1]', 'FLOAT')
            , C.value('(@VTIME)[1]', 'INT')
            , C.value('(@X)[1]', 'INT')
            , C.value('(@Y)[1]', 'INT')
            , C.value('(@FW)[1]', 'INT')
            , C.value('(@FH)[1]', 'INT')
            , C.value('(@READ_DATETIME)[1]', 'DATETIME')
        FROM
```

```

        @p_smartscan_data.nodes('READS/READ') t(C)

SELECT @oExecResult = 1
        ,@oExecMessage= 'Success!'
        ,@oResult      = NULL

END TRY
BEGIN CATCH
    DECLARE @ExceptionMessage NVARCHAR(max),
            @ErrorSeverity INT = ERROR_SEVERITY(),
            @ErrorState INT = ERROR_STATE(),
            @ErrorProcedure VARCHAR(50) =
                N'[IDS].[usp_set_smarttrack_readings]';--ERROR_PROCEDURE();

    SELECT @oExecResult = 0,
            @ExceptionMessage = [LOG].[fn_get_Formated_Exception_Message]
            (ERROR_MESSAGE(),
            @ErrorProcedure,
            ERROR_LINE(),
            default)

    EXECUTE [LOG].[usp_Create_User_Log_Error] @ErrorProcedure,
            @ExceptionMessage

    SELECT @oExecResult = 0,
            @oExecMessage = @ExceptionMessage

END CATCH

```

Código 12 – Procedimento responsável por inserir dados no Armazém de Dados

O procedimento recebe os seguintes parâmetros:

- p_smartscan_data: Dados a serem inseridos. Encontram-se no formato XML, sendo utilizadas funções já existentes no SQL Server que permitem decodificar facilmente este formato.
- oExecResult: assume o valor 1 no caso de sucesso, ou 0 no caso de insucesso.
- oExecMessage: mensagem informativa do resultado da execução do procedimento. Caso execute com sucesso, apenas assume o valor “Success!”, caso contrário assume a descrição do erro.

Dois destes parâmetros estão configurados como *Output*, ou seja, após o procedimento ser executado, retornam o valor correspondente de cada um para a aplicação que o invocou.

Neste caso, o *Web Service* após receber o parâmetro “oExecResult”, verifica se foi executado com sucesso, e retorna para a loja que o invocou o valor correspondente, tal como descrito anteriormente.

4.2.2 Criação de Cubo de Dados

Após os dados provenientes das lojas se encontrarem no Armazém de Dados, a tabela de factos encontra-se assim preenchida com informação relativa a dados demográficos dos visitantes de cada loja, como se pode verificar na Figura 40.

ID	NODE_ID	ISP_ID	STARTSTOP	GENDER	AGE	VTIME	X	Y	FW	FH	READ_DATETIME	INTEGRATION_STATUS	INTEGRATION_DATE
137250	126	0	0	0	0	0	0	0	0	0	2014-10-06 06:01:48.000	R	2014-10-06 04:06:08.520
137251	13	0	0	0	0	0	0	0	0	0	2014-10-06 06:27:30.000	R	2014-10-06 06:31:08.220
137252	13	17	0	1	1	112	369	185	217	160	2014-10-06 06:50:15.000	R	2014-10-06 06:51:07.980
137253	13	17	2	1	0	210	242	236	217	160	2014-10-06 06:50:18.000	R	2014-10-06 06:51:07.980
137254	14	97	0	1	5	112	366	140	63	197	2014-10-06 07:11:31.000	R	2014-10-06 07:18:30.583
137255	14	97	2	1	5	250	417	140	63	197	2014-10-06 07:11:42.000	R	2014-10-06 07:18:30.583
137266	14	121	0	2	4	124	238	442	61	52	2014-10-06 07:22:00.000	R	2014-10-06 07:28:30.640
137267	14	121	2	2	4	124	340	259	61	52	2014-10-06 07:22:02.000	R	2014-10-06 07:28:30.640
137268	14	125	0	2	3	104	403	209	160	143	2014-10-06 07:23:57.000	R	2014-10-06 07:28:30.640
137269	14	125	2	2	3	104	201	245	160	143	2014-10-06 07:23:59.000	R	2014-10-06 07:28:30.640
137276	14	165	0	0	4	112	232	126	259	149	2014-10-06 07:58:46.000	R	2014-10-06 08:03:30.770

Figura 40 – Tabela DW - STAG_SMARTSCAN_READS

Com esta informação, torna-se assim possível a realização de análises por parte do utilizador final.

A tecnologia utilizada para permitir esta exploração foi o SQL Server Analysis Services (SSAS), da Microsoft. O SSAS permite realizar a construção de cubos de exploração de dados, cujo acesso pode ser feito através do Excel.

A construção do cubo de exploração de dados é composta por quatro partes, descritas de seguida.

4.2.2.1 Views dimensões

A primeira componente essencial diz respeito à sua fonte de dados.

Neste caso a fonte trata-se do Armazém de Dados, sendo necessário fornecer dados ao cubo através de *views*. Isto é aplicável tanto para as suas dimensões, como para os factos propriamente ditos.

Segue-se então a descrição das *views* que utilizam as tabelas do AD descritas no capítulo anterior. Para este cubo em específico foram assim criadas as seguintes *views*:

- Vw_dim_calendar
- Vw_dim_time
- Vw_dim_age
- Vw_dim_gender
- Vw_dim_location

- Vw_vst_demographics
- Vw_vst
- Vw_sls_tkt
- Vw_weather

Optou-se por criar *views*, ao invés de se efetuar leituras diretas às tabelas, uma vez que desta forma é possível apresentar a informação com um formato mais apelativo, fator este importante na navegação no cubo por parte do utilizador final.

A relação entre as *views* pode ser vista no próximo diagrama. A sua estrutura, no que diz respeito à ligação entre *views*, é semelhante à das tabelas, descrita anteriormente.

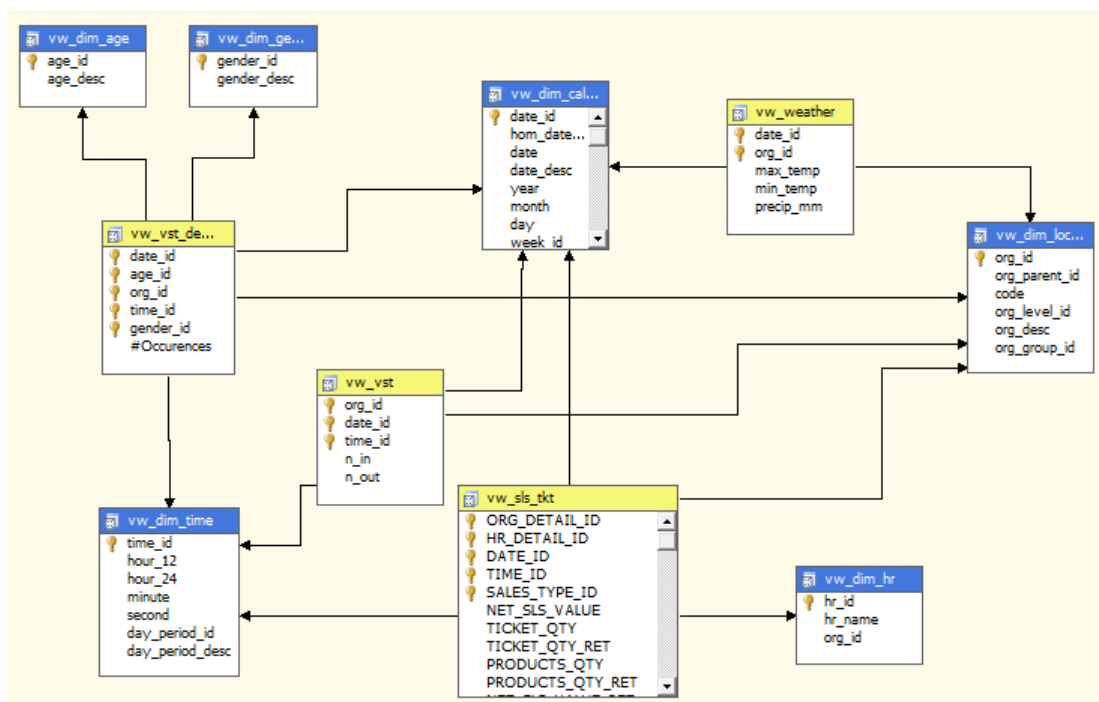


Figura 41 – Diagrama dados do cubo multidimensional

Segue-se uma descrição de cada uma das *views*, assim como do respetivo resultado já no cubo. As *views* vw_dim_hr, vw_vst e vw_weather, apesar de presentes no cubo, não serão descritas, uma vez que não foram desenvolvidas no âmbito da presente dissertação.

4.2.2.1.1 Vw_dim_calendar

Esta *view* é responsável por preencher a dimensão “Data” do cubo. A tabela de origem encontra-se no Armazém de Dados e já foi pré-preenchida.

```

create view [ext_out].[vw_dim_calendar] as
SELECT d.[date_id] as [date_id],
       d.hom_id_last_year_adj as [hom_date_id],
       d.full_date as [date],
       d.week_day_desc as [date_desc],
       d.[year],
       convert(varchar(20),LEFT(d.[MONTH_DESC],3)) ++ '-' ++
convert(varchar(20),d.[YEAR]) month,
       d.[month] month_nr,
       convert(varchar(20),d.DAY) ++ '-' ++
convert(varchar(20),LEFT(d.[MONTH_DESC],3)) ++ '-' ++ convert(varchar(20),d.[YEAR])
day,
       d.[day] day_nr,
       d.week_id,
       convert(varchar(20),LEFT(d.[week_id],4)) ++ '-' ++
convert(varchar(20),RIGHT(week_id,2)) week,
       d.week_day_number,
       d.week_day_desc
from icommon.d_date d

```

Código 13 – View vw_dim_calendar

Esta *view* resulta na seguinte informação:

date_id	hom_date_id	date	date_desc	year	month	month_nr	day	day_nr	week_id	week	week_day_number	week_day_desc
20140404	20130405	2014-04-04	Sexta-feira	2014	Abr-2014	4	4-Abr-2014	4	201414	2014-14	5	Sexta-feira
20140405	20130406	2014-04-05	Sábado	2014	Abr-2014	4	5-Abr-2014	5	201414	2014-14	6	Sábado
20140406	20130407	2014-04-06	Domingo	2014	Abr-2014	4	6-Abr-2014	6	201414	2014-14	7	Domingo
20140407	20130408	2014-04-07	Segunda-feira	2014	Abr-2014	4	7-Abr-2014	7	201415	2014-15	1	Segunda-feira
20140408	20130409	2014-04-08	Terça-feira	2014	Abr-2014	4	8-Abr-2014	8	201415	2014-15	2	Terça-feira
20140409	20130410	2014-04-09	Quarta-feira	2014	Abr-2014	4	9-Abr-2014	9	201415	2014-15	3	Quarta-feira
20140410	20130411	2014-04-10	Quinta-feira	2014	Abr-2014	4	10-Abr-2014	10	201415	2014-15	4	Quinta-feira
20140411	20130412	2014-04-11	Sexta-feira	2014	Abr-2014	4	11-Abr-2014	11	201415	2014-15	5	Sexta-feira
20140412	20130413	2014-04-12	Sábado	2014	Abr-2014	4	12-Abr-2014	12	201415	2014-15	6	Sábado

Figura 42 – Resultado *view* vw_dim_calendar

Ao preencher a dimensão com o conteúdo desta *view*, foram também criadas algumas hierarquias, de forma a que, durante a navegação, seja possível efetuar o *drill-down* e *roll-up*⁴⁰ em alguns atributos. Isto é especialmente importante nas dimensões do tipo Data, uma vez que possuem centenas ou milhares de registos, consoante o pré-carregamento da tabela fonte.

Neste caso em concreto, encontram-se carregados 20 anos, ou seja: 20 x 365 registos = 7300 registos. Do ponto de vista do utilizador, as hierarquias na dimensão Data facilitam assim a sua usabilidade.

Foram criadas duas hierarquias: Date Hierarchy e Week Hierarchy. A primeira contém os seguintes atributos, por ordem crescente de granularidade:

⁴⁰ Capacidade de expandir e colapsar uma estrutura de dados.

- Ano
- Mês
- Dia

Por sua vez, a Week Hierarchy contém os seguintes atributos:

- Semana: Encontrando-se este atributo no formato YYYY-WW
- Dia

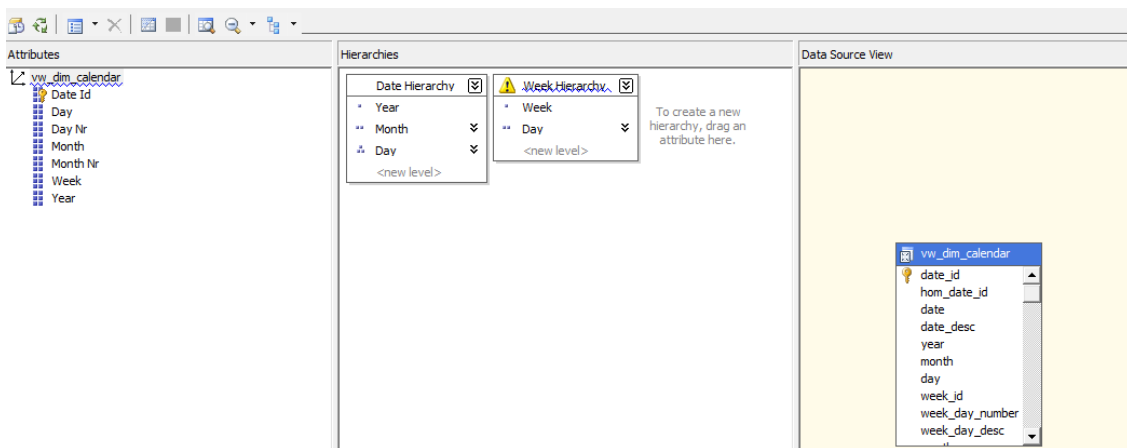


Figura 43 – Hierarquias da dimensão “Data”

Após esta construção, torna-se então possível navegar na dimensão através de cada uma das hierarquias. Na Figura 44, encontra-se expandida a Hierarquia Date até ao nível do dia.

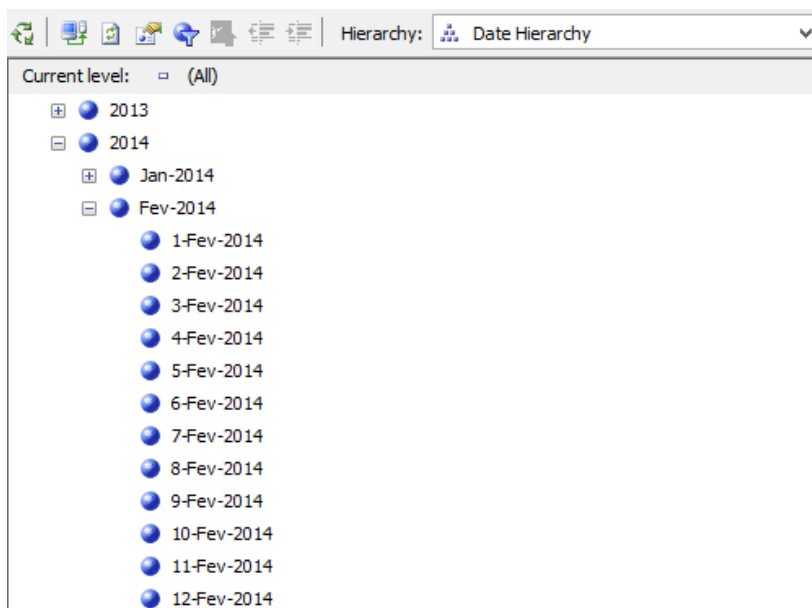


Figura 44 – Navegação dimensão “Data”, hierarquia “Date Hierarchy”

4.2.2.1.2 Vw_dim_time

Esta *view* é responsável por preencher a dimensão “Tempo” do cubo. A tabela de origem encontra-se no Armazém de Dados e já foi pré-preenchida.

```
create view [ext_out].[vw_dim_time] as
select distinct time_id,
    format(hour_12, '00') hour_12,
    format(hour_24, '00') hour_24,
    format(minute, '00') minute,
    format(second, '00') second,
    day_period_id,
    day_period_desc
from iCommon.d_time
```

Código 14 – View vw_dim_time

Esta *view* resulta na seguinte informação:

time_id	hour_12	hour_24	minute	second	day_period_id	day_period_desc
84132	08	08	41	32	1	Dia
84133	08	08	41	33	1	Dia
84134	08	08	41	34	1	Dia
84135	08	08	41	35	1	Dia
84136	08	08	41	36	1	Dia
84137	08	08	41	37	1	Dia
84138	08	08	41	38	1	Dia
84139	08	08	41	39	1	Dia
84140	08	08	41	40	1	Dia
84141	08	08	41	41	1	Dia

Figura 45 – Resultado *view* vw_dim_time

Foi então criada a hierarquia, que neste caso desce até ao nível do segundo, sendo esta a estrutura do resultado final:

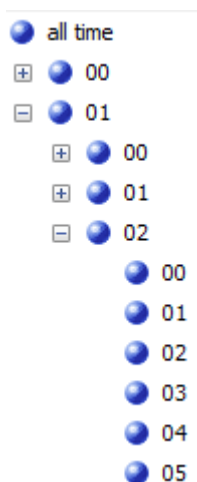


Figura 46 – Navegação dimensão “Tempo”

4.2.2.1.3 Vw_dim_age

Esta *view* é responsável por preencher a dimensão “Faixa Etária” do cubo. A tabela de origem encontra-se no Armazém de Dados e o seu preenchimento está fora do âmbito da presente dissertação.

```
create view [ext_out].[vw_dim_age] as
SELECT convert(float,ag.[CODE]) as age_group_id,
       ag.[Description] as age_group,
       ag.[LONG_DESCRIPTION] age_group_desc
FROM [DATA].[D_AGE_GROUP] ag
```

Código 15 – View vw_dim_age

Esta *view* resulta na informação presente na Figura 47. Tal como referido anteriormente, o registo referente a *Teenager*, não se encontra atualmente a ser disponibilizado pela API do Intel AIM.

age_group_id	age_group	age_group_desc
0	Age Unknown	Age Unknown
1	Child	Age Under 16
2	Teenager	Age Between 13 And 19
3	Young Adult	Age Between 16 And 34
4	Older Adult	Age Between 35 And 64
5	Senior	Age Greater Than 65

Figura 47 – Resultado *view* vw_dim_age

Após o processamento da dimensão, obtém-se os seguintes dados. Neste caso, não existe nenhuma hierarquia, uma vez que se trata de uma estrutura bastante simples, apenas com um atributo.

- Age Unknown
- Child
- Teenager
- Young Adult
- Older Adult
- Senior

Figura 48 – Navegação dimensão “Faixa Etária”

4.2.2.1.4 Vw_dim_gender

Esta *view* é responsável por preencher a dimensão “Sexo” do cubo. A tabela de origem encontra-se no Armazém de Dados e o seu preenchimento está fora do âmbito da presente dissertação.

```

create view [ext_out].[vw_dim_gender] as
SELECT convert(float,g.[CODE]) as gender_id,
       g.[DESCRIPTION] as gender_desc
FROM [DATA].[D_GENDER] g

```

Código 16 – View vw_dim_gender

Esta *view* resulta na seguinte informação:

gender_id	gender_desc
0	Gender Unknown
1	Male
2	Female

Figura 49 – Resultado *view* vw_dim_gender

Tal como na dimensão “Faixa Etária”, também nesta não é necessário criar nenhuma hierarquia, devido à sua baixa complexidade. Sendo assim, obtém-se os seguintes valores possíveis durante a navegação do cubo:

- Gender Unknown
- Male
- Female

Figura 50 – Navegação dimensão “Sexo”

4.2.2.1.5 Vw_dim_location

Esta *view* é responsável por preencher a dimensão “Loja” do cubo. A tabela de origem encontra-se no Armazém de Dados e o seu preenchimento está fora do âmbito da presente dissertação.

```

create view [ext_out].[vw_dim_location] as
SELECT s.[ORG_DETAIL_ID] as org_id,
       s.[org_detail_parent_id] as org_parent_id,
       s.[CODE] as code,
       'S' as org_level_id,
       cast(s.[ORG_DETAIL_ID] as nvarchar) ++ ' - ' ++ cast(s.[NAME] as varchar) as
org_desc,
       '-1' as org_group_id,
       (select cast(si.[code] as nvarchar) ++ ' - ' ++ cast(si.[NAME] as varchar)
from [iCommon].[vwOrganizationalHierarchyMembers] si where si.ORG_DETAIL_ID =
s.ORG_DETAIL_PARENT_ID) org_parent_desc
FROM [iCommon].[vwOrganizationalHierarchyMembers] s
WHERE s.[ORG_HEADER_ID] = 3
and status = 1

```

Código 17 – View vw_dim_location

Esta view resulta na seguinte informação:

org_id	org_parent_id	code	org_level_id	org_desc	org_group_id	org_parent_desc
51	600000009	51	S	51 - San Se Reyes	-1	EN - Espanha Norte
58	600000009	58	S	58 - Alcorcon	-1	EN - Espanha Norte
59	600000009	59	S	59 - Alcala Henares	-1	EN - Espanha Norte
63	600000010	63	S	63 - Marbella	-1	ES - Espanha Sul
65	600000010	65	S	65 - Mijas	-1	ES - Espanha Sul
67	600000010	67	S	67 - Roquetas Mar	-1	ES - Espanha Sul
966	600000009	966	S	966 - Santander	-1	EN - Espanha Norte
1030	600000010	1030	S	1030 - Malaga	-1	ES - Espanha Sul

Figura 51 – Resultado view vw_dim_location

Como se pode visualizar, a última coluna diz respeito à zona geográfica da respetiva loja. Isto é usado para criar uma hierarquia, como se pode comprovar de seguida. Desta forma, é facilitada a navegação nas lojas da organização.

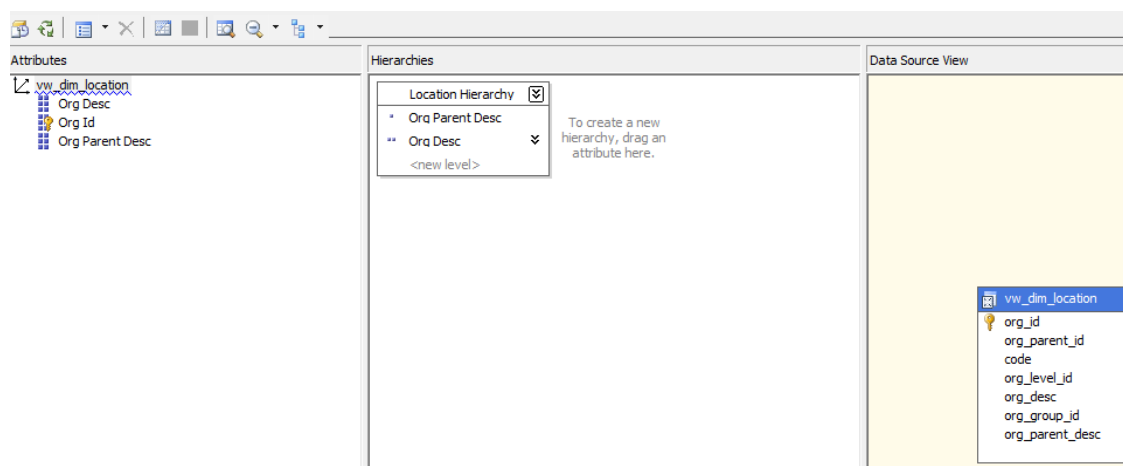


Figura 52 - Hierarquias da dimensão “Loja”

A navegação, utilizando esta hierarquia, resulta então no seguinte:



Figura 53 – Navegação dimensão “Loja”

4.2.2.2 Views factos

Tal como para as dimensões são utilizadas *views* de forma a carregar informação, também para os factos é usada a mesma lógica.

Uma vez que um cubo pode cruzar vários domínios de dados, foram criadas quatro *views*, uma para cada domínio, o que deu origem a quatro *Measure Groups*. Um *Measure Group* diz respeito a um conjunto de métricas de determinado domínio e, neste caso, foram criados os seguintes:

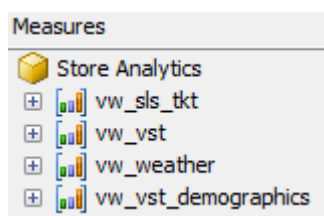


Figura 54 – *Measure Groups* criados

Segue-se uma descrição das *views* mais relevantes para a presente dissertação, a *vw_sls_tkt* e a *vw_vst_demographics*.

4.2.2.2.1 Vw_sls_tkt

Esta *view* diz respeito à informação de vendas. Possui métricas relacionadas com os *tickets* de cada loja, tais como quantidades vendidas, valor de venda, ou até mesmo o próprio número de *tickets*.

Esta informação já se encontra presente em tabelas do Armazém de Dados, e tem a sua origem proveniente de outras fontes.

A *query* de extração destes dados de vendas é a seguinte:

```
create view [ext_out].[vw_sls_tkt] as
select sls.[ORG_DETAIL_ID],
       [HR_DETAIL_ID],
       [DATE_ID],
       hour_24_id as [TIME_ID],
       [SALES_TYPE_ID],
       [TICKET_QTY],
       [TICKET_QTY_RET],
       [PRODUCTS_QTY],
       [PRODUCTS_QTY_RET],
       [NET_SLS_VALUE],
       [NET_SLS_VALUE_RET],
       [NET_SLS_VALUE_DAY_OTD],
       [BASKET_SIZE],
       [BASKET_VALUE],
       [CLIENT_IN_QTY],
       [TARGET_VALUE],
       [BUDGET_VALUE],
       [CONVERSION_RATE],
       [HOM_TICKET_QTY],
       [HOM_TICKET_QTY_RET],
       [HOM_PRODUCTS_QTY],
       [HOM_PRODUCTS_QTY_RET],
       [HOM_NET_SLS_VALUE],
       [HOM_NET_SLS_VALUE_RET],
       [HOM_CONVERSION_RATE]
from [DATA].[F_EMPLOYEE_LOCATION_METRICS_AH] sls inner join
iccommon.organizational_detail od on od.[ORG_DETAIL_ID] = sls.[ORG_DETAIL_ID]
```

Código 18 – View vw_sls_tkt

Trata-se de uma *view* bastante simples, uma vez que os eventuais cálculos de cada métrica já foram efetuados durante o preenchimento das tabelas que lhe servem de base. O resultado desta *view* é o seguinte:

ORG_DETAIL_ID	HR_DETAIL_ID	DATE_ID	TIME_ID	SALES_TYPE_ID	TICKET_QTY	TICKET_QTY_RET	PRODUCTS_QTY	PRODUCTS_QTY_RET	NET_SLS_VALUE	NET_SLS_VALUE_RET	NET_SLS_VALUE_DAY_OTD
51	510	20130401	100000	8106	8	2	7	2	239.96	-42.14	239.9600
51	510	20130401	100000	810999	8	2	7	2	239.96	-42.14	239.9600
51	510	20130401	110000	8106	14	3	13	3	510.30	-336.36	510.3000
51	510	20130401	110000	810999	14	3	13	3	510.30	-336.36	510.3000
51	510	20130401	120000	8106	19	2	20	3	1310.61	-425.98	1310.6100
51	510	20130401	120000	810999	19	2	20	3	1310.61	-425.98	1310.6100
51	510	20130401	130000	8106	25	3	31	3	2371.79	-482.48	2371.7900
51	510	20130401	130000	810999	25	3	31	3	2371.79	-482.48	2371.7900

Figura 55 – Resultado *view* vw_sls_tkt

Já no cubo propriamente dito, foram criadas as seguintes métricas relativas a este *Measure Group*:

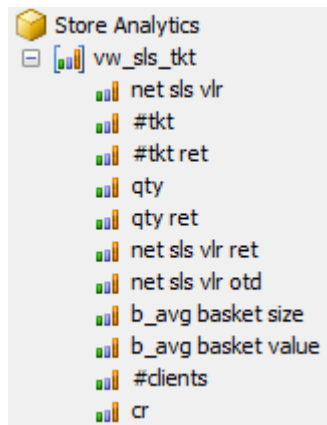


Figura 56 – Measure Group vw_sls_tkt

Segue-se uma descrição de cada uma destas métricas:

Tabela 7 – Descrição das métricas do cubo

<i>Métrica</i>	<i>Significado</i>
<i>Net sls vlr</i>	<i>Valor de venda líquida</i>
<i>#tkk</i>	<i>Número de tickets</i>
<i>#tkk ret</i>	<i>Número de tickets devolvidos</i>
<i>Qty</i>	<i>Número de unidades vendidas</i>
<i>Qty ret</i>	<i>Número de unidades devolvidas</i>
<i>Net sls vlr ret</i>	<i>Valor de venda líquida devolvido</i>
<i>Net sls vlr otd</i>	<i>Valor de venda líquido acumulado (open to date)</i>
<i>B_avg basket size</i>	<i>Número médio de artigos por ticket</i>
<i>B_avg basket value</i>	<i>Valor médio de artigos por ticket</i>
<i>#clients</i>	<i>Número de clientes</i>
<i>Cr</i>	<i>Taxa de Conversão</i>

4.2.2.2.2 Vw_vst_demographics

Esta *view* é responsável por carregar para o cubo informação relativa aos dados demográficos dos visitantes. Tem por base a tabela de factos “STAG_SMARTSCAN_READS”, cujo fluxo de preenchimento foi detalhado até ao momento.

```

create view [ext_out].[vw_vst_demographics] as
select ed.ORG_DETAIL_ID org_id,
       cast(cast(year(sr.READ_DATETIME) as nvarchar) ++
REPLACE(STR(cast(month(sr.READ_DATETIME) as nvarchar), 2), SPACE(1), '0') ++
REPLACE(STR(cast(day(sr.READ_DATETIME) as nvarchar), 2), SPACE(1), '0') as int) as
date_id,
       cast(REPLACE(STR(cast(DATEPART(hh,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') ++ REPLACE(STR(cast(DATEPART(minute,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') ++ REPLACE(STR(cast(DATEPART(ss,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') as int) as time_id,
       sr.AGE as age_id,
       sr.GENDER as gender_id,
       count(1) as #Occurences
from [IDS].[STAG_SMARTSCAN_READS] sr
INNER join [iCommon].[EQUIPMENT_DETAIL] ed on ed.[EQUIP_DETAIL_ID] = sr.[NODE_ID]
where sr.STARTSTOP = 0
group by ed.[ORG_DETAIL_ID] ,
       cast(cast(year(sr.READ_DATETIME) as nvarchar) ++
REPLACE(STR(cast(month(sr.READ_DATETIME) as nvarchar), 2), SPACE(1), '0') ++
REPLACE(STR(cast(day(sr.READ_DATETIME) as nvarchar), 2), SPACE(1), '0') as int),
       cast(REPLACE(STR(cast(DATEPART(hh,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') ++ REPLACE(STR(cast(DATEPART(minute,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') ++ REPLACE(STR(cast(DATEPART(ss,sr.READ_DATETIME) as nvarchar), 2), SPACE(1),
'0') as int),
       sr.AGE,
       sr.GENDER

```

Código 19 – View vw_vst_demographics

Uma vez que a tabela de factos apenas possui o ID do equipamento, é necessário cruzar com a tabela EQUIPMENT_DETAIL, que possui o mapeamento entre os equipamentos e as lojas a que pertencem.

A execução da *view* retorna a seguinte informação:

org_id	date_id	time_id	age_id	gender_id	#Occurences
1029	20140819	180833	3	2	1
59	20140816	160835	4	1	2
59	20140816	160854	4	1	1
59	20140816	170828	4	1	1
59	20140823	180830	3	1	1
59	20140827	170858	3	1	1
59	20140910	90915	3	1	1
1029	20140827	170832	4	1	1
59	20140831	90812	5	2	1
1029	20140901	180916	4	1	1

Figura 57 – Resultado *view* vw_vst_demographics

Interpretando os resultados obtidos, é possível verificar que esta *view* retorna cinco atributos cuja finalidade passa por efetuar a ligação com as dimensões descritas anteriormente. Além

disso, é retornada uma métrica denominada “#Occurrences”, ou seja, o número de ocorrências. Analisando, por exemplo, a segunda linha, chega-se à conclusão de que no dia 16 de Agosto de 2014, na loja 59, às 16:08:35, foram detetadas duas pessoas com a idade correspondente à faixa etária do grupo 4 (Older Adult – 35 a 64 anos), e do sexo 1 (Masculino).

Com o cubo e a navegação nas dimensões que este proporciona, este processo torna-se algo bastante fácil e rápido.

4.2.2.3 Associação Dimensões e *Measure Groups*

Uma vez que as dimensões e os factos se encontram preenchidos, o passo seguinte diz respeito à associação entre ambos.

Isto é feito através de um processo bastante simples, em que se associa cada *Measure Group* às dimensões que lhe digam respeito.

Dimensions	vw_sls_tkt	vw_vst	vw_weather	vw_vst_demographics
vw_dim_hr	hr id			
vw_dim_time	time_id	time_id		time_id
vw_dim_calendar	Date Id	Date Id	Date Id	Date Id
vw_dim_age				Age Group Id
vw_dim_gender				Gender Id
vw_dim_location	Org Id	Org Id	Org Id	Org Id

Figura 58 – Associação *Measure Group* a Dimensões

Como se pode constatar, nem todas as dimensões estão associadas a todos os *Measure Groups*, como por exemplo a dimensão relativa à faixa etária. Esta apenas diz respeito a dados demográficos, e não tem qualquer relação com dados de *ticket* ou de qualquer outro *Measure Group*.

Nota para o facto de o *Measure Group* vw_vst, relativo a visitantes, dizer respeito a outro sistema de contadores em funcionamento paralelo com o Intel AIM. Desta forma, não é possível associar os dados demográficos a este *Measure Group*, uma vez que este não retorna informação com este detalhe. Por isso mesmo, foi necessário criar um *Measure Group* distinto, neste caso o vw_vst_demographics.

4.2.2.4 Processamento do Cubo

O último passo diz respeito ao processamento do cubo.

Com a associação entre as dimensões e os *Measure Groups*, apenas resta processar o cubo, ou seja, importar os dados utilizando as *views* definidas anteriormente.

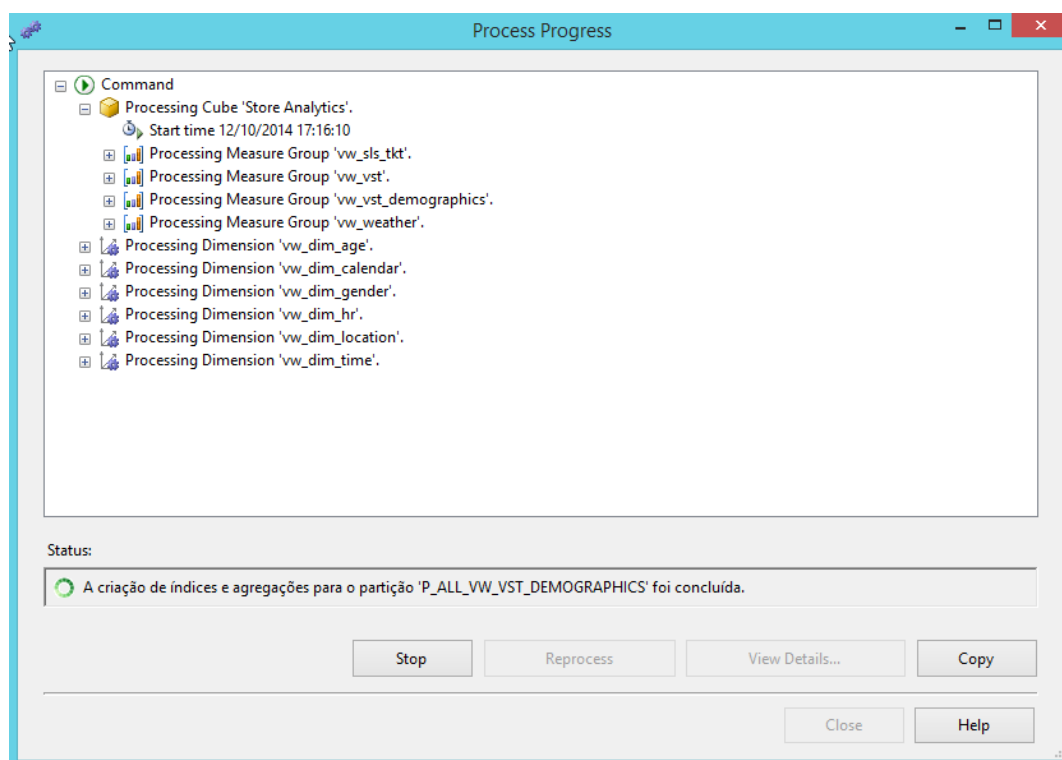


Figura 59 – Processamento do cubo

Uma vez que se trata da primeira vez a ser processado, tanto as dimensões como os *Measure Groups* fazem parte deste processamento.

Obtém-se assim o cubo já com a informação proveniente do Armazém de Dados processada, e pronto a ser explorado.

4.3 Componente Utilizador

O último componente diz respeito ao objetivo final da solução, ou seja, à exploração da informação de uma forma fácil e rápida.

Com o cubo já processado, basta ao utilizador final efetuar uma ligação a este e proceder à exploração. A ferramenta mais utilizada em ligações a cubos de Analysis Services é o Excel. Basta, no próprio Excel, aceder ao seguinte menu:

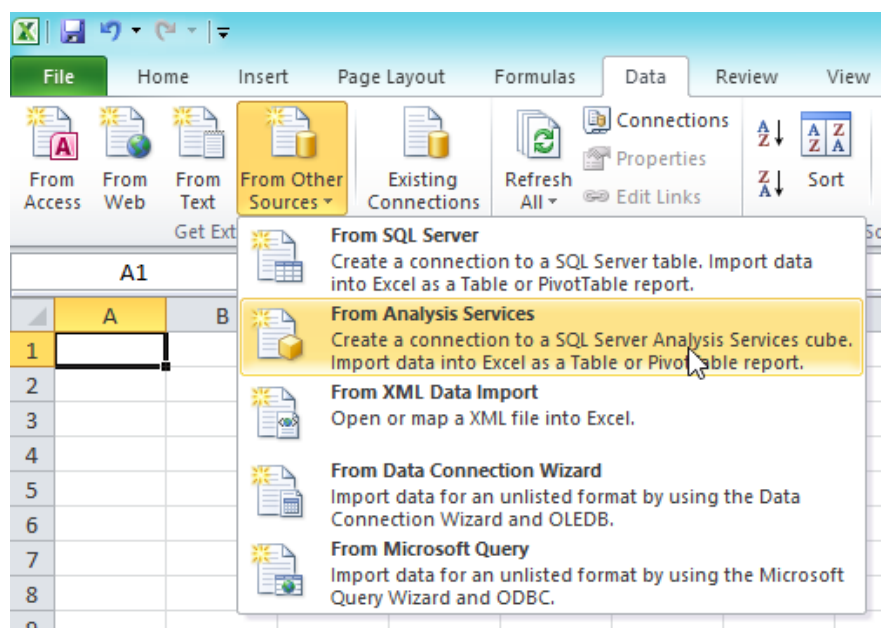


Figura 60 – Estabelecimento ligação ao cubo através do Excel

Após se introduzir o endereço do servidor Analysis Services, surgem as Bases de Dados de Analysis Services e os cubos disponíveis, bastando seleccionar o pretendido, neste caso o “Store Analytics” localizado na Base de Dados SEEPLUS_DEMO.

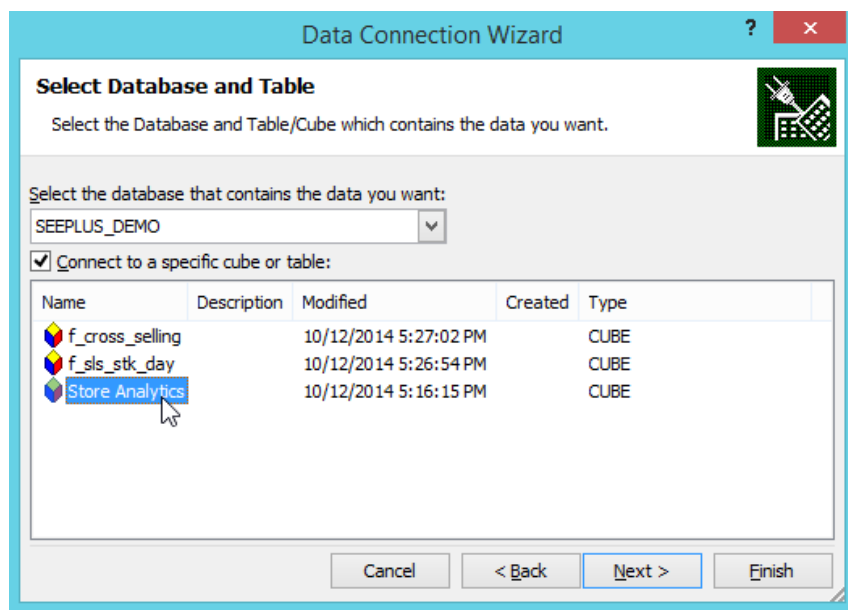


Figura 61 – Seleção do cubo

Após se efetuar a ligação ao cubo, é criada uma *pivot table* com a estrutura do cubo. É então possível arrastar dimensões e métricas para a *pivot* de forma a criar relatórios de análise do negócio.

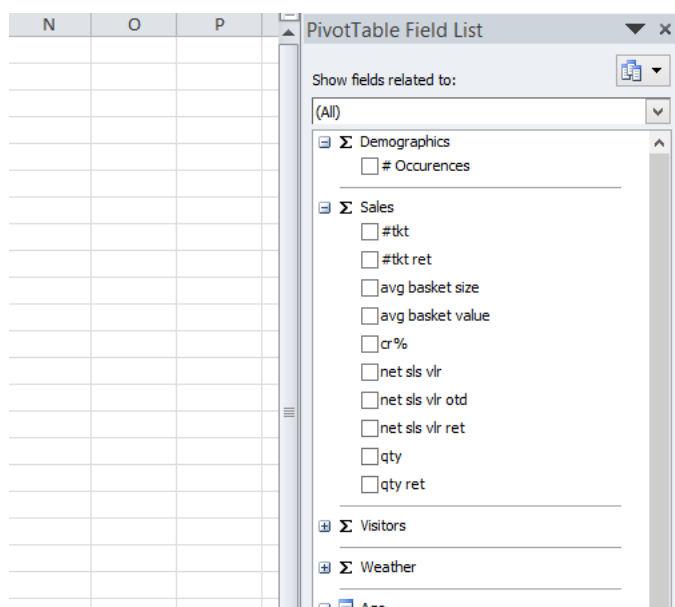


Figura 62 – Pivot Table com base no cubo

No capítulo seguinte encontram-se alguns exemplos de relatórios criados utilizando-se este cubo para tal.

5 Resultados obtidos

Este capítulo diz respeito aos resultados obtidos. Tal como referido anteriormente, uma vez desenvolvido e processado o cubo, são os utilizadores finais, normalmente analistas de negócio, os responsáveis pela realização de relatórios.

Serão, portanto, demonstrados alguns exemplos de relatórios tendo por base os dados de produção compreendidos entre 29 de Julho de 2014 e dia 11 de Setembro de 2014, e referentes às duas lojas que, a essa data, possuíam a solução implementada.

Nota especial que as lojas estão localizadas em Espanha e que, devido a fatores geográficos e culturais, as análises têm por isso resultados diferentes dos que seriam de esperar em determinados períodos da semana e do dia, caso estas fossem realizadas em Portugal.

Por questões de simplificação, as lojas serão referidas como Loja 1 e Loja 2.

5.1 Distribuição por Sexo

Os seguintes gráficos representam, para a semana 34 de 2014, qual a percentagem de Homens e Mulheres para cada uma das lojas.

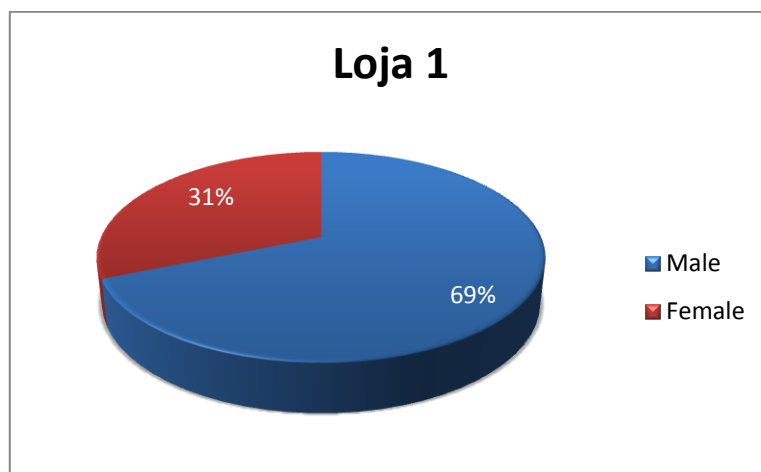


Figura 63 - Visitantes por Sexo na Loja 1

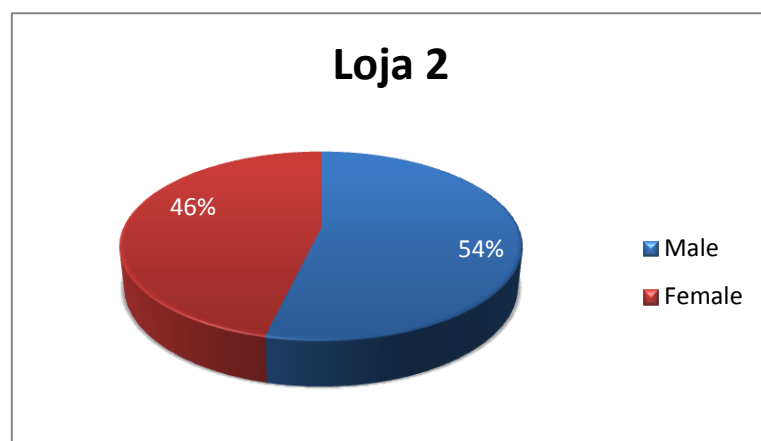


Figura 64 - Visitantes por Sexo na Loja 2

Facilmente se comprova que a Loja 1 tem maior afluência de visitantes do sexo Masculino, enquanto que na Loja 2 não existe uma discrepância tão elevada, situando-se perto da casa dos 50%.

Outra análise interessante diz respeito à afluência de cada sexo em cada período do dia:

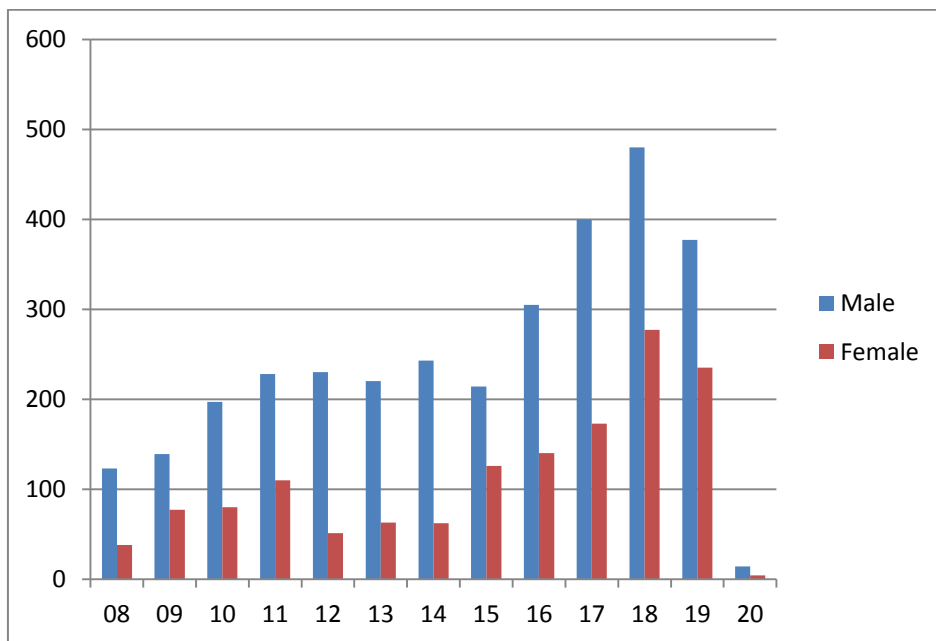


Figura 65 - Visitantes por Sexo e Hora – Loja 1

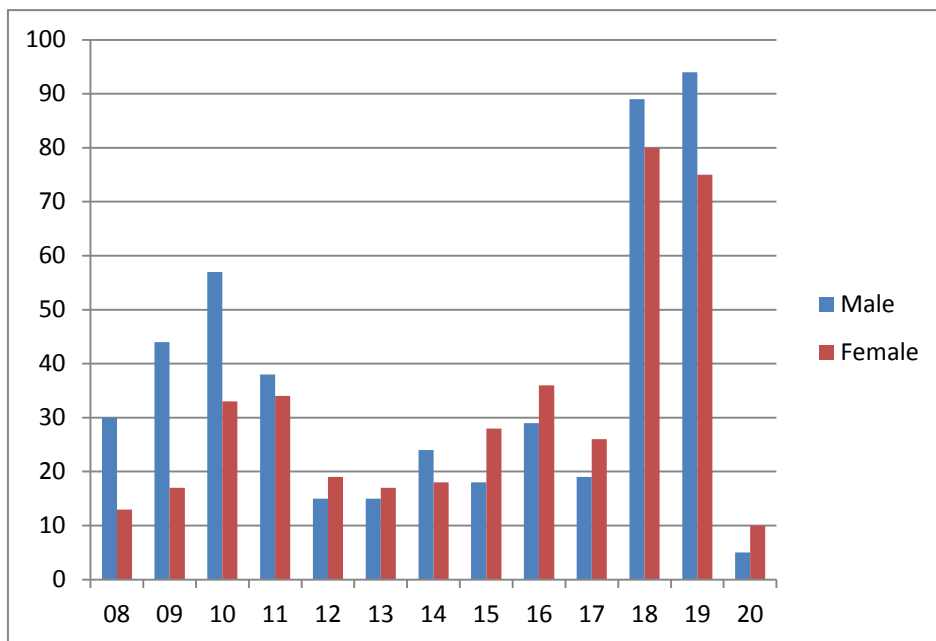


Figura 66 - Visitantes por Sexo e Hora – Loja 2

Após verificação dos dois gráficos anteriores, é possível visualizar que a Loja 1 tem predominantemente mais visitantes do sexo Masculino durante todo o dia, ao passo que a

Loja 2 é mais equilibrada, chegando mesmo o público Feminino a superar o Masculino durante alguns períodos. Isto vai de encontro à análise anterior.

5.2 Distribuição por Faixa Etária

Uma das análises bastante úteis que se torna assim possível de ser efetuada, é a distribuição dos visitantes de acordo com a sua faixa etária. Isto é algo bastante afetado consoante a localização geográfica de cada loja.

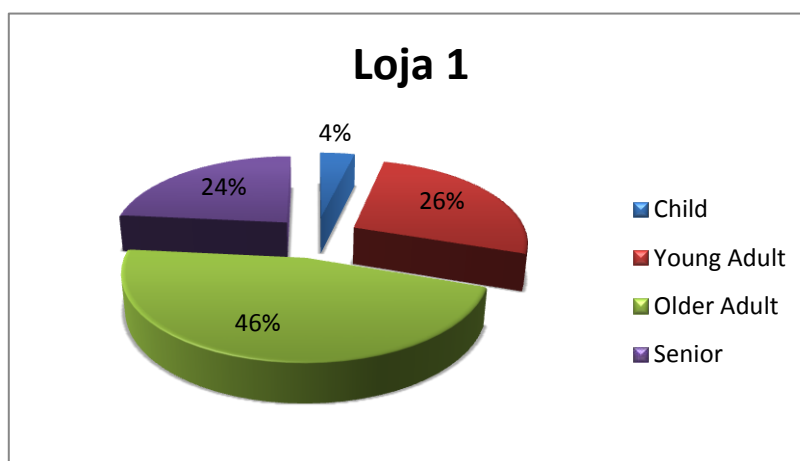


Figura 67 - Visitantes por faixa etária – Loja 1

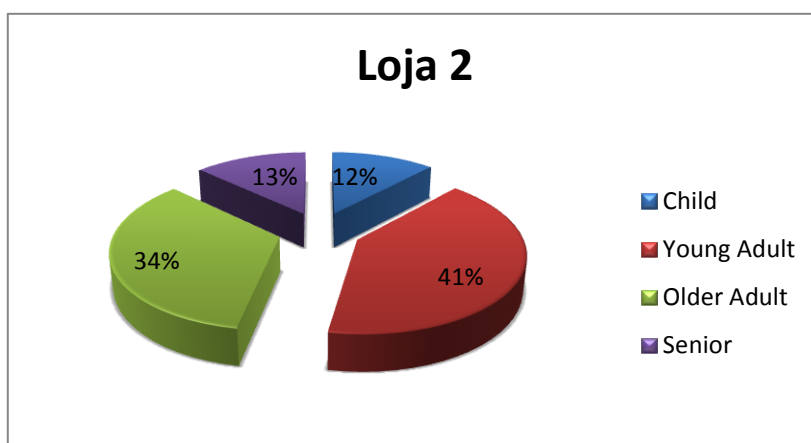


Figura 68 - Visitantes por faixa etária – Loja 2

No que diz respeito à faixa etária, também aqui se encontra uma diferença significativa. Enquanto que na Loja 1 é predominante uma faixa etária entre os 35 e os 64 anos, na Loja 2 trata-se da faixa etária imediatamente inferior, ou seja, dos 16 aos 34 anos.

Para cada loja, foram também efetuadas análises à faixa etária predominante em cada dia. Segue-se um exemplo de *Drill-Down*, que permite navegar, neste caso, na hierarquia relativa à dimensão Data.

# Occurrences	Column Labels	Child	Young Adult	Older Adult	Senior	Grand Total
2014		573	1900	1894	771	5138
Ago-2014		341	1088	1147	384	2960
Set-2014		232	812	747	387	2178
1-Set-2014		36	91	94	29	250
2-Set-2014		39	133	98	59	329
3-Set-2014		23	75	81	32	211
4-Set-2014		27	88	65	30	210

Figura 69 – *Drill-Down* dimensão Data

De seguida apresenta-se graficamente estes dados, relativos à semana 34 de 2014.

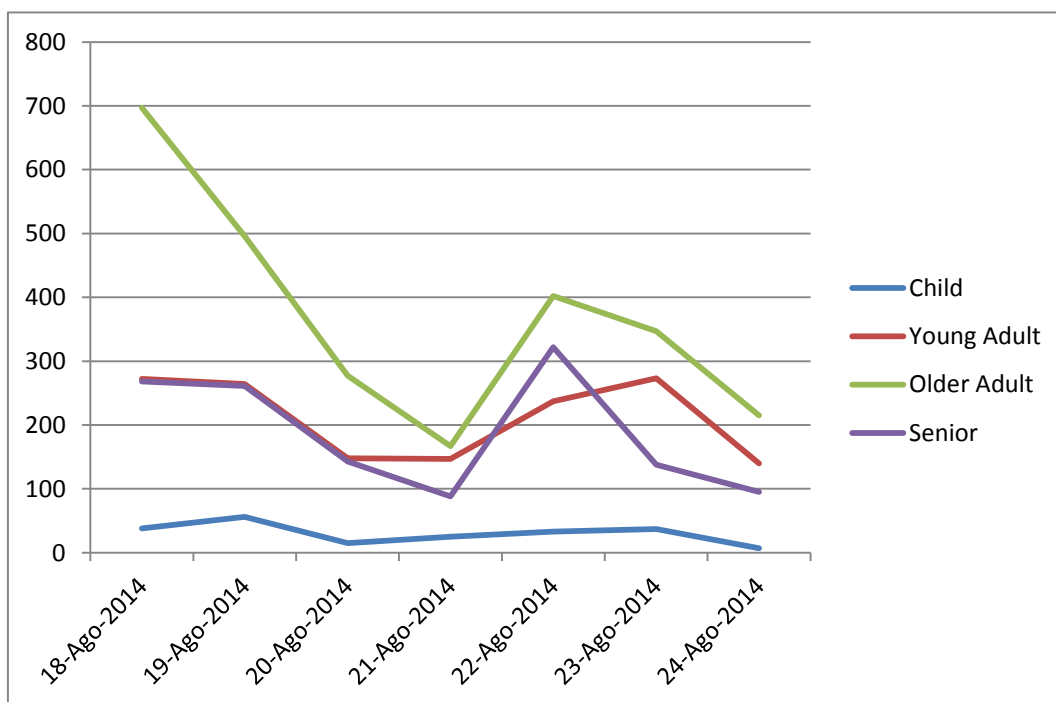


Figura 70 - Visitantes por faixa etária e dia da semana – Loja 1

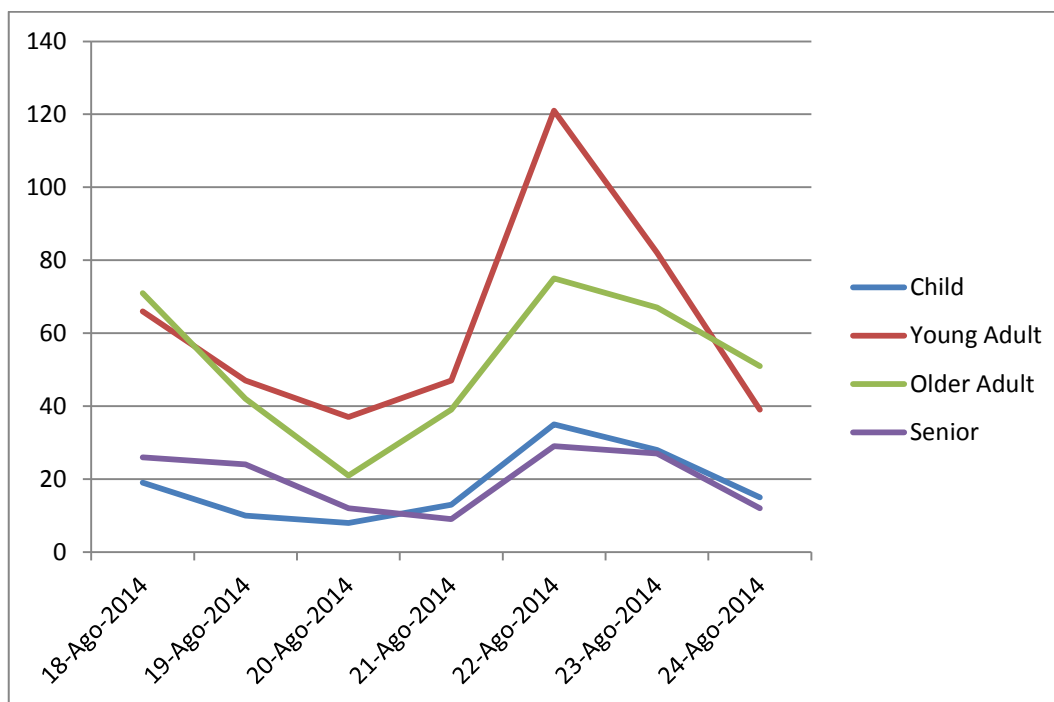


Figura 71 - Visitantes por faixa etária e dia da semana – Loja 2

Em ambas as lojas as faixas etárias predominantes são as compreendidas entre os 35 e os 64 na Loja 1, e entre os 16 e os 34 na Loja 2.

5.3 Vendas

Um dos pontos fulcrais na criação desta solução diz respeito à análise das vendas. Desta forma, é possível verificar o desempenho das lojas. O seguinte gráfico apresenta, para cada dia da semana 34 de 2014, o total de vendas líquidas em Euros da Loja 1 e da Loja 2.

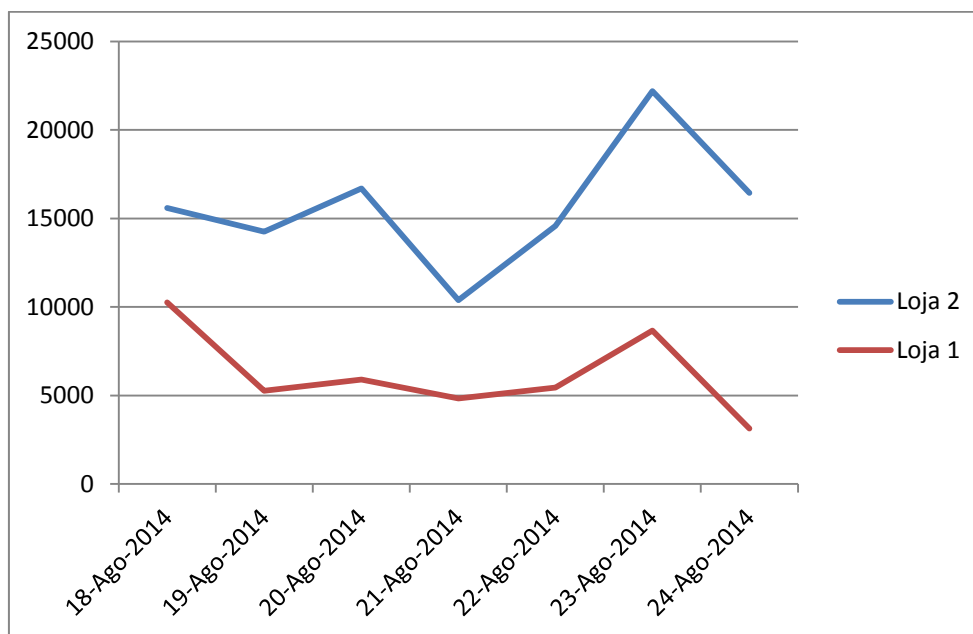


Figura 72 - Venda líquida na Semana 34

Outra métrica importante na análise de vendas diz respeito à Taxa de Conversão. Para as mesmas condições, foi analisada esta taxa ao longo dos dias, sendo os resultados apresentados no gráfico seguinte:

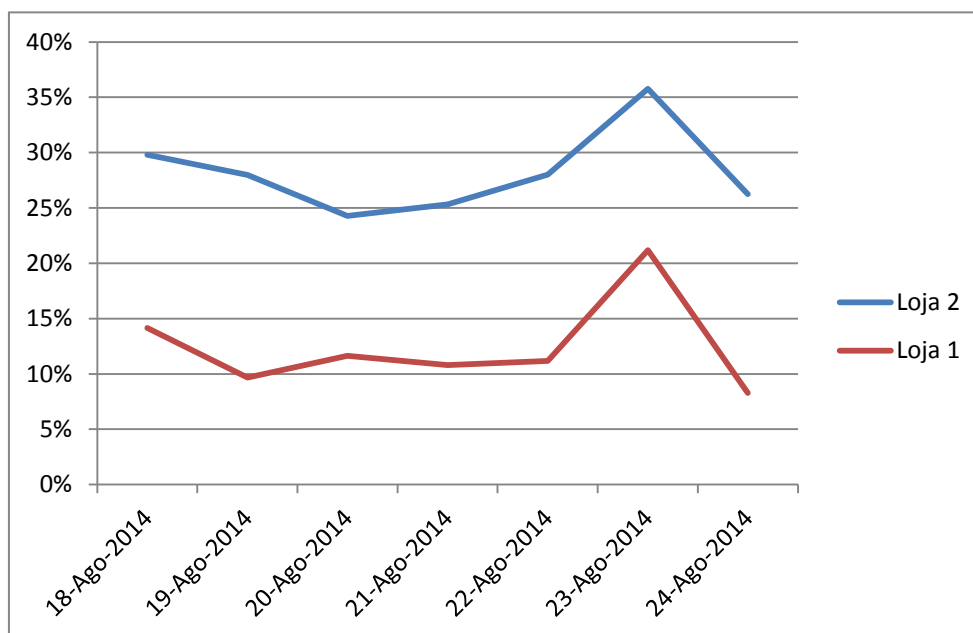


Figura 73 - Taxa de Conversão Loja 1 na Semana 34

Tal como seria de esperar, o valor de venda líquida “dispara” durante o Sábado. Por outro lado, apesar de a Taxa de Conversão atingir o pico efetivamente no Sábado, à segunda-feira surge também um valor bastante elevado.

5.4 Impacto dos fatores demográficos nas vendas

Cruzando a informação anterior, é possível construir *Dashboards* no próprio Excel. Desta forma, o utilizador final fica com uma perceção instantânea do desempenho da companhia. O uso de gráficos é um fator determinante, uma vez que transmite esta informação de uma forma visual, e exige um menor esforço na compreensão por parte do utilizador final.

Segue-se o exemplo de uma *Dashboard* criada, onde foi utilizada a semana 36 e a Loja 59 (referida até ao momento como Loja 1).

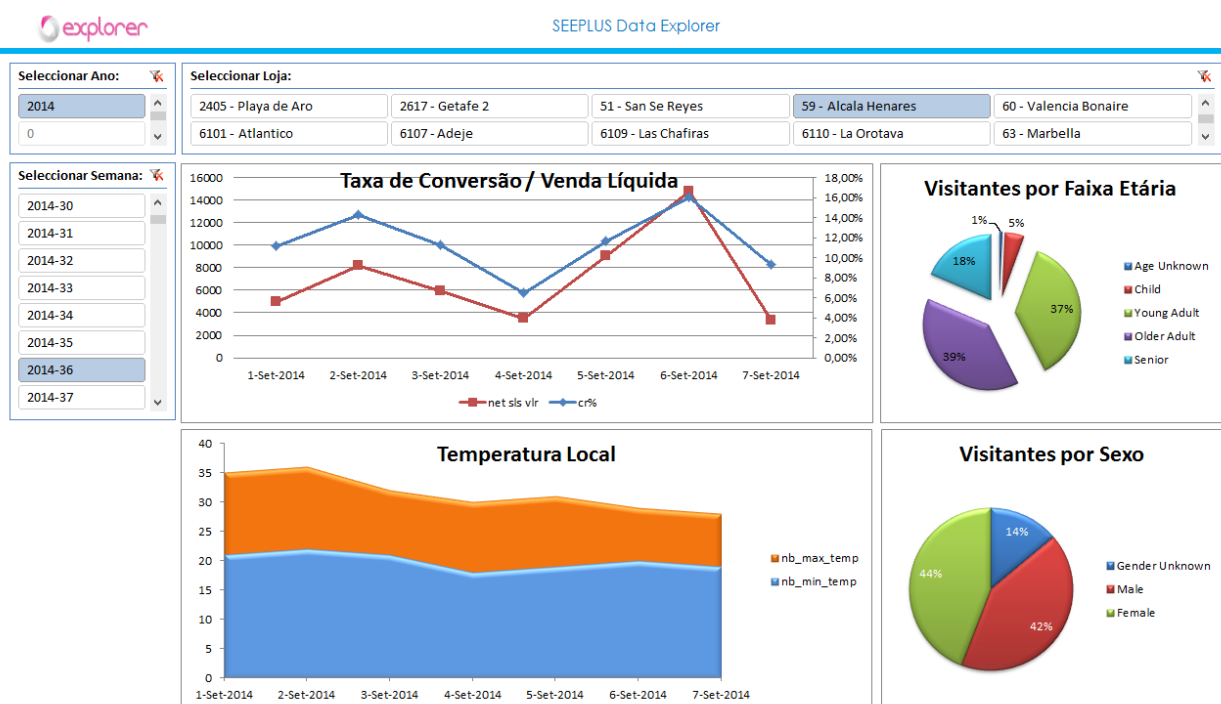


Figura 74 - Dashboard em Excel (Semana 36)

Como se pode visualizar, esta *Dashboard* possui assim informação relativa a vários domínios:

- Vendas
- Fatores Demográficos
- Meteorologia

Uma vez que os filtros relativos às dimensões Data e Loja são partilhados entre todos os gráficos, facilmente se efetua a alteração a um filtro e se visualiza o resultado obtido em todos eles. Desta forma, consegue-se ter uma perceção do impacto entre os vários domínios de informação. Na Figura 75 encontra-se exatamente a mesma informação, mas para a semana anterior (Semana 35).

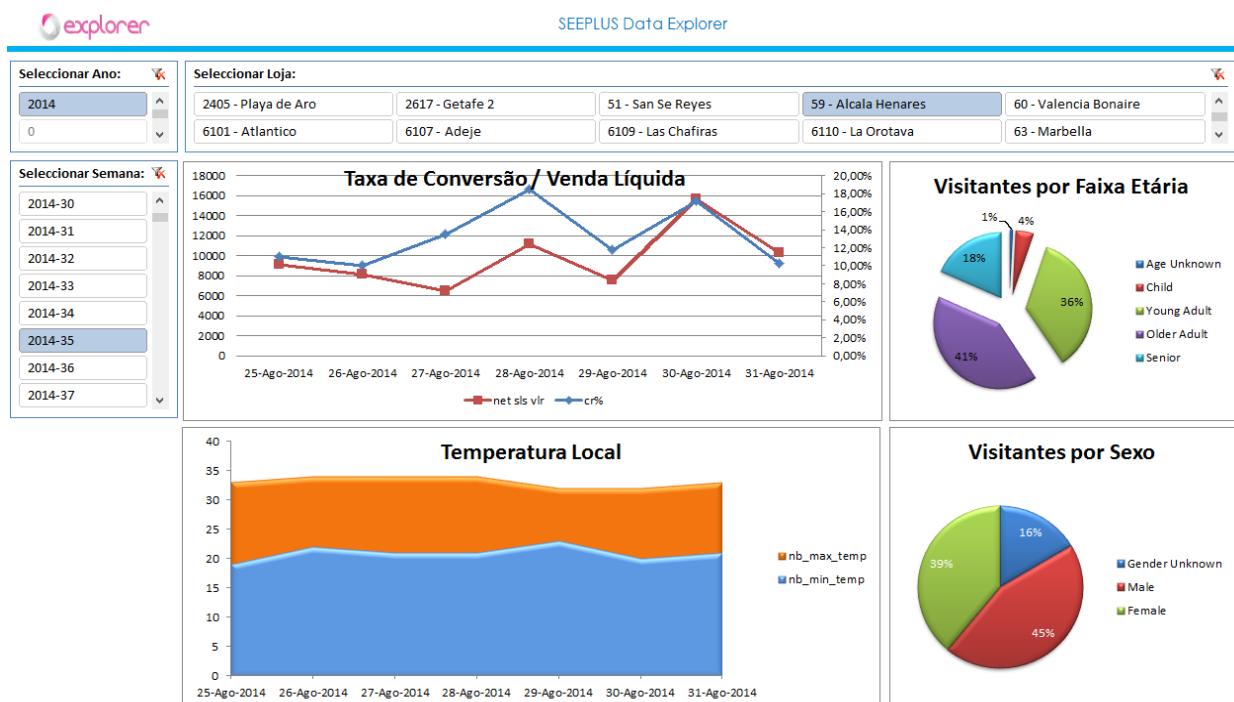


Figura 75 – Dashboard em Excel (Semana 35)

Nota-se assim uma diferença de valores na taxa de conversão e no valor de venda líquida, especialmente a meio da semana. No entanto, os fatores demográficos sofreram apenas ligeiros desvios, sendo a diferença mais significativa a relativa ao sexo.

6 Conclusão

A presente dissertação consistiu no desenvolvimento de uma solução e consequente integração no SeePlus, produto já existente e proprietário da InovRetail.

Denominado SmartScan, este novo módulo proporciona assim aos utilizadores finais a realização de análises de negócio com informação demográfica dos visitantes das suas lojas, informação esta inexistente até ao momento. Estes novos indicadores são uma mais-valia do ponto de vista da informação de gestão, uma vez que se torna assim possível comparar se está a ser atingido com sucesso o público-alvo pretendido, e tomar ações caso os resultados não sejam os esperados.

Aliando esta nova informação a dados relativos a outros domínios, como por exemplo vendas, torna-se também possível ver o seu impacto no desempenho de cada loja.

O desenvolvimento efetuado tinha como objetivo, assim que possível, passar a piloto num dos clientes da InovRetail. Tendo isto em conta, este foi concluído com êxito e encontra-se, de facto, em fase de piloto com dados provenientes do ambiente de produção do cliente. Neste momento, encontram-se duas lojas deste cliente com a solução instalada, sendo que o objetivo passa pela adição de mais lojas à medida que o piloto vai avançando de fase.

Ao longo do desenvolvimento, foram encontradas algumas dificuldades, tais como:

- Restrição do tamanho da mensagem HTTP: na fase do carregamento dos dados recolhidos pelo AIM no Armazém de Dados, estava a ser ultrapassado o limite configurado do tamanho da mensagem HTTP. De forma a contornar este problema, apenas são enviadas no máximo 50 registos da Base de Dados Local.
- *Firewall*: Foram encontradas dificuldades ao ser estabelecida a ligação com o *Web Service* devido à *Firewall* do cliente onde o piloto foi implementado. No entanto, após cooperação com o departamento técnico do cliente, foi configurado um *proxy* que permitiu efetuar esta comunicação com sucesso.

6.1 Melhorias / Trabalho Futuro

Uma vez que o desenvolvimento efetuado teve o objetivo de passar a protótipo o quanto antes, existem alguns pontos de melhoria:

- Adição da dimensão Produto ao cubo. Esta dimensão já se encontra presente no projeto do cubo Analysis Services, no entanto é necessário associar os artigos à tabela de factos. Além disso, é também necessário efetuar ajustes nesta dimensão, de forma a melhorar o seu aspeto para o utilizador final.
- Consolidação da informação do AIM: uma vez que, após o carregamento no Armazém de Dados, os dados se encontram numa *Staging Area*, um dos pontos de trabalho futuro passa por consolidar esta informação numa tabela final.
- Utilização de nomenclaturas: uma vez que o objetivo passou por uma implementação no cliente o quanto antes, devido a tratar-se de um piloto, ficaram algumas nomenclaturas por respeitar.
- Instâncias do *Socket* do AIM: encontrando-se uma ligação efetuada a um *Socket* do Intel AIM, após ser aberta a ferramenta de visualização em tempo real da própria Intel, este *Socket* é desligado, sendo assim necessário restabelecer uma nova ligação. Isto faz com que os dados não sejam mais capturados, sendo que o melhoramento passa por elaborar um mecanismo que detete esta perda de ligação, e efetue automaticamente uma nova.

Referências

- "buy-facebook-fans", 2014. *Advanced Facebook Facial Recognition*. [Online]
Available at: <http://facebook-fans-gurus.com/advanced-facebook-facial-recognition/>
[Accessed Setembro 2014].
- Absolut-e, 2009. *Comparison of Bill Inmon and Ralph Kimball paradigm*. [Online]
Available at: <http://absolut-e-blog.blogspot.pt/2009/06/comparison-of-bill-inmon-and-ralph.html>
[Accessed Outubro 2014].
- Ackerman, D., 2013. *Xbox One and PlayStation 4: Facial recognition shootout*. [Online]
Available at: <http://www.cnet.com/news/xbox-one-and-playstation-4-facial-recognition-shootout/>
[Accessed Setembro 2014].
- Anthony, S., 2014. *Facebook's facial recognition software is now as accurate as the human brain, but what now?*. [Online]
Available at: <http://www.extremetech.com/extreme/178777-facebooks-facial-recognition-software-is-now-as-accurate-as-the-human-brain-but-what-now>
[Accessed Agosto 2014].
- Atasoy, H., 2014. *Rapid Object Detection in .NET*. [Online]
Available at: <http://www.codeproject.com/Articles/436521/Rapid-Object-Detection-in-NET>
[Accessed Setembro 2014].
- Baer, H., 2010. *Partitioning with Oracle Database 11g Release 2*. Redwood Shores: Oracle.
- Ballard, C. et al., 1998. *Data Modeling Techniques for Data Warehousing*. s.l.:IBM.
- Bolme, D. S., Strout, M. & Beveridge, J. R., 2007. *FacePerf: Benchmarks for Face Recognition Algorithms*. Fort Collins: s.n.
- Breslin, M., 2004. *Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models*. s.l.:Business Intelligence Journal.
- Bright, P., 2013. *A new Skype experience in the living room, with Xbox One*. [Online]
Available at: <http://arstechnica.com/gaming/2013/11/a-new-skype-experience-in-the-living-room-with-xbox-one/>
[Accessed Agosto 2014].
- Centhala, S., 2009. *Comparison of Bill Inmon and Ralph Kimball paradigm*. [Online]
Available at: <http://srinicenthala.blogspot.pt/2009/01/comparison-of-bill-inmon-and-ralph.html>
[Accessed Agosto 2014].

- Chandler, N., 2011. *How Facebook Photo Tags Work*. [Online]
Available at: <http://computer.howstuffworks.com/internet/tips/facebook-photo-tags.htm>
[Accessed Setembro 2014].
- Chaudhuri, S. & Dayal, U., 1997. *An Overview of Data Warehousing and OLAP Technology*.
- Conroy, P. & Bearse, S., 2006. *The Changing Nature of Retail: Planting the seeds for sustainable growth Customer Conversion*. s.l.:Deloitte.
- Couceiro, F. d. S., 2012. *Agile Modeling Data Warehouse Development*. Porto: ISEP.
- DatabaseETL, 2014. *ETL Tools – Top 10 ETL Tools Reviews*. [Online]
Available at: <http://www.databasetl.com/etl-tools-top-10-etl-tools-reviews/>
[Accessed Agosto 2014].
- Datawarehouse4u.info, 2009. *Datawarehouse4u.info*. [Online]
Available at: <http://Datawarehouse4u.info>
- DeWitt, D. J., Madden, S. & Stonebraker, M., n.d. *How to Build a High-Performance Data Warehouse*.
- Experian Footfall, 2014. *Experian Footfall*. [Online]
Available at: (<http://www.footfall.com/>)
[Accessed Outubro 2014].
- Experian, 2006. *Experian acquires FootFall Ltd*. [Online]
Available at: <http://www.experianplc.com/news/company-news/2006/05-01-2006.aspx>
[Accessed Outubro 2014].
- FinalVerdict, 2013. *Xbox One vs. Playstation 4: Kinect 2.0 vs. Playstation 4 Camera*. [Online]
Available at: <http://www.ign.com/blogs/finalverdict/2013/11/02/xbox-one-vs-playstation-4-kinect-20-vs-playstation-4-camera>
[Accessed Agosto 2014].
- Future Travel Experience, 2011. *Russian airport trials facial recognition biometrics*. [Online]
Available at: <http://www.futuretravelexperience.com/2011/03/russian-airport-trials-facial-recognition-biometrics/>
[Accessed Setembro 2014].
- Homeland Security News Wire, 2011. *Largest Moscow airport testing of facial biometric system*. [Online]
Available at: <http://www.homelandsecuritynewswire.com/largest-moscow-airport-testing-facial-biometric-system>
[Accessed Agosto 2014].

IBM, 2014. [Online]

Available at: <http://www-01.ibm.com/software/data/infosphere/>

[Accessed Outubro 2014].

Inmon, W. H., 2002. *Building the Data Warehouse: Third Edition*. s.l.:Wiley.

Inmon, W. H., 2010. *A Tale of Two Architectures*. s.l.:s.n.

Introna, L. D. & Nissenbaum, H., 2009. *Facial Recognition Technology: A Survey of Policy and Implementation Issues*. New York: s.n.

Jiang, B., 2012. *Is Inmon's Data Warehouse Definition Still Accurate?*. [Online]

Available at: <http://www.b-eye-network.com/view/16066>

[Accessed Agosto 2014].

Johnson, J., 2009. *Digital Camera Face Recognition: How It Works*. [Online]

Available at: <http://www.popularmechanics.com/technology/how-to/4218937>

[Accessed Setembro 2014].

Kimball, R. & Ross, M., 2002. *The Data Warehouse Toolkit: The Complete Guide To Dimensional Modeling*. 2nd ed. s.l.:Wiley.

Kimball, R. & Ross, M., 2013. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. 3rd ed. Indianapolis: Wiley.

Langley, H., 2013. *How Skype could help Xbox One edge the PS4 out of the living room*. [Online]

Available at: <http://www.techradar.com/news/gaming/consoles/how-skype-could-help-the-xbox-one-edge-the-ps4-out-of-the-living-room-1200959>

[Accessed Setembro 2014].

LASE PeCo, 2014. *LASE PeCo*. [Online]

Available at: <http://www.peoplecounter.de/en/>

[Accessed Agosto 2014].

Luhn, H. P., 1958. A Business Intelligence System. *IBM Journal*.

Mayer, O., 2014. *Overcoming Legacy Database Integration Challenges in the Era of Big Data*.

[Online]

Available at: <http://www.attunity.com/blog/overcoming-legacy-database-integration-challenges-era-big-data>

[Accessed Outubro 2014].

Microsoft, 2014. [Online]

Available at: <http://www.microsoft.com/en-us/server-cloud/products/sql-server/default.aspx>

[Accessed Outubro 2014].

Moody, D. L. & Kortink, M. A., 2000. *From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design*. Melbourne: s.n.

Oliveira, P., 2013. *Armazenamento e Processamento Analítico de Dados*, Porto: ISEP.

OpenCV, 2014. *Face Detection Using Haar Cascades*. [Online]

Available at:

http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

[Accessed Setembro 2014].

Oracle, 2012. *Data Integration Architectures for Operational Data Warehousing*. Redwood Shores: Oracle.

Oracle, 2014. *Oracle*. [Online]

Available at: <http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>

[Accessed Outubro 2014].

Patterson, D. A., Gibson, G. & Katz, R. H., 1988. *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. Berkeley: s.n.

Perdikaki, O., Kesavan, S. & Swaminathan, J. M., 2012. *Effect of Traffic on Sales and Conversion Rates of Retail Stores*. s.l.:s.n.

Protalinski, E., 2013. *Overview: Here's how Skype will work on the Xbox One*. [Online]

Available at: <http://thenextweb.com/microsoft/2013/11/20/skype-works-doesnt-work-xbox-one/>

[Accessed Setembro 2014].

SAP, 2014. [Online]

Available at: <http://www.sap.com/portugal/pc/tech/enterprise-information-management/software/data-integrator/index.html>

[Accessed Outubro 2014].

SAS, 2014. [Online]

Available at: <http://support.sas.com/software/products/etls/>

[Accessed Outubro 2014].

Shah, A., 2013. *Inside Samsung Galaxy S4's face and eye-tracking technology*. [Online]

Available at: <http://www.computerworld.com/article/2495384/smartphones/inside-samsung-galaxy-s4-s-face-and-eye-tracking-technology.html>

[Accessed Agosto 2014].

Silva, B., 2014. *Inside Visions: A 'startup' que aposta no futuro*. [Online]

Available at: <http://saldopositivo.cgd.pt/empresas/inside-visions-startup-que-aposta-em->

tecnologias-futuristas

[Accessed Outubro 2014].

Solution Providers For Retail, 2013. *As Digital Signage Goes Mainstream, the Value Is in Customization*. [Online]

Available at: <http://www.solutionprovidersforretail.com/article/digital-signage-goes-mainstream-value-customization>

[Accessed Outubro 2014].

Sousa, R. S., 2011. *Modelo Dimensional para Data Warehousing*. [Online]

Available at: <http://ruisalgueirosousa.wordpress.com/2011/03/10/modelo-dimensional-para-data-warehousing-2/>

[Accessed Agosto 2014].

Standen, J., 2008. *Data Warehouse vs Data Mart*. [Online]

Available at: <http://www.datamartist.com/data-warehouse-vs-data-mart>

[Accessed Agosto 2014].

Taigman, Y., Yang, M., Ranzato, M. A. & Wolf, L., 2014. *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*. s.l.:s.n.

Wayman, J., Jain, A., Maltoni, D. & Maio, D., 2005. *An Introduction to Biometric Authentication Systems*. s.l.:Springer.

Xbox.com, 2014. *Como configurar e utilizar o início de sessão do Kinect*. [Online]

Available at: <http://support.xbox.com/pt-PT/xbox-one/kinect/setup-kinect-sign-in>

[Accessed Setembro 2014].