

AMANUAL DE INTEGRACION API FISCAL

PNPDLL.DLL

V1.8

Cubre los modelos

PF-950

PF-675

PF-220-II

PF-250

PF-300-II



Escanea con tu celular este código para que agregues nuestros datos

URL: www.desarrollospnp.com E-mail: integracion@abacco.com

Todos los derechos reservados. Ninguna parte de este documento podrá ser reproducida, almacenada en un sistema que permita su recuperación posterior o transmitida en cualquier forma o por cualquier medio sin previa autorización escrita de DESARROLLOS PNP, C.A.. Si bien se han tomado todas las precauciones del caso en la preparación de este libro, DESARROLLOS PNP, C.A. no asume ninguna responsabilidad por errores u omisiones; tampoco se asume ninguna responsabilidad por daños y perjuicios resultantes del uso de la información contenida en el presente manual.



REVISIONES

Revisiones	
Fecha	Observaciones
17/11/2010	Revisión general. H1.
29/11/2010	Revisión general. Se incluye función de consulta.
19/06/2012	Revisión General. Se anexa manejo de funciones con descripción y nombre de empresas ampliadas. Se maneja descripciones de hasta 140 caracteres en 3 líneas y razones sociales de máx 80 caracteres en 2 líneas.
02/07/2012	Se corrige en el manual el estatus T y V que no estaban documentados aunque si aparecen en el manual de protocolo.



INTRODUCCIÓN

Este documento contiene la descripción de todas las funciones manejadas por la librería cargada dinámicamente (dll) pfdll.dll versión 1.8

Este dll les una librería cargable dinámicamente que no requiere registro alguno solo debe estar accesible a la aplicación que la importe sea en su mismo directorio o en la ruta path definida en el equipo.

Para su correcto funcionamiento es recomendable que sea cargada una única vez por la aplicación que la utiliza y mantenerla enlazada hasta el cierre de la misma.

Si requiere soporte adicional sobre el manejo de la misma o sugerencias que le hagan más sencillo su uso, no dude en contactarnos por el email integracion@abacco.com

Para facilitar el manejo general de aplicaciones todas las respuestas de las funciones son tipo texto (string) y están basadas en el manual de protocolo de nuestros equipos que pueden conseguir en nuestra página www.desarrollosnpn.com

La mayoría de las funciones tiene como respuesta tres textos: OK si la transmisión / recepción fue correcta, ER si hay presencia de error y TO si se excedió el tiempo de respuesta esperado del equipo.

El paquete incluye una versión de desarrollo que le permite integrar los comandos en su aplicación sin la necesidad de conectar un equipo fiscal y obteniendo en pantalla un resultado parecido al presentado en la impresora.





TABLA DE CONTENIDO

Declaracion de funciones en Delphi	7
Declaracion de funciones en C#	8
Declaracion de funciones en VISUAL BASIC.....	10
Declaracion de funciones en FoxPRO	12
FUNCION PFAbreFISCAL:string;	13
FUNCION PFAbreNF:string;	14
FUNCION PFAbrepuerto(numero:string):string;	15
FUNCION PFBarra(edbarra:string):string;	16
FUNCION PFcaNCELADOC(MOD,MON:string):string;	18
FUNCION PFcambiatasa(t1,t2,t3:string):string;	19
FUNCION PFcambiofecha(edfecha,edhora:string):string;	20
FUNCION PFCambtipoContrib(tip:string):string;	21
FUNCION PFCheque2(mon,ben,fec,c1,c2,c3,c4,campo1,campo2:string):string;	22
FUNCION PFCierraNF:string;	23
FUNCION PFCortar:string;	24
FUNCION PFDevolucion(razon,rif,comp,maqui,fecha,hora:string):string;	25
FUNCION PFDisplay950(edlinea:string):string;	26
FUNCION PFendoso(campo1,campo2,campo3,tiposendoso:string):string;	27
FUNCION PFestatus(edlinea:string):string;	28
FUNCION PFGaveta:string;	31
FUNCION PFLeereloy: STRING;	32
FUNCION PFLINEANF (edlinea: STRING): STRING;	33
FUNCION PFLogo:string;	34
FUNCION PFparcial:string;	35
FUNCION PFrenglon(Descripcion,cantidad,monto,iva:string):string;	36
FUNCION PFrepMemNF(desf,hasf,modmem:string):string;	37
FUNCION PFrepMemoriaNumero(desn,hasn,modmem:string):string;	38
FUNCION PFrepx:string;	39
FUNCION PFrepz:string;	40
FUNCION PFreset:string;	41
FUNCION PFSerial:string;	42
FUNCION PFSLIPOFF:string;	43
FUNCION PFSlipON:string;	44
FUNCION PFTfiscal(edlinea:string):string;	45
FUNCION PFTIPOIMP(MODELO:string):string;	46
FUNCION PFtotal:string;	47



FUNCION PFtoteconomico;	48
FUNCION PFultimo:string;	49
FUNCION PFvalida675(campo1,campo2,campo3,campo4:string):string;	50
FUNCION PFversion:string;	51
FUNCION PFVoltea:string;	52
SECUENCIA sugerido para generar una factura	53
SECUENCIA para la generacion de documento no fiscal.....	53
VERSIONES DISPONIBLES DE LA LIBRERIA	53



DECLARACION DE FUNCIONES

DECLARACION DE FUNCIONES EN DELPHI

```
function PFabrefiscal(Razon,RIF:pchar):pchar; stdcall external 'pnpdll.dll'
function PFTotal:pchar; stdcall external 'pnpdll.dll'
function PFRepz:pchar; stdcall external 'pnpdll.dll'
function PFRepX:pchar; stdcall external 'pnpdll.dll'
function PFrenlon(Descripcion,cantidad,monto,iva:pchar):pchar; stdcall external 'pnpdll.dll'
function PFabrepuerto(numero:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCierrapuerto:pchar; stdcall external 'pnpdll.dll'
function PFDisplay950(edlinea:pchar):pchar; stdcall external 'pnpdll.dll'
function PFAbreNF:pchar; stdcall external 'pnpdll.dll'
function PFLineaNF(edlinea:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCierraNF:pchar; stdcall external 'pnpdll.dll'
function PFDescuento(edbarra,monto:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCortar:pchar; stdcall external 'pnpdll.dll'
function PFTfiscal(edlinea:pchar):pchar; stdcall external 'pnpdll.dll'
function PFparcial:pchar; stdcall external 'pnpdll.dll'
function PFSerial:pchar; stdcall external 'pnpdll.dll'
function PFToteconomico:pchar; stdcall external 'pnpdll.dll'
function PFCancelaDoc(edlinea,monto:pchar):pchar; stdcall external 'pnpdll.dll'
function PFGaveta:pchar; stdcall external 'pnpdll.dll'
function PFDevolucion(razon,rif,comp,maqui,fecha,hora:pchar):pchar; stdcall external 'pnpdll.dll'
function PFSlipON:pchar; stdcall external 'pnpdll.dll'
function PFSLIPOFF:pchar; stdcall external 'pnpdll.dll'
function PFestatus(edlinea:pchar):pchar; stdcall external 'pnpdll.dll'
function PFreset:pchar; stdcall external 'pnpdll.dll'
function PFendoso(campo1,campo2,campo3,tipoendoso:pchar):pchar; stdcall external
'pnpdll.dll'
function PFvalida675(campo1,campo2,campo3,campo4:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCheque2(mon,ben,fec,c1,c2,c3,c4,campo1,campo2:pchar):pchar; stdcall external
'pnpdll.dll'
function PFCambiofecha(edfecha,edhora:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCambiatasa(t1,t2,t3:pchar):pchar; stdcall external 'pnpdll.dll'
function PFBarra(edbarra:pchar):pchar; stdcall external 'pnpdll.dll'
function PFVoltea:pchar; stdcall external 'pnpdll.dll'
function PFLeereloy:pchar; stdcall external 'pnpdll.dll'
function PFRepMemNF(desf,hasf,modmem:pchar):pchar; stdcall external 'pnpdll.dll'
function PFRepMemoriaNumero(desn,hasn,modmem:pchar):pchar; stdcall external 'pnpdll.dll'
function PFCambtipoContrib(tip:pchar):pchar; stdcall external 'pnpdll.dll'
function PFTultimo:pchar; stdcall external 'pnpdll.dll'
function PFTipoImp:pchar; stdcall external 'pnpdll.dll'
```



DECLARACION DE FUNCIONES EN C#

```
using System;
using System.Runtime.InteropServices;
namespace FiscalPrinterPnP
{
    /// <summary>
    /// Clase con la declaración de las funciones PNPDLL.dll
    /// </summary>

    public class PNPF132
    {
        public PNPF132()
        {
        }
    }

    #region DECLARACIÓN DE LAS FUNCIONES de PNPDLL.DLL

    [DllImport("PNPDLL.dll")] public static extern string PFAbreNF();
    [DllImport("PNPDLL.dll")] public static extern string PFAbreNFiscal(String Razon, String RIF);
    [DllImport("PNPDLL.dll")] public static extern string PFTotal();
    [DllImport("PNPDLL.dll")] public static extern string PFRepz();
    [DllImport("PNPDLL.dll")] public static extern string PFRepX();
    [DllImport("PNPDLL.dll")] public static extern string PFReglon(String Descripcion, String
    cantidad, String monto, String iva);
    [DllImport("PNPDLL.dll")] public static extern string PFAbrepuerto(String numero);
    [DllImport("PNPDLL.dll")] public static extern string PFCierrapuerto();
    [DllImport("PNPDLL.dll")] public static extern string PFDisplay950(String edlinea);
    [DllImport("PNPDLL.dll")] public static extern string PFAbreNF();
    [DllImport("PNPDLL.dll")] public static extern string PFLineaNF(String edlinea);
    [DllImport("PNPDLL.dll")] public static extern string PFCierraNF();
    [DllImport("PNPDLL.dll")] public static extern string PFDescuento(String edbarra, String
    monto);
    [DllImport("PNPDLL.dll")] public static extern string PFCortar();
    [DllImport("PNPDLL.dll")] public static extern string PFTfiscal(String edlinea);
    [DllImport("PNPDLL.dll")] public static extern string PFparcial();
    [DllImport("PNPDLL.dll")] public static extern string PFSerial();
    [DllImport("PNPDLL.dll")] public static extern string PFToteconomico();
    [DllImport("PNPDLL.dll")] public static extern string PFCancelaDoc(String edlinea, String
    monto);
    [DllImport("PNPDLL.dll")] public static extern string PFGaveta();
    [DllImport("PNPDLL.dll")] public static extern string PFDevolucion(String razon, String rif,
    String comp, String maqui, String fecha, String hora);
    [DllImport("PNPDLL.dll")] public static extern string PFSlipON();
```




```
[DllImport("PNPDLL.dll")] public static extern string PFSLIPOFF();
[DllImport("PNPDLL.dll")] public static extern string PFestatus(String edlinea);
[DllImport("PNPDLL.dll")] public static extern string PFreset();
[DllImport("PNPDLL.dll")] public static extern string PFendoso(String campo1, String campo2,
String campo3, String tipoendoso);
[DllImport("PNPDLL.dll")] public static extern string PFvalida675(String campo1, String
campo2, String campo3, String campo4);
[DllImport("PNPDLL.dll")] public static extern string PFCheque2(String mon, String ben, String
fec, String c1, String c2, String c3, String c4, String campo1, String campo2);
[DllImport("PNPDLL.dll")] public static extern string PFCambiofecha(String edfecha, String
edhora);
[DllImport("PNPDLL.dll")] public static extern string PFCambiatasa(String t1, String t2, String
t3);
[DllImport("PNPDLL.dll")] public static extern string PFBarra(String edbarra);
[DllImport("PNPDLL.dll")] public static extern string PFVoltea();
[DllImport("PNPDLL.dll")] public static extern string PFLeerelej();
[DllImport("PNPDLL.dll")] public static extern string PFrepMemNF(String desf, String hasf,
String modmem);
[DllImport("PNPDLL.dll")] public static extern string PFrepMemoriaNumero(String desn, String
hasn, String modmem);
[DllImport("PNPDLL.dll")] public static extern string PFCambtipoContrib(String tip);
[DllImport("PNPDLL.dll")] public static extern string PFultimo();
[DllImport("PNPDLL.dll")] public static extern string PFTipoImp();

#endregion
}
}
```



DECLARACION DE FUNCIONES EN VISUAL BASIC

Funciones de PNPDLL.DLL

Public Declare Function Pfabrefiscal Lib "PNPDLL.DLL" (ByVal Razon As String, ByVal RIF As String) As String

Public Declare Function Pftotal Lib "PNPDLL.DLL" As String

Public Declare Function PFrepz Lib "PNPDLL.DLL" As String

Public Declare Function PFrepx Lib "PNPDLL.DLL" As String

Public Declare Function PFrenglon Lib "PNPDLL.DLL"
(ByVal Descripcion As String, ByVal cantidad As String, ByVal monto As String, ByVal iva As String) As String

Public Declare Function Pfabrepuerto Lib "PNPDLL.DLL" (ByVal numero As String) As String

Public Declare Function Pfcierrapuerto Lib "PNPDLL.DLL" As String

Public Declare Function PFDisplay950 Lib "PNPDLL.DLL" (ByVal edlinea As String) As String

Public Declare Function PFAbreNF Lib "PNPDLL.DLL" As String

Public Declare Function PFLineaNF Lib "PNPDLL.DLL" (ByVal edlinea As String) As String

Public Declare Function PFCierraNF Lib "PNPDLL.DLL" As String

Public Declare Function PFDescuento Lib "PNPDLL.DLL" (ByVal edbarra As String, ByVal monto As String) As String

Public Declare Function PFCortar Lib "PNPDLL.DLL" As String

Public Declare Function PFTfiscal Lib "PNPDLL.DLL" (ByVal edlinea As String) As String

Public Declare Function PFparcial Lib "PNPDLL.DLL" As String

Public Declare Function PFSerial Lib "PNPDLL.DLL" As String

Public Declare Function Pftoteconomico Lib "PNPDLL.DLL" As String

Public Declare Function PFCancelaDoc Lib "PNPDLL.DLL" (ByVal edlinea As String, ByVal monto As String) As String

Public Declare Function PFGaveta Lib "PNPDLL.DLL" As String

Public Declare Function PFDevolucion Lib "PNPDLL.DLL" (ByVal razon As String, ByVal rif As String, ByVal comp As String, ByVal maqui As String, ByVal fecha As String, ByVal hora As String) As String

Public Declare Function PFSlipON Lib "PNPDLL.DLL" As String

Public Declare Function PFSLIPOFF Lib "PNPDLL.DLL" As String

Public Declare Function PFestatus Lib "PNPDLL.DLL" (ByVal edlinea As String) As String

Public Declare Function PFreset Lib "PNPDLL.DLL" As String

Public Declare Function PFendoso Lib "PNPDLL.DLL" (ByVal campo1 As String, ByVal campo2 As String, ByVal campo3 As String, ByVal tipoendoso As String) As String

Public Declare Function PFvalida675 Lib "PNPDLL.DLL" As String
(campo1, campo2, campo3, campo4: pchar; pchar; stdcall external 'pnpdll.dll')

Public Declare Function PFCheque2 Lib "PNPDLL.DLL" (ByVal mon As String, ByVal ben As String, ByVal fec As String, ByVal c1 As String, ByVal c2 As String, ByVal c3 As String, ByVal c4 As String, ByVal campo1 As String, ByVal campo2 As String) As String

Public Declare Function PFCambiofecha Lib "PNPDLL.DLL" (ByVal edfecha As String, ByVal edhora As String) As String

Public Declare Function PFCambiatasa Lib "PNPDLL.DLL" (ByVal t1 As String, ByVal t2 As String, ByVal t3 As String) As String



```
Public Declare Function PFBarra Lib "PNPDLL.DLL" (ByVal edbarra As String) As String
Public Declare Function PFVoltea Lib "PNPDLL.DLL" As String
Public Declare Function PFLeerelobj Lib "PNPDLL.DLL" As String
Public Declare Function PFrepMemNF Lib "PNPDLL.DLL" (ByVal desf As String, ByVal hasf As
String, ByVal modmem As String) As String
Public Declare Function PFrepMemoriaNumero Lib "PNPDLL.DLL" (ByVal desn As String, ByVal
hasn As String, ByVal modmem As String) As String
Public Declare Function PFCambtipoContrib Lib "PNPDLL.DLL" (ByVal tip As String) As String
Public Declare Function PFultimo Lib "PNPDLL.DLL" As String
Public Declare Function PFTipoImp Lib "PNPDLL.DLL" As String
```



DECLARACION DE FUNCIONES EN FOXPRO

* Funciones de pnpdll.dll

```
DECLARE STRING PFabrefiscal IN PNPDLL.DLL STRING,STRING
DECLARE STRING PFtotal IN PNPDLL.DLL
DECLARE STRING PFrepz IN PNPDLL.DLL
DECLARE STRING PFrepz IN PNPDLL.DLL
DECLARE STRING PFrenglon IN PNPDLL.DLL STRING,STRING,STRING,STRING
DECLARE STRING PFabrepuerto IN PNPDLL.DLL STRING
DECLARE STRING PFCierrapuerto IN PNPDLL.DLL
DECLARE STRING PFDisplay950 IN PNPDLL.DLL STRING
DECLARE STRING PFAbreNF IN PNPDLL.DLL
DECLARE STRING PFLineaNF IN PNPDLL.DLL STRING
DECLARE STRING PFCierraNF IN PNPDLL.DLL
DECLARE STRING PFDescuento IN PNPDLL.DLL STRING,STRING
DECLARE STRING PFCortar IN PNPDLL.DLL
DECLARE STRING PFTfiscal IN PNPDLL.DLL STRING
DECLARE STRING PFparcial IN PNPDLL.DLL
DECLARE STRING PFSerial IN PNPDLL.DLL
DECLARE STRING PFtoteconomico IN PNPDLL.DLL
DECLARE STRING PFCancelaDoc IN PNPDLL.DLL STRING,STRING
DECLARE STRING PFGaveta IN PNPDLL.DLL
DECLARE STRING PFDevolucion IN PNPDLL.DLL STRING, STRING, STRING, STRING,
STRING, STRING
DECLARE STRING PFSlipON IN PNPDLL.DLL
DECLARE STRING PFSLIPOFF IN PNPDLL.DLL
DECLARE STRING PFestatus IN PNPDLL.DLL STRING
DECLARE STRING PFreset IN PNPDLL.DLL
DECLARE STRING PFendoso IN PNPDLL.DLL STRING,STRING,STRING,STRING
DECLARE STRING PFvalida675 IN PNPDLL.DLL STRING,STRING,STRING,STRING
DECLARE STRING PFCheque2 IN PNPDLL.DLL
STRING,STRING,STRING,STRING,STRING,STRING,STRING,STRING,STRING,STRING
DECLARE STRING PFCambiofecha IN PNPDLL.DLL STRING,STRING
DECLARE STRING PFCambiatasa IN PNPDLL.DLL STRING,STRING,STRING
DECLARE STRING PFBarra IN PNPDLL.DLL STRING
DECLARE STRING PFVoltea IN PNPDLL.DLL
DECLARE STRING PFLeerelej IN PNPDLL.DLL
DECLARE STRING PFrepMemNF IN PNPDLL.DLL STRING,STRING,STRING
DECLARE STRING PFRepMemoriaNumero IN PNPDLL.DLL STRING,STRING,STRING
DECLARE STRING PFCambtipoContrib IN PNPDLL.DLL STRING
DECLARE STRING PFultimo IN PNPDLL.DLL
DECLARE STRING PFTipoImp IN PNPDLL.DLL
```



FUNCIONES EN ORDEN ALFABETICO

FUNCION PFABREFISCAL:STRING;

Descripción: Inicia documento fiscal

Parámetros: Razon: String de máximo 80 caracteres. El Api automáticamente genera líneas adicionales para aquellas razones sociales que superen los 38 caracteres de establecidos en la función por compatibilidad a modelos anteriores. Si el largo del parámetro supera estas dimensiones es truncado automáticamente.

RIF: String de máximo 12 caracteres

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Delphi

Respuesta:= PFAbreFISCAL(Razon,Rif);

VB

Respuesta = PFAbreFISCAL(Razon,Rif)

FoxPro

Respuesta = PFAbreFISCAL(Razon,Rif)



FUNCION PFABRENF:STRING;

Descripción: Inicia documento no fiscal

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Delphi

Respuesta:= PFAbreNF;

VB

Respuesta = PFAbreNF

FoxPro

Respuesta = PFAbreNF



FUNCION PFABREPUERTO(NUMERO:STRING):STRING;

Descripción: Abre puerto serial físico del equipo para comunicación con impresora. Se requiere ejecutarse solo una vez.

Parámetros: (string) Numero de puerto 1,2,3,4

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfabrepuerto(numero);



FUNCION PFBARRA(EDBARRA:STRING):STRING;

Descripción: Envía código de barras dependiendo del modelo de impresora.

Parámetros: (string) valor de la barra (código)

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfbarra(edbarra);





FUNCION PFCANCELADOC(MOD,MON:STRING):STRING;

Descripción: Cancela la emisión del documento activo actual. Genera el texto en la impresora de documento cancelado.

Parámetros: (string) MOD: Modo comando para cancelación de documento "C"

MON: Monto parámetro "0"

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfcanceladoc('C','0');



FUNCION PFCAMBIATASA(T1,T2,T3:STRING):STRING;

Descripción: Cambia las tasas activas del equipo. Se requiere la generación de un reporte Z antes de su ejecución de lo contrario presenta error.

Parámetros: (string) T1: Tasa GENERAL

en formato XXDD donde XX: Entero DD Decimal. Ej.: 1200 = 12%

T2: Tasa Reducida

T3: Tasa Aumentada

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfcambiatasa(t1,t2,t3);



FUNCION PFCAMBIOFECHA(EDFECHA,EDHORA:STRING):STRING;

Descripción: Cambia la fecha del equipo. Solo se puede adelantar los días y retrasar las horas. Requiere un reporte Z antes de su ejecución.

Parámetros: (string) EDFECHA: Fecha a cambiar en formato DDMMAA
(string) EDHORA: Hora a cambiar en formato HHMM

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Delphi

Respuesta:= pfcambiofecha(edfecha,edhora);



FUNCION PFCAMBTIPOCONTRIB(TIP:STRING):STRING;

Descripción: Cambia el tipo de contribuyente al equipo.

Parámetros: (string) Tipo de Contribuyente:

- 1:Ordinario
- 2:No sujeto al IVA
- 3:Formal

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfcambiotipocontrib(tip);



FUNCION

PFCHEQUE2(MON,BEN,FEC,C1,C2,C3,C4,CAMPO1,CAMPO2:STRING):STRING;

Descripción: Envía los datos para llenado de cheques en equipos que soportan impresión de cheques.

Parámetros: (string) MON: Monto en formato XXXXXXDD

BEN: Beneficiario

FEC: Fecha

C1,C2,C3,C4:Valores numéricos para cuadro de campos

CAMPO1,CAMPO2: Valores a emitir en el endoso para equipos que lo soporten

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfCheque2(mon,ben,fec,c1,c2,c3,c4,campo1,campo2);



FUNCION PFCIERRANF:STRING;

Descripción: Cierra documento no fiscal.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfCierranf;



FUNCION PFCORTAR:STRING;

Descripción: Envía comando corte de papel para equipos compatibles.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfCortar;



FUNCION

PFDEVOLUCION(RAZON,RIF,COMP,MAQUI,FECHA,HORA:STRING):STRING;

Descripción: Abre documento fiscal de nota de crédito/devolución.

Parámetros: (string) RAZON: Nombre de la razón social

RIF: Numero de RIF sin espacios ni guiones del cliente.

COMP: Numero de factura objeto de la nota de crédito

MAQUI: Serial del equipo fiscal que generó la factura.

FECHA: Fecha de la facture original en formato DDMMAA

HORA: Hora en formato HHMM

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfDevolucion(razon,rif,comp,maqui,fecha,hora);



FUNCION PFDISPLAY950(EDLINEA:STRING):STRING;

Descripción: Genera texto en el visor del cliente conectado en los modelos de equipo compatibles. PF-950

Parámetros: (string) EDLINEA: Línea de texto a enviar?

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfDisplay950(edlinea);



FUNCION PFENDOSO(CAMPO1,CAMPO2,CAMPO3,TIPOENDOSO:STRING):STRING;

Descripción: Emite datos de endoso en cheques.

Parámetros: (string) CAMPO 1,2,3: Textos a generar en el endoso.

TIPOENDOSO: Tipo del endoso. Para depósitos o cobros

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfendoso(campo1,campo2,campo3,tipoendoso);



FUNCION PFESTATUS(EDLINEA:STRING):STRING;

Descripción: Solicita el estatus fiscal del equipo.

Parámetros: (string) EDLINEA: Letra clave del tipo de estatus solicitado.

'N'= datos de los contadores
'E'= valor de ventas exentas
'A'= valor de base imponible a + impuestos
'B'= valor de base imponible b + impuestos
'C'= valor de base imponible c + impuestos
'D' = valor de descuentos + impuestos
'R'= valor de devoluciones + impuestos
'F'= near end papel de recibo
'J'= near end papel de auditoría.
'S'= desactiva el autocutter.
'T'= Numero de última nota de crédito generada.
'U'= Verifica si hay papel en el slip de la impresora.
'V'= Envía versión del FW del equipo.

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Respuestas Obtenidas en función ultima separados por coma.

RESPUESTA AL COMANDO STATUS IF (0X38) CON EL PARÁMETRO 'N'

Campo 1 Estado Impresora <0000>
Campo 2 Estado Fiscal <0000>
Campo 3 Ultimo valor de Secuencia
Campo 4 Código del status actual de la impresora
Campo 5 Código último comando ejecutado
Campo 6 Fecha en la IF <AAMMDD>
Campo 7 Hora en la IF <HHMMSS>
Campo 8 # Factura fiscal del periodo fiscal
Campo 9 # DNF del periodo fiscal
Campo 10 # Factura fiscal acumulado
Campo 11 # DNF acumulado
Campo 12 # último reporte Z

Detalle de la respuesta al comando PFESTATUS con el parámetro "N":



Campo 3: Contiene el valor de secuencia almacenado en la memoria fiscal que será utilizado para validar el próximo comando. Se puede enviar el comando STATUS para fijar un valor de secuencia en la impresora fiscal.

Campo 4: Contiene un código hexadecimal de dos bytes que indica el estado actual de la impresora:

_ 00 = Impresora lista para abrir una factura, abrir un documento no fiscal, hacer un reporte Z o un reporte de memoria fiscal.

_ 01 = Factura fiscal en curso. Esperando por un ítem, cerrar/cancelar la factura. Solo se admitieran comandos relacionados a la factura fiscal.

_ 02 = Documento no fiscal en curso. Esperando por línea de texto, cerrar el documento. Solo se admitieran comandos relacionados a documentos no fiscales.

_ 03 = SLIP activo. Solo se admitieran comandos relacionados a documentos no fiscales o comandos para el formato de cheques.

_ 04 = Mas de un día desde el último reporte Z. Es necesario un reporte Z. Para poder realizar una venta se deberá efectuar previamente un reporte Z.

_ 05 = Primeras líneas descriptivas de una factura fiscal impresas.

_ 08 = Equipo bloqueado a la espera de impresión de cierre Z. Esto ocurre solo en caso de producirse un error durante la impresión de un cierre Z. Por ejemplo, si se acaba el papel cuando se está imprimiendo un cierre Z. Se debe hacer un *RESET* al equipo.

_ 10= Error critico. Error en BCC RAM. Es necesaria la intervención del servicio técnico.

_ 11= Error critico. Error en BCC ROM. Es necesaria la intervención del servicio técnico.

_ 12= Error critico. Error de formato de FECHA en RAM. Es necesaria la intervención del servicio técnico.

_ 13= Error critico. Error de formato de datos al realizar un Z. Es necesaria la intervención del servicio técnico.

_ 14= Error critico. Limite de memoria fiscal. Es necesaria la intervención del servicio técnico.

Para verificar que no se ha producido ningún error critico se puede comprobar que el byte más significativo sea igual a cero ("0" = 0x30). Si este byte no es cero el equipo presenta un error critico de hardware o la data contenida en la memoria de trabajo (RAM) o memoria fiscal (ROM) está corrompida. Se debe llamar a servicio técnico.

Campo 5: Contiene un código hexadecimal de dos bytes que indica cual fue el último comando procesado con éxito.

Campo 6: Fecha en la impresora fiscal. Formato <AAMMDD>

Campo 7: Hora en la impresora fiscal. Formato <HHMMSS>

Campo 8: Facturas fiscales emitidos en el periodo.

Campo 9: Documentos no fiscales emitidos en el periodo.

Campo 10: Facturas fiscales acumuladas.

Campo 11: Documentos no fiscales acumulados.

Campo 12: Es el # del último reporte Z efectuado.



RESPUESTA AL COMANDO STATUS IF (0X38) CON EL PARÁMETRO 'E'

Campo 1 Estado Impresora <0000>
Campo 2 Estado Fiscal <0000>
Campo 3 Ultimo valor de Secuencia
Campo 4 Código del status actual de la impresora
Campo 5 Código último comando ejecutado
Campo 6 Fecha en la IF <AAMMDD>
Campo 7 Hora en la IF <HHMMSS>
Campo 8 Ventas exentas

RESPUESTA AL COMANDO STATUS IF (0X38) CON EL PARÁMETRO 'A' , 'B' , 'C' , 'D' , 'R'

Campo 1 Estado Impresora <0000>
Campo 2 Estado Fiscal <0000>
Campo 3 Ultimo valor de Secuencia
Campo 4 Código del status actual de la impresora
Campo 5 Código último comando ejecutado
Campo 6 Fecha en la IF <AAMMDD>
Campo 7 Hora en la IF <HHMMSS>
Campo 8 Ventas según el parámetro = bases A, B, C, Descuentos, Devoluciones.
Campo 9 Impuestos según el parámetro = imp. A, B, C, Descuentos, Devoluciones.

RESPUESTA AL COMANDO STATUS IF (0X38) CON EL PARÁMETRO 'F' , 'J' , 'S' Y 'U'

Campo 1 Estado Impresora <0000>
Campo 2 Estado Fiscal <0000>

Con los parámetros "F" y "J" es posible bloquear a la impresora hasta que el cajero coloque otro rollo de papel. Si se envía este comando y el sensor de near end está activo la impresora se bloquea y no permite ejecutar ningún comando de impresión.

Ejemplos:

Respuesta:= pfestatus(edlinea);
Resultado := pfultimo;



FUNCION PFGAVETA:STRING;

Descripción: Abre gaveta de dinero conectada al equipo.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfgaveta;



FUNCION PFLEERELOJ: STRING;

Descripción: Realiza la lectura del reloj de la impresora. Revisar FUNCION PFultimo para obtener resultados.

Parámetros: (string) Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfleerelaj;

Respuesta:= pfUltimo;



FUNCION PFLINEANF (EDLINEA: STRING): STRING;

Descripción: Envía una línea de texto en documentos no fiscales.

Parámetros:

(string) EDLINEA: Texto de línea no fiscal a generar.
80 caracteres para el modelo PF-300 y 40 para el resto de los modelos. Si envía un texto de mayor tamaño este será truncado.

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfLineaNf(edlinea);



FUNCION PFLOGO:STRING;

Descripción: Envía la impresión del logo almacenado en la memoria de la impresora para los modelos que lo soportan y fueron configurados correctamente.

Parámetros: ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfLogo;



FUNCION PFPARCIAL:STRING;

Descripción: Genera un cierre parcial de documento fiscal tipo facture y nota de crédito. Después de ejecutado este comando toda la información de la facture es almacenada en la memoria fiscal

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfparcial;



FUNCION PFREGLON(DESCRIPCION,CANTIDAD,MONTO,IVA:STRING):STRING;

Descripción: Envía un renglón (Item o producto) de la facture.

Parámetros:

(string) DESCRIPCION: Texto descriptivo del producto. El largo máximo para el modelo PF-220 es de 20 caracteres y 40 para la PF-300.

En la versión 1.8 soporta descripciones extendidas y el API añade tantas líneas extras necesarias para poder generar la descripción completa.

(string) CANTIDAD: Cantidad a facturar en formato numérico con **punto** decimal de máximo 3 decimales. Importante que solo contenga punto como símbolo decimal y no coma.

(string) MONTO: Monto unitario sin impuesto del producto: Numérico con **punto** decimal. Importante que solo contenga punto como símbolo decimal y no coma.

(string) IVA: Valor del porcentaje de IVA a calcular en el formato: EEDD. Debe coincidir con el registrado en el equipo. Tamaño de 4 caracteres. Ej. 0000 1200 0800

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfrenglon(Descripcion,cantidad,monto,iva);

Respuesta:= pfrenglon('PRODUCTO 1','10.21','25.01','0000');



FUNCION PFREPMEMNF(DESF,HASF,MODMEM:STRING):STRING;

Descripción: Genera un reporte de memoria fiscal almacenado en el equipo.

Parámetros: (string) DESF: Fecha desde en formato DDMMAA
HASF: Fechas hasta en formato DDMMAA
MODMEM: Modo del reporte resumen o detallado

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfpemMemNF(desf,hasf,modmem);



FUNCION PFREPMEMORIANUMERO(DESN,HASN,MODMEM:STRING):STRING;

Descripción: Genera reportes de memoria fiscal basados en los números de reporte Z.

Parámetros: (string) DESN,HASN: Numero de reporte Z inicial y final

MODMEM: Modo del reporte resumen o detallado

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfRepMemoriaNumero(desn,hasn,modmem);



FUNCION PFREPX:STRING;

Descripción: Genera un reporte X en el equipo.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfrepx;



FUNCION PFREPZ:STRING;

Descripción: Genera un reporte de cierre diario o reporte Z.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfRepz;



FUNCION PFRESET:STRING;

Descripción: Reinicia el equipo. Comando no necesario pero puede ayudar en resolución de situaciones de desarrollo.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfReset;



FUNCION PFSERIAL:STRING;

Descripción: Solicita el serial del equipo. Se debe ejecutar el comando último para obtener su valor

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfSerial;



FUNCION PFSLIPOFF:STRING;

Descripción: Desactiva el modo validador SLIP en determinados modelos.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfslipoff;



FUNCION PFSLIPON:STRING;

Descripción: Activa el modo validación para los equipos que soportan el comando.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfSlipon;



FUNCION PFTFISCAL(EDLINEA:STRING):STRING;

Descripción: Envía una línea de texto fiscal al equipo. Estas líneas se pueden usar 4 al inicio del documento fiscal. 3 entre cada producto y 20 al final de la facture después de un cierre parcial.

Parámetros: (string) EDLINEA: Línea de texto fiscal a generar.

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pftfiscal(edlinea);



FUNCION PFTIPOIMP(MODELO:STRING):STRING;

Descripción: Cambia el modelo de la impresora para permitir soporte de largos de líneas para el modelo PF-300-II

Parámetros: (string) MODELO: Modelo del equipo '300' para el modo PF-300-II. Cualquier otro texto para el resto de los modelos.

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pftipoimp(modelo);



FUNCION PFTOTAL:STRING;

Descripción: Genera el final de una facture, imprime el total si este no fue generado por un comando parcial previo y corta el papel.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pftotal;



FUNCION PFTOTECONOMICO;

Descripción: Igual que total pero mientras avanza para el corte de papel genera el encabezado de la próxima facture para ahorrar papel.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pftotecnomico;



FUNCION PFULTIMO:STRING;

Descripción: Devuelve la ultima respuesta obtenida del impresor donde cada campo coincide con los establecidos en el manual de protocolo y para facilitar su manejo están separados por comas.

Parámetros: (string) 1,2,3,4

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfultimo;



FUNCION PFVALIDA675(CAMPO1,CAMPO2,CAMPO3,CAMPO4:STRING):STRING;

Descripción: Genera validación en la impresora modelo PF-675.

Parámetros: (string) CAMPO1,CAMPO2,CAMPO3,CAMPO4: Líneas de texto para generar en la validación superior

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfvalida675(campo1,campo2,campo3,campo4);



FUNCION PFVERSION:STRING;

Descripción: Solicita la versión de producción del equipo fiscal.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,

ER: Existe un error

NP: Puerto No Abierto

TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfversion;



FUNCION PFVOLTEA:STRING;

Descripción: Volea el cheque para equipos compatibles.

Parámetros: Ninguno

Resultados: (string)

OK: Transmisión/recepción fue correcta,
ER: Existe un error
NP: Puerto No Abierto
TO: Se excedió el tiempo de respuesta esperado del equipo.

Ejemplos:

Respuesta:= pfvoltea;



ANEXOS

SECUENCIA SUGERIDO PARA GENERAR UNA FACTURA

El siguiente ejemplo le guiara en la lógica basada en los comandos del api para la generación de una factura.

```
PFAbrePuerto('1');  
PFAbreFiscal('CLIENTE','RIF');  
PFRenglon('PRODUCTO 1','1','10','1200');  
PFTOTAL;
```

SECUENCIA PARA LA GENERACION DE DOCUMENTO NO FISCAL

A continuación la secuencia de comandos a ejecutar para la generación de un documento no fiscal.

```
PFAbrePuerto('1');  
PFAbreNF;  
PFLineaNF(' TEXTO ');  
PFCierraNF;
```

VERSIONES DISPONIBLES DE LA LIBRERIA

Existe una versión pnpdll.dll y otra pnpdlltest.dll y esta última le permitirá simular una impresora para probar toda la integración además de que tiene la facultad de monitorear la comunicación con la impresora conectada para propósitos de revisión de problemas.

Se entrega con el paquete una aplicación de prueba (testdllpnp.exe) que le permitirá ejecutar los comandos principales y poder probar la funcionalidad de la librería sin la necesidad de generar código alguno.