

# BANCO DE DADOS

Roni Francisco Pichetti



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS



# Projeto de BD: criando BD

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Construir bancos de dados nos modelos lógico e físico.
- Desenvolver um projeto lógico de banco de dados com Workbench.
- Praticar a modelagem física de banco de dados com MySQL.

## Introdução

Os bancos de dados (BDs) são utilizados para armazenar dados que serão úteis para atender às necessidades de seus usuários. O processo de construção de um BD envolve conhecer o ambiente para o qual ele está sendo criado, para que o projeto contemple a maior quantidade possível de informações. As informações necessárias são utilizadas para elaborar os modelos que fazem parte da documentação do BD, a qual é importante, pois serve como base para futuras implementações, adequações ou manutenções que se mostrarem necessárias para aumentar o desempenho de sistemas que utilizem os BDs.

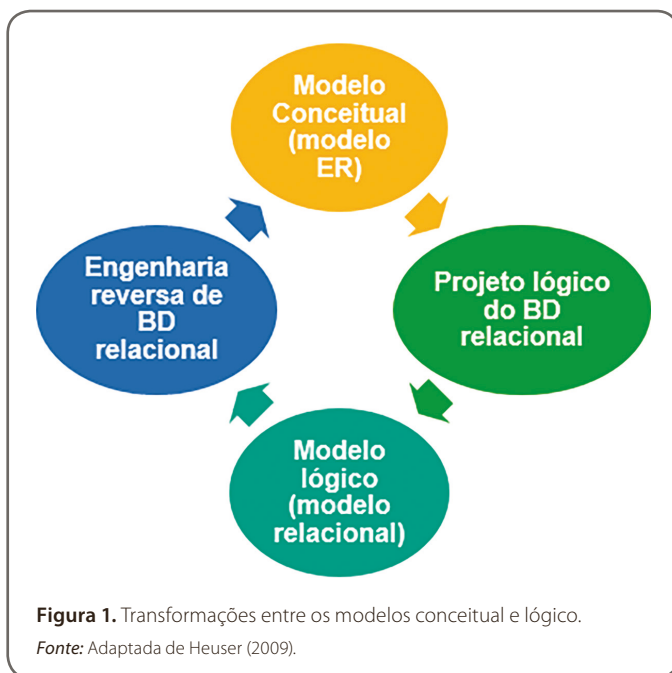
Neste capítulo, você vai estudar a construção dos modelos lógico e físico de BDs. Você também vai verificar exemplos de desenvolvimento desses modelos em duas ferramentas específicas: Workbench e MySQL.

## 1 Construção de banco de dados utilizando os modelos lógico e físico

A construção de um **BD relacional** conta com diferentes fases, que contribuem para a definição de suas relações de forma efetiva, visando ao bom desempenho no armazenamento e na manipulação de dados. Cada fase possui uma função específica, e se utilizam modelos para implementar cada etapa. O resultado de cada modelo é utilizado para a construção do modelo seguinte.

A primeira fase costuma utilizar o **modelo conceitual**, que não leva em conta o sistema de gerenciamento de BD (SGBD) que será utilizado, pois é uma forma mais abstrata da definição dos relacionamentos entre as tabelas ou entidades que vão compor o BD. Em seguida, utiliza-se o **modelo lógico**, que se preocupa em representar os dados da forma como serão armazenados no SGBD. Esse modelo precisa ser descrito em alguma linguagem, que é o papel da terceira fase, quando ocorre a construção do **modelo físico**. O processo inverso também pode ser necessário, ou seja, construir um modelo conceitual a partir de um modelo lógico — esse processo é chamado de **engenharia reversa**, conforme leciona Heuser (2009).

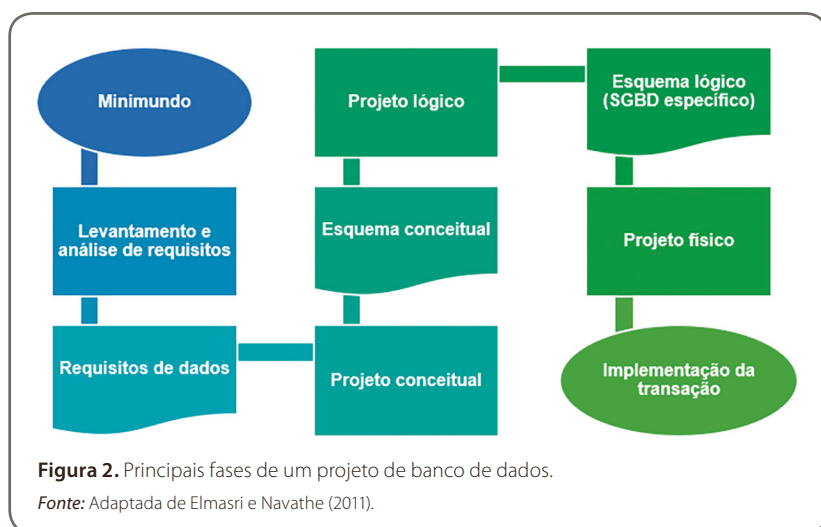
A transição entre os modelos conceitual e lógico e a engenharia reversa são representadas na Figura 1.



A criação de um BD robusto, flexível e preciso em suas possibilidades está relacionada ao desenvolvimento adequado de um **projeto de BD**. Nesse sentido, a teoria de projetos é considerada a base científica para os projetos de BD, enquanto o **modelo relacional** é a base científica mais comum para ferramentas de BDs. Por isso, todos os profissionais envolvidos com projetos de BDs necessitam compreender o modelo relacional, do qual fazem parte os modelos conceitual, lógico e físico (DATE, 2015).

A engenharia reversa entre modelos é útil quando não se tem modelos de um BD ou de um modelo lógico que já existe. Ou, ainda, quando um BD tenha passado por modificações por um período de tempo e elas não tenham sido documentadas em um modelo. Esse processo contribui para uma maior compreensão do relacionamento entre as entidades presentes no BD. O projeto lógico de BD relacional faz parte da transformação de um modelo conceitual em lógico e de um modelo lógico em físico. Os modelos conceituais podem ser implementados por meio de variados modelos lógicos, sendo que cada um deles tem influência direta no desempenho do sistema que utilizará o BD. Isso porque esses modelos podem provocar maior ou menor facilidade de desenvolvimento e manutenção do sistema (HEUSER, 2009).

Para que a construção de um BD tenha sequência, após a definição do modelo lógico, é preciso definir o modelo físico, aquele que realizará a comunicação do BD com a aplicação utilizada pelo usuário. Todo esse processo de desenvolvimento de um projeto de BD é ilustrado de forma resumida na Figura 2.



O primeiro passo é a identificação do **minimundo**, o qual o BD deverá representar. No minimundo, é realizado o levantamento e a análise dos requisitos. Os requisitos precisam ser especificados da forma mais completa possível, o que auxiliará a construção do projeto conceitual por meio de um esquema. O **esquema conceitual** é composto por uma descrição concisa dos requisitos, que envolve as entidades, as restrições e os relacionamentos necessários. A etapa seguinte, também presente na Figura 2, é o **projeto lógico**. No modelo lógico, deve ser considerado o modelo de dados adotado no projeto de BD — nesse caso, trata-se do modelo relacional (ELMASRI; NAVATHE, 2011).

As partes principais dos modelos relacionais são as **tabelas ou relações**, que são compostas por uma ou mais colunas. As colunas que servem para identificar as tabelas são chamadas de **chaves primárias**. Essas chaves primárias podem ser compostas por uma ou mais colunas da tabela.



### Exemplo

Pense em uma tabela de alunos de uma faculdade, na qual constam o nome, a matrícula e o CPF de cada aluno. Tanto a matrícula quanto o CPF podem ser a chave primária. Essas chaves auxiliam na construção dos relacionamentos entre as tabelas (COSTA, 2006).

Geralmente, os relacionamentos são representados pela migração da chave primária de uma tabela para outra, e essa coluna será uma **chave estrangeira**. A chave estrangeira possibilita a correlação entre linhas de diferentes tabelas.



### Exemplo

Sobre a chave estrangeira, considere duas tabelas em um BD de uma empresa: colaborador e departamento. Cada colaborador está alocado em um departamento, e na tabela “Departamento” existe a coluna “Código\_Departamento”. Para que no BD ocorra essa relação entre as duas tabelas, na tabela “Colaborador” existirá também uma coluna “Código\_Departamento”, a qual será uma chave estrangeira, referenciada à sua tabela original.

Outra informação importante sobre a criação de tabelas é que elas podem representar os atributos de forma especializada ou genérica. A **especialização** se trata de informações ou dados particulares sobre um conjunto de ocorrências de uma tabela, enquanto a **generalização** trata as informações genéricas. Por exemplo, em uma tabela de clientes, estes podem possuir dados especializados sobre seu cadastro, como CNPJ, para empresas, e CPF, para pessoas físicas. Ou, ainda, em um cadastro de funcionários de uma empresa, alguns dados são comuns a todos, como nome, endereço, data de nascimento, entre outros. Porém, alguns podem possuir particularidades de sua carreira ou formação, como administradores possuem o número de registro no Conselho Regional de Administração e engenheiros, no Conselho Regional de Engenharia e Agronomia (COSTA, 2006).

A fase final é o **projeto físico**, como pode ser visto na Figura 2, no qual são consideradas as características específicas do SGBD que será utilizado na implantação do BD. No projeto físico, além dos relacionamentos, das tabelas e das colunas, estão descritos os tipos de dados para cada coluna, e há a preocupação com a linguagem de criação das tabelas no BD (ELMASRI; NAVATHE, 2011).



### Fique atento

O processo de construção do BD, do projeto conceitual ao físico, é uma das responsabilidades do administrador do BD, ou DBA (do inglês *database administrator*). Cabe a ele realizar a interação com os usuários do BD, para entender quais dados necessitam ser armazenados no SGBD e como eles serão utilizados. Somente com esse conhecimento será possível projetar tanto o esquema conceitual quanto o físico (RAMAKRISHNAN; GEHRKE, 2011).

Portanto, todas as etapas de construção de um projeto de BD são interdependentes. Isso quer dizer que uma depende do resultado da outra, mas também que uma compreensão incorreta da análise dos requisitos, por exemplo, tem influência direta no resultado final e pode comprometer o posterior funcionamento do BD como um todo.

## 2 Projeto lógico de banco de dados com *Workbench*

Conforme citado anteriormente, o projeto lógico é a etapa do projeto de BD em que é criado um esquema lógico de acordo com o SGBD que for selecionado, para representar os esquemas da modelagem conceitual. Essa fase pode ser descrita em duas partes principais:

- mapeamento do modelo de dados independente do sistema, em que ainda não são levadas em conta as especificações do SGBD; e
- ajuste para um SGBD específico, momento no qual há a preocupação com os recursos de modelagem e restrições próprias.

O resultado da segunda parte são os comandos DDL (Data Definition Language, ou Linguagem de Definição de Dados) na linguagem utilizada no SGBD escolhido para o projeto físico (ELMASRI; NAVATHE, 2011).

Veja agora um exemplo, para ilustrar a construção de um projeto de BD. Trata-se do BD chamado “Escola”. Ele é utilizado para o registro de professores, alunos, turmas e disciplinas — portanto, esse é o seu minimundo, a parte da escola observada. Trata-se de uma escola que atende a alunos do ensino fundamental. A descrição dos requisitos de dados, obtida no levantamento e na análise dos requisitos, é apresentada a seguir.

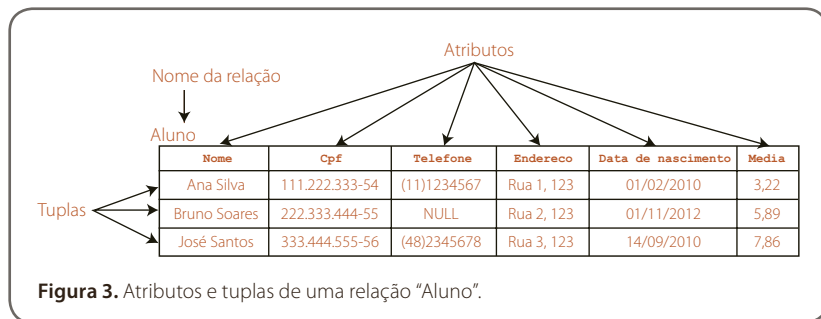
- A escola conta com diferentes professores, que, de acordo com sua formação, ministram diferentes disciplinas. Eles possuem como exclusivos os dados nome, CPF e um código interno, o código do professor. Portanto, um professor pode atender a diferentes disciplinas.
- Já as turmas são utilizadas para alocar os alunos, bem como para dividir os alunos de acordo com o ano em que estão matriculados. Para os alunos, são registrados dados pessoais como nome, endereço, data de nascimento e contato.

Para a modelagem lógica deste exemplo, utilizou-se o **MySQL Workbench** e, para sua transformação em projeto físico, o SGBD **MySQL Community**. Essas duas ferramentas são classificadas como open source (código aberto), o que permite sua utilização de forma gratuita, desde que não seja para fins comerciais. Para fins comerciais, existe uma licença específica, que precisa ser paga. Seu *download* está disponível na página do MySQL.

O MySQL Workbench é uma ferramenta visual que possibilita a modelagem de dados com modelos relacionais, voltados para a Linguagem de Consulta Estruturada (SQL, do inglês Structured Query Language), entre outras funções. Ela é compatível com os sistemas operacionais Linux, Mac OS X e Windows (MYSQL, 2020).

O desenvolvimento do projeto do BD “Escola” depende da definição de diferentes entidades, entre elas: professor, alunos, turmas e disciplinas. Cada entidade representa uma parte do minimundo que existe de forma independente, que pode ser física, como um aluno, ou apenas conceitual, como um departamento da escola.

O modelo relacional trata um BD como um conjunto de relações. Nesse caso, as **entidades**, representadas no modelo conceitual, são transformadas em **tabelas**, que, na terminologia formal do modelo relacional, são as **relações**. Os **atributos** das entidades passam a ser tratados por **colunas** dessas tabelas, também chamadas de **cabeçalhos**. E as **linhas** das tabelas, que armazenam os dados propriamente ditos, são chamadas de **tuplas**. Veja na Figura 3 a relação “Aluno”, na qual são especificadas as tuplas e os atributos.



Entre os atributos, são escolhidos identificadores de cada relação, os quais compõem as chaves primárias, quando internos, e as chaves externas ou estrangeiras, quando externos às relações. Quando as tabelas armazenarem dados, eles serão alocados nas linhas de acordo com as informações respectivas à coluna. Além disso, é necessário definir o tipo de dados que pode ser inserido em cada tupla; esse **tipo** é representado por um **domínio** de possíveis valores.



No caso da entidade “Professor”, seus atributos são: CPF, nome, endereço, sexo, salário, CPF do coordenador e código da disciplina. Dessa forma, a tabela “Professor” possui respectivamente as seguintes colunas: `Cpf`, `NomeProfessor`, `Endereco`, `Sexo`, `Salario` e `CodDisciplina`, tendo como código identificador e chave primária a coluna `Cpf`. Já a entidade “Disciplina” conta com: nome da disciplina (`NomeDisciplina`), código da disciplina (`CodDisc`) e CPF do coordenador (`CpfCoordenador`). Já “Turma” possui: nome (`TurmaNome`), código (`CodTurma`) e sala (`TurmaSala`). Neste exemplo, as chaves primárias são o CPF do professor, o código da disciplina e o código da turma.

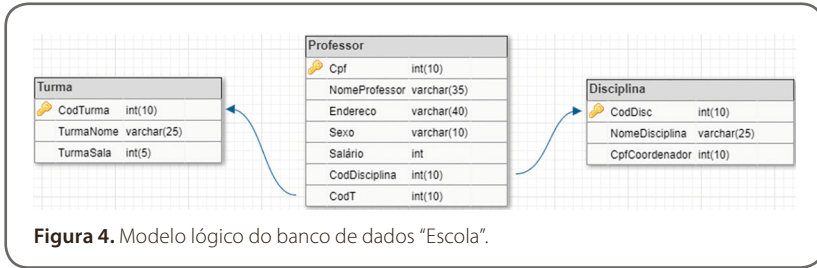
O esquema relacional das tabelas “Professor”, “Disciplina” e “Turma” pode ser descrito da seguinte forma, com a chave primária sublinhada:

```
Professor (Cpf, NomeProfessor, Endereco, Sexo, Salario,
CodDisciplina, CodT)
    CodDisciplina referencia Disciplina
    CodT referencia Turma
Disciplina (CodDisc, NomeDisciplina, CpfCoordenador)
Turma (CodTurma, TurmaNome, TurmaSala)
```

O que significa que a tabela “Professor” tem relação com as tabelas “Disciplina” e “Turma” por meio de suas chaves primárias, que, em cada uma das outras tabelas, são descritas de formas diferentes, mas remetem ao mesmo valor. É importante perceber essas relações, pois contribuem na construção do modelo seguinte, o físico.

Não se aconselha transcrever diretamente os nomes dos atributos para os nomes das colunas. É preciso ter em mente que esses nomes serão referenciados com frequência em sistemas; por isso, para diminuir o trabalho dos programadores, costuma-se utilizar nomes curtos. Da mesma forma, em SGBDs relacionais, os nomes das colunas não podem conter espaços em branco ou hífens — para evitá-los, utilizam-se abreviações (HEUSER, 2009).

Na Figura 4, é apresentado o modelo lógico da relação entre as entidades do exemplo anterior. Veja que foi necessário definir o tipo dos dados para os atributos. Nesse exemplo, foram utilizados os tipos *char*, em atributos cujos dados possuem tamanhos predefinidos, *varchar* para caracteres variáveis, *date* para datas e *int* para valores inteiros. Da mesma forma que o esquema de dados relacional, nesse modelo lógico, as relações entre as entidades são realizadas de acordo com as chaves primárias e estrangeiras.



Nesse caso, a entidade com relação por chave secundária depende da chave primária da outra entidade. Isso pode ser identificado no atributo "CodTurma", que é a chave primária da tabela "Turma" e se relaciona como secundária para a tabela "Professor". Na próxima seção, o modelo lógico do BD "Escola" será utilizado para a construção do modelo físico.

### 3 Modelo físico de banco de dados para MySQL

O projeto e o modelo físico do BD envolvem a escolha de estruturas específicas para armazenar os dados de arquivos e como realizar o acesso a esses dados, para que as aplicações que utilizem o BD apresentem bom desempenho. Esse desempenho depende diretamente do tamanho e do número de registros em cada arquivo. Por isso, é necessário estimar a quantidade de parâmetros em cada arquivo e o seu padrão de atualização e recuperação em cada transação (ELMASRI; NAVATHE, 2011).

Isso quer dizer que cabe ao DBA elaborar os modelos do BD, de acordo com o fluxo de utilização das tabelas, a fim de não sobrecarregar com muitos atributos uma única tabela, por exemplo. Ainda, cabe ao DBA pensar em como os relacionamentos entre as entidades ou tabelas do BD podem seguir de forma mais fluida, facilitando o seu tempo de acesso e de processamento.

Outro fator que deve ser levado em conta na modelagem do projeto físico do BD é a estimativa de crescimento de determinado arquivo, seja relacionada ao tamanho de seus registros, ao número de atributos ou ao número de registros. Isso resultará em uma determinação inicial dos caminhos de acesso e do armazenamento dos arquivos. É inicial pois, geralmente, é necessário realizar alterações no projeto, baseadas no seu desempenho, que pode ser observado somente após o BD ser implementado (ELMASRI; NAVATHE, 2011).

Dessa forma, o modelo físico é utilizado para especificar detalhes adicionais sobre o armazenamento do BD. Ele resume como as relações descritas nos modelos conceituais e lógicos serão armazenadas. Por isso, é utilizado para definir de que forma os arquivos serão armazenados e também para criar estruturas auxiliares de dados, os índices, que auxiliam na aceleração das operações de recuperação de dados (RAMAKRISHNAN; GEHRKE, 2011).

Agora, cabe voltar ao exemplo anterior, de criação de um projeto de BD chamado “Escola”. O MySQL Workbench, utilizado para a construção do modelo lógico, possibilita exportá-lo para o código MySQL, gerando o código necessário para a criação das tabelas, linhas e colunas, que é empregado no modelo físico. Posteriormente ele pode ser editado, para inclusão dos parâmetros adicionais necessários. Essa transformação acontece de acordo com os relacionamentos definidos ainda no modelo conceitual. O exemplo a seguir apresenta o código gerado a partir do modelo lógico do BD “Escola”, detalhado anteriormente na Figura 4.



### Exemplo

```
CREATE TABLE `Professor` (  
    `Cpf` int(10) NOT NULL,  
    `NomeProfessor` varchar(35) NOT NULL,  
    `Endereco` varchar(40) NOT NULL,  
    `Sexo` varchar(10) NOT NULL,  
    `Salário` int NOT NULL,  
    `CodDisciplina` int(10) NOT NULL,  
    `CodT` int(10) NOT NULL,  
    PRIMARY KEY (`Cpf`)  
);  
  
CREATE TABLE `Disciplina` (  
    `CodDisc` int(10) NOT NULL AUTO_INCREMENT,  
    `NomeDisciplina` varchar(25) NOT NULL,  
    `CpfCoordenador` int(10) NOT NULL,  
    PRIMARY KEY (`CodDisc`)  
);
```

```
CREATE TABLE `Turma` (  
    `CodTurma` int(10) NOT NULL AUTO_INCREMENT,  
    `TurmaNome` varchar(25) NOT NULL,  
    `TurmaSala` int(5) NOT NULL,  
    PRIMARY KEY (`CodTurma`)  
);  
  
ALTER TABLE `Professor` ADD CONSTRAINT `Professor_fk0` FOREIGN  
KEY (`CodDisciplina`) REFERENCES `Disciplina`(`CodDisc`);  
  
ALTER TABLE `Professor` ADD CONSTRAINT `Professor_fk1` FOREIGN  
KEY (`CodT`) REFERENCES `Turma`(`CodTurma`);
```

No exemplo, é apresentado o código utilizado na DDL do MySQL, que conta com diferentes comandos, entre eles `CREATE` e `ALTER TABLE`, que são utilizados para criar e alterar tabelas, respectivamente. Nesse caso específico, o `CREATE TABLE` aparece no começo da expressão para criar cada uma das tabelas, com seus respectivos atributos, acompanhados dos tipos de dados e, em alguns casos, do tamanho de cada campo, entre parênteses. Já o `ALTER TABLE` está sendo utilizado no código do exemplo para incluir as informações sobre as chaves estrangeiras (ELMASRI; NAVATHE, 2011).

Com as tabelas “Professor”, “Disciplina” e “Turma” criadas, cabe destacar possíveis consultas que podem ser obtidas com esse projeto de BD. A tabela que conta com mais atributos é a que armazenará os dados sobre os professores. A partir dela e de suas relações, é possível recuperar os dados pessoais de um professor específico, bem como as disciplinas e as turmas nas quais ele trabalha. Já quanto à tabela “Turma”, é possível verificar o nome da turma, o seu código e o número da sala na qual a turma é alocada, permitindo o acompanhamento por um professor ou coordenador, por exemplo.

Portanto, ao longo deste capítulo, você pode verificar que a construção de um projeto de BD é importante tanto para os desenvolvedores de sistemas quanto para os seus usuários. Assim, ele precisa ser adequado às necessidades dos usuários, mas também depende de um bom planejamento, para que todas as etapas se complementem e alcancem o resultado esperado.



### Saiba mais

A Secretaria Municipal de Educação (SME) de Duque de Caxias disponibiliza em seu *site* um material completo sobre a ferramenta MySQL. Para consultar o material, digite “SME duque de caxias manual MySQL” no seu mecanismo de busca e acesse o documento intitulado “MySQL Reference Manual”. No capítulo 2 do documento, são descritas as instruções básicas sobre a instalação do MySQL, o que é útil para a utilização de modelos físicos de BDs.



### Referências

COSTA, R. L. de C. *SQL: guia prático*. 2. ed. Rio de Janeiro: Brasport, 2006.

DATE, C. J. *Projeto de banco de dados e teoria relacional: formas normais e tudo o mais*. São Paulo: Novatec, 2015.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson, 2011.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2009. (Série Livros Didáticos Informática UFRGS).

MYSQL. *MySQL workbench*. [S. l.], 2020. Disponível em: <https://www.mysql.com/products/workbench/>. Acesso em: 27 maio 2020.

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: AMGH, 2011.



### Fique atento

Os *links* para *sites* da *web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS