



**INSTITUTO SUPERIOR
DE TECNOLOGIAS AVANÇADAS**

Licenciatura Informática

2012/2013

***Sistema de Gestão Equipamentos Informáticos
Caso do Município do Seixal***

Realizado por:

Linda Zara da Luz Valentim

Nº1750 - Turma B - Lisboa

Coordenador: Professor Doutor Nuno Correia

Orientador Metodológico: Professor Doutor Pedro Brandão

Orientador de Especialidade: Professor Doutor Joaquim Aleixo

*“Viva como se fosse morrer amanhã,
aprenda como se fosse viver para sempre”*

Mahatma Gandhi

Dedicatória

Dedico este trabalho a todas as pessoas, que contribuíram directa ou indirectamente para a minha formação moral, ética e profissional.

Agradecimentos

Agradeço ao Instituto Superior de Tecnologias Avançadas, professores e colaboradores pelo apoio e disponibilidade sempre demonstrados ao longo dos 3 anos de curso.

Agradeço o apoio de todos os meus colegas, em especial ao João Silva e ao Pedro Rodrigues pelas longas noites de estudo que se revelaram sempre eficazes na compreensão da matéria, bem como nos resultados dos exames.

Agradeço aos meus colegas de trabalho pela sua disponibilidade incansável em especial aos colegas Adolfo Ameixa, Pedro Costa e Luis Milheiriço pelo seu contributo directo na execução deste projecto.

Um especial agradecimento ao meu irmão Daniel da Luz que embarcou comigo nesta aventura, não me deixando perder o foco, nem a vontade nos momentos mais difíceis, mantendo a sempre a minha atenção virada para este objectivo com a sua insistência e paciência ao longo destes últimos 3 anos.

Resumo e palavras-chave

Este documento foi desenvolvido no âmbito da disciplina Projecto Global, do curso Licenciatura em Informática ministrado no Instituto Superior de Tecnologias Avançadas – ISTEAC.

Foi elaborado um projecto que dotou o Município do Seixal de um Sistema de Gestão de Equipamentos Informáticos (SGEI), que consegue gerir os equipamentos colocados nas várias escolas do concelho do seixal, a sua distribuição nas salas de aula, o respectivo *software* e *hardware* instalado, as suas características, e a assistência técnica prestada. A aplicação permite ainda adicionar, remover ou alterar *software* e *hardware*.

Assente em plataformas gratuitas têm como Sistema de Gestão de Base de Dados (SGBD) o MySQL e como linguagem de programação foi utilizado o PHP, o acesso é via navegador de Internet (*Internet browser*).

Palavras-chave: SGEI, Base de Dados, Gestão de Equipamentos, Assistência Técnica

Abstract

This document was developed under the discipline Global Project Course Degree in Computer Science taught at the Institute of Advanced Technologies - ISTECS.

A project that provided the Municipality of Seixal a Management System Computer Equipment (SGEI), which can manage the equipment placed in various schools of the county Seixal, their distribution in classrooms, our software and hardware installed was prepared , its features, and technical assistance. The application also lets you add, remove, or change software and hardware.

Based on free platforms have as Management System Database (DBMS) like MySQL and PHP programming language was used, access is via Internet browser.

Palavras-chave: SGEI, Data Base, Equipment Management, Technical Assistance

Conteúdo

Dedicatória	III
Agradecimentos.....	IV
Resumo e palavras-chave	V
Abstract	VI
Lista de Ilustrações.....	IX
Abreviaturas.....	X
Introdução.....	1
Capitulo I – Estado da Arte	3
1.1 Introdução.....	3
1.2 Breve História das Bases de Dados	3
1.3 Sistemas de Gestão de bases dados – SGBD	6
1.4 Evolução dos SGBD	7
1.4.1 Processamento Básico de Dados	7
1.4.2 SGF – Sistemas de Gestão de Ficheiros	8
1.4.3 SGBD – Sistemas de Gestão de Bases de Dados	9
1.5 Soluções semelhantes no Mercado	13
1.6 Conclusões	14
Capitulo II – Tecnologia utilizada - Base de Dados Relacionais.....	15
2.1 Propriedades de uma base de dados relacional:.....	15
2.1.1 Chaves.....	15
2.1.1.1 Cardinalidade no Relacionamento	16
2.2 Normalização.....	16
2.2.1 Primeira Forma Normal – 1FN	17
2.2.2 Segunda Forma Normal – 2FN.....	17
2.2.3 Terceira Forma Normal – 3FN	17
2.2.4 Outras	17

Capitulo III – Definição e Manipulação de Dados	18
3.1 DL – Linguagem de Definição de Dados	18
3.2 DML – Linguagem de Manipulação de dados.....	18
3.3 DLL – Biblioteca de Comunicação Dinâmica	19
Capitulo IV – SGEI - Caso Município do Seixal	20
4.1 Contextualização	21
4.2 Recolha e Análise das Especificações	22
4.3 Especificações da Base de Dados.....	23
4.4 Interface gráfica utilizada	26
4.4.1 SGBD.....	26
4.5 Desenho Conceptual	27
4.6 Esquema conceptual	35
4.6.1 Diagrama de Relações	35
4.6.2 Definição e Manipulação de Dados	36
Capitulo V- SGEI - Interface gráfico.....	41
Capitulo VI - SGEI – Testes de Usabilidade	43
Conclusão.....	44
Bibliografia.....	45
Anexos e Apêndices	47

Lista de Ilustrações

Ilustração 1 - Sistema de Gestão Base de Dados	6
Ilustração 2 - A evolução no processamento de dados	7
Ilustração 3 - Sistema de Gestão de Ficheiros	8
Ilustração 4 – o <i>ranking</i> dos SGBD.....	9
Ilustração 5 - Processo de desenho duma base de dados	20
Ilustração 6 – Computador de Implementação	22
Ilustração 7 - Diagrama de Relações.....	35
Ilustração 8 - Criação de tabela recorrendo ao SQL.....	36
Ilustração 9 - PK Chaves Primárias	36
Ilustração 10 - FK - Chaves Estrangeiras	37
Ilustração 11 - Auto_increment.....	37
Ilustração 12 - Inserir dados	38
Ilustração 13 - Eliminação de dados.....	38
Ilustração 14 - Alteração de dados	39
Ilustração 15 - Consulta de dados	39
Ilustração 16 - <i>View</i> Equipamentos	40
Ilustração 17 - Iniciar Sessão em SGEI	41
Ilustração 18 - Consultas em SGEI.....	42

Abreviaturas

BD – Base de Dados

COBOL - *Common Business Oriented Language*

CODASYL - *Conference on Data Systems Languages*

DDL – Linguagem de Definição de Dados

DER - Diagrama de Entidade-Relação

DL/I – *Data Language/Interface*

DLL – Biblioteca de Comunicação Dinâmica

DML – Linguagem de Manipulação de dados

FK - *Foreign Key*

GLPI - *Gestionnaire Libre de Parc Informatique*

ICS - *Information Control System*

MER - Modelo de Entidade-Relação

PHP – *HyperText Preprocessor*

PK - *Primary Key*

SGBD – Sistema de Gestão de Base de Dados

SGEI – Sistema de Gestão de Equipamentos Informáticos

SQL – *Structured Query Language*

UML – *Unified Modeling Language*

WAMP: *Windows, Apache, MySql e PHP.*

Introdução

Os padrões de qualidade impostos às tecnologias de informação em Portugal, através de legislação e regulamentações específicas, não se podem dissociar do advento e desenvolvimento dos sistemas de informação e tecnologias desenvolvidas.

Com esta evolução, surge um grande fluxo de informação e a sua importância para as empresas, é vital aceder à informação de uma forma rápida e eficaz. As bases de dados são consideradas como um conjunto de elementos ligados entre si, que se relacionam de forma lógica, abrangendo um leque alargado de entidades, públicas e privadas.

Os municípios em geral e o município do Seixal em particular, em virtude das competências que assumem na gestão, fornecimento e assistência técnica de equipamentos informáticos às Escolas Básicas de 1º Ciclo e Jardins de Infância, têm necessidade de utilizar um sistema que consiga gerir os equipamentos colocados nas várias escolas do concelho do Seixal, bem como a sua distribuição nas salas de aula. No entanto, e apesar de existirem inúmeras soluções no mercado para gerir e controlar os equipamentos e respectivas assistências técnicas, as condições financeiras das autarquias não são favoráveis à aquisição dessas soluções (Livro Branco do Sector Empresarial Local). Assim e neste enquadramento, a inexistência de um sistema que controle as medidas urgentes relativas aos equipamentos informáticos, de carácter técnico, logístico e económico a adaptar em resposta aos problemas concretos das escolas existentes no município do Seixal, inviabilizam ou remetem para segundo plano, compreensivelmente, a garantia e eficácia da qualidade dos meios e serviços técnicos.

Esta dissertação expõe a importância das Bases de Dados e dos Sistemas de Gestão de Base Dados e apresenta um sistema de Gestão de Equipamentos Informáticos, a implementar no município do Seixal, que terá em consideração as especificações técnicas regulamentadas em matéria de gestão de base dados.

Tendo como principal objectivo a redução de custos, será demonstrado, como ferramentas gratuitas respondem às necessidades apresentadas e às

questões colocadas de forma simples e viável. As ferramentas utilizadas serão o MySQL Workbench e o interface gráfico será desenvolvido em linguagem PHP.

Para uma melhor compreensão de toda a temática do projecto, a informação sobre o mesmo foi dividida na seguinte estrutura:

Capítulo I – Estado da Arte: reflexão e apresentação sobre o estado actual do objecto de estudo, as principais aplicações e tendências relativamente aos SGBD;

Capítulo II – Tecnologia utilizada - Base de Dados Relacionais: principais conceitos da tecnologia utilizada;

Capítulo III – Definição e Manipulação de Dados a estrutura e armazenamento das informações pertinentes a tabelas, consultas, registos, definições de dados e atributos;

Capítulo IV – SGEI - Caso Município do Seixal: desenvolvimento e implementação da base de dados;

Capítulo V- SGEI - Interface gráfico: desenvolvimento e desenho da aplicação e respectivos testes de usabilidade;

Capítulo I – Estado da Arte

1.1 Introdução

Sistemas de Gestão de bases de dados são programas que permitem criar e manipular bases de dados, em que os dados são estruturados com independência (isto é, de forma a ser possível alterar a estrutura dos dados de uma base de dados, quer a nível físico quer a nível conceptual, sem que isso implique a necessidade de reformular o programa que opera com os dados) relativamente aos programas de aplicação que os manipulam. (Thomas M. Connolly, Carolyn E. Begg, Anne D. Strachan, 1996)

1.2 Breve História das Bases de Dados

Na década de 1950 e nos primeiros anos da década de 1960, o armazenamento e acesso a dados era ainda bastante rudimentar. Existiam alguns projectos mais avançados em desenvolvimento, mas restritos a um pequeno número de pessoas, a grande maioria armazenava dados em arquivos de texto, com um tamanho fixo e apenas com simples operações de leitura e escrita. Embora esta fosse uma metodologia bastante simples para armazenamento de dados, não tardou para que se percebesse que não era a forma mais eficiente na maioria dos casos.

Em 1957, o Departamento de Defesa dos Estados Unidos da América, inaugurou a *Conference on Data Systems Languages* (Conferência sobre as Linguagens de Sistemas de Dados), também conhecida simplesmente por CODASYL, com o objectivo de desenvolver linguagens de programação e famosa pela criação da linguagem de programação COBOL, CODASYL também foi responsável pela criação da primeira base de dados.

Em 1963, numa conferência intitulada "*Development and Management of a Computer-Centered Data Base*" ("Desenvolvimento e Gestão de uma Base de Dados para Computadores"), foi concebido e definido o termo base de dados, com a seguinte definição: conjunto de arquivos (tabelas), onde um arquivo é uma colecção

ordenada de registos (linhas), e um registo consiste em uma ou mais chaves e dados.

Em 1965, CODASYL formou um grupo chamado de *List Processing Task Force* (Força Tarefa de Processamento de Listas), que mais tarde tornou-se o *Data Base Task Group* (Grupo Tarefa de Base de Dados), este grupo emitiu um importante relatório em 1971 delineando o *Network Data Model* (Modelo de Dados em Rede), também conhecido como *CODASYL Data Model* (Modelo de Dados CODASYL), ou ainda *DBTG Data Model* (Modelo de Dados DBTG). Esse modelo de dados definiu muitos conceitos importantes para base de dados, que foram posteriormente incorporados na linguagem de programação COBOL. Na mesma época em que o CODASYL criava o Modelo de Dados em Rede, outros esforços eram feitos no sentido de criar a primeira base de dados hierárquica. (Conference on Data Systems Languages Records, 1959-1987)

Em 1966, membros da IBM, *North American Rockwell* e a *Caterpillar Tractor* juntaram-se para começar o projecto de desenvolvimento do *Information Control System - ICS* (Sistema de Controlo de Informações) e da *Data Language/I - DL/I* (Linguagem de Dados/I). O ICS era responsável pelo armazenamento e recuperação de dados, enquanto a DL/I era responsável pela linguagem de consulta necessária para aceder os dados.

Em 1970, Edgar Codd, um cientista britânico que trabalhava na IBM, publicou um importante artigo chamado "*A Relational Model of Data for Large Shared Data Banks*" ("Um modelo relacional de dados para grandes bases de dados compartilhados"), no qual, ele introduziu a ideia de *relational model* (modelo relacional). Neste modelo, Codd enfatizou a importância da separação de tipos de dados genéricos primitivos das implementações de tipos de dados específicos implementados em um determinado modelo de computador e expôs uma linguagem simples de consulta de alto-nível para acesso a dados. Este modelo permitiria que os programadores utilizassem operações em um conjunto completo de dados de uma única vez, ao invés de trabalhar com um registo de cada vez.

Uma base de dados relacional consiste num conjunto de tabelas a duas dimensões. Ao todo existem apenas 4 conceitos a compreender: tabela, linha, coluna e campo. (Pedro Bizarro, 2000)

Em 1976 uma abordagem conhecida como Modelo de Entidade-Relação (MER) foi criada pelo Dr. Pin-Shan, este modelo consiste na representação conceitual abstracta da estrutura de dados, gerando um diagrama conhecido como Diagrama de Entidade-Relação (DER), cujas convenções devem ser conhecidas por todos os participantes do processo de análise, projecto e implementação de uma base de dados.

Baseados nas ideias de Edgar Codd surgiram dois projectos. O "*System R*" ("Sistema R") desenvolvido pela IBM e o "*Ingres*" oriundo da Universidade da Califórnia em Berkeley. Durante o desenvolvimento do "Sistema R", uma nova linguagem de consulta conhecida como *Structured Query Language* (SQL) (Linguagem Estruturada de Consulta) foi criada. O projecto "Ingres" era bastante similar ao "Sistema R", mas com uma vantagem que aumentou o seu tempo de utilização, o código do "Ingres" foi disponibilizado publicamente, ainda que mais tarde tenha sido comprado e comercializado pela *Computer Associates* (na década de 1980). (Peter Pin-Shan,1976)

Na ultima década, guiadas pelas ideias deixadas, houve uma grande evolução das bases de dados, surgindo bases de dados como *Oracle*, Microsoft SQL Server, MySQL e LDAP. Muito mais evoluídas permitem manipulação de grande volume de transacções e manuseamento de grandes quantidades de dados, além de oferecerem altos índices de escalabilidade e confiabilidade. (Heuser, 1999)

1.3 Sistemas de Gestão de bases dados – SGBD

“O programa ou grupo de programas que fornece acesso a uma base de dados é conhecido como um sistema de gestão de base de dados” (Turban,2003). Esta definição, apesar de simples, enfatiza que um SGBD do inglês DBMS (*data base management system*) é, antes de mais, um *software* que permite aceder a uma base de dados, como podemos verificar na figura abaixo apresentada.

Uma outra definição mais detalhada é dada por O'Brien: “Um sistema de gestão de base de dados é um conjunto de programas de computador que controla a criação, manutenção e uso das bases de dados por uma organização e por seus utilizadores finais” (O'Brien,2003).

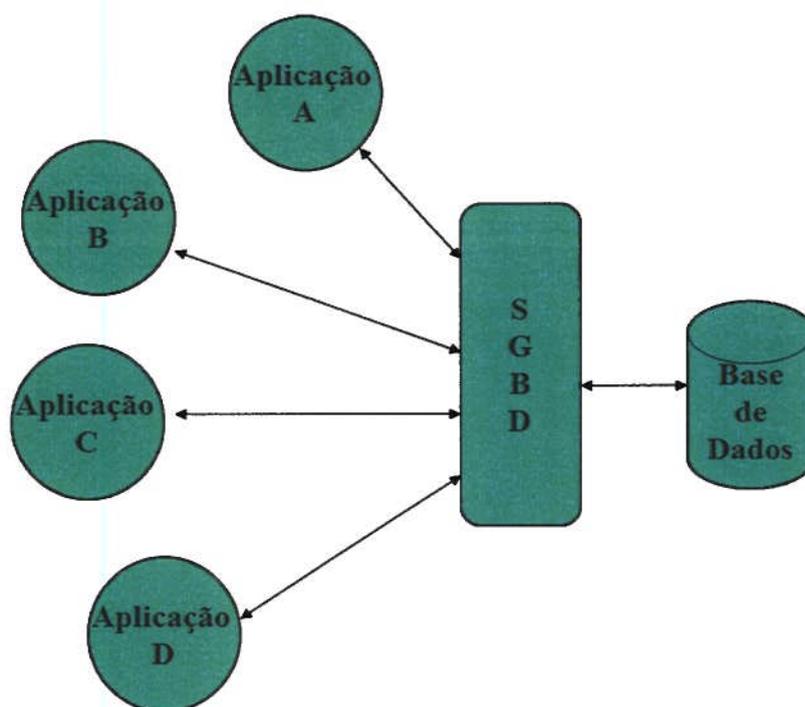


Ilustração 1 - Sistema de Gestão Base de Dados

1.4 Evolução dos SGBD

A evolução temporal do processamento de dados pode ser vista da seguinte forma:

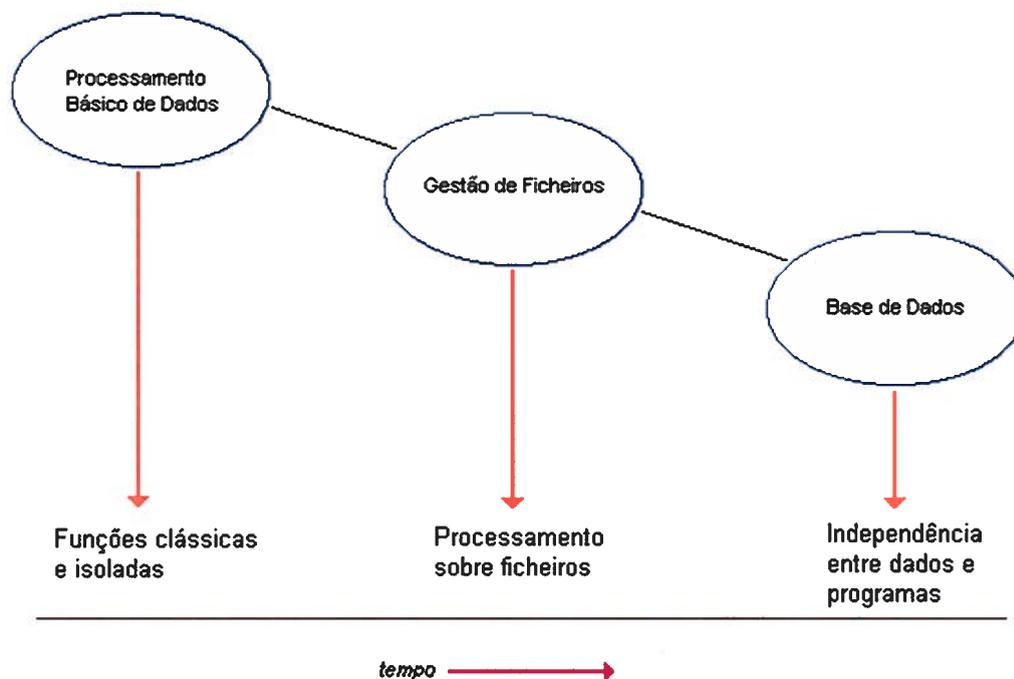


Ilustração 2 - A evolução no processamento de dados

Fonte: http://www.portalwebmarketing.com/Tecnologia/mdr_introducao_modelo_dados_relacional

1.4.1 Processamento Básico de Dados

O Processamento básico (ficheiros elementares, anos 50/60). Caracterizou-se por trabalhos isolados de programação. Cada programa tinha os seus ficheiros.

A manipulação dos dados estava reduzida a simples funções, tais como: ordenação, classificação, e realização de somatórios. O *Software* pouco mais fazia do que a entrada e saída sobre o mecanismo de armazenamento, normalmente numa banda magnética. Qualquer alteração à forma como os dados deveriam estar armazenados, implicava modificações nos programas, a sua recompilação e teste. A alteração num registo conduzia à criação dum novo ficheiro, o antigo continuava a existir e assim sucessivamente. A grande maioria dos ficheiros era utilizada numa só aplicação. “Havia, portanto, um alto nível de redundância, com os mesmos dados multiplicados por n ficheiros” (C.Caldeira,2008).

1.4.2 SGF – Sistemas de Gestão de Ficheiros

Os antecessores dos SGBD foram os SGF (Sistema de Gestão de ficheiros).

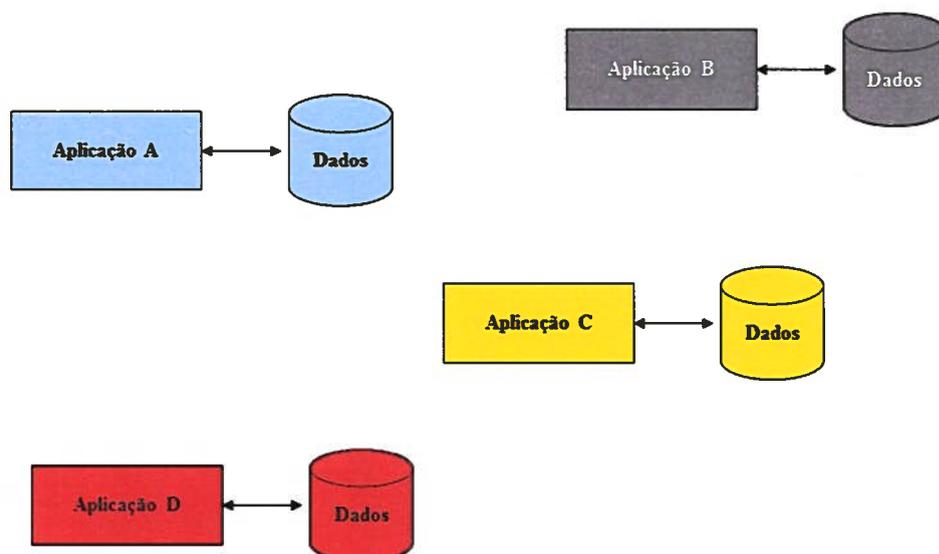


Ilustração 3 - Sistema de Gestão de Ficheiros

Dos vários SGF existentes destacaram-se os desenvolvidos com recurso ao COBOL, estes vieram automatizar tarefas até então realizadas manualmente, tornando-as essencialmente mais rápidas, no entanto tinham enormes deficiências, tais como, o facto dos mesmos dados serem armazenados em ficheiros diferentes, criados pelas diversas aplicações. Para além deste facto, os SGF caracterizavam-se ainda por uma difícil gestão, uma vez que, numa organização, cada departamento tratava e armazenava a informação de modo diferente e a falta de comunicação das alterações aos outros departamentos originava a inconsistências nos dados.

Perante este cenário houve um crescimento exponencial da informação e por consequência do número de base de dados. Era necessário um sistema mais eficaz e eficiente, que permitisse a eliminação dos dados redundantes, um sistema caracterizado por ser a única entidade que manipulava a base de dados, ou seja, centralizava em si o acesso à base de dados, implementando um nível de abstracção para as aplicações que utilizavam os dados por eles disponibilizados e colmatando assim as limitações dos sistemas utilizados anteriormente SGF (J.L.Pereira,1998).

1.4.3 SGBD – Sistemas de Gestão de Bases de Dados

Os SGBD foram projectados para gerir grandes grupos de informações. A gestão de dados envolve a definição de estruturas para armazenamento de informações e o fornecimento de mecanismos para manipulá-las. Além dessas informações, o SGBD deve fornecer total segurança das informações armazenadas e contra tentativas de acessos não-autorizados.

Com grande evolução e optimização durante as ultimas décadas, entre muitos destacam-se 3 grandes sistemas de gestão de Base de Dados:

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



204 systems in ranking, November 2013

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Oracle ↗	Relational DBMS	1617.19	+33.35
2.	2.	MySQL ↗	Relational DBMS	1254.27	-77.07
3.	3.	Microsoft SQL Server ↗	Relational DBMS	1234.46	+27.45
4.	4.	PostgreSQL ↗	Relational DBMS	190.83	+13.82
5.	5.	DB2 ↗	Relational DBMS	165.90	-9.93
6.	6.	MongoDB ↗	Document store	161.87	+12.40
7.	7.	Microsoft Access ↗	Relational DBMS	141.60	-0.89
8.	8.	SQLite ↗	Relational DBMS	78.78	+0.90
9.	9.	Sybase ↗	Relational DBMS	77.75	+4.09
10.	10.	Teradata ↗	Relational DBMS	60.12	+5.70
11.	11.	Cassandra ↗	Wide column store	57.58	+4.66
12.	12.	Solr ↗	Search engine	46.53	+3.20
13.	13.	Redis ↗	Key-value store	40.57	+4.56
14.	14.	FileMaker ↗	Relational DBMS	35.10	-0.21
15.	↑ 17.	Memcached ↗	Key-value store	28.50	+0.38

Ilustração 4 – o ranking dos SGBD

Fonte: DB-Engines Ranking¹

¹ <http://db-engines.com/en/ranking>

Oracle: Ao longo dos últimos 30 anos a Oracle tem vindo a aperfeiçoar o seu principal produto, mantendo-se assim líder de mercado. Procurando destacar-se dos seus concorrentes adicionando novas funcionalidades ao seu SGBD, por exemplo, com a versão 8, lançado em 1997, foi adicionado a possibilidade de armazenamento de objectos na base de dados, já a versão 8i, lançada em 1999, iniciou melhorias que colocaram o Oracle8i focado na Internet. O Oracle8i iniciou a tradição de *marketing*² da Oracle de adicionar uma letra como sufixo no nome da versão, ou seja, o "i" presente no Oracle8i serve para ressaltar o foco na Internet. O Oracle8i, focado na Internet, foi o primeiro objecto da base de dados relacional (ORDBMS), sendo apenas um esboço das características para o então Oracle9i, lançado em 2000 (ano posterior ao de lançamento do Oracle8i) e considerado por publicações oficiais da própria Oracle (Oracle Press Books) como a sua primeira base de dados relacional (RDBMS) que oferece uma plataforma de suporte e desenvolvimento para a Internet. Além de oferecer toda uma infra-estrutura para o suporte ao modelo Internet, o Oracle9i inovou com uma ferramenta para Servidores Paralelos chamada *Real Application Clusters*, melhorou o suporte a documentos em XML e realizou melhorias fundamentais de performance e gestão no SGBD para o trabalho via Internet.

Apesar do pioneirismo do Oracle8i na introdução de característica Internet (JServer e OAS), o Oracle9i foi responsável por oferecer um ambiente completo de desenvolvimento de aplicações voltadas para a Internet.

Os produtos da Oracle são licenciados com base principalmente em dois parâmetros: *Named Utilizador Plus* ou *Processor*, sendo o primeiro utilizado em ambientes cujos utilizadores e/ou dispositivos podem ser facilmente identificados e contados e o segundo (*Processor*), utilizado em ambientes cujos utilizadores não podem ser identificados e contados com facilidade. (Oracle,2013)

² Em português, *marketing* costuma traduzir-se para *mercadologia* (Dicionário da Língua Portuguesa na Infopédia, acesso em 11 de Novembro de 2013)

MySQL: é um sistema de gestão de base de dados, que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês, *Structured Query Language*) como interface. Segundo a DB-Engines Ranking³ o *MySQL* é atualmente uma das bases de dados mais populares, com mais de 10 milhões de instalações pelo mundo⁴. O sucesso do *MySQL* deve-se em grande medida à fácil integração com o PHP, incluído quase que obrigatoriamente, nos pacotes de hospedagem de *sites*⁵ da Internet oferecidos actualmente. “Empresas como Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics e Texas Instruments usam o *MySQL* em aplicações de missão crítica. A Wikipédia é um exemplo de utilização do *MySQL* em *sites* de grande audiência”⁶.

As principais características da linguagem são: Portabilidade (suporta praticamente qualquer plataforma actual); Compatibilidade; Excelente desempenho e estabilidade; Pouco exigente quanto a recursos de novos *hardwares*; É um Software Livre com base na GNU GPL ou simplesmente GPL⁷ (*General Public License* - Licença Pública Geral); Suporta *Triggers*; Suporta Cursores; Suporta *Stored Procedures* e *Functions*; Replicação e é facilmente configurável;

Microsoft SQL Server: é um sistema de gestão de base de dados, desenvolvido pela Microsoft, criado em parceria com a Sybase em 1988 inicialmente para a plataforma OS/21. Esta parceria durou até 1994, com o lançamento da versão para Windows NT apenas a Microsoft mantém a manutenção do produto. Como uma base de dados, é um produto de *software*, cuja principal função é a de armazenar e recuperar dados solicitados por outras aplicações de software, estejam em execução no mesmo computador ou em execução em outro computador através de uma rede (incluindo a Internet), há pelo menos uma dúzia de diferentes edições do *Microsoft SQL Server* destinadas a públicos diferentes e para diferentes cargas de trabalho (variando de pequenas aplicações que armazenam e recuperam dados no mesmo computador, a milhões de utilizadores e computadores que acedam a

³ <http://db-engines.com/en/ranking>

⁴ <http://www.mysql.com/why-mysql/>

⁵ sítio electrónico.

⁶ <http://www.mysql.com/why-mysql/case-studies/>

⁷ Designação da licença para *software* livre idealizada por Richard Matthew Stallman em 1989, no âmbito do projecto GNU GPL da Free Software Foundation (FSF).

grandes quantidades de dados a partir da *Internet* ao mesmo tempo). As suas linguagens de consulta primárias são *T-SQL* e *ANSI SQL*.

Com o *SQL Server 2012*, a Microsoft continua a oferecer aos clientes uma variedade de opções de licenciamento (Nielsen Paul, 2005).

1.5 Soluções semelhantes no Mercado

Os sistemas de Gestão de Equipamentos informáticos existentes no mercado não se restringem apenas ao controlo de equipamentos e avarias, como o nosso SGEI. Na sua maioria são sistemas complexos, com grande aplicabilidade e com custos associados, tendo como base o nosso único e principal objectivo, a inexistência de custos, eis alguns exemplos de soluções gratuitas existentes no mercado.

GLPI⁸ - Gestão total do parque informático (*Gestionnaire libre de parc informatique*) - é um sistema de gestão de recursos que oferece uma interface muito intuitiva e um conjunto de funcionalidades que permitem manter um conjunto de informações sobre todos os recursos de uma rede. Esta solução não é viável para resposta ao meu projecto na medida em que para que tirar partido desta, é necessário instalar esta solução numa rede e no nosso caso o controlo que queremos fazer é o de equipamentos fora da nossa rede.

SpiceWorks IT Desktop⁹ - é uma aplicação gratuita que inclui várias funcionalidades, das quais se destacam a monitorização de redes, inventário da rede, *software* de *HelpDesk*, auditoria de redes, etc. É uma ferramenta bastante interessante, que disponibiliza as funcionalidades essenciais para se gerir mais comodamente qualquer rede informática. Mais uma vez não, não podemos usufruir das suas funcionalidades devido ao nosso parque informático estar em modo *offline* (fora da nossa rede).

⁸ <http://www.glpi-project.org/spip.php?article315>

⁹ <http://www.spiceworks.com/>

1.6 Conclusões

Como podemos verificar ao longo deste estudo houve uma tentativa de otimizar e garantir a eficácia das bases dados e respectivos sistemas de gestão das mesmas, podemos afirmar que a informação é de vital importância para as instituições. Não é exagerado falar que a informação é um dos activos mais importantes, pois está directamente ligada ao sucesso.

Realço e chamo a atenção para o uso da palavra informação e não de dados. Muitos são os que utilizam estas duas palavras como se fossem sinónimas, quando na realidade não o são. Para fazer a devida distinção entre os conceitos, peço licença para utilizar um pequeno exemplo:

Supondo que temos uma mina onde os mineiros procuram ouro conhecido por todos pelo seu grande valor, no entanto, este misturado a outros minerais não têm valor nenhum. O ouro é retirado do solo em seu estado bruto, é separado dos demais materiais e vendido numa forma rústica, porém, pura. O ourives, por sua vez, recebe o ouro oriundo da mina e transforma-o numa jóia de grande valor, excedendo várias vezes o valor da matéria-prima.

Neste simples exemplo, o ouro em seu estado natural compara-se a uma massa de dados, sem valor. Já a jóia feita com o ouro, representa a informação, uma lapidação de valor feita com os dados.

Assim como neste exemplo, a qualidade dos dados determina a qualidade da informação. Dados incorrectos ou parcialmente correctos e mal estruturados comprometem as informações que podem ser extraídas a partir deles e causam transtornos e prejuízos a quem deles faz uso.

Outro factor importante é a protecção desses mesmos dados, estes nunca devem estar desprotegidos ou acessíveis a qualquer um.

Agora, então, pergunto: Como armazenar, proteger e disponibilizar grandes volumes de dados consistentes aos mais diversos interessados?

A resposta está no uso de sistemas de gestão de base de dados (SGBD), um conjunto de programas responsáveis por, dentre outras coisas, armazenar e proteger massas de dados.

Capítulo II – Tecnologia utilizada - Base de Dados Relacionais

Antes de apresentar a solução para o projecto aqui proposto, irei sucintamente falar sobre os principais conceitos da tecnologia que irei usar, assim temos:

Base de dados relacional é vista como um conjunto de tabelas bidimensionais, tendo por base 4 conceitos: Tabelas; Colunas; Linhas e Campos; (Caldeira C.,2011)

2.1 Propriedades de uma base de dados relacional:

Uma base de dados relacional é uma colecção de tabelas, sendo uma tabela tem o formato colunas/linhas; Existe um conjunto de operadores que permitem separar e combinar as tabelas; As ligações são efectuadas com base nos dados não existindo apontadores. O utilizador não sabe "onde", nem como os dados estão fisicamente organizados; Existe total independência dos dados (os dados são os mesmos para qualquer aplicação que os utiliza).

Uma tabela isolada deve satisfazer as seguintes propriedades:

Não existirem linhas em duplicado; Não existirem nomes de colunas em duplicado; A ordem das linhas não é significativa; A ordem das colunas não é significativa; As tabelas relacionam-se umas as outras através de **chaves**; (Caldeira C.,2011)

2.1.1 Chaves

Uma chave é um conjunto de um ou mais atributos que determinam a unicidade de cada registo.

Chave primária: (*PK - Primary Key*) é a chave que identifica cada registo dando-lhe unicidade. A chave primária nunca se repetirá.

Chave Estrangeira: (*FK - Foreign Key*) é a chave formada através de um relacionamento com a chave primária de outra tabela. Define um relacionamento entre as tabelas e pode ocorrer repetidas vezes. Caso a chave primária seja composta na origem, a chave estrangeira também o será. (Caldeira C.,2011)

2.1.1.1 Cardinalidade no Relacionamento

O número máximo e mínimo de ocorrências de uma entidade que estão associadas às ocorrências de outra entidade que participa no relacionamento. Ou seja, a cardinalidade é importante para ajudar a definir o relacionamento, pois define o número de ocorrências num relacionamento.

Um para um (**1 para 1**) - indica que as tabelas têm relação unívoca entre si. Escolhemos qual tabela vai receber a chave estrangeira;

Um para muitos (**1 para N**) - a chave primária da tabela que tem o lado 1 vai para a tabela do lado N. no lado N será chamada de chave estrangeira;

Muitos para muitos (**N para N**) - quando tabelas têm entre si relação N.N, é necessário criar uma nova tabela com as chaves primárias das tabelas envolvidas, ficando assim uma chave composta, ou seja, formada por diversos campos-chave de outras tabelas. A relação reduz-se para uma relação 1.N, sendo que o lado n ficará com a nova tabela criada. (Caldeira C.,2011)

2.2 Normalização

A teoria da normalização (conjunto de normas associadas ao modelo relacional) garante que um mesmo facto não apareça em mais de um sítio.

O objectivo da normalização é evitar os problemas provocados por falhas no projecto, bem como eliminar a "mistura de assuntos" e as correspondentes repetições desnecessárias de dados.

O Processo de Normalização aplica uma série de Regras sobre as Tabelas de uma Base de Dados, para verificar se estas estão correctamente projectadas.

Embora existam 5 formas normais (ou regras de Normalização), na prática usamos um conjunto de 3 Formas Normais. Normalmente após a aplicação das Regras de Normalização, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente existente. Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção. (Sobrinho A.P.,2011)

2.2.1 Primeira Forma Normal – 1FN

"Uma Tabela está na Primeira Forma Normal quando seus atributos não contêm grupos de Repetição".

Por isso dizemos que uma Tabela que possui Grupos de Repetição não está na Primeira Forma Normal.

2.2.2 Segunda Forma Normal – 2FN

"Ocorre quando a chave Primária é composta por mais de um campo. Neste caso, devemos observar se todos os campos que não fazem parte da chave dependem de todos os campos que compõem a chave. Se algum campo depender somente da parte da chave composta, então este campo deve pertencer a outra tabela".

2.2.3 Terceira Forma Normal – 3FN

"Na definição dos campos de uma entidade podem ocorrer casos em que um campo não seja dependente directamente da chave primária ou de parte dela, mas sim dependente de um outro campo da tabela, campo este que não a Chave Primária". Quando isto ocorre, dizemos que a tabela não está na Terceira Forma Normal.

2.2.4 Outras

FNBC (Forma Normal Boyce-Codd), 4FN e 5FN

Os temas neste capítulo foram abordados de forma sucinta, tendo em conta apenas a necessidade deste projecto, podem ser seguidos em vários manuais referenciados na bibliografia em referência.

Capítulo III – Definição e Manipulação de Dados

As bases de dados possuem uma estrutura que armazena as informações pertinentes a tabelas, consultas, registros, definições de dados e atributos. Apesar de possuírem nomes ou identificações diferentes, todo SGBD possui um esquema e um subesquema.

Esquema da Base de Dados - armazena o “projeto geral” (estrutura) da base de dados e possui as seguintes características: não muda com frequência e há um esquema para cada nível de abstração e um subesquema para cada visão do utilizador.

Para que seja possível transformar um modelo conceitual/lógico num modelo físico é necesssária a utilização de uma linguagem apropriada, esta linguagem é conhecida como DDL – *Data Definition Language*. E a sua transcrição ocorre por meio de uma ferramenta CASE – *Computer Aided Software Engineer* (algo como: linguagem de construção para auxiliar ao processamento). (Alvaro Sobrinho, 1999)

3.1 DL – Linguagem de Definição de Dados

Permite especificar o esquema da base de dados, através de um conjunto de definições de dados. A compilação dos comandos em DDL é armazenada no dicionário (ou diretório) de dados. Neste ponto são criados os metadados¹⁰ que irão compor o esquema do projeto que estiver em transcrição. (Alvaro Sobrinho, 1999)

3.2 DML – Linguagem de Manipulação de dados

A manipulação dos dados acontece através da recuperação da informação armazenada, da inserção de novas informações, exclusão de informações e modificação de dados armazenados.

Para que os eventos acima sejam possíveis é necessário utilizar uma linguagem apropriada para interagir com a base de dados de tal maneira que, através da leitura do esquema, consiga entender o dado e a sua posição na tabela. Esta linguagem é conhecida como DML – *Data Manipulation Language* – Linguagem e Manipulação de Dados que permite ao utilizador aceder ou manipular

¹⁰ Metadados, por definição, são dados que definem outros dados. Assim os comandos compilados são metadados dos comandos originais.

os dados, vendo-os da forma como foram definidos no nível de abstração mais alto do modelo de dados utilizado.

O processo de interação no SGBD é feito através de um conjunto de procedimentos que denominados por *QUERY* (Consulta). Portanto, uma consulta (“query”) é um conjunto de comandos que requisitam uma recuperação de informação à base de dados. Por essa razão, a parte de uma DML que envolve recuperação de informação é chamada linguagem de consulta. (Alvaro Sobrinho, 1999)

3.3 DLL – Biblioteca de Comunicação Dinâmica

As consultas podem ser construídas diretamente no SGBD, ou podem ser feitas recorrendo ao uso de uma linguagem de alto nível (orientada a objeto ou não) para efetuar a interação com a base de dados. Neste caso, alguns programas fazem uso de uma DLL – *Dynamic Link Library* – Biblioteca de Comunicação Dinâmica, que é instalada no computador, e nada mais é do que códigos que várias aplicações compartilham, o que reduz de forma significativa a utilização da memória do computador. (Alvaro Sobrinho, 1999)

Capítulo IV – SGEI - Caso Município do Seixal

O desenvolvimento de uma base de dados é uma tarefa complexa, dado que não existe um único grupo de especificações¹¹, mas vários conjuntos que se sobrepõem uns aos outros.

O primeiro passo centra-se na recolha e análise das especificações. Para isso, foram feitas uma série de entrevistas aos utilizadores, e reunida toda a documentação necessária para a execução da actividade em estudo.

Na ilustração abaixo podemos ver uma descrição simplificada do processo de desenho de uma base de dados.



Ilustração 5 - Processo de desenho duma base de dados

Fonte: <http://www.portalwebmarketing.com/Tecnologia/Introdu%C3%A7%C3%A3oaoModelodeDadosRelacional/MDRDesenvolvimentodeBasedeDados/tabid/657/Default.aspx>

¹¹ Aquilo que o utilizador final pretende que a base de dados faça.

4.1 Contextualização

Actualmente, a assistência prestada às Escolas Básicas de 1º Ciclo e Jardins de Infância no município do Seixal circula via correio electrónico, formulários, faxes e em alguns casos via telefonemas sem qualquer registo. Cada técnico gere e armazena os dados das assistências que ele próprio efectua. Relativamente ao controlo dos equipamentos existentes nas escolas ou em *stock*, este é feito via ficheiros de dados individuais. Face a este cenário e perante o estudo apresentado neste trabalho, a solução mais eficaz será a criação de uma base de dados e por consequência de um sistema de gestão para essa mesma base de dados.

Face ao exposto e após entrevistar os vários intervenientes no processo, os requisitos apresentados pelo município são as seguintes: Custos reduzidos com o projecto, de preferência utilizar os meios ao dispor e/ou ferramentas gratuitas, uma vez que existem várias escolas por agrupamento escolar, é necessário saber a quantidade de equipamentos colocado em cada escola e as respectivas características, o *Software* e *Hardware* e a assistência técnica, tal como a descrição das avarias associadas aos equipamentos; e os técnicos que efectuaram a assistência solicitada;

Foi efectuada uma análise aos sistemas existentes no mercado, vista a sua complexidade e respectivo licenciamento (custo associado), e sendo o principal objectivo deste projecto a redução de custos, será demonstrado, como ferramentas gratuitas respondem às necessidades apresentadas e às questões colocadas de forma simples e viável. As ferramentas utilizadas serão o MySQL Workbench e o interface gráfico será desenvolvido recorrendo à Linguagem PHP.

No final deste trabalho teremos uma solução em que os mesmos dados vão poder ser utilizados, provavelmente com diferentes níveis de agregação, por utilizadores a diferentes níveis na organização, não só pelos operacionais, mas também por gestores aos níveis tático e estratégico.

4.2 Recolha e Análise das Especificações

Requisitos necessários ao desenvolvimento e implementação deste projecto já existem, são eles apenas um computador com as seguintes características:

Processador Intel Core i3 CPU 3.10GHz, com 4Gb RAM, sistema operativo Windows 7 Professional N e um disco de 500GB.

Imagem abaixo apresenta as características do equipamento em utilização:



Ilustração 6 – Computador de Implementação

4.3 Especificações da Base de Dados

Após várias entrevistas com os utilizadores/intervenientes no processo, apuramos que a base de dados deveria responder directamente às seguintes questões:

1. Escolas por Agrupamento.
2. Coordenador(a) responsável pela escola.
3. Equipamentos por sala /escola.
4. Salas sem equipamento informático.
5. Numero total de avarias reportadas:
6. Numero total de avarias por equipamento.
7. Numero total de reparações por técnico.
8. *Software* instalado por PC.
9. Hardware instalado por PC.
10. Características dos Equipamentos.

Caldeira (2011) afirma que “Uma base de dados é composta por um conjunto de tabelas e associações entre as tabelas. A associação entre os dados é o ponto forte dos sistemas relacionais.” Tal como dito anteriormente será criada uma base de dados relacional e os dados serão manipulados através da linguagem SQL (*Structured Query Language*). Para o funcionamento base e estrutural foram recolhidas as características fundamentais dos intervenientes activos na aplicação, assim temos:

Sobre Escola: Uma escola é caracterizada pelo Nome, Morada, Telefone, Correio electrónico e Código Postal;

Uma escola tem vários equipamentos informáticos, sendo possível adicionar/remover equipamentos à mesma, também existe um responsável e um(a) coordenador(a); Esta pertencerá a um agrupamento e um agrupamento possuirá várias escolas. Cada escola pode registar avarias dos equipamentos afectos a si.

Sobre Agrupamento: O Agrupamento é caracterizado pelo Nome, Morada, Telefone e Código Postal;

Um agrupamento tem várias escolas mas uma escola só pertence a um agrupamento;

Sobre Equipamento: Os equipamentos são caracterizados pela Etiqueta ou Nº de inventário, Nº de série, Modelo, Marca, Categoria e Local instalação;

Um equipamento é identificado pela sua categoria, pode ser uma impressora, *switch*, *router*, monitor ou *scanner*, este (PC) pode ter *hardware* e *software*, e podem ser instalados/colocados ou retirados, que por sua vez está numa escola – colocado em salas, e uma sala pode ter vários equipamentos.

Sobre avaria: A avaria é caracterizada pela Escola associada, Tipo, Data criação, Equipamento associado e Data resolução;

Uma avaria pode acontecer a vários equipamentos, e um equipamento pode ter várias avarias que deve ter uma descrição, com um tipo associado, ou seja pode ser: *software*, *hardware*, vírus, internet ou outros. A avaria é registada por uma escola e resolvida por um técnico.

Sobre Técnico: O técnico é caracterizado pelo Nome, Telefone serviço, Correio electrónico e Categoria;

Um técnico pode ter várias avarias para resolver, depois de resolver a avaria pode fazer uma descrição da resolução, encerrar as avarias através do preenchimento da data de resolução e pode consultar todas as avarias existentes.

4.4 Interface gráfica utilizada

4.4.1 SGBD

Como já referido anteriormente, para a manipulação das informações nas bases de dados e desenvolvimento de programas aplicativos, é necessário a utilização de um SGBD, assim sendo, e de acordo com o exposto neste projecto foi utilizado o MySQL, por ser um sistema de gestão de base de dados relacionais de código aberto, e recorreu-se à linguagem *SQL* por este sistema utilizada. Os requisitos de utilização deste SGBD são um servidor e uma aplicação cliente.

Sendo o servidor o responsável por armazenar os dados, responder aos pedidos, controlar a consistência dos dados, bem como a execução de transacções concomitantes entre outras e o cliente de comunicar com este através do *SQL*.

Foi utilizado o MySQL Workbench, por ser gratuito e possuir o servidor e a interface gráfica cliente no que respeita à BD, de forma a garantir o funcionamento deste foi criada ligação entre o nosso SGBD e a nossa BD, atribuído o nome SGEI, e utilizado o método Standard TCP/IP para o endereço local 127.0.0.1 (*localhost*) porto 3306 e utilizador *root*.

Por fim foi criada a base de dados, com o mesmo nome SGEI, recorrendo a um dos assistentes existentes, que gerou o seguinte código:

```
CREATE SCHEMA 'sgei' DEFAULT CHARACTER SET utf8;
```

Será explicado em Definição e Manipulação de dados como foram criadas e manuseadas as tabelas.

4.5 Desenho Conceptual

Consiste na definição de um modelo de dados conceptual, incluindo a descrição das entidades da BD, dos atributos das entidades, dos relacionamentos entre entidades e das possíveis restrições. Desta forma, facilita-se o entendimento dos dados por parte dos não-especialistas, a detecção de conflitos e simplifica eventuais correcções a fazer, simplificando a implementação.

Tabela - Agrupamento

Atributos

idagrupamento int(11) not null auto_increment,
agrup_nome varchar(100) not null,
agrup_morada varchar(300) not null,
agrup_codpost varchar(25),
agrup_telefone varchar(10),
agrup_email varchar(80),

Chave Primária

Idagrupamento

Tabela – Escola

Atributos

idescola int(11) not null auto_increment,
escola_nome varchar(100) not null,
escola_morada varchar(300) not null,
escola_codpost varchar(25),
escola_telefone varchar(10),
escola_email varchar(80),
escola_coordenacao varchar(100) not null,
escola_observacao varchar(500),
idescolaagrupamento int(11) not null,

Chave Primária

Idescola

Chave Estrangeira

idescolaagrupamento:Agrupamento(idagrupamento)

Tabela – SalaAtributosidsala int(11) not null auto_increment,
sala_nome varchar(100) not null,
idsalaescola int (11) not null,Chave Primária

idsala

Chave Estrangeira

idsalaescola:Escola(idescola)

Tabela – EquipamentoAtributosidequipamento int(11) not null auto_increment,
etiqueta int(11) not null,
numserie varchar(20) not null,
equip_observacoes varchar(300),
idequipamentosala int(11),
idequipamentocategoria int(11),
idequipamentomodelo int(11),Chave Primária

Idequipamento

Chave Estrangeira

idequipamentosala:Sala(idsala)
idequipamentomodelo:Modelo(idmodelo)
idequipamentocategoria:Categoria(idcategoria)

Tabela – UtilizadorAtributos

idutilizador int(11) not null auto_increment,
utilizador_nome varchar(100) not null,
utilizador_login varchar(40) not null,
utilizador_password varchar(20) not null,
utilizador_observacoes varchar(500),

Chave Primária

Idutilizador

Tabela – TécnicoAtributos

idtecnico int(11) not null auto_increment,
permissoes char(11) not null,
observacao varchar(300),
idtecnicoutilizador int(11) not null,

Chave Primária

idtecnico

Chave Estrangeira

idtecnicoutilizador:Utilizador(idutilizador)

Tabela – CategoriaAtributos

idcategoria int(11) not null auto_increment,
nomecategoria varchar(100) not null,

Chave Primária

Idcategoria

Tabela – MarcaAtributos

idmarca int(11) not null auto_increment,
marca_nome varchar(100) not null,

Chave Primária

idmarca

Tabela – ModeloAtributos

idmodelo int(11) not null auto_increment,
modelo_nome varchar(100) not null,
idmodelomarca int(11),

Chave Primária

Idmodelo

Chave Estrangeira

idmodelomarca:Marca(idmarca)

Tabela – SoftwareAtributos

idsoftware int(11) not null auto_increment,
software_nome varchar(100) not null,
software_versao varchar(20) not null,
software_descricao varchar(200) not null,
softw equip_id int(11) not null,

Chave Primária

idsoftware

Tabela – HardwareAtributos

idhardware int(11) not null auto_increment,
hardware_nome varchar(100) not null,
hardware_versao varchar(20) not null,
hardware_capacidade varchar(20),

Chave Primária

idhardware

Tabela – equip_sotwareAtributos

idequip_sotware int(11) not null auto_increment,
id Equipamento int(11) not null,
idequipsotware int(11) not null,

Chave Primária

idequip_sotware

Chave Estrangeira

id Equipamento:Equipamento(idequipamento)
idequipsotware:Software(idsotware)

Tabela – equip_hardwareAtributos

idequip_hardware int(11) not null auto_increment,
id Equipamento int(11) not null,
idequiphardware int(11) not null,

Chave Primária

idequip_hardware

Chave Estrangeira

id Equipamento:Equipamento(idequipamento)
idequiphardware:hardware(idhardware)

Tabela – AvariaAtributos

idavaria int(11) not null auto_increment,
avaria_data date not null,
avaria_descricao varchar(500) not null,
idavariautilizador int(11) not null,
idavariaequipamento int(11) not null,
idavaria_tipo int(11) not null,

Chave Primária

idavaria

Chave Estrangeira

idavariautilizador:Utilizador(idutilizador),
idavariaequipamento:Equipamento(idequipamento),
idavaria_tipo:Avariatiipo(idavariatiipo),

Tabela – AvariaTipoAtributos

idavariatiipo int(11) not null auto_increment,
avariatiipo_nome varchar(50) not null,

Chave Primária

idavariatiipo

Tabela – ReparaçãoAtributos

idreparacao int(11) not null auto_increment,
repar_data date not null,
repar_descricao varchar(500) not null,
idrepararavaria int(11) not null,
idreparartecnico int (11) not null,

Chave Primária

idreparacao

Chave Estrangeira

idrepararavaria:Avaria(idavaria)
idreparartecnico:Tecnico(idtecnico)

4.6 Esquema conceptual

4.6.1 Diagrama de Relações

"The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It incorporates some of the important semantic information about the real world." (Peter Chen, 1976)

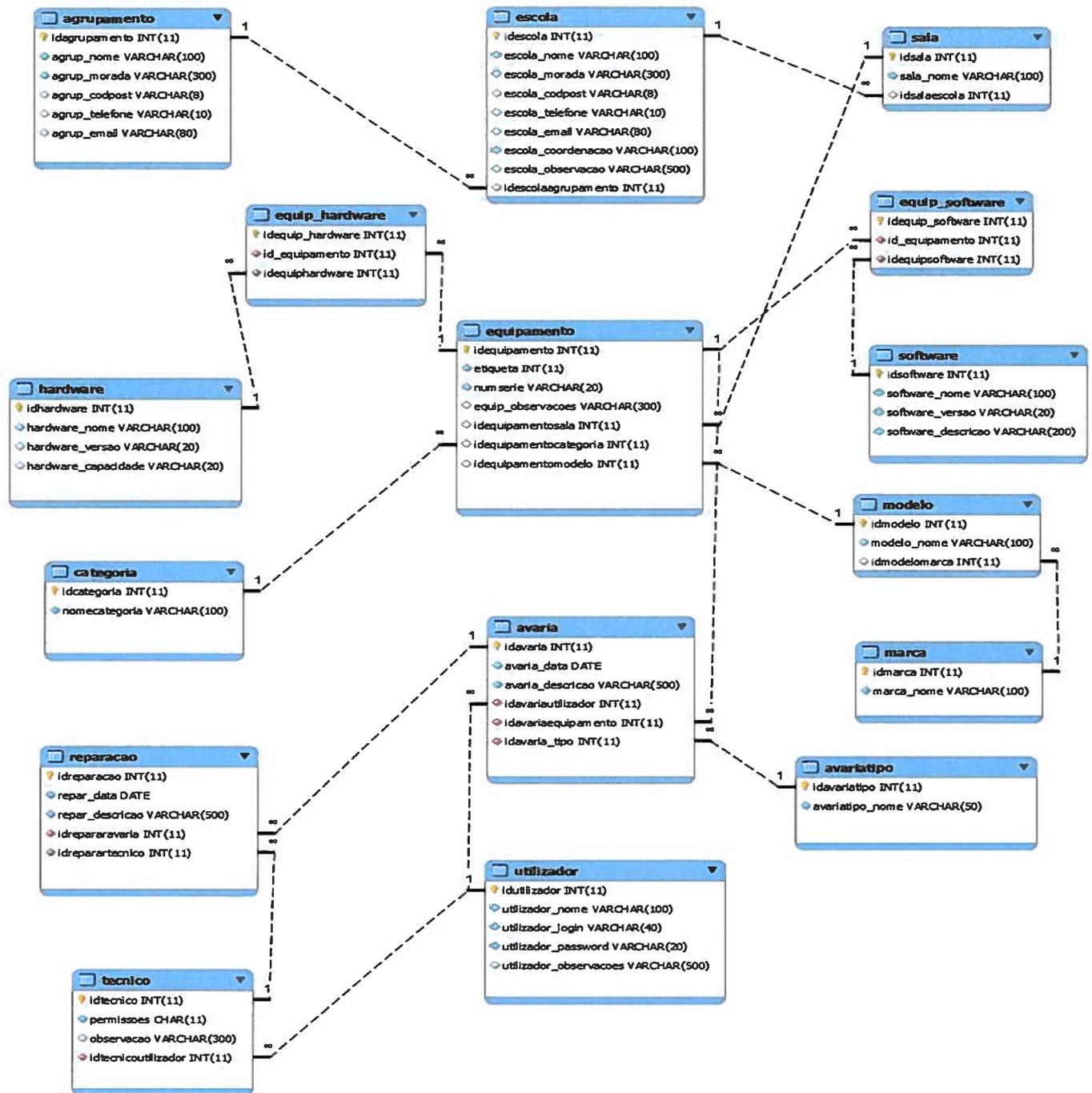
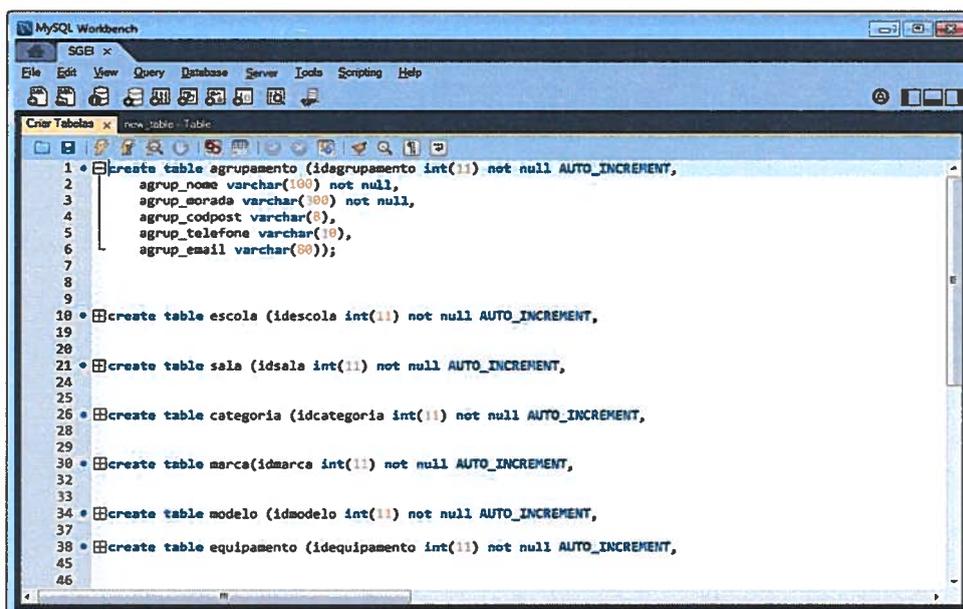


Ilustração 7 - Diagrama de Relações

4.6.2 Definição e Manipulação de Dados

Tal como descrito anteriormente (Capítulo III), é possível criar, atribuir e manipular a base de dados e respectiva informação através de linguagem específica, assim sendo, foram executados vários comandos em SQL:

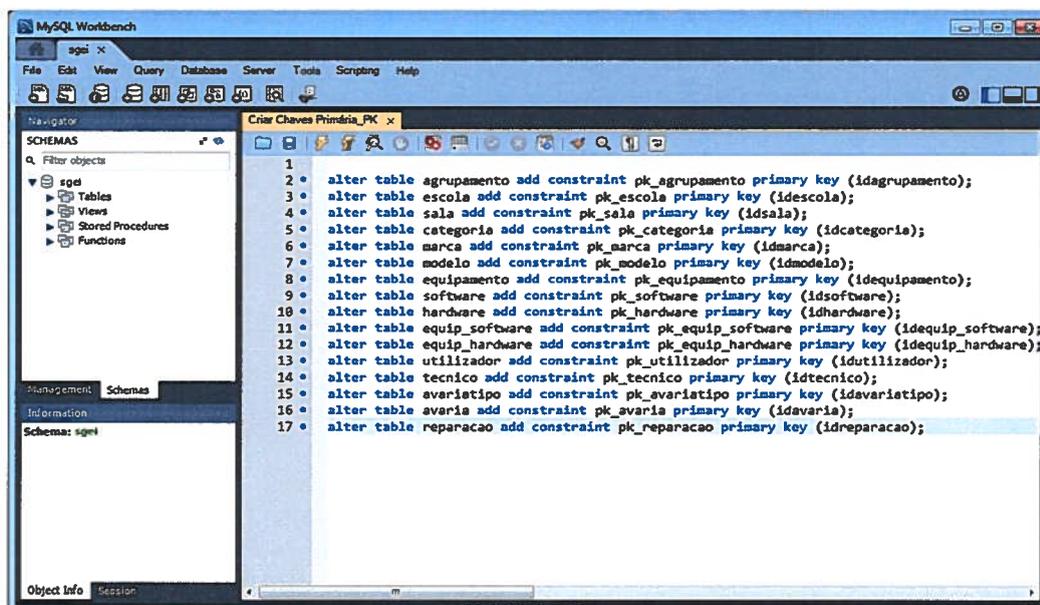
Criação de Tabelas:

A screenshot of the MySQL Workbench interface. The main window is titled 'Criar Tabelas' and contains a list of SQL commands for creating tables. The commands are:

```
1 create table agrupamento (idagrupamento int(11) not null AUTO_INCREMENT,  
2   agrup_nome varchar(100) not null,  
3   agrup_borada varchar(100) not null,  
4   agrup_codpost varchar(6),  
5   agrup_telefone varchar(10),  
6   agrup_email varchar(60));  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18 create table escola (idadescola int(11) not null AUTO_INCREMENT,  
19  
20  
21 create table sala (idsala int(11) not null AUTO_INCREMENT,  
22  
23  
24  
25  
26 create table categoria (idcategoria int(11) not null AUTO_INCREMENT,  
27  
28  
29  
30 create table marca(idmarca int(11) not null AUTO_INCREMENT,  
31  
32  
33  
34 create table modelo (idmodelo int(11) not null AUTO_INCREMENT,  
35  
36  
37  
38 create table equipamento (idequipamento int(11) not null AUTO_INCREMENT,  
39  
40  
41  
42  
43  
44  
45  
46
```

Ilustração 8 - Criação de tabela recorrendo ao SQL

Criação de Chaves Primária _PK:

A screenshot of the MySQL Workbench interface. The main window is titled 'Criar Chaves Primária_PK' and contains a list of SQL commands for adding primary keys to various tables. The commands are:

```
1 alter table agrupamento add constraint pk_agrupamento primary key (idagrupamento);  
2  
3 alter table escola add constraint pk_escola primary key (idadescola);  
4  
5 alter table sala add constraint pk_sala primary key (idsala);  
6  
7 alter table categoria add constraint pk_categoria primary key (idcategoria);  
8  
9 alter table marca add constraint pk_marca primary key (idmarca);  
10  
11 alter table modelo add constraint pk_modelo primary key (idmodelo);  
12  
13 alter table equipamento add constraint pk_equipamento primary key (idequipamento);  
14  
15 alter table software add constraint pk_software primary key (idsoftware);  
16  
17 alter table hardware add constraint pk_hardware primary key (idhardware);  
18  
19 alter table equip_software add constraint pk equip_software primary key (idequip_software);  
20  
21 alter table equip_hardware add constraint pk equip_hardware primary key (idequip_hardware);  
22  
23 alter table utilizador add constraint pk_utilizador primary key (idutilizador);  
24  
25 alter table tecnico add constraint pk_tecnico primary key (idtecnico);  
26  
27 alter table avariastipo add constraint pk_aviariastipo primary key (idavariastipo);  
28  
29 alter table avaria add constraint pk_avaria primary key (idavaria);  
30  
31 alter table reparacao add constraint pk_reparacao primary key (idreparacao);
```

Ilustração 9 - PK Chaves Primárias

Criação de Chaves Estrangeira _FK:

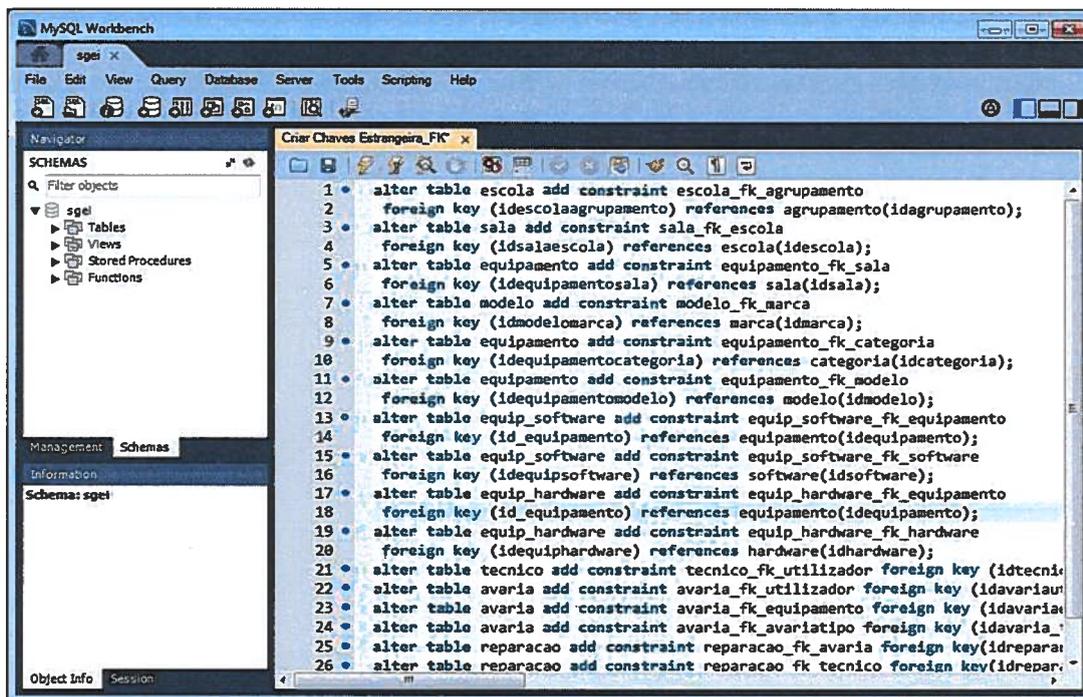


Ilustração 10 - FK - Chaves Estrangeiras

Atribuição de Auto Incremento (Query Alter):

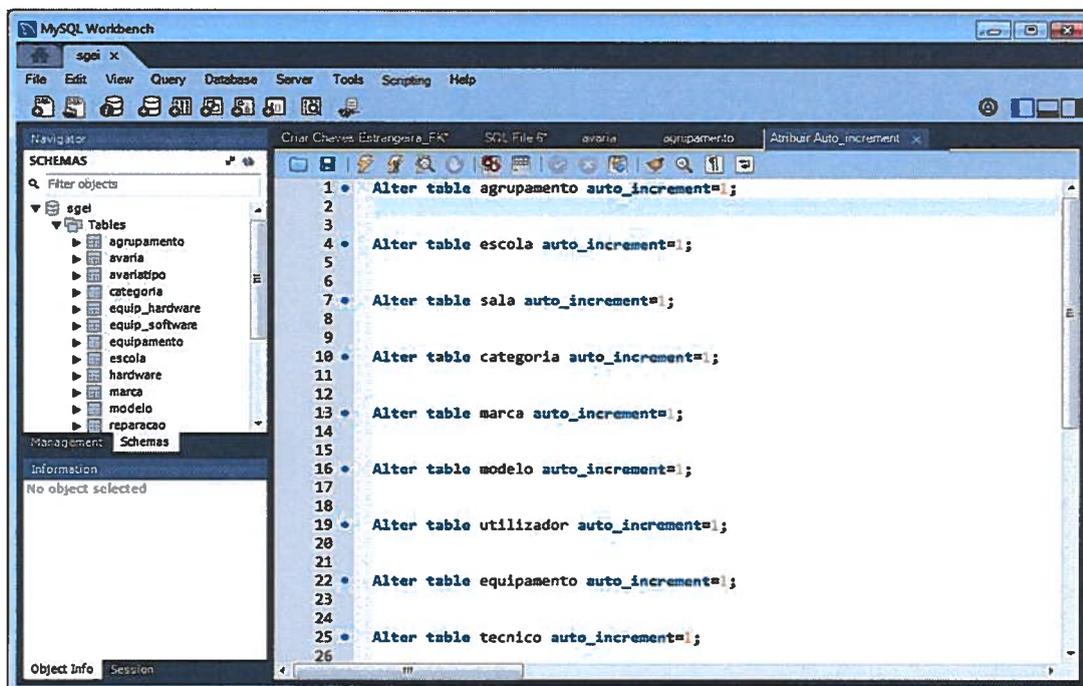


Ilustração 11 - Auto_increment

Inserção de dados (*Query Insert*):

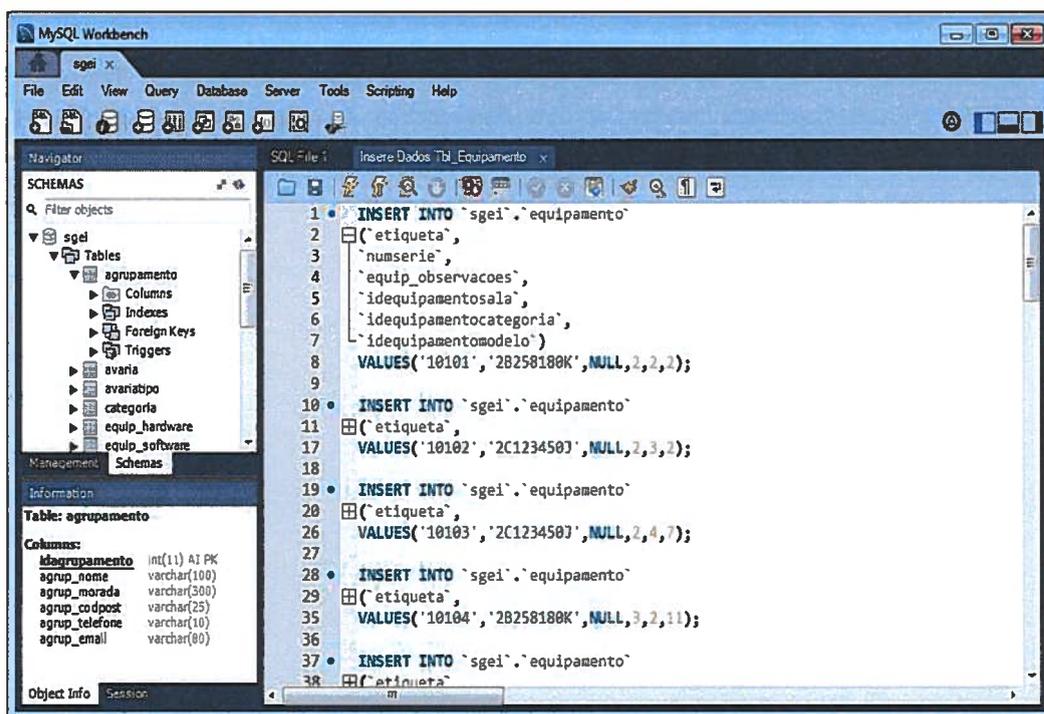


Ilustração 12 - Inserir dados

Eliminação de dados (*Query Delete*):

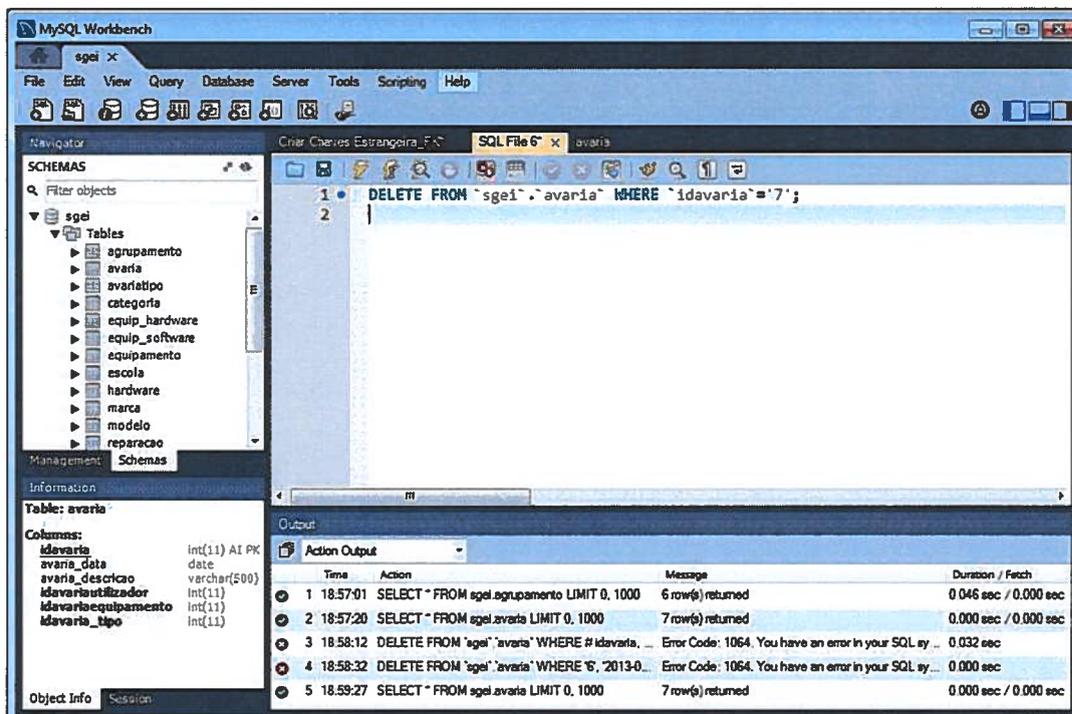


Ilustração 13 - Eliminação de dados

Alteração de dados (Query Update):

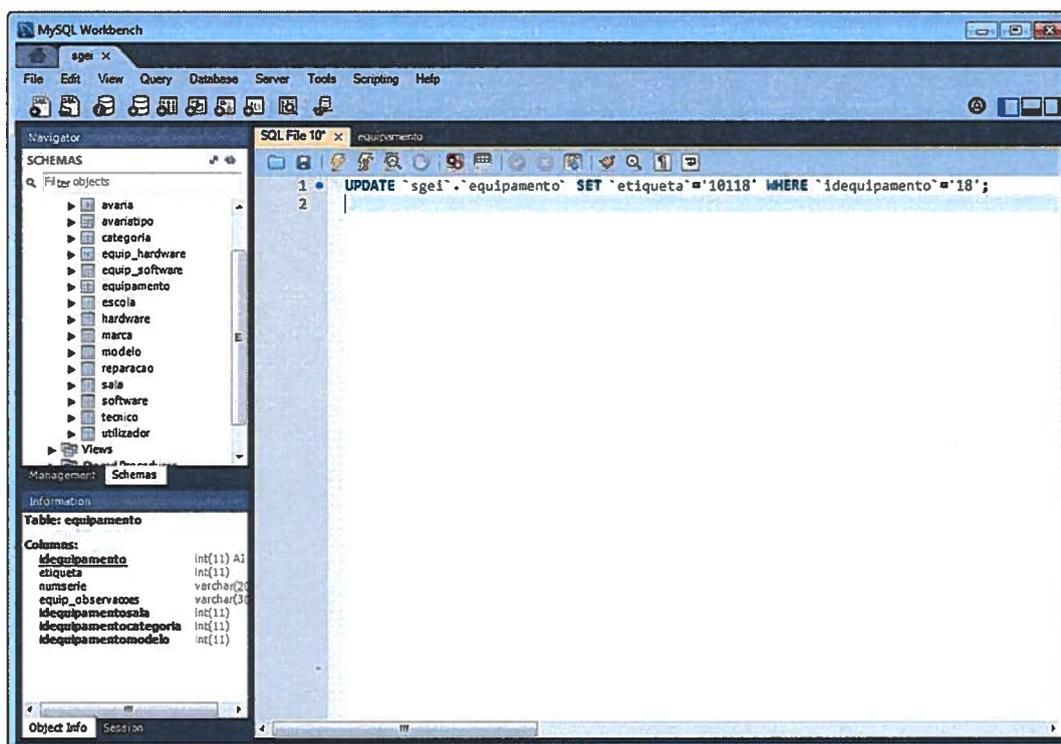


Ilustração 14 - Alteração de dados

Consulta de dados (Query Select):

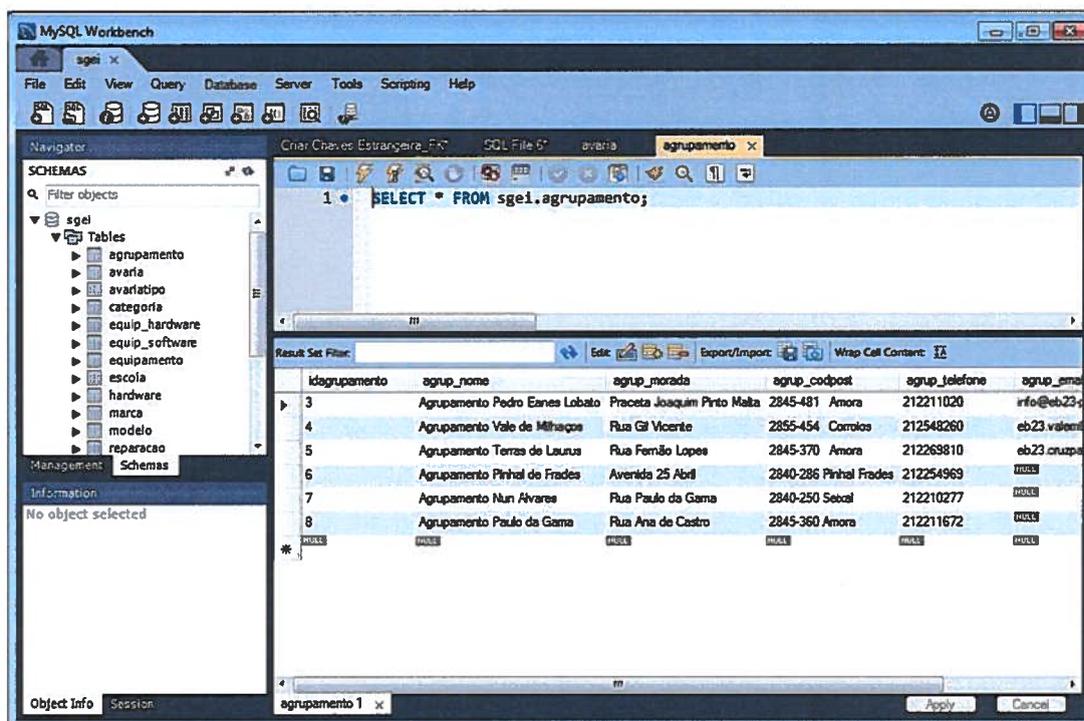
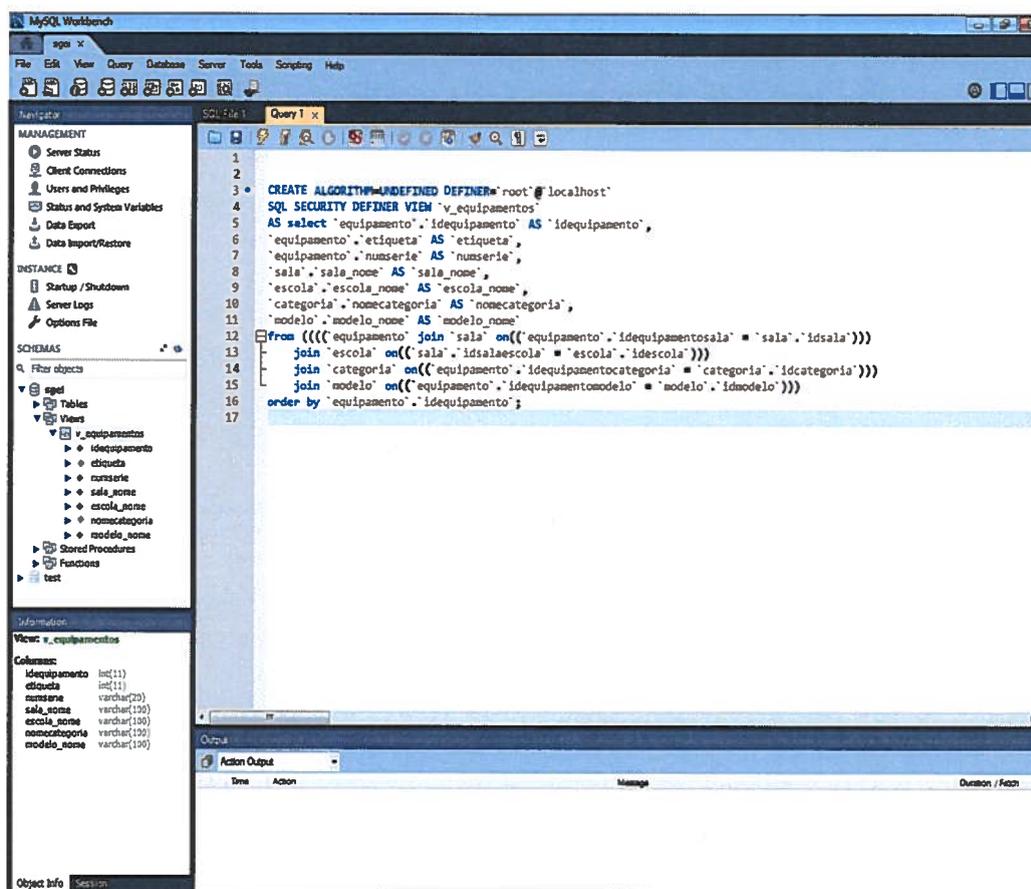


Ilustração 15 - Consulta de dados

Criação de visualizações (views):

Tendo como objectivo a optimização da base de dados, uma *View* é a representação virtual de uma tabela, e pode conter colunas de uma ou varias tabelas físicas ou até mesmo de outras *Views*. Na maioria das vezes, as *Views* não armazenam os dados na base de dados, apenas consultam os dados que foram seleccionados nas tabelas de origem. É possível utilizar uma *View* em qualquer lugar onde se possa utilizar uma tabela.

Na *view* abaixo podemos ver todos os equipamentos:



The screenshot shows the MySQL Workbench interface. The main window displays the following SQL code for creating a view:

```
1  
2  
3 CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost'  
4 SQL SECURITY DEFINER VIEW `v_equipamentos`  
5 AS select `equipamento`.`idequipamento` AS `idequipamento`,  
6 `equipamento`.`etiqueta` AS `etiqueta`,  
7 `equipamento`.`numserie` AS `numserie`,  
8 `sala`.`sala_nome` AS `sala_nome`,  
9 `escola`.`escola_nome` AS `escola_nome`,  
10 `categoria`.`nomecategoria` AS `nomecategoria`,  
11 `modelo`.`modelo_nome` AS `modelo_nome`  
12 from (((`equipamento` join `sala` on((`equipamento`.`idequipamentosala` = `sala`.`idsala`)))  
13 join `escola` on((`sala`.`idsalaescola` = `escola`.`idescola`)))  
14 join `categoria` on((`equipamento`.`idequipamentocategoria` = `categoria`.`idcategoria`)))  
15 join `modelo` on((`equipamento`.`idequipamentomodelo` = `modelo`.`idmodelo`)))  
16 order by `equipamento`.`idequipamento`;  
17
```

The left sidebar shows the 'SCHEMAS' tree with the 'v_equipamentos' view selected. The 'Information' panel shows the columns and their data types:

Column	Data Type
idequipamento	int(11)
etiqueta	int(11)
numserie	varchar(20)
sala_nome	varchar(100)
escola_nome	varchar(100)
nomecategoria	varchar(100)
modelo_nome	varchar(100)

Ilustração 16 - View Equipamentos

Capítulo V- SGEI - Interface gráfico

Tendo como objectivo a criação de uma aplicação dinâmica de forma gratuita, tal como referido anteriormente, recorri ao PHP. No entanto, e para trabalhar com PHP, para além da base de dados criada anteriormente, é necessário ter instalado um servidor que irá interpretar o código PHP e gerar um código HTML.

Uma vez mais, tendo em conta o objectivo principal deste projecto, de tudo ser gratuito, optei por o WAMPServer, um aplicativo gratuito para Windows que instala e configura automaticamente na máquina o servidor Apache, a Base de dados MySQL e o PHP.

Não sendo objecto deste projecto, explicar o conceito de servidor de aplicações, deixo apenas a anotação sobre o que significa WAMP: abreviatura de Windows, ApacheMySQL e PHP.

Windows = Sistema operativo base;

Apache = servidor de publicações;

MySQL = servidor da Base de Dados;

PHP = execução da linguagem / interface gráfico.

O WAMPServer vai permitir que o utilizador final, com uma única instalação (única ferramenta), disponha de tudo o que é necessário para o funcionamento do SGEI.

O acesso ao SGEI será feito via motor de busca, acedendo nesta fase ao seguinte endereço com as seguintes credenciais: <http://localhost:8080/sgei/login.php>
utilizador: lvalentim, palavra-chave: teste



Ilustração 17 - Iniciar Sessão em SGEI

As consultas serao efectuadas através de um simples interface grafico, tambem desenvolvido neste projeto:

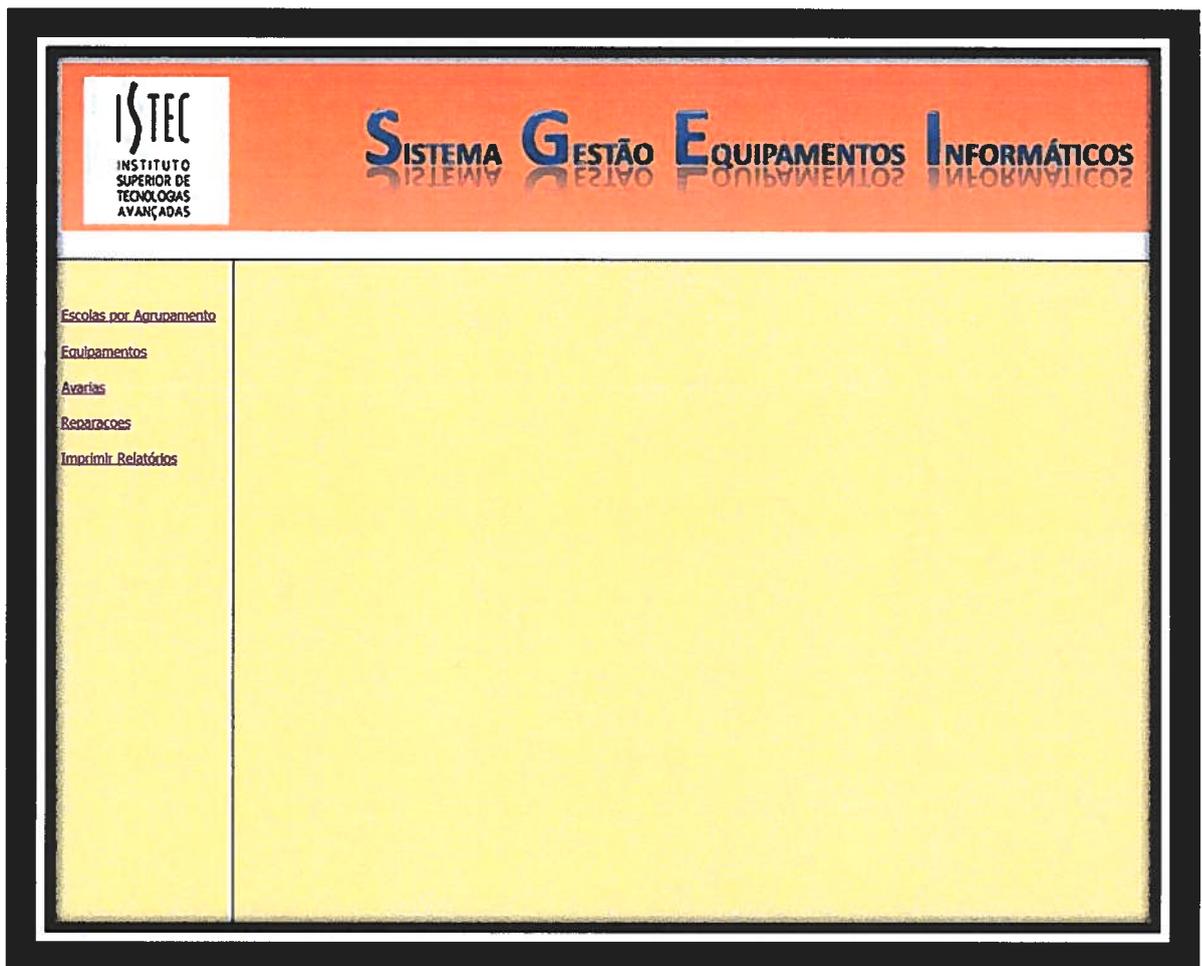


Ilustração 18 - Consultas em SGEI

Capítulo VI - SGEI – Testes de Usabilidade

Utilizando o interface gráfico acima apresentado e de acordo com o solicitado pelos intervenientes no processo, foram efectuados testes, durante a realização dos mesmos obteve-se um comportamento normal, performance satisfatória, não foi posto em causa o bom funcionamento dos equipamentos utilizados.

Foram criados relatórios individuais de acordo com cada pedido efectuado, a sua criação poderá ser consultada nos seguintes anexos:

rel_escolas por agrupamento.sql ; rel_coordenador(a) responsável pela escola.sql ; rel_equipamentos por sala /escola. sql ; rel_salas sem equipamento informático. sql ; rel_numero total de avarias reportadas sql ; rel_numero total de avarias por equipamento. sql ; rel_numero total de reparações por técnico. sql ; rel_software instalado por pc. sql ; rel_hardware instalado por pc. sql ; rel_características dos equipamentos. sql

Testemunhos:

“

Após a realização deste projecto, procedemos à sua utilização como ferramenta de gestão, controlo de equipamentos e do serviço prestado, estamos muito satisfeitos com a performance da base de dados e conseguimos otimizar e responder a questões que antes se verificavam morosos em tempo útil muito satisfatório.

Coordenador do Serviço de Informática

Alfredo Ribeiro, 20 de Maio de 2014

“

“

Na sequencia da instalação desta base de dados é possível verificar em tempo real o trabalho em curso, tal como, os equipamentos existentes, escolas e responsáveis, o que facilita em muito o apoio técnico por mim prestado às escolas.

Técnico de Informática

Adolfo Ameixa, 10 de Junho 2014

“

Conclusão

Analisando os objectivos deste trabalho, desenvolvido no sentido da criação de uma base de dados, pode considerar-se que o projecto foi bem-sucedido tendo sido atingidos os seguintes objectivos:

Desenvolvimento de uma base de dados, capaz de responder às especificações solicitadas pelos intervenientes no processo, permitindo o aumento da qualidade de serviço de informática no fornecimento e assistência técnica de equipamentos informáticos às Escolas Básicas de 1º Ciclo e Jardins de Infância, do concelho do Seixal.

Aplicação de técnicas e conceitos abordados durante a licenciatura de Informática e simultaneamente aprofundar os conhecimentos e o estudo de novas ferramentas e tecnologias de desenvolvimento.

Do âmbito deste último ponto, foi possível ampliar os conhecimentos adquiridos relativos ao desenvolvimento de aplicações usando tecnologia gratuita e permitiu também o aprofundamento das competências no uso de ferramentas de base de dados.

É ainda importante referir que após a apresentação do trabalho junto do serviço de informática desta entidade, o mesmo passou a ser utilizado como ferramenta de gestão e controlo de equipamentos e do serviço prestado. Assim sendo, este é um factor que deverá contribuir para se considerar o desenvolvimento da aplicação um sucesso.

Bibliografia

Livros:

Caldeira, C. (2011). A Arte das Bases de Dados. (1ª edição). Lisboa: Edições Silabo, Lda.

Carriço, J. A. & Carriço, A. J. (2004). Desenho de Base de Dados e Linguagem SQL. Lisboa: Edições Chambel.

PEREIRA, José Luís (1998) - Tecnologias de Bases de Dados, FCA Editora;

Peter Chen, Peter Pin-Shan (March 1976). "The Entity-Relationship Model - Toward a Unified View of Data". ACM Transactions on Database Systems

Thomas Connolly, Carolyn Begg, Anne Strachan. Database Systems, A Practical Approach to Design, Implementation, and Management, 2nd Edition > acedido em 15/05/2014

Heuser, Carlos A. Projeto de Banco de Dados. 2ª edição. Sagra Luzzatto. 1999.

Sítios da *Internet*:

Vasco Pereira (2012). Introdução ao PHP > Acedido em <http://www.cidadela.org/>

The Importance of Data Modeling > acedido em 16/12/2013

<Http://www.4guysfromrolla.com/Internettech/datamodeling.shtml>

DB-Engines, The most popular database management systems, 2013

> acedido em 21/11/2013

MySQL, Top Reasons to Use MySQL, 2013

><Http://www.mysql.com/why-mysql/case-studies/>> acedido em 21/11/2013

Norton, Peter. Introdução à Informática. Makron Books. 1996 > acessado em 10/09/2013 > [Http://www.sirmacstronger.eti.br/bd/introdbd.php](http://www.sirmacstronger.eti.br/bd/introdbd.php)

Carlos A. Projecto de Banco de Dados. 2ª Edição. Sagra Luzzatto. 1999> [Http://www.sirmacstronger.eti.br/bd/introdbd.php](http://www.sirmacstronger.eti.br/bd/introdbd.php)> acessado em 10/09/2013

McClure, Wallace B.; Beamer, Gregory A.; Croft IV, John J.; Little, J. Ambrose; Ryan, Bill; Winstanley, Phil; David, Yack; Zongker, Jeremy. Professional ADO.NET 2 Programming with SQL Server 2005, Oracle, and MySQL. Wiley Publishing, Inc. 2006. > <http://www.sirmacstronger.eti.br/bd/introdbd.php> > acessado em 10/09/2013

Nielsen, Paul. , SQL Server 2005 Bible. Wiley Publishing, Inc. 2007 > [Http://www.sirmacstronger.eti.br/bd/introdbd.php](http://www.sirmacstronger.eti.br/bd/introdbd.php)> acessado em 10/09/2013

Sobrinho, Álvaro Caetano Pimentel. Open Access: Ser ou não ser? > <http://artigocientifico.tebas.kinghost.net/pesquisadores/?mnu=2&smnu=5&id=29536> > acessado em 09/01/2014

<http://www.glpi-project.org/spip.php?article315> > acessado em 01/02/2014

[Http://www.spiceworks.com/](http://www.spiceworks.com/) > acessado em 01/02/2014

Banco de dados> TrabalhosFeitos.com.> Acessado em 05/07/ 2013, de <http://www.trabalhosfeitos.com/ensaios/Banco-De-Dados/856955.html>

Conference on Data Systems Languages Records, 1959-1987 > Acessado em 07/09/2013 > <http://discover.lib.umn.edu>

Introdução às Bases de Dados Oracle, Pedro Bizarro, 2000> acessado em> http://eden.dei.uc.pt/~bizarro/files/manual_praticas_bd1_v3.pdf

<http://www.linhadecodigo.com.br/artigo/1308/melhorando-desempenho-de-consultas-utilizando-views-indexadas.aspx#ixzz35XySHc4z>

Outros:

Documento Verde da Reforma da Administração Local, Gabinete do Ministro Adjunto e dos Assuntos Parlamentares

Anexos e Apêndices

SGEI.sql
Criar Tabelas.sql
Criar Chaves Estrangeira_FK.sql
Criar Chaves Primária_PK.sql
Atribuir Auto_increment.sql
Inserer Dados Tbl_Agrupamento.sql
Inserer Dados Tbl_Avaria.sql
Inserer Dados Tbl_AvariaTipo.sql
Inserer Dados Tbl_Categoria.sql
Inserer Dados Tbl_Equip_Hardware.sql
Inserer Dados Tbl_Equip_software.sql
Inserer Dados Tbl_Equipamento.sql
Inserer Dados Tbl_Escola.sql.
Inserer Dados Tbl_Hardware.sql
Inserer Dados Tbl_Marca.sql
Inserer Dados Tbl_Modelo.sql
Inserer Dados Tbl_Reparacao.sql
Inserer Dados Tbl_Sala.sql
Inserer Dados Tbl_Software.sql
Inserer Dados Tbl_Tecnico.sql
Inserer Dados Tbl_Utilizador.sql
v_avarias.sql
v_agrupamentos.sql
v_reparacoes.sql
v_equipamentos.sql
v_agrupvalemilhacos.sql
v_agrupterraslaurus.sql
v_agruppinhalfrades.sql
v_agruppedroeaneslobato.sql
v_agruppaulogama.sql
v_hardequip.sql
v_swequip.sql

v_numtotalavariaequip.sql
v_salassemequip.sql
rel_escolas por agrupamento.sql
rel_coordenador(a) responsável pela escola.sql
rel_equipamentos por sala /escola. sql
rel_salas sem equipamento informático. sql
rel_numero total de avarias reportadas sql
rel_numero total de avarias por escola. sql
rel_numero total de reparações por técnico. sql
rel_software instalado por pc. sql
rel_hardware instalado por pc. sql
rel_características dos equipamentos. sql

NOTA: todos estes ficheiros encontram-se em formato digital.

