

1. Consulta de Preços

1.1. Especificação do Arquivo Via Texto

1.1.1. Referência:

Consulta de Preços na Autorização

1.1.2. Objetivo

Permitir a consulta do Preço Praticado no Varejo no momento da Autorização através da consulta de um arquivo texto contendo a lista dos produtos a consultar.

1.1.3. Descrição

No momento da Autorização é necessário consultar o Preço Praticado no Varejo, ou seja, o preço da loja, considerando inclusive as promoções.

1.1.4. Definição do formato

Nome do Arquivo

Definido pelo desenvolvedor (Ex: Tabela de Preços.TXT)

Parâmetros de Chamada (conteúdo do arquivo)

- a) Código do Produto (EAN ou interno do varejo) - Numérico (13)
- b) Descrição do Produto - Alfanumérico (40), alinhado à esquerda com espaços à direita, Se não for possível fornecer uma descrição, preencher com "X"
- c) Preço Máximo ao Consumidor - Numérico (7), em centavos, com zeros à esquerda. Quando não disponível retorna zeros
- d) Preço Praticado no Varejo - Numérico (7), em centavos, com zeros à esquerda. Preço de venda na loja considerando as promoções. Nunca pode ser zero.
- e) Código da Categoria - Numérico (8), com zeros à esquerda. Para implementação futura, retornar sempre zeros.

Caminho da Consulta (diretório)

Definido pelo desenvolvedor

Acesso no aplicativo POS:

- Modo de operação
- Consulta de Preços
- Importação de Arquivo
- Nome do arquivo de importação de preços (Ex: C:\e-Pharma\Tabela_de_preços.txt).

Observações

- As linhas de textos devem ser divididas pelos caracteres 0Dh e 0Ah (CR e LF na tabela ASCII).
- O arquivo deve ser ordenado pelo código de barras do menor para o maior.
- **O** preço de venda da loja não pode ser zero.

1.2. Especificação do via DLL

1.2.1. Referência:

Consulta de Preços na Autorização

1.2.2. Objetivo

Permitir a consulta do Preço Praticado no Varejo no momento da Autorização a partir do aplicativo e-Pharma diretamente ao sistema de frente de loja.

1.2.3. Descrição

No momento da Autorização é necessário consultar o Preço Praticado no Varejo, ou seja o preço da loja, considerando inclusive as promoções. Nas redes de varejo onde o software utilizado no PDV não possibilitar o desenvolvimento de uma interface para esta consulta foi criado um acesso padrão, via DLL, que permite a estas aplicações disponibilizar estes dados no ato da autorização.

1.2.4. Definição do acesso

- Parâmetro de Chamada: Estrutura conforme descrição abaixo preenchida com o código EAN do produto.
- Parâmetro de Retorno: Valor numérico indicando o sucesso (0 = zero) ou falha (diferente de zero) da pesquisa.

Informações da Estrutura de Chamada e Retorno

- Código do Produto (EAN ou interno do varejo)
 - Numérico (13), com zeros a esquerda
 - O mesmo recebido como parâmetro
- Descrição do Produto
 - Alfanumérico (40), alinhado à esquerda com espaços à direita
 - Se não for possível fornecer uma descrição, preencher com "X"
- Preço Máximo ao Consumidor
 - Numérico (7), em centavos, com zeros a esquerda
 - Quando não disponível retorna zeros
- Preço Praticado no Varejo
 - Numérico (7), em centavos, com zeros a esquerda
 - Preço de venda na loja considerando as promoções
- Código da Categoria
 - Numérico (8), com zeros a esquerda
 - Para implementação futura, retornar sempre zeros.

1.2.5. Observações

Quando o produto não for localizado na base da loja a função deve retornar TRUE (diferente de zero), neste caso o conteúdo dos campos da estrutura de retorno é desprezado.

Exemplo de formato dos preços: R\$ 75,49 retorna 0007549.

Definição das Informações

Código do produto EAN – Código EAN do Produto.

Descrição do produto – Descrição do Produto + Apresentação.

Preço Máximo ao Consumidor – em centavos, com zeros à esquerda.

Preço Praticado pelo Varejo no momento da transação – em centavos, com zeros a esquerda.

Categoria do produto – quando não utilizado deve ser zeros.

```
// Declaração da função da DLL
```

```
function PrecoEPharma(var Produto : TProduto) : ShortInt; stdcall; export;
```

O formato desta estrutura deve ser o seguinte, em C:

```
typedef struct {
    WORD wSize;
    char szEAN[13]; // Código do produto EAN
    char szDescricao[40]; // Descrição do produto
    char szPrecoMaximo[7]; // Preço Máximo ao Consumidor
    char szPrecoVarejo[7]; // Preço Praticado pelo Varejo no momento da transação
    char szCategoria[8]; // Categoria do produto
} PRECO;
```

Em VB ela fica:

```
Structure qual_quer_nome
    iSize as short
    szEAN as string * 13 ' Código do produto EAN
    szDescricao * 40 ' Descrição do produto
    szPrecoMaximo as string * 7 ' Preço Máximo ao Consumidor
    szPrecoVarejo as string * 7 ' Preço Praticado pelo Varejo no momento da transação
    szCategoria as string * 8 Categoria do produto
End Structure
```

A estrutura e a declaração da função em Delphi:

```
// Declaração da Estrutura
```

```
TProduto = Record
Tamanho: Integer;
Cod_Barra : array [1.. 13] of char;
// Código do produto EAN
Nom_Produto : array [1..40] of char; // Descrição do produto
Vlr_Unitario : array [1..7] of char; // Preço Máximo ao Consumidor
Vlr_Venda : array [1..7] of char; // Preço Praticado pelo Varejo no momento da transação
Tip_Categoria : array [1..8] of char; // Categoria do produto end;
```

O conteúdo do campo iSize será sempre o tamanho total da estrutura e szEAN conterá o código do produto.

A DLL pode ter qualquer nome, bastará indicar o mesmo no campo correspondente, na tela de configuração de modo de operação, se não for indicado o nome, apenas o diretório, o programa assume como sendo o nome "PrecoEPharma.dll"

A função interna da DLL deve-se chamar "PrecoEPharma", respeitando este formato de maiúsculos e minúsculos e utilizará como parâmetro a estrutura.

Ela deve retornar ZERO se funcionou corretamente, ou diferente de zero se deu erro, se o valor for diferente de zero, poderá ser um indicador do tipo de erro que ocorreu, este dado será gravado no log, para consulta.