



XML Documentation

Documentação dos COLETOR RF de WMS LanLimp

- Este documento descreve como ocorre as principais rotinas de COLETOR RF no WMS da LanLimp dentro de cada rotina prevista no fluxo de processo de entrada de mercadorias.

Elaborado por



Elaborado para





XML Documentation

SUMÁRIO

1. APRESENTAÇÃO	03
2. OBJETIVOS	03
3. Nome da Unidade de Programa identificada.....	03
Formulário para realizar a verificação do produto no endereço bipado pelo usuário	
Unit untVerificaProduto.....	03
4. XML Documentation	04

XML Documentation

1. Apresentação

Dia após dia, nós programadores escrevemos códigos e mais códigos, seja implementando uma nova rotina, ou dando manutenção em processos existentes (o que é muito comum, principalmente em sistemas legados).

Já parou para pensar quanto tempo “perdemos” tentando decifrar um código, para então poder dar a manutenção adequada? Entender o “jeito”, o raciocínio e ideia que outro programador teve para implementar determinada rotina, é muito complicado, principalmente quando o código é muito grande, poluído ou muito avançado para o nosso nível.

Muitas vezes essa situação ocorre com o nosso próprio código. Convenhamos, é muito difícil, se não impossível, lembrarmos de cada detalhe que utilizamos na escrita dos códigos. Para isso existem os comentários. Um recurso muito utilizado, que permite descrever rotinas, regras de negócios, ou até mesmo justificar o porque de tal código ser daquela maneira.

2. Objetivo

XML Documentation

O XML Documentation nada mais é do que um padrão muito utilizado para documentar códigos fontes. Este padrão não é algo exclusivo, na verdade surgiu em outras linguagens e devido ao seu grande aceite entre os desenvolvedores, foi introduzido ao RAD Studio também.

3. Nome da Unidade de Programa Identificada

O nome do arquivo fonte que contém as linhas de Formulário para realizar a verificação do produto no endereço bipado pelo usuário no WMS é: **untVerificaProduto**

XML Documentation

4. XML Documentation

Tags

Segue a lista das possíveis *tags* que podem ser utilizadas para formatar a documentação do código fonte.

summary: Descrição da função (cabeçalho);

para: Criação de parágrafos;

C: Deixa o conteúdo com a fonte *courier*, semelhante ao da IDE;

code: Semelhante a *tag* acima, porém pula uma linha antes e depois do conteúdo. Exemplo: `<code>Item := nil</code>;`

remarks: Utilizado para fazer uma observação sobre o uso do código;

param: Utilizado para documentar os parâmetros de uma função. Exemplo: `<param name="NomeDoParametro">Exemplo...</param>;`

see: Referência a um tipo, variável, parâmetro, classe, etc... Exemplo: `<see cref="Referencia"/>;`

returns: Descreve o retorno da função;

exception: Descreve as exceções que podem ser retornadas ao utilizar a função. Exemplo: `<exception cref="EExceptionTypeName"></exception>`. O texto que vai entre ">" e "<" é a descrição, da exceção;

permission: Descreve as permissões que devem ter para utilizar a rotina. Exemplo: `<permission cref="PermissionType"></permission>`. Semelhante a *tag exception*, o que tiver entre ">" e "<" é a descrição.

<code><file name></code>	" untVerificaProduto "
<code><instruction name></code>	Formulário para realizar a verificação do produto no endereço bipado pelo usuário
<code><type></code>	"PAS "
<code><caller></code>	caller=" untVerificaProduto "
<code><summary></code>	<code><summary></code> Formulário para realizar a verificação do produto no endereço bipado pelo usuário. <code></summary></code> <code><instruction name="VerificaProduto"></code> <code><summary></code> Procedimento com a finalidade de permitir ao usuário realizar a verificação do produto a partir do endereço bipado. <code></summary></code> <code></instruction name="VerificaProduto"></code>

XML Documentation

	<pre> </summary> <instruction name="BiparEndereco"> <summary> Procedimento com a finalidade de permitir ao usuário bipar o endereço do produto. </summary> <instruction name="BiparEAN"> <summary> Procedimento com a finalidade de permitir ao usuário bipar o código EAN do produto. </summary> <instruction name="BiparEAN"> <summary> Procedimento com a finalidade de permitir ao usuário bipar o código EAN do produto. </summary> <instruction name="SelecionaUnidade"> <summary> Procedimento com a finalidade de permitir ao usuário informar a unidade do produto. </summary> </pre>
<param>	<pre> <param name="Endereco" type="String"> <para>Parâmetro de entrada/saída, utilizado para informar o endereço do produto que está sendo verificado.</para> </param> <param name="idProdutoEAN" type="Integer"> <para>Parâmetro de entrada/saída, utilizado para informar o código do produto relacionado ao EAN bipado.</para> </param> <param name="EAN" type="String"> <para>Parâmetro de entrada/saída, utilizado para informar o código EAN do produto que está sendo verificado.</para> </param> <param name="idProdutoEAN" type="Integer"> <para>Parâmetro de entrada/saída, utilizado para informar o código do produto relacionado ao EAN bipado.</para> </pre>

XML Documentation

	<p></param></p> <p><param name="idUnidade" type="Integer"></p> <p> <para>Parâmetro de entrada/saída, utilizado para informar o código da unidade do produto.</para></p> <p></param></p> <p><param name="idProduto" type="Integer"></p> <p> <para>Parâmetro de entrada, utilizado para informar o código do produto.</para></p> <p></param></p>
<returns>	
<exception>	
<remarks>	<p><remarks></p> <p> Informações complementares:</p> <p> <para>- Durante a verificação do produto, caso o EAN bipado pelo usuário não exista na base de dados, este EAN será cadastrado.</para></p> <p></remarks></p> <p><remarks></p> <p> Informações complementares:</p> <p> <para>- Só será permitido bipar endereços válidos e previamente cadastrados.</para></p> <p></remarks></p>