

# **SWEDA**

## **SWMFD.DLL**

## **SWMFD.SO**

Versão 1.26 / Novembro 2009

Este manual tem caráter técnico-informativo, sendo propriedade da SWEDA INFORMÁTICA LTDA. Todos os direitos reservados.

É permitida a reprodução deste documento, desde que preservada a integridade do seu conteúdo, com autorização prévia da SWEDA.

Todos os cuidados foram tomados no preparo deste manual. Entretanto, a SWEDA não assume nenhuma responsabilidade por erros, imprecisões ou omissões em seu conteúdo, assim como por quaisquer tipos de danos, perdas ou lucros cessantes decorrentes do uso da informação aqui contida.

As informações contidas neste manual estão sujeitas a alteração sem aviso prévio.

# ÍNDICE

<a href="#">Apresentação.....</a>	<a href="#">3</a>
<a href="#">Composição do Kit.....</a>	<a href="#">4</a>
<a href="#">Arquivo SWMFD.INI.....</a>	<a href="#">5</a>
<a href="#">SWMFD.DLL.....</a>	<a href="#">6</a>
<a href="#">Função SWEDA_CapturarDispositivo.....</a>	<a href="#">6</a>
<a href="#">Estrutura TInfoCallBackCD.....</a>	<a href="#">7</a>
<a href="#">Eventos notificados à rotina de callback.....</a>	<a href="#">7</a>
<a href="#">Função SWEDA_ExtrairInformacoesECF.....</a>	<a href="#">10</a>
<a href="#">Estrutura TPropriedadesDispositivo.....</a>	<a href="#">11</a>
<a href="#">Função SWEDA_ArquivoSINTEGRA.....</a>	<a href="#">12</a>
<a href="#">A Rotina de CallBack.....</a>	<a href="#">15</a>
<a href="#">Estrutura TInfoCallBackSintegra.....</a>	<a href="#">15</a>
<a href="#">Eventos notificados à rotina de callback.....</a>	<a href="#">16</a>
<a href="#">Função SWEDA_ReproduzirLeituraMemoriaFiscal.....</a>	<a href="#">19</a>
<a href="#">Função SWEDA_ReproduzirFitaDetalhe.....</a>	<a href="#">20</a>
<a href="#">Função SWEDA_ArquivoEletronicoCOTEPE.....</a>	<a href="#">22</a>
<a href="#">Função SWEDA_TabelaMercadoriasServicos.....</a>	<a href="#">24</a>
<a href="#">Função SWMFD_ObterParametrosConfiguracao.....</a>	<a href="#">26</a>
<a href="#">Função SWMFD_AlterarParametrosConfiguracao.....</a>	<a href="#">26</a>
<a href="#">Estrutura SWMFD_INI.....</a>	<a href="#">26</a>
<a href="#">SWEDA_VerificarAssinaturaDigital.....</a>	<a href="#">26</a>
<a href="#">Função SWEDA_SWMFDVersao.....</a>	<a href="#">27</a>
<a href="#">Funções Complementares.....</a>	<a href="#">27</a>
<a href="#">Função SWEDA_DetectarECF.....</a>	<a href="#">27</a>
<a href="#">Função SWEDA_ConectarECF.....</a>	<a href="#">29</a>
<a href="#">Função SWEDA_EmitirLeituraX.....</a>	<a href="#">30</a>
<a href="#">Função SWEDA_EmitirLeituraMemoriaFiscal.....</a>	<a href="#">30</a>
<a href="#">Função SWEDA_EmitirLeituraParametros.....</a>	<a href="#">31</a>
<a href="#">Função SWEDA_ImprimirFitaDetalhe.....</a>	<a href="#">31</a>
<a href="#">Códigos de Erro da SWMFD.DLL.....</a>	<a href="#">31</a>

## APRESENTAÇÃO

A biblioteca SWMFD.DLL é composta por um conjunto de funções para captura e processamento das informações armazenadas nos dispositivos de memória fiscal e de fita-detalhe das impressoras fiscais da Linha Térmica Sweda (Modelos: IF ST100, IF ST120, IF ST200 e IF ST1000), e possibilitam:

- Reproduzir a fita-detalhe
- Reproduzir a Leitura da Memória Fiscal.
- Gerar o arquivo SINTEGRA, conforme as especificações do convênio COTEPE/ICMS nº 57/95 e 20/04 (Registros 10, 11, 60M, 60A, 60D, 60I, 60R, 75 e 90).
- Gerar o arquivo eletrônico, segundo as especificações técnicas do ato COTEPE/ICMS nº 17/04 de 29 de Março de 2004

## COMPOSIÇÃO DO KIT

Arquivo	Descrição	Linguagem
SWMFD.DLL	Biblioteca de Ligação Dinâmica	-
SWMFD.LIB	Arquivo .LIB para importação de funções da biblioteca	C/C++
SWMFD.H	Declaração de protótipos e estruturas de dados	C
TSWMFD.H	Declaração da classe TSWMFD	C++
TSWMFD.CPP	Código fonte do objeto TSWMFD	C++
UnitTSWMFD.PAS	Classe TSWMFD	Pascal
UnitSWMFD.PAS	Protótipos das funções da biblioteca SWMFD.DLL	Pascal

O objeto TSWMFD, implementado para as linguagens C++ e Pascal (Delphi), contém as mesmas funções e protótipos da SWMFD.DLL, incluindo funções para o carregamento e o descarregamento dinâmico da biblioteca. Observe sua utilização nos programas de exemplo que acompanham o kit.

Pasta	Descrição
Source\C	Arquivos-fonte para Linguagem C
Source\C++	Arquivos-fonte para Linguagem C++
Source\Delphi	Arquivos-fonte para Delphi/Pascal
Exemplos\CBuilder	Programa de exemplo para CBuilder v. 3.0
Exemplos\Delphi	Programa de exemplo para Delphi v. 6.0
Exemplos\BIN	Informações do dispositivo de MFD
Validador SINTEGRA 2006 (5.2.0)	Contém o programa de instalação do validador de arquivos SINTEGRA, desenvolvido conjuntamente pelas secretarias de fazenda dos estados de Sergipe, Minas Gerais e Rio Grande do Sul.

# ARQUIVO SWMFD.INI

A SWMFD.DLL lê os parâmetros de configuração inicial do arquivo SWMFD.INI.

```
[Comunicação]                               Opções de configuração da porta comunicação
  Porta: 0                                   Porta serial (0 = Detectar; 1 = COM1, 2 = COM2...)
  Velocidade: 0                             Baud-rate (0 = Detectar, 9600, 19200, 38400, 57600, 115200)

[ECF]                                         Dados sobre o ECF local
  FAB: SW010100000000004262                 Número serial de fabricação
  MFD: MFDSW000000001612                   Número serial do dispositivo de memória de fita-detache (MFD)

[Sweda]                                      Diretório de gravação dos arquivos binários
  Diretório: C:\SWEDA

[SINTEGRA]                                   Dados de identificação do contribuinte
  CNPJ: "53.485.215/0001-06"
  IE: "111.072.115.110"
  IM: ""
  RAZÃO SOCIAL: "Sweda Informática Ltda"
  LOGRADOURO: "Rua Dona Brígida"
  NUMERO: "713"
  COMPLEMENTO: ""
  BAIRRO: "Vila Mariana"
  CIDADE: "São Paulo"
  UF: "SP"
  CEP: "0920-410"
  TELEFONE: "2106-5644"
  FAX: "5574-5644"
  CONTADO: "SAC"

[ASSINATURA DIGITAL]                       Opções de assinatura digital (registro EAD) dos arquivos gerados
                                              pela biblioteca: SIM, NÃO, OPCIONAL*
  FITA-DETALHE:                             OPCIONAL
  LEIUTRA MEMÓRIA FISCAL:                   OPCIONAL
  COTEPE:                                   OPCIONAL
  MEMÓRIA FISCAL:                           OPCIONAL
  MEMÓRIA DE FITA-DETALHE:                 OPCIONAL
  SOFTWARE BÁSICO:                         OPCIONAL
```

*Arquivo SWMFD.INI*

Quando distribuído, o arquivo SWMFD.INI deve ser copiado para a pasta do sistema operacional:  
WINDOWS\SYSTEM ou WINDOWS\SYSTEM32.

Se o arquivo não for encontrado pela DLL será criado um arquivo, contendo a configuração padrão.

# SWMFD.DLL

## Função SWEDA\_CapturarDispositivo

**Descrição:** Função para transferência das informações armazenadas na memória fiscal ou na memória fita-detalle do ECF, e do Software Básico.

**Parâmetro(s):** *Dispositivo*

Seleção do dispositivo:

Constante	Valor	Descrição
OpCapturarMFD	1	Memória de Fita-Detalhe
opCapturarMF	2	Memória Fiscal
opCapturarSoftwareBasico	4	Software Básico
opAssinarArquivo	512	Assina digitalmente o arquivo

*Opções*

Opções de performance da função, opcional:

Constante	Valor	Descrição
opMostrarJanela*	0	Exibir uma janela de acompanhamento
opCancelamento	1024	Permite ao usuário interromper a captura.
opOcultarJanela	2048	Ocultar a janela de acompanhamento
opSegundoPlano	4096	Processar em segundo plano
opAutoDetectarECF	16384	Detecta, se necessário, as configuração da porta de comunicação com o ECF

\*Padrão

*Destino*

Caso 1: Endereço de string de caracteres, terminada por caracter nulo, contendo o nome e diretório do arquivo de saída.

Caso 2: Endereço para buffer da aplicação para retorno do nome padrão do arquivo, gerado pela função a partir do número de fabricação do ECF e do número de série do dispositivo transferido. Neste caso, o próximo argumento (*Extensão*) é obrigatório.

Caso 3: Endereço nulo. A função gera o arquivo de nome padrão, sem retorna-lo à aplicação.

Argumento opcional, valor padrão: endereço nulo.

*Extensão*

Capacidade do buffer informado no parâmetro *Destino* (Caso 2), em bytes. Parâmetro opcional, valor padrão: 0

*Timeout*

Tempo máximo de espera da função pela resposta do ECF à solicitação de transferência de informações, definido em milisegundos (ms); Parâmetro opcional, valor padrão: 15000 ms (ou 15 segundos). Se igual 0 o tempo de espera é infinito.

*CallBack* Rotina de callback (tipo TCallBackCD)  
 Parâmetro opcional , valor padrão: endereço nulo.  
 A rotina de callback é de implementação é facultativa. Quando implementada, a rotina será chamada durante a execução da função *SWEDA\_CapturarDispositivo* e receberá por parâmetro, em uma estrutura do tipo TInfoCallBackCD, informações sobre o processo.

*Instancia* Dado(s) da aplicação, repassado(s) pela função *SWEDA\_CapturarDispositivo* à rotina de callback.  
 Parâmetro opcional, valor padrão: endereço nulo.

**Retorno:**

- 2 Executando em segundo plano!
- 1 Processo interrompido pela aplicação!
- 0 Ok
- 1 Falha na abertura do arquivo de saída!
- 2 Falha de gravação!
- 3 Falha de comunicação com o ECF (Timeout)
- 4 Falha na criação da instancia!
- 5 ECF off-line ou não detectado!

#### Estrutura TInfoCallBackCD

Informações repassadas pela função *SWEDA\_CapturarDispositivo* à rotina de callback.

Campo	Descrição	Tipo	Extensão
Arquivo	Nome do arquivo destino	Caracter	260
NS	Número de série do dispositivo	Caracter	22
FAB	N/S de fabricação do ECF	Caracter	22
Modelo	Modelo do equipamento	Caracter	22
Evento	Código do evento	Inteiro	2
Dispositivo	Dispositivo em transferência: 1 = Memória de Fita-Detalhe 2 = Memória Fiscal 3 = Software Básico	Inteiro	2
Contador	Valores: máximo, mínimo e corrente do contador.	TContador	-
Retorno	Valor retornado à aplicação na conclusão da função	Inteiro	2
TempoEstimado	Tempo, em segundos, estimado para conclusão da transferência.	Inteiro	4

#### Eventos notificados à rotina de callback

	Evento*	Descrição
<i>Inicialização...</i>	100	Início...
	200	Detectando o ECF...
	110	Obtendo informações sobre os dispositivos instalados...
<i>Capturara de informações do ECF...</i> (opção: <i>opTransferirMFD</i> )	130	Iniciada a transferência de dados do dispositivo...
	140	Incremento do contador (bytes transferidos)

	160	Ocioso, aguardando resposta do ECF...
<i>Encerramento da função</i>	197	Erro! (código ref. ao erro na variável <i>Retorno</i> )
	198	Processo interrompido pela aplicação!
	199	Transferência concluída com sucesso!

\*valor indicado no campo Evento da estrutura TInfoCallBackCD.



## Exemplo:

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo.c

// Rotina de CallBack

void __stdcall CallBackCD(TInfoCallBackCD *Info, void *Instancia)
{
    switch (Info->Evento)
    {
        // Evento 100 - Iniciando a função
        // Evento 110 - Obtendo informações sobre o ECF...
        case 110: printf ("Comunicando-se com o ECF...\r\n");
                 break;

        // Evento 130 - Iniciando a transferência de um dispositivo...
        case 130: if (Info.Dispositivo == 1)
                 printf ("Capturando informações da memória de fita-detache...\r\n");
                 if (Info.Dispositivo == 2)
                 printf ("Capturando informações da memória fiscal...\r\n");
                 if (Info.Dispositivo == 3)
                 printf ("Capturando o software básico...\r\n");
                 break;

        // Evento 140 - Incremento do contador (bytes transferidos)
        case 140: printf "\r " %ld/%ld bytes transferidos", Info->Contador.Valor, Info->Contador.Maximo);
                 break;

        // Evento 160 - Aguardando retorno do ECF

        // Evento 197 - Erro (Código de erro na variável Retorno)!
        case 197: printf ("\r\nErro: %d\r\n\n", Info->Retorno);
                 break;

        // Evento 198 - Processo interrompido pela aplicação!
        case 198: break;

        // Evento 199 - Processo concluído!
        case 199: printf ("\r\nConcluído!\r\n\n");
                 break;
    }
}

void main ()
{
    char szNomePadrao[512];

    SWEDA_CapturarDispositivo (
        1,
        opOcultarJanela | opSegundoPlano,
        NULL,
        0,
        15000,
        &CallBackCD,
        NULL);

    // Obtém o nome padrão do arquivo de saída, gerado pela função a partir dos números do série do
    // equipamento

    if (SWEDA_CapturarDispositivo (
        1,
        opOcultarJanela | opSegundoPlano,
        szNomePadrao,
        sizeof (szNomePadrao),
        15000,
        &CallBackCD,
        NULL) == 0)
        printf ("\r\n\nO nome padrão é: \"%s\"\r\n\n", szNomePadrao);
}
```

## Função SWEDA\_ExtrairInformacoesECF

**Descrição:** A função *SWEDA\_ExtrairInformacoesECF* extrai de um arquivo proveniente de um dos dispositivos do ECF, informações sobre o equipamento que o originou.

**Parâmetros(s)**

<i>Nome do arquivo</i>	Nome completo do arquivo (string)
<i>buffer</i>	Endereço de uma estrutura de dados do tipo <i>TPropriedadesDispositivo</i>
<i>extensão</i>	Extensão do buffer (inteiro)

### Exemplo:

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo C

void main ()
{
    TPropriedadesDispositivo Propriedades;

    // Obtém as informações do arquivo C:\SW0105000000000336.BIN
    if (SWEDA_ExtrairInformacoesECF ("C:\SW0105000000000336.BIN"
        , &Propriedades, sizeof (Propriedades)) > 0)
    {
        switch (Propriedades.Arquivo.Tipo)
        {
            case 1: // 0 arquivo contém informações registradas na memória de fita-detelhe
            case 2: // 0 arquivo contém informações registradas na memória fiscal
                printf ("Tipo: %s\n"
                    "Modelo: %s\n"
                    "FAB: %s\n"
                    "Dispositivo: %s\n"
                    , Propriedades.ECF.Tipo
                    , Propriedades.ECF.Modelo
                    , Propriedades.ECF.FAB
                    , Propriedades.Arquivo.Dispositivo);

                // Informações da memória de fita-detelhe...
                if (Propriedades.Arquivo.Tipo == 1)
                    printf ("N/S: %s\n"
                        "Capacidade: %s\n"
                        "Consumo: %s\n"
                        , Propriedades.MFD.NS
                        , Propriedades.MFD.Capacidade
                        , Propriedades.MFD.Consumo);

                // Informações sobre o(s) usuário(s) do ECF
                for (i = 0; i < Propriedades.Usuarios; i++)
                {
                    printf ("\n"
                        "Usuário: %d\n"
                        "  C.N.P.J.: %s\n ", i+1
                        "  I.E.: %s\n"
                        "  I.M.: %s\n"
                        "  MFD: %s\n"
                        , Propriedades.Usuario[i].CNPJ
                        , Propriedades.Usuario[i].IE
                        , Propriedades.Usuario[i].IM
                        , Propriedades.Usuario[i].MFD);
                }
                break;

            case 3: // software básico
                printf ("Software Básico\n" +
                    "  Fabricante: %s\n"
                    "  Tipo: %s\n "
                    "  Modelo: %s\n "
                    "  Versão: %s\n "
                    "  Checksum: %s\n "
                    , Propriedades.ECF.Marca
                    , Propriedades.ECF.Tipo
                    , Propriedades.ECF.Modelo
                    , Propriedades.SB.Versao
                    , &Propriedades.SB.Checksum[4]);
                break;
        }
    }
}
```

```
}
}
```

**Retorno:**

- 2 O arquivo não existe
- 1 Erro na abertura do arquivo
- 0 Arquivo desconhecido
- 1 Memória de fita-detalle
- 2 Memória fiscal
- 3 Software Básico

#### Estrutura TPropriedadesDispositivo

Campo	Descrição	Tipo	Extensão
Arquivo	Informações sobre o arquivo de origem	TArquivo	-
Tipo	Tipo de arquivo: 0 = Arquivo desconhecido 1 = Memória de fita-detalle 2 = Memória fiscal 3 = Software Básico	Inteiro	2
Dispositivo	Descrição do dispositivo de origem.	Caracter	40
Data	Data de atualização do arquivo	Caracter	11
Horário	Horário de atualização do arquivo	Caracter	10
Nome	Nome do arquivo	Caracter	260
Extensao	Tamanho total do arquivo	Inteiro	4
ECF	Informações sobre o equipamento	TECF	-
Marca	Marca: 'Sweda'	Caracter	21
Modelo	Modelo	Caracter	21
Tipo	Tipo: 'ECF-IF'	Caracter	21
ECF	Número sequencial do ECF no estabelecimento	Caracter	10
Loja	Indicador de loja	Caracter	20
FAB	Número serial de fabricação	Caracter	22
Berco	Berço: ' ' ou 'A'	Caracter	1
Usuario[]	Informações sobre usuários do ECF	TUsuario	[12]
CNPJ	C.N.P.J.	Caracter	21
IE	Inscrição Estadual	Caracter	21
IM	Inscrição Municipal	Caracter	21
MFD	Número serial do dispositivo de MFD	Caracter	22
Moeda	Símbolo da moeda	Caracter	7
GT	Valor acumulado no totalizador geral	Caracter	25
Usuarios	Número de usuários	Inteiro	2
SB	Informações sobre o software básico	SB	2
Versão	Versão do software básico instalado	Caracter	12
Checksum	Checksum do arquivo	Caracter	9
MFD	Informações sobre o dispositivo de MFD instalado	TMFD	-
NS	Número de série de fabricação	Caracter	22
Berço	Berço	Caracter	1
Capacidade	Capacidade	Inteiro	4
Consumo	Consumo	Inteiro	4
DataInstalacao	Data de inicialização no ECF	Caracter	11
Contadores	Valores acumulados nos contadores fiscais	TContadores	-
COO	Valo acumulado no COO (Contador de Ordem de Operação)	Caracter	7
CRZ	Valo acumulado no CRZ (Contador de Reduções Z)	Caracter	7

CRO	Valo acumulado no CRO (Contador de Reinício de Operação)	Caracter	7
-----	---	----------	---

## Função SWEDA\_ArquivoSINTEGRA

**Descrição:** Gera o arquivo SINTEGRA segundo as especificações do convênio COTEPE/ICMS 57/95 e 20/04, a partir das informações armazenadas na memória de fita-detalle do ECF, transferidas para arquivo magnético.

Registros gerados:

10	Mestre do Estabelecimento
11	Dados complementares do informante
60M	Mestre
60A	Analítico
60D	Resumo Diário
60I	Item
60R	Resumo Mensal
75	Código de Produto ou Serviço
90	Totalização do arquivo

A função permite consolidar informações provenientes de diversos ECFs em um único arquivo.

**Parâmetros(s):**

<i>Destino</i>	Nome e diretório do arquivo destino (string).
<i>Inicio</i>	Data inicial (string). Formato: [dd/]mm/aaaa, [dd/]mm/aa ou ddmmaaaa
<i>Fim</i>	Data final (string), opcional. Formato: [dd/]mm/aaaa, [dd/]mm/aa ou ddmmaaaa

*Estabelecimento* Conjunto de informações sobre o contribuinte.  
Endereço de uma estrutura de dados do tipo *TdadosEstabelecimento*:

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Extensão</b>
CNPJ	CNPJ	Character	21
IE	Inscrição Estadual	Character	21
IM	Inscrição Municipal	Character	21
RazaoSocial	Nome comercial do contribuinte	Character	51
Logradouro	Logradouro	Character	51
Numero	Numero	Character	11
Complemento	Complemento	Character	31
Bairro	Bairro	Character	41
Município	Município	Character	41
UF	Unidade da Federação	Character	3
CEP	Código de Endereçamento Postal	Character	16
Telefone	Número de telefone para contato	Character	21
FAX	Fax	Character	21
Contato	Pessoa responsável para contato	Character	41

Os campos CNPJ, IE e IM são opcionais, quando informados a função os utiliza para selecionar os arquivos daquele usuário.

*Registros* Seleção do conteúdo do arquivo por registros:

<b>Constante</b>	<b>Valor</b>	<b>Registro(s) Selecionado(s)</b>
opSintegraTipo10	1	Tipo 10 - Mestre do Estabelecimento
opSintegraTipo11	2	Tipo 11 - Dados complementares do informante
opSintegraTipo60M	4	Tipo 60M – Mestre
opSintegraTipo60A	8	Tipo 60A – Analítico
opSintegraTipo60D	16	Tipo 60D - Resumo Diário
opSintegraTipo60I	32	Tipo 60I – Item
opSintegraTipo60R	64	Tipo 60R - Resumo Mensal
opSintegraTipo60x	125	Tipos: 60M, 60A, 60D, 60I, 60R
opSintegraTipo75	128	Tipo 75 - Código de produto ou serviço
opSintegraTipo90	256	Tipo 90 - Totalização do arquivo

*Opções* Opções de performance da função (inteiro), opcional:

Constante	Valor	Descrição
opArquivosPasta	1	Processar todos os arquivos da pasta <i>Origem</i>
opIncluirSubpastas	3	Processar todos os arquivos da pasta e subpasta(s) <i>Origem</i>
opListaEmArquivo	4	Processar a relação de arquivos listados no arquivo <i>Origem</i>
opTransferirMFD	8	Captura as informações da MFD, atualizando o arquivo de origem antes de processá-lo.
opWindowsCodePage*	0	Página de Código: Windows Code Page
opCodePage860	128	Página de Código: CP860 (Português)
opCodePage850	256	Página de Código: CP850 (Multilingual)
opMostrarJanela*	0	Exibir uma janela de acompanhamento
opCancelamento	1024	Possibilita ao usuário interromper o processo.
opOcultarJanela	2048	Ocultar a janela de acompanhamento
opSegundoPlano	4096	Processar em segundo plano
opAutoDetectarECF	16384	Faz auto-deteção do ECF, se necessário

\*Padrão

*Origem* Arquivo, pasta ou lista de arquivos a processar:  
Parâmetro opcional; Valor padrão: nulo.

*Caso 1.* Nome do arquivo binário a ser processado.

*Caso 2.* Nome de arquivo em formato texto, que contém a relação dos nomes de arquivos a processar.

*Caso 3.* Pasta a ser processada.

Quando omitido este parâmetro, a função assume o diretório de gravação configurado na chave 'Diretório' do arquivo de inicialização (SWMFD.INI). E, se não selecionada a opção opArquivosPasta ou opIncluirSubpastas, o arquivo correspondente ao ECF local.

*Log* Nome do arquivo de log, opcional

*CallBack* Rotina de callback. Através dela função comunica à aplicação os eventos internos ocorridos durante a execução.  
Parâmetro opcional; Tipo: TCallBackSintegra

*Instância* Endereço para estrutura de dados da aplicação repassada à rotina de callback, opcional

*Mascara* Seleciona os registro que a aplicação deseja analisar antes da gravação (evento nº 82 da rotina de callback), opcional. Usar as mesmas constantes do parâmetro *Registros*, opcional.

**Importante:** Observe as seguintes adequações do aplicativo de frente de caixa, quanto ao registro de itens:

- Não agregar à descrição do produto qualquer outra informação, tais como: quantidade comercializada, unidade de medida, valor unitário, acréscimos ou

descontos.

- Informar o código real do produto e não o código de barras lidos das etiquetas de preço impressas em balanças eletrônica;
- Informar precisamente o valor unitário do produto e a quantidade comercializada.

As adequações possibilitam a consolidação das vendas por produto, preservam os recursos da memória de fita-detelhe (ampliando a vida-útil do equipamento) e não implicam em aumento do consumo de papel utilizado na impressão do registro.

**Retorno:**

- 2 Executando em segundo plano!
- 1 Processo interrompido pela aplicação!
- 0 Ok
- 1 Falha de abertura do arquivo log!
- 2 Data inválida!
- 3 Faixa inválida!
- 4 O número de arquivos excede a capacidade da fila! (100 arquivos)
- 5 Nenhum arquivo encontrado! (opção: opArquivosPasta)
- 6 Nenhum arquivo selecionado! (opção: opListaEmArquivo)
- 7 Arquivo não encontrado! (*Origem*)
- 8 Memória insuficiente!
- 9 Falha em arquivo temporário!
- 10 Não foi possível abrir o arquivo de saída!
- 11 Arquivo inválido!
- 12 Falha na criação da instância!
- 13 O mês selecionado não está encerrado!

### A Rotina de Callback

A rotina de callback é chamada pela função SWEDA\_ArquivoSINTEGRA durante sua execução e recebe por parâmetro informações relativas aos processos internos da função (uma estrutura do tipo TInfoCallbackSintegra). Sua implementação é facultativa.

### Estrutura TInfoCallbackSintegra

Informações repassadas pela função de geração do SINTEGRA à rotina de callback.

Campo	Descrição	Tipo	Extensão
Estabelecimento	Informações sobre o estabelecimento	TDadosEstabelecimento	-
Fila[100]	Lista de arquivos na fila de processamento	TArquivoOrigem	-
Arquivos	Total de arquivos na fila	Inteiro	2
Arquivo	Informações sobre o arquivo em processamento	TArquivoOrigem	-
Sequencial	Número seqüencial do arquivo na fila	Inteiro	2
Contador	Valores: máximo, mínimo e corrente do contador.	TContador	-
Destino	Nome do arquivo destino	Caracter	260
Evento	Código do evento	Inteiro	2
Retorno	Valor retornado à aplicação	Inteiro	2
Tipo	Tipo do registro (Evento 82)	Inteiro	2
Subtipo	Subtipo do registro (Evento 82)	Caracter	1
Registro	Registro (Evento 82)	Ponteiro	4

### Estrutura TContador

Valores: máximo, mínimo e corrente do contador:

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Extensão</b>
Mínimo	Valor inicial do contador	Inteiro	4
Máximo	Valor final do contador	Inteiro	4
Valor	Valor corrente do contador	Inteiro	4

### Estrutura TArquivoOrigem

Dados sobre o arquivo de origem:

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Extensão</b>
Nome	Nome do arquivo de origem	Caracter	260
Data	Data de atualização do arquivo	Caracter	11
Horario	Horário de atualização do arquivo	Caracter	10
Dispositivo	Dispositivo de origem	Ponteiro / String	4
Extensao	Tamanho do arquivo em bytes	Inteiro	4
Loja	Indicador de loja	Caracter	5
ECF	Número sequencial do ECF de origem	Caracter	4
FAB	N/S de fabricação do ECF de origem	Caracter	22
+Tipo	Tipo de arquivo: 1 = Memória de fita-detalle 2 = Memória fiscal	Inteiro	2

### Eventos notificados à rotina de callback

	<b>Evento*</b>	<b>Descrição</b>
<i>Inicialização da função</i>	10	Verificando a consistência dos parâmetros...
<i>Capturara de informações do ECF... (opção: opTransferirMFD)</i>	100	Início do processo de transferência do dispositivo.
	200	Detectando o ECF...
	110	Obtendo informações sobre o dispositivos instalados...
	130	Transferindo informações da memória de fita-detalle... O contador é reiniciado, referindo-se á quantidade de bytes a ser transferido.
	140	Incremento do contador (bytes transferidos)
	160	Ocioso, aguardando resposta do ECF...
<i>Iniciando o processamento dos arquivos...</i>	20	Arquivos inseridos na fila de processamento!
<i>Processando os arquivos de origem...</i>	40	Iniciando o processamento de um arquivo de memória de fita-detalle... O contador é reiniciado, referindo-se a documentos processados (COO).
	41	Incremento do contador (documentos processados, evento 40)
<i>Consolidação, classificação e gravação de registros...</i>	80	Consolidando as informações do arquivo. O contador é reiniciado, referindo-se a registros gravados.
	81	Incremento do contador (registros gravados)
	82	Gravando registro



Encerramento da função	97	Erro! (código ref. ao erro na variável <i>Retorno</i> )
	98	Processo interrompido pela aplicação!
	99	Processamento concluído!

\*valor indicado no campo Evento da estrutura TinfoCallBackSintegra.

### Exemplo:

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo.c

// Rotina de CallBack

void __stdcall CallBackSintegra(TInfoCallBackSintegra *Info, void *Instancia)
{
    int i;

    switch (Info->Evento)
    {
        // Evento 10 - Iniciando o processamento...
        case 10: break;

        // Evento 20 - Parâmetros e arquivos de origem Ok!
        case 20: printf ("CNPJ: %s\r\n"
                        "IE: %s\r\n"
                        "IM: %s\r\n\n"
                        , Info->CNPJ, Info->IE, Info->IM);

                for (i = 0; i < Info->Arquivos; i++)
                {
                    printf ("Loja: %-7.7s ECF: %-5.5s FAB: %-22.22s\r\n"
                            "Arquivo: %s (%s %s Dispositivo: %s)\r\n\n"
                            , Info->Fila[i].Loja
                            , Info->Fila[i].ECF
                            , Info->Fila[i].FAB
                            , ExtractFileName (Info->Fila[i].Nome)
                            , Info->Fila[i].Data
                            , Info->Fila[i].Horario
                            , Info->Fila[i].Dispositivo);
                }
                break;

        // Evento 100 Iniciando a captura das informações registradas na memória de fita-detache
        // Evento 200 Detectando o ECF
        // Evento 110 Obtendo informações sobre os dispositivos instalados no ECF...
        case 110: printf ("Comunicando-se com o ECF...\r\n"
                        break;

        // Evento 130 Transferindo informações do dispositivo...
        case 130: printf ("Capturando informações da memória de fita-detache...\r\n"

        // Evento 140 - Incremento do contador (bytes transferidos)
        case 140: printf ("\r " " %ld/%ld bytes transferidos", Info->Contador.Valor, Info->Contador.Maximo);
                break;

        // Evento 40 - Iniciando o processando de informações da memória de fita-detache em arquivo...
        case 40: printf (
                    "Processando informações do arquivo %s\r\n\r\n"
                    " LOJA: %s\n"
                    " ECF: %s\n"
                    " FAB: %s\n"
                    , Info->Arquivo.Nome
                    , Info->Loja
                    , Info->ECF
                    , Info->FAB);
                break;

        // Evento 41 - Processando informações da memória de fita-detache
        // Incremento do contador (C00)
        case 41: printf ("\r " "C00: %ld/%ld", , Info->Contador.Valor, Info->Contador.Maximo);
                break;

        // Evento 80 - Iniciando a geração o arquivo de saída...
```

```

        case 80:
            printf (
                "Consolidando os registros...\n\n" +
                "Arquivo destino: " + ExtractFileName (Info->Destino);
            Progresso->lbInfo->Update();
            break;

// Evento 81 - Incremento do contador (registros gravados)
        case 81: printf "\r" "Registros: %ld/%ld", , Info->Contador.Valor, Info->Contador.Maximo);
            break;

// Evento 82 - Gravação de registros
        case 82: printf ("%s", Info->Registro);
            break;

// Evento 97 - Erro (Código de erro na variável Retorno)!
        case 97: printf ("\r\nErro: %d\r\n\n", Info->Retorno);
            break;

// Evento 98 - Processo interrompido pela aplicação!
        case 98: break;

// Evento 99 - Processo concluído!
        case 99: printf ("\r\nConcluído!\r\n\n");
            break;
    }
}

void main ()
{
// Dados do estabelecimento informante do SINTEGRA

TDadosEstabelecimento Estabelecimento =
{
    "53.485.21/0001-06", // CNPJ (opcional)
    "111.072.115.110", // Inscrição Estadual (opcional)
    "111.072.115.110", // Inscrição Municipal (opcional)
    "Sweda Informática Ltda", // Nome comercial (Razão Social) do contribuinte
    "Rua Dona Brigida", // Logradouro
    "713", // Número
    "", // Complemento
    "Vila Mariana", // Bairro
    "São Paulo", // Município onde está domiciliado o estabelecimento informante
    "SP", // Unidade da Federação referente ao Município
    "04111-081", // Código de Endereçamento Postal
    "11 5574-5644", // Número de telefone para contato
    "5572-4799", // Fax
    "SAC", // Pessoa responsável para contato
};

SWEDA_ArquivoSINTEGRA (
    "12/2005", "",
    "C:\\SINTEGRA.TXT",
    &Estabelecimento,
    opSintegraTipo10 | opSintegraTipo11 | opSintegraTipo60x | opSintegraTipo75 | opSintegraTipo90,
    opTransferirMFD | opAutoDetectarECF,
    "",
    &CallBackSintegra,
    "C:\\LOG.TXT");
}

```

## Função SWEDA\_ReproduzirLeituraMemoriaFiscal

**Descrição:** Reproduz a Leitura da Memória Fiscal a partir das informações da memória fiscal do ECF, transferidas para um arquivo magnético.  
Se informados o buffer de saída e sua extensão (parâmetros 7 e 8) a leitura é gerada no buffer.

**Parâmetros(s)**

*Origem* Nome e diretório do arquivo de origem (string).

*Destino* Nome e diretório do arquivo destino (string) , opcional.

*Inicio* Redução inicial (string)  
Seleção por CRZ, dia ou mês, exemplos:  
CRZ: 0039 '0039' ou '39'  
Dezembro de 2005 '12/2005', '12/05'  
Dia 12/10/2006 '12/10/06', '12/10/2006' ou '12102006'

*Fim* Redução final (string), opcional  
Seleção por CRZ, dia ou mês, exemplos:  
CRZ: 0040 '0040', '40'  
Janeiro de 2006 '01/2006', '01/06'  
Dia 12/10/2006 '12/10/06', '12/10/2006' ou '12102006'

*Tipo* Tipo da leitura: Completa (1) ou simplificada (0), (booleano)  
A leitura simplificada imprime apenas as totalizações mensais dentro da faixa selecionada. A leitura completa inclui o detalhamento dos valores registrados em cada redução.

*Opções* Opções de performance da função (inteiro):

Constante	Valor	Descrição
opAnexar	16	Anexar a captura ao final do arquivo
opOcultarJanela	2048	Ocultar a janela de acompanhamento
opWindowsCodePage*	0	Página de Código: Windows Code Page
opCodePage860	128	Página de Código: CP860 (Português)
opCodePage850	256	Página de Código: CP850 (Multilingual)
opAssinarArquivo	512	Assina digitalmente o arquivo

\*Padrão

*Buffer* Buffer de saída (ponteiro), opcional.

*Extensão* Extensão do buffer (inteiro) , opcional

### Exemplo:

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo.c

void main ()
{ char LMF[30000];

// 1. Leitura simplificada, direcionada para arquivo com seleção da faixa
// de reduções por CRZ (1 a 12)

    SWEDA_ReproduzirLeituraMemoriaFiscal ("C:\\SW0105000000000336.BIN"
        , "C:\\TESTE.TXT", "1", "0012", 0);
```

```

// 2. Leitura completa, direcionada para o buffer da aplicação com seleção da faixa de
// reduções por período.

    if (SWEDA_ReproduzirLeituraMemoriaFiscal ("C:\\SW0105000000000336.BIN"
        , "", "01/01/2005", "31/01/2005", 1, 0, LMF, sizeof (LMF) == 0)
        printf (LMF);

// 3. Leitura completa, direcionada para o buffer da aplicação com seleção da faixa de
// reduções por período mensal (02/05), omitido o parâmetro de seleção final.

    if (SWEDA_ReproduzirLeituraMemoriaFiscal ("C:\\SW0105000000000336.BIN"
        , "", "02/05", "", 1, 0, LMF, sizeof (LMF) == 0)
        printf (LMF);
}

```

**Retorno:**

- 0 Ok
- 2 Data inválida!
- 3 Faixa inválida!
- 7 Falha na abertura do arquivo de origem!
- 10 Falha na abertura do arquivo de saída!
- 14 Buffer insuficiente

## Função SWEDA\_ReproduzirFitaDetalhe

**Descrição:** Reproduz a fita-detalle a partir das informações da memória de fita-detalle do ECF transferidas para um arquivo magnético.  
Se informados o buffer de saída e sua extensão (parâmetros 6 e 7) o texto é armazenada no buffer.

**Parâmetros(s):**

<i>Origem</i>	Nome e diretório do arquivo de origem (string).
<i>Destino</i>	Nome e diretório do arquivo destino (string) , opcional.
<i>Inicio</i>	Documento inicial (string) Seleção por COO, dia ou mês, exemplos: COO: 000039        '000039' ou '39' Dezembro de 2005    '12/2005', '12/05' Dia 12/10/2006       '12/10/06', '12/10/2006' ou '12102006'
<i>Fim</i>	Documento final (string), opcional Seleção por COO, dia ou mês, exemplos: COO: 000040        '000040' ou '40' Janeiro de 2006       '01/2006', '01/06' Dia 12/10/2006       '12/10/06', '12/10/2006' ou '12102006'
<i>Opções</i>	Opções de performance da função (inteiro):

Constante	Valor	Descrição
opAnexar	16	Anexar a saída ao final do arquivo
opOcultarJanela	2048	Ocultar a janela de acompanhamento
opWindowsCodePage*	0	Página de Código: Windows Code Page
opCodePage860	128	Página de Código: CP860 (Português)
opCodePage850	256	Página de Código: CP850 (Multilingual)
opAssinarArquivo	512	Assina digitalmente o arquivo

\*Padrão

<i>Buffer</i>	Buffer de saída (ponteiro), opcional.
<i>Extensão</i>	Extensão do buffer (inteiro) , opcional

**Exemplo:**

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo.c

void main ()
{ char FitaDetalhe[30000];

// 1. Saída direcionada para arquivo com seleção da faixa de documentos por C00 (000001 a 000012)
SWEDA_ReproduzirFitaDetalhe ("C:\\SW0105000000000336.BIN"
, "C:\\TESTE.TXT", "1", "000012");

// 2. Saída direcionada para o buffer da aplicação com seleção da faixa de documentos
// por período de datas
if (SWEDA_ReproduzirFitaDetalhe ("C:\\SW0105000000000336.BIN"
, "", "01/01/2005", "31/01/2005", 0, FitaDetalhe, sizeof (FitaDetalhe)) == 0)
printf (FitaDetalhe);

// 3. Seleção por período mensal (fevereiro e março de 2005)
SWEDA_ReproduzirFitaDetalhe ("C:\\SW0105000000000336.BIN", " C:\\TESTE.TXT "
, "02/05", "03/2005");

// 4. Seleção de um único documento (C00: 000129)
SWEDA_ReproduzirFitaDetalhe ("C:\\SW0105000000000336.BIN", " C:\\TESTE.TXT ", "129", "");
}
```

**Retorno:**

0	Ok
2	Data inválida!
3	Faixa inválida!
7	Falha na abertura do arquivo de origem!
10	Falha na abertura do arquivo de saída!
14	Buffer insuficiente

## Função SWEDA\_ArquivoEletronicoCOTEPE

**Descrição:** Gera o arquivo eletrônico, segundo as especificações técnicas do ato COTEPE/ICMS nº 17/04 de 29 de Março de 2004, a partir das informações da memória de fita-detalle ou memória fiscal do ECF, transferidas para um arquivo magnético.

**Parâmetros:**

<i>Origem 1</i>	Nome e diretório do arquivo em formato binário, obtido a partir da transferência das informações armazenadas na MF ou MFD do ECF.
<i>Origem 2</i>	Nome e diretório do arquivo em formato binário, obtido a partir da transferência das informações armazenadas na MF ou MFD do ECF. O segundo arquivo de origem é necessário somente quando selecionada a opção TDM.
<i>Destino</i>	Nome e diretório do arquivo destino (string)

**Parâmetros:**

**(cont.)**

*Início* Documento inicial (string)  
 Seleção por CRZ (MF), COO (MFD), dia ou mês, exemplos:  
 CRZ: 0001 '0001' ou '1'  
 COO: 000039 '000039' ou '39'  
 Dezembro de 2005 '12/2005', '12/05'  
 Dia 12/10/2006 '12/10/06', '12/10/2006' ou '12102006'

*Fim* Documento final (string), opcional  
 Seleção por CRZ (MF), COO (MFD), dia ou mês, exemplos:  
 CRZ: 9999 '9999'  
 COO: 000040 '000040' ou '40'  
 Janeiro de 2006 '01/2006', '01/06'  
 Dia 12/10/2006 '12/10/06', '12/10/2006' ou '12102006'

*Opções* Opções de geração do arquivo (inteiro)

Constante	Valor	Descrição
opCOTEPE1704	0*	Formato definido no ato COPETE 17/2004
opCSV	2	Gera arquivo .CSV
opTDM	4	Opção TDM
opRS2L	32	Extrair a Razão Social da 2ª linha do clichê **
opWindowsCodePage*	0	Página de Código: Windows Code Page
opCodePage860	128	Página de Código: CP860 (Português)
opCodePage850	256	Página de Código: CP850 (Multilingual)
opAssinarArquivo	512	Assina digitalmente o arquivo

\*Padrão

\*\* A função extrai a Razão Social do contribuinte dos dados de identificação programados no ECF. Quando os campos Razão Social e Nome Fantasia são corretamente programados, a Razão Social é impressa na 1ª linha do clichê. Se as posições forem invertidas a Razão Social será impressa na 2ª linha. A inversão é desaconselhável, mas, a fim de evitar a inconsistência do arquivo-eletrônico, a inversão, se ocorrer, deve ser indicada pela opção opRS2L.

**Exemplo:**

```
#include "SWMFD.h"

// Exemplo.c

void main ()
{ SWEDA_ArquivoEletronicoCOTEPE1704 ("C:\\Sw010500000000336.BIN", "", "C:\\TESTE.TXT");
}
```

**Retorno:**

0 Ok  
 2 Data inválida!  
 3 Faixa inválida!  
 7 Falha na abertura do arquivo de origem!  
 10 Falha na abertura do arquivo de saída!

## Função SWEDA\_TabelaMercadoriasServicos

**Descrição:** A função SWEDA\_TabelaMercadoriasServicos gera, a partir das informações registradas na memória de fita-detalle (MFD) do ECF, a tabela de mercadorias e serviços do usuário; conforme as especificações do ATO COTEPE/ICMS N° 25/04, DE 8 DE JUNHO DE 2004 (alterado pelo ATO COTEPE/ICMS N° 34/04).

Na tabela, haverá apenas uma entrada para cada produto ou serviço.

São consideradas apenas as informações mais recentes do item; contidas no último registro efetuado pelo ECF no período selecionado.

**Parâmetros:**

<i>Destino</i>	Nome e diretório do arquivo de saída (string)
<i>Inicio</i>	Data inicial (string) Seleção por dia ou mês, exemplos: Dezembro de 2005    '12/2005', '12/05' Dia 12/10/2006    '12/10/06', '12/10/2006' ou '12102006'
<i>Fim</i>	Data final (string), opcional Seleção por dia ou mês, exemplos: Janeiro de 2006    '01/2006', '01/06' Dia 12/10/2006    '12/10/06', '12/10/2006' ou '12102006'

*Opções*                      Opções de performance da função (inteiro), opcional:

Constante	Valor	Descrição
OpTransferirMFD	8	Atualizar as informações do arquivo de MFD antes do processamento.
opMostrarJanela*	0	Exibir uma janela de acompanhamento
OpCancelamento	1024	Possibilita ao usuário interromper o processo.
OpOcultarJanela	2048	Ocultar a janela de acompanhamento
opAutoDetectarECF	16384	Faz auto-deteccção do ECF, se necessário
opWindowsCodePage *	0	Página de Código: Windows Code Page
opCodePage860	128	Página de Código: CP860 (Português)
opCodePage850	256	Página de Código: CP850 (Multilingual)

\*Opção padrão

*Origem*                      Nome e diretório do arquivo de origem, parâmetro opcional (string).  
Quando omitido este parâmetro, a função assume o diretório de gravação configurado na chave 'Diretório' do arquivo de inicialização (SWMFD.INI). E, o arquivo de nome padrão, correspondente ao ECF local.

*Log*                          Nome do arquivo log, opcional

**Retorno:**

0	Ok
2	Data inválida!
3	Faixa inválida!
7	Falha na abertura do arquivo de origem!
10	Falha na abertura do arquivo de saída!





## Função SWMFD\_ObterParametrosConfiguracao

**Descrição:** A função *SWMFD\_ObterParametrosConfiguracao* retorna o conjunto de parâmetros de configuração da SWMFD.DLL, que são lidos do arquivo SWMFD.INI na inicialização da DLL.

**Parâmetros:** *Buffer* Endereço do buffer de retorno, informado pela aplicação, que recebe a configuração;  
Tipo: SWMFD\_INI.  
*Extensão* Extensão em bytes do buffer informado no parâmetro anterior.

**Retorno:** Nenhum

## Função SWMFD\_AlterarParametrosConfiguracao

**Descrição:** A função *SWMFD\_AlterarParametrosConfiguracao* altera os parâmetros de configuração da SWMFD.DLL. A nova configuração será registrada no arquivo SWMFD.INI no encerramento da DLL.

**Parâmetros:** *Buffer* Endereço para estrutura de dados do tipo SWMFD\_INI.  
*Extensão* Extensão em bytes do buffer informado no parâmetro anterior.

**Retorno:** Nenhum

## Estrutura SWMFD\_INI

Parâmetros de configuração da SWMFD.DLL

Campo	Descrição	Tipo	Extensão
DiretorioSweda	Diretório raiz para gravação dos arquivos de dados transferidos do ECF.	Caracter	260
FAB	Número de Fabricação do ECF local	Caracter	22
MFD	Número serial do dispositivo de MFD instalado no ECF local	Caracter	22
Porta	Porta de comunicação serial (0 = Detectar, 1=COM1, ...)	Inteiro	2
Velocidade	Velocidade de comunicação (0 = Detectar, 9600, 19200, ...)	Inteiro	4
Estabelecimento	Dados de identificação do estabelecimenot	TDadosEstabelecimento	

## SWEDA\_VerificarAssinaturaDigital

**Descrição:** Verifica a assinatura digital de um arquivo.  
Os arquivos são assinados pelo fabricante através da biblioteca SWMFD (registro EAD)

**Parâmetros:** Nome e diretório do arquivo assinado

<b>Retorno:</b>	0	Assinatura Válida
	8	Arquivo não encontrado
	27	Arquivo não assinado
	28	Assinatura Inálida

### **Função SWEDA\_SWMFDVersao**

**Descrição:** Retorna valor numérico correspondente a versão da SWMFD.DLL

**Parâmetros:** Nenhum

**Retorno:** 118      que corresponde à versão 1.18

No código fonte, é declarada a constante SWMFD\_VERSAO com o número correspondente à versão atual da SWMFD.DLL, que compõe o kit de desenvolvimento (118).

## **FUNÇÕES COMPLEMENTARES**

Também, estão disponíveis funções complementares que permitem detectar e obter informações sobre o ECF; e, comandar a emissão de leituras fiscais ou a impressão da fita-detalhe.

### **Função SWEDA\_DetectarECF**

**Descrição:** Detecta as configurações relativas a porta de comunicação serial do PC utilizada.

**Parâmetros:***Opções*

Seleção de portas (inteiro), opcional:

Constante	Valor	Descrição
-	0*	Procura em todas as portas
-	1	Procurar exclusivamente na COM1
-	2	Procurar exclusivamente na COM2
-	3	Procurar exclusivamente na COM3
-	4	Procurar exclusivamente na COM4
-	5	Procurar exclusivamente na COM5
-	6	Procurar exclusivamente na COM6
-	7	Procurar exclusivamente na COM7
-	8	Procurar exclusivamente na COM8
opExcluirCOM1	16	Excluir a porta COM1
opExcluirCOM2	32	Excluir a porta COM2
opExcluirCOM3	64	Excluir a porta COM3
opExcluirCOM4	128	Excluir a porta COM4
opExcluirCOM5	256	Excluir a porta COM5
opExcluirCOM6	512	Excluir a porta COM6
OpExcluirCOM7	1024	Excluir a porta COM7
OpExcluirCOM8	2048	Excluir a porta COM8

\*Seleção padrão

A exclusão de portas, evita possíveis interferências do processo de detecção sobre outros periféricos.

*Porta*

Endereço de variável alocada pela aplicação, do tipo inteiro (2 bytes), que recebe o número correspondente à porta utilizada, ou zero, se o ECF não é detectado; parâmetro opcional.

*Velocidade*

Endereço de variável alocada pela aplicação, do tipo inteiro (4 bytes), que recebe o valor literal da velocidade de comunicação: 9600, 19200, 38400, 57600, 115200, ou zero, quando o ECF não é detectado; parâmetro opcional.

**Exemplo:**

```
#include "SWMFD.h"

// Exemplo.c

void main ()
{
    int Porta;
    long Velocidade;

    // Detecta o ECF em todas as portas de comunicação (COM1 a COM6)
    SWEDA_DetectarECF ();

    // Detecta o ECF conectado à COM2
    SWEDA_DetectarECF (2);

    // Detecta o ECF em todas as portas, exceto COM2 e COM3
    SWEDA_DetectarECF (opExcluirCOM2 + opExcluirCOM3);

    // Obtém o resultado da detecção
    if (SWEDA_DetectarECF (0, &Porta, &Velocidade) == true)
        printf ("\r\nECF detectado na COM%d a %ld bps\r\n\n", Porta, Velocidade);
    else
        printf ("\r\nECF não detectado!\r\n\n", Porta, Velocidade);
}
```

**Retorno:**

0 ECF não detectado!  
1 ECF detectado!

Quando detectado o ECF, as configurações da porta são atualizadas no arquivo de inicialização da DLL SWMFD.INI, que é encontrado na pasta do sistema ou da aplicação.

## Função SWEDA\_ConectarECF

**Descrição:** A função SWEDA\_ConectarECF obtém informações sobre o ECF conectado a porta de comunicação de seu computador

**Parâmetros(s)**

<i>Opções</i>	0: Somente conexão 1: Conexão com detecção automática dos parâmetros de configuração da porta de comunicação serial.
<i>buffer</i>	Endereço de uma estrutura de dados do tipo TPropriedadesDispositivo, parâmetro opcional;
<i>extensão</i>	Tamanho da estrutura TPropriedadesDispositivo (inteiro)

### Exemplo:

```
#include "SWMFD.h"
#include "stdio.h"

// Exemplo C

void main ()
{
    TPropriedadesDispositivo Propriedades;

    if (SWEDA_ConectarECF (1, &Propriedades, sizeof (Propriedades), 0) > 0)
    {
        switch (Propriedades.Arquivo.Tipo)
        {
            case 1: // 0 arquivo contém informações registradas na memória de fita-detelhe
            case 2: // 0 arquivo contém informações registradas na memória fiscal

                printf ("Tipo: %s\n"
                    "Modelo: %s\n"
                    "FAB: %s\n"
                    "Dispositivo: %s\n"
                    , Propriedades.ECF.Tipo
                    , Propriedades.ECF.Modelo
                    , Propriedades.ECF.FAB
                    , Propriedades.Arquivo.Dispositivo);

                // Informações da memória de fita-detelhe...
                if (Propriedades.Arquivo.Tipo == 1)
                printf ("N/S: %s\n"
                    "Capacidade: %s\n"
                    "Consumo: %s\n"
                    , Propriedades.MFD.NS
                    , Propriedades.MFD.Capacidade
                    , Propriedades.MFD.Consumo);

                // Informações sobre o(s) usuário(s) do ECF
                for (i = 0; i < Propriedades.Usuarios; i++)
                {
                    printf ("\n"
                        "Usuário: %d\n"
                        "  C.N.P.J.: %s\n ", i+1
                        "  I.E.: %s\n"
                        "  I.M.: %s\n"
                        "  MFD: %s\n"
                        , Propriedades.Usuario[i].CNPJ
                        , Propriedades.Usuario[i].IE
                        , Propriedades.Usuario[i].IM
                        , Propriedades.Usuario[i].MFD);
                }
                break;
        }
    }
}
```

```

        case 3: // software básico

            printf ("Software Básico\n" +
                "    Fabricante: %s\n"
                "    Tipo:      %s\n "
                "    Modelo:    %s\n "
                "    Versão:     %s\n "
                "    Checksum:   %s\n "
                , Propriedades.ECF.Marca
                , Propriedades.ECF.Tipo
                , Propriedades.ECF.Modelo
                , Propriedades.SB.Versao
                , &Propriedades.SB.Checksum[4]);

            break;
    }
}

```

**Retorno:** Valor booleano, verdadeiro (1) se obtiver sucesso na conexão com o ECF e falso (0), caso contrário.

## Função SWEDA\_EmitirLeituraX

**Propósito:** Emite a Leitura X.

**Parâmetro(s):** Nenhum

**Retorno:**

- 18 Erro na impressora (tampa aberta!)
- 19 Erro na impressora (sem papel!)
- 20 Erro na impressora (trocar bobina!)
- 21 Erro de sintaxe! Verificar parâmetros.
- 22 Comando inválido!
- 23 Falha na execução no comando!

## Função SWEDA\_EmitirLeituraMemoriaFiscal

**Propósito:** Emitir Leitura da Memória Fiscal do ECF.

**Parâmetro(s):**

<i>De</i>	Redução inicial String
<i>A</i>	Redução final String
<i>Completa</i>	Tipo de listagem: Numérico, 1: completa ou 0: simplificada

A faixa de reduções é selecionada por CRZ (Contador de Reduções Z) ou data. O CRZ deve ser informado com até 4 dígitos e a data em um dos seguintes formatos: *dd/mm/aaaa*, *dd/mm/aa* ou *ddmmaaaa*.

Dentro da faixa de reduções selecionada, a leitura simplificada imprime apenas as totalizações mensais referente ao período selecionado. A completa detalha os valores registrados a cada redução.

**Retorno:** 2 Data inválida!

- 3 Faixa inválida!
- 18 Erro na impressora (tampa aberta!)
- 19 Erro na impressora (sem papel!)
- 20 Erro na impressora (trocar bobina!)
- 21 Erro de sintaxe! Verificar parâmetros.
- 22 Comando inválido!
- 23 Falha na execução no comando!

### Função SWEDA\_EmitirLeituraParametros

**Propósito:** Emite Leitura dos Parâmetros da impressora fiscal.

**Parâmetro(s):** Nenhum

**Retorno:**

- 18 Erro na impressora (tampa aberta!)
- 19 Erro na impressora (sem papel!)
- 20 Erro na impressora (trocar bobina!)
- 21 Erro de sintaxe! Verificar parâmetros.
- 22 Comando inválido!
- 23 Falha na execução no comando!

### Função SWEDA\_ImprimirFitaDetalhe

**Propósito:** Imprimir a fita-detalle armazenada na MFD (Memória de Fita-Detalhe) do ECF.

**Parâmetro(s):**

<i>De</i>	COO ou data inicial. String.
<i>A</i>	COO ou data final. String.

A faixa de documento poderá ser selecionada ou por COO (Contador de Ordem de Operação) ou data. O COO deve ser informado em até 6 dígitos e a data em um dos seguintes formatos: *dd/mm/aaaa*, *dd/mm/aa* ou *ddmmaaaa*.

**Descrição:** Por força de legislação, a impressão da fita-detalle poderá ocorrer somente quando o ECF estiver em MIT (Modo de Intervenção Técnica).

**Retorno:**

- 2 Data inválida!
- 3 Faixa inválida!
- 18 Erro na impressora (tampa aberta!)
- 19 Erro na impressora (sem papel!)
- 20 Erro na impressora (trocar bobina!)
- 21 Erro de sintaxe! Verificar parâmetros.
- 22 Comando inválido!
- 23 Falha na execução no comando!

### Códigos de Erro da SWMFD.DLL

A tabela a baixo lista os códigos de erro retornados pelas funções da SWMFD.DLL e seus significados:

<b>Cod.</b>	<b>Descrição</b>
-2	Executando em segundo plano!
-1	Processo interrompido pela aplicação!
0	Ok
1	Falha de abertura do arquivo log!
2	Data inválida!
3	Faixa inválida!
4	O número de arquivos excede a capacidade da fila! (100 arquivos)
5	Nenhum arquivo encontrado! (opção: opArquivosPasta)
6	Nenhum arquivo selecionado! (opção: opListaEmArquivo)
7	Arquivo não encontrado! ( <i>Origem</i> )
8	Memória insuficiente!
9	Falha em arquivo temporário!
10	Não foi possível abrir o arquivo de saída!
11	Arquivo inválido! ( <i>Origem</i> )
12	Falha na criação da instância!
13	O mês selecionado não está encerrado!
14	Buffer insuficiente
15	Falha de gravação!
16	Falha de comunicação com o ECF (Timeout)
17	ECF off-line ou não detectado!
18	Erro na impressora (tampa aberta!)
19	Erro na impressora (sem papel!)
20	Erro na impressora (trocar bobina!)
21	Erro de sintaxe! Verificar parâmetros.
22	Comando inválido!
23	Falha na execução no comando!
24	Modelo incompatível.
25	Não houve movimento no período selecionado.
26	Não é possível gerar o arquivo tipo TDM a partir dos arquivos informados. O movimento referente ao período selecionado não está registrado em ambos os dispositivos.
27	Arquivo não assinado!
28	Assinatura inválida!