

PEX - Preencher grade com informações de formulários

No **DOX** existe a possibilidade de inserir registros em uma grade de dados por meio de outros campos da mesma atividade. Para fazer isso, utilizamos o recurso **PEX** ao sair de um botão, que insere os dados digitados em um campo texto na grade de dados.

[Clique aqui](#) para baixar o processo deste exemplo

Neste exemplo, há um campo para digitar a chave de acesso de uma nota fiscal eletrônica, um botão para executar a inserção e a grade, onde será gravado as informações de cada nota informada.

[image.1649170072682.jpg](#)

Crie os seguintes formulários:

- Dissertativa "Chave de acesso" - /*CHAVEACESSO*/ (Texto)
- Botão "inserir"
- Grade de dados "Notas informadas" - /*GRADENOTAS*/ (Grade de dados)

[image.1649245013531.png](#)

No formulário de grade crie as colunas de acordo com a imagem abaixo:

A grade foi configurada com os campos de informações que contém na chave de acesso, incluindo um campo invisível para a chave.

[image.1649245074939.png](#)

O código PEX foi criado no formulário do tipo **Botão**. Foi utilizado recurso para percorrer o registro de grade para evitar duplicidade de registros, algumas validações das informações da chave de acesso e o recurso de inserir novo registro em grade.

```
const
    csChave = '1'; //código do formulário da chave de acesso
    csGrade = '3'; //código do formulário da grade de
notas
var
    lsChave : string;
    lsCnpj : string;
```

```

lsSerie : string;
lsNumeroNf : string;
lsEmissao : string;
liIdRegistro: Integer;
loRegistro: TJSONObject;
loNovoRegistro : TJSONObject;
liSequencia : Integer;

// declara as variáveis e seus respectivos tipos
begin
    //verifica se chave está completa
    if Length(aoFormularios.GetJSON(csChave).GetStr(' TEXTO' )) <> 44 then
        begin
            aoMensagem.SetStr(' MENSAGEM', ' Erro! Chave inválida.' );
            aoMensagem.SetStr(' TIPO', ' ERRO' );
            aoMensagem.SetInt(' TIMEOUT', 6000);
            exit;
        end;

        lsChave := aoFormularios.GetJSON(csChave).GetStr(' TEXTO' );

        //verifica se a nota já foi inserida
        for liIdRegistro := 0 to Pred(aoFormularios.GetJSON(' 3' ).GetArrayJSON(' DADOS' ).Count)
        do
            begin
                loRegistro :=
aoFormularios.GetJSON(' 3' ).GetArrayJSON(' DADOS' ).GetItemAsJson(liIdRegistro);

                if loRegistro.GetStr(' CHAVE' ) = lsChave then
                    begin
                        aoMensagem.SetStr(' MENSAGEM', ' Erro! Nota já inserida.' );
                        aoMensagem.SetStr(' TIPO', ' ERRO' );
                        aoMensagem.SetInt(' TIMEOUT', 6000);
                        exit;
                    end;
                end;

                //verifica se CNPJ é válido
                if not TSTR.Testa_CNPJ(copy(lsChave, 7, 14)) then
                    begin
                        aoMensagem.SetStr(' MENSAGEM', ' Erro! CNPJ inválido.' );
                        aoMensagem.SetStr(' TIPO', ' ERRO' );
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        aoMensagem.SetInt(' TIMEOUT' , 6000);
        exit;
    end;

    lsEmissao := copy(lsChave, 5, 2) + '/'20' +
                copy(lsChave, 3, 2);

    //verifica se a data é válida
    if (StrToDate('01/' + lsEmissao) > date) or (StrToInt(copy(lsEmissao, 1, 2)) > 12)
then
    begin
        aoMensagem.SetStr(' MENSAGEM' , 'Erro! Data de emissão inválida. ');
        aoMensagem.SetStr(' TIPO' , 'ERRO' );
        aoMensagem.SetInt(' TIMEOUT' , 6000);
        exit;
    end;

    //formata CNPJ
    lsCnpj := copy(lsChave, 7, 2) + '.' +
              copy(lsChave, 9, 3) + '.' +
              copy(lsChave, 12, 3) + '/' +
              copy(lsChave, 15, 4) + '-' +
              copy(lsChave, 19, 2);

    //retira zero da frente do número da nota e da série
    lsSerie := IntToStr(StrToInt(copy(lsChave, 23, 3)));
    lsNumeroNf := IntToStr(StrToInt(copy(lsChave, 26, 9)));

    //incrementa sequencial
    liSequencia := 1;

    for liIdRegistro := 0 to Pred(aoFormularios.GetJSON('3').GetArrayJSON('DADOS').Count)
do
    begin
        loRegistro :=
aoFormularios.GetJSON('3').GetArrayJSON('DADOS').GetItemAsJson(liIdRegistro);

        if loRegistro.GetInt(' SEQUENCIA' ) >= liSequencia then
        begin
            liSequencia := loRegistro.GetInt(' SEQUENCIA' )+1;
        end;
    end;
end;

```

```
end;

//insere informações na grade
loNovoRegistro := TJSONObject.Create;
loNovoRegistro.AddPair(' NUMERONF', lsNumeroNf);
loNovoRegistro.AddPair(' SERIE', lsSerie);
loNovoRegistro.AddPair(' EMISSAO', lsEmissao);
loNovoRegistro.AddPair(' CNPJEMITENTE', lsCnpj);
loNovoRegistro.AddPair(' CHAVE', lsChave);
loNovoRegistro.AddPair(' SEQUENCIA', IntToStr(liSequencia));
aoFormularios.GetJSON(csGrade).GetArrayJSON(' DADOS' ).Add( loNovoRegistro);

//limpa campos de chave de acesso
aoFormularios.GetJSON(csChave).SetStr(' TEXT0', '');

end;
```

Revisão #11

Criado 5 Abril 2022 10:39:33 por Nicolly Andrielly

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