

	<b>Normas Gerais do Projeto</b> <b>Nomenclatura na Implementação</b>		Página 1 de 6
			<b>NGP. 04</b>
	<b>Projeto Biblioteca</b>		<b>2º Ano – Turma B</b>

Nome	Nº Chamada	Matrícula
Alberto de Mattos	01	58
Granhan Bell F. Ribeiro	09	71
Rafael R. Nascimento	21	62
Robson S. Martins	22	48
Rute A. R. Silva	25	81
Sandra R. C. Ramires	26	40

Revisão	Alterações	Originador	Verificador	Data
<b>0</b>	Liberação de Documento	Robson	Alberto	15/06/2001
<b>A</b>	Alteração na Página 6 - Portabilidade	Robson	Alberto	05/10/2001
<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
3.5	PORTABILIDADE .....	6

## **1 Objetivo**

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

## **2 Documentos de Referência**

Não Aplicável.

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

O aplicativo do Projeto Biblioteca será implementado através do ambiente de programação Borland Delphi™ versão 5.0 (Enterprise).

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:

- Arquivo de Projeto

O arquivo de projeto (.dpr) terá por nome Biblioteca.dpr, gerando um executável com nome Biblioteca.exe.

Os arquivos .cfg .dof e .res terão, por consequência, o mesmo nome, porém com suas respectivas extensões.

- Units

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

unit\_ParaQueServeaUnit.pas

onde:

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

unit\_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 Biblioteca devem seguir o seguinte padrão:

xxxx\_yyyy

onde:

xxxx é o tipo de componente (usar o próprio nome gerado pelo Delphi™, excluindo a seqüência numérica do sufixo).

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

Por exemplo:

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

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

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

Para os forms, deve ser seguido o padrão:

form\_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.

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.

### **3.5 Portabilidade**

O Sistema Biblioteca será implementado com uso do BD Microsoft Access™.

A implantação do Sistema no cliente proverá a licença de uso do BD citado.

O método usado para permitir o acesso ao BD é o ADO (ActiveX Data Objects), que dispensa o uso do BDE (Borland Data Engine).

Para a implementação, pode ser usado o Delphi 5.0 Enterprise ou o Delphi 5.0 Professional + ADOExpress.

Para obter esse acesso ao BD, a máquina alvo deve ter os pacotes:

- DCOM95 (só para o Windows 95);
- MDAC 2.5 ou posterior (incluindo o MSJET 4.0).