



Residência
em Software

Residência em Tecnologia da Informação e Comunicação

JDBC: o Padrão DAO

Professor:

Alvaro Degas Coelho



INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



**Parecia que o mundo
iria para os PCs...**

Mas não foi!

Porque superar a “era da conectividade”

- Downsizing não funcionou
 - Sistemas corporativos essenciais permaneceram nos MainFrames
 - Código maduro e funcional
 - Bug do milênio “bugou”
 - *Rightsizing*
 - Manter os Sistemas corporativos nos MainFrames e os PCs para novas aplicações
- Problemas da “Era da conectividade”
 - Custo alto de manutenção (máquinas complexas e localizada em pontos distantes)
 - Problemas cada vez mais remotos (e mais complexos)



Residência
em Software

Como assim tenho que pagar para ir em Cabobreira do Sul e em Guajarinha do Oeste para reinstalar o sistema da filial de lá?

Com o MainFrame eu não tenho esse tipo de problema! É tudo centralizado na sede!





Residência
em Software

**Vamos observar o
que houve...**

As famosas 3 camadas

- Uma definição
- Os Sistemas são compostos de três partes distintas
- Interface do Usuário
 - Parte do sistema que conversa com as pessoas (e com outros sistemas)
- Regras de Negócio
 - Parte do sistema que executa o processamento dos dados de acordo com as regras da organização
- (Acesso a) Dados
 - Parte do sistema que garante a persistência (e consistência) dos dados



Residência
em Software

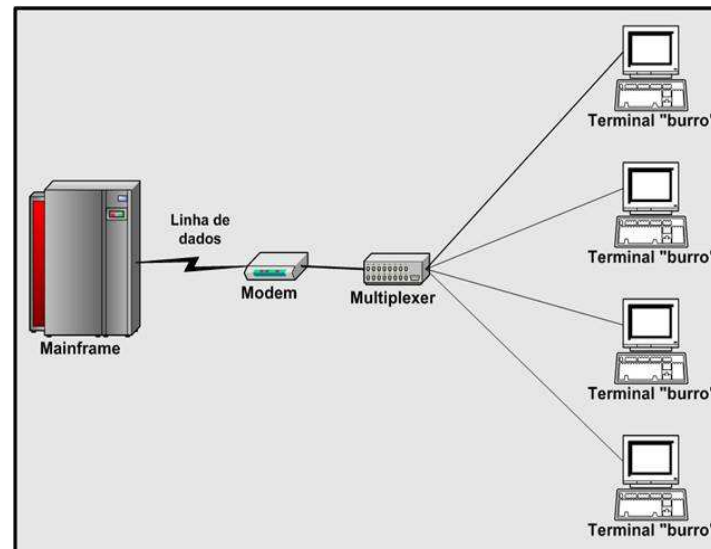


Sistemas “Duas camadas”

- Um programa (Delphi, VB, etc.)
 - Implementa a Interface e as Regras de Negócio
- Um servidor (Oracle, Sql-Server, etc.)
 - Implementa Acesso a Dados
- *Fat Client*
 - Difícil de manter (reinstalação)
 - Máquina robusta (e cara)



Vimos que os MainFrames são “Uma camada”

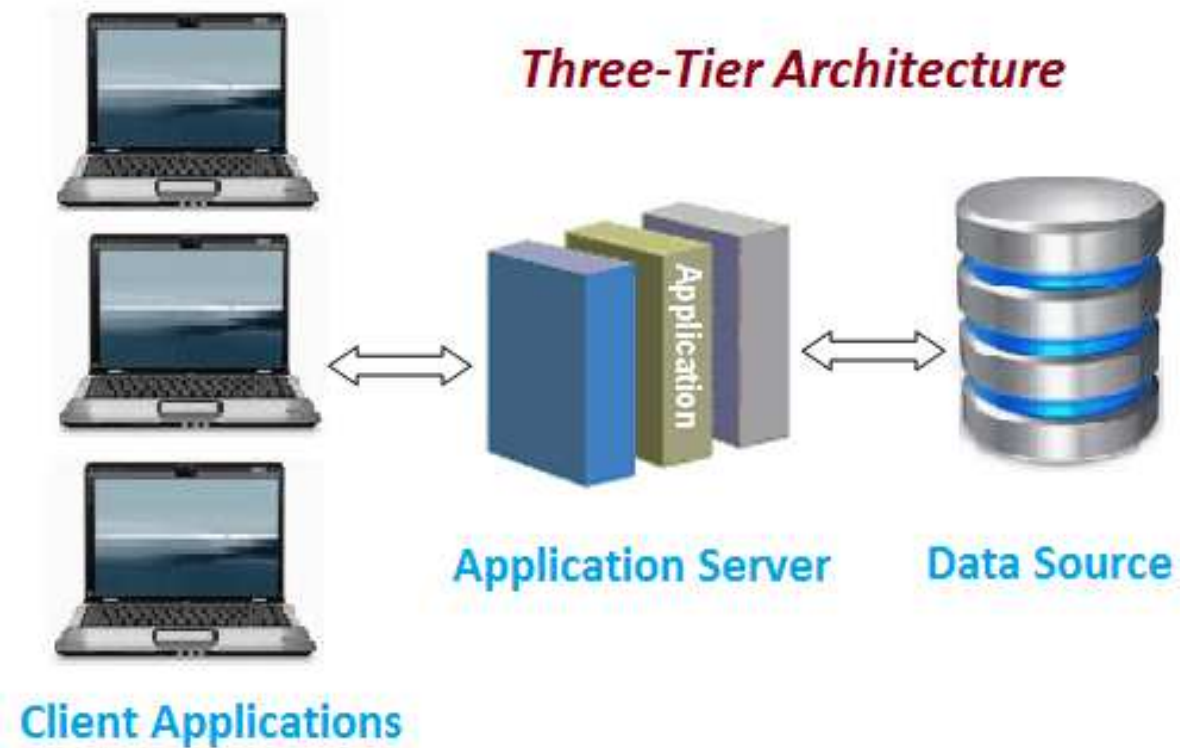


Sistemas de 3 camadas

- Ideia Geral
 - “Apartar” Regras de Negócios de Interface
- Surge o “Servidor de Aplicações”
- Na camada de interface
 - Thin Client
 - Fácil de instalar (basta o SO e um Browser)
 - Máquinas menos robustas



Residência
em Software



Porque é melhor?

- Permite clientes “magros”
 - Basicamente um SO e um Browser
- Permite gerenciamento centralizado
 - Aplicações e bancos de dados localizados
- Do jeito que o MainFrame fazia... e além
 - Flexibilização de fornecedores
 - Tanto de Hardware quanto de Software



Residência
em Software



Múltiplas Camadas

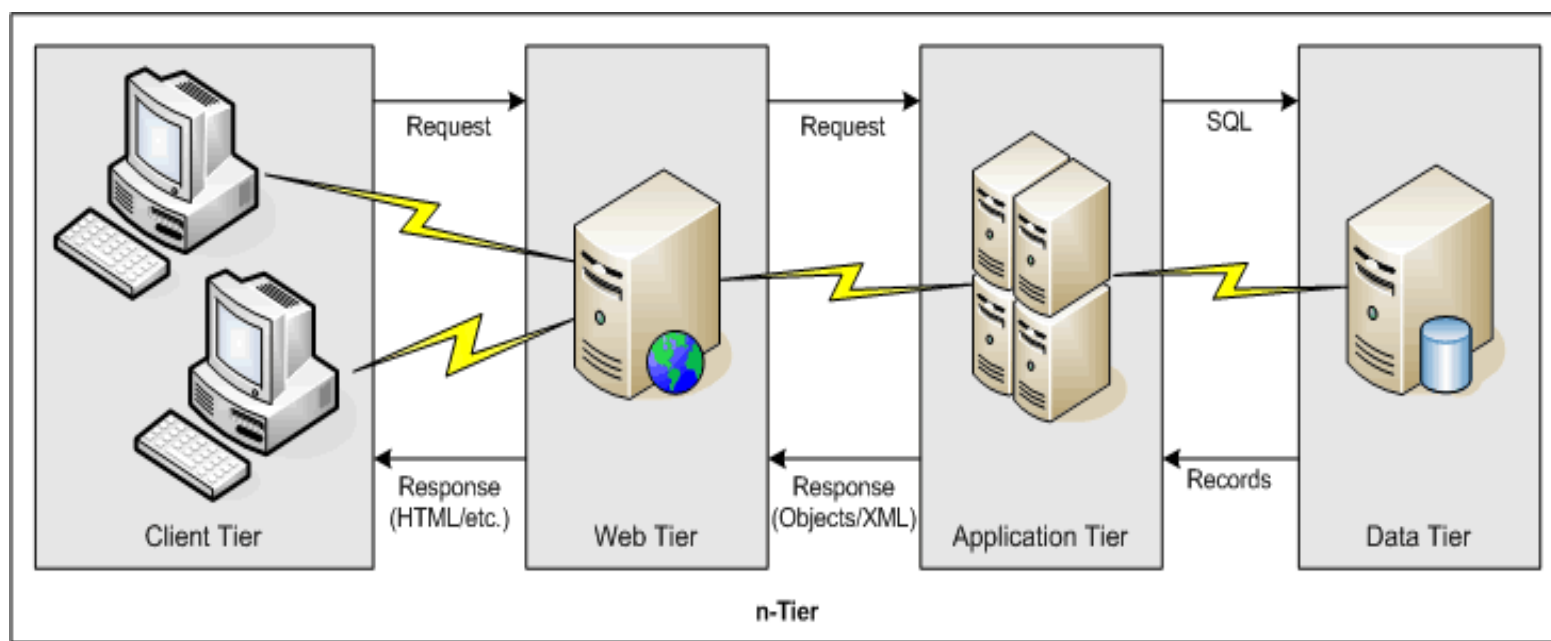
- A camada de Regras de Negócio pode ser repetidamente “dividida”
 - Melhorar escala e reuso
 - Componentes
- Componentes ideais
 - Baixo acoplamento
 - Alta coesão
- Via protocolos de Internet: conectividade entre componentes



Formato atual de TODO sistema corporativo



Residência
em Software

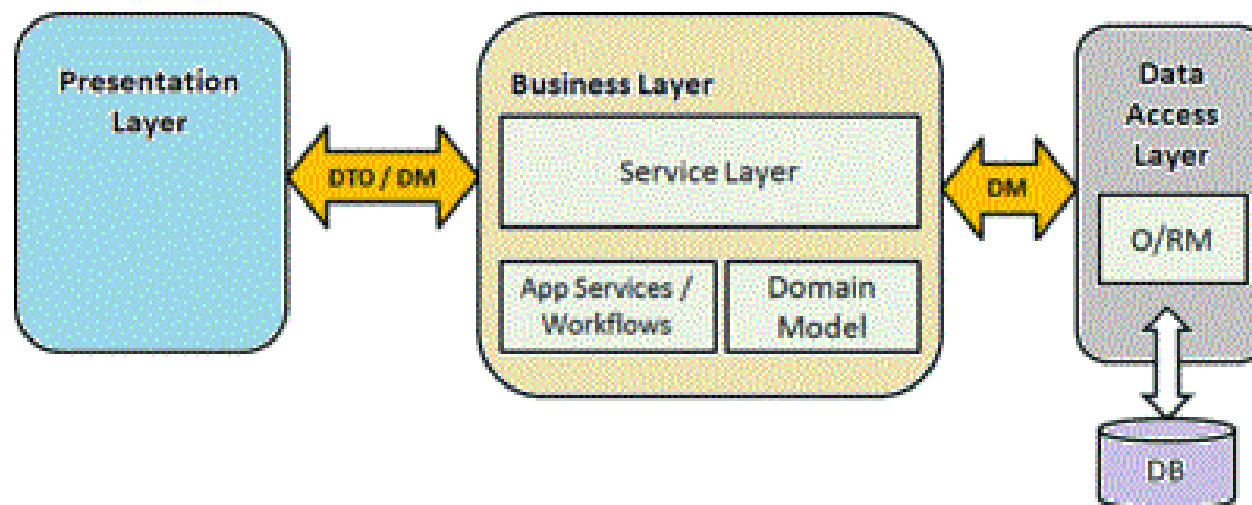


O que te tem DAO a ver com isso?

- DAO é a ideia de separar absolutamente a camada de Aplicação da camada de Acesso a dados
 - Mas elas precisam manter uma interface para conversar entre si
- Do padrão Expert
 - Alguém tem que se especialista em acessar os dados
- Refinando
 - Alguém tem que ser especialista em acessar CADA TIPO DE DADOS



Residência
em Software





Residência
em Software

Seguiremos construindo nosso DAO

O Objeto Connection

- Connection é criado após as credenciais terem sido devidamente validadas
 - User, Password, Banco de Dados, URL, Porta
- Um objeto connection permite acesso a uma série de outros objetos para manipulação do Banco de Dados
 - O primeiro deles: PreparedStatement

PreparedStatement

- Implementa uma instrução válida em SQL
- Permite que ela seja executada no SGBD
 - A partir de um objeto do tipo Connection
- Tipicamente usada em contexto dinâmico
 - Podem haver múltiplas operações ao mesmo tempo
- PreparedStatement tem vários métodos
 - Voltaremos em breve.

Um exemplo do uso do PreparedStatement

- Supondo um objeto Connection (conn) já instanciado
 - No BD há uma tabela Estudante(#Matricula, Nome, CPF, \$CodCurso)
- Temos um objeto Estudante (Leo) cadastrado no sistema
 - Dados: (1111, “Leo”, “111.222.333-44”, 10021)
- Qual seria a sentença SQL para inserir Leo no BD?
 - INSERT INTO Estudante (Matricula, Nome, CPF, CodCurso)
VALUES (1111, “Leo”, “111.222.333-44”, 10021)
- Como criamos a sentença correta?
 - PreparedStatement
- Como enviamos a sentença correta para execução no BD?
 - PreparedStatement (usando a conexão)

Organizando as responsabilidades

- A que Classe cabe a responsabilidade de Inserir o novo Estudante?
- Padrão DAO: duas soluções possíveis
 - Uma Classe especializada em salvar qualquer coisa
 - Um conjunto de classes, cada uma especializada em um tipo de objeto
- No nosso exemplo ficaria: uma package “dao”
 - Várias classes DAO
 - Uma delas EstudanteDAO: responsabilidade de inserir um novo estudante
 - O PreparedStatement para executar o insert é implementado ali