

# **CONCEITOS BÁSICOS**

## **1. Conceitos básicos de BD, SBD e SGBD**

A importância da informação para a tomada de decisões nas organizações tem impulsionado o desenvolvimento dos sistemas de processamento de informações. Desta evolução, surgem alguns conceitos:

**BD (Banco de Dados):** é uma coleção de dados interligados, representando informações sobre um domínio específico. Exemplos: lista telefônica, acervo de uma biblioteca.

**SGBD (Sistema Gerenciador de Banco de Dados):** é um software com recursos específicos para facilitar a manipulação das informações de um BD e o desenvolvimento de programas aplicativos. Exemplos: Oracle, Paradox, MySQL, Access, Interbase, Sybase.

**SBD (Sistema de Banco de Dados):** é um sistema de manutenção de registros por computador envolvendo quatro componentes principais: dados, hardware, software e usuários.

Um SBD possui como objetivos isolar o usuário de detalhes mais internos do BD (abstração de dados) e prover independência de dados às aplicações (estrutura física de armazenamento e estratégia de acesso).

As vantagens da utilização de um SBD em relação aos sistemas tradicionais de gerenciamento de arquivos são:

- Rapidez na manipulação e no acesso à informação;
- Redução do esforço humano no desenvolvimento e utilização das aplicações;
- Disponibilização da informação no tempo necessário;
- Controle integrado de informações distribuídas fisicamente;
- Redução da redundância e de inconsistência de informações;
- Compartilhamento de dados;
- Aplicação automática de restrições de segurança;
- Redução de problemas de integridade.

### **SGBD x GA**

As diferenças entre o sistema tradicional de gerenciamento de dados, denominado Gerenciador de Arquivos (GA) e um SGBD são listadas a seguir. Se um sistema, candidato a SGBD, não contiver todas as características abaixo, este poderá ser, no máximo, um GA de alta qualidade. Quase um SGBD, mas não um SGBD. Um SGBD deve obedecer integralmente as seis regras. Caso contrário, será um GA.

**Regra 1: Auto-Contenção-** Um SGBD não contém apenas os dados em si, mas armazena completamente toda a descrição dos dados, seus relacionamentos e formas de acesso. Em um GA, em algum momento ao menos, os programas aplicativos declaram estruturas (algo que ocorre tipicamente em C, Pascal e COBOL), ou geram os relacionamentos entre os arquivos (típicos do ambiente xBase). Quando se define a forma do registro dentro do próprio programa, não está se lidando com um SGBD.

**Regra 2: Independência dos Dados-** Quando as aplicações estiverem realmente imunes as mudanças na estrutura de armazenamento ou na estratégia de acesso aos dados, pode-se dizer que esta regra foi atingida. Portanto, nenhuma definição dos dados deverá estar contida nos programas da aplicação. Quando a criação de uma nova forma de acesso ou um novo índice precisar ser feito dentro do código da aplicação, não está se lidando com um SGBD.

**Regra 3: Abstração dos Dados-** Em um SGBD real, é fornecida ao usuário somente uma representação conceitual dos dados, o que não inclui maiores detalhes sobre sua forma de armazenamento real. O chamado Modelo de Dados é um tipo de abstração utilizada para fornecer esta representação conceitual. Neste modelo, um esquema das tabelas, seus relacionamentos e suas chaves de acesso é exibido ao usuário, porém nada é afirmado sobre a criação dos índices, ou como serão mantidos, ou então quais as relações existentes entre as tabelas.

**Regra 4: Visões-** Um SGBD deve permitir que cada usuário visualize os dados de forma diferente daquela existente previamente no BD. Uma visão consiste de um subconjunto de dados do BD, mas não necessariamente estes deverão estar armazenados no BD. Portanto, uma replicação de uma estrutura, para fins de acesso de forma diferenciada por outros aplicativos, não caracteriza o uso de um SGBD.

**Regra 5: Transações-** Um SGBD deve gerenciar completamente a integridade referencial definida em seu esquema, sem precisar, em tempo algum, do auxílio do programa aplicativo. Desta forma exige-se que o BD tenha ao menos instruções que permitam a gravação e o cancelamento de uma série de modificações simultâneas. Por exemplo: um cadastro de pedido de um cliente o qual deseja reservar cinco itens do estoque. Se existir algum bloqueio financeiro deste cliente (duplicatas em atraso) que impeçam a venda, a transação deverá ser desfeita com apenas uma instrução ao BD, sem quaisquer modificações suplementares nos dados. Qualquer acesso complementar para a correção da reserva não caracteriza a utilização de um SGBD.

**Regra 6: Acesso Automático-** Num GA, uma situação típica é o chamado *dead-lock*, o abraço mortal. Esta situação indesejável pode ocorrer toda vez que um usuário travou um registro em uma tabela e, seu próximo passo, será travar um registro em uma tabela relacionada à primeira. Porém, se este registro estiver previamente travado por outro usuário, o primeiro usuário ficará paralisado, pois, estará esperando o segundo usuário liberar o registro em uso, para que então possa travá-lo e prosseguir sua tarefa. Se, por hipótese, o segundo usuário necessitar travar o registro travado pelo primeiro usuário, diz-se que ocorreu um *dead-lock*, pois cada usuário travou um registro e precisa travar um outro, justamente o registro anteriormente travado pelo outro. Por exemplo: a pessoa responsável pelos pedidos acabou de travar o registro Item de Pedido, e necessita travar um registro no Cadastro de Produtos, para indicar uma nova reserva. Se, concomitantemente, estiver sendo realizada uma tarefa de atualização de pendências na tabela de Itens, e para tanto, previamente este segundo usuário

travou a tabela de Produtos, ocorre o *dead-lock*. Se a responsabilidade de evitar esta ocorrência for da aplicação, não está se lidando com um SGBD.

## 2. Abstração de dados

O sistema de banco de dados (SBD) deve prover uma visão abstrata de dados para os usuários, isolando, desta forma, detalhes mais internos do BD. A abstração se dá em três níveis:

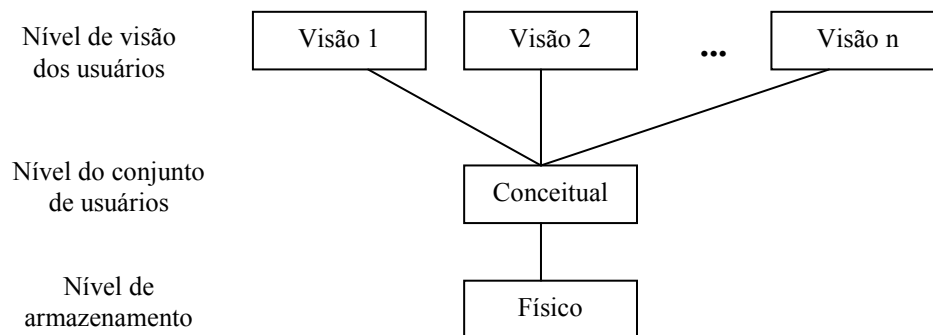


FIGURA 1.1 – NÍVEIS DE ABSTRAÇÃO

**Nível físico:** também chamado de “Esquema interno”, é o nível mais baixo de abstração. Descreve como os dados estão realmente armazenados, englobando estruturas complexas de baixo nível.

**Nível conceitual (ou lógico):** conhecido também como “Esquema Conceitual”, descreve quais os dados estão armazenados e seus relacionamentos. Neste nível, o BD é descrito através de estruturas relativamente simples, que podem envolver estruturas complexas no nível físico.

**Nível de visões do usuário:** é o nível externo, descrevendo partes do BD que serão visualizadas pelos usuários de acordo com suas necessidades. Uma visão é um subconjunto de dados do BD, sem que exista a necessidade de estarem armazenados no BD.

Esta arquitetura de três níveis, além de prover a abstração de dados, provê também a independência lógica e física dos dados. Uma independência lógica possui a capacidade de mudar o esquema conceitual sem a necessidade de modificar programas da aplicação e esquemas externos, enquanto que a física tem a capacidade de mudar o esquema interno sem a necessidade de alterar os esquemas conceitual e externo. O acréscimo de uma informação num esquema conceitual é um exemplo de independência lógica e a reorganização física dos arquivos e a criação de estruturas de acesso adicionais são exemplos de independência física.

### 3. Modelos lógicos de dados

É o conjunto de ferramentas conceituais para a descrição dos dados, dos relacionamentos entre os mesmos e das restrições de consistência e integridade. Em outras palavras, é o conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados (tipos de dados, relacionamentos e restrições que devem ser mantidas sobre os dados).

Os modelos dividem-se em dois grupos: baseados em objetos e baseados em registros. No modelo orientado a objetos, o código executável é parte integrante do modelo de dados enquanto que no orientado a registros, o banco é estruturado em registros de formato fixo, onde cada registro tem sua coleção de atributos. Além disto, o modelo baseado em registro possuem linguagens para expressar consultas e atualizações no BD. São exemplos de modelos baseados em registro os modelos de Redes, Hierárquico e Relacional.

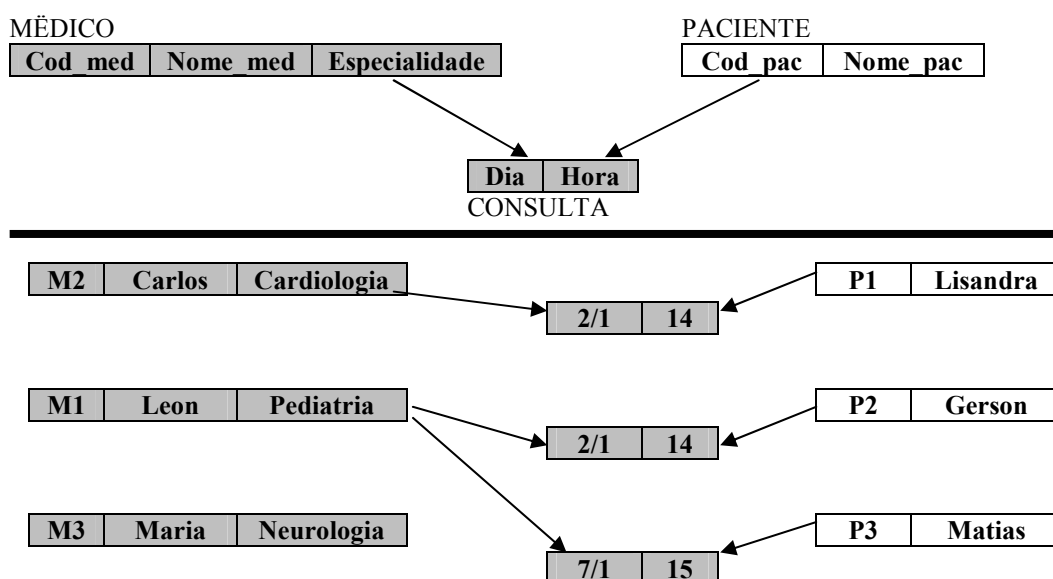
Em termos de evolução, o primeiro modelo utilizado foi o de **Redes** (1964) e, em seguida, o **Hierárquico** (1965-66), ambos fortemente influenciados por características físicas do banco de dados. O **Relacional**, proposto em 1970, esteve disponível comercialmente em meados dos anos 80.

#### Modelo de Redes

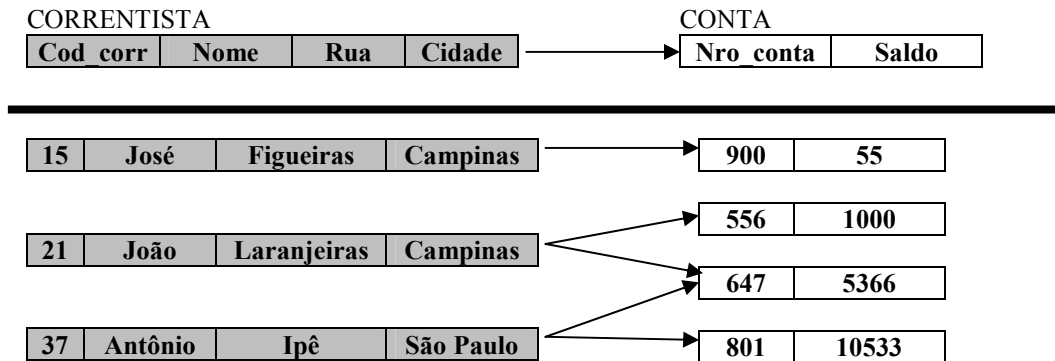
Inicialmente apresentado no relatório do grupo de trabalho CODASYL (*Conference on Data Systems Language*), em 1971, sendo chamado de modelo DBTG (*Data Base Task Group*). Foi revisado em 1978 e 1981.

Apresenta como características os dados organizados em vários tipos de registros e tipos de *sets* (estabelecem os relacionamentos) e um problema de estratégia de acesso aos registros "filhos".

Exemplo 1



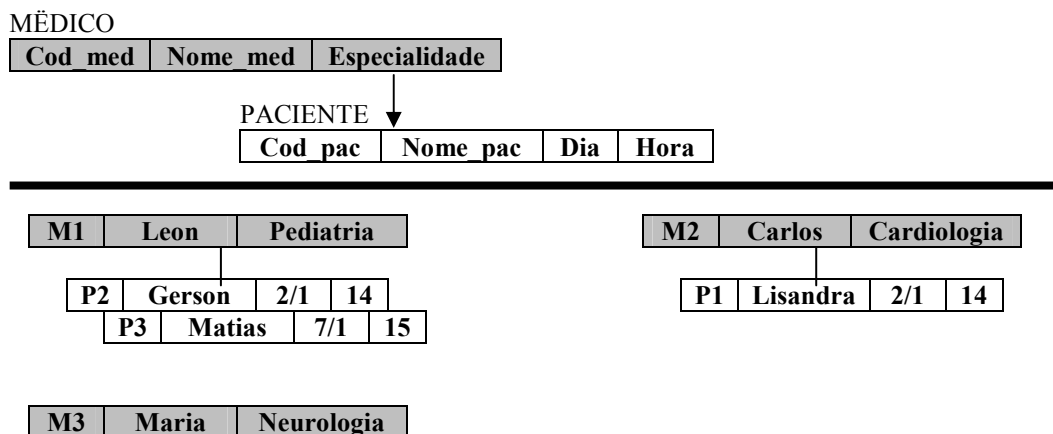
## Exemplo 2



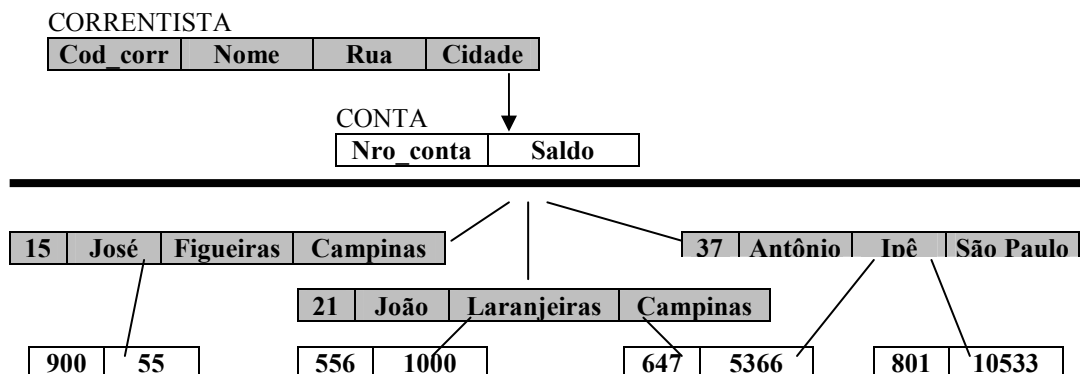
## Modelo Hierárquico

Da mesma forma que o modelo de redes, no modelo hierárquico os dados são organizados em vários tipos de registros e o relacionamento dos registros "pai" e "filho" são explícitos. Apresenta como vantagens a simplicidade de processamento e o uso natural para organizações hierárquicas de dados.

## Exemplo 1



## Exemplo 2



**Modelo Relacional**

Introduzido por E.F. Codd, em 1970, possui como características: base de dados visualizada como um conjunto de tabelas, cada uma representando uma relação; relacionamentos representados por valores de dados; simetria nas consultas; várias linguagens definidas (álgebra relacional, DDL e DML em SQL). É uma desvantagem do modelo relacional a representação não natural de objetos complexos.

**Exemplo 1****MÉDICO**

Cod_med	Nome_med	Especialidade
---------	----------	---------------

**PACIENTE**

Cod_pac	Nome_pac
---------	----------

**CONSULTA**

Cod_med	Cod_pac	Dia	Hora
---------	---------	-----	------

---

**MÉDICO**

Cod_med	Nome_med	Especialidade
M1	Leon	Pediatria
M2	Carlos	Cardiologia
M3	Maria	Neurologia

**PACIENTE**

Cod_pac	Nome_pac
P1	Lisandra
P2	Gerson
P3	Matias

**CONSULTA**

Cod_med	Cod_pac	Dia	Hora
M1	P2	2/1	14
M1	P3	7/1	15
M2	P1	2/1	14

**Exemplo 2****CORRENTISTA**

Cod_corr	Nome	Rua	Cidade
----------	------	-----	--------

**CORRENTISTA-CONTA**

Cod_corr	Nro_conta
----------	-----------

**CONTA**

Nro_conta	Saldo
-----------	-------

---

**CORRENTISTA**

Cod_corr	Nome	Rua	Cidade
15	José	Figueiras	Campinas
21	João	Laranjeiras	Campinas
37	Antônio	Ipê	São Paulo

**CONTA**

Nro_conta	Saldo
900	55
556	1000
647	5366
801	10533

**CORRENTISTA-CONTA**

Cod_corr	Nro_conta
15	900
21	556
21	647
37	647
37	801

## 4. DDL e DML

O esquema de um banco de dados é a estrutura geral do BD. Sua estrutura não muda com frequência e há um esquema para cada nível de abstração e um sub-esquema a cada visão do usuário. Um esquema pode ser representado de uma forma **textual** ou **gráfica**. Normalmente, utiliza-se o Diagrama de Entidade Relacionamento (descrito mais adiante) para a representação gráfica do esquema de um BD. São exemplos de esquemas apresentados na forma textual:

Medico (Cod\_med, Nnome\_med, Especialidade)  
Paciente (Cod\_pac, Nome\_pac)  
Consulta (Cod\_med, Cod\_pac, Dia, Hora)

Os modelos de dados baseados em registros possuem linguagens para expressar consultas e atualizações no BD, chamadas de Linguagens de SGBD. Algumas delas atuam sobre o esquema do BD, outras sobre os dados armazenados. Abaixo, uma síntese destas linguagens:

### **DDL (*Data Definition Language* ou Linguagem de Definição de Dados)**

Permite especificar o esquema do BD, através de um conjunto de definições de dados. O esquema do BD fica armazenado numa área do SGBD chamada de Dicionário de Dados. Na maioria das vezes, uma DDL atua nos esquemas conceitual e interno.

### **DML (*Data Manipulation Language* ou Linguagem de Manipulação de Dados)**

Em relação a BD, a manipulação dos dados dá-se pelas seguintes operações:

- Recuperação da informação armazenada. Esta recuperação é no sentido de tornar a informação armazenada no banco disponível ao usuário e/ou programa aplicativo;
- Inserção de novas informações;
- Exclusão de informações;
- Modificação de dados armazenados.

Uma DML permite ao usuário acessar ou manipular os dados, vendo-os da forma como são definidos no nível de abstração mais alto do modelo de dados utilizado. No nível mais alto, no de visões de usuário, uma DML é: não-procedural, declarativa, orientada a conjuntos e dita linguagem de consulta (*query language*). Em baixo-nível, no físico, uma DML é orientada a registros e considerada uma sublinguagem de dados, embutida em uma linguagem hospedeira.

A linguagem SQL (*Structured Query Language*) é um bom exemplo, pois contempla uma DDL e uma DML.

## 5. Papéis humanos em SBD

As pessoas envolvidas num sistema de banco de dados podem ser divididas em usuário e administrador. A um usuário, cabe a tarefa de realizar operações de manipulação de dados dum BD, enquanto que um administrador fica responsável pelo controle o SBD.

Usuários	Atuação
Programador de aplicação	Desenvolve programas aplicativos que manipulam banco de dados
Usuário sofisticado	Capaz de manipular dados num BD sem o uso de aplicativos utilizando as linguagens de consulta
Usuário "ingênuo"	Manipula BD somente pelas interfaces definidas nos programas aplicativos

Administradores	Atuação
Administrador de dados	Define e atualiza do esquema do BD
Administrador do SGBD	<ul style="list-style-type: none"><li>- Define estrutura de armazenamento e estratégia de acesso;</li><li>- Concede autorização de acesso a dados;</li><li>- Define controles de integridade;</li><li>- Define estratégias para <i>backup</i>;</li><li>- Monitora desempenho.</li></ul>

QUADRO 1.1 – PAPÉIS NUM SBD

## 6. Estrutura geral de um SGBD

Um Sistema Gerenciador de Banco de Dados é um módulo de programa que fornece a interface entre dados de baixo nível armazenados num banco de dados e os programas aplicativos ou as solicitações submetidas ao sistema. O SGBD é o responsável por todo o acesso aos dados armazenados. O papel da gerência do BD, de forma conceitual, passa pelas etapas:

- O usuário emite uma solicitação de acesso;
- O SGBD intercepta a solicitação e a analisa;
- O SGBD inspeciona os esquemas externos (ou sub-esquemas) relacionados àquele usuário, os mapeamentos entre os três níveis e a definição da estrutura de armazenamento;
- O SGBD realiza as operações solicitadas no BD.

As principais tarefas de um SGBD são:

- Interação com o sistema de arquivos do sistema operacional;
- Cumprimento da integridade;
- Cumprimento da segurança;
- Cópias de segurança (*backup*) e recuperação;
- Controle da concorrência.



O quadro abaixo apresenta as classificações de um SGBD segundo vários critérios:

Critério	Classificação
Modelo de dados	<ul style="list-style-type: none"> <li>- Rede</li> <li>- Hierárquico</li> <li>- Relacional</li> </ul>
Número de usuários	<ul style="list-style-type: none"> <li>- Monousuário</li> <li>- Multiusuário</li> </ul>
Localização da base de dados	<ul style="list-style-type: none"> <li>- Centralizado</li> <li>- Distribuído (SGBDD)               <ul style="list-style-type: none"> <li>• Homogêneo</li> <li>• Heterogêneo (BD Federados)</li> </ul> </li> </ul>

QUADRO 1.2 – CLASSIFICAÇÕES DE SGBD

De um modo geral, um Sistema de Banco de Dados apresenta a seguinte estrutura:

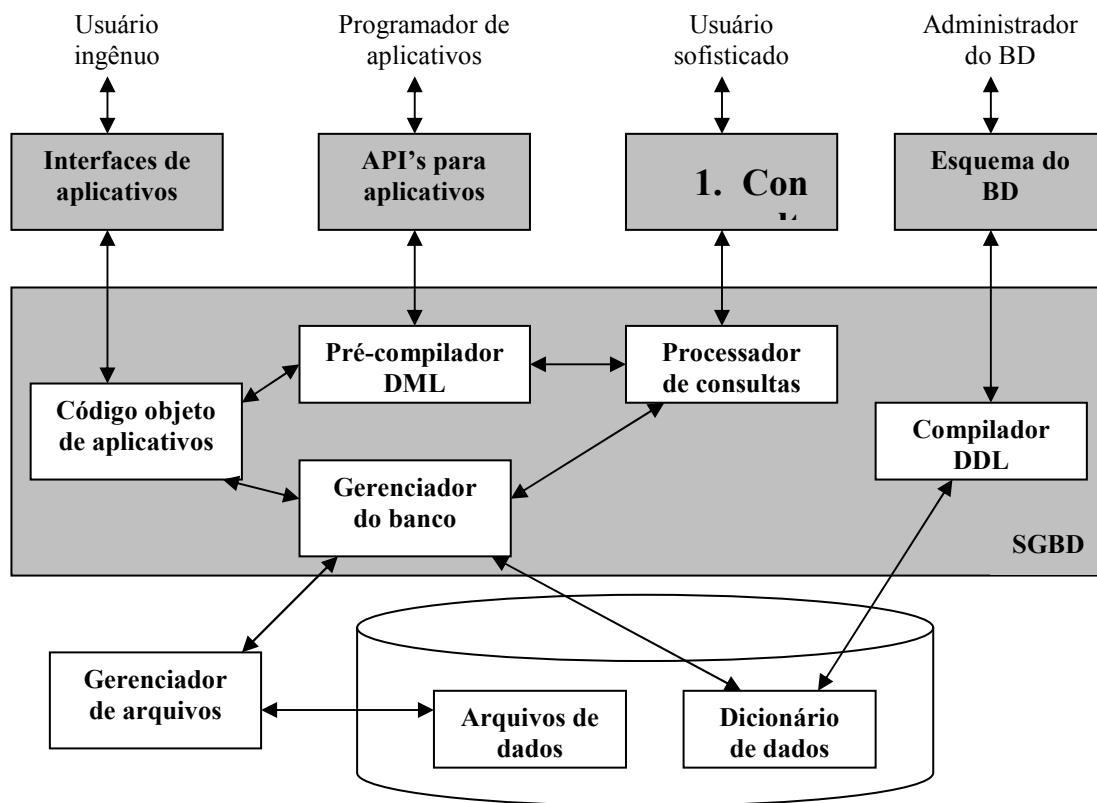


FIGURA 1.2 –ESTRUTURA DE UM SGBD