

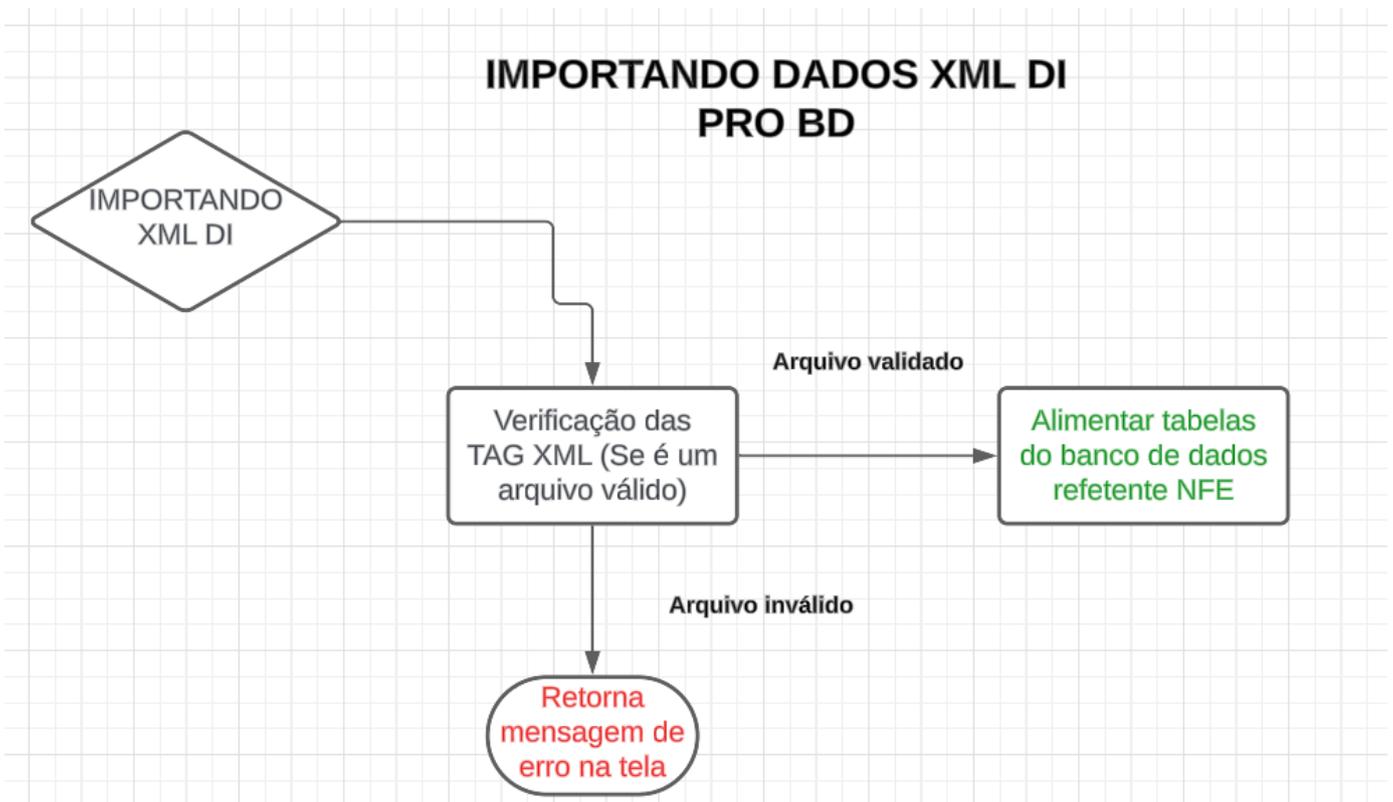
2. Validação XML DI

Softwaresistemaslogpe-ukh.com	Projeto/Sistema: Função importar DI XML - GA	Versão do Template: 1.1
	Processo: Importando dados XML DI no GA	Versão do Documento: 1.0
	Responsável(eis): Gustavo Fernandes	Data: 02/10/2024

1. Introdução

Este processo visa exemplificar como deve ser validado os dados do XML DI para o sistema GA.

2. Processo validação dados XML DI



I - Fluxograma Exemplo

Processo	Envolvidos	Dados de entrada	Dados de Saída
Importando dados XML DI para BD	XML DI e Sistema	Verificação das TAG XML (Se é um arquivo válido)	Retorna se é válido ou não arquivo.
Arquivo inválido	XML DI e Sistema	Validando arquivo	Retorna mensagem de erro da tela
Arquivo validado	XML DI e Sistema	Validando arquivo	Alimentar tabelas do BD referente a NFe

Processos Relacionados

[Protótipo de tela para importar DI](#)

Especificação Funcional

PROC002 VERIFICAÇÃO DAS TAG XML:

Este procedimento tem por objetivo demonstrar como será a validação XML DI.

Segundo as regras de validação: [Documentação de Referência do Siscomex DI](#)

Protótipo de Tela:

Sucesso! XML carregado

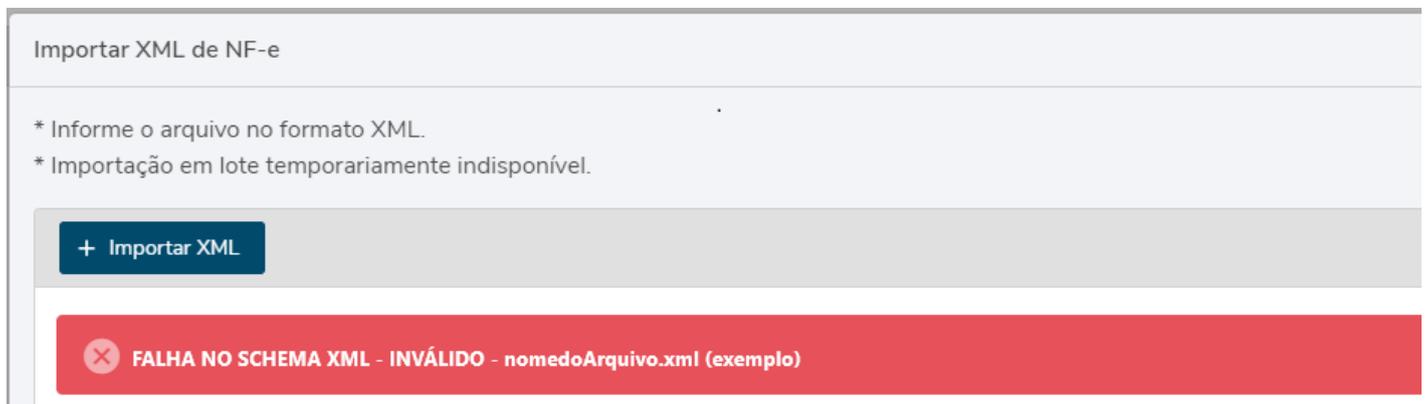
Série	N° Fiscal	CFOP	Cliente	Dt. Emissão	Finalidade
Nenhum registro foi encontrado					

Importar XML DI

Deseja importar o emitente dessa nota como destinatário (Cliente)?

Sim Não

I - Tela XML DI - retorno XML válido



II - Tela XML DI - retorno XML inválido

Regras de Interface:

1. Seguir as mesmas regras já determinadas no GA para importação de XML, contudo, validar se o SCHEMA XML esta válido.

1.2 Se estiver válido, seguir rotina normal de importação XML.

1.3 Se estiver inválido, mostrar na tela a FALHA NO SCHEMA XML, se possível retornar qual TAG esta com erro.

Regras de Negócio:

1. Ao importar o XML DI deverá ser validado se esta no padrão correto, com as informações obrigatório.

ARQUIVO DI.XML (modelo)

1.2 Se estiver válido, seguir rotina normal de importação XML.

1.3 Se estiver inválido, parar importação.

2. Após validar arquivo, alimentar as tabelas do banco de dados de TAG XML para campo respectivo.

Questões Técnicas:

1. Alimentar os dados referente a NFe temporária, conforme a tela:

Importar XML DI para NFe

- Seguir numeração do sistema
- Importar com data atual (Emissão e Saída)

Chave de Acesso

NFe35241007587030000195550010000000241118036452

Número

24

Valor

R\$

124.917,95

Remetente

NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FI!

Destinatário

GENUINE AUTOPARTS LTD

Itens da NF-e

Descrição	Unid.	Quant.	VI. Unitário	VI. Total	Selecione o Produto
6PK986 - CORREIA POLY V (CONTITECH) BMW 116I 11-17	PC	5,00	15,34	76,70	Cadastrar Produto

Fechar

Importar

1.2 ESPECIFICAÇÃO DOS CAMPOS

TAG DO XML	CAMPO DO SISTEMA	VALOR DO CAMPO	OBS
-	Chave de acesso	NFe35241007587...	Seguir mesmo processo padrão importação XML, gerando a chave de acesso automático!
-	Número	24	Seguir numeração sequência do sistema, por padrão. (Explicação a baixo)
-	Valor	124.917,95	Cálculos no código para chegar no valor total do item, para calcular e chegar no VI.Total da NFe
-	Remetente	NF-E EMITIDA EM AMBIENTE...	Seguir padrão do sistema
<fornecedorNome>	Destinatário	GENUINE AUTOPARTS LTD	Nome fornecedor do XML
<descricaoMercadoria>	Descrição	6PK986 - CORREIA POLY V...	Descrição item do XML
<unidadeMedida>	Unid.	PC	Descrição und do XML
<quantidade>	Quant.	5,00	Quant do XML

<valorUnitario>	VL.Unitário	15,34	Cálculos no código para chegar no vl.unitário, explicação a baixo.
-	VI. Total	76,70	Cálculos no código para chegar no vl.Total, explicação a baixo.

1.3 Como chegar no vl.unitário/vl.total do item, exemplo do que é feito no SIEM:

```

}
else if (tagXML == "valorUnitario")
{
    produtos.Adicao = Adicao;
    produtos.Descricao = Descricao;
    produtos.Sequencia = Seq;
    produtos.Qtd = Qtd;
    produtos.UND = Und;
    produtos.Ncm = pNcm;
    vUni = Div7(lerXml.Value);
    produtos.Valor = vUni;
    vTotal = (Qtd * vUni) + vTotal;
    if (bcP == 0) bcP = vUni * Qtd;
    ListProdutos.Add(produtos);
}

```

```

private decimal Div7(string QtdOrig)
{
    decimal QtdNova = (Convert.ToDecimal(QtdOrig) / 10000000);
    return QtdNova;
}

```

```
vUni = Div7(lerXml.Value);
```

A função `Div7` é chamada, e ela recebe um valor vindo do XML (`lerXml.Value`).

A função `Div7` recebe uma string (`QtdOrig`), que será convertida para decimal e dividida por 10.000.000:

```

private decimal Div7(string QtdOrig)
{
    decimal QtdNova = (Convert.ToDecimal(QtdOrig) / 10000000);
    return QtdNova;
}

```

O objetivo dessa divisão por 10.000.000 é ajustar o valor bruto vindo do XML para um valor mais preciso e utilizável. (provavelmente o valor está em uma unidade diferente e precisa ser ajustado).

1.4 Cálculo do valor total (vTotal):

- Após calcular o valor unitário ajustado ($vUni$), o código calcula o valor total do produto

```
vTotal = (Qtd * vUni) + vTotal;
```

Aqui, ele multiplica a quantidade (Qtd) pelo valor unitário ($vUni$) e soma ao valor total já existente.

- **Condição de BCP:**

- Se o valor de bcp for igual a zero, ele recalcula o valor de bcp como $vUni * Qtd$.

- **Adição do produto à lista:**

- Por fim, o produto com todos os atributos e cálculos é adicionado à lista

$ListProdutos$.

OBS IMPORTANTE: No momento este cálculo para chegar no vl.unitário foi o mais perto que chegamos, ainda estamos analisando o código do SIEM e documentações online

©SOFTEN SISTEMAS 2024

Revision #32

Created 2 October 2024 14:30:26 by Gustavo Henrique Braga Fernandes

Updated 31 October 2024 16:55:30 by Gustavo Henrique Braga Fernandes