



**Normas Gerais do Projeto**  
**Nomenclatura na Implementação**

Página 1 de 5

**NGP. 04**

**Processo Seletivo**

Projeto Integrado 5º Semestre

<b>Nome</b>	<b>Matrícula</b>
Denise Aparecida de Lima	04
Evaldo Aurelio Alves de Lavos	31
Mara Cristina Leite	19
Rogério Luiz Alves Lopes	50
Cícero Marcelo da Silva	52
Cláudio Alves de Lima	25
Clóvis Alves de Lima	02
Robson de Sousa Martins	48

<b>Revisão</b>	<b>Alterações</b>	<b>Originador</b>	<b>Verificador</b>	<b>Data</b>
<b>0</b>	Liberação de Documento	Robson	Denise	18/03/2002
<b>A</b>				
<b>B</b>				
<b>C</b>				
<b>D</b>				
<b>E</b>				
<b>F</b>				
<b>G</b>				
<b>H</b>				
<b>I</b>				
<b>J</b>				
<b>L</b>				

## SUMÁRIO

<b>1</b>	<b>OBJETIVO .....</b>	<b>3</b>
<b>2</b>	<b>DOCUMENTOS DE REFERÊNCIA .....</b>	<b>3</b>
<b>3</b>	<b>PADRÕES PARA A IMPLEMENTAÇÃO DO APLICATIVO .....</b>	<b>3</b>
3.1	NOMENCLATURA DE ARQUIVOS .....	3
3.2	NOMENCLATURA DOS COMPONENTES .....	4
3.3	ORGANIZAÇÃO DO CÓDIGO-FONTE .....	5
3.4	VERSÃO DO CÓDIGO-FONTE .....	5

## **1 Objetivo**

Este documento tem por objetivo estabelecer as normas para a nomenclatura de arquivos, componentes e itens relativos à implementação do Projeto Sistema Seletivo, proporcionando um padrão que permitirá uma melhor manutenibilidade e compreensão do sistema.

## **2 Documentos de Referência**

Não Aplicável.

## **3 Padrões para a Implementação do Aplicativo**

O aplicativo do Projeto Processo Seletivo será implementado através do ambiente de programação Borland Delphi™ versão 6.0.

A seguir estão os padrões para a nomenclatura dos arquivos e componentes utilizados no ambiente de programação.

### **3.1 Nomenclatura de arquivos**

Os arquivos-fonte gerados pelo Borland Delphi™ deverão seguir o padrão descrito:

- Units

As units (.pas) terão seu nome composto da seguinte forma:

u\_ParaQueServeaUnit.pas

onde:

ParaQueServeaUnit é uma descrição que representa o uso da unit, sem uso de espaços, hífen ou *underlines*. Por exemplo:

u\_CadastroAlunos.pas – é uma unit que contém código relativo ao form de cadastro de alunos.

Os arquivos que contém os forms (.dfm) terão conseqüentemente os mesmos nomes dos arquivos das units correspondentes, porém com a extensão .dfm.

- Outros arquivos

Se existirem outros arquivos, como arquivos de configuração, etc. estes terão o nome de arquivo e extensão que melhor convém ao projeto.

### **3.2 Nomenclatura dos componentes**

Os componentes usados na implementação do projeto devem seguir o seguinte padrão:

xxx\_yyyy

onde:

xxx é o tipo de componente (usar tabela que será definida na próxima revisão deste documento).

yyyy é um nome, sem espaços, hífen e *underlines*, único, que representa o significado do componente.

Por exemplo:

edt\_Senha : um componente TEdit que serve para a digitação de uma senha.

edt\_Telefone : um componente TDBEdit que representa o campo Telefone em uma tabela.

Imgl\_32 : um componente TImageList que contém ícones de 32x32 pixels.

Para os forms, deve ser seguido o padrão:

frm\_ParaQueServeaUnit

Observe que ParaQueServeaUnit é a mesma descrição usada no nome do arquivo da unit correspondente a esse form.

### **3.3 Organização do código-fonte**

Para melhorar o entendimento do código-fonte, é necessário a inclusão de comentários, que podem ser redigidos livremente pelo programador.

A indentação do código deverá seguir as normas especificadas posteriormente na próxima revisão deste documento.

As units que correspondem a forms deverão conter código relativo ao próprio form. Processamentos especiais ou comuns a mais de uma unit deverão estar alocados em uma unit separada.

Deverá ser utilizada para a codificação a OOP, evitando-se uso de estruturas, variáveis ou funções repetidas e não-reutilizáveis.

Para possibilitar um melhor desempenho da aplicação (uso de memória e recursos de sistemas otimizado) deve-se utilizar o método de criação dinâmica de objetos, incluindo forms, matrizes e outras estruturas complexas, como as consultas ao BD e os relatórios.

### **3.4 Versão do código-fonte**

Para um melhor controle do código-fonte, ele deverá ser marcado com uma versão, sendo 0.0.0.0 a primeira versão (em tempo de implementação) e 1.0.0.0 a primeira versão final ao cliente.