

Curso de Bacharelado em Biblioteconomia na Modalidade a Distância

José Simão de Paula Pinto

Planejamento e Elaboração de Bases de Dados

Semestre

8

Curso de Bacharelado em Biblioteconomia na Modalidade a Distância

José Simão de Paula Pinto

Planejamento e Elaboração de Bases de Dados

Semestre

8

Brasília, DF



1920 | 2020
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Rio de Janeiro

Faculdade de Administração
e Ciências Contábeis
Departamento
de Biblioteconomia

2018



Permite que outros remixem, adaptem e criem a partir do seu trabalho para fins não comerciais, desde que atribuam o devido crédito ao autor e que licenciem as novas criações sob termos idênticos.

Presidência da República

Ministério da Educação

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)

Diretoria de Educação a Distância (DED)

Sistema Universidade Aberta do Brasil (UAB)

Universidade Federal do Rio de Janeiro (UFRJ)

Núcleo de Educação a Distância (NEAD)

Faculdade de Administração e Ciências Contábeis (FACC)

Departamento de Biblioteconomia

Leitor

Leobino Nascimento Sampaio

Comissão Técnica

Célia Regina Simonetti Barbalho

Helen Beatriz Frota Rozados

Henriette Ferreira Gomes

Marta Lúcia Pomim Valentim

Comissão de Gerenciamento

Mariza Russo (*in memoriam*)

Ana Maria Ferreira de Carvalho

Maria José Veloso da Costa Santos

Nadir Ferreira Alves

Nysia Oliveira de Sá

Equipe de Apoio

Eliana Taborda Garcia Santos

José Antonio Gameiro Salles

Maria Cristina Paiva

Miriam Ferreira Freire Dias

Rômulo Magnus de Melo

Solange de Souza Alves da Silva

Coordenação de

Desenvolvimento Instrucional

Cristine Costa Barreto

Desenvolvimento Instrucional

Fernanda Felix

Diagramação

André Guimarães de Souza

Revisão de Língua Portuguesa

Patrícia Sotello

Projeto Gráfico e Capa

André Guimarães de Souza

Patrícia Seabra

Normalização

Lamas Consultoria

P659p Pinto, José Simão de Paula.

Planejamento e elaboração de bases de dados / José Simão de Paula Pinto; [leitor] Leobino Nascimento Sampaio. – Brasília, DF : CAPES : UAB ; Rio de Janeiro, RJ : Departamento de Biblioteconomia, FACC/UFRJ, 2018.

164 p. : il.

Inclui bibliografia.

ISBN 978-85-85229-77-1 (brochura)

ISBN 978-85-85229-69-6 (e-book)

1. Bases de dados. 2. Bases de dados em rede. I. Sampaio, Leobino Nascimento. II. Título.

CDD 020.92

CDU 023.4

Caro leitor,

A licença CC-BY-NC-AS, adotada pela UAB para os materiais didáticos do Projeto BibEaD, permite que outros remixem, adaptem e criem a partir desses materiais para fins não comerciais, desde que lhes atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. No interesse da excelência dos materiais didáticos que compõem o Curso Nacional de Biblioteconomia na modalidade a distância, foram empreendidos esforços de dezenas de autores de todas as regiões do Brasil, além de outros profissionais especialistas, a fim de minimizar inconsistências e possíveis incorreções. Nesse sentido, asseguramos que serão bem recebidas sugestões de ajustes, de correções e de atualizações, caso seja identificada a necessidade destes pelos usuários do material ora apresentado.

LISTA DE FIGURAS

Figura 1 - Esquema comparativo entre o funcionamento do sistema de arquivos e o sistema de gerenciamento de banco de dados.....	22
Figura 2 - Os SGBDs permitem que façamos compras e investimentos on-line	23
Figura 3 - Assim como apenas guardar as notas fiscais não vai ajudar uma pessoa a gerenciar suas finanças	29
Figura 4 - Uma base de dados pode ser composta por vários bancos de dados	30
Figura 5 - A usabilidade e a performance de um SGBD interferem na sua percepção de uso de um determinado sistema	30
Figura 6 - Níveis de abstração na arquitetura de bancos de dados ANSI-SPARC	34
Figura 7 - Exemplo de tabela de banco de dados relacionais.....	36
Figura 8 - Ao fazer uma compra via <i>internet</i> , você certamente estará interagindo com um banco de dados	37
Figura 9 - Síntese dos conceitos apresentados	38
Figura 10 - Representação da comunicação pela rede no ambiente cliente x servidor	42
Figura 11 - O par login/identificador (<i>user name</i>) e senha (<i>password</i>), após o usuário fazer a digitação das informações nos respectivos campos	43
Figura 12 - Ciclo de vida de um sistema de informação para modelagem de negócios, de acordo com <i>Martin e Filkelstein</i> (1984)	46
Figura 13 - Antes de planejar modelar um banco de dados, é preciso conhecer os seus futuros usuários. No caso de uma biblioteca de uma universidade.....	49
Figura 14 - É importante observar a rotina do usuário, seu modo de buscar informações e os tipos de informações pesquisadas.....	50
Figura 15 - Exemplo de diagrama de casos de uso.....	51
Figura 16 - A identificação dos usuários padrão e sua colaboração são fundamentais para a elaboração da modelagem do banco de dados.....	54
Figura 17 - A participação do usuário é central no processo, desde a concepção, passando pelo desenho, pela implementação e pela validação de um banco de dados.....	56

Figura 18 - Representação da sequência de projeto de bancos de dados	58
Figura 19 - Criação de um modelo conceitual a partir dos requisitos do usuário	58
Figura 20 - Representação gráfica genérica do Modelo Entidade x Relacionamento (MER)	68
Figura 21 - Representação de uma entidade e seus tipos de atributos	71
Figura 22 - Exemplo de representação MER no programa <i>Ferret</i>	72
Figura 23 - Exemplo de generalização e especialização	73
Figura 24 - Exemplo de modelagem com especialização	74
Figura 25 - Modelagem de associações entre as entidades pessoa e livro.....	75
Figura 26 - <i>British Museum Library</i> e <i>Magee Public Library</i> . Modelar significa entender as necessidades de cada instituição e modelar os dados	76
Figura 27 - Exemplos de autorrelacionamento entre as entidades ...	76
Figura 28 - Exemplos de entidade	77
Figura 29 - Exemplos de relacionamento	77
Figura 30 - Exemplo de MER contendo relacionamento quaternário...	78
Figura 31 - Exemplo de possível solução para o estudo de caso 1 ...	81
Figura 32 - Exemplo de mapeamento entre os modelos ER e Relacional.....	91
Figura 33 - Exemplo de MER com aspectos físicos	97
Figura 34 - Exemplo de criação de BD	98
Figura 35 - Exemplo do comando <i>USE</i> para indicar qual BD será usado para os próximos comandos	99
Figura 36 - Exemplo do comando <i>CREATE TABLE</i> para criar as tabelas no BD em uso.....	102
Figura 37 - Exemplo do comando <i>SHOW TABLES</i> para mostrar as tabelas no BD em uso.....	103
Figura 38 - Exemplo do comando <i>DESC</i> para mostrar a estrutura de uma tabela no BD em uso.....	103
Figura 39 - Exemplo do comando <i>DESC</i> para mostrar a estrutura de uma tabela no BD em uso.....	104
Figura 40 - Exemplo do comando <i>INSERT INTO</i> para mostrar a inserção de dados em uma tabela no BD em uso, no caso, a Tabela Autores.....	106

Figura 41 - Exemplo do comando <i>INSERT INTO</i> usado para tentar inserir novamente dados já cadastrados na tabela autores no BD em uso	106
Figura 42 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Usuarios no BD em uso	107
Figura 43 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Livros no BD em uso.....	109
Figura 44 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Livro-autores no BD em uso	110
Figura 45 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Livro-autores no BD em uso	110
Figura 46 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Livro-autores no BD em uso	111
Figura 47 - Exemplo do comando <i>INSERT INTO</i> usado para inserir dados na tabela Empréstimos no BD em uso.....	113
Figura 48 - Exemplo do comando <i>UPDATE</i> usado para inserir dados de devolução na tabela Empréstimos do BD	114
Figura 49 - Exemplo do comando <i>SELECT</i> usado para selecionar e exibir dados da tabela Livros do BD em uso	116
Figura 50 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso.....	116
Figura 51 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso, selecionando por ISBN.....	117
Figura 52 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso, selecionando por Título	117
Figura 53 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso, selecionando por Título, em ordem alfabética	118
Figura 54 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso, selecionando por Título, em ordem descendente	118
Figura 55 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso, selecionando por ISBN.....	119
Figura 56 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso.....	119
Figura 57 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso	120
Figura 58 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso	121
Figura 59 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso.....	121

Figura 60 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso	121
Figura 61 - Exemplo de resultado de lista de dados da tabela Livros do BD em uso	122
Figura 62 - Exemplo de resultado usando o comando <i>WHERE</i>	122
Figura 63 - Exemplo de resultado sem usar o comando <i>WHERE</i>	123
Figura 64 - Exemplo de resultado usando o comando <i>Select</i> para filtrar por CPF	124
Figura 65 - Exemplo de resultado sem usar o comando <i>WHERE</i> buscando o nome dos usuários	124
Figura 66 - Exemplo de resultado usando o comando <i>SELECT DISTINCT</i> e <i>WHERE</i> buscando o nome dos usuários com base no CPF aplicando <i>NOT IN</i>	125
Figura 67 - Exemplo de resultado usando o comando <i>SELECT DISTINCT</i> e <i>WHERE</i> buscando o nome dos usuários com base no CPF aplicando <i>NOT IN</i>	125
Figura 68 - Exemplo de resultado usando o comando <i>DATE_</i> <i>FORMAT</i>	126
Figura 69 - Exemplo de resultado usando o comando <i>DATEDIFF</i> ...	126
Figura 70 - Exemplo de resultado usando o comando <i>CURDATE</i> ...	126
Figura 71 - Exemplo de resultado usando o comando <i>CURDATE</i> na tabela Empréstimos.....	127
Figura 72 - Exemplo de resultado usando o comando Componentes	128
Figura 73 - Exemplo de resultado usando o comando <i>SELECT</i>	129
Figura 74 - Exemplo de resultado usando o comando <i>ALTER TABLE</i> ...	129
Figura 75 - Exemplo de resultado usando o comando <i>DELETE</i> para dados menores que 100	130
Figura 76 - Exemplo de resultado usando o comando <i>DELETE</i> para dados maiores que 99	130
Figura 77 - Exemplo de resultado usando o comando <i>INSERT INTO</i> para verificar se os dados foram totalmente apagados do BD	131
Figura A.1 - Tela inicial da ferramenta brMODELO.....	142
Figura A.2 - Utilização da ferramenta brMODELO para criação de uma nova entidade pessoa	143
Figura A.3 - Utilização da ferramenta brMODELO para criação de atributos na entidade pessoa	144

Figura A.4 - Utilização da ferramenta brMODELO para ajuste do atributo composto endereço na entidade pessoa....	144
Figura A.5 - Utilização da ferramenta brMODELO para ajuste de características dos atributos conforme o dicionário de dados do modelo	145
Figura A.6 - Criação do modelo lógico a partir do modelo conceitual	145
Figura A.7 - Criação do modelo lógico a partir do modelo conceitual: outras opções	146
Figura A.8 - Utilização da ferramenta brMODELO para gerar modelo lógico: cardinalidade ajustada	147
Figura A.9 - Utilização da ferramenta brMODELO para gerar o esquema físico	147
Figura B.1 - Execução de comandos no <i>MySQL</i> em interface texto por meio do terminal no sistema <i>Ubuntu</i>	153
Figura B.2 - Criação de conexão com o <i>MySQL</i> dentro do <i>workshop</i>	154
Figura B.3 - Fornecer senha e confirmar conexão com o <i>MySQL</i>	155
Figura B.4 - Visão da tela do <i>MySQL Workbench</i> após logado	155
Figura B.5 - Auxílio sensível ao contexto	156
Figura B.6 - Barras de ferramentas do <i>MySQL Workbench</i> e seus ícones.....	156
Figura B.7 - Principais comandos presentes no quadro de navegação e gerenciamento	160
Figura B.8 - Edição de diagramas no <i>Workbench</i>	161
Figura B.9 - Aspecto da tela de edição de diagramas	161
Figura B.10 - Criação de tabelas	161
Figura B.11 - Tela de inserção de dados na tabela criada	162
Figura B.12 - Criação de relacionamentos entre as tabelas no editor de diagramas	162

LISTA DE QUADROS

Quadro 1 – Exemplo de aplicação de banco de dados baseada em sistema de arquivos	20
Quadro 2 - Exemplo de diferenciação entre dado e metadado usando o contexto de informações de obras em uma biblioteca	25
Quadro 3 - Tipos de dados usuais em bancos de dados relacionais ...	27
Quadro 4 - Tipos de integridade de dados em SGBDs	40
Quadro 5 - Tipos de entidades e suas descrições	69
Quadro 6 - Sugestões de implementações de tabelas a partir da Modelagem ER	92
Quadro 7 - Exemplo de dicionário de dados	94
Quadro A.1 - Dicionário de dados	144
Quadro A.2 - Funções dos ícones da barra de ferramentas básicas do <i>software</i> brMODELO	148
Quadro A.3 - Funções dos ícones da barra de ferramentas para modelagem conceitual do <i>software</i> brMODELO	149
Quadro A.4 - Funções dos ícones da barra de ferramentas para modelagem lógica da ferramenta brMODELO	150
Quadro B.1 - Funções dos ícones da barra de ferramentas de assistentes	157
Quadro B.2 - Funções dos ícones da barra de ferramentas do editor de comandos	158

SUMÁRIO

	APRESENTAÇÃO	15
1	UNIDADE 1: CONCEITOS RELATIVOS A BANCOS DE DADOS	17
1.1	OBJETIVO GERAL.....	17
1.2	OBJETIVOS ESPECÍFICOS.....	17
1.3	PRÉ-REQUISITOS	17
1.4	A IMPORTÂNCIA DA ESTRUTURAÇÃO DAS BASES DE DADOS E A BIBLIOTECONOMIA.....	19
1.4.1	Das aplicações de bancos de dados ao sistema gerenciador de bancos de dados	20
1.4.2	Conceitos	24
1.4.2.1	<i>Dados, metadados, informação e conhecimento</i>	24
1.4.2.2	<i>Bancos de dados e bases de dados</i>	29
1.4.2.3	Atividade	33
1.4.2.4	<i>Níveis de abstração em bancos de dados</i>	33
1.4.2.5	<i>Tarefas do sistema gerenciador de bancos de dados</i>	36
1.4.2.6	<i>Ambiente de rede e segurança</i>	42
1.4.3	Desenvolvimento de sistemas e bancos de dados	44
1.4.3.1	<i>Levantamento de dados e requisitos do sistema</i>	48
1.4.3.2	<i>Participação do usuário</i>	54
1.4.3.3	<i>Breves considerações a respeito da tecnologia</i>	56
1.4.4	Os níveis de abstração e a modelagem do banco de dados	57
1.4.5	Atividade	59
	CONCLUSÃO	61
	RESUMO	62
2	UNIDADE 2: PLANEJAMENTO E DESENHO DE BANCOS DE DADOS POR MEIO DE MODELAGEM CONCEITUAL	65
2.1	OBJETIVO GERAL.....	65
2.2	OBJETIVOS ESPECÍFICOS.....	65
2.3	PRÉ-REQUISITOS	65
2.4	A MODELAGEM COMO UM MODO DE VER A REALIDADE.....	67
2.5	MODELO ENTIDADE RELACIONAMENTO – MER.....	67
2.5.1	Entidades	69
2.5.2	Atributos	70
2.5.3	Relacionamentos	72
2.5.4	Formas de modelagem MER – Relacionamentos	73
2.6	DESENHO DE BANCOS DE DADOS BASEADO NO MODELO ENTIDADE RELACIONAMENTO (MER).....	79
2.6.1	Atividade	79
2.6.2	Estudo de caso 1: modelagem de uma situação de compra e venda	80
2.6.3	Atividade	81

2.6.4	Estudo de caso 2: ampliando o modelo de compra e venda com a inclusão dos fornecedores	82
2.6.5	Atividade	82
2.6.6	Atividade	83
	CONCLUSÃO	85
	RESUMO	86
3	UNIDADE 3: MANIPULAÇÃO DE BANCOS DE DADOS	89
3.1	OBJETIVO GERAL	89
3.2	OBJETIVOS ESPECÍFICOS	89
3.3	PRÉ-REQUISITOS	89
3.4	INTRODUÇÃO	91
3.5	INTRODUÇÃO À SQL	95
3.6	PRÁTICA DE DECLARAÇÕES SQL	96
3.6.1	Criar o banco de dados	98
3.6.2	Criar as tabelas necessárias para implementar o modelo	100
3.6.3	Criar <i>constraints</i> de integridade: chaves primárias e estrangeiras	104
3.6.4	Inserir dados de teste	105
3.6.4.1	<i>Dados para a tabela Autores</i>	105
3.6.4.2	<i>Dados para a tabela Usuarios</i>	107
3.6.4.3	<i>Dados para a tabela Livros</i>	108
3.6.4.4	<i>Dados para a tabela livro_autores</i>	109
3.6.4.5	<i>Dados para a tabela emprestimos</i>	112
3.6.5	Atualizar dados	114
3.6.6	Praticar recuperação de dados e pesquisas no banco de dados	115
3.6.6.1	<i>Ver informações dos livros cadastrados</i>	115
3.6.6.2	<i>Ver informações dos livros cadastrados, ordenados pelo título do livro</i> ..	118
3.6.6.3	<i>Ver informações dos livros cadastrados, ordenados pelo título do livro em ordem descendente</i>	118
3.6.6.4	<i>Determinar quantos livros estão cadastrados</i>	119
3.6.6.5	<i>Determinar quantos livros estão cadastrados, fornecendo o nome de Quantidade para a coluna que conterà a resposta</i>	120
3.6.6.6	<i>Listar livros cujos títulos comecem com a palavra Modelagem</i>	120
3.6.6.7	<i>Uso da cláusula WHERE para unir dados de tabelas diferentes</i>	122
3.6.6.7.1	Atividade	123
3.6.6.8	<i>Aninhamento de consultas</i>	124
3.6.6.9	<i>Trabalhar formatação de datas no comando select</i>	125
3.6.7	Campos de numeração automática	127
3.6.8	Apagar dados e destruir objetos	129
3.6.8.1	<i>Apagar dados da tabela Componentes</i>	130
3.6.8.2	<i>Apagar um objeto do banco de dados</i>	131
3.6.8.3	Atividade	132

	CONCLUSÃO	136
	RESUMO	136
	SUGESTÃO DE LEITURA	138
	REFERÊNCIAS	138
	APÊNDICE A	141
	APRESENTAÇÃO DA FERRAMENTA BR MODELO PARA MODELAGEM ENTIDADE <i>VERSUS</i> RELACIONAMENTO	141
	APÊNDICE B	151
	APRESENTAÇÃO DA FERRAMENTA MYSQL WORKSHOP PARA TRABALHO COM SQL E BANCOS DE DADOS RELACIONAIS MYSQL	151
2.1	CRIAR CONEXÃO NO <i>MYSQL WORKSHOP</i>	154
2.2	TRABALHAR NO <i>MYSQL WORKBENCH</i>	155
2.3	CRIAR DIAGRAMAS NO <i>MYSQL WORKBENCH</i>	160

APRESENTAÇÃO

A disciplina *Planejamento e elaboração de bases de dados* faz parte do eixo de Tecnologias da Informação e da Comunicação (TIC) na formação do bibliotecário. Guarda relação com as disciplinas do referido eixo, em especial *Introdução às Tecnologias da Informação e da Comunicação*.

Oferece conhecimentos que visam a capacitar tecnicamente o bibliotecário para participar de projetos de bancos de dados, atentando para questões humanísticas, porém, com maior concentração na técnica.

Além de atender à necessidade de formação básica do conhecimento necessário ao planejamento de bancos de dados nas organizações, esta disciplina pode também atender às necessidades específicas na execução da disciplina futura de *Desenvolvimento do Trabalho de Conclusão de Curso* (TCC), além de possibilitar a criação de um pequeno serviço de informações baseado em banco de dados e de bibliotecas digitais.

Em alguns momentos, poderá ser necessário o conhecimento prévio de conceitos e técnicas abordados em outras disciplinas, cabendo revisão destas, se necessário, em especial: *Gestão da Informação e do Conhecimento*; *Políticas de Informação*; *Instrumentos de Representação Descritiva da Informação*; *Organização, Sistemas e Métodos Aplicados a Ambientes de Informação*; e, *Ambientes, Serviços e Sistemas Informacionais*.

Quanto ao conteúdo, este foi dividido em três unidades e aborda conceitos de bancos de dados, focando no modelo relacional.

A primeira delas apresenta alguns conceitos importantes e abordagens relativas ao planejamento e ao projeto de bancos de dados, discutindo implicações de sua utilização em ambiente de rede e levantamento de requisitos.

A segunda unidade discute os aspectos de modelagem de bancos de dados, focando o Modelo Entidade Relacionamento (MER) e algumas formas de levantamento de requisitos. Também apresenta e discute dois casos de modelagem, para que o aluno possa praticá-la.

A parte final destina-se a uma introdução à linguagem *Structured Query Language* (SQL) e à modelagem e à implementação de bancos de dados. Por meio de um exemplo, todos os comandos são apresentados passo a passo para maior clareza e facilidade de estudo a distância.

Os exemplos adotados, contrariamente à prática usual e a exemplos em outras obras do autor, não são os finais, os perfeitos. Propositamente, os modelos são imperfeitos, tal qual geralmente ocorre no primeiro pensamento do projetista. Dessa forma, podem ser sentidos e discutidos os problemas advindos de uma modelagem superficial e as dificuldades em melhorá-la posteriormente ao início da implementação (ou, até, quando já em produção).

No decorrer da disciplina, há exemplos e atividades de modelagem (Unidade 2) e implementação de bancos de dados (Unidade 3). É importante que os exemplos sejam reproduzidos e as atividades realizadas, para facilitar sua aprendizagem. Os conteúdos da Unidade 2 podem ser realizados com lápis e papel, embora, preferencialmente, devam ser realizados com apoio de algum *software*. Para o caso da Unidade 3, o uso de um *software* é essencial.

Ao final, dois apêndices apresentam, de forma rápida, porém suficiente para o propósito desta disciplina, duas ferramentas úteis: uma para modelagem conceitual, outra para implementação e uso de um banco de dados.

Após a leitura, o aluno possuirá conhecimentos para elaboração de modelos de dados e sua implementação em um banco de dados relacionais, servindo a aplicações desde pequena até média complexidade, em ambientes controlados pelo próprio aluno ou disponibilizados em pequenas redes.

Deu-se preferência a ferramentas ditas de **software livre** para a execução dos exemplos. Boa leitura, boas práticas e boas soluções. **Sucesso!**

UNIDADE 1

CONCEITOS RELATIVOS A BANCOS DE DADOS

1.1 OBJETIVO GERAL

Apresentar os conceitos relacionados a bancos de dados e ao seu projeto.

1.2 OBJETIVOS ESPECÍFICOS

Espera-se que, ao final desta unidade, você seja capaz de:

- a) conceituar dado, metadado, tipos de dados, informação e conhecimento no contexto de bases de dados;
- b) compreender a organização de bancos de dados, em especial do modelo relacional;
- c) compreender os diferentes níveis de abstração e visão no projeto de bancos de dados;
- d) reconhecer a importância de uma boa modelagem de dados com participação ativa dos usuários.

1.3 PRÉ-REQUISITOS

Noções de tecnologia da informação e de conceitos de dado, informação e seu ciclo.

1.4 A IMPORTÂNCIA DA ESTRUTURAÇÃO DAS BASES DE DADOS E A BIBLIOTECONOMIA

Você já passou pela situação de guardar um documento em qualquer lugar e, quando precisou dele, não sabia onde estava? Pode ser a certidão de nascimento, o título de eleitor, um certificado de conclusão de curso ou uma conta de luz, por exemplo. Talvez já tenha vivenciado o contrário, ou seja, de guardar tão bem um documento que depois não sabia onde o havia guardado.

É muito comum isso acontecer quando não temos um sistema de gestão de nossos dados organizados de modo que facilite a nossa consulta e a busca pela informação desejada, quer seja em pastas, quer seja em um fichário.

Assim como a organização pessoal facilita o armazenamento e a localização de um dado documento quando precisamos, o mesmo pensamento se aplica às informações armazenadas em bancos de dados.

Dentre os problemas típicos que temos com a informação e sua gestão, encontram-se o “como” selecionar e o “onde” conseguir a informação desejada.

Uma das formas adotadas para auxiliar na solução desse problema é o armazenamento sistemático das informações ou dos dados que nos propiciam obtê-las.

Vamos ver um exemplo?

Podemos destacar a área da saúde, que:

- a) já em 1867, *John Billings*, por meio da *National Library of Medicine* (NLM), inicia o *Index Medicus*;
- b) em 1958, a NLM introduz o computador;
- c) em 1989, passa a produzir o *MEDLINE CD-ROM*, que cresce a cada ano com a inclusão de centenas de milhares de artigos de milhares de revistas indexadas;
- d) atualmente, possui várias bases de dados em saúde, as quais, tem por finalidade concentrar as informações importantes e oferecer um mecanismo de busca e recuperação dos conteúdos desejados. Exemplos: BIREME, Lilacs, PubMed.

De forma geral, todas as áreas do conhecimento possuem essa mesma necessidade sentida pela área da saúde, dando origem às **bases eletrônicas de dados**, nas quais, obviamente, o computador é uma poderosa ferramenta de auxílio.

Agora, reflita sobre a seguinte questão: como você vê a importância das bases de dados em sua área de atuação?

Base de dados

O termo “banco de dados” é normalmente utilizado para categorizar um conjunto de dados específicos ou relacionados, enquanto o termo “bases de dados” pode ser utilizado para o conjunto de bancos de dados. Por exemplo, podemos ter uma base de dados em saúde contendo os bancos de dados de farmacologia e anatomia.



Para a Biblioteconomia, as bases de dados são fundamentais, pois facilitam o “como fazer” a busca das informações e localizá-las! Por isso, é importante conhecermos o histórico de sua evolução, como chegamos aos modelos que usamos atualmente, e também os principais conceitos a elas relacionados.

Então, mãos à obra e bons estudos!

1.4.1 Das aplicações de bancos de dados ao sistema gerenciador de bancos de dados

Em uma visão histórica, nos primórdios da computação, as aplicações de bancos de dados eram implementadas em sistemas de arquivos. Isso ocorreu, principalmente, entre os anos 1950 a 1970 – embora o processo ainda possa ser utilizado hoje em dia.

Usar sistemas de arquivos corresponde a escrever programas de acesso aos dados para cada nova operação necessária (busca, inclusão, alteração etc.). Modernamente, os sistemas gerenciadores de bancos de dados, que veremos nesta disciplina, são os responsáveis pela realização dessas operações, por meio de comandos SQL, que veremos na sequência.

Um exemplo hipotético de arquivo pode ser visto na imagem (Quadro 1) a seguir. Nele, temos números de *International Standard Book Number* (ISBN), nomes de obras e de autores, separados pelo caracter | (que não pode fazer parte dos nomes de obras nem de seus autores, por ser um separador). Note que é necessário saber de antemão a sequência e a posição dos conteúdos para localizá-los, o que exige a elaboração de um programa, como comentado.

Quadro 1 – Exemplo de aplicação de banco de dados baseada em sistema de arquivos

ISBN	nomes de obras	nomes de autores
...		
8508028121	Teoria da informação	<i>Isaac Epstein</i>
007460575	Modelagem de dados	<i>Peter Chen</i>
978857605923	Sistemas de informação gerenciais	<i>Jane P. Laudon-Kenneth C. Laudon</i>
9788586846588	<i>Microsoft SQL Server</i>	<i>José Simão de Paula Pinto</i>
...		

Fonte: Produção do próprio autor

O quadro poderia também ser escrito do seguinte modo:

...|8508028121|Teoria da informação|Isaac Epstein|007460575|
Modelagem de dados|Peter Chen|9788576059233|Sistemas de informação gerenciais|Jane P. Laudon-Kenneth C. Laudon|9788586846588|
Microsoft SQL Server|José Simão de Paula Pinto|...

Observe que foi utilizada uma separação (com -) para os nomes dos autores de um dos livros (*Sistemas de informação*, do casal *Laudon e Laudon*); isso tem que ser previsto na modelagem e impede que o caractere escolhido para separação (-) seja utilizado nos nomes.

Tal técnica funcionou e foi utilizada durante muito tempo, mas trazia as seguintes desvantagens:

- a) redundância de dados e **inconsistência**;
- b) **múltiplos formatos de arquivos**;
- c) duplicação da informação em arquivos diferentes;
- d) dificuldades no acesso aos dados;
- e) necessidade de reescrever um novo programa para atender cada nova tarefa;
- f) **isolação dos dados**, devido aos múltiplos arquivos e formatos;
- g) **problemas de integridade** e *constraints* (ou regras) de integridade fazendo parte dos códigos dos programas, trazendo dificuldades para adicionar novas regras ou modificar as já existentes.

Se você já trabalhou em ambientes assim, deve ter sentido a necessidade de possuir um mecanismo melhor para tratar desses problemas. Como exemplo, podemos citar os sistemas antigos de gestão, incluindo as bibliotecas, que eram muitas vezes baseados em conjuntos de arquivos, os quais tinham de ser mantidos e reindexados frequentemente e dependiam de alterações na programação para a inclusão de pequenas novas funcionalidades.

Uma solução é o Sistema Gerenciador de Bancos de Dados (SGDB), que permite a criação de bancos de dados diversos e seu gerenciamento, cuidando da localização física dos dados, da segurança do acesso e da integridade do conteúdo (PINTO, 2000, p. 6).

Nesta disciplina, estudaremos os SGBDs do tipo relacionais, que têm seu funcionamento baseado no modelo proposto por *Cood* no início da década de 1970 e teve seu primeiro modelo comercial (o *Oracle*) comercializado a partir do final da mesma década. Veremos esse modelo mais adiante.

Agora que você já teve uma breve introdução sobre o sistema baseado em arquivos e o SGBD, que tal fazermos uma comparação entre eles para melhor observarmos as suas formas de funcionamento?

Um comparativo entre um sistema baseado em arquivos e gerenciador de bancos de dados pode ser visto na imagem (Figura 1) que segue.

Inconsistência – ocorre quando os dados cadastrados possuem valores diferentes de acordo com a fonte de leitura; por exemplo, numa mesma biblioteca existem dois cadastros de usuários, no qual um tem endereço de correspondência atualizado e outro não.

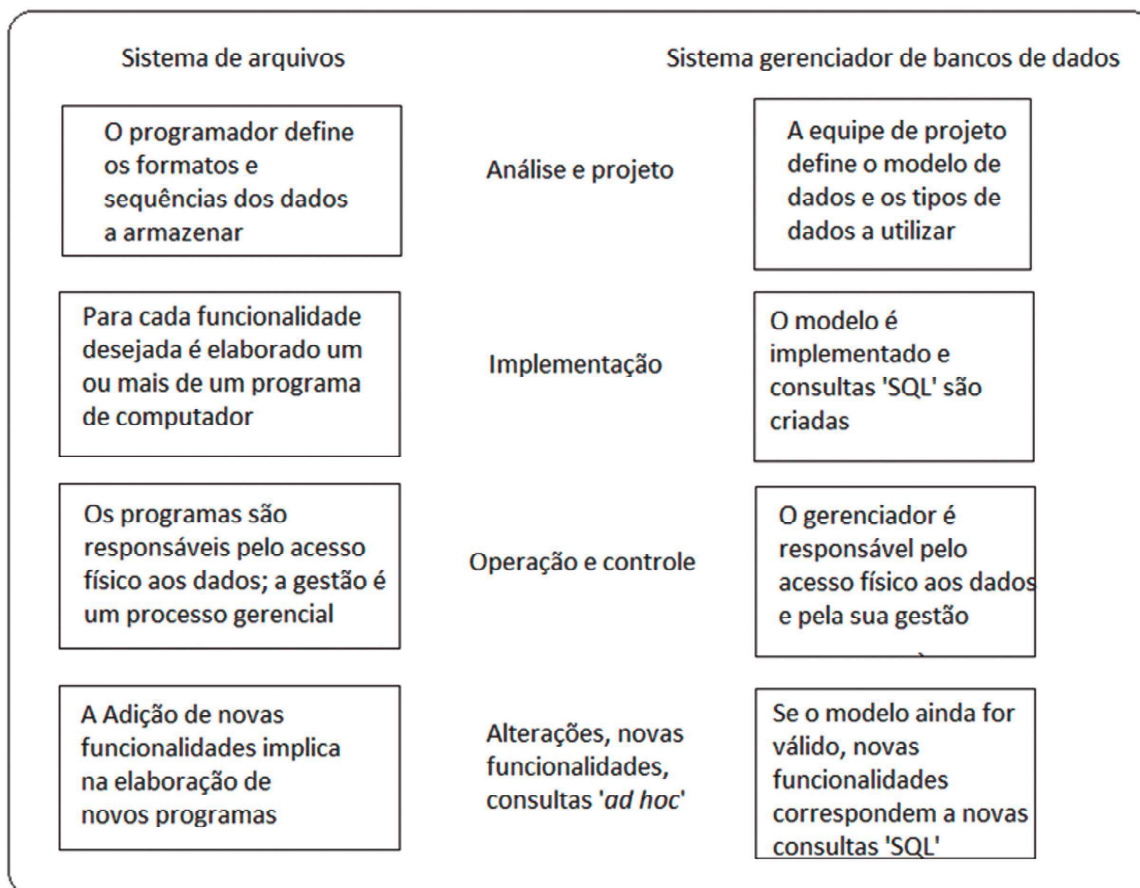
Múltiplos formatos de arquivos – ocorre quando diferentes programadores utilizam diferentes sequências de dados ou formas de compactação para cadastrar os dados. Por exemplo, se, no Quadro 1, um programador gravar na sequência do nome da obra o nome do autor, quando forem recuperadas as informações, elas estarão incorretas.

Isolação dos dados – a alteração de um não pode forçar a alteração de outro. Por exemplo: trocar em um arquivo uma sequência numérica para cadastro de ISBNs por uma alfanumérica (para permitir cadastrar o caractere **X**), não pode afetar outros arquivos.

Problemas de integridade – uma situação facilmente resolvida nos modernos gerenciadores por meio de chaves e que pode causar situações confusas quando trabalhamos com arquivos isolados. Imagine um cadastro de dependentes, no qual o responsável não foi cadastrado ou foi removido um cadastro e seus dependentes não. Modernamente, as *constraints* impedem que esse erro ocorra, desde que, obviamente, tenham sido bem projetadas e implementadas.



Figura 1 - Esquema comparativo entre o funcionamento do sistema de arquivos e o sistema de gerenciamento de banco de dados



Fonte: Produção do próprio autor

Devemos reforçar uma grande vantagem a favor dos gerenciadores: a possibilidade de realização das consultas *ad hoc*, ou seja, aquelas que são realizadas, em certo momento, para atender a uma necessidade específica e talvez nem sejam mais úteis posteriormente, sem a elaboração de um novo programa.

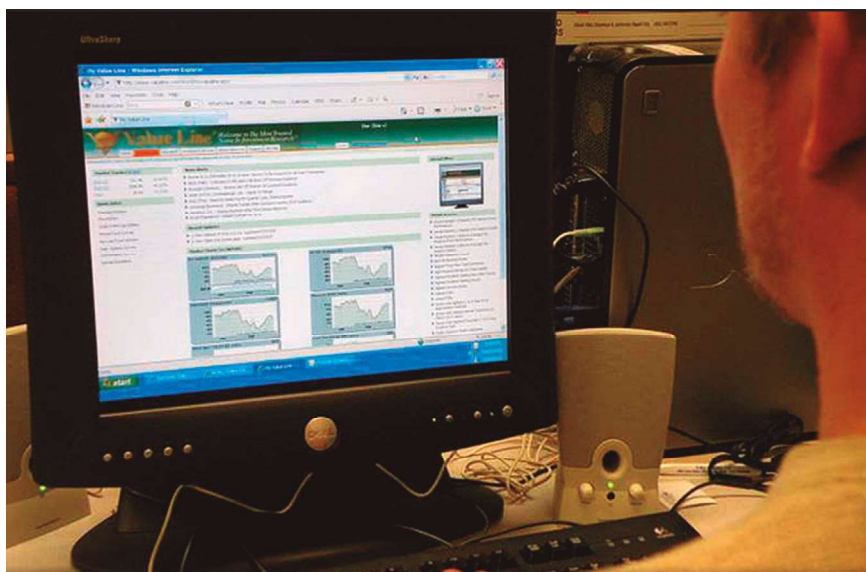
Por exemplo: imagine que foi solicitado a você um levantamento de quantos livros de certo autor foram emprestados pela manhã e devolvidos à tarde.

Evidentemente, nesse exemplo, presumimos que o modelo utilizado deve ter previsto armazenar os horários de empréstimo e devolução para que isso possa ser executado.

Evidencia-se, assim, a necessidade de um bom modelo.

Cabe aos SGBDs, ainda, cuidar da consistência ou atomicidade das atualizações (os termos serão explicados adiante). Por exemplo, cuidar para que em uma transferência de fundos ocorra, conjuntamente, um débito em uma conta e um crédito em outra. Gerenciar o acesso concorrente (simultâneo) aos recursos do banco de dados e seu desempenho são exemplos de outras tarefas importantes executadas pelos SGBDs.

Figura 2 - Os SGBDs permitem que façamos compras e investimentos on-line, por exemplo, de modo que as transações possam ocorrer simultaneamente e com confiabilidade entre o comprador e o vendedor ou investidor e bolsa de valores



Fonte: Flickr¹

Podemos compreender o SGBD como sendo um programa que, a partir de **scripts**, permite a criação e o gerenciamento de bancos de dados. Posteriormente, por meio de linguagens de acesso, permite a manipulação dos dados e objetos armazenados.

Um exemplo disso, em linguagem de consulta estruturada, SQL (a qual estudaremos na Unidade 3), poderia ser:

```
SELECT titulo FROM obras WHERE ISBN = 8508028121;
```

No exemplo, estamos solicitando o título do livro, cujo ISBN é 8508028121, e que está cadastrado na tabela obras. O ponto e vírgula é o terminador desse comando, e é obrigatório. O gerenciador irá pesquisar em suas tabelas e fornecer o resultado (o título é Teoria da Informação, para o exemplo dado). Note que, embora não seja necessária a elaboração de um programa, será necessária a elaboração da consulta em SQL; mas, isso é bem mais fácil e, ainda, existem muitos gerenciadores que oferecem formas gráficas de elaboração de consultas, gerando o SQL automaticamente.

Para fins de comparação, volte ao Quadro 1 e imagine como o nome do autor poderia ser obtido a partir de um arquivo como o da Figura 1.

Uma outra característica importante dos SGBDs é que a complexidade das operações e manipulações de dados e arquivos fica escondida do usuário, ou seja, não precisamos manipular *bits* para criar bancos de dados.

É interessante que você compreenda que, se o banco de dados possui uma interface gráfica, como o bastante difundido Microsoft Access, aquela interface permite uma interação com o usuário, facilitada; mas, internamente, aquilo que você selecionar na interface será executado por meio de um script, ou seja, as tarefas ficarão escondidas, e a complexidade de operação será menor. E, para alguns casos, a flexibilidade também.

Script é um conjunto de instruções que são executadas sequencialmente para atingir certo objetivo, por exemplo, localizar todos os livros que pertencem a uma coleção, em um banco de dados de uma biblioteca.



¹ FLICKR. Rapid City Public Library. Disponível em: <https://www.flickr.com/photos/rcpl/4385363391/in/photostream/>. Acesso em: 21 dez. 2018.

Nesta disciplina, estudaremos como criar e utilizar bancos de dados usando como exemplo prático um SGBD com arquitetura relacional, também conhecido como SGBD-R (em inglês, *Relational Database Management System* – RDBMS), iniciando com a explicação dos conceitos, a modelagem conceitual e evoluindo para a prática de bancos de dados.

Embora os exemplos sejam fornecidos em um SGBD-R específico, você poderá utilizar qualquer outro que seja capaz de executar as atividades propostas.

Para que possamos criar nossas próprias bases de dados, estudaremos, nesta unidade, conceitos e técnicas de planejamento e elaboração das mesmas.

Para a finalidade desta disciplina, e dada a ampla difusão dos bancos de dados relacionais, ao nos referirmos a SGBD, estaremos falando de SGBD-R, ok?

Então, vamos em frente!

1.4.2 Conceitos

Para início de nossos estudos, vamos resgatar e introduzir alguns conceitos para que possamos trabalhar com eles do ponto de vista de bancos de dados.

1.4.2.1 Dados, metadados, informação e conhecimento

Para garantir a qualidade dos dados, precisamos resgatar alguns conceitos já vistos em disciplinas precedentes e nos atermos à sua importância no contexto dos bancos de dados.

Conforme você já viu em outras disciplinas do curso, **dado** é o conteúdo mais básico, também conhecido como informação sem processamento. Em um banco de dados, é o elemento primário, aquilo que queremos armazenar para consulta ou transformação futura.



Explicativo

O ruído em um banco de dados

Ressaltamos aqui que, quando pesquisamos dados em uma organização, sempre haverá um subtipo de dado classificado como **ruído**, que corresponde aos dados incompletos, desatualizados, inexistentes ou aos desnecessários no contexto de interesse. Por exemplo, em uma biblioteca, cadastrar um livro com ano incorreto ou com título incompleto, esse dado pode ser considerado um ruído.

Meta é um prefixo que, na maioria das aplicações tecnológicas, significa uma definição ou descrição de algo. Então, **metadados** são a descrição ou a definição dos dados.

Em uma biblioteca, poderíamos ter, como exemplo de metadados, o ISBN de uma obra, seu título e seu autor. Os dados correspondentes seriam a combinação de valores para cada caso, desta forma:

Quadro 2 - Exemplo de diferenciação entre dado e metadado usando o contexto de informações de obras em uma biblioteca

Metadado	Dado
ISBN	8508028121
Título	Teoria da informação
Autor	<i>Isaac Epstein</i>

Fonte: Produção do próprio autor



Multimídia

Metadados úteis a bibliotecas e coleções

Para saber mais a respeito de metadados úteis a bibliotecas e coleções, pesquise na internet sobre o Padrão *Dublin Core* e o formato *Machine Readable Cataloging* (MARC).

Já a **metalinguagem** é uma definição ou descrição de uma linguagem.

Em seu sentido geral, refere-se a tornar mais compreensivo ou fundamental um conjunto de regras. Um exemplo, seria tomarmos o conjunto de regras formado por uma gramática e um dicionário, que poderiam ser utilizados para definir e descrever as formas de falar e escrever um determinado idioma. Isso se refere à metalinguagem, ou seja, a uma linguagem utilizada para explicar outra.

A linguagem tipicamente utilizada para descrever as estruturas de um banco de dados relacional (e, posteriormente, realizar as consultas ao seu conteúdo) é a linguagem de consulta estruturada (SQL): ela é utilizada para traduzir os conceitos do mundo real para a **linguagem** falada pelo gerenciador de bancos de dados.

Em bancos de dados, **metadado** refere-se à descrição do dado: descreve a estrutura e o conteúdo do dado, assim como seu **formato** em termos de **tipo de dado**.

Vamos ver um exemplo?



Poderíamos ter em uma biblioteca a descrição do nome do autor de uma obra com a especificação para que sejam aceitos até 40 caracteres alfanuméricos, de tal forma que, em bancos de dados, seria:

NOME_AUTOR CHAR(40);

Assim, temos NOME_AUTOR como o metadado que identifica o nome do autor da obra; CHAR, o representativo para caracteres alfanuméricos; e, 40, o tamanho máximo, ou a quantidade de caracteres.

Em muitos casos, também estará associado ao conjunto de metadados, o **domínio de valores**, ou seja, a faixa de valores que o dado pode assumir, ou os valores possíveis, caso sejam enumeráveis, e, ainda, a sua **unidade**. Veja, a seguir, dois exemplos de domínio de valores:

- a) a temperatura com valor numérico real, com uma casa decimal, variando de – 10 a + 50 graus centígrados;
- b) os valores das siglas possíveis para estados da região sul, necessariamente a escolher somente um do conjunto {PR, SC, RS}.

Nos exemplos acima, os domínios para temperatura são os valores numéricos entre – 10 e + 50 e, para as siglas da região sul, o conjunto dado (PR, SC, RS); assim, quaisquer outros valores não fazem parte do domínio e não podem ser aceitos.

Em uma biblioteca, poderíamos ter um metadado que descreve o ISBN, por exemplo, aceitando 11 números, somente inteiros. Poderia haver uma regra que determinaria que não pode ser aceito o número zero.

E onde essas informações relativas a metadados, formatos e tipos de dados ficam armazenadas?

Toda a **informação** para gerenciamento do banco de dados, armazenada no chamado **catálogo**, é um conteúdo referente aos metadados de interesse do SGBD (ELMASRI; NAVATHE, 2011, p. 26). O catálogo em questão é uma tabela ou um conjunto delas, mantida pelo sistema gerenciador, que possui as informações dos metadados, dados e arquivos físicos que compõem o banco de dados.

Foi mencionado anteriormente que os dados podem ser classificados em vários tipos diferentes. Vamos conhecê-los?

O computador armazena tudo em forma binária (*bits* 0 e 1). Então, para distinguir como as sequências de *bits* serão interpretadas, devemos informar qual **tipo de dado** queremos naquela posição, por exemplo, um número inteiro, um número decimal ou caracteres alfanuméricos.

Veja, a seguir, alguns tipos de dados usuais (os valores mínimos, máximos e limites em geral devem ser consultados para cada produto, já que variam de um SGBD para outro):

Quadro 3 - Tipos de dados usuais em bancos de dados relacionais

TIPOS DE DADOS USUAIS EM BANCOS DE DADOS RELACIONAIS	
Tipo	Descrição
CHAR(tamanho)	<p>Para armazenamento de caracteres em geral (letras, números, símbolos). O tamanho (número inteiro) refere-se à quantidade de caracteres que será utilizada, e é o máximo, ou seja, o dado poderá ter um tamanho menor. O tamanho declarado será sempre ocupado no banco de dados como espaço de armazenamento, ou seja, se declararmos CHAR(100), estaremos reservando 100 posições de armazenamento, quer existam ou não dados a armazenar.</p> <p>Exemplo: nome CHAR(40) - isto faria com que, para o cadastro dos dados referentes ao nome, pudessem ser utilizados até 40 caracteres alfanuméricos e sempre seriam utilizadas as 40 posições no armazenamento, mesmo que o nome fosse menor.</p>
VARCHAR(tamanho)	<p>Igual ao CHAR, porém somente serão armazenados os caracteres efetivamente ocupados, e não o tamanho total. Por exemplo, se declararmos VARCHAR(100) e o dado contiver somente 50 caracteres, o tamanho total não será utilizado. Esse armazenamento consome um pouco mais de processamento.</p> <p>Exemplo: nome VARCHAR(40) - isso faria com que, para o cadastro dos dados referentes ao nome, pudessem ser utilizados até 40 caracteres alfanuméricos, utilizando o espaço de armazenamento de acordo com o tamanho do conteúdo.</p>
INTEGER	<p>Para o armazenamento de números inteiros.</p> <p>Exemplo: código INTEGER - isto faria com que, para o cadastro dos dados referentes ao código, pudessem ser utilizados somente números inteiros (o limite dos números depende de cada gerenciador).</p>
DECIMAL	<p>Para o armazenamento de números não inteiros, que possuem a parte decimal. Podemos especificar o tamanho da parte inteira e o da parte decimal, por exemplo, DECIMAL (10,2), com 10 casas para a parte inteira e 2 casas decimais.</p> <p>Exemplo: valor DECIMAL (10,2) - isto faria com que, para o cadastro dos dados referentes ao valor, pudessem ser utilizados números com parte inteira de 10 dígitos e duas casas decimais.</p>
BLOB ou BINARY	<p>Para armazenamento de sequências de <i>bits</i>, que podem ser arquivos digitais, em geral para texto, imagens, sons, vídeos.</p> <p>Exemplo: imagem BLOB - isso permitiria cadastrar uma sequência longa tal como uma imagem.</p>

Fonte: Produção do próprio autor

Em síntese, o tipo de dado define **o que** podemos armazenar e, em alguns casos, o **como** será armazenado.

Com relação à **informação**: esse termo se refere aos dados após terem sido agregados a algum contexto, utilidade ou processamento. Por exemplo, podemos considerar um código de usuário em uma biblioteca como sendo um dado e o conjunto de códigos, ou seja, quantos são, como sendo a informação do número de usuários cadastrados na biblioteca.



Explicativo

A importância dos usuários e do conhecimento do negócio para a modelagem de um banco de dados

No projeto de bancos de dados, a modelagem conceitual inicia-se a partir da coleta das informações pertinentes àquilo que se está modelando; nesse caso, os códigos de usuário. Prosseguiria modelando-se informações que se quer obter a partir dos dados; para o exemplo, o número de usuários. Notar a importância do conhecimento do negócio que está sob análise.

Já o **conhecimento** refere-se àquilo que é obtido a partir da contextualização e da utilização dos dados e informações e, para fins de bancos de dados, àquilo que será agregado como sistema de informação em uma organização.

Note que modelar o total de usuários (ou o período do ano em que ocorrem mais empréstimos para poder, com isso, estimar atendentes, se fosse o caso) tem muito mais a ver com a contextualização do uso do sistema do que com o sistema em si: é necessário saber o que e como o **negócio**, nesse caso uma biblioteca, utilizará as informações para gestão. Essa característica refere-se ao conhecimento, e não está intrinsecamente ligada aos dados ou aos metadados (ou às informações que possam ser geradas a partir deles), mas ao seu uso.

Portanto, o banco de dados pode ser uma fonte do conhecimento empresarial, enquanto o sistema de informações que o utiliza, ou que nele é baseado, corresponde à sistematização do conhecimento de uma organização. O uso que se fará desse conhecimento liga-se ao conceito de inteligência empresarial, ou de negócios, que transcende esta disciplina.

Figura 3 - Assim como apenas guardar as notas fiscais não vai ajudar uma pessoa a gerenciar suas finanças se ela não fizer uma análise delas, o mesmo acontece com os dados em um SGDB. Se não servirem de subsídios para o conhecimento, ficarão apenas armazenados para consulta

BIBLIOTECA DE PESQUISA
COMPROVANTE DE MOVIMENTAÇÃO
23/03/2016 15:20:35
=====

USUÁRIO: 0101 Francisco da Silva

EMPRÉSTIMO(S):
1) 2554 - História e vida integrada
* Data de devolução: 30/03/2016

RENOVAÇÕES:

DEVOLUÇÕES

ASS. FUNC. ASS. USUARIO
=====

MULTA POR DIA DE ATRASO R\$ 0,50 - CONSULTA LOCAL DEVOLVER ATÉ AS
14:00 DAS 8:00 E 20:00

Fonte: Produção do próprio autor



Atenção

Lembre-se: o banco de dados não é um depósito de dados a esmo! Não estamos querendo um arquivo morto sem planejamento, no qual jogaremos dados somente para guardá-los. Pelo contrário, desejamos projetar como e para que serão recuperados os dados, de forma a ter o melhor armazenamento possível que atenda a essas necessidades. Caso contrário, teremos um grande arquivo no qual, quando necessitamos de algo, não conseguimos encontrar...

1.4.2.2 Bancos de dados e bases de dados

O termo **banco de dados** é, normalmente, utilizado para categorizar um conjunto de dados específicos ou relacionados, enquanto o termo **bases de dados** pode ser utilizado para o conjunto de bancos de dados. Por exemplo, podemos ter, em uma base de dados em saúde, os bancos de dados de farmacologia e anatomia.

Refleta: que bancos de dados você utiliza em seus afazeres diários? Como eles são organizados, atualizados e disponibilizados?

Pensando um pouco, você verá que são vários os exemplos, que vão de sua agenda de telefones e de compromissos a dados bancários, incluindo-se o histórico escolar do curso de Biblioteconomia, na modalidade à distância, que você está cursando!

Figura 4 - Uma base de dados pode ser composta por vários bancos de dados. Por exemplo, a área de Educação pode conter dados de Filosofia e Psicologia. As bases de dados analógicas eram consultadas nas estantes das bibliotecas, atualmente elas são digitais



Fonte: Flickr²

Vamos ver um exemplo?

Você certamente já utilizou um caixa eletrônico para fazer transações bancárias. Já aconteceu de você acessar a sua conta em um caixa eletrônico de seu banco e, de repente, você perceber que a interface está toda diferente? Que já não sabe onde estão as opções que você sempre utilizava e você tem a impressão de que ficou mais difícil usá-lo? Essa situação de facilidade e uso intuitivo de uma interface de um sistema está relacionada à usabilidade.

Figura 5 - A usabilidade e a performance de um SGBD interferem na sua percepção de uso de um determinado sistema



Fonte: Flickr³

² FLICKR. Philippe Boivin. Disponível em: <https://www.flickr.com/photos/philippeboivin/4789456531/>. Acesso em: 21 dez. 2018.

³ FLICKR. Sascha Kohlmann. Disponível em: <https://www.flickr.com/photos/skohlmann/15540194507/>. Acesso em: 21 dez. 2018.

Uma situação envolvendo a performance poderia ser a de quando você solicita, nesse mesmo caixa eletrônico, a consulta ao seu extrato do mês anterior e ele, rapidamente, carrega as informações. Nesse caso, poderemos dizer que o sistema apresentou uma alta performance no processamento da sua solicitação.

O mesmo pode ocorrer em um sistema de uma biblioteca. Ele pode ser fácil e intuitivo de ser usado por alunos, professores e bibliotecários, cada um com sua visão do sistema ou ser extremamente complexo. Pode carregar rapidamente o resultado de buscas por livros e periódicos e dar baixas em materiais, assim como ser extremamente trabalhoso ou vargasoso. Evidentemente, a usabilidade e a performance dependerão de como o sistema foi concebido e desenvolvido.

Agora, vamos conhecer um pouco mais sobre a estruturação de um banco de dados.

Os SGBDs possuem mais de uma forma de organização interna. Ela define a forma de trabalhar com os SGBDs e também algumas de suas características, como a **performance** e a **usabilidade**.

Assim, ao longo dos anos de desenvolvimento e pesquisas na área, surgiram modelos de organização interna, como:

- a) organização hierárquica dos componentes internos, como *Adaptable Data Base System* (Adabas) e *IP Multimedia Subsystem* (IMS) (e, quando em arquivos, *Extensible Markup Language* (XML));
- b) organização em rede, como o *Integrated Database Management System* (CAIDMS), baseado no modelo da *Conference on Data Systems Languages* (CODASYL);
- c) orientados a relações, muito difundidos no mercado hoje, como o *My-SQL*, que usaremos nesta disciplina, o *SQL Server* e o *MS-Access*;
- d) orientados a objetos, como *Parallel Object-Oriented Environment and Toolkit* (POET) e *ObjectDB*;
- e) aqueles que misturam essas duas últimas possibilidades, os objeto-relacionais, como *DB-2*, *PostgreSQL* e versões atuais do *Oracle*.


Os modelos orientados a objetos visam a atender, em especial, às linguagens de programação orientadas a objetos, porém, não possuem uma álgebra universal (capaz de descrever o SGBD lógica ou matematicamente, como a álgebra relacional dos SGBD-R), o que dá origem a diversas implementações diferentes entre si. Não são grandemente difundidos no mercado e muitas de suas aplicações são experimentais.

Já os modelos orientados a objeto-relacionais procuram associar o moderno mundo da orientação a objetos com os tradicionais e largamente utilizados SGBDs relacionais, ou SGBD-R, oferecendo uma interface de objetos para as tabelas do banco de dados. Dessa forma, compreender os bancos de dados relacionais é útil também para essas aplicações. Modelos relacionais baseiam-se em relações; você verá mais sobre eles mais adiante na disciplina.


Ao longo dos anos, as práticas comerciais voltaram-se à grande expansão de modelos hierárquicos nos ambientes de **mainframe**, seguidos por enorme expansão do uso de bancos de dados relacionais. Um dos aspectos que apoiou o crescimento dos SGBD-R é a existência de uma

Performance – ou desempenho, relaciona-se, em bancos de dados, com a capacidade de atender os vários usuários simultâneos rapidamente.

Usabilidade – nesse contexto, tem a ver com a facilidade de implementação e/ou uso do sistema. Também pode referir-se ao ambiente necessário ao seu pleno funcionamento (funciona na *web*? E em equipamentos móveis?), ou, ainda, mas não menos importante, ao uso do sistema por pessoas com problemas de visão/de cognição/de audição, quando for o caso de sistemas com resposta audível.



Mainframe – termo utilizado na informática para referir-se a computadores de grande porte.



Álgebra relacional – simbolismo utilizado para representar as operações matemáticas que podem ser realizadas com conjuntos de relações (em nosso caso, tabelas). É a base para o funcionamento dos SGBD-R.

Arquivos binários – são os arquivos que armazenam os dados na forma de sequências de zeros e uns (0, 1).

álgebra relacional que fornece uma base sólida a seu funcionamento e uso: os conjuntos de relações e suas manipulações seguem uma estrutura formal bem definida, o que possibilita garantia de integridade ao usuário do sistema.

Do ponto de vista de *software*, é comum a referência a nomes como *MySQL*, *Oracle*, *SQL Server*, como sendo bancos de dados. Tecnicamente, são SGBDs. Como implementam o modelo relacional, podem ser referenciados como sendo SGBD-R. Esses sistemas permitem construir, catalogar e manter grandes coleções de dados, sob a forma de **arquivos binários** armazenados.

Neste curso, utilizaremos os bancos de dados relacionais. Devemos citar, porém, que os arquivos XML, de utilização cada vez maior nos aplicativos atuais, em especial, naqueles voltados a dispositivos móveis, são baseados em modelos hierárquicos e muito se assemelham aos bancos de dados hierárquicos.

Veja, a seguir, um exemplo de conteúdo de arquivo XML:

```
<?xml version = "1.0">
  <obras>
    <isbn>"8508028121"</isbn><titulo>"Teoria da
informação"</titulo><autor>"Isaac Epstein"</autor>
    <isbn>"0074605755"</isbn><titulo>"Modelagem de
dados"</titulo><autor>"Peter Chen"</autor>
    <isbn>"9788576059233"</isbn><titulo>"Sistemas de
informação gerenciais"</titulo><autor>"Jane P. Laudon"
</autor><autor>"Kenneth C. Laudon"</autor>
    <isbn>"9788586846588"</isbn><titulo>"Microsoft SQL
Server"</titulo><autor>" José Simão de Paula Pinto "
</autor>
  </obras>
```

No exemplo, o conteúdo exibido fornece informações de um conjunto de obras (identificado pela marcação, ou *tag*, <obras>, ou seja, o metadado). As obras estão listadas entre os marcadores <obras> e </obras> (sendo que este segundo, precedido de /, indica o final do conteúdo). Cada obra é identificada por meio das marcações <isbn>, <titulo> e <autor> (e seus terminadores </isbn>, </titulo> e </autor>); o conteúdo entre aspas entre os terminadores, por exemplo **8508028121** (para o caso de <isbn> e </isbn>), corresponde aos dados. Esse tipo de arquivo tem sido bastante utilizado para troca de dados, integração de sistemas e em aplicações na *internet* (por exemplo, as notas fiscais eletrônicas ou NF-es).

Do ponto de vista de uso, não deve ser levado ao usuário o conhecimento de questões relativas à organização interna do banco de dados. O utilizador deve possuir apenas uma boa interface para interação com as funcionalidades de que necessita. Porém, podemos concluir com a visão de que bancos de dados orientados a objetos e objeto-relacionais são mais próximos das metodologias de desenvolvimento orientadas a objetos; o modelo relacional é o mais tradicional no mercado e utilizado igualmente em aplicações orientadas a objetos e estruturadas. O modelo hierárquico possui alta performance, porém, exige programação específica para cada nova rota de acesso.



1.4.2.3 Atividade

Pesquise na *internet* dois SGBD-R pagos e dois gratuitos.

Resposta comentada

Exemplos de SGBD-R pagos, entre outros, são o *Microsoft SQL Server*, o *IBM DB2* e o *Oracle*; entre os gratuitos destacam-se o *MySQL* e o *PostgreSQL*.

1.4.2.4 Níveis de abstração em bancos de dados

Quando falamos do desenho de um banco de dados, ou de outros sistemas de computação, devemos ter em mente que existem vários níveis de pensamento, ou **abstração**, envolvidos: a tela de consultas que o usuário vai utilizar, os dados que serão armazenados e seus tipos (números, letras) e como será armazenado fisicamente o arquivo que contém o banco, por exemplo.

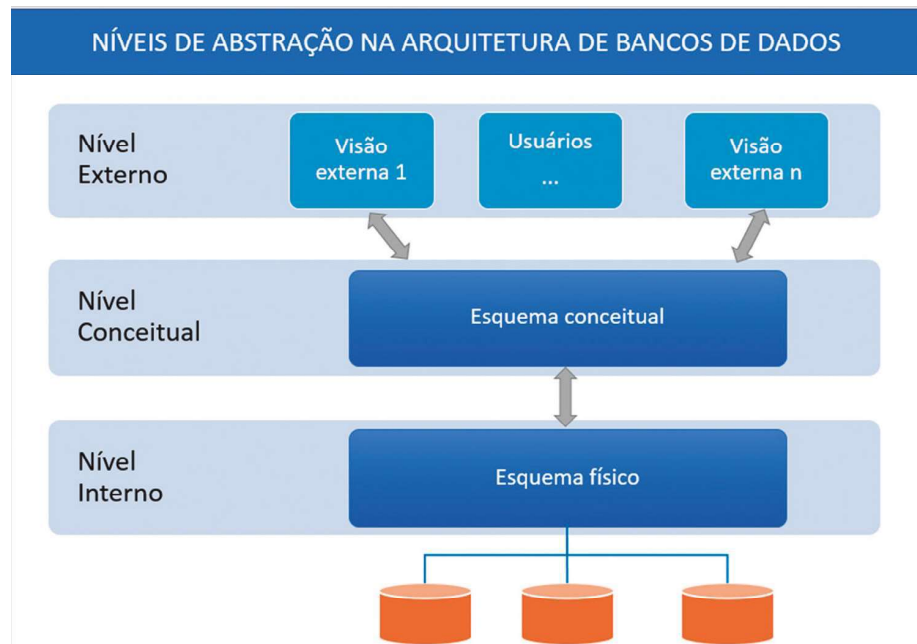
A **arquitetura** sugerida pelo *American National Standards Institute* (ANSI) e *Standard Planning and Requirements Comitee* (SPARC) prevê diferentes níveis de abstração. Veja de forma mais detalhada, a seguir:

Abstração – processo mental por intermédio do qual nos concentramos nos aspectos relevantes de um conjunto de objetos, desconsiderando suas diferenças.

Arquitetura – no contexto dos sistemas gerenciadores de bancos de dados, refere-se à forma de organização de suas estruturas internas e àquelas acessíveis aos usuários.



Figura 6 - Níveis de abstração na arquitetura de bancos de dados ANSI-SPARC



Fonte: Adaptado de Elmasri; Navathe (2004, p. 22)

Na representação dos níveis de abstração apresentada, notamos a existência de diferentes esquemas (ou modelos) em cada nível.

A imagem (Figura 6) é utilizada para dar uma visão geral da estrutura (ou arquitetura), sendo que as setas representam os fluxos de dados; os pequenos cilindros na base correspondem à forma usual de representação de conjuntos de dados.

Que tal vermos agora uma descrição de cada elemento?

Então, vamos lá:

- Nível físico ou interno** - descreve como um registro é armazenado (ex.: cliente). Neste nível, teremos as informações de localização do arquivo (ou arquivos) que compõe nosso banco de dados, seu tamanho etc. Um exemplo de nível físico é quando você guarda um arquivo no disco e precisa saber seu nome e local de armazenamento para acessá-lo: `c:\windows\planilha.xls`;
- Nível lógico ou conceitual** - descreve os dados armazenados em um banco de dados e relacionamentos entre os dados. Por exemplo, em uma linguagem de programação, poderíamos ter a definição de um tipo de dado para cliente declarado da seguinte maneira:

```
type cliente = record  
    nome : string;  
    rua : string;  
    numero : integer;  
end;
```

Na representação, temos a palavra *type* que representa um novo tipo de registro (*record*) de dados, e essa definição termina com a palavra *end*.

O ponto e vírgula é o terminador desse comando e é obrigatório. Os termos “nome”, “rua” e “numero” são os metadados, e as palavras *string* e *integer* representam os tipos de dados: a primeira indica que o conteúdo poderá ser formado por números e letras; a segunda, que somente são aceitos números inteiros;

- c) **Nível de visão ou externo** - programas de aplicação escondem detalhes dos tipos de dados e também informação, para propósitos de segurança e facilidade de utilização (ex.: por lei, a informação sobre o valor do salário de um funcionário não é visível para todos os usuários, mas apenas aqueles que precisam dessa informação; já o nome é uma informação que pode ficar visível, mas seu tipo de dado e codificação no banco não).

Devemos notar que uma das finalidades dessa arquitetura é oferecer níveis crescentes de abstração e facilidade de uso. As camadas mais baixas (no nível físico) devem trabalhar com a complexidade da manipulação dos *bits/bytes* dentro da máquina, algoritmos, programas; as mais externas (visão de usuários) somente utilizam os dados, sem se preocupar com como e onde estão armazenados.



Atenção

Nesta disciplina, vamos trabalhar, essencialmente, com o modelo (ou esquema) conceitual. Note que, ao longo deste texto, mostraremos os sinônimos dos conceitos e daremos preferência a um deles. A existência de vários termos semelhantes deve-se às diferentes nomenclaturas e interpretações de autores diversos, assim como às traduções. Não se preocupe, trabalharemos os mais usuais e, até o final desta disciplina, você conhecerá o necessário.

Por fim, os gerenciadores dos bancos de dados relacionais trabalham com relações, que seguem conceitos da teoria de conjuntos. Mas, na prática, as relações são chamadas de tabelas.

As tabelas são uma espécie de planilha, na qual as colunas correspondem aos atributos da relação, ou seja, o conteúdo, os dados. Na prática, os atributos, as colunas de nossa planilha exemplo, são chamadas de campos.

Na nomenclatura de bancos de dados relacionais, as linhas de nossa hipotética planilha são chamadas de **tuplas**, e correspondem a um conjunto de todas as colunas que guardam relação entre si. São comumente chamadas de linhas.



Observe a imagem a seguir.

Figura 7 - Exemplo de tabela de banco de dados relacionais

METADADOS (ISBN, TITULO, AUTOR) - as colunas são os ATRIBUTOS

	A	B	C
1	ISBN	TITULO	AUTOR
2	8508028121	Teoria da informação	Isaac Epstein
3	0074605755	Modelagem de dados	Peter Chen
4	9788576059233	Sistemas de informação gerenciais	Jane P. Laudon
5	9788576059233	Sistemas de informação gerenciais	Kenneth C. Laudon
6	9788586846588	Microsoft SQL Server	José Simão de Paula Pinto

As linhas contêm os DADOS - em bancos de dados chamamos as linhas de tuplas

Tabela LIVROS

Fonte: Produção do próprio autor

Dessa forma, ao trabalharmos com bancos de dados relacionais trabalharemos com tabelas compostas de colunas, que possuem muitas linhas de dados cadastrados. A tabela e as linhas compõem as informações, enquanto os campos representam os metadados. Por exemplo, poderíamos ter uma tabela de livros, com os metadados (colunas) autor, título, editora e ano. Cada linha dessa tabela corresponde ao conjunto de dados que fornece informações a respeito de um único livro.

Note na imagem (Figura 7) que o conteúdo cadastrado nas linhas (ou tuplas) 2 a 6 corresponde aos dados, contidos na tabela (ou relação) LIVROS, cujos metadados estão descritos no cabeçalho das colunas (chamadas em bancos de dados de atributos ou campos) da linha 1. Note também que as linhas 4 e 5 correspondem a algo que chamamos de redundância: para podermos ter mais de um autor, no modelo de planilha utilizado (e seria semelhante no desenho de um arquivo), optou-se por repetir as informações de ISBN e título. Uma alternativa seria pensar em termos de primeiro autor, segundo autor etc., em colunas separadas; mas, essa abordagem traria o inconveniente de termos várias colunas em branco para o caso de obras de um único autor, consistindo em desperdício.

A modelagem deverá levar em conta esse tipo de situação (veremos isso na Unidade 2). A solução adequada é termos uma tabela com os títulos de livros e outra com os nomes de autores, ligando-os por meio de um atributo comum, nesse caso, o ISBN. A arte desse tipo de desenho desenvolve-se por meio da normalização, mas não será abordada nesta disciplina, que é destinada a estudos introdutórios, dentro de 30h. Por fim, fechando os conceitos vistos, na planilha, cada célula pode ter uma formatação quanto aos tipos de dados que aceitará. Em um banco de dados, a primeira linha, que possui os metadados, e os tipos de dados aceitos em cada célula, estariam contidos no catálogo, conforme já explicado.

1.4.2.5 Tarefas do sistema gerenciador de bancos de dados

Já aconteceu de você estar fazendo uma transação via *internet*, como uma compra on-line e, de repente, a *internet* cair ou acabar a luz no meio da compra?

O que aconteceu em seguida?

Você provavelmente ficou preocupado se a compra foi realmente realizada, se o valor foi corretamente debitado da sua conta ou se a compra não foi efetivada em virtude desse contratempo.

É justamente essa a tarefa de um gerenciador de banco de dados! Vamos conhecê-la melhor?

Figura 8 - Ao fazer uma compra via *internet*, você certamente estará interagindo com um banco de dados que estará gerenciando essa transação visando a assegurar a integridade lógica e física dos dados



Fonte: Flickr⁴

A garantia de integridade dos dados, lógica e física, é uma das responsabilidades do gerenciador de bancos de dados. A integridade lógica refere-se à modelagem correta e à manutenção dela por meio da implementação (por exemplo, as chaves); a física refere-se à integridade do conteúdo, dos dados. Considerando-se que, atualmente, os bancos de dados são um dos pilares das aplicações comerciais e, em diversas áreas do conhecimento, como a Biblioteconomia, essa característica torna-se primordial. Estima-se que mais de 90% das aplicações comerciais modernas utilizem bancos de dados.

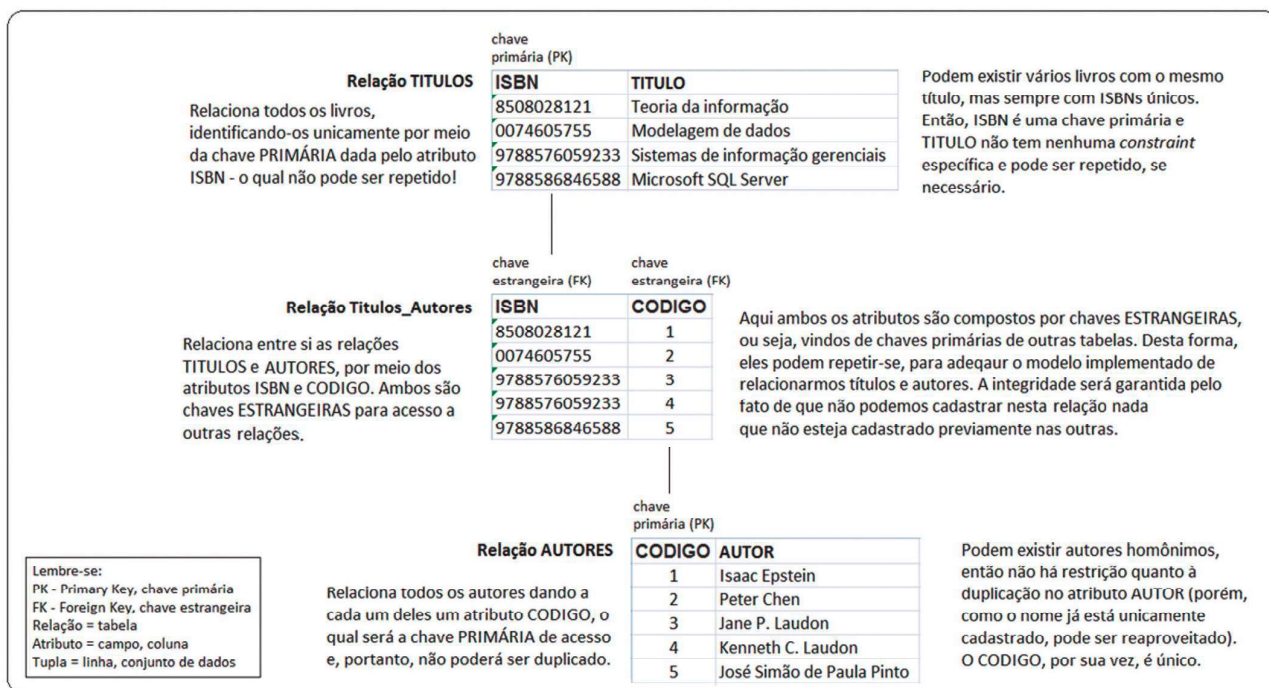
Nos bancos de dados relacionais, teremos a integridade garantida por meio de *constraints*, que são regras para o funcionamento do sistema. Entre elas, destacamos:

- a) campos que não podem ter valor nulo ou campos cujo preenchimento é obrigatório;
- b) limites de valores;
- c) campos de valor único (que não podem ser duplicados na mesma tabela, como o CPF), que podem ser campos identificadores ou chaves, primárias e estrangeiras (às vezes, chamadas de secundárias).

⁴ FLICKR. Barn Images. Disponível em: <https://www.flickr.com/photos/barnimages/21796692282/>. Acesso em: 21 dez. 2018.

Observe a imagem (Figura 9) que segue.

Figura 9 - Síntese dos conceitos apresentados



Fonte: Produção do próprio autor

A partir da imagem (Figura 9), podemos revisar alguns conceitos apresentados. A entidade (do mundo real) correspondente a um título de livro foi modelada a partir da relação (ou tabela) TITULOS, a qual possui os atributos ISBN (que é uma chave primária, *Primary Key* = PK) e TITULO.

Sempre que representarmos uma obra, ela será formada pelo conjunto (tupla) de seus dois atributos (correspondendo a uma entidade do mundo real).

Como o ISBN é uma chave, ele não poderá ser duplicado nem suprimido, nem poderá ser nulo (pois é um identificador, uma chave de acesso a uma certa tupla, um certo conjunto de dados, uma entidade).

O mesmo conceito ocorre com a relação correspondente aos AUTORES, sendo, naquele caso, a chave primária o atributo CODIGO.

Já a relação Titulos_Autores só será formada no momento em que tivermos os títulos e os autores cadastrados; seus atributos, ambos chaves estrangeiras (*Foreign Key* = FK), têm seus valores advindos de outras relações (TITULOS e AUTORES).

Dessa forma, é impossível cadastrar, por exemplo, um código de autor com valor zero, pois ele não foi cadastrado na relação de autores. Mas, por não ser uma chave primária, um mesmo valor pode ser repetido, utilizado mais de uma vez (note na figura o caso do livro *Sistemas de informação gerencial*, ISBN = 9788576059233, que tem mais de um autor).

Da mesma forma, temos uma garantia de integridade no conjunto de dados, pois não podemos, por exemplo, apagar um autor ou um título, caso ele esteja em uso na relação que faz sua junção (tal operação poderia disparar um gatilho – *trigger* – informando que a operação

não é possível em função da existência de valores em outra relação). Esse tipo de modelagem permite que se garanta a integridade e a não redundância da base.

Note que podemos cadastrar (inserir na relação ou tabela) os títulos e os autores de forma independente e em qualquer ordem, e descartá-los antes que sejam usados. Porém, no momento em que relacionarmos autores e livros, o sistema garantirá que sempre exista esse relacionamento.

Por fim, observe que sempre haverá uma dupla metadado/dado (por exemplo ISBN e 9788576059233); o valor do dado terá sua integridade também garantida pelo tipo atribuído (por exemplo, somente números inteiros, caracteres etc.), não sendo permitido o cadastramento de dados incorretos. Na Unidade 2, estudaremos um pouco mais a respeito, modelando entidades e seus relacionamentos.



Explicativo

Bancos de dados utilizam-se de *constraints*, que são restrições ou regras às quais o modelo é submetido para garantir sua integridade. Por exemplo, ao informarmos que certo campo não pode receber valores nulos (*not null*) ou que seus valores devem ser únicos (*unique*), estamos utilizando esse conceito.

Na modelagem de bancos de dados, com frequência aparecem duas restrições que são parte essencial do relacionamento entre as tabelas, as chaves primárias e estrangeiras.

Definimos a chave primária (*primary key*) como sendo o atributo ou conjunto de atributos que permitem identificar unicamente uma relação (tabela). Nesse caso, não podem existir para esse atributo (esta coluna) dois valores iguais em tuplas (linhas) diferentes.

A chave estrangeira (*foreign key*) é a chave primária de uma relação exportada para outra. Por se tratar de uma representação de relacionamento entre relações, considerando uma repetição de relacionamento, ocorrerá uma repetição do valor desse atributo em tuplas diferentes. Às vezes, a chave estrangeira é chamada de secundária, em oposição à primária.

Uma das atribuições da etapa de **desenho lógico** de um banco de dados é a criação de chaves primárias e secundárias de acesso aos dados. Essas chaves serão utilizadas não somente para auxílio na recuperação, mas também na garantia de integridade e na **indexação**. Por meio delas, o SGBD poderá manter as integridades: das entidades, de domínio, referencial e aquelas definidas pelo usuário. A imagem (Quadro 4), a seguir, resume alguns tipos de integridade garantidos pelas chaves (também conhecidas como *constraints* e/ou *keys*).

Desenho lógico – é o desenho que representa logicamente as relações existentes entre os dados que estão sendo modelados. Por exemplo, uma entidade Editora ligada com todos os seus Autores e suas Obras (veremos detalhadamente na Unidade 2).

Indexação – corresponde a colocar em ordem os conteúdos dos atributos definidos em uma relação. Em sistemas de bancos de dados, podemos ter vários índices diferentes para uma mesma tabela, facilitando a recuperação dos dados (embora com implicação no momento da inserção, pois os índices devem ser atualizados). Por exemplo, podemos, em uma tabela de livros, ter índices para ISBN por ordem crescente e decrescente, por títulos em ordem alfabética ou alfabética inversa.



Quadro 4 - Tipos de integridade de dados em SGBDs

TIPOS DE INTEGRIDADE DE DADOS EM SGBDs		
Tipo de integridade de dados	Garantida por meio de:	Descrição
Entidade (<i>Entity</i>)	Chave primária (<i>Primary key</i>) Identificador único (<i>Unique constraint</i>) Atributo identidade (<i>Identity attribute</i>)	Define cada tupla (linha) em uma relação (tabela) como sendo única. A integridade de entidades é garantida (<i>enforced</i>) pela criação de atributos (colunas) identificadores em uma relação (tabela) e garantindo que os valores desses atributos (colunas) serão únicos por meio de uma chave primária (<i>primary key</i>), um identificador único (<i>unique constraint</i>) ou um atributo identidade (<i>identity attribute</i>).
Domínio (<i>Domain</i>)	Tipos de dados (<i>Data type</i>) Nulabilidade (<i>Nullability</i>) Chave estrangeira (<i>Foreign key</i>) Valor padrão (<i>Default constraint</i>) Regra (<i>Check constraint</i>)	A integridade de domínio visa a garantir que os dados de entrada em um atributo sejam validados pela limitação dos valores que podem assumir. Os tipos de dados que podem ser armazenados em um atributo é restrito pelos tipos de dados desse atributo. O formato dos dados é definido pelos valores das regras. A faixa de valores permitida para um atributo é especificada por meio de chaves estrangeiras, regras e <i>constraints default</i> .
Referencial (<i>Referential</i>)	Chave estrangeira (<i>Foreign key</i>)	A integridade referencial mantém a integridade dos dados por meio de ligação entre as relações (tabelas), testando-as quando tuplas (registros, linhas) são inseridas, apagadas ou modificadas. As ligações são definidas no projeto por meio das chaves primárias e estrangeiras.
Definidas pelo usuário (<i>User defined</i>)	Regras (<i>Check constraints</i>) Gatilhos (<i>Triggers</i>)	As regras definidas pelo usuário são baseadas nas regras de negócio específicas de cada banco de dados.

Fonte: Adaptado de Elmasri; Navathe (2006, p. 44-64)

Entre as tarefas a serem exercidas pelo SGBD, devemos também ressaltar o aspecto **transação**, que corresponde a uma coleção de operações que realizam uma função lógica simples, em uma aplicação de bancos de dados.

O SGBD possui propriedades que visam a garantir a integridade dos dados durante uma transação, geralmente conhecidas como **ACID: ATOMICIDADE; CONSISTÊNCIA; ISOLAMENTO; DURABILIDADE**.

Os significados desses termos são:

- a) **Atomicidade** é a característica de conclusão única (atômica) de todas as atividades dentro de uma transação. Por exemplo, se for necessário debitar uma conta e creditar outra, as duas operações farão parte de uma transação atômica: se uma não for realizada, a outra também não será, ou, se já foi, será desfeita;

- b) **Consistência** é a obediência às regras estabelecidas, às *constraints* do banco. Por exemplo, se um campo for definido como *UNIQUE* (valor único), digamos para o CPF de uma pessoa, esse valor não poderá ser duplicado;
- c) **Isolamento** quer dizer que as transações ocorrem sem se misturarem, cada uma independente da outra. Durante a execução de uma transação, os dados em uso ficarão **travados** para outras transações de forma a garantir que valores intermediários ou incorretos não venham a ser utilizados em outros processos;
- d) **Durabilidade** é o propósito final da sequência de operações e refere-se ao fato de que, uma vez executada, a transação gera resultados definitivos e estáveis, os quais somente poderão ser alterados por uma outra transação.



Atenção

Lembre-se: o termo ACID refere-se às características necessárias a uma transação e correspondem ao principal objetivo do uso de SGBDs: Atomicidade; Consistência; Isolamento; Durabilidade.

Vejamos um exemplo. As informações de uma obra (por exemplo ISBN, título e autor) são obtidas a partir de uma tela única de cadastro (digamos uma página de entrada de dados na *internet*). Mas, devem ser gravadas em relações (tabelas) separadas, conforme exemplificamos anteriormente. Nesse caso, poderíamos realizar a inclusão da obra a partir de uma sequência de operações:

Início da transação;

 Cadastrar o ISBN e título;

 Cadastrar o código e autor;

 Cadastrar a relação do título ao autor usando o ISBN e o código do autor;

Fim da transação.

Então, a execução de uma transação garantiria o cadastramento completo da obra na base: o título, o autor e o relacionamento. Para o exemplo proposto, as três operações (cadastro do livro, do autor e do relacionamento) seriam executadas entre o início (*BEGIN TRANSACTION*) e o final da transação. Ou seja, atomicidade garantida. Mas, e se após cadastrar o ISBN e o título, ou em qualquer outro momento a conexão falhasse? Nesse caso, todas as operações realizadas durante a transação (entre o início e o fim) seriam desfeitas (*ROLLBACK TRANSACTION*), voltando o banco de dados à situação anterior, garantindo sua consistência (a qual também seria garantida pelos mecanismos de integridade do gerenciador, já que, por exemplo, não seria possível cadastrar um relacionamento de um título com um autor inexistente).



Durante a execução dessa transação fictícia, outras com o mesmo acesso aos dados não seriam permitidas, evitando que os dados e/ou comandos se **misturassem**, nem seriam afetadas outras transações já realizadas, isto é, o isolamento. Por fim, se todas as operações da transação ocorrerem normalmente, ela será gravada definitivamente (*COMMIT TRANSACTION*) e as alterações realizadas permanecerão gravadas e disponíveis enquanto necessárias – durabilidade.

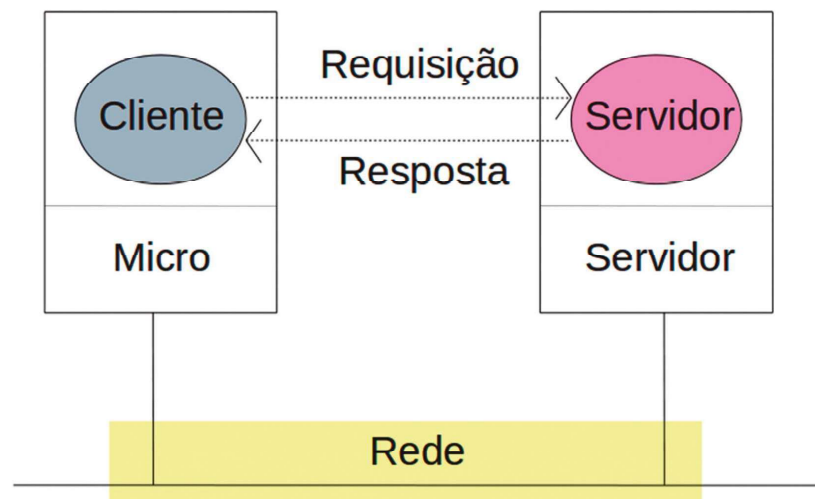
O SGBD é o responsável pela gerência de transações, por meio de seu gerenciador de transações, que é o componente que garante que o banco de dados permaneça consistente (correto) apesar de eventuais falhas de sistema (ex.: falta de energia, falhas no sistema operacional) e da execução de suas transações.

O gerenciador de controle de concorrência controla a utilização simultânea de recursos, a concorrência entre diferentes transações do sistema, garantindo a consistência, enquanto o gerenciamento de armazenamento é um módulo que provê interface entre os dados em baixo nível armazenados pelo banco de dados e os programas de aplicação e consultas. É responsável pelas seguintes tarefas: interação com o gerente de arquivos; armazenagem eficiente para recuperação e atualização de dados.

1.4.2.6 Ambiente de rede e segurança

Consideraremos aqui que as operações realizadas com o SGBD serão executadas em um ambiente tipo Cliente/Servidor, no qual um programa, chamado cliente, envia requisições para outro programa, chamado servidor, localizado em um computador remoto, por meio de uma rede de comunicação. Observe a imagem, a seguir.

Figura 10 - Representação da comunicação pela rede no ambiente cliente x servidor



Fonte: Adaptado de Silberschatz, Korth e Sudarshan (2006, p.18)

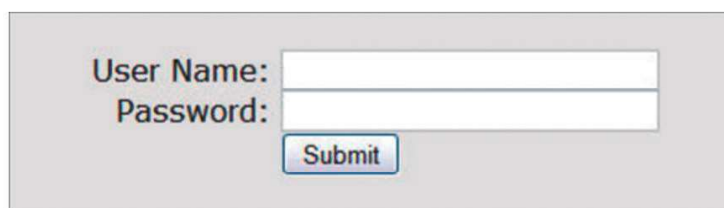
Esse é o modelo básico utilizado na *internet*, no qual o cliente é o navegador (*browser*) e o servidor é o programa que atende às requisições, por exemplo, um servidor de páginas HTML, como o *Apache* (PINTO, 2001). Para o caso dos SGBDs, o servidor poderia ser um *MySQL*, um *PostgreSQL*, um *Oracle*, um *DB2*, um *MS-SQL Server*, entre tantos outros.

Esse modelo também é adotado por quaisquer aplicativos que façam o uso do padrão *Common Gateway Interface* (CGI) para utilização de bancos de dados por meio da *internet*. Corresponde a um padrão de comunicação e troca de mensagens e formatos padronizados, os quais permitem que um navegador (por exemplo o *Firefox*) execute uma aplicação em um servidor (por exemplo, um usuário consultando seus empréstimos em uma biblioteca), e receba os resultados (PINTO, 2001).

Devemos observar que as transações ocorrem em um ambiente de rede e, portanto, podem ser monitoradas. Isso implica pensar em aspectos de segurança, como a utilização de **criptografia** e a **identificação segura de usuários**, além, obviamente, de uma boa política de senhas. Lembre-se de que o profissional de Biblioteconomia é um profissional de informação, e é responsável pela segurança e pela integridade desta.

Do ponto de vista do banco de dados, teremos sempre uma comunicação com um cliente identificado. No início da conexão, será fornecida uma identificação e uma senha correspondente (para este, usualmente, chamado de **login**).

Figura 11 - O par login/identificador (*user name*) e senha (*password*), após o usuário fazer a digitação das informações nos respectivos campos e clicar no botão enviar (*submit*), estabelece a comunicação do cliente identificado com o servidor do banco de dados. Se uma das informações for digitada incorretamente pelo usuário, a conexão não será concretizada

A imagem mostra um formulário de login simples. À esquerda, há duas rotulagens: "User Name:" e "Password:". À direita, há dois campos de entrada de texto empilhados verticalmente. Abaixo dos campos, há um botão retangular com o texto "Submit" em azul.

Fonte: Flickr⁵

Após a conexão ser efetivada com sucesso, podemos armazenar todas as transações realizadas pelo usuário/cliente, o que inclui registrar quem realizou, em qual momento (data/hora) e o que fez. Isso é chamado de LOG.

Se o LOG será ou não armazenado e, se for, por quanto tempo são decisões que transcendem o projeto do banco de dados e enquadram-se nas normas e **objetivos das políticas de informação da organização**. Do ponto de vista funcional, devemos observar que armazenar todas as transações realizadas gera uma sobrecarga de trabalho para o gerenciador e, evidentemente, um consumo de espaço de armazenamento e uma necessidade de cópias.


O cálculo foge ao escopo desta disciplina e do objetivo desta unidade, mas podemos estimar, a partir do número de transações e do número de usuários do sistema a quantidade de espaço que será ocupada – embora o armazenamento seja de forma compactada. E como o sistema tende a continuar a ser utilizado ao longo do tempo, essa quantidade tenderá a crescer indefinidamente, forçando o administrador do sistema a limpar os arquivos de tempos em tempos, eventualmente guardando cópias do que foi apagado – o que também se relaciona diretamente às políticas de segurança da organização.

Criptografia – é a operação de trocar os dados por outros seguindo uma chave secreta, de forma a impedir que sejam compreendidos. É bastante utilizada para a proteção de senhas e outras informações sensíveis.

Identificação segura de usuários – corresponde ao conjunto de mecanismos (incluindo-se *hardware* e *software*) utilizados para garantir que o usuário do sistema realmente seja o usuário cadastrado. A forma mais comum é o conjunto de login formado pelo par de identificação de usuário e senha, podendo ser ampliado por meio do uso de um *token* (uma espécie de *pen drive* que possui uma chave secreta para criptografia).



⁵ FLICKR. Paul O'Rear. Disponível em: <https://bit.ly/2LrxaWz>. Acesso em: 21 dez. 2018.



Quanto ao assunto cópias de segurança, esta é outra preocupação que o projetista do sistema deverá ter. Como e onde serão armazenadas e com que frequência serão realizadas as cópias e atualizadas são as razões de preocupação, porque, em caso de falha no sistema (de *software*, de *hardware*, por causa de **ataques** cibernéticos) deve haver uma forma de recuperar os conteúdos.

Será que os usuários devem ser responsáveis pela execução e pelo armazenamento das cópias de segurança dos bancos de dados?

Uma reflexão que ajuda a responder essa questão é a seguinte: você já perdeu algum arquivo por ter esquecido de fazer *backup*? De repente, aquele trabalho que você tinha de entregar para o seu chefe ou para a faculdade e, na hora da entrega, o arquivo estava corrompido ou com vírus... Se você tivesse feito uma cópia de segurança do arquivo em bom estado, isso provavelmente poderia ter sido evitado.

O mesmo acontece com os dados de um banco de dados. Não é aconselhável deixar nas mãos do usuário um procedimento tão crítico, melhor é deixar as atividades para um sistema automatizado ou para o pessoal técnico.

Nos ambientes organizacionais, o profissional responsável pela gerência dos sistemas de bancos de dados é conhecido como *Data Base Administrator* (DBA), ou administrador de bancos de dados.

Esse profissional coordena todas as atividades do sistema de bancos de dados e possui um bom entendimento sobre os recursos e necessidades de informação da empresa. Seus serviços incluem: a definição do esquema, a definição da estrutura de armazenamento e acesso, a modificação da organização do esquema físico, a delegação de autoridade de acesso ao banco de dados, a especificação de *constraints* de integridade, a atuação com os usuários, o monitoramento da performance e o atendimento às necessidades de melhoria (SILBERSCHATZ; KORTH; SUDARSHAN, 2011, p. 18).

1.4.3 Desenvolvimento de sistemas e bancos de dados

Desenvolver sistemas de informação é uma arte e inúmeros são os problemas a ela relacionados. Bancos de dados incluem-se nessa situação, seja como parte de um sistema de informações, seja representando o próprio sistema.

Com o passar dos anos, diversos pesquisadores têm proposto soluções para resolver ou atenuar alguns dos problemas continuamente pesquisados e relatados e, para facilitar o desenvolvimento dos sistemas, como não atender os requisitos do negócio ou dos usuários e no tocante à documentação atualizada. Entre os tópicos comuns, estão o bom senso e a forte presença de técnicas de documentação (STAIR, 1998, p. 34; LAUDON; LAUDON, 1999, p. 43-44).

Uma das técnicas de documentação surgidas é a engenharia da informação, que procura ligar os conceitos de sistema de informação com as necessidades estratégicas de informação das organizações (MACHADO; ABREU, 2004, p. 8-9).

Surgida nos anos 1980, essa técnica alinha-se às soluções atuais de modelagem de negócios descritas no *Data Management Body Of Knowledge* (DMBOK), no TOGAF, no *Archimate*, nas quais a participação das tecnologias da informação está fortemente ligada à estratégia da organização e à atenção ao usuário.



Explicativo

O termo DMBOK refere-se ao corpo do conhecimento em administração de dados, que para o livro é sinônimo de gestão da informação. O DMBOK apresenta dez funções a serem realizadas para garantir a governança da informação nas organizações. Entre essas funções, destacam-se a de gestão de dados mestres, metadados e de dados.

O TOGAF e o *Archimate* são modelos destinados à modelagem da arquitetura de informações das organizações com base em seu negócio principal, possuem objetivos semelhantes e apresentam estruturas (*frameworks*) para levantamento, conhecimento e gestão dos processos informacionais da organização, sendo que a ligação com a estratégia está sempre presente. Ou seja, não há implementação de soluções sem que exista um bom motivo estratégico, um objetivo, uma meta para a existência delas.

Em todos os casos é importante notar que existem usuários envolvidos no processo, seja para fornecer informações a respeito de um processo ou controle, seja como consumidores dos serviços que serão criados.

Já a engenharia da informação, segundo *James Martin*, citado por *Nascimento* (1993, p. 54) é o:

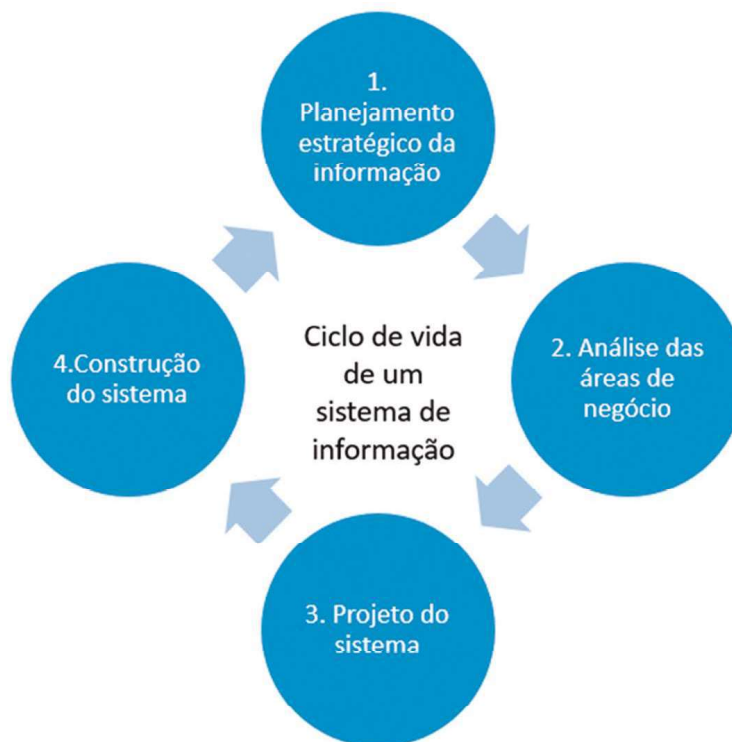
[...] conjunto de técnicas formais pelas quais os modelos de negócios, modelo de dados e modelo de processo são construídos em uma base de conhecimento compreensível e são usadas para desenvolver e manter sistemas de processamento de dados (NASCIMENTO, 1993, p. 54).

Assim como a informação tem um ciclo de vida, que vai de sua obtenção ao descarte ou armazenamento final, os sistemas possuem igualmente um ciclo de vida. Em algum momento eles deverão ser substituídos, em geral, porque o negócio a que atendem mudou e sua implementação não é mais compatível, ou porque houve uma evolução tecnológica que deve ser incorporada.

Modelo de dados – descreve quais são os dados importantes para uma organização incluindo as entidades; os metadados; os tipos de dados; os proprietários e a origem dos dados; e, a temporalidade de atualização e o armazenamento.



Figura 12 - Ciclo de vida de um sistema de informação para modelagem de negócios, de acordo com *Martin e Filkelstein (1984)*



Fonte: Produção do próprio autor

Para *Martin e Filkelstein (1984, p. 56-63)*, as fases do ciclo de vida de um sistema de informação são:

- a) **planejamento estratégico da informação** – aqui se definem a missão e os objetivos da instituição, por meio de análise dos planos de negócios, objetivos e seus fatores críticos de sucesso, durante um período de tempo de (por exemplo) 3 a 5 anos. O resultado é o modelo de dados e o **modelo funcional** da organização. Entre as atividades requeridas encontram-se organizar e controlar o projeto de planejamento de sistemas de informação, definir as estratégias, desenvolver o modelo inicial de informações e o modelo detalhado, o **perfil dos sistemas de informação** existentes e criar os relatórios de **necessidade de informação** (em caso de dúvidas, retorne ao conteúdo da disciplina de *Organização, Sistemas e Métodos*).

Entre os produtos dessa fase, temos: o relatório de necessidade de informações, o modelo de informações institucionais, o relatório de perfis de sistemas de informação (levantamento de sistemas existentes, dados e outros recursos) e o plano de sistema de informação (definição de projetos prioritários, com escalonamento de desenvolvimento).

- b) **análise das áreas de negócio** – área de negócio é uma parte conceitual da organização responsável pela execução de um conjunto de atividades que produz resultados bem definidos. Para cada área de negócio encontrada no planejamento estratégico das informações, um modelo lógico é desenvolvido, descrevendo um fluxo de dados, conteúdo e política de negócios e projetos alternativos.

Modelo funcional – conforme visto na disciplina de OSM, descreve como a organização está estruturada, o seu organograma.

Necessidade de informação – corresponde a um estudo de usuários e suas necessidades de informação, tendo foco no modelo de dados, no funcional e nos sistemas de informação existentes.

Perfil dos sistemas de informação – em conjunto com o modelo de dados, descreve quais os sistemas de informação que a organização possui e suas finalidades.



É selecionada uma estratégia de implementação. Entre as atividades dessa etapa estão: organizar e controlar o projeto da análise das áreas de negócio; a definição da área de negócios e a divisão das áreas de negócios, modelos existentes e modelos futuros; e, os requisitos de técnicas de documentação e abordagem de implementação.

Os produtos gerados nessa fase são: o modelo de informações para a área de negócios, os requisitos para a área de negócios, os relatórios de requisitos técnicos na área de negócios, a análise de lucros empresariais (se houver necessidade) e a abordagem da implementação na área de negócios (projeto inicial de sistema de alto nível).

A preocupação em **como** será implantado aquilo que foi definido na fase de análise. Os requisitos definidos pela análise das áreas de negócios são utilizados para produzir uma especificação detalhada de sistema. A participação do usuário é forte, assim como a utilização de protótipos. Uso de *Linguagens de Quarta Geração* (L4Gs).

- c) **projeto do sistema** – refere-se ao planejamento do projeto, e ao projeto em si. As atividades necessárias são: organizar e controlar o anteprojeto de projeto do sistema, determinar a(s) tecnologia(s) de apoio, projetar as mudanças culturais, projetar os bancos de dados, projetar os **sistemas de conversão**, projetar os testes de sistema e projetar os **programas de treinamento**.

Os produtos dessa fase são: detalhar o projeto do sistema, a referência de usuário (protótipo de interfaces, procedimentos manuais, mudanças organizacionais), o projeto de sistema de conversão, o projeto de teste de sistema e o projeto de programa de treinamento.

- d) **construção do sistema** – nessa fase, são gerados ou escritos **códigos-fonte** (ou L4G), testes, implantação de bases de dados, procedimentos de conversão de bases de dados, assim como o treinamento dos usuários e a implementação do sistema.

Suas atividades são: organização e controle do projeto e da construção, preparar a construção e testar o ambiente, implementar as especificações do projeto, implementar o desenvolvimento do produto, implementar o pacote de aplicação, compilar a documentação com o projeto de banco de dados e do repositório, desenvolver e conduzir o programa de treinamento, testar e instalar o sistema e passar para os usuários.

Essa fase gera os seguintes produtos: pacote de documentação de sistemas, banco de dados operacional, sistema operacional instalado, procedimentos de operação do sistema e programa de treinamento.

Em todas as etapas, a presença do usuário final é considerada fundamental, para que suas necessidades e requisitos sejam perfeitamente compreendidos e atendidos.

Para que todos sejam envolvidos, deveremos:

- a) determinar quais são os interessados (em inglês, *stakeholders*), selecionando os usuários mais experientes na área de negócio que será atendida e verificando quais são suas prioridades;

Sistemas de conversão – são utilizados para a conversão de tipos e estruturas de dados de um sistema para outro, por exemplo, na aquisição de um novo *software*.

Programas de treinamento – são as ações do RH em conjunto com a TI para garantir que tanto o pessoal técnico quanto os usuários saibam operar um sistema de informações.

Códigos-fonte – são as listas/arquivos contendo os códigos dos programas e *scripts* gerados para a implementação de um programa ou de um sistema de informação. Modernamente são escritos em uma linguagem de programação tal como C ou JAVA (em caso de dúvidas, retorne ao conteúdo da disciplina de *software livre*).



- b) verificar o organograma, o modelo de dados da organização e as políticas de segurança e de informação, mapeando as necessidades de informação e os requisitos de segurança;
- c) verificar com os usuários selecionados quais são as implicações para suas áreas de negócio: o que poderia melhorar, quais problemas poderiam acontecer (mapear os riscos);
- d) determinar se existem alternativas e documentar todas as informações obtidas e as decisões tomadas;
- e) determinar com os usuários do sistema as reais necessidades de dados e informações a serem armazenados e obtidos do sistema (pode ser feito por meio de entrevistas, observações, análise de documentos, produção de casos de uso etc. — ver adiante neste material);
- f) produzir uma visão integrada das entidades do mundo real envolvidas no processo e seus relacionamentos, originando um modelo que demonstre o que será armazenado em banco de dados (isso será abordado na Unidade 2);
- g) analisar a necessidade de treinamento da equipe e produzir um plano para supri-lo, se for o caso.

Além da questão usuário, você notou que o primeiro passo descrito prevê o alinhamento estratégico da solução a ser implementada com as necessidades da organização?

A análise deve ser voltada às necessidades do negócio, de forma que as ferramentas e os processos de análise e modelagem de negócios, como *Business Analysis Body of Knowledge* (**BABOK**) e *Business Process Model and Notation* (**BPMN**), por exemplo, poderão ser fortes aliados. Qualquer que seja a ferramenta, porém, estará sujeita ao efetivo levantamento e compreensão dos dados e informações pertinentes.

1.4.3.1 Levantamento de dados e requisitos do sistema

Para modelar um banco de dados, deveremos levantar os requisitos necessários ao nosso modelo. Com **nosso modelo** queremos dizer o modelo que atenda às necessidades da instituição e, conseqüentemente, dos nossos usuários. Esse modelo é único para cada organização, existindo diferenças de dados e sua organização mesmo que os objetos sejam semelhantes: por exemplo, o modelo de uma universidade não é necessariamente único, igual ao de outras universidades, apesar de todas trabalharem com alunos, professores e seus relacionamentos.

BABOK – corpo do conhecimento em análise de negócios: reúne as melhores práticas para descrever e analisar ambientes de negócio, registrando documentos e desenhos de entrada, ferramentas e técnicas, elementos e documentos de saídas. É uma boa fonte de informações para compreender os dados organizacionais e seus fluxos.

BPMN – notação para modelagem de processos de negócio: trata-se de uma técnica e um diagrama correlato que permite o desenho dos processos de negócio da organização segundo procedimentos padronizados, o que permite uma visão clara de como a organização efetivamente funciona e de quais são os dados e fluxos envolvidos.



Figura 13 - Antes de planejar modelar um banco de dados, é preciso conhecer os seus futuros usuários. No caso de uma biblioteca de uma universidade, por exemplo, significa saber quem são as pessoas que a utilizam, para que fins e quais as suas necessidades, para entender de que modo o banco de dados poderia contribuir para facilitá-las



Fonte: Flickr⁶

Considerando que o levantamento de requisitos, também conhecido como elicitación de requisitos, é a etapa inicial de um projeto de sistemas, devemos ter em mente que o objetivo desse levantamento é a determinação dos requisitos de negócio que a solução irá atender e quais necessidades tem o usuário (ou usuários) do sistema.

No projeto do banco de dados, levantar requisitos significa determinar: quais saídas (consultas, relatórios) o banco de dados deverá atender; o que deve ser armazenado e o que deverá ser calculado (e como calcular) para atender às saídas necessárias; a forma de armazenar os dados e suas associações; e, os tipos de dados necessários e/ou compatíveis com a realidade em estudo.

Para tal, devemos definir: quem é ou quem são os usuários e quais os seus papéis.

O levantamento dos dados necessários à implementação de um sistema de informação, do qual o banco de dados é uma das partes constituintes, passa pela análise de necessidades e requisitos dos usuários.

Independentemente da tecnologia ou da metodologia a ser utilizada, entra em cena o caráter científico do levantamento de dados, por exemplo, com pesquisas documentais, análises de documentos e técnicas de coleta de dados.

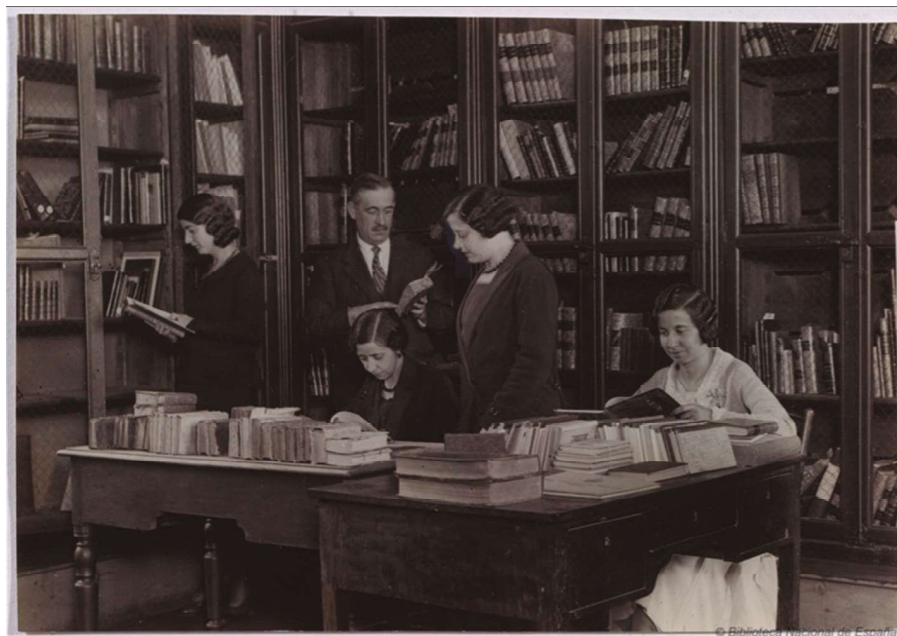
Destacamos aqui a observação participante das rotinas dos usuários:

Situação de pesquisa onde observador e observado encontram-se face a face, e onde o processo de coleta de dados se dá no próprio ambiente natural de vida dos observados, que passam a ser vistos não mais como objetos de pesquisa, mas como sujeitos que interagem em dado projeto de estudos (SERVA; JAIME, 1995, p. 69).

⁶ FLICKR. Universidad de Salamanca. Disponível em: <https://www.flickr.com/photos/universidaddesalamanca/16717060027/>. Acesso em: 21 dez 2018.

E, também, a pesquisa-ação, dada a necessidade de “[...] orientação para o futuro, facilitando a criação de soluções voltadas para um futuro desejável pelos interessados” e “[...] colaboração entre pesquisadores e clientes” (SUSMAN; EVERED, 1978, p. 589-590).

Figura 14 - É importante observar a rotina do usuário, seu modo de buscar informações e os tipos de informações pesquisadas, pensando em suprir as suas necessidades atuais, mas também orientadas para o futuro, pois os modos e as necessidades também são passíveis de mudanças com o passar do tempo



Fonte: Flickr⁷

Devemos ter em mente que o projeto de um banco de dados deverá levar em conta o que o casal *Laudon e Laudon* (1999, p. 149) chama de “[...] projeto de procedimentos e controle: especificar como cada aplicação deve ser executada, bem como o que deve ser feito para reduzir a possibilidade de crimes e fraudes”. Tais especificações incluem auditoria, cópias de segurança e distribuição de saídas (o que se obtém e para quem será entregue), quem pode consultar, inserir, alterar e apagar os dados, assim como definir a quais dados cada um tem acesso.

Isso tem a ver com os processos e as políticas de informação da organização, que são as fontes de consulta para esses requisitos. Por exemplo, poderia estar definido para o caso de uma biblioteca: a consulta ao acervo é disponível para todos os usuários, porém sua manutenção (inserção, alteração e exclusão, se for o caso) somente pode ser realizada pela bibliotecária. Da mesma forma poderia estar definido que todo usuário pode consultar seu histórico de empréstimos, porém somente a gestora da biblioteca poderá consultar os históricos de todos os usuários.

a) metodologia de casos de uso

Metodologias modernas, como a *Unified Modeling Language (UML)*, consagram o papel do usuário iniciando a modelagem por meio dos chamados casos de uso (*use cases*) e a descrição de cenários, nos quais o sistema é visto sob a ótica de seu utilizador.

A **UML**, linguagem de modelagem unificada, é formada por uma metodologia e um conjunto de diagramas que permitem documentar necessidades de informações a partir da visão do ambiente como sendo formado por objetos. A partir da visão e dos documentos oferecidos, fornece subsídios à implementação de sistemas de informação.



⁷ FLICKR. Vídea. Biblioteca Nacional da Espanha. Disponível em: <https://www.flickr.com/photos/bibliotecabne/6573937711/>. Acesso em: 21 dez. 2018.

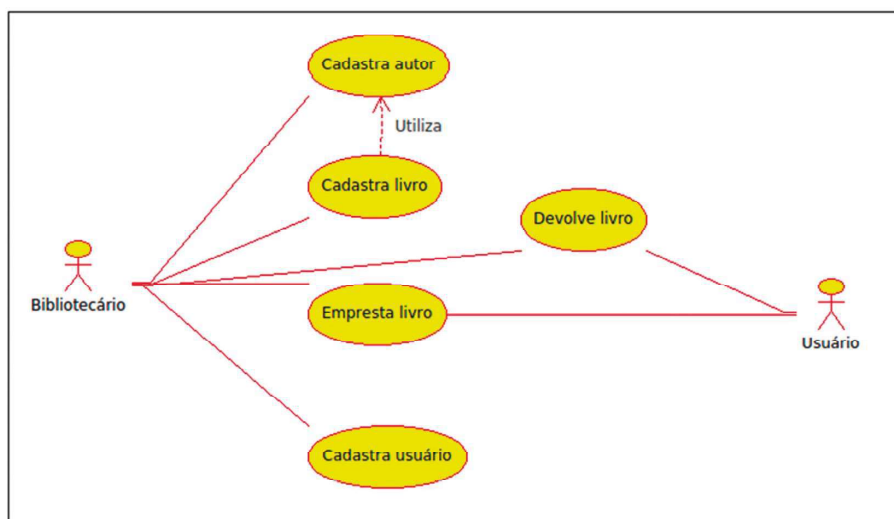
Mas, onde estão essas informações? As fontes a serem utilizadas para o levantamento de requisitos incluem: desenhos e descrições de processos de negócio, fluxogramas, diagramas de blocos, documentos da organização, entrevistas com usuários, observação de seu trabalho e documentos utilizados e observação das rotinas.

Um diagrama cuja descrição é uma boa fonte de informações é o caso de uso, da notação UML (Figura 3). Os diagramas de casos de uso são bastante utilizados atualmente para a modelagem de sistemas, e um de seus méritos é possibilitar o envolvimento dos usuários no processo de modelagem: na definição de entidades, de dados, de necessidades, de formatos, de responsabilidades e direitos sobre os dados e informações gerados e armazenados.

Vamos ver um exemplo?

Observe a imagem (Figura 15) a seguir.

Figura 15 - Exemplo de diagrama de casos de uso



Fonte: Produção do próprio autor⁸

A imagem (Figura 15) representa um diagrama de casos de uso de um sistema de biblioteca hipotético. No diagrama, os atores que participam da interação com o sistema são representados pelos bonecos, os quais estão ligados aos usos que fazem do sistema (os casos de uso, representados por elipses) por meio de linhas.

De forma resumida, a construção de diagramas de caso de uso passa pelo desenho do contexto e sua posterior documentação (conforme visto no exemplo em páginas anteriores). Para isso, primeiro determinam-se quais são os atores: quem vai interagir com o sistema? Seus usuários e seus administradores. No exemplo anteriormente apresentado, os atores foram resumidos em bibliotecário e usuário, que são duas classes diferentes de atuação, e também têm a ver com as citadas políticas de informação da organização. Depois são determinadas as ações principais a serem realizadas; no exemplo anterior, foram: cadastrar usuário, livro e autor, e emprestar e devolver livros; esses são os casos de uso do sistema (a figura, então, corresponde a um diagrama de casos de uso).

⁸ Produção do próprio autor utilizando o programa *Umbrello*.

A seguir, são ligados os casos (representados por elipses) aos atores correspondentes (representados por bonecos). Note que nem todos os atores possuem acesso a todos os casos de uso, implicando a implementação das políticas de informação e de segurança da informação. Finalizado o desenho, cada caso de uso será explicado (o que poderá levar à revisão do desenho e da descrição).

Quando utilizamos a metodologia de elaboração de casos de uso, é possível (e altamente recomendável) envolver os usuários do sistema, tanto para o desenho da usabilidade de pretendem ter quanto para termos noção da descrição de como gostariam de utilizar o sistema. Dessa forma, nenhuma funcionalidade desejada será esquecida.

Após o desenho, cada caso de uso será descrito em forma textual, no tocante às suas particularidades. Uma estrutura possível para essa descrição é: identificação do caso de uso; descrição resumida de sua finalidade; pré-condições, ou seja, o que deve ocorrer antes de o caso de uso ser utilizado ou executado; fluxo principal, contendo a sequência de operações a serem realizadas; fluxo alternativo, correspondendo, se for o caso, às ações a serem realizadas, caso o fluxo principal não possa ser executado; fluxo de exceção, o qual mostra as ações em caso de problemas; e, as pós-condições, ou seja, como ficará o sistema após a execução do caso de uso. Do texto descritivo do caso de uso, retiram-se as informações de modelagem.

Veja a seguir um possível descritivo de um dos casos de uso.

Caso de uso: Devolve livro

Descrição resumida: o usuário do sistema de biblioteca devolve o livro previamente emprestado e o controle de empréstimos é atualizado, juntamente com o de livros à disposição.

Pré-condições: pelo menos um livro emprestado

Fluxo principal:

- a) o usuário devolve o livro ao bibliotecário;
- b) o bibliotecário escaneia o código de barras do livro devolvido;
- c) o bibliotecário confirma a devolução.

Fluxo alternativo:

- a) o bibliotecário digita o código do livro devolvido.

Fluxo de exceção:

- a) o sistema informa que o código informado não existe.

Pós-condições: as tabelas de empréstimo e disponibilidade são atualizadas pelo sistema.

Note que a descrição desse exemplo é bastante resumida, mas oferece uma estrutura para pesquisa dos dados necessários à implementação do sistema. Nesse caso, sabemos que devemos trabalhar, minimamente, com o código do livro; que haverá operação de devolução (que pressupõe a de empréstimo prévio), de usuários e operadores do sistema previamente cadastrados e identificados, de atualização de tabelas em bancos de dados para efetivar as transações.

Segundo *Larman* (2004, p. 23-24), a partir de uma descrição textual, podemos extrair informação útil à modelagem por meio da seguinte técnica (por exemplo, a partir da análise do descritivo de um caso de uso):

- a) sublinhar substantivos para identificar conceitos no domínio do problema;
- b) usar os substantivos sublinhados para criar conceitos no modelo conceitual;
- c) poderão ser utilizados alguns dos substantivos, se eles identificarem tipos de dados simples, para criar os atributos dos conceitos, assim como os adjetivos;
- d) criar associações entre os conceitos.

Ao utilizarmos essa técnica, surgirão naturalmente os verbos nas frases analisadas. Já que os mesmos correspondem a ações, sua aparição dá origem a processos, funções ou métodos, conforme a metodologia em uso. Em bancos de dados, representam as operações a serem realizadas com os dados e podem dar origem a procedimentos armazenados (*stored procedures*).

Por exemplo, vejamos o seguinte texto: O usuário toma emprestados livros da biblioteca; os livros são identificados por seu ISBN, que é único, e pela combinação de título e autor, podendo ainda possuírem diferentes edições.

Desse texto, podemos verificar que temos identificados o livro e o usuário, os quais relacionam-se por meio do empréstimo. Já na descrição do livro, temos seu ISBN, seu título, seu autor, sua edição, que são características importantes da entidade livro.

Exemplificando, vamos imaginar uma situação comum, uma compra de mercadorias. Vamos a um supermercado e escolhemos alguns itens nas prateleiras. Depois, no caixa, ao passarem as mercadorias pelos leitores de códigos de barras, os códigos e descrições dos produtos podem ser impressos no cupom fiscal. Também no cupom será impressa a quantidade de itens e, em função dela, será calculado o preço daquele conjunto de itens, em função do preço unitário deles. Por último, o terminal Ponto de Vendas (PDV) calcula os totais e os impostos envolvidos na transação. Tudo rápido, automático e transparente para o cliente. Mas, de onde vem isso tudo? De um banco de dados, ou de mais de um, claro. Analisando o breve enunciado (descrição do procedimento), vemos que apareceu a palavra **calcular**.

Nem tudo o que teremos de apresentar em uma saída, por exemplo, o preço total de um conjunto de itens adquiridos, deverá necessariamente ser ou estar armazenado. Há muitas situações nas quais, para economizar espaço em disco, não iremos armazenar os dados, mas sim obtê-los a partir de outros dados. Por exemplo, armazenamos as parcelas e calculamos o total em uma venda, ou armazenamos as obras e calculamos o total de obras emprestadas.

Há situações intermediárias entre desempenho e dados de históricos, que geram grande número de consultas, nas quais poderemos ter uma tabela que especificamente traga dados calculados ou sumarizados a partir de outras tabelas. Essa condição é mais comumente utilizada quando temos dados históricos de consulta frequente, por exemplo,

quantos livros foram emprestados por mês no ano anterior? Nesse caso, é melhor calcular os valores para cada mês e para cada ano anterior, e armazená-los, já que não mudarão.

Lembre-se de que o seu projeto do banco de dados deverá levantar essas situações e documentar os conteúdos utilizados para a tomada de decisões no projeto e a situação final.

Definir o que será armazenado e o que será calculado é uma decisão que você tomará ao longo do projeto de seu banco de dados.

1.4.3.2 Participação do usuário

A partir do final da década de 1960 e do início até o meio da década de 1970, surgiu a discussão referente à chamada **crise do software**, relativa aos projetos de sistemas que apresentavam problemas de orçamentos e prazos não cumpridos e de qualidade baixa no produto entregue, em especial por, muitas vezes, não atender aos requisitos necessários.

Muitos estudos foram realizados e surgiu, como possível resposta, a disciplina e as técnicas da engenharia de *software*. Foram desenvolvidas e/ou aprimoradas diversas metodologias e processos para atender ao ciclo de vida do *software*, do levantamento inicial de requisitos, passando por modelagem, projeto, implementação, instalação e manutenção. Cada vez mais, os estudos mostraram que a participação do usuário era essencial para a garantia da qualidade do produto final.

Dentro dessa evolução de pensamento, as metodologias passam a não somente dar cada vez mais importância à participação do usuário, mas também passam a tentar garantir essa participação ao longo de todo o processo.

Ao desenvolvermos um novo sistema, então, deveremos primeiramente selecionar um usuário, ou um grupo deles, para colaborar no processo. Nessa escolha, definiremos o que podemos chamar de usuário padrão: aquele que conhece bem o negócio que estamos informatizando, que será seu principal usuário.

Figura 16 - A identificação dos usuários padrão e sua colaboração são fundamentais para a elaboração da modelagem do banco de dados



Fonte: Flickr⁹

⁹ FLICKR. Bibliotecas Redondela. Disponível em: <https://www.flickr.com/photos/bibliotecasredondela/11212147513/in/photostream/>. Acesso em: 21 dez. 2018.

Selecionados os colaboradores, verificaremos como será a interação com eles: entrevistas, reuniões, elaboração conjunta de modelos, utilização comum de ambientes virtuais para troca de informações, análise de documentos, observação. A combinação das técnicas varia de organização para organização e conforme o andamento do projeto.

Uma boa forma de começar o envolvimento é pedir ao seu usuário que descreva, da forma mais exata possível, o que ele precisa e como ele vê a solução que o atende. O usuário será auxiliado nessa tarefa pelo modelo de documentação de sistemas adotado na organização. Para facilitar, a UML, por exemplo, utiliza os casos de uso, de forma que se possa deixar que o próprio usuário retrate, no momento inicial, como ele vê o sistema a ser desenvolvido.

Mas, há outras formas:

- a) solicite ao seu usuário que desenhe um relatório (saída) ou o resultado de uma consulta que deseja que o sistema ofereça. Analisando os dados de saída, você poderá detectar o que será necessário como entrada, seja para armazenar, seja para calcular. Por exemplo, dados de entrada poderiam ser os ISBNs dos livros emprestados; e, os dados de saída, os 10 títulos mais emprestados em ordem decrescente;
- b) peça ao usuário que esboce a cópia de uma tela de entrada de dados ou um formulário. Nesse caso, ele estará fornecendo os dados e entidades que lhe são importantes. A partir dele, pode ser efetuada a modelagem conceitual, que também pode ser confirmada com o usuário (isso será visto na próxima unidade);
- c) converse com o usuário pedindo que ele lhe explique o que ele faz e que informações ele utiliza. O que ele precisa fazer com essas informações? Organizá-las, tabular, contar, gerar outras? A conversa pode ser uma entrevista estruturada ou não estruturada, desde que sejam registrados os pontos principais;
- d) seu usuário já possui um sistema não informatizado que ele utiliza para controle de seus afazeres? Nesse caso, deverá haver documentos que podem ser analisados para a verificação dos conteúdos a serem informatizados;
- e) podemos utilizar a observação dos processos realizados pelos seus usuários, com acompanhamento e registro das rotinas e dados necessários para realizá-las;
- f) outra forma é a observação participante, na qual você fará o papel do usuário do serviço e tentará identificar e registrar os pontos importantes. Colocar-se no lugar do outro pode esclarecer muitas coisas; porém, você deverá ter algum domínio das rotinas/processos a realizar. Nesse caso, é bom contar com o apoio de um **usuário padrão** e com uma boa revisão.

Uma vez levantados os dados, você deverá tomar cuidado para não cometer um erro clássico: não informatize o que não é o melhor; se há falhas percebidas ou melhorias a executar, faça isso antes de colocar a situação em um sistema ou banco de dados.

Outra situação a evitar é não se ater ao presente: hoje funciona assim, portanto, vamos informatizar assim que o negócio será atendido. Não, nem sempre. Alguns sistemas levam tempo considerável para serem



entregues, e os negócios são dinâmicos. **Procure sempre enxergar o futuro: como gostaríamos que fosse essa solução?**

Após a realização do modelo, virá sua implantação. Nas unidades seguintes, serão dados exemplos de desenho e implementação de bancos de dados.

Nessas duas fases, a participação do usuário continua a ser importantíssima, pois teremos primeiro o modelo, o roteiro a ser seguido para a implementação do sistema, que deve ser validado.

Figura 17 - A participação do usuário é central no processo, desde a concepção, passando pelo desenho, pela implementação e pela validação de um banco de dados



Fonte: Flickr¹⁰

Quando o banco de dados for efetivamente criado, poderemos ainda lançar mão da metodologia de prototipagem: um modelo funcional será entregue ao usuário que realizará críticas e testes sob acompanhamento, garantindo que, uma vez instalado, ele efetivamente cumprirá o papel para o qual foi concebido.

1.4.3.3 *Breves considerações a respeito da tecnologia*

Por fim, uma vez determinado que é necessário um banco de dados, assim como o que será armazenado nele e as operações necessárias, devemos realizar um estudo da sua viabilidade tecnológica, relativa às condições necessárias para a implantação e a manutenção da solução ao longo do tempo. Esta última garante a disponibilidade e a continuidade do negócio, do processo e das informações que serão requeridas do sistema ao longo do tempo.

¹⁰ FLICKR. Jisc infoNet. Disponível em: <https://www.flickr.com/photos/jiscinfonet/287757737/>. Acesso em: 21 dez. 2018.

Essa análise é importante, pois devemos ter em mente que a implantação de um sistema em geral não é tarefa rápida; uma solução não pode ser substituída de pronto por outra, sempre há um estágio de planejamento, implantação e adaptação. Assim, é importante fundamentar as escolhas de forma a garantir que aquilo que for escolhido continue funcionando sob condições originalmente previstas por algum tempo (em geral, 2 a 5 anos).

Algumas das perguntas a serem feitas, e respondidas, são:

- a) Já existe uma solução pronta?
- b) Quanto custa?
- c) Vale mais a pena comprar ou desenvolver?
- d) Qual é a complexidade envolvida?
- e) Dominamos a tecnologia?

Independentemente de a solução ser comprada ou desenvolvida, devemos considerar que tipos de equipamentos são necessários, qual sistema operacional deve ser usado, qual é a competência das pessoas envolvidas, se há a necessidade de treinamento e de suporte e, claro, qual é o custo de tudo isso.

As decisões passarão pela questão da tecnologia proprietária x livre, que o aluno já teve em outra disciplina deste curso. Mas não custa lembrar que “livre” não significa “sem custos”, e que “pago” não significa necessariamente “continuidade de negócio”, nem que o valor pago inicialmente cobrirá todas as manutenções futuras.

1.4.4 Os níveis de abstração e a modelagem do banco de dados

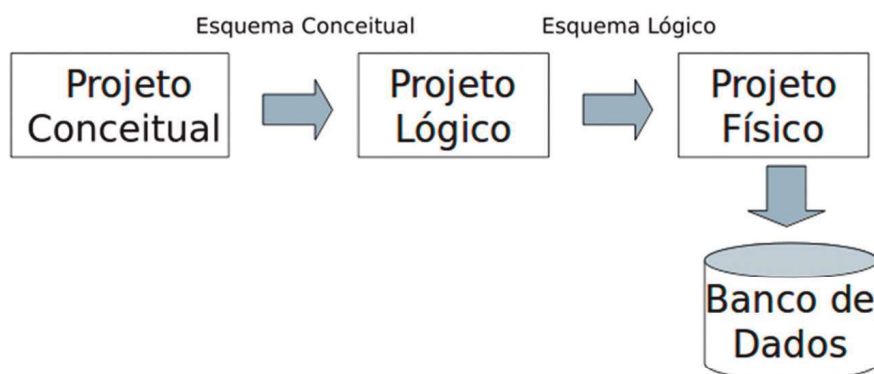
Ao trabalharmos com bancos de dados, temos de lidar com o nível de abstração a ser utilizado, referente às diferentes visões que um mesmo projeto de banco de dados pode apresentar. Essas visões referem-se aos níveis conceitual, lógico e físico, e correspondem às visões dos diferentes interessados envolvidos no processo, usuários de níveis diferentes:

- a) analistas de negócio;
- b) clientes e fornecedores;
- c) programadores;
- d) administrador de bancos de dados.

Cada uma dessas visões dará origem a esquemas (ou diagramas) diferentes, os quais simbolizam, em síntese, a mesma realidade. Veja na imagem (Figura 18) a seguir.



Figura 18 - Representação da sequência de projeto de bancos de dados



Fonte: Adaptado de Elmasri e Navathe (2011, p. 132)

Na imagem (Figura 18), vemos que o banco de dados depende de seu projeto físico, que corresponde às estruturas dos dados que serão armazenados. O projeto físico, por sua vez, depende da visão de relacionamentos lógicos que são abstraídas a partir da visão conceitual. Destacaremos aqui o projeto conceitual.

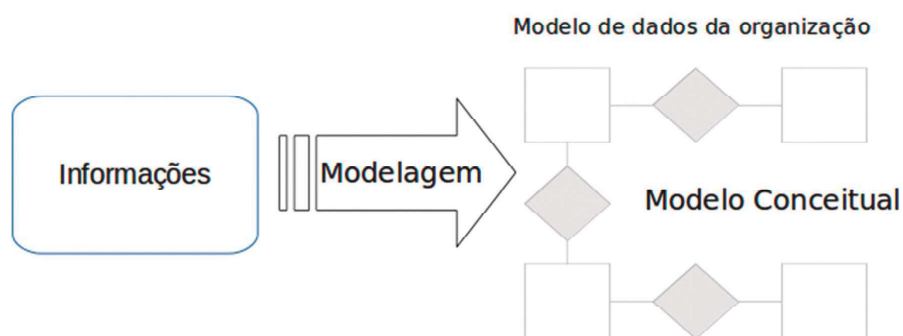
O **modelo conceitual** corresponde ao ponto de vista do projetista, e do usuário, e facilita o entendimento do modelo da organização, tornando o projeto final mais estável. Esse modelo independe do SGBD, o que facilita modificações.

Por possibilitar uma visão comum entre projetista e usuário, um modelo facilita a comunicação e, conseqüentemente, o entendimento para obter o produto desejado, além de auxiliar no controle do projeto.

A modelagem visa a resolver questões relativas à necessidade de informação da organização, tais como: quantos funcionários trabalham na seção de vendas, ou quantos não tiraram férias em janeiro, ou que tipo de produtos os clientes compram nos meses de inverno?

A imagem (Figura 19) a seguir, representa a ligação entre as informações disponíveis na organização e sua representação em modelagem conceitual. Note que a origem do modelo vem da modelagem das informações importantes ao usuário/organização.

Figura 19 - Criação de um modelo conceitual a partir dos requisitos do usuário



Fonte: Adaptado de Machado e Abreu (2004, p. 24)

Enfim, a modelagem conceitual atua na identificação da funcionalidade e dos dados necessários para um sistema, e na determinação da melhor forma de agrupamento dos dados. Essas atividades não dependem de um sistema gerenciador de bancos de dados específico! São atividades que geram um modelo, uma representação, aplicável a quaisquer sistemas de processamento de dados e informações (incluindo os não informatizados). Observe que a modelagem é o processo que leva ao modelo conceitual, e ela tem como entrada o conjunto de informações fornecido pelos usuários do sistema. Para a modelagem de um sistema de bibliotecas, devemos pensar em quais serão seus usuários, entrevistá-los, obter informações de sistemas já existentes (incluindo não informatizados) e, a partir daí, esboçar o modelo e submetê-lo a críticas/validação.

Trata-se de um processo de modelagem de dados que tem por essência transformar um universo infinito de informações em um universo finito e relacionado de entidades. Atua capturando a visão estática de um problema ou situação.



Explicativo

Modelagem de dados é o “processo de estruturar as informações do mundo real segundo um modelo de dados” e tem por objetivo obter um bom modelo, que garanta transações corretas e eficientes. Transação é a “execução de um programa que acessa ou modifica o conteúdo de um banco de dados”.



1.4.5 Atividade

Imagine que você, como bibliotecário, foi solicitado a acompanhar o projeto de desenvolvimento de um SGBD para a sua biblioteca. Na etapa inicial do projeto, você, como usuário, foi convidado a desenhar a tela inicial do sistema envolvendo o empréstimo e devolução de livros.

Desenhe no espaço a seguir a tela de como você imagina a sequência de empréstimo e devolução de livros.

A partir do seu desenho, responda às seguintes questões:

Qual é a importância de você, como bibliotecário, fazer parte da equipe de desenvolvimento de um novo SGBD para uma biblioteca?

A partir do seu desenho, identifique o que seria um dado, um metadado e um tipo de dado.

Nesta etapa, o seu desenho representa que tipo de visão do banco de dados? Justifique.

Resposta comentada

<p>Sistema <u>BIBLIOTECA</u></p> <p>EMPRESTIMO</p> <p>Usuário: <input type="text"/></p> <p>ISBN: <input type="text"/></p> <p>EMPRESTAR</p>	<p>Sistema <u>BIBLIOTECA</u></p> <p>DEVOLUÇÃO</p> <p>Usuário: <input type="text"/></p> <p>ISBN: <input type="text"/></p> <p>DEVOLVER</p>
--	--

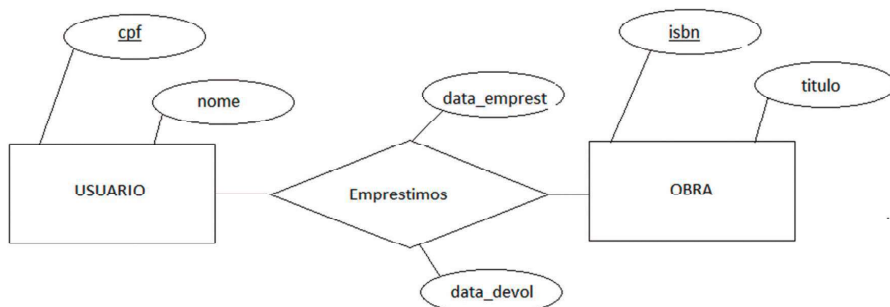
Com relação à questão (a), a sua participação no desenvolvimento de um SGBD para a sua biblioteca seria fundamental, considerando que vimos que uma etapa do projeto é justamente envolver os diversos tipos de usuário para que identifiquem o processo e suas respectivas necessidades para que o sistema possa melhor atender às demandas. Em resposta à questão (b), fizemos um desenho genérico apenas para ilustrar como você poderia ter desenhado. Em nosso desenho, os metadados seriam o código do usuário e o ISBN do livro, os dados seriam o código do usuário e o número do ISBN, e os tipos de dados seriam números inteiros para ISBN e

o mesmo (ou caracteres alfanuméricos) para o código do usuário (isso pode variar, de acordo com a organização em estudo). Um bom identificador para o usuário poderia ser seu CPF, formado por 11 números inteiros. Por fim, na questão (c), o tipo de visão que o seu desenho representa é a das informações que serão utilizadas para a elaboração do modelo lógico, porque identificam os metadados de algumas das entidades que participarão do processo de empréstimo e de devolução, que constituirão uma entidade de relacionamento entre as anteriores. Note que o desenho de telas somente fornece informações, não o modelo lógico que será utilizado **por trás**, o qual será construído a partir dessas informações.

Observação: ao desenhar a tela, você está indiretamente interagindo com pelo menos três entidades do banco de dados: a entidade usuário, que representa as pessoas; a entidade livro ou obra, que representa o objeto físico à disposição do usuário; e, o relacionamento que você quer dar a eles: empréstimo ou devolução. Embora a concepção da tela seja simples, na verdade, o banco de dados, ao receber os dados relativos ao código do usuário e ao ISBN deverá:

- testar primeiramente para verificar se ambos existem, retornando um erro se um dos dois ou ambos estiverem incorretos;
- verificar se é um empréstimo; e, nesse caso, armazenar as informações de qual usuário tomou emprestada qual obra;
- se for devolução, pesquisar na tabela de empréstimos a combinação usuário e ISBN e marcar como devolvido. Isso sem contar que pode haver uma tabela de reserva a ser consultada antes, para verificar se a obra de fato está disponível.

De qualquer forma, desenhos de tela são úteis para envolver os usuários no processo. Um MER capaz de representar parcialmente o que foi aqui apresentado poderia ser:



CONCLUSÃO

O enorme conjunto de funcionalidades e características pertinentes aos SGBDs é que lhes dá a amplitude de aplicações que hoje possuem. Em especial, note que a complexidade referente à execução das tarefas de controle e garantia dos bancos de dados por eles gerenciados não fica visível aos usuários, os quais, em contrapartida, não precisam ter profun-

dos conhecimentos de ciência da computação para utilizá-los. Basta saber informar o modelo e executar as consultas. Isso será visto na sequência desta disciplina. Mas, para chegarmos às consultas e, antes, no modelo, precisamos levantar as informações necessárias, e documentá-las bem.

Nas unidades seguintes desta disciplina, criaremos modelos e os transformaremos em um banco de dados funcional, ou seja, naquele pronto para uso pelos usuários, cujo modelo foi validado e está pronto para utilização. O início da modelagem, porém, depende dos conceitos vistos nesta unidade, para o levantamento dos dados necessários à estratégia de negócios.

Por isso, lembre-se de que suas fontes de informação principais serão a documentação do negócio e seus processos e os usuários que fazem parte destes, seja no papel de alimentadores de dados e informações, seja no papel de consumidores ou **clientes finais**.

RESUMO

Bancos de dados fazem parte das aplicações que nos prestam serviço em nosso dia a dia. São controlados por sistemas gerenciadores, resultando na sigla SGBD. Os mais utilizados atualmente são os relacionais, e seus gerenciadores são chamados SGBD-R.

Os SGBDs trabalham basicamente armazenando, transformando e recuperando dados, parte básica e com significado limitado, definidos pelos metadados, que descrevem os atributos e tipos, tais como numéricos ou caracteres. Essas operações visam à obtenção e ao gerenciamento de informações, coleções ou interpretações dos dados, agora ligados a um significado útil em certo contexto.

Nos bancos de dados relacionais (SGBD-R), as entidades de negócio serão representadas por tabelas, as quais possuem colunas, que definem os metadados de interesse. As linhas das tabelas conterão dados, que, em conjunto, são chamados também de informações (por exemplo, uma linha em uma tabela de pessoas representa o conjunto de informações de uma pessoa específica).

O gerenciamento das informações leva à formalização dos conhecimentos organizacionais, que, se introduzidos nos ambientes informacionais, nos sistemas de informação, garantem um bom ciclo de vida aos dados de interesse.

Na implementação de bancos de dados serão utilizadas restrições (*constraints*) para garantir sua operação íntegra dentro do modelo proposto. Entre os exemplos de restrições comumente utilizadas, temos declarações para impedir que sejam cadastrados valores nulos, ou seja, que deixemos certo campo essencial sem preenchimento, para garantir unicidade de valores e das chamadas chaves: a chave primária, que identifica unicamente uma entidade, e a chave estrangeira, utilizada para o relacionamento entre as diversas entidades do sistema (as entidades serão implementadas na forma de tabelas).

Na modelagem de bancos de dados, com frequência aparecem duas restrições que são parte essencial do relacionamento entre as tabelas, as chaves primárias e estrangeiras. Definimos a chave primária (*primary key*) como sendo o atributo ou conjunto de atributos que permitem identificar unicamente uma relação (tabela). Nesse caso, não podem existir para esse atributo (essa coluna) dois valores iguais em tuplas (linhas) diferentes. A chave estrangeira (*foreign key*) é a chave primária de uma relação exportada para outra. Por se tratar de uma representação de relacionamento entre relações, considerando uma repetição de relacionamento, ocorrerá uma repetição do valor desse atributo em tuplas diferentes.

Os gerenciadores trabalham garantindo o controle de transações, que são as operações submetidas aos SGBDs que devem ser processadas e devolvidas. Por funcionarem em ambiente de rede, atualmente em especial na *internet*, devemos ter um bom projeto de segurança ao implementá-los, pois informações essenciais à organização poderão estar envolvidas.

As transações devem atender aos preceitos do termo ACID: respeitar a Atomicidade (conclusão única ou atômica de todas as atividades dentro de uma transação), a Consistência (obediência às regras estabelecidas para o banco), o Isolamento (cada transação ocorre em seu espaço sem misturar-se a outras) e a Durabilidade (o propósito final, gerar resultados estáveis ao final da sequência de operações). Essas garantias são implementadas pelo SGBD.

Os bancos de dados são implementados a partir de um modelo conceitual, o qual analisa os metadados, os dados e as informações da organização, para depois sugerir suas associações. Como visam a atender à organização e todos os seus usuários, a participação destes é fundamental.

Outro ponto importante a ser considerado é no tocante ao domínio da tecnologia que será utilizada e no seu custo, tanto de aquisição quanto de manutenção. Uma vez que dados da organização tenham sido inseridos em um SGBD, ele terá que ser mantido.

Vale lembrar que a participação do usuário é muito importante para a modelagem de um banco de dados, desde a concepção, passando pelo desenho até a implementação e o teste do sistema.

