

Validando campos ou dados via PEX

Este tópico tem a finalidade de abordar os principais validadores presentes no PEX, como bem sabemos, o PEX nas versões maiores que 12 possuem um assistente PEX , no PEX existem também muitas rotinas e/ou funções publicadas que podem ser acessadas via menu de contexto, conforme imagem abaixo:

image-1640261677944.png

As funções abordadas serão:

```
TSTR.Testa_CNPJ(Codigo);
TSTR.Testa_CPF(Codigo);
TSTR.of_IsNotNullEMA(asText);
TSTR.of_IsNullEMA(asText);
TSTR.of_VerificaEmail(asTexto, abVerificaNull);
TSTR.of_TemNumero(s);
TSTR.of_TemTexto(s, abValida);
TSTR.of_UFValido(asUF);
TSTR.of_IsCodigoBarra(Codigo);
```

Valida CNPJ

Esta função deve ser utilizada com uma variável do PEX, variável do processo e/ou formulário do processo afim de validar um CNPJ, retornando False (Booleana) no caso de digitado um CNPJ inválido por engano ou algo do gênero.

```
var
    CNPJ : string;
    Resposta : Boolean;

begin

    CNPJ := '07297774000175'; //CNPOJ Informado em um formulario e/ou variavel do tipo texto
    Resposta := TSTR.Testa_CNPJ(CNPJ); -- Retorna True ou False para varaivel boleana

end;
```

Valida CPF

Esta função deve ser utilizada com uma variável do PEX, variável do processo e/ou formulário do processo afim de validar um CPF, retornando False (Booleana) no caso de digitado um CPF inválido por engano ou algo do gênero.

```
var
    CPF : string;
    Resposta : Boolean;

begin

    CPF:= '06972928840'; //CPF Informado em um formulario e/ou variavel do tipo texto
    Resposta := TSTR. Testa_CPF(CPF); -- Retorna True ou False para varaivel boleana

end;
```

Ou, dessa forma:

```
const
    cs_cpf = '7'; //Aqui é informada a ID referente ao campo.
var
    lsCPF : string;
begin

    lsCPF := aoFormularios. GetJSON(cs_cpf). GetStr(' TEXTO' );
    lsCPF := TSTR. StrTran(lsCPF, '-', '');
    lsCPF := TSTR. StrTran(lsCPF, '.', '');

    // Esta parte abaixo, formata a mensagem de erro caso aconteça

    if (not TSTR. Testa_CPF(lsCPF)) and (aoFormularios. GetJSON(cs_cpf). GetBol(' OBRIGATORIO' ) =
true) then
        begin

            aoMensagem. SetStr(' MENSAGEM' , ' CPF INVÁLIDO!' );
            aoMensagem. SetBol(' ABORTA' , True);
            aoMensagem. SetInt(' FOCO' , 9);
            aoMensagem. SetInt(' TIMEOUT' , 6000{Milissegundos});
            aoMensagem. SetStr(' TIPO' , ' ERRO' );
```

```
end;
```

```
end;
```

Verifica se Campo ou Variável é null / não preenchido:

```
IF TSTR.of_IsNotNulleMA(variavelpex) THEN // Como retorna uma boleada, podemos utializar  
diretamente no IF, assim se a variavel nao for vazia, a condicional será verdadeira.
```

```
IF TSTR.of_IsNulleMA(variavelpex) THEN // Como no exemplo acima, neste caso a condicional  
será verdadeira se a variavel estiver em branco/nula.
```

Valida E-mail :

Esta função verifica se o e-mail preenchido na variável ou campo é válido, ou seja, possui pelo menos um @ e ponto, e tem a opção de validar ou não se o campo estiver em branco nos casos de e-mail obrigatório:

```
//Padrão  
TSTR.of_VerificaEmail(asTexto, abVerificaNull);  
  
//Exemplo  
TSTR.of_VerificaEmail(VariavelDopex, True);
```

Ou seja, estamos verificando o conteúdo da variável e no segundo parâmetro estou verificando se não foi deixado em branco.

Como podemos perceber os nomes são bem intuitivos e podem ser facilmente interpretados, em todos os casos citados o retorno da função será uma booleana, ou seja, True/False, abaixo demais exemplos ainda não citados:

```
TSTR.of_TemNumero(variavel); // Verifica se possuem numeros na variavel.  
TSTR.of_TemTexto(variavel); // Verifica se possuem caracteres texto na variavel.  
TSTR.of_UFValido(variavel); // Verifica se a UF preenchida é uma UF válida dentro do  
Brasil.  
TSTR.of_IsCodigoBarra(variavel); // Verifica se código de barras é válido.
```

Revisão #4

Criado 23 December 2021 09:10:47 por Nicolly Andrielly

Atualizado 6 April 2022 09:07:47 por Nicolly Andrielly