

# Módulo 9

Banco de Dados



## Lição 1

Introdução a Sistemas de Bancos de Dados

*Versão 1.0 - Fev/2009*

**Autor**

Ma. Rowena C. Solamo

**Equipe**

Rommel Feria

Rick Hillegas

John Paul Petines

**Necessidades para os Exercícios****Sistemas Operacionais Suportados****NetBeans IDE 5.5** para os seguintes sistemas operacionais:

- Microsoft Windows XP Professional SP2 ou superior
- Mac OS X 10.4.5 ou superior
- Red Hat Fedora Core 3
- Solaris™ 10 Operating System (SPARC® e x86/x64 Platform Edition)

**NetBeans Enterprise Pack**, poderá ser executado nas seguintes plataformas:

- Microsoft Windows 2000 Professional SP4
- Solaris™ 8 OS (SPARC e x86/x64 Platform Edition) e Solaris 9 OS (SPARC e x86/x64 Platform Edition)
- Várias outras distribuições Linux

**Configuração Mínima de Hardware****Nota:** IDE NetBeans com resolução de tela em 1024x768 pixel

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	500 MHz Intel Pentium III workstation ou equivalente	512 MB	850 MB
Linux	500 MHz Intel Pentium III workstation ou equivalente	512 MB	450 MB
Solaris OS (SPARC)	UltraSPARC II 450 MHz	512 MB	450 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Série 1.8 GHz	512 MB	450 MB
Mac OS X	PowerPC G4	512 MB	450 MB

**Configuração Recomendada de Hardware**

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	1 GB
Linux	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	850 MB
Solaris OS (SPARC)	UltraSPARC IIIi 1 GHz	1 GB	850 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Series 1.8 GHz	1 GB	850 MB
Mac OS X	PowerPC G5	1 GB	850 MB

**Requerimentos de Software**

NetBeans Enterprise Pack 5.5 executando sobre Java 2 Platform Standard Edition Development Kit 5.0 ou superior (JDK 5.0, versão 1.5.0\_01 ou superior), contemplando a Java Runtime Environment, ferramentas de desenvolvimento para compilar, depurar, e executar aplicações escritas em linguagem Java. Sun Java System Application Server Platform Edition 9.

- Para **Solaris, Windows, e Linux**, os arquivos da JDK podem ser obtidos para sua plataforma em <http://java.sun.com/j2se/1.5.0/download.html>
- Para **Mac OS X**, Java 2 Platform Standard Edition (J2SE) 5.0 Release 4, pode ser obtida diretamente da Apple's Developer Connection, no endereço: <http://developer.apple.com/java> (é necessário registrar o download da JDK).

Para mais informações: <http://www.netbeans.org/community/releases/60/relnotes.htm>

**Java™ DB System Requirements**

Java™ DB is supported on the Solaris, Linux and Windows operating systems and Sun Java 1.4 or later.

**Colaboradores que auxiliaram no processo de tradução e revisão**

Aécio Júnior	Carlos Hilner Ferreira Costa	Kleberth Bezerra Galvão dos Santos
Alberto Ivo da Costa Vieira	Daniel Noto Paiva	Luiz Fernandes de Oliveira Junior
Alexandre Mori	Daniel Wildt	Maria Carolina Ferreira da Silva
Alexis da Rocha Silva	Denis Mitsuo Nakasaki	Maricy Caregnato
Aline Sabbatini da Silva Alves	Fábio Antonio Ferreira	Maurício da Silva Marinho
Allan Wojcik da Silva	Givailson de Souza Neves	Paulo Oliveira Sampaio Reis
Angelo de Oliveira	Jacqueline Susann Barbosa	Ronie Dotzlaw
Aurélio Soares Neto	Jader de Carvalho Belarmino	Seire Pareja
Bruno da Silva Bonfim	João Vianney Barrozo Costa	Sergio Terzella
Carlos Fernando Gonçalves	José Francisco Baronio da Costa	Thiago Magela Rodrigues Dias

**Auxiliadores especiais**

Revisão Geral do texto para os seguintes Países:

- **Brasil** – Tiago Flach
- **Guiné Bissau** – Alfredo Cá, Bunene Sisse e Buon Olossato Quebi – ONG Asas de Socorro

**Coordenação do DFJUG**

- **Daniel deOliveira** – JUGLeader responsável pelos acordos de parcerias
- **Luci Campos** - Idealizadora do DFJUG responsável pelo apoio social
- **Fernando Anselmo** - Coordenador responsável pelo processo de tradução e revisão, disponibilização dos materiais e inserção de novos módulos
- **Rodrigo Nunes** - Coordenador responsável pela parte multimídia
- **Sérgio Gomes Veloso** - Coordenador responsável pelo ambiente JEDI™ (Moodle)

**Agradecimento Especial**

**John Paul Petines** – Criador da Iniciativa JEDI™

**Rommel Feria** – Criador da Iniciativa JEDI™

# 1. Objetivos

Nesta lição veremos uma introdução a sistemas de bancos de dados. A primeira parte descreve o ambiente de banco de dados utilizando o conceito de Gerenciamento de Recursos de Informações ou IRM (*Information Resource Management*) em que as empresas tem necessidade de gerenciar seus dados e informações.

A segunda seção discute o processo de desenvolvimento de bases de dados utilizando o framework a Arquitetura de Sistema de Informação ou ISA (*Information System Architecture*). Isto ajudará a entender o lugar do banco de dados no processo de desenvolvimento e manutenção de sistemas de informação em uma empresa.

Ao final desta lição, o estudante será capaz de:

- Descrever o ambiente de banco de dados utilizando o conceito de Gerenciamento de Recursos de Informação
- Descrever a Arquitetura de Sistemas de Informação como um *framework* para construção de sistemas de informação
- Entender a importância do banco de dados dentro do desenvolvimento geral de sistemas de informação
- Discutir os processos de análise, design e implementação de um banco de dados

## 2. Ambiente de Bases de Dados

Com o passar dos anos, as empresas perceberam a importância do dado e da informação em suas operações diárias. IRM (Gerenciamento de Recursos de Informação) é o conceito de que a informação é um recurso corporativo muito importante e deve ser gerenciado utilizando alguns princípios básicos que são utilizados para gerenciar outros ativos da companhia como pessoal, equipamento e recursos financeiros.

A seguir vemos os princípios básicos do qual IRM é derivado:

1. Empresas utilizam recursos que fluem em seu ambiente
2. Meio ambiente. Provê recursos retornando ele para seu ambiente de origem
3. Existem dois tipos básicos de recursos para serem gerenciados, especificamente:
  - Recursos Físicos que são, por exemplo: pessoal, materiais, máquinas, entre outros
  - Recursos Conceituais como dados e informações
4. Com o crescimento das operações organizacionais, torna-se difícil gerenciar os recursos físicos utilizando observações. Portanto, gerentes de negócios são forçados a depender de recursos conceituais
5. Os mesmos princípios básicos utilizados para o gerenciamento de recursos físicos podem ser utilizados para gerenciar os recursos conceituais
6. O gerenciamento de dados e informações, envolve:
  - Aquisição de dados e informações antes que sejam necessárias
  - Medidas de segurança para proteger recursos contra invasão, uso indevido e destruição
  - Garantia de qualidade
  - Procedimentos de liberação de recursos quando não são mais necessários à organização
7. Compromisso organizacional é necessário para gerenciar os dados e informações.

Para implementar IRM, as seguintes funcionalidades são consideradas:

1. Gerenciamento de operações, tais como: agendamento, planejamento de capacidade, segurança de operações e desastres de recuperação de dados e informações
2. Garantia de qualidade para garantir que informações sejam fornecidas quando necessário
3. Gerenciamento de comunicações de LAN ou WAN
4. Gerenciamento de recursos de dados tais como análise de dados, design de bancos de dados, administração de dados e administração do banco de dados
5. Gerenciamento de projeto
6. Planejamento de Sistemas de Informação corporativos
7. Desenvolvimento e manutenção de sistemas

Dados são fatos acerca de pessoas, objetos e eventos. Informação, por outro lado, é o dado que foi devidamente processado e apresentado num formulário para a interpretação humana, freqüentemente com o propósito de revelar tendências e padrões. Existem 5 tarefas envolvidas na conversão de dados em informações. São eles:

1. Aquisição
2. Armazenamento
3. Manipulação
4. Recuperação

## 5. Distribuição

Para dar suporte aos 5 itens citados acima é necessária a utilização de uma base de dados. Um banco de dados é uma coleção compartilhada de dados lógicos relacionados, de tal forma a prover informação para múltiplos usuários em uma organização. Existem duas arquiteturas genéricas de bancos de dados: bancos de dados centralizados e bancos de dados distribuídos.

### 2.1. Banco de Dados Centralizado

Em um banco de dados centralizado, os dados estão armazenados em um único local. Estes dados são acessados por meio de equipamentos de comunicação. Fornecem um mecanismo de controle de acesso e atualização dos dados melhor do que os bancos de dados distribuídos, porém são mais vulneráveis a falhas já que dependem da disponibilidade de recursos de um único local. Três exemplos de bancos de dados centralizados são discutidos logo abaixo:

1. **Banco de Dados de Computadores Pessoais.** Neste ambiente o banco é utilizado por um único usuário. Este usuário cria o banco de dados, atualiza e mantém os dados, produz os dados e gera relatórios ou informação com base nestes dados. Normalmente o banco de dados está alocado em um computador pessoal onde um ou um número limitado de aplicações acessam este banco. Este tipo de banco de dados pode ser encontrado em pequenos negócios. Um exemplo de aplicação pode ser o gerenciamento de um estoque.
2. **Bancos de Dados em Computadores Centrais.** Neste ambiente o banco de dados é normalmente localizado numa máquina central e é chamado de host. Os dados são acessados por terminais e meios de comunicações de dados. O computador é normalmente um mainframe ou servidor encontrado em grandes negócios que necessitam de um intenso acesso aos dados por um grande número de usuários. Típicas aplicações podem ser: sistemas de reservas aéreas, instituições financeiras e companhias de entrega.
3. **Bancos de Dados Cliente/Servidor.** Neste ambiente a arquitetura cliente-servidor é utilizada, onde muitos clientes podem compartilhar o serviço de um único servidor. O servidor é um software de aplicação que provê serviços (chamados de funções back-end, tais como impressões, gerenciamento de bancos de dados ou de arquivos, gerenciamento de comunicações, entre outros) para os clientes que os requisitam. Um cliente (que fornece as funções de *front-end*) é um software de aplicações que requisita o serviço de um ou mais servidores. O ponto forte de uma arquitetura cliente-servidor é permitir que a aplicação do cliente acesse dados gerenciados pelo servidor.

### 2.2. Bancos de Dados Distribuído

Um banco de dados distribuído é representado por um único banco de dados lógico que é separado fisicamente em vários computadores que podem estar situados em vários locais. Existem duas categorias genéricas.

1. **Bancos de dados homogêneos.** A tecnologia de bancos de dados utilizada é a mesma ou pelo menos compatível em todos os locais. Certas condições devem ser respeitadas antes de um banco de dados ser considerado homogêneo:
  - O sistema operacional utilizado em cada uma das localidades deve ser o mesmo ou pelo menos um que seja altamente compatível
  - O modelo de dados utilizado em cada uma das localidades deve ser o mesmo
  - O sistema gerenciador de banco de dados utilizado em cada uma das localidades deve ser o mesmo ou pelo menos que sejam altamente compatíveis entre si
  - Os dados em suas várias localidades devem possuir definições e formatos comuns
2. **Bancos de dados Heterogêneos.** A tecnologia utilizada varia e pode não ser a mesma em todas as localidades. Um banco de dados pode utilizar uma tecnologia de gerenciamento relacional enquanto que outro utilize arquivos convencionais ou antigos bancos de dados hierárquicos. Estes bancos de dados ficam conectados.

### 3. Arquitetura de Sistemas de Informação (ISA)

Para desenvolver sistemas de informação, um *framework* é utilizado para fornecer uma base para o planejamento estratégico, desenvolvimento e utilização de sistemas de informação que dão suporte a visão geral dos objetivos da organização. A arquitetura de sistemas de informação é um exemplo de *framework*. Representa um modelo conceitual ou plano que ilustra a estrutura dos sistemas de informação que são necessários para a organização. Fornece a base para planejamentos estratégicos e comunicações para a direção de toda a tecnologia da informação e o contexto principal para a tomada de decisão em uma área. A tabela 1 descreve este *framework*.

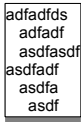
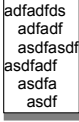
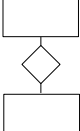

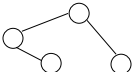
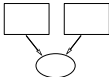
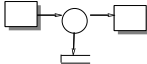
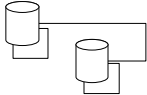


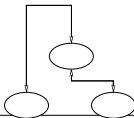
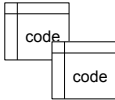
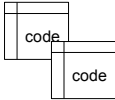
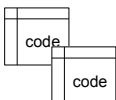
		Dado	Processos	Rede
1	Escopo do negócio	Lista as entidades importantes de um negócio 	Lista as funções que o negócio deve realizar 	Lista de localidades em que o negócio opera
2	Modelo do negócio	Entidade de negócio e seus inter-relacionamentos 	Decomposição de funções e processos 	Links de comunicação entre as localidades do negócio 
3	Modelo do sistema de informação	Modelo de negócios e seus inter-relacionamentos 	Fluxos entre os processos 	Distribuição em redes 
4	Modelo de tecnologia	Design do banco de dados 	Especificação dos processos 	Design de configuração 
5	Definição da tecnologia	Definição da estrutura e subestrutura do banco de dados 	Códigos do programa e blocos de código 	Definição de configuração 
6	Sistema de informação	Dados e informações	Aplicações	Configuração do sistema

Tabela 1: Arquitetura de Sistemas de Informação

Os três componentes principais neste *framework* são dados, processos e rede. Estes são as três colunas da tabela.

1. **Dados** consistem de entidades de dados e o relacionamento entre si. Representam o “o quê?” em um sistema de informação. Este é o componente com o qual bancos de dados são construídos.
2. **Processos** são seqüências de passos que convertem entradas em saídas (ou dados em informação). Representam o “como” em um sistema de informação.

3. **Rede** descreve a localização onde os dados são armazenados, onde os processos são realizados bem como a conexão entre as localidades. Representam o “onde” em um sistema de informação.

Cada linha representa uma camada arquitetural na construção de sistemas de informação de uma organização. Fornecem as 6 regras e perspectivas de cada camada.

1. **Escopo do Negócio** – provê uma visão geral da estratégia dos sistemas de informação de uma organização. Define o escopo, missão e direção do negócio e quais sistemas de informação lhes dá suporte. Uma lista de entidades importantes, funções necessárias para que o sistema possa realizar aquilo que deve ser feito e identificar onde o negócio da empresa é realizado. Os proprietários do negócio são responsáveis por definir o escopo, missão e direção do negócio
2. **Modelo de Negócio** – desenvolve os modelos que representam o escopo do negócio, missões e direções que o negócio irá tomar. Nesta camada, as entidades do negócio e seus inter-relacionamentos são definidos. A decomposição de funções e processos são identificadas. É definida as ligações entre as localidades de negócios. O arquiteto do sistema de informação é a pessoa que desenvolve estes modelos
3. **Modelo do Sistema de Informação** – desenvolve o modelo da informação que dá suporte ao negócio da organização. Nesta camada, os dados e seus relacionamentos são modelados em detalhes. Os fluxos entre as aplicações também são processados e definidos. A distribuição pela rede também é identificada. O designer é responsável pelo desenvolvimento do modelo da informação
4. **Modelo de Tecnologia** – converte o modelo do sistema de informação em um design que possa se adaptar às características e especificações da tecnologia. O projeto de um banco de dados, especificação de processos e configurações de rede são criados. O construtor desenvolve o design do sistema de informação
5. **Definição da Tecnologia** – converte os modelos de tecnologia em declarações para gerar o sistema de informação. O projeto do banco de dados é traduzido em esquema e sub-esquemas do banco de dados. As especificações de processo são codificadas como códigos de programa e blocos de controle. Os projetos de configuração da rede são traduzidos em definição de configuração. O contratante traduz modelos de tecnologia em códigos
6. **Sistema de Informação** – gerencia, utiliza e opera o sistema de informação como um todo. Neste ponto, os usuários do sistema utilizam os dados e informações através de aplicativos especificados na configuração de sistema

Para melhor utilizar este *framework*, duas regras simples são empregadas:

1. Cada processo é mapeado para o dado que o utiliza. Ambos, dados e processos, são mapeados para as localizações na rede ou objetos onde serão distribuídos. Isto ajuda na garantia de que os vários componentes serão integrados aos demais
2. A transformação de dados, processos e rede ocorrem simultaneamente de uma linha para a próxima. Esta regra evita inconsistência e retrabalho

Este curso irá se concentrar no COMPONENTE DE DADOS do *framework* da Arquitetura do Sistema de Informação (ISA) na construção de um banco de dados.

## 4. Metodologia de Engenharia da Informação

O *framework* ISA fornece um contexto de desenvolvimento e integração do sistema de informação. Sugere o tipo de processo no desenvolvimento de modelos em cada uma das camadas da arquitetura. Entretanto, não fornece uma forma para o desenvolvimento destes modelos. Portanto, uma organização deve utilizar uma ou mais metodologias e um conjunto de ferramentas de modelagem para desenvolver a representação arquitetural requisitada em cada perspectiva.

Uma metodologia define o “como” ou conjunto de passos para se realizar determinado objetivo,



juntamente com um conjunto de objetos de design que são manipulados para auxiliar o processo. Deve fornecer regras para ajudar a garantir que um conjunto consistente de padrões e procedimentos serão utilizados durante todo o processo de desenvolvimento e o sistema resultante se adequar aos objetivos especificados.

Uma variedade de ferramentas de modelagem, manuais ou automatizadas, são necessárias para auxiliar no desenvolvimento do sistema de informação.

1. **Engenharia de Software Auxiliada por Computador (CASE)** – São produtos de software que fornecem suporte automatizado para algumas partes do processo de desenvolvimento do sistema.
2. **CASE Integrado (I-CASE)** – Conjunto de ferramentas CASE que podem dar suporte a todas as fases do processo de desenvolvimento de um software.

Engenharia da Informação é uma metodologia formal que é utilizada na criação e manutenção de sistemas de informação. É um padrão *top-down* que se inicia com os modelos de negócio. A partir destes modelos, os modelos de dados e modelos de processos são derivados e então é criado o modelo de negócio. Porque utilizar Engenharia da Informação?

1. Desenvolvido com uma visão baseado na empresa que permite que a organização desenvolva um sistema de informação integrado
2. Baseado em dados ao invés de processos. Um modelo baseado em dados utiliza as seguintes etapas:
  - Identificação de entidades ou coisas que a organização deve gerenciar
  - Identificação de atributos, propriedades e características das entidades
  - Identificação dos relacionamentos entre as entidades
  - Identificação de regras de negócio que governam como as entidades são gerenciadas e utilizadas
  - Desenvolver o aplicativo ou programa baseado em como os dados estão sendo gerenciados e utilizados

Um modelo baseado em processos utiliza as seguintes etapas:

- Identificação e análise dos processos organizacionais
- Modelagem dos fluxos de dados entre os processos
- Especificação das entradas e saídas de dados
- Especificação da lógica necessária para converter dados de entrada em dados de saída
- Design dos arquivos de dados

Designers de Sistemas descobriram que um equilíbrio entre modelo baseado em dados e modelo baseado em processos são normalmente o mais apropriado.

3. É compatível com o *framework* ISA

A Metodologia da Engenharia de Informação é dividida em fases que podem ser mapeadas para o *framework* ISA. São normalmente chamadas de: fase de planejamento, fase de análise, fase de projeto e fase de implementação.

#### **4.1. Fase de Planejamento**

O objetivo desta fase é alinhar a tecnologia da informação às estratégias do negócio de uma organização. Para isto é necessário que haja uma cooperação entre os gerentes do negócio e os gerentes do sistema de informação. Existem três grandes passos na fase de planejamento. São eles:

1. Identificação de fatores de planejamento de estratégia que incluam os objetivos do negócio, fatores críticos de sucesso e problemas das áreas do negócio.
2. Identificação de objetos de planejamento corporativo. Estes objetos são:
  - Unidades da organização que consistem de vários departamentos ou outros

- componentes da organização
  - Localizações que mostrem os componentes organizacionais em mais de um único lugar
  - Funções de negócio relacionados a grupos de processos de negócio que dão suporte a alguns aspectos da missão da empresa e que não são os mesmos da unidade organizacional
  - Tipos de entidades
3. Desenvolver os modelos da empresa utilizando as técnicas e ferramentas de modelagem a seguir:
- Técnica de decomposição funcional que consiste na quebra das funções de uma organização em busca de níveis de detalhes cada vez maiores, utilizando o **Diagrama de Fluxo de Dados** para modelar as funções ou processos
  - Técnica de análise situacional que consiste no processo de análise e identificação de entidades que representam os dados importantes para a organização e o **Diagrama de Entidade e Relacionamento** utilizado para modelar a estrutura dos dados
  - Matriz de planejamento para vincular as funções identificadas na decomposição funcional com às entidades com o propósito de identificar órfãos. Por exemplo, identifica quais funções não fazem uso de nenhuma entidade ou entidades que não são utilizadas por nenhuma função

```
Lagyan Cards Incorporated
Business Goals:
1. Increase distribution of cards by 50% for the next three years.
2. Increase network of dealers nationwide by 10% for the next year.

Organizational Units:
1. Sales Department
2. Accounting Department
3. Financial Department
4. Business Centers

Business Functions:
1. Inventory of e-Prepaid Cards
2. Accounting of all financial transaction
3. Sales monitoring of e-Prepaid Cards

List of Entity Types
1. Customers, Direct Resellers, Dealers
2. e-Prepaid cards, phone cards and internet access cards
3. e-prepaid card transactions
```

Figura 1: Objetos de Planejamento Corporativo

A fase de planejamento gera vínculos com a camada de Escopo do Negócio do *framework* ISA.

## 4.2. Fase de Análise

Esta fase é também conhecida como a fase da Engenharia de Requisitos. O propósito é desenvolver especificações detalhadas para o sistema de informação requisitado pela organização. Essas especificações incluem suporte a decisões e sistemas de informações executivos bem como sistemas de processamentos transacionais. Cobre o estudo da atual situação do negócio bem como a determinação dos requisitos para o novo sistema. Lida com uma área de negócios por vez que é o agrupamento de funções e entidades coesas que dão base para o desenvolvimento do sistema de informação.

CrITÉRIOS para a definição da Área de Negócios

1. Área de negócios deve ser claramente delimitada.
2. Pequena o bastante para ser bem entendida e facilmente gerenciável.
3. Grande o suficiente para necessitar de bancos de dados compartilhados.



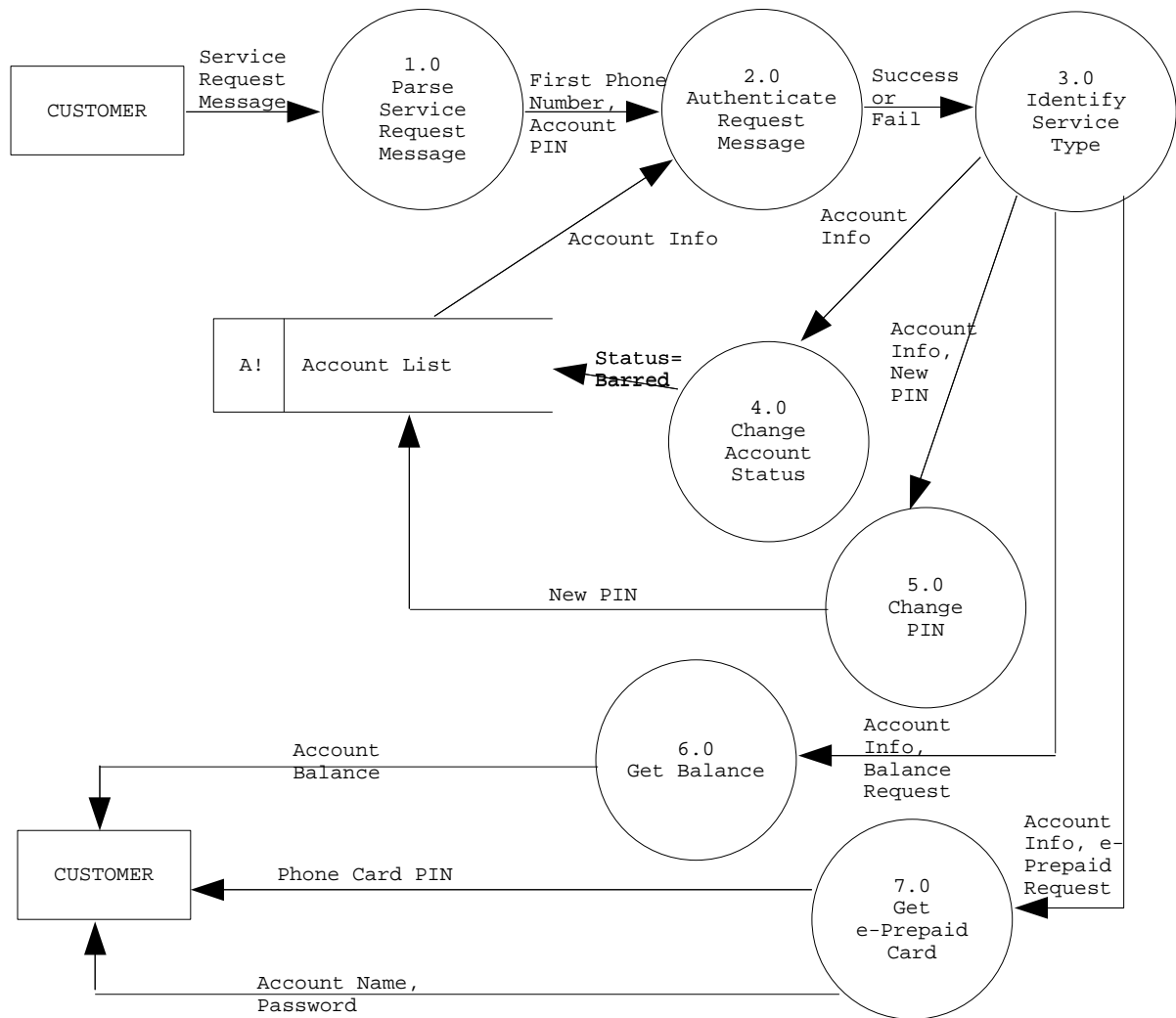


Figura 3: DFD Nível 0 – Serviço de Requisição do Sistema de Mensagem

A fase de análise está relacionada com as camadas **Modelo do Negócio** e **Modelo do Sistema de Informação** do *framework* ISA.

### 4.3. Fase de Projeto

O objetivo da Fase de Projeto é transformar o **Modelo Conceitual** e o **Modelo de Processo** desenvolvidos durante a Fase de Análise em modelos que se adequar a tecnologia a ser utilizada na implantação do sistema de informação. Dois modelos de projeto são criados, **Projeto de Banco de Dados** e **Projeto de Processo**.

#### Passos da Fase de Projeto

##### 1. Desenvolvimento do Projeto do Banco de Dados

O propósito desse passo é mapear o **Modelo Conceitual de Dados** para o **Modelo de Implementação** que um Sistema Gerenciador de Banco de Dados (SGBD) pode processar com uma performance aceitável para todos os usuários da organização. Para construir o banco de dados dois projetos são criados.

- **Projeto Lógico do Banco de Dados** que mapeia o Modelo Conceitual desenvolvido na análise das estruturas específicas do SGBD. A figura 4 mostra um trecho de um projeto lógico de banco de dados.
- **Projeto Físico do Banco de Dados** é o mapeamento das estruturas do Projeto Lógico do Banco de Dados para estruturas de armazenamento físico, tais como, arquivos e entidades. Índices também são especificados assim como métodos de acesso e outros

fatores físicos. O objetivo principal de se ter o design físico do banco de dados é fornecer performance adequada para usuários de aplicações em termos de tempo de resposta, taxas de transferência, entre outros. Cópias de segurança e sua restauração são considerados no Projeto Físico.

## ACCOUNT

CellPhoneNo	PIN	Balance	Limit	Status	Type
09192345678	1234	\$500,00	\$2.500,00	ACT	2
09174561234	2345	\$100,00	\$2.000,00	ACT	2
09205467234	4523	\$25.000,00	\$300.000,00	ACT	1
09165647342	7812	\$30.000,00	\$300.000,00	ACT	1

## STATUS

CODE	DESCRIPTION
ACT	Active Account
BAR	Barred Account
TER	Terminated Account

## TYPE

CODE	DESCRIPTION
1	Dealer Account
2	Direct Reseller

## PHONECARDTRANSACTION

CELLPHONENO	CARDNO	LOADDATE	RECIPIENT
09205467234	2346253782	DEC-24-2006	9223456173
09205467234	8736237634	DEC-24-2006	9178746345

Figura 4: Exemplo de Projeto Lógico do Banco de Dados

## 2. Desenvolvimento do **Projeto de Processo**.

O propósito desse passo é especificar a lógica de cada um dos processos e incluir todas as referências para as entidades relevantes. Existem dois subpassos para o Projeto de Processo:

- Especificar a lógica detalhada de cada processo
- Desenhar interfaces de usuários que podem ser: menus, formulários e relatórios, entre outros

A figura 5 é um exemplo da especificação de um Projeto de Processo. O trecho da especificação com fonte vermelha deve ser tratado como uma transação.

Project Name: e-Lagyan Distribution System  
 Company: Lagyan Cards Incorporated

Process Specifications: Phone Card Service Request

INPUT: PAccountNo, PBalance, VFirstCellNo, VSecondCellNo,  
 VService, VDenomination

```

IF PBalance < VDenomination THEN
  MESSAGE "Insufficient Balance"
  Terminate Transaction
ELSE
  SELECT InventoryCount
  INTO PCount
  FROM CellCardSupply
  WHERE Code IN (SELECT Code
                  FROM CellCardType
                  WHERE Denomination = VDenomination
                  AND SPID = VService)

  IF (PCount < 0) THEN
    MESSAGE "No Card Available" TO VFirstCellNo
  ELSE
    SELECT FirstAvailableSerialNo, PIN, CardType
    INTO PSerialNo, PPIN, PCardType
    FROM CellCard
    WHERE CardType IN (SELECT Code
                      FROM CellCardType
                      WHERE SPID = VService
                      AND Denomination = VDenomination)
    AND Status = "VALID" or "UNSOLD"
    IF VSecondCellNo <> NULL THEN
      INSERT INTO CellCardTrans VALUES
        (PAccountNo, PSerialNo, SYSDate, VSecondCellNo)
    ELSE
      INSERT INTO CellCardTrans VALUES
        (PAccountNo, PSerialNo, SYSDate, VFirstCellNo)
    ENDIF
    UPDATE Account SET
      Balance = Balance - Denomination
    WHERE AccountNo = PAccountNo
    UPDATE CellCardSupply SET
      Count = Count - 1
    WHERE Code = PCardType
    UPDATE CellCard SET
      Status = 'Invalid' or 'Sold'
    WHERE SerialNo = PSerialNo
    IF VSecondCellNo <> NULL THEN
      SEND PPIN to VSecondCellNo
    ELSE
      SEND PPIN to VFirstCellNo
    ENDIF
  ENDIF
ENDIF

```

Figura 5: Especificação do Projeto de Processo dos cartões telefônicos e-Prepaid

#### 4.4. Fase de Implementação

O propósito da fase de implementação é construir e instalar o sistema de informação de acordo com os planos e designs. Isto envolve uma série de passos que acarretará a operacionalização do sistema de informação que incluem criar definições de banco de dados, criar códigos de programas, testar o sistema, desenvolver procedimentos operacionais e documentação.

## 5. Componentes para um SGBD

Sistema Gerenciador de Banco de Dados (SGBD) são softwares altamente sofisticados e complexos que tem por objetivo fornecer serviços para gerenciar dados de uma organização. Nessa seção, os tipos genéricos de funções e serviços serão discutidos e uma possível arquitetura de um SGBD será apresentada.

### 5.1. Serviços e Funções de um SGBD

Codd<sup>1</sup> definiu oito serviços que devem ser fornecidos por um SGBD de alta-escala.

1. Um SGBD deve fornecer aos usuários uma forma de armazenar, recuperar e atualizar dados no banco de dados. Essa é a função fundamental de um SGBD.
2. Um SGBD deve fornecer um catálogo com descrição de itens de dados armazenados e acessíveis aos usuários. A chave para esse serviço é o **catálogo de sistema** ou **dicionário de dados**, que é um repositório de informações descrevendo os dados no banco de dados. É referenciado como dados sobre dados ou **meta-dados**.
3. Um SGBD deve fornecer um mecanismo que assegure que todas as atualizações correspondentes a uma dada transação sejam feitas ou que nenhuma delas seja feita. A chave para esse serviço é o conceito de transação. Uma **transação** é uma série de ações que acessam ou modificam o conteúdo do banco de dados. Quando executada, deve garantir que o banco de dados esteja sempre num estado estável e consistente.
4. Um SGBD deve fornecer um mecanismo que assegure que o banco de dados seja atualizado corretamente quando múltiplos usuários atualizam o banco de dados concorrentemente. A chave para esse serviço é permitir que muitos usuários acessem o banco de dados simultaneamente. O acesso concorrente é relativamente fácil quando os usuários estão apenas lendo os dados. Entretanto, quando os usuários estão acessando o banco de dados simultaneamente e pelo menos um deles está escrevendo dados, podem ocorrer interferências que podem resultar em inconsistências. O SGBD deve garantir que acessos concorrentes não deixem o banco de dados num estado inconsistente.
5. Um SGBD deve fornecer um mecanismo para recuperação do banco de dados no caso de dados serem danificados de alguma maneira. Isso está relacionado com as transações. Quando uma transação falhar, o SGBD deve retornar ao estado consistente que estava antes da transação ser executada.
6. Um SGBD deve fornecer um mecanismo para garantir que apenas usuários autorizados tenham acesso ao banco de dados. A chave para esse serviço é a segurança, que se refere à proteção do banco de dados contra acessos não autorizados, intencionais ou não.
7. Um SGBD deve ser capaz de se integrar com softwares de comunicação.
8. Um SGBD deve fornecer um meio de garantir que os dados e suas mudanças sigam certas regras. A integridade de dados se refere-se à exatidão e consistência dos dados armazenados no banco de dados. É usualmente expressa em termos de restrições (*constraints*) que são regras de consistência que o banco de dados não deve violar. Também é conhecido como implementação de regras de negócio.

Outros serviços a serem considerados e fornecidos por um SGBD, são:

1. Um SGBD deve incluir facilidades de suporte à independência dos programas da estrutura do banco de dados. A chave para isso é o conceito de visões (*views*) e subesquemas.
2. Um SGBD fornece um conjunto de serviços utilitários:
  - facilidades de importação que carreguem o banco de dados utilizando arquivos, e facilidades de exportação que descarreguem o banco para arquivos.
  - facilidades de monitoramento da utilização e operação do banco de dados

<sup>1</sup> Codd E. F. The 1981 ACM Turing Award Lecture: Relational Database: A Practical Foundation for Productivity. Comm. ACM, 25(2), 109-117

- análises de performance e estatísticas de utilização
- facilidades para reorganização de índices e suas sobrecargas
- *garbage collection* e realocação para remover fisicamente os registros eliminados para consolidar o espaço liberado e realocá-lo quando houver necessidade

## 5.2. Arquitetura de Banco de Dados

Não é possível generalizar a estrutura de componentes de um SGBD, pois isto varia de sistema para sistema. No entanto, é útil entender o sistema de banco de dados visualizando seus componentes e como eles se relacionam. Figura 6<sup>1</sup> é uma arquitetura possível de SGBD.

Um SGBD é modularizado em diversos componentes de softwares relacionados. Cada componente foi desenvolvido para executar operações específicas. O diagrama mostra como cada componente interage com os outros. A seguir listamos os componentes.

- **Processador de Consultas.** É considerado um dos principais componentes do SGBD. Transforma consultas em uma série de instruções de baixo-nível que são direcionadas ao software gerenciador do banco de dados
- **Gerenciador de Banco de Dados.** Interage com aplicações e consultas enviadas pelo usuário. Aceita consultas e examina esquemas externos e conceituais para determinar quais registros conceituais são requeridos para satisfazer uma consulta. Chama o gerenciador de arquivos para atender à requisição. Possui os seguintes componentes:
  - **Controle de Autorização.** Verifica se o usuário possui a autorização necessária para executar a operação requerida
  - **Processador de Comandos.** Responsável por controlar a operação quando se sabe que o usuário possui a autorização necessária
  - **Validador de Integridade.** Verifica se a operação requisitada satisfaz as restrições de integridade necessárias (como as restrições de chaves)
  - **Otimizador de Consultas.** Determina a estratégia para execução otimizada de consultas
  - **Gerenciador de Transações.** Executa o processamento das operações requeridas. Recebe as validações de integridade das transações
  - **Agendador.** Garante que operações concorrentes no banco de dados procedam sem conflitos entre si. Controla a ordem relativa em que as transações são executadas
  - **Gerenciador de Recuperação.** Garante que o banco de dados permaneça num estado consistente quando uma falha ocorrer (como violação de restrições). É responsável por confirmar ou abortar uma transação
  - **Gerente de Buffer.** É responsável por transferir dados entre a memória principal e o disco
- **Gerenciador de Arquivos.** Manipula os arquivos internos de armazenamento e gerencia a alocação de espaço em disco. Não é responsável por gerenciar diretamente entradas e saídas de dados. Ao contrário, passa a requisição para o método de acesso apropriado
- **Processador DML.** Converte as instruções da Linguagem de Manipulação de Dados encontradas nas aplicações em chamadas de funções do banco de dados. Interage com o processador de consultas para gerar o código apropriado.
- **Compilador DDL.** Converte as instruções da Linguagem de Definição dos Dados em um conjunto de tabelas que contém os meta-dados. As tabelas são armazenadas no catálogo de sistema enquanto que informações de controle são armazenadas no cabeçalho dos arquivos de dados
- **Gerenciador de Catálogo.** Gerencia acesso e mantém os catálogos de sistema

<sup>1</sup> Codd E. F. (1982) *The 1981 ACM Turing Award Lecture: Relational Database: A Practical Foundation for Productivity*. Comm. ACM, 25(2), 109-117



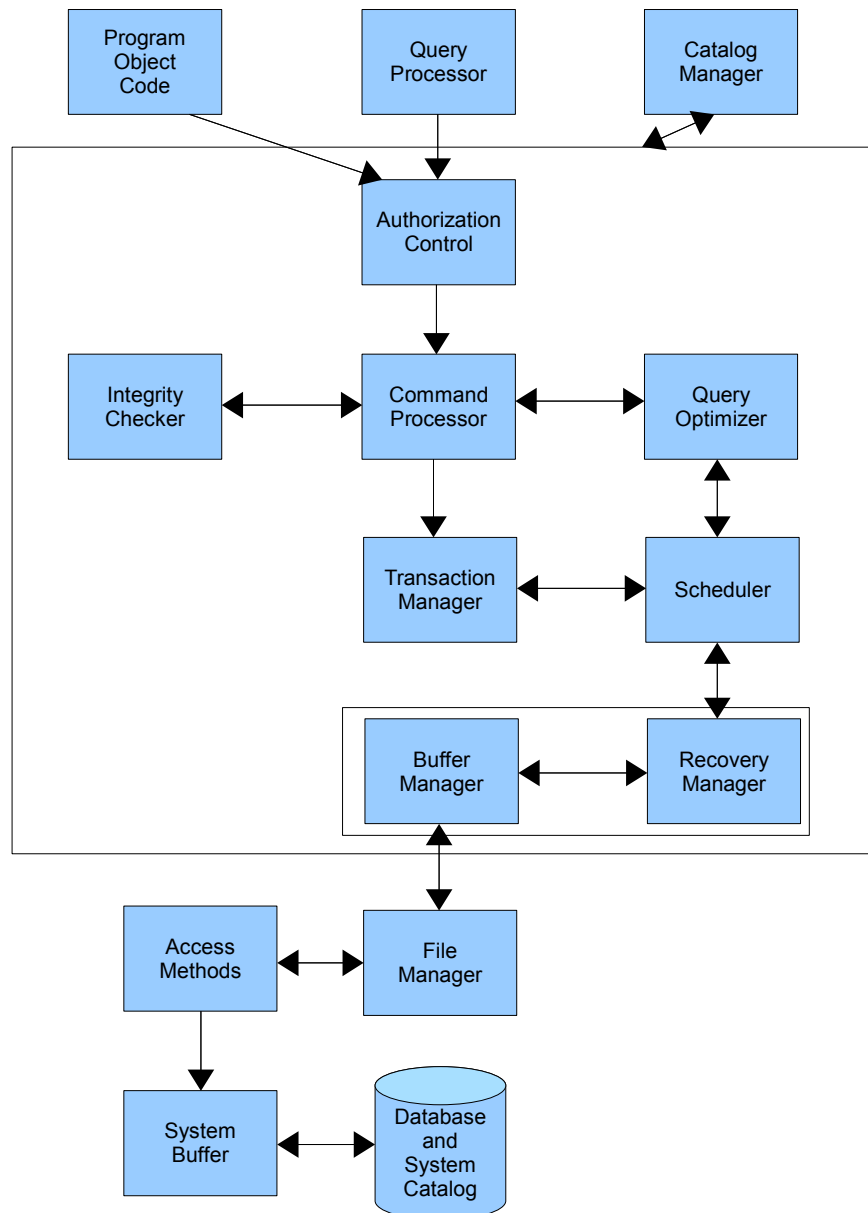


Figura 6: Arquitetura de Banco de Dados

## Parceiros que tornaram JEDI™ possível



### ***Instituto CTS***

Patrocinador do DFJUG.

### ***Sun Microsystems***

Fornecimento de servidor de dados para o armazenamento dos vídeo-aulas.

### ***Java Research and Development Center da Universidade das Filipinas***

Criador da Iniciativa JEDI™.

### ***DFJUG***

Detentor dos direitos do JEDI™ nos países de língua portuguesa.

### ***Politec***

Suporte e apoio financeiro e logístico a todo o processo.

### ***Instituto Gaudium***

Fornecimento da sua infra-estrutura de hardware de seus servidores para que os milhares de alunos possam acessar o material do curso simultaneamente.