

A decorative graphic on the right side of the page. It features three concentric blue circles of different sizes, each with a lighter blue outer ring. Two thin blue lines intersect at a point between the top and middle circles, extending towards the top-left and bottom-right corners of the page. A larger concentric blue circle is partially visible at the bottom right corner.

# Padrões de Desenvolvimento Java V.1

## Nomenclatura

Este documento foi criado com o objetivo de estabelecer um padrão de desenvolvimento em aplicações Java SE quanto a nomenclatura de classes, interfaces, métodos, variáveis, componentes visuais.

**Elissandro Messias Santos**  
**01/04/2010**

# Índice

- 1. Arquitetura em Camadas
  - 1.1 - Apresentação
  - 1.2 - Negocio
  - 1.3 - Acesso a Dados
  - 1.4 - Pastas Auxiliares
- 2. Padrão de Nomenclatura
  - 2.1 - Classes
  - 2.2 - Interfaces
  - 2.3 - Métodos
  - 2.4 - Variáveis
  - 2.5 - Componentes Visuais
- 3. Padrão de Telas

# 1. Arquitetura em camadas

O estilo de estrutura adotado para o desenvolvimento do sistema será o estilo em camadas. Serão utilizadas três camadas: Apresentação, Negócio e Acesso a Dados. Cada uma delas será um pacote, como mostrado na **Figura 1**. Também serão criadas as pastas auxiliares: lib, imagens, relatórios e artefatos.

## 1.1 - Apresentação

- Contém as classes responsáveis pela comunicação do usuário com o sistema;
- As classes desta camada estarão visíveis apenas para as classes da camada de Negócio e vice-versa;
- O nome destas classes seguem o padrão de nomenclatura como descrito na seção Nomenclatura acrescido do sufixo **-Apr**.

Exemplo:

ClienteApr, PessoaFisicaApr, ...

## 1.2 - Negocio

- Contém as classes de objetos e as classes de negócio para as comunicações: Apresentação-Negócio e Negócio-Acesso a dados;
- Possui também uma classe abstrata chamada “UtilNeg” a qual implementará métodos auxiliares de uso geral de validação, formatação e outros, como, por exemplo, validar CPF, colocar data no formato exigido pelo banco de dados;
- As classes desta camada estarão visíveis apenas para as classes da camada de Acesso a Dados e vice-versa;
- O nome das classes de objeto e negócio seguem o padrão de nomenclatura como descrito na seção Nomenclatura. O nome das classes de objeto não recebem sufixo, já o das classes de negócio é acrescido do sufixo **-Neg**.

Exemplo:

Classes de Objetos: Cliente, Funcionario, Produto.

Classes de Negócio: ClienteNeg, FuncionarioNeg, ProdutoNeg, UtilNeg.

## 1.3 - Acesso a Dados (DAO)

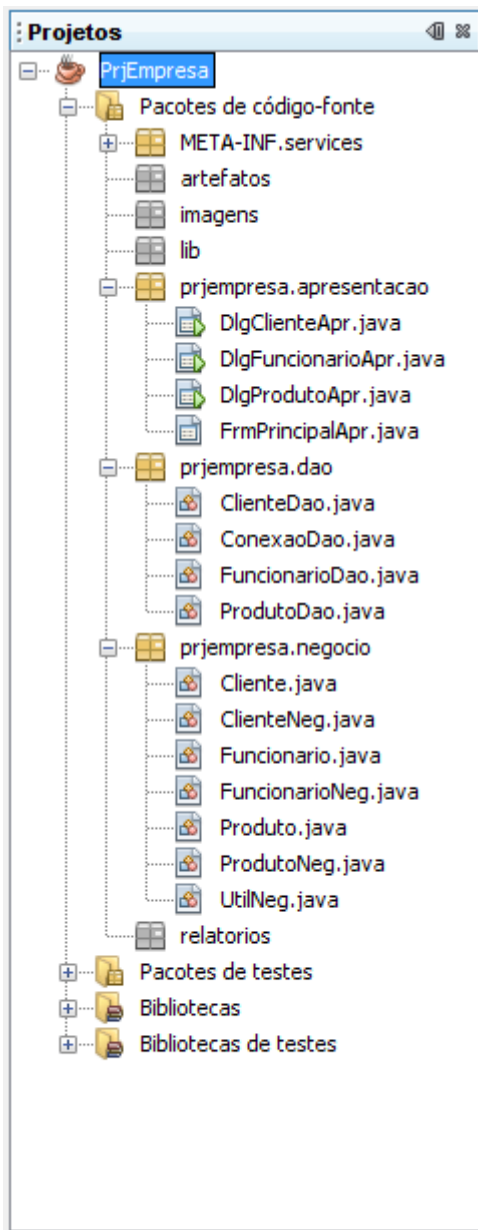
- Contém as classes que implementam os métodos acesso a dados através de comandos SQL;
- Possui também uma classe chamada “ConexaoDao” a qual implementará os métodos de conexão com o banco de dados, como, por exemplo, abrir e fechar conexão com o banco, transação e outros. Ela só será visível para as classes desta camada;
- As classes desta camada serão visíveis apenas para as classes da camada de Negócio e vice-versa, exceto ConexaoDao;
- O nome destas classes seguem o padrão de nomenclatura como descrito na seção Nomenclatura acrescido do sufixo **-Dao**.

Exemplo:

ClienteDao, FuncionarioDao, ProdutoDao, ConexaoDao.

#### 1.4 - Pastas Auxiliares

- **lib**: conterá arquivos JAR utilizados pelo sistema. Ex: driver's JDBC
- **imagens**: arquivos de imagem utilizados pelo sistema.
- **relatórios**: arquivos de relatórios gerados pelo iReport, sendo utilizados pelo sistema.
- **artefatos**: arquivos gerados automaticamente pelo sistema. Ex: recibos e contratos em PDF.



**Figura 1.** Disposição das camadas

# Padrões de Nomenclatura

## 1. Classes

A primeira letra deve ser maiúscula e se for composta por mais de uma palavra, devem ser escritas juntas e cada uma inicia com letra maiúscula. Este formato é chamado de “camelCase”.

Exemplo:

```
class Automovel {  
  
    ...  
  
}  
  
class CarroDoMeuPai extend Automovel {  
  
    ...  
  
}
```

## 2. Interfaces

O nome da interface segue o mesmo padrão adote para classes.

Exemplo:

```
Interface MinhaInterface {  
  
    void freio();  
  
}  
  
class Automovel implements MinhaInterface {  
  
    ...  
  
}
```

## 3. Métodos

A primeira letra minúscula e se composta por mais de uma palavra a partir da segunda deve seguir a regra “camelCase”.

Exemplo:

```
public void setNomeDoCarro(string nmCarro) {  
  
    ...  
}
```

### 3.1 - Prefixo dos Métodos

	Prefixo	Significado	Exemplo
1	set	Atribuir valor	void setNome(String nome)
2	get	Obter valor	Date getDtNascimento()
3	is	Obter valor booleano	boolean isVencido(Date data)
4	add	Adicionar evento	void addActionListener(MeuListener m)
5	remove	Remover evento	void removeMeuEvento(MeuListener m)

## 4. Variáveis

O nome da variável segue o mesmo padrão adote para nome de métodos.

Obs<sup>1</sup>: Usar nomes sugestivos, ou seja, que indiquem o que as variáveis estão armazenando.

Ex:

idAluno, nmCliente, dsCarroDoMeuPai,...

Obs<sup>2</sup>: Se o nome for ou tiver uma sigla deve seguir a mesma idéia acima.

Ex:

cdRg, cdCpf, cdCep,...

### 4.1 - Prefixo das variáveis:

Toda variável terá um prefixo que indicará a informação que ela armazena. A tabela abaixo tem a lista de prefixos a serem utilizados:

	Prefixo	Significado	Conteúdo	Exemplo
1	id	Identificador (chave primária)	1	idCliente
2	sq	Sequencial	12	sqAluno
3	cd	Código	456.300.100-23	cdCpf, cdRg,cdMatrícula, cdCep
4	no	Número (Nº)	1000	noEndereco
5	qt	Quantidade	100	qtProduto
6	vl	Valor	R\$: 3,00; 70 Kg	vlRenda, vlPeso
7	pc	Percentual	3%	pcVotos

8	tx	Taxa	R\$: 4,50; 5,4%	txMulta
	nm	Nome	“João Pedro”	nmAluno
	sg	Sigla	SE, BA,...	sgEstado
	ds	Descrição	“rua 2”, “joão@gmail.com”	dsEndereco, dsEmail
	tp	Tipo	‘M’, ‘F’, ...	tpSexo
	fl	Flag	(0 ou 1, True ou False)	flAtivado
	im	Imagem	<Imagem qualquer>	imFoto3x4
	arq	Arquivo	.pdf,. doc, .mp3, ...	arqEdital
	dh	Data e Hora	2010-10-01 14:02:00	dhSaque
	dt	Data	01/03/2010	dtVencimento
	dd	Dia de uma data	01 (01/05/2010)	ddConsulta
	mm	Mês de uma data	05 (01/05/2010)	mmConsulta
	aa	Ano de uma data	2010 (01/05/2010)	aaConsulta
	hr	Hora	08:10:55	

## 5. Componentes Visuais

A tabela a seguir refere-se a alguns componentes visuais Java GUI Swing.

Dica: Para saber se um componente visual é um Swing, basta observar se o nome da classes que representam o componente inicia com a letra “J” (“Jota”).

### 5.1 - Prefixo para os nomes dos componentes visuais

Janelas			
	Componente	Prefixo	Exemplo
1	<b>JFrame</b>	<b>frm</b>	<b>frmPrincipal</b>
<p>Quadro JFC (Swing). Frames (Quadros) são tipicamente utilizados como <b>janelas</b> de nível superior autônomas como a <b>interface principal</b> do usuário para um aplicativo. <b>A maioria dos aplicativos Swing são construídos iniciando a partir deste formulário.</b></p>			
2	<b>JDialog</b>	<b>dlg</b>	<b>dlgFuncionario</b>
<p>Caixa de diálogo JFC (Swing). Caixas de diálogo são janelas modais ou não-modais que tipicamente são utilizadas para requerer entrada de dados aos usuários. Por exemplo, “<b>janelas filhas</b>”, sendo o Frame a “<b>janela Pai</b>”.</p>			
3	<b>JFileChooser</b>	<b>flc</b>	<b>flcAnexarArquivo, flcSelecionarFoto</b>
<p>Um painel de controles projetado para permitir que o usuário selecione um arquivo no computador.</p>			
Menus			
	Componente	Prefixo	Exemplo
	<b>JMenuBar</b>	<b>mnb</b>	<b>mnbPrincipal, mnbCliente</b>

Um contêiner para menus e itens de menu.

	<b>JMenu</b>	<b>mnu</b>	<b>mnuPrincipal, mnuAluno</b>
--	--------------	------------	-------------------------------

Menu para itens de menu e submenu.

	<b>JMenuItem</b>	<b>mni</b>	<b>mniCadastrarCliente</b>
--	------------------	------------	----------------------------

Um item de menu.

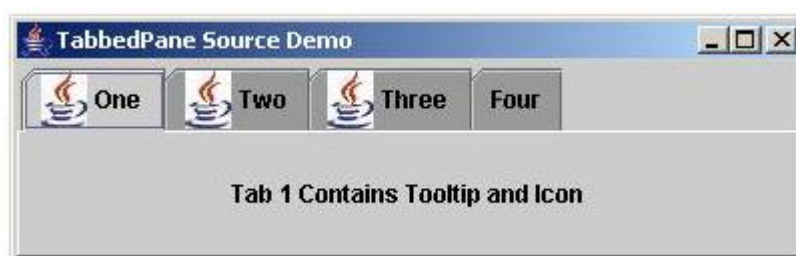
## Contêineres

	Componente	Prefixo	Exemplo
4	<b>JPanel</b>	<b>pnl</b>	<b>pnlDadosPessoais, pnlDadosParaContato</b>

Painel JFC (Swing). Painéis são freqüentemente utilizados para **agrupar** muitos outros **componentes** em um único local. Um JPanel pode ser utilizado como um componente **dentro** de um **JFrame** ou **JDialog**.

	<b>JTabbedPane</b>	<b>tbp</b>	<b>tbp</b>
--	--------------------	------------	------------

Painel tabulado. Componente que permite que o usuário alterne entre um grupo de componentes clicando em uma guia (aba) com um determinado título e/ou ícone.



	<b>JToolBar</b>	<b>tlb</b>	
--	-----------------	------------	--

Um componente que é útil para exibir ações ou controles usados com freqüência.



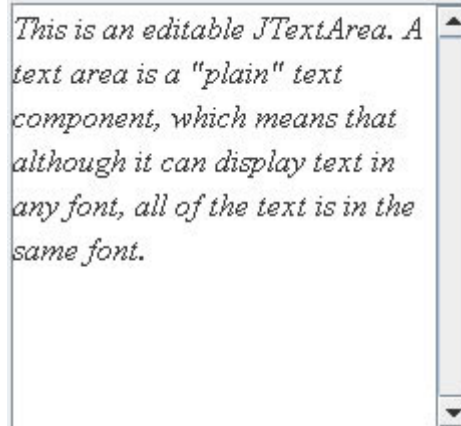
## Controles

	Componente	Prefixo	Exemplo
	<b>JLabel</b>	<b>lbl</b>	<b>lblNome, lblCPF, lblFotoAluno</b>

Componente para exibição de texto e/ou imagem.



	<b>JButton</b>	<b>btn</b>	<b>btnGravar, btnAlterar</b>
Botão simples “de ação”.			
	<b>JCheckBox</b>	<b>ckb</b>	<b>ckbSexo</b>
Componente que pode ser selecionado ou desmarcado. Geralmente, mais de uma caixa de seleção pode ser selecionada em grupo desde tipo.			
	<b>JRadioButton</b>	<b>rdb</b>	<b>rdbMasculino, rdbFeminino</b>
Componente que pode ser selecionado ou desmarcado. Usado com o ButtonGroup.			
	<b>ButtonGroup</b>	<b>btg</b>	<b>btgSexo</b>
É este não começa com “J” (Jota), mas é um Swing. Usado para agrupar um conjunto de JRadioButton’s em que é possível selecionar apenas um botão por vez.			
	<b>JComboBox</b>	<b>cbx</b>	<b>cbxEstadoCivil, cbxSexo</b>
Componente que permite ao usuário selecionar um objeto em uma lista que exibida ao clicar no botão associado ao componente.			
	<b>JList</b>	<b>lst</b>	<b>lstAssuntosDeInteresse</b>
Componente que permite o usuário selecionar um ou mais itens em uma lista.			
	<b>TextField</b>	<b>txf</b>	<b>txfNome</b>
Componente que permite editar uma única linha de texto simples.			
<div>JTextField: <input type="text" value="Hello"/></div>			
	<b>JFormattedTextField</b>	<b>ftx</b>	<b>ftxDataNascimento, ftxCEP, ftxSalario</b>
Componente que permite editar uma única linha de texto, possibilitando aplicar uma “máscara”.			
<div>JFormattedTextField: <input type="text" value="Feb 20, 2007"/></div>			
	<b>TextArea</b>	<b>txa</b>	<b>txaObservacao</b>
Componente para várias linhas de texto simples.			

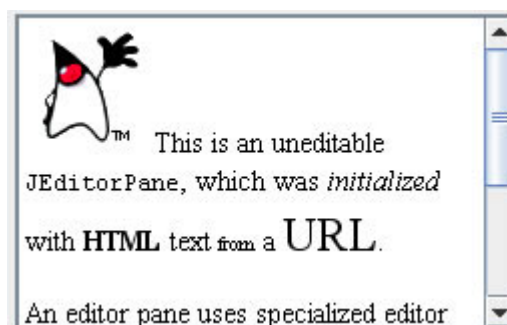


**JEditorPane**

**edp**

**edpArtigo**

Componente para várias linhas de texto que pode ser formatado (bold, italic, etc).

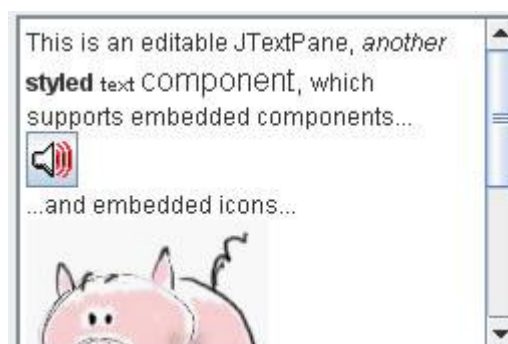


**JTextPane**

**txp**

**txpArtigoDoBlog**

Versão mais 'refinada' do JEditorPane (subclasse deste). Componente para várias linhas de texto que pode ser formatado (bold, italic, etc), especializado para uso com StyledDocument e StyleEditorKit.



**JPasswordField**

**pwf**

**pwfSenha**

Componente que permite editar uma única linha de texto, sem exibir os caracteres originais.

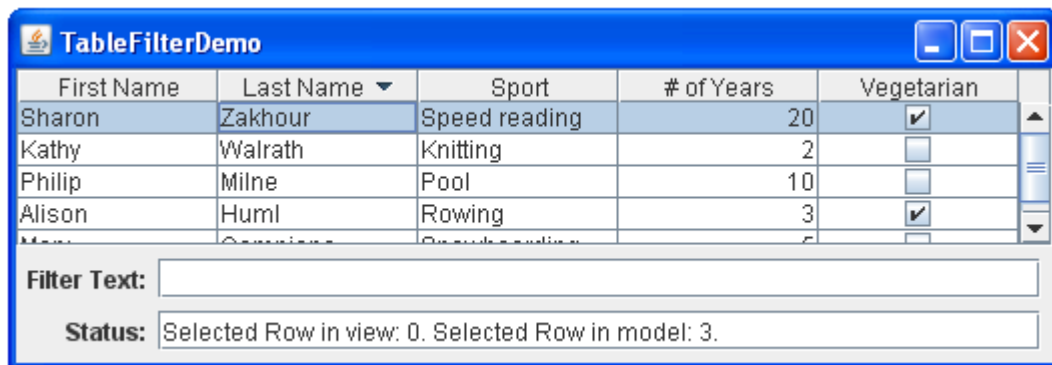


**JTable**

**tbl**

**tblListarAluno**

Componente usado para exibir uma lista de informações em linhas divididas em colunas.



**JProgressBar**

**pgb**

**pgbCarregarSistema**

Barra de Progresso. Geralmente comunica o progresso de algum trabalho exibindo sua porcentagem de conclusão e até exibir de forma textual essa porcentagem.

## 6. Telas

As telas devem seguir o formato da tela abaixo:

**Cadastro de Usuário**

**Pessoa**

CPF:  Nome:  ... +

Usuário:  Perfil de Acesso: **Administrador** ▼

Senha:

Confirmar senha:

Observação:

**Lista de Usuários:**

CPF	Nome	Usuário	Perfil de Acesso
123456789-01	Maria Silva	maria	Administrador
111111111-11	Ana Carla	ana	Auxiliar
000000000-00	João Pedro	pedro	Corretor

Gravar Alterar Consultar Excluir Sair

CPF: <input type="text"/>	Se o usuário souber o CPF, ao digitá-lo e apertar Enter ou mudar o foco para outro componente deve trazer todos dados da pessoa referente a essa tela se existir ou exibir uma mensagem informado que não existe.
<input type="button" value="..."/>	Ao ser clicado chama a tela de pesquisa do elemento ao qual está vinculado, na tela acima seria Pessoa.
<input type="button" value="+"/>	Ao ser clicado chama a tela de cadastro, na tela acima seria a tela de Cadastro de Pessoa.

## Referências

Revista .Net Magazine Design Patterns com C#.

<http://java.sun.com/docs/books/tutorial/uiswing/components/>