

Bases de Dados

2. Bases de Dados

- De uma forma genérica podemos dizer que uma base de dados é um conjunto de dados relacionados, acessíveis a uma comunidade de utilizadores.
- Desde tempos imemoriais o homem guardou em armários, gavetas (...) os dados importantes relativos às suas diversas actividades – as primeiras bases de dados.

2. Bases de Dados

- A partir de meados do século XX verificou-se um grande crescimento das organizações o que provocou um grande aumento no volume dos dados nelas utilizados.
- Ao mesmo tempo verificou-se um grande desenvolvimento dos computadores, que se tornaram mais acessíveis.
- Surge assim, de forma natural, a introdução dos computadores nas organizações com o objectivo de servir de suporte às bases de dados nelas existentes.

2. Bases de Dados

- Numa primeira fase, surgem os chamados Sistemas de Ficheiros e só posteriormente são desenvolvidos os Sistemas de Bases de Dados.
- Com ambos se pretende obter acesso rápido aos dados gerando a informação necessária às diversas actividades.
- Os dados são guardados em ficheiros e a partir deles são gerados relatórios com a informação pretendida.
- Actualmente, quando nos referimos a bases de dados, queremos referir-nos às bases de dados em suporte informático.

2. Bases de Dados

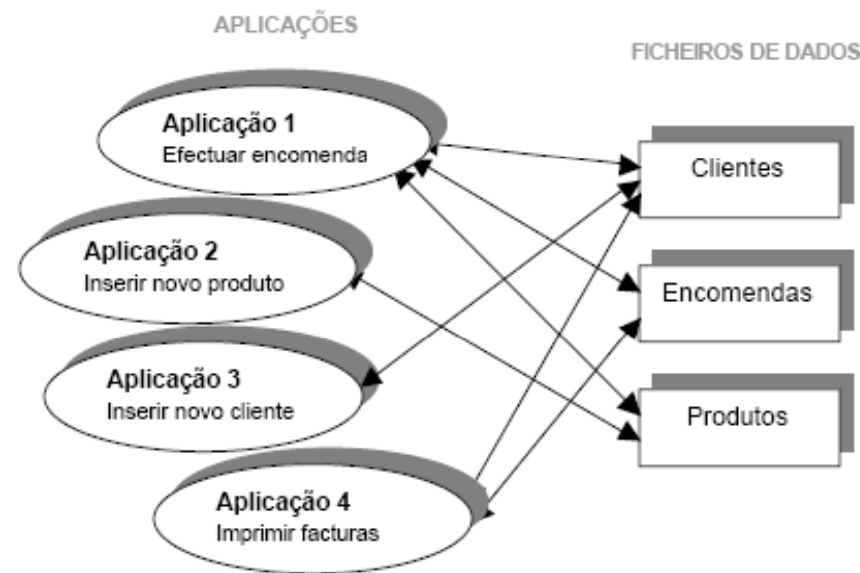
- Embora se possa considerar que os Sistemas de Ficheiros estão desactualizados e que não fará sentido abordá-los, só se podem compreender as potencialidades de um Sistema de Bases de Dados se houver a compreensão dos Sistemas de Ficheiros, que os antecederam.

2. Bases de Dados

- Neste capítulo, iremos começar por abordar, em 2.1, as raízes históricas das bases de dados, nomeadamente, os Sistemas de Ficheiros.
- Em 2.2, veremos as características de um Sistema de Bases de Dados e os seus intervenientes. Abordaremos ainda alguns aspectos importantes de um Sistemas de Gestão de Bases de Dados.
- Os modelos de Bases de Dados serão abordados em 2.3.
- Por fim, em 2.4, abordaremos algumas tendências de evolução na área das bases de dados.

2.1 As Raízes Históricas: Ficheiros e Sistemas de Ficheiros

- Num Sistema de Ficheiros guardam-se diversos registos, de forma permanente, em vários ficheiros.
- Existem também diversas aplicações ou programas que permitem efectuar uma variedade de operações com esses ficheiros: inserir dados, recuperar dados, alterar dados e apagar dados (Figura I.2.1).



2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Os ficheiros de um Sistema de Ficheiros possuem uma determinada estrutura física, que tem de ser conhecida para que possa ser elaborada cada uma das aplicações do sistema.
- Na figura I.2.2, apresenta-se um exemplo do ficheiro Clientes, pertencente ao sistema de ficheiros apresentado na figura I.2.1.

	CLI-COD	CLI-NOME				CLI-MORADA				CLI-TELEF													
campo	1	a	n	a			p	o	r	t	o			1	2	3	4	5	6	7	8	9	
	2	m	a	r	i	a								4	4	4	5	5	5	6	6	6	
dado	3	p	s	é										7	7	7	8	8	8	9	9	9	
	4	t	i	a	g	o								9	9	9	8	8	8	9	9	9	
	5	p	e	d	r	o								6	6	6	5	5	5	4	4	4	
registo	6	j	o	a	q	u	i	m			m	a	i	a				3	3	3	2	2	2
																				1	1	1	

Figura I.2.2 Exemplo do ficheiro Clientes

2.1.1 Gestão de dados nos Sistemas de Ficheiros

Antes de avançarmos, vamos definir alguma terminologia básica sobre ficheiros:

- **Dado** - factos em “bruto”, ou seja, a mais pequena porção de dados que o computador reconhece; pode ser um número, um carácter,...;
- **Campo** - um carácter ou conjunto de caracteres com um significado específico, como por exemplo um nome;
- **Registo** - um conjunto de um ou mais campos relacionados, que descrevem uma pessoa, um lugar ou um objecto (concreto ou abstracto);
- **Ficheiro** - conjunto de registos relacionados.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

Cada ficheiro presente no sistema de gestão de ficheiros precisa, pelo menos, de cinco aplicações que permitam:

- Criar a estrutura do ficheiro;
- Adicionar dados ao ficheiro;
- Eliminar dados do ficheiro;
- Modificar dados contidos no ficheiro;
- Listar o conteúdo do ficheiro.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

Cada uma dessas aplicações envolve várias operações, mais ou menos complexas. Por exemplo, para eliminar o cliente **Ana**, do ficheiro de **clientes** do exemplo, será necessário:

- Abrir o ficheiro de **clientes**;
- Procurar o registo relativo ao cliente **Ana**;
- Eliminar o registo encontrado;
- Fechar o ficheiro **clientes**.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Podemos, através deste exemplo simples, reconhecer a dificuldade e morosidade existentes na implementação de um sistema de ficheiros, se pensarmos que todo este trabalho terá de ser multiplicado pelo número de ficheiros existentes no sistema.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Por outro lado, a sua manutenção também é difícil.
- Consideremos novamente o exemplo da figura 1.2.1. Depois de implementado o sistema, os gestores da empresa onde ele foi implementado, consideram importante obter relatórios mensais, contendo informações sobre os valores das encomendas realizadas por cada um dos clientes.
- Passa a haver a necessidade de, em cada mês, elaborar uma aplicação que possa produzir esse relatório.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Conforme se conseguem satisfazer algumas necessidades surgem imediatamente novas necessidades, o que implica a escrita de novas aplicações.
- Muitas vezes torna-se incomportável fazer uma gestão eficiente do sistema, devido ao elevado número de aplicações necessárias à sua manutenção.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Outro aspecto a ter em conta na gestão dos dados prende-se com a integridade dos dados, isto é, os valores guardados nos ficheiros devem satisfazer determinadas regras – as restrições de integridade.
- No exemplo já apresentado, considerando o ficheiro encomendas, “só pode existir uma encomenda se contiver algum produto” ou “só são aceites encomendas de valor superior a 5 €”, são exemplos de duas restrições de integridade.
- Estas restrições são introduzidas no sistema adicionando código apropriado em várias aplicações, implicando um grande esforço adicional.

2.1.1 Gestão de dados nos Sistemas de Ficheiros

- Todas estas considerações têm a ver com sistemas utilizados por um utilizador.
- Imagine-se o esforço necessário para implementar e manter um sistema no qual exista a possibilidade de vários utilizadores (não simultâneos).
- Neste caso, seria preciso definir para, cada utilizador, se ele poderia ou não aceder aos dados, se poderia acedê-los apenas para consulta ou se poderia alterar os dados, etc.

2.1.2 Dependência dados-aplicações

- Num sistema de ficheiros, as aplicações estão directamente relacionadas com os ficheiros, isto é, quando uma aplicação acede a um ficheiro, deve “dizer” ao computador não só “a que aceder” mas também “como acedê-lo”.
- Assim sendo, qualquer alteração na estrutura dos ficheiros provocará alterações em todas as aplicações que com ela se relacionam, com todos os inconvenientes daí resultantes.

2.1.3 Redundâncias nos dados

- Nos sistemas de ficheiros é difícil evitar que existam dados repetidos em diversos ficheiros.
- Quando isto acontece, diz-se que existe redundâncias nos dados.
- No exemplo da figura I.2.1, os dados relativos a cada cliente no ficheiro *clientes*, surgem repetidos no ficheiro *encomendas*.
- Estas redundâncias conduzem a custos mais elevados ao gravar e ao aceder aos dados.

2.1.3 Redundâncias nos dados

- Um aspecto importante a considerar é o facto de estas redundâncias poderem gerar inconsistências nos dados, isto é, várias cópias do mesmo dado não serem coincidentes.
- Consideremos, no exemplo da figura 1.2.1, a situação de, ao realizar uma encomenda se verificar que a morada do cliente que fez essa encomenda foi alterada.
- Como o utilizador está a efectuar o pedido de encomenda com a aplicação destinada para esse efeito, inclui a nova morada apenas no ficheiro *encomendas*.

2.1.3 Redundâncias nos dados

- Nesse momento surge um pedido de encomenda de um outro cliente.
- Como, para efectuar a alteração da morada do cliente anterior, é necessário recorrer a outra aplicação, o utilizador aproveita o facto de estar a utilizar a aplicação destinada ao pedido de encomendas para efectuar a nova encomenda.
- Entretanto esqueceu-se da alteração que deveria ter sido realizada! Desta forma, os dados relativos à morada do referido cliente não são coincidentes, isto é, são *inconsistentes*.

2.2 Sistemas de Bases de Dados

- Um Sistema de Bases de Dados (SBD) é um sistema baseado em computador(es) que tem como principal objectivo armazenar dados de modo a permitir aos utilizadores consultar e actualizar a informação solicitada.
- Os Sistemas de Bases de Dados surgiram como resposta aos problemas apresentados pelos sistemas de ficheiros. Ao contrário destes, aqueles caracterizam-se por um repositório de dados que é acedido pelos diversos utilizadores através de um sistema – o Sistema de Gestão de Bases de Dados (SGBD) (figura I.2.3).

2.2 Sistemas de Bases de Dados

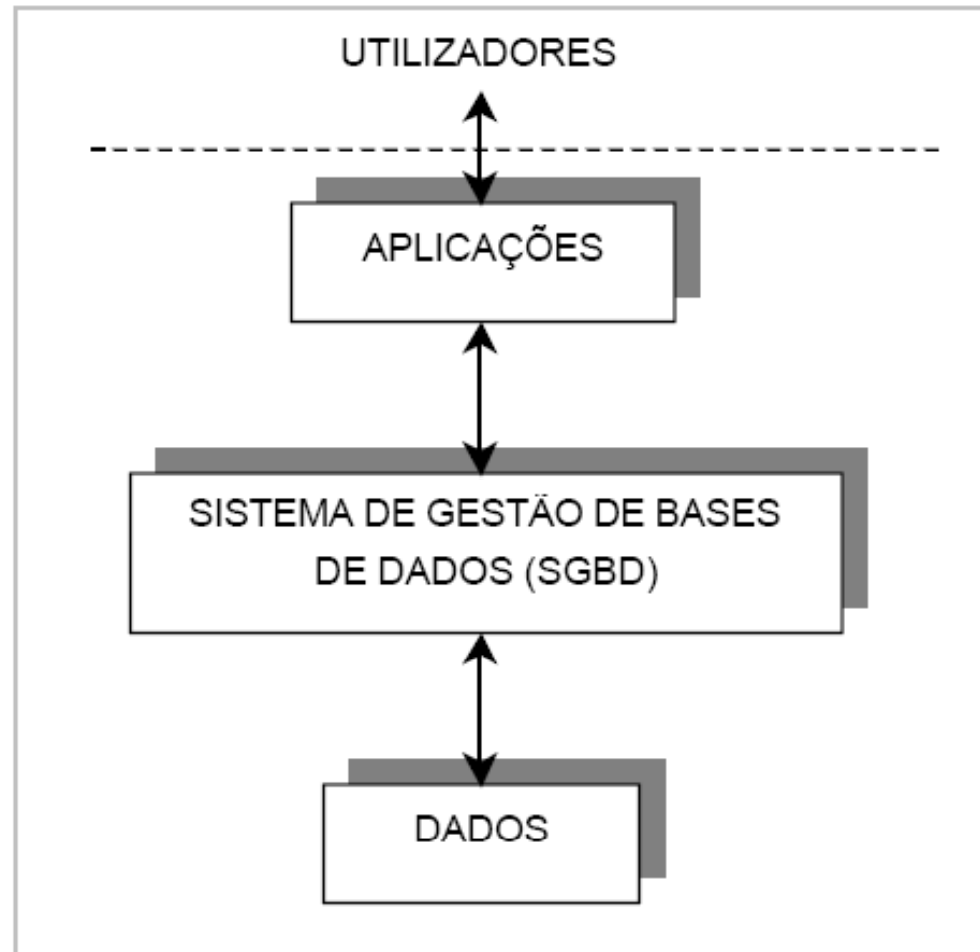


Figura I.2.3 Imagem simplificada de um Sistema de Bases de Dados

2.2.1 Características de um Sistema de Bases de Dados

Várias características distinguem os Sistemas de Bases de Dados dos Sistemas de Ficheiros:

1. **Catálogo do sistema:** uma das características fundamentais dos Sistemas de Bases de Dados é que contém, não só a base de dados, mas também a definição completa ou descrição da base de dados – o catálogo do sistema ou dicionário de dados. A informação contida no catálogo é chamada de *metadata* (dados sobre dados). O catálogo do sistema está acessível ao Sistema de Gestão de Bases de Dados e aos utilizadores e é automaticamente criado pelo sistema. A existência do catálogo do sistema permite que um só SGBD possa ser eficientemente utilizado com diversas bases de dados;

2.2.1 Características de um Sistema de Bases de Dados

- 2. Independência** **Dados-Programas:** outra característica importante dos Sistemas de Bases de Dados é a independência dados-programas: a estrutura dos ficheiros de dados está gravada no catálogo do SGBD, separada dos programas que lhe acedem;

2.2.1 Características de um Sistema de Bases de Dados

3. Abstracção dos dados: o Sistema de Base de Dados deve fornecer ao utilizador uma representação abstracta dos dados, isto é, o sistema deve esconder detalhes sobre a forma como os dados são fisicamente armazenados e mantidos – abstracção dos dados.

2.2.1 Características de um Sistema de Bases de Dados

4. **Vistas múltiplas dos dados:** normalmente, uma base de dados tem vários utilizadores. Cada utilizador necessita de ter acesso a perspectivas diferentes da base de dados – vista da base de dados (Figura I.2.4). Uma vista pode ser um subconjunto da base de dados, como por exemplo, a área de produção, do marketing, etc.

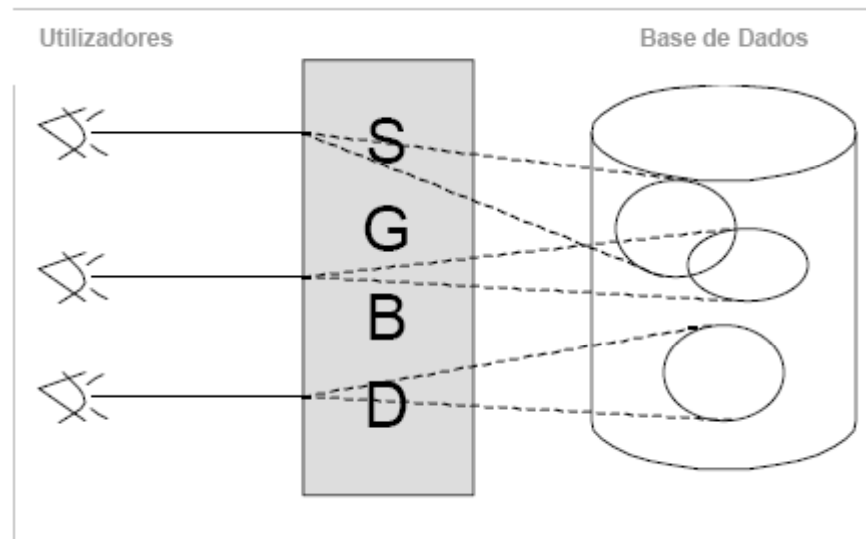


Figura I.2.4 Vistas da Base de Dados

2.2.2 Intervenientes num Sistema de Bases de Dados

Podemos classificar as pessoas intervenientes num Sistema de Bases de Dados em quatro categorias (que por vezes se sobrepõem):

1. **Administrador de Dados e Administrador da Base de Dados:** o Administrador de Dados é o gestor responsável pela administração dos recursos da base de dados, passando pela planificação, pelo estabelecimento de políticas e procedimentos. O Administrador da Base de Dados tem o controlo centralizado dos dados e dos programas que acedem a esses dados. É o responsável técnico que assegura que todo o sistema funciona de forma conveniente. Em algumas organizações não existe distinção entre estes dois papéis.

2.2.2 Intervenientes num Sistema de Bases de Dados

- 2. Analistas da base de dados:** são as pessoas que fazem o desenho da base de dados.

São os responsáveis pela identificação dos dados a serem armazenados na base de dados e pela definição das estruturas físicas necessárias para armazenar esses dados.

Para que a sua função seja cumprida de forma efectiva, os analistas devem estar envolvidos com as regras do negócio e com os utilizadores finais.

2.2.2 Intervenientes num Sistema de Bases de Dados

3. **Programadores de aplicações:** implementam as funcionalidades pretendidas pelos utilizadores finais, obtidas a partir de especificações fornecidas pelo analista de sistemas.

Além disso, testam, corrigem erros, documentam e mantêm os programas escritos.

Podem utilizar linguagens de programação de 3ª geração (por exemplo, C) ou de 4ª geração (por exemplo, INFORMIX-4GL).

2.2.2 Intervenientes num Sistema de Bases de Dados

4. **Utilizadores finais:** são as pessoas que acedem ao sistema para obter a informação necessária ao seu desenvolvimento na organização.

Alguns desses utilizadores acedem à base de dados a partir de programas e interfaces já definidos, que pretendem tornar o acesso o mais simplificado possível.

Existem outros utilizadores mais sofisticados, familiarizados com as facilidades fornecidas pelo SGBD que podem eles próprios escrever as suas aplicações.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

- O Sistema de Gestão de Bases de Dados (SGBD) é uma aplicação informática que faz a gestão de toda a base de dados fornecendo, aos seus utilizadores, acessos controlados.
- Vamos, de seguida, apresentar algumas das funções de um SGBD.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

- 1. Gestão do armazenamento dos dados:** o SGBD cria as estruturas necessárias ao armazenamento, e utilização dos dados, libertando o utilizador da difícil tarefa de definir e programar as suas características físicas.

É criado o dicionário de dados que facilita este processo.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

2. Gestão da Interface: o SGBD formata os dados consultados de modo a que tenham um aspecto facilmente perceptível pelo utilizador.

São disponibilizadas determinadas linguagens de programação que permitem essa interface.

Os SGBD's mais recentes apresentam interfaces amigáveis que facilitam grandemente o acesso à base de dados por parte de utilizadores não especializados.

Fornecem também capacidades para comunicar em ambiente de redes de computadores, nomeadamente através da Internet.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

- 3. Gestão da Segurança:** o SGBD cria um sistema de segurança que protege a base de dados de acessos não autorizados.

São impostas regras que definem quais os utilizadores que podem ter acesso à base de dados, e para os utilizadores autorizados a forma como pode ser feito esse acesso, isto é, quais os ficheiros que podem ser acedidos e as operações permitidas (ler, adicionar, apagar ou modificar).

Deverão existir procedimentos que efectuem cópias de segurança e recuperação dos dados em caso de falhas, de modo a assegurar a segurança e a integridade dos dados.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

- 4. Manutenção da Integridade dos dados:** o SGBD assegura a verificação de restrições de integridade de modo a que a base de dados contenha sempre dados válidos, minimizando a redundância nos dados e maximizando a sua consistência. Existem restrições de integridade intrínsecas ao SGBD e outras definidas pelos utilizadores do sistema.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

(4. continuação)

Outro aspecto a considerar é a gestão das transacções. Uma transacção é uma série de acções levada a cabo por um utilizador ou por uma aplicação, como por exemplo, inserir um novo cliente na base de dados. Se uma transacção é interrompida antes do seu fim (devido, por exemplo, a uma falha de energia ou um sector danificado no disco), a base de dados estará num estado de inconsistência, isto é, algumas alterações foram efectuadas e outras não. O SGBD deverá ser capaz de desfazer o que foi feito até essa altura, de modo a devolver a base de dados a um estado de consistência.

2.2.3 As funções do Sistema de Gestão de Bases de Dados

5. **Controlo de acessos por vários utilizadores em simultâneo:** em bases de dados multi-utilizador, pode acontecer que tentem aceder aos mesmos dados mais do que um utilizador em simultâneo.

O SGBD deve fornecer mecanismos que assegurem que a base de dados é actualizada correctamente, quando isso acontece – controlo da concorrência.

2.2.4 Arquitectura do Sistema de Gestão de Bases de Dados e Independência dos Dados

- Um dos objectivos de um Sistema de Base de Dados é o de fornecer aos utilizadores uma visão abstracta dos dados, isto é, o sistema esconde detalhes sobre a forma como os dados são armazenados e mantidos.
- Na arquitectura ANSI/SPARC7, em 1975, são propostos três níveis de abstracção que podemos ver na figura I.2.5.

2.2.4 Arquitectura do Sistema de Gestão de Bases de Dados e Independência dos Dados

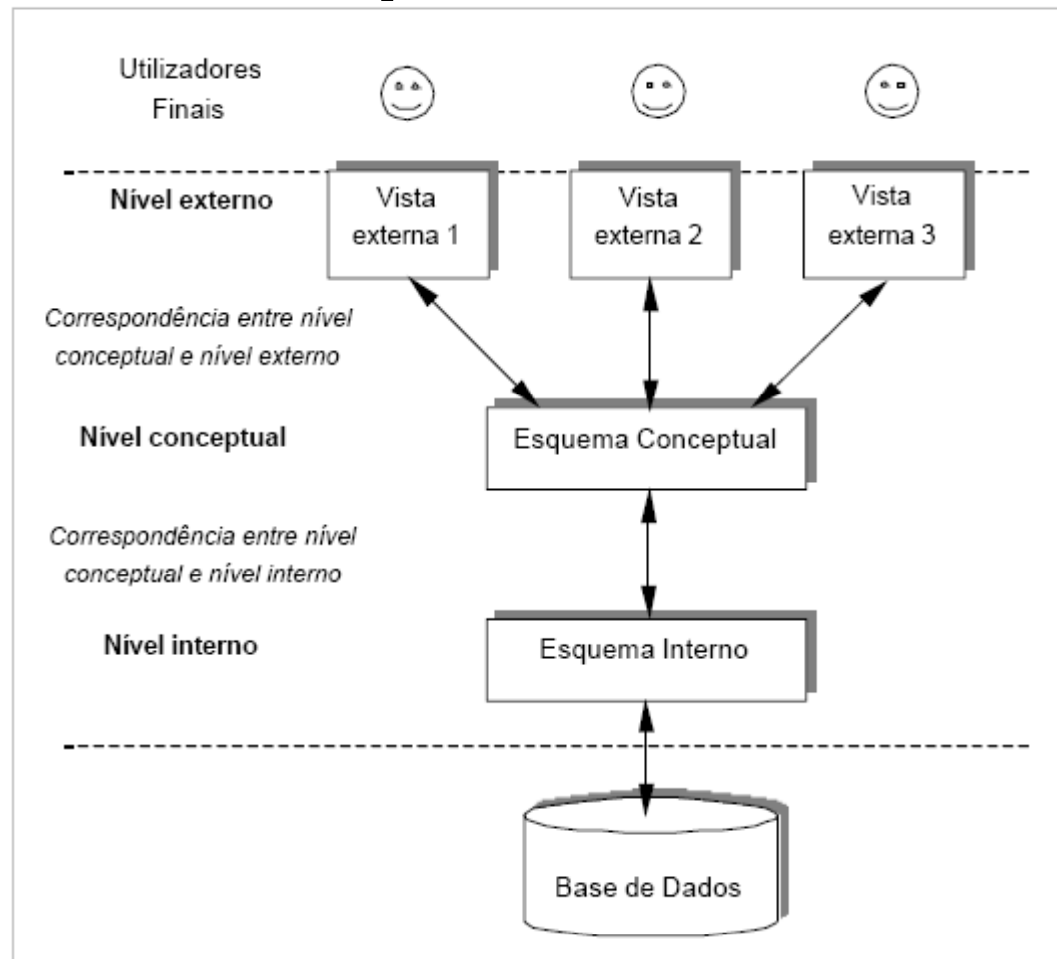


Figura 1.2.5 A arquitectura ANSI/SPARC

2.2.4 Arquitectura do Sistema de Gestão de Bases de Dados e Independência dos Dados

- **Nível Interno:** é o nível mais baixo de abstracção. Existe um esquema interno, o qual descreve a forma como os dados são fisicamente armazenados. Neste nível as estruturas de baixo nível dos dados são descritas em pormenor.
- **Nível conceptual:** no nível seguinte existe um esquema conceptual que descreve a estrutura dos dados para uma comunidade de utilizadores: quais os dados que estão armazenados e quais as relações existentes entre esses dados.
- **Nível externo:** o nível mais elevado de abstracção inclui um número de esquemas externos ou vistas do utilizador, que descrevem apenas uma parte da base de dados.

2.2.4 Arquitectura do Sistema de Gestão de Bases de Dados e Independência dos Dados

- Esta arquitectura de três níveis pode ser utilizada para explicar o conceito de independência dos dados, a qual pode ser definida como a capacidade de alterar o esquema de um determinado nível do sistema de base de dados sem alterar o esquema do próximo nível [Elmasri e Navathe, 1989].
- Podemos definir dois tipos de independência dos dados que são a Independência lógica e a Independência física.

2.2.4 Arquitectura do Sistema de Gestão de Bases de

Dados e Independência dos Dados

- **Independência lógica:** é a capacidade de alterar o esquema conceptual sem ter de alterar os esquemas externos ou programas de aplicação.

Podemos alterar o esquema conceptual para expandir a base de dados ou para a reduzir. No segundo caso, os esquemas externos, relativamente aos dados que se mantêm inalterados, não devem ser afectados.

Num SGBD que suporta independência lógica, apenas é preciso alterar as definições de vistas e as correspondências entre os dois níveis [Elmasri e Navathe, 1989]. Podem também ocorrer mudanças nas restrições existentes, sem que sejam afectados os esquemas externos.

2.2.4 Arquitectura do Sistema de Gestão de Bases de

Dados e Independência dos Dados

- **Independência física:** é a capacidade de alterar o esquema interno sem alterar o esquema conceptual (ou o externo) [Elmasri e Navathe, 1989].

As alterações no esquema interno podem ser feitas por ser necessário reorganizar fisicamente os ficheiros, para melhorar o desempenho do sistema. Se a base de dados continuar a ter os mesmos dados, não deverão haver alterações no esquema conceptual.

É mais fácil conseguir a independência física do que a independência lógica, porque aquela tem apenas a ver com o isolamento das aplicações e das estruturas físicas armazenadas.

2.2.5 Linguagens fornecidas pelo SGBD

O SGBD coloca ao dispor de cada utilizador várias linguagens. Cada uma dessas linguagens inclui uma sublinguagem de dados que consiste de duas partes:

- uma linguagem para manipulação de dados – DML (Data Manipulation Language);
- uma linguagem de definição de dados – DDL (Data Definition Language).

Alguns SGBD`s permitem que as sublinguagens sejam embutidas numa linguagem de alto nível como por exemplo COBOL, PASCAL ou C (linguagens de 3ª geração).

2.2.5 Linguagens fornecidas pelo SGBD

- Uma DDL é utilizada para definir um esquema ou para modificar um esquema já existente.
- Uma DML é utilizada para efectuar todas as operações básicas de manipulação de dados, tais como, inserir novos dados, modificar os dados armazenados ou apagar dados da base de dados.

2.2.5 Linguagens fornecidas pelo SGBD

- Existem DML`s imperativas, que permitem que o utilizador diga ao sistema como se devem efectuar as operações pretendidas e DML`s não imperativas, que permitem ao utilizador definir o que é necessário mas não como obtê-lo (linguagens de 4ª geração).
- Exemplos deste último tipo de DML`s são a SQL (Structured Query Language), QBE (Query By Example), geradores de formulários, geradores de relatórios (na parte II veremos exemplos para o SGBD Access).

2.2.6 Classificação dos SGBD`s

- A classificação de um SGBD pode ser feita atendendo a vários critérios, nomeadamente, o modelo no qual o SGBD se baseia, o número de utilizadores e o número de locais pelos quais a base de dados está distribuída.
- Atendendo ao modelo, os SGBD`s classificam-se em relacionais, em rede, hierárquicos ou outros.
- Em 2.3 abordaremos mais em pormenor estes modelos.

2.2.6 Classificação dos SGBD`s

- Quanto ao número de utilizadores, podemos classificar os SGBD`s em SGBD`s mono-utilizador, se suportam apenas um utilizador de cada vez, ou multi-utilizador, se suportam a utilização simultânea de vários utilizadores.
- Os SGBD`s mono-utilizador estão normalmente ligados à utilização dos computadores pessoais. Exemplos de SGBD`s multi-utilizador encontram-se em sistemas para reservas de viagens de avião ou sistemas para gestão de contas correntes em bancos, etc.
- Um SGBD multi-utilizador deverá possuir capacidades de controlo da concorrência, isto é, possibilitar o acesso simultâneo de vários utilizadores mantendo a base de dados num estado de integridade.

2.2.6 Classificação dos SGBD`s

- Quanto à distribuição, os SGBD`s podem-se classificar em centralizados, se os dados estiverem armazenados apenas num local, ou SGBD`s distribuídos se a base de dados e as suas aplicações estiverem distribuídas por diversos locais, ligados através de uma rede.
- Os SGBD`s distribuídos apresentam características internas específicas e uma maior complexidade que os SGBD`s centralizados.
- No entanto, isso não deve ser visível para os utilizadores.

2.2.7 Vantagens e desvantagens dos Sistemas de Bases de Dados

Ao longo deste capítulo, foram já apresentadas algumas vantagens da utilização dos Sistemas de Bases de Dados, tais como:

- tornar mais fácil a eliminação de inconsistências e de anomalias de dados presentes nos Sistemas de Ficheiros;
- eliminar as dependências estruturais presentes nos sistemas de ficheiros;
- reduzir o tempo destinado ao desenvolvimento de aplicações;
- aumentar a flexibilidade do sistema, permitindo uma maior facilidade nos procedimentos necessários para efectuar alterações na sua estrutura;
- facilitar a obtenção de informação actualizada.

2.2.7 Vantagens e desvantagens dos Sistemas de Bases de Dados

No entanto, os Sistemas de Bases de Dados apresentaram algumas desvantagens que dificultaram a sua generalização imediata, como por exemplo:

- o SGBD é um sistema de grande complexidade, ocupa muito espaço em disco e é mais caro;
- existem custos adicionais de hardware aquando da conversão do sistema existente para um Sistema de Bases de Dados, além dos custos inerentes à própria conversão;
- em caso de falha do sistema, existem normalmente maiores impactos, visto que um Sistema de Bases de Dados é mais abrangente, embora existam mecanismos e procedimentos que minimizam esses impactos.

2.3 MODELOS DE BASES DE DADOS

- Diz-nos Chiavenato que “para se estudar ou analisar um sistema, utilizam-se modelos” e que “modelo é a representação de um sistema”.
- Ao longo dos anos, têm vindo a ser delineados vários modelos para representar um Sistema de Bases de Dados.
- Podemos distinguir entre dois tipos de modelos, os modelos conceptuais e os modelos de implementação.
- Os modelos conceptuais são utilizados para se obter uma descrição lógica do sistema. Um dos mais populares é o modelo Entidade- Relacionamento (E-R).

2.3 MODELOS DE BASES DE DADOS

- Os modelos de implementação descrevem a forma como os dados estão representados na base de dados. Podemos dividir estes modelos em três gerações.
- Na primeira geração colocam-se o modelo hierárquico, que abordaremos em 2.3.1, e o modelo em rede, que será abordado em 2.3.2.
- Na segunda geração temos o modelo relacional, que será abordado em 2.3.3.
- No ponto 2.4 abordaremos os modelos de 3ª geração.

2.3.1 Modelo hierárquico

- As duas estruturas básicas do modelo hierárquico são os registos e os relacionamentos pai-filhos.
- Como o próprio nome do modelo indica, existem vários registos relacionados através de uma estrutura hierárquica do tipo pai-filhos, com vários níveis.
- Cada registo fornece informação sobre uma entidade ou instância de um relacionamento.

2.3.1 Modelo hierárquico

- Um relacionamento do tipo pai-filhos é um relacionamento de 1:M (um para muitos), entre dois registros.
- O primeiro registro é o registro pai e os registros que com ele estão relacionados são os registros filhos.
- O esquema de uma base de dados hierárquica pode ser representado por uma estrutura como a apresentada na figura I.2.6, que representa uma parte da base de dados de uma empresa.

2.3.1 Modelo hierárquico

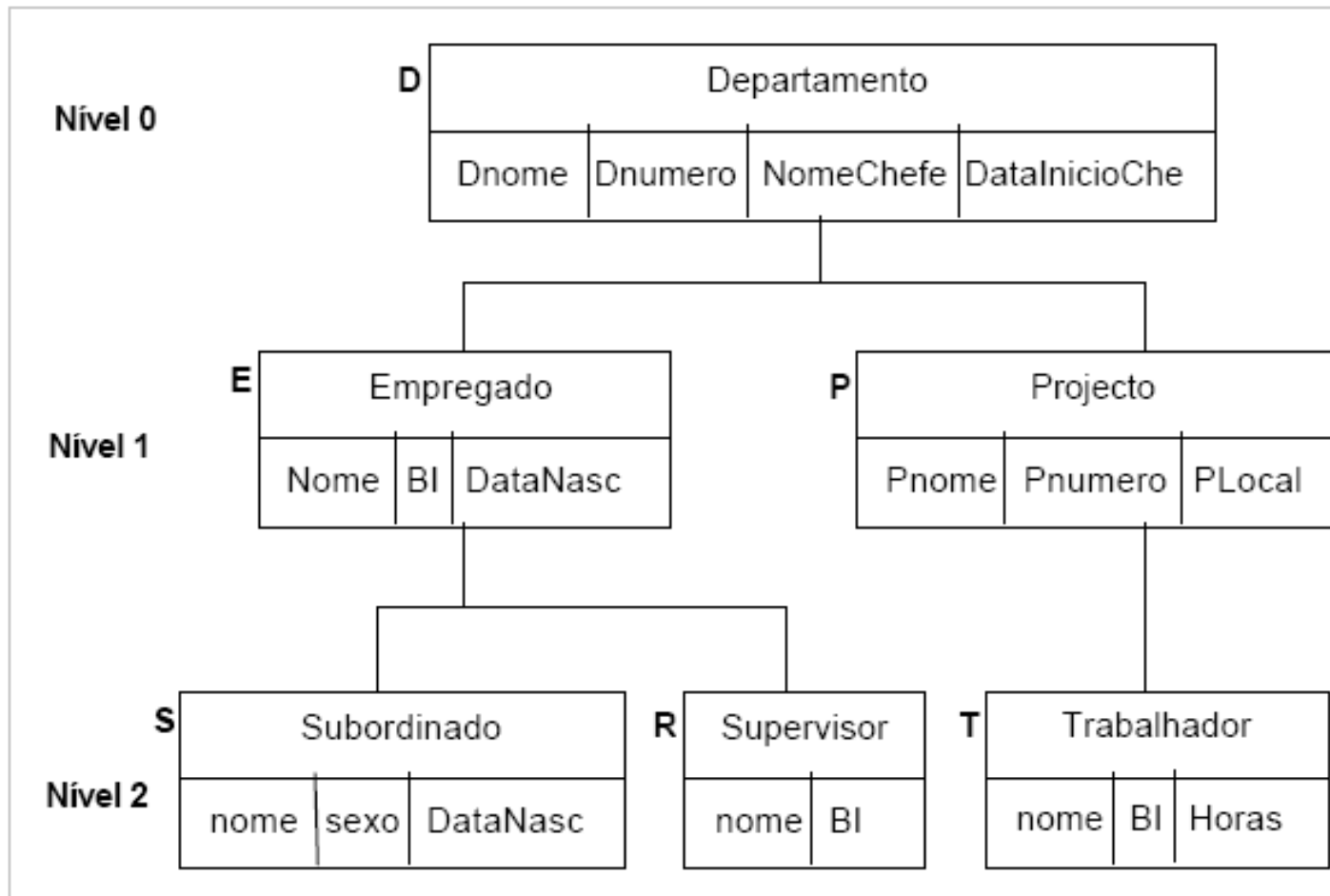


Figura I.2.6 Uma estrutura hierárquica^{1o}

2.3.1 Modelo hierárquico

- Neste diagrama, cada registo é representado por um rectângulo e o relacionamento por uma linha que liga o registo pai ao registo filho.
- Ao registo que se encontra no nível 0, chama-se raiz.
- O acesso a qualquer um dos registos deve ser feito a partir da raiz e correndo cada um dos nós da árvore da esquerda para a direita e de cima para baixo.
- No exemplo considerado, para aceder ao registo projecto (P), teria de se percorrer os registos D, E, S, R e P.

2.3.1 Modelo hierárquico

Existem algumas restrições inerentes ao modelo hierárquico, como por exemplo:

- apenas o registo raiz pode ocorrer sem estar relacionado com um registo pai;
- se um registo filho tem dois ou mais registos pai, então o filho deve ser duplicado para cada um dos pais.

2.3.1 Modelo hierárquico

As bases de dados hierárquicas apresentam alguns problemas, tais como:

- dificuldades de representação de relacionamentos do tipo M:N (muitos para muitos);
- existem redundâncias nos dados, inerentes ao próprio modelo, pois quando um registo filho tem mais do que um pai, deve ser repetido para cada um dos pais;
- ao eliminar um registo, eliminam-se os seus filhos, correndo-se o risco de eliminar informação relevante;
- a programação nos Sistemas de Bases de Dados hierárquicos é muito exigente.

2.3.1 Modelo hierárquico

- Um dos mais difundidos SGBD`s concebidos segundo o modelo hierárquico, comercialmente disponível, foi desenvolvido pela IBM na década de 70 e ficou conhecido como IMS (*Information Management System*).

2.3.2 Modelo em rede

- O modelo em rede surgiu tendo por base o modelo hierárquico; assim sendo, apresenta com ele algumas semelhanças.
- Neste modelo, um relacionamento chama-se **set** e é-lhe atribuído um nome.
- O registo pai chama-se **owner** e o registo filho chama-se **member**.
- Ao contrário do que acontece no modelo hierárquico, no qual um registo filho só pode ter um pai, um *member* pode ter mais do que um *owner*.

2.3.2 Modelo em rede

- O acesso aos dados é feito de forma semelhante ao do modelo hierárquico.
- O esquema de dados em rede pode ser representado por uma estrutura como a representada na figura I.2.7.

2.3.2 Modelo em rede

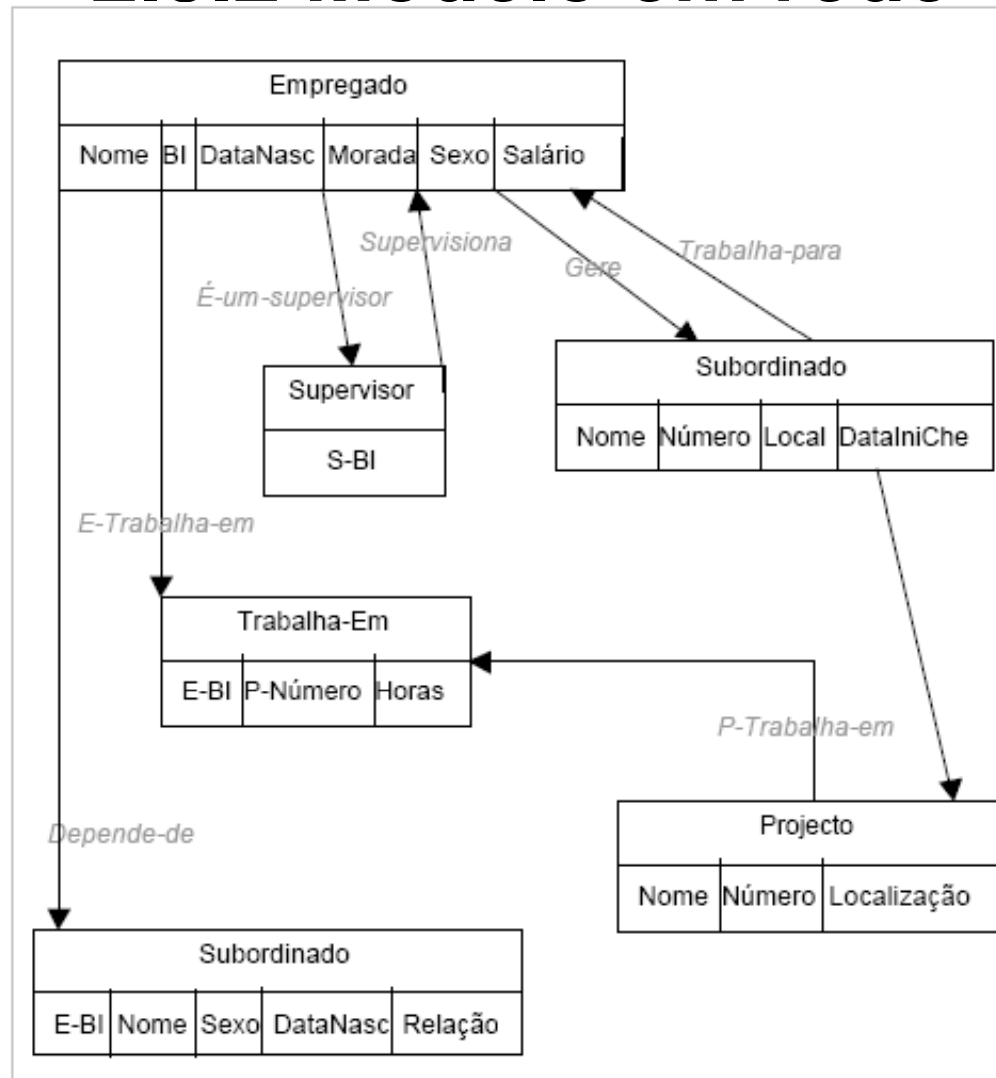


Figura I.2.7 Uma estrutura em rede

2.3.2 Modelo em rede

- Existem algumas restrições que devem ser impostas aos membros de um *set* e que devem ser definidas no momento do desenho da base de dados, dependendo da forma como queremos que eles se comportem.
- Existem opções chamadas de inserção (manuais e automáticas) e opções chamadas de retenção (opcionais, mandatórias ou fixas).

2.3.2 Modelo em rede

- Apesar de apresentar várias vantagens, nomeadamente possibilitar a representação de relacionamentos de M:N e do acesso aos dados ser mais flexível do que no modelo hierárquico, o modelo em rede apresenta também algumas desvantagens.
- Uma das grandes desvantagens tem a ver com o facto de o sistema ser muito complexo, exigindo um conhecimento profundo por parte dos programadores, para que a sua utilização seja feita de forma eficiente, e um grande esforço na sua manutenção, devido ao facto de não existir verdadeiramente independência funcional.
- Além disso, o ambiente é pouco amigável para os utilizadores finais.

2.3.2 Modelo em rede

- Ao contrário daquilo que acontecia no modelo hierárquico, para o qual não existiam normas definidas, o modelo em rede surgiu a partir das normas estabelecidas pelo DBTG, indicado pela CODASYL em 1971.

2.3.3 O Modelo Relacional

- O modelo relacional de bases de dados, apresentado por Edgar F. Codd em 1970, num artigo intitulado “A Relational Model of Data for Large Shared Data Banks”, representa uma quebra com os modelos anteriores e uma verdadeira revolução.
- A estrutura básica do modelo relacional é a tabela, também chamada relação.
- Uma base de dados relacional é formada por um conjunto de tabelas que se relacionam através da partilha de atributos comuns. (Figura I.2.8)

2.3.3 O Modelo Relacional

- Apesar da tabela do modelo parecer um ficheiro, existe uma diferença fundamental entre eles: uma tabela fornece total independência quer dos dados quer funcional, pois é uma estrutura puramente lógica.

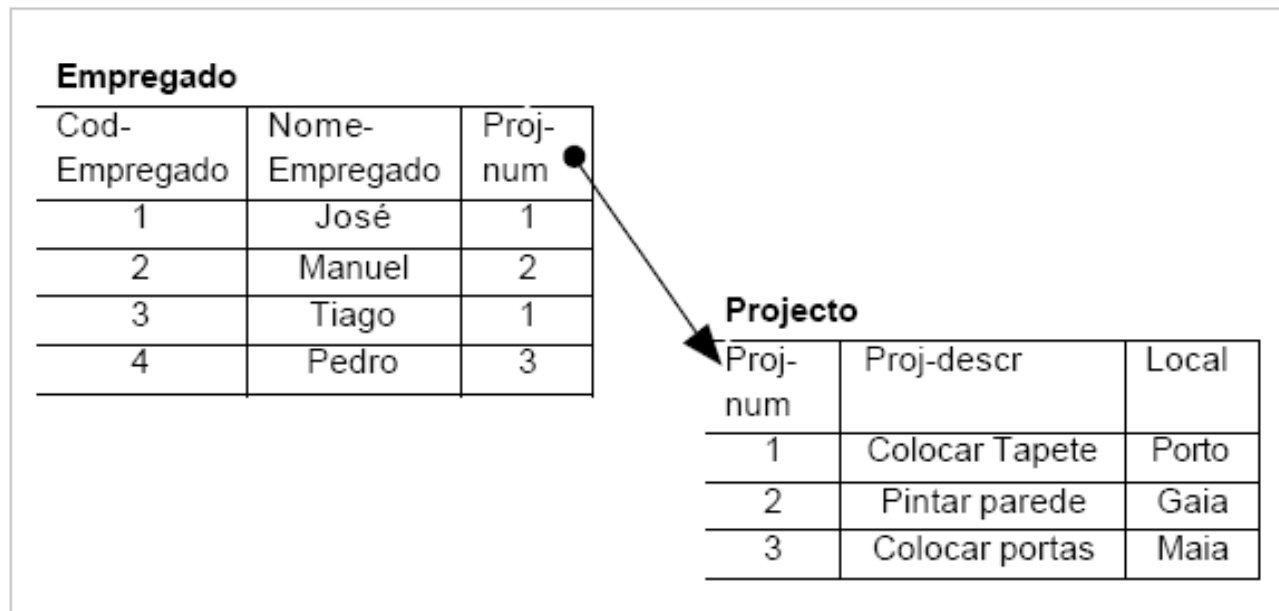


Figura I.2.8 Tabelas do Modelo Relacional

2.3.3 O Modelo Relacional

- Um dos motivos que levaram ao grande sucesso do modelo relacional é a facilidade que possui para responder a questões Ad-hoc, através de uma linguagem própria – SQL (Structured Query Language).
- Devido à sua importância abordaremos mais em pormenor este modelo no próximo capítulo.

2.3.3 O Modelo Relacional

- Apesar de todas as vantagens apresentadas, o modelo relacional apresenta também algumas desvantagens.
- Uma delas tem a ver com o facto de o SGBD relacional ser muito exigente quer a nível do hardware, quer a nível do sistema operativo, devido à sua complexidade.
- Este aspecto tem vindo a ser ultrapassado devido à crescente capacidade dos computadores.

2.3.3 O Modelo Relacional

- Outro aspecto a considerar é o facto de, devido ao seu ambiente amigável, os SGBD`s relacionais serem utilizados sem ter em linha de conta o desenho cuidado das estruturas necessárias, gerando muitas vezes problemas semelhantes aos dos sistemas de ficheiros.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- As bases de dados relacionais têm tido grande aceitação em aplicações clássicas nas organizações, nomeadamente no processamento de ordens de encomendas, no controlo de inventários, marcação de reservas e muitas outras.
- Nos últimos anos, as bases de dados têm aumentado quer em tamanho quer em complexidade nas suas aplicações, colocando exigências com as quais é difícil lidar com a utilização de tecnologias relacionais clássicas.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Nestas aplicações, incluem-se os Sistemas de Informação Geográfica (SIG), aplicações médicas e científicas, os sistemas CAD e CAM, os sistemas multimédia, entre outras.
- Em qualquer destes casos, o volume de informação a tratar é grande e o tipo de objectos a guardar na base de dados é complexo.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Desta forma, surgiram novos modelos de representação de dados, nomeadamente os modelos orientados a objectos, que têm origem nas linguagens de programação orientadas a objectos.
- Como resposta surgiram também extensões ao modelo relacional, de modo a suportarem características de modelos orientados a objectos.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Um **objecto** é uma representação conceptual de uma entidade do mundo real, contendo os **atributos** que o descrevem e o seu comportamento, ou seja, as acções que lhe estão associadas – **encapsulamento** significa que um objecto contém quer a estrutura dos dados quer o conjunto de operações que podemos definir.
- Estas acções chamam-se **métodos**.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Um atributo pode ser uma referência a outro objecto.
- Cada objecto contém um **identificador** único, que é independente dos seus atributos e invisível para o utilizador.
- O estado de um objecto é o conjunto de valores que os atributos do objecto têm num dado momento.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Os métodos podem ser utilizados para alterar o estado dos objectos.
- Os objectos comunicam entre si através de mensagens.
- Uma **mensagem** é um pedido de um objecto a outro (pode ser o mesmo objecto) solicitando-lhe que execute um dos seus métodos.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Uma **classe** é uma colecção de objectos semelhantes que partilham a estrutura e o comportamento, permitindo que os atributos e os métodos associados possam ser definidos uma vez para a classe em vez de separadamente para cada objecto.
- As classes estão organizadas através de uma estrutura hierárquica, na qual os objectos de uma classe herdam os atributos e métodos das suas super-classes – **herança** – além de possuírem atributos e métodos próprios.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- A herança permite que uma classe seja definida como caso especial de uma classe mais geral.
- Os casos especiais são conhecidos como sub-classes e os casos mais gerais são conhecidos como super-classes.
- Podemos falar de herança simples, sendo que neste caso um objecto pode ter apenas uma superclasse pai e de herança múltipla, sendo que neste caso um objecto pode ter duas ou mais super-classes pai.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- O **polimorfismo** é uma característica importante dos sistemas orientados a objectos.
- Significa que a mesma mensagem enviada a objectos pertencentes a classes diferentes, origina a resposta correcta em cada caso, ou seja, podemos utilizar o mesmo nome para métodos em classes diferentes, pois o seu comportamento será adequadamente diferente em resposta à mesma mensagem.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- O processo de seleccionar o método apropriado baseado num tipo de objecto chama-se **binding**.
- Quando este tem lugar ao executar o programa, diz-se que existe *binding* dinâmico.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Um SGBD orientado a objectos resulta da combinação das características tradicionais de um SGBD (tais como gestão do armazenamento dos dados, gestão da segurança, manutenção da integridade dos dados) e das características de sistemas orientados a objectos, tais como, herança, encapsulamento e polimorfismo.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- O “Manifesto dos sistemas de bases de dados orientados a objectos” apresentado por Atkison, et al em 1989 e citado por [Connolly e Beg, 1999], apresenta treze regras que descrevem todas as características desejáveis de um SGBD Orientado a Objectos.
- O manifesto contém dois conjuntos de regras apresentadas na figura I.2.9.
- Já na década de 90, foi formado por vários vendedores de SGBD`s um grupo de trabalho que pretendeu definir normas no que concerne a este assunto – o Object Database Management Group (ODMG).

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

Este grupo propõe um modelo de dados orientado a objectos que consiste de três partes:

- um modelo de objectos;
- uma linguagem de definição de objectos (equivalente a uma linguagem de definição de dados de um SGBD convencional);
- uma linguagem para questões, em ambientes orientados a objectos, com sintaxe semelhante à do SQL.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

Deve ser um sistema orientado a objectos	Deve ser um SGBD
1. O sistema deve suportar objectos complexos	9. Os dados devem ser permanentes
2. O sistema deve suportar a identidade de um objecto	10. O SGBD deve ser capaz de lidar com bases de dados muito grandes
3. O sistema deve suportar encapsulamento	11. O SGBD deve suportar utilizadores concorrentes
4. Devem ser suportados tipos ou classes	12. O SGBD deve ser capaz de recuperar de falhas de hardware e software
5. O sistema deve suportar herança na estrutura hierárquica de classes	13. As questões a colocar sobre os dados devem poder ser efectuadas de forma simples
6. O sistema deve suportar Binding dinâmico	
7. A linguagem de manipulação de dados deve ser completa	
8. O conjunto de tipos de dados deve ser extensível	

Figura I.2.9 Regras do "Manifesto dos Sistemas de Bases de Dados Orientados a Objectos"

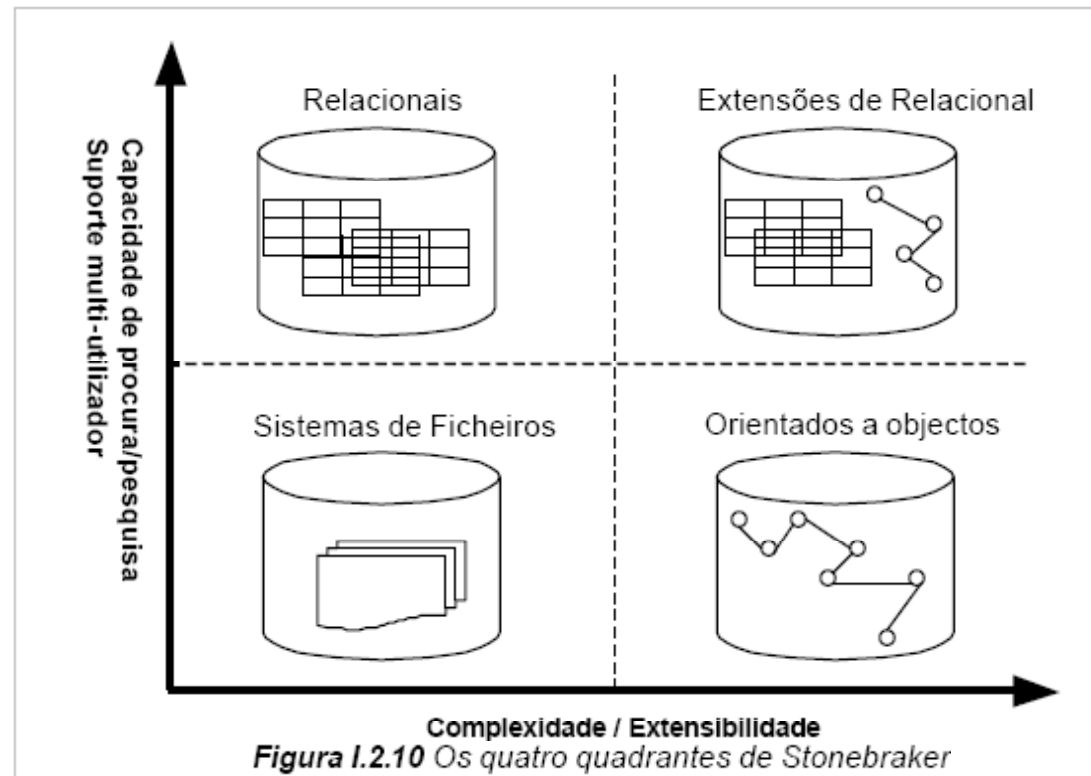
2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- As rápidas mudanças que se têm vindo a verificar no ambiente das bases de dados, forçaram os defensores das bases de dados relacionais a uma resposta ao desafio colocado pelos ambientes orientados a objectos, através da criação de extensões ao modelo relacional.
- Não existe um modelo relacional extendido único.
- Existem sim várias propostas que diferem na forma como as extensões são feitas.
- Em comum existem as tabelas do modelo relacional, a linguagem para questões e a incorporação de alguns conceitos de modelos orientados a objectos.

2.3.4 Novos modelos: Modelos Orientados a Objectos e

Extensões ao Modelo Relacional

- Stonebraker`s propôs uma visualização do mundo das bases de dados classificando-as segundo quatro quadrantes, como apresentado na figura I.2.10.



2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Apesar de esta classificação ser interessante, existem muitas aplicações de bases de dados que não se revêm facilmente num dos quadrantes.
- Além disso, com o trabalho realizado pelo grupo ODMG (Object Data Management Group) a distinção entre extensão do modelo relacional e modelo orientado a objectos é cada vez menos clara.

2.3.4 Novos modelos: Modelos Orientados a Objectos e Extensões ao Modelo Relacional

- Actualmente existem debates entre os proponentes dos dois modelos.
- Ambos estão de acordo em que o modelo relacional é insuficiente para certos tipos de aplicações, diferindo, no entanto, na sua opinião quanto à melhor solução.
- É difícil prever se uma das abordagens se tornará dominante ou se cada uma delas ocupará um nicho de mercado próprio.

2.4 TENDÊNCIAS FUTURAS

- No final da década anterior surgiram novos desenvolvimentos na área das bases de dados.
- Dado que se encontram numa fase inicial do seu desenvolvimento, intitulamos esta secção de Tendências Futuras”.

2.4.1 Data Warehouses

- Uma Data Warehouse (armazém de dados) é uma base de dados que agrupa um grande volume de dados.
- Esses dados são otimizados para futuras análises, servindo de base na tomada de decisão nas organizações.
- O chamado “pai das Data Warehouses”, Bill Inmon, define-as como uma colecção de dados integrados, orientada para determinados assuntos, variável no tempo e não volátil, que fornece apoio na tomada de decisão.

2.4.1 Data Warehouses

- Os dados dizem-se integrados pois numa Data Warehouse consolidam-se os dados provenientes de diversas fontes, muitas vezes com formatos distintos.
- A integração dos dados implica um esforço bem organizado para definir e padronizar todos os dados, de modo a torná-los consistentes, para que possam ser apresentados aos utilizadores de forma centralizada.

2.4.1 Data Warehouses

- A colecção de dados diz-se orientada para determinados assuntos visto estar organizada de modo a fornecer respostas a questões provenientes de diversas áreas funcionais de uma companhia (tais como, clientes, produtos, vendas,...).
- Na medida em que os dados, na Data Warehouse, representam o fluxo de dados ao longo de um determinado período (a última semana, o último mês, o último ano,...) diz-se que são variáveis no tempo.

2.4.1 Data Warehouses

- Além disso, a variabilidade no tempo também está presente na medida em que os dados são periodicamente actualizados, provocando que todas as agregações que dependem da variável tempo, sejam novamente calculadas.
- Visto que os dados, na Data Warehouse, não são actualizados em tempo real, mas sim regularmente recolhidos a partir dos sistemas operacionais, (OLTP's "On Line Transaction Processing" ou sistemas de processamento de transacções) e visto que os dados existentes não são nunca removidos sendo os novos dados adicionados continuamente, diz-se que os dados são não voláteis.

2.4.1 Data Warehouses

- Pela breve descrição apresentada, podemos verificar que estas bases de dados têm características completamente diferentes das bases de dados dos chamados sistemas de processamento de transacções, que apresentámos ao longo das secções 2.1 a 2.3.
- Estas são concebidas de modo a maximizar a capacidade de processamento de um grande número de transacções que são previsíveis, repetitivas e utilizadas muitas vezes, enquanto que aquelas são concebidas de modo a maximizar a capacidade de resposta a um número relativamente baixo de transacções que são de natureza imprevisível e requerem respostas a questões Ad hoc, não estruturadas e heurísticas.

2.4.1 Data Warehouses

- Naturalmente, sendo de naturezas distintas, apresentam características distintas.
- Na figura I.2.11 apresentamos uma comparação entre as bases de dados utilizadas nos sistemas OLTP e as Data Warehouses, utilizadas pelos sistemas de apoio à decisão (DSS – Decision Support Systems).

2.4.1 Data Warehouses

- Dado que a criação das Data Warehouses envolve custos elevados, muito tempo despendido e um esforço considerável por parte da gestão, muitas empresas iniciam a sua incursão no mundo das Data Warehouses, por um conjunto de dados mais manejáveis, orientados para pequenos grupos dentro da organização – as Data Marts.
- Numa tradução directa de Mart, obtemos mercado, centro de comércio.

2.4.1 Data Warehouses

- Podemos definir uma Data Mart como um subconjunto de uma Data Warehouse que suporta as exigências de um dado departamento ou de uma função do negócio.
- Uma Data Mart pode estar isolada ou pode estar ligada a uma Data Warehouse central.

2.4.1 Data Warehouses

Bases de dados utilizadas em OLTP's	Data Warehouses
Mantêm dados actuais	Mantêm dados históricos
Guardam dados detalhados	Guardam dados detalhados e grande-mente resumidos
Os dados são dinâmicos	Os dados são maioritariamente estáticos
Processamentos repetitivos	Processamentos Ad hoc, não estrutura-dos e heurísticos
Grande número de transacções	Médio a baixo nível de transacções
Padrão de utilizações previsível	Padrão de utilizações não previsível
Orientadas para transacções	Orientadas para análise
Orientadas para aplicações	Orientadas para determinados assuntos
Suportam decisões do dia-a-dia	Suportam decisões estratégicas
Servem um grande número de utiliza-dores de nível operacional	Servem um número relativamente baixo de utilizadores de nível administrativo

Figura I.2.11 Comparação entre bases de dados utilizadas em OLTP's e Data Warehouses

2.4.1 Data Warehouses

- Os dados armazenados numa Data Warehouse servirão para responder a questões sobre determinados factos analisados de várias formas – perspectiva multi-dimensional.
- O *star schema* é uma técnica de modelação de dados que gera um modelo facilmente utilizável numa perspectiva de análise multi-dimensional dos dados, preservando, ainda assim, as estruturas relacionais.

2.4.1 Data Warehouses

- Até há muito pouco tempo, as ferramentas disponíveis para aceder a grandes volumes de dados forneciam apenas uma análise limitada e relativamente simplista desses dados.
- O nascimento das Data Warehouses tem sido acompanhado por uma exigência crescente por parte dos seus utilizadores no que concerne a ferramentas de acesso mais poderosas.
- Como resposta, têm vindo a ser desenvolvidas ferramentas tais como On-Line Analytical Processing (OLAP`s) e Data Mining, que fornecem capacidades analíticas avançadas.

2.4.2 OLAP`s

- OLAP, do inglês On-Line Analytical Processing, são ferramentas que proporcionam ambientes para análise avançada de dados, fazendo a síntese, análise e consolidação de grandes volumes de dados, armazenados numa perspectiva multi-dimensional.
- A perspectiva multi-dimensional dos dados permite que a análise dos dados do negócio seja feita sob várias perspectivas (dimensões): vendas por cliente, vendas por região, vendas por vendedor, etc.

2.4.2 OLAP`s

As técnicas de análise de dados numa perspectiva multi-dimensional podem ser ampliadas através das seguintes funções:

- funções de representação de dados tais como gráficos 3-D;
- funções para agregação, consolidação e classificação de dados que permitem ao analista dos dados do negócio a criação de níveis de agregação múltiplos, a visualização dos dados sob diferentes perspectivas e a apresentação dos dados pormenorizados, contidos nos dados consolidados;
- funções de cálculo, tais como variáveis de negócio (margens de venda, quotas de mercado, etc.), rácios financeiros e contabilísticos (lucro, retorno, etc), funções estatísticas, etc.;
- funções de modelação de dados tais como cenários do tipo Se-Então e programação linear.

2.4.2 OLAP`s

- As características dos OLAP`s tornam-se muito mais úteis se o acesso a elas for simples.
- Assim sendo, é importante a existência de interfaces amigáveis.

2.4.2 OLAP`s

- Um dos objectivos do processamento de dados é processar os dados contidos em bases de dados com uma complexidade crescente, sem sacrificar os tempos de resposta.
- As arquitecturas cliente/servidor permitem dividir o sistema OLAP em vários componentes que podem ser colocados no mesmo computador ou podem ser distribuídos por vários computadores.

2.4.2 OLAP`s

As ferramentas OLAP podem ser organizadas segundo duas categorias principais:

- ***MOLAP (Multidimensional OLAP)*** As ferramentas MOLAP utilizam estruturas de dados apropriadas e SGBD`s multi-dimensionais para organizar, navegar e analisar os dados. Exemplos de ferramentas MOLAP incluem Analysis Server, Essbase, Express Server;
- ***ROLAP (Relational OLAP)*** As ferramentas ROLAP são a categoria de OLAP`s que se tem desenvolvido mais rapidamente. As ferramentas ROLAP utilizam SGBD`s relacionais, através da utilização de camadas de metadados, evitando assim a criação de estruturas de dados multidimensionais estáticos. Exemplos de ferramentas ROLAP incluem Axsys, DSS Agent/DSS Server, Metacube.

2.4.2 OLAP`s

- A tendência actual é que as ferramentas MOLAP e ROLAP convirjam, produzindo um novo tipo de SGBD`s que utilizem as melhores capacidades de cada uma das ferramentas.

2.4.3 Data Mining

- Só conseguimos perceber o verdadeiro valor de uma Data Warehouse se formos capazes de extrair características dos dados, relações, dependências ou tendências (padrões) previamente desconhecidos. As ferramentas de Data Mining são uma das melhores formas de o conseguir.
- As ferramentas de Data Mining são pró-activas, isto é, em vez de ser o utilizador a definir o problema e seleccionar os dados e as ferramentas que vão analisá-los, as ferramentas de Data Mining pesquisam os dados de forma automática, à procura de anomalias e relações, identificando problemas não identificados previamente pelo utilizador final.

2.4.3 Data Mining

- As ferramentas de Data Mining são recentes e assim sendo, não existem normas estabelecidas.
- Existem sim, famílias de ferramentas de Data Mining dirigidas a determinados nichos de mercado (marketing, finanças, saúde,...), utilizando algoritmos que vão ser aplicados a diversos dados de modo a obter conhecimento.
- Os algoritmos tipicamente utilizados podem ser baseados em redes neuronais, árvores de decisão, regras de indução, algoritmos genéticos, classificação e árvores de regressão, visualização de dados, etc.

2.4.3 Data Mining

- Um exemplo de um resultado obtido pelas ferramentas de Data Mining pode ser que 90% dos clientes que compraram uma televisão têm uma apetência de 88% de comprar um vídeo nas 4 semanas seguintes.
- Este resultado pode ser utilizado na tomada de decisão relativa a uma campanha de marketing de vídeos. Exemplos de ferramentas de Data Mining incluem Clementine, Itelligent Miner, Mineset, etc.
- Para um maior aprofundamento deste tema sugerimos a leitura de [Fayad et al, 1996].

2.4.4 Integração com a Web

- A World Wide Web ou de forma abreviada a Web ou WWW, é para muitas pessoas sinónimo de Internet.
- O pai da Web é Tim Berners-Lee, do CERN (laboratório Europeu de Física das Partículas), que teve a ideia de criar uma teia electrónica de informação sobre investigação.
- Hoje encontramos na Web informação sobre quase tudo. Foi desenvolvida uma linguagem chamada HTML – Hyper Text Mark-up Language – que consiste num conjunto de instruções, inseridas pelo programador, em texto simples.

2.4.4 Integração com a Web

- Cada uma das páginas Web é um documento em Hipertexto.
- O conceito de Hipertexto foi inventado nos anos 50, por Ted Nelson, e significa que uma porção de texto num documento pode remeter para outros pedaços de texto dentro do mesmo documento ou de outros documentos.
- Inicialmente as páginas da Web eram baseadas apenas em texto mas actualmente apresentam características de multimédia.

2.4.4 Integração com a Web

- Nos últimos tempos, têm-se verificado o desenvolvimento de tecnologias que permitem a integração de sistemas de gestão de bases de dados com a Web, através da utilização de arquitecturas adequadas.
- Existem páginas Web cujo conteúdo não se altera a não ser que o próprio ficheiro (em HTML) seja alterado – páginas estáticas.
- É cada vez mais comum a existência de páginas Web que são geradas de cada vez que são acedidas – páginas dinâmicas.

2.4.4 Integração com a Web

- As bases de dados, pelas suas características eminentemente dinâmicas, prestam-se a uma integração na Web através da criação de páginas dinâmicas.
- Nesta situação, o hipertexto deve ser automaticamente gerado pelos servidores, através de *scripts* que fazem a conversão de diferentes formatos de dados para HTML.
- Existem algumas aproximações possíveis, tais como utilização da técnica CGI (*Common Gateway Interface*), *cookies*, as linguagens *JavaScript* ou *VBScript*.
- O Ms Access inclui algumas funcionalidades para a Web.