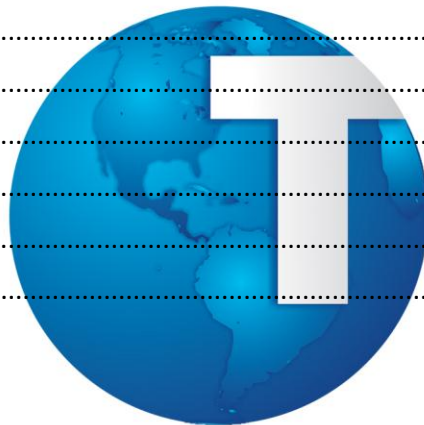


## Sumário

Sumário .....	1
1. Objetivo.....	2
2. Integração .....	3
3. Pré-requisitos: Instalação, Implantação e Utilização.....	9
4. Integração – Banco a Banco.....	9
5. Integração – Webservice .....	17
6. Integração - nativa .....	20
7. Controle de Versão .....	155
8. Fluxo das Informações .....	156
9. Como fazer .....	159



# TOTVS

Integração TAF



## 1. Objetivo

Disponibilizar no produto TAF rotinas de integração que poderão ser executadas tanto via menu como via *job*, ou seja, rotinas *scheduladas* conforme configuração de *Schedule* do sistema (SIGACFG - Configurador).

As formas de integração são disponibilizadas de 2 (duas) maneiras:

### 1. Integração Nativa

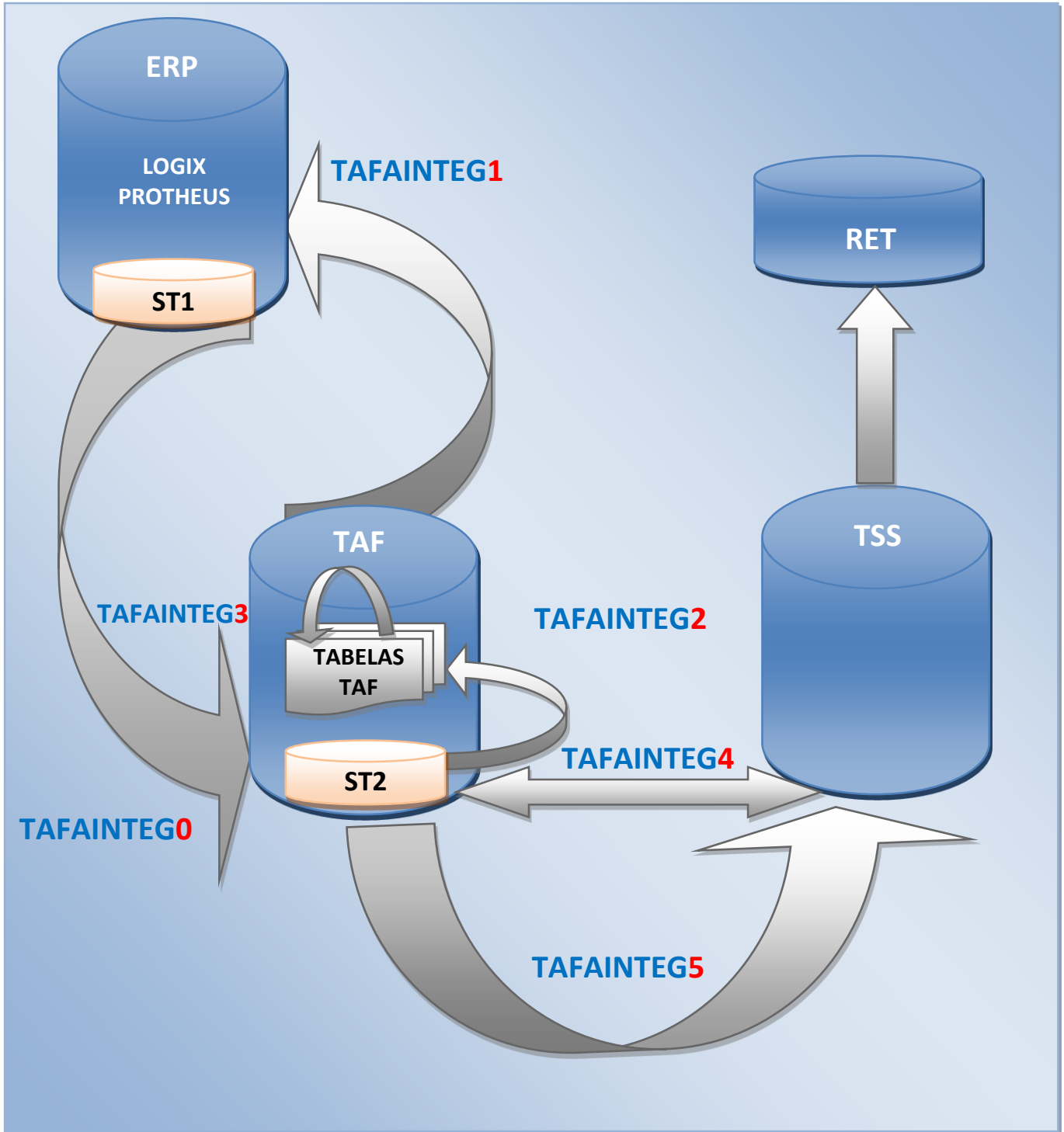
Neste cenário a integração do ERP grava as informações diretamente nas entidades do TAF, utilizando o próprio MVC. Este cenário serve para quando o ERP utilizar a mesma base (Dicionário de Dados/RPO) do produto TAF.

### 2. Integração banco-a-banco

Neste cenário utiliza-se conexão banco-a-banco para realizar a integração das informações. Este conceito utiliza a própria ferramenta DBAcces/TopConnect.

Com isso, a aplicação grava em uma tabela compartilhada e sob seu domínio, ou seja, no mesmo database, o XML criado por sua rotina de integração. Após gravá-lo, o TAF através de suas rotinas de monitoramento, processará os XMLs disponíveis e transportará para uma tabela de controle dentro de seu ambiente de processamento (TAF).

## 2. Integração





## TAF McIntGO

**Programa(s):** Deverá utilizar o nome **Integração ERP x TAF**.

**Descrição:** Busca as informações na ST1 no ERP e integra para a ST2 (TAF). Alterando o *flag* desses registros na ST1 (STATUS=2 - Integrados).

**Tabela(s):** Deverá utilizar a(s) tabela(s) **TAFST1 (Tafe Ésse Tê Um)**, **TAFST2 (Tafe Ésse Tê Dois)**. Ambas tratam-se de tabelas compartilhadas sendo ST na base de dados do ERP e ST2 na base de dados do TAF.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.

## TAF McIntG1

**Programa(s):** Deverá utilizar o nome **Limpar TAFST1**.

**Descrição:** Verifica os registros na TAFST1 (STATUS=2) integrados para limpá-lo, tornando ST1 volátil.

**Tabela(s):** Deverá utilizar a(s) tabela(s) **TAFST1 (Tafe Ésse Tê Um)** que é a tabela compartilhada na base de dados do ERP.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.

## TAF McIntG2

**Programa(s):** Deverá utilizar o nome **Integração TAFST2 x TAF**.

**Descrição:** Realiza a integração da TAFST2 para as tabelas internas do TAF. Integra as informações mesmo com inconsistências alimentando o *flag* nas tabelas de registros.

**Tabela(s):** Deverá utilizar a(s) tabela(s) **TAFST2 (Tafe Ésse Tê Dois)**.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.

## TAF McIntG3

**Programa(s):** Deverá utilizar o nome **Validações dos Dados**.

**Descrição:** Rotina que varre a base de dados procurando as inconsistências para apresentá-las no monitor de integrações.

**Tabela(s):** Deverá utilizar a(s) tabela(s) **TAFST2 (Tafe Ésse Tê Dois)** e todas as tabelas de base de dados do TAF.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.

## TAF McIntG4

**Programa(s):** Deverá utilizar o nome **Transmitir eventos**.

**Descrição:** Rotina verifica as informações válidas para transmissão ao TSS.

**Tabela(s):** Deverá utilizar a(s) tabela(s) da base de dados TAF.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.

## TAF McIntG5

**Programa(s):** Deverá utilizar o nome **Monitorar informações TAF**.

**Descrição:** Busca retorno dos registros transmitidos ao TSS e alimentar protocolo.

**Tabela(s):** Deverá utilizar a(s) tabela(s) da base de dados TAF.

**Opções de menu:** Será uma rotina de processamento que terá perguntas básicas para filtros do processamento.



Todas as tabelas referentes aos eventos terão o grupo de campos abaixo:

<b>Campo</b>	??_ID
<b>Tipo</b>	C
<b>Tamanho</b>	6
<b>Título</b>	ID
<b>Descrição</b>	Identificador do registro
<b>Picture</b>	@!
<b>Uso</b>	Usado por todos os módulos e Não Obrigatório.
<b>Inic.Padrão</b>	GetSx8Num("???", "??_ID")
<b>Browse</b>	Sim
<b>Visual</b>	Visualizar
<b>Contexto</b>	Real
<b>Help de Campo</b>	Representa o código de identificação do registro. Este código é controlado internamente pelo sistema de forma incremental e único por filial.

<b>Campo</b>	??_VERSAO
<b>Tipo</b>	C
<b>Grupo Campos</b>	071
<b>Tamanho</b>	14
<b>Título</b>	Id. Ver. Reg
<b>Descrição</b>	ID da versão do registro
<b>Picture</b>	@!
<b>Uso</b>	Não Usado
<b>Propriedade</b>	Microsiga
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Help de Campo</b>	Campo de uso interno do sistema. Representa a identificação da versão do registro. Será composto por Data + Hora da ação (Inclusão/Alteração/Exclusão).

<b>Campo</b>	??_VERANT
<b>Tipo</b>	C
<b>Grupo Campos</b>	071
<b>Tamanho</b>	14



<b>Título</b>	Ver Ant Reg
<b>Descrição</b>	Versão anterior registro
<b>Picture</b>	@!
<b>Uso</b>	Não Usado
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Help de Campo</b>	Campo de uso interno do sistema. Representa a identificação da versão anterior do registro. Este campo herdará a versão do registro anterior, quando se tratar de uma alteração/exclusão.

<b>Campo</b>	??_STATUS
<b>Tipo</b>	C
<b>Grupo Campos</b>	072
<b>Tamanho</b>	1
<b>Título</b>	Status Reg.
<b>Descrição</b>	Status do registro
<b>Picture</b>	@!
<b>Valid</b>	Pertence(" 012349")
<b>Uso</b>	Não Usado
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Combo Box</b>	0=Reg.Válido 1=Reg.Inválido 2=Reg.Transmitido 3=Reg.Transmitido com inconsistência 4=Reg.Transmitido válido 9=Em Processamento
<b>Help de Campo</b>	Campo de uso interno do sistema. Representa o status do registro:  (Vazio) = Registro Novo 0 = Registro Válido (integração) 1 = Registro invalido (integração) 2 = Registro transmitido 3 = Registro transmitido com inconsistência(s) 4 = Registro transmitido válido 9 = Em Processamento



<b>Campo</b>	??_PROTUL
<b>Tipo</b>	C
<b>Grupo Campos</b>	073
<b>Ordem</b>	16
<b>Tamanho</b>	44
<b>Título</b>	Ult. Prot.
<b>Descrição</b>	Prot. última transmissão
<b>Picture</b>	@!
<b>Uso</b>	Não Usado
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Help de Campo</b>	Campo de uso interno do sistema.  Deve representar neste campo o protocolo da última transmissão do registro

<b>Campo</b>	??_PROTPN
<b>Tipo</b>	C
<b>Grupo Campos</b>	073
<b>Ordem</b>	17
<b>Tamanho</b>	44
<b>Título</b>	Pnlt. Prot.
<b>Descrição</b>	Prot. penúltima transm.
<b>Picture</b>	@!
<b>Uso</b>	Não Usado
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Help de Campo</b>	Campo de uso interno do sistema. Representa o protocolo da penúltima transmissão do registro.

<b>Campo</b>	??_EVENTO
<b>Tipo</b>	C
<b>Grupo Campos</b>	074
<b>Tamanho</b>	1
<b>Título</b>	Id. Evento



<b>Descrição</b>	Identificação do evento
<b>Picture</b>	@!
<b>Valid</b>	Pertence("IAE")
<b>Uso</b>	Não Usado
<b>Inicio Padrão</b>	"I"
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Combo Box</b>	I=Inclusão A=Alteração E=Exclusão
<b>Help de Campo</b>	<p>Campo de uso interno do sistema. Representa a identificação do evento. Deve ser preenchido com:</p> <p>I = Inclusão A = Alteração E = Exclusão</p>

<b>Campo</b>	??_ATIVO
<b>Tipo</b>	C
<b>Grupo Campos</b>	075
<b>Tamanho</b>	1
<b>Título</b>	Reg. Ativo?
<b>Descrição</b>	Registro ativo?
<b>Valid</b>	Pertence("12")
<b>Uso</b>	Não Usado
<b>Inicio Padrão</b>	"1"
<b>Browse</b>	Não
<b>Visual</b>	Alterar
<b>Contexto</b>	Real
<b>Combo Box</b>	1=Sim 2=Não
<b>Help de Campo</b>	<p>Campo de uso interno do sistema. Indica se o registro está ativo:</p> <p>1 = Sim 2 = Não</p>





		Ação Inclusão	Ação Alteração/Retificação	Ação Exclusão (apenas para eventos cadastrais)
Inclusão EVENTO = "I"	Não Transmitido	Alteração direta do registro, mantendo o Evento como uma Inclusão	Alteração direta, mantendo o Evento como uma inclusão	Exclusão direta
	Transmitido	Gera um novo registro como um evento de Alteração e inativa o registro original na base TAF	Gera um novo registro como um evento de Alteração e inativa o registro original na base TAF	Gera um novo registro como um evento de Exclusão e inativa o registro original na base TAF
Alteração EVENTO = "A"	Não Transmitido	Alteração direta do registro, mantendo o Evento como uma Alteração	Alteração direta do registro, mantendo o Evento como uma Alteração	Exclusão direta e Ativa o registro anterior na base TAF
	Transmitido	Gera um novo registro como um evento de Alteração e inativa o registro original na base TAF	Gera um novo registro como um evento de Alteração e inativa o registro original na base TAF	Gera um novo registro como um evento de Exclusão e inativa o registro original na base TAF
Exclusão EVENTO = "E"	Não Transmitido	Inclui um novo registro como um evento do Inclusão	Alteração direta, alterando o Evento como uma Alteração	-
	Transmitido	Gera um novo registro como um evento de Inclusão	Gera um novo registro como um evento de Inclusão	Gera um novo registro herdando todas as informações do registro original e como um evento de Exclusão

### 3. Pré-requisitos: Instalação, Implantação e Utilização

- TAF – Protheus V. 11.90
- TOTVS DBAccess – V. 20131204

### 4. Integração – Banco a Banco

Primeiramente é necessário realizar a instalação do produto TAF. Mais informações na manual de instalação TAF. // *em desenvolvimento*.

Instalação do TOTVS DBAcces. Mais informações no *link*:

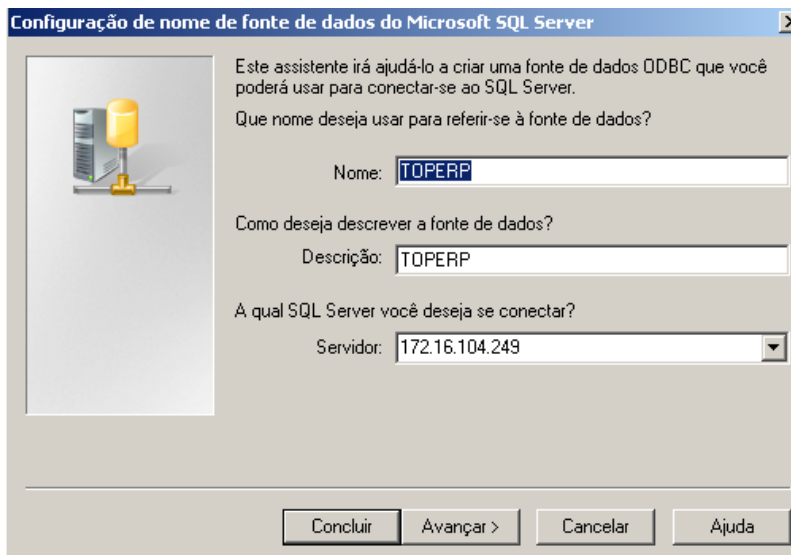
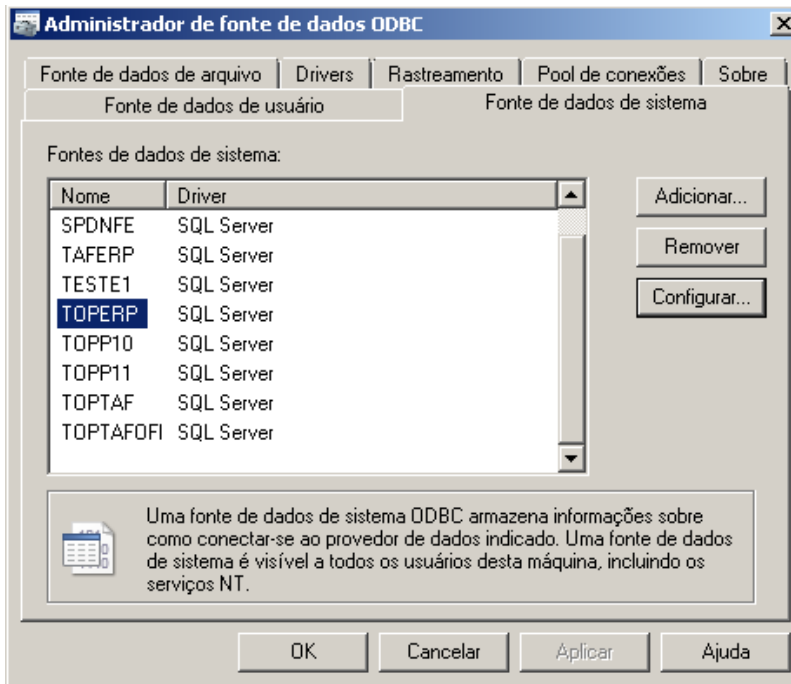
<http://tdn.totvs.com.br/display/tec/TOTVS++DBAccess+--+29106>.

Depois de configurado o ambiente local TAF, será necessário realizar as configurações para informar o local da base de dados do ERP, onde será criada e mantida a tabela compartilhada **TAFST1**.

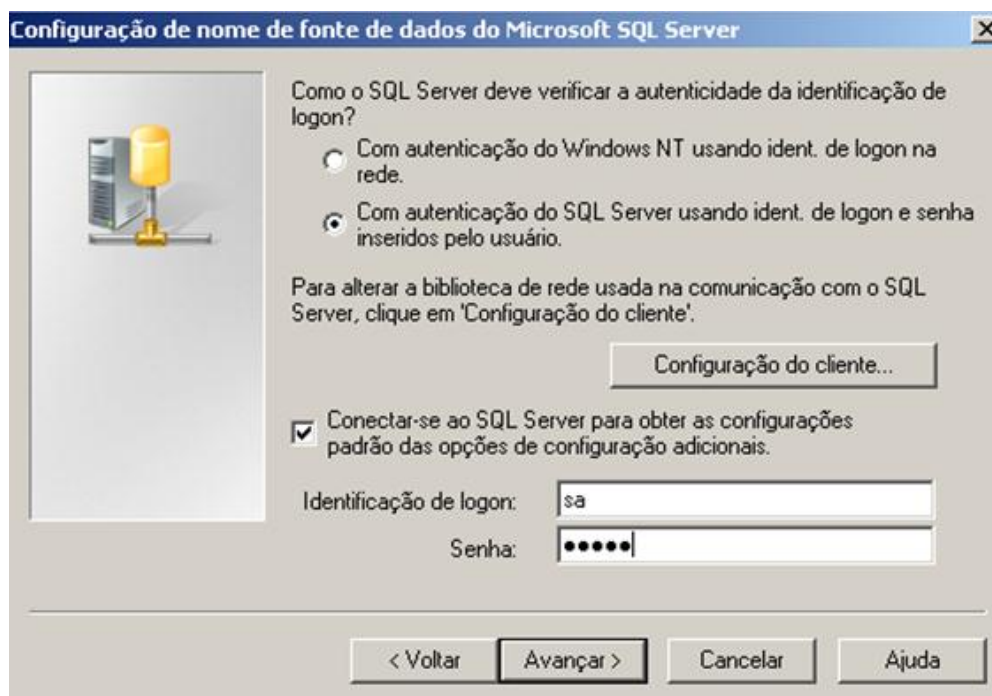
Para isso, utiliza-se o próprio ODBC (*Open Database Connectivity*) do sistema operacional, criando uma Fonte de dados do sistema, apontando para essa database.

**Exemplo:**

Fonte de dados ODBC Windows, base de dados SQL:

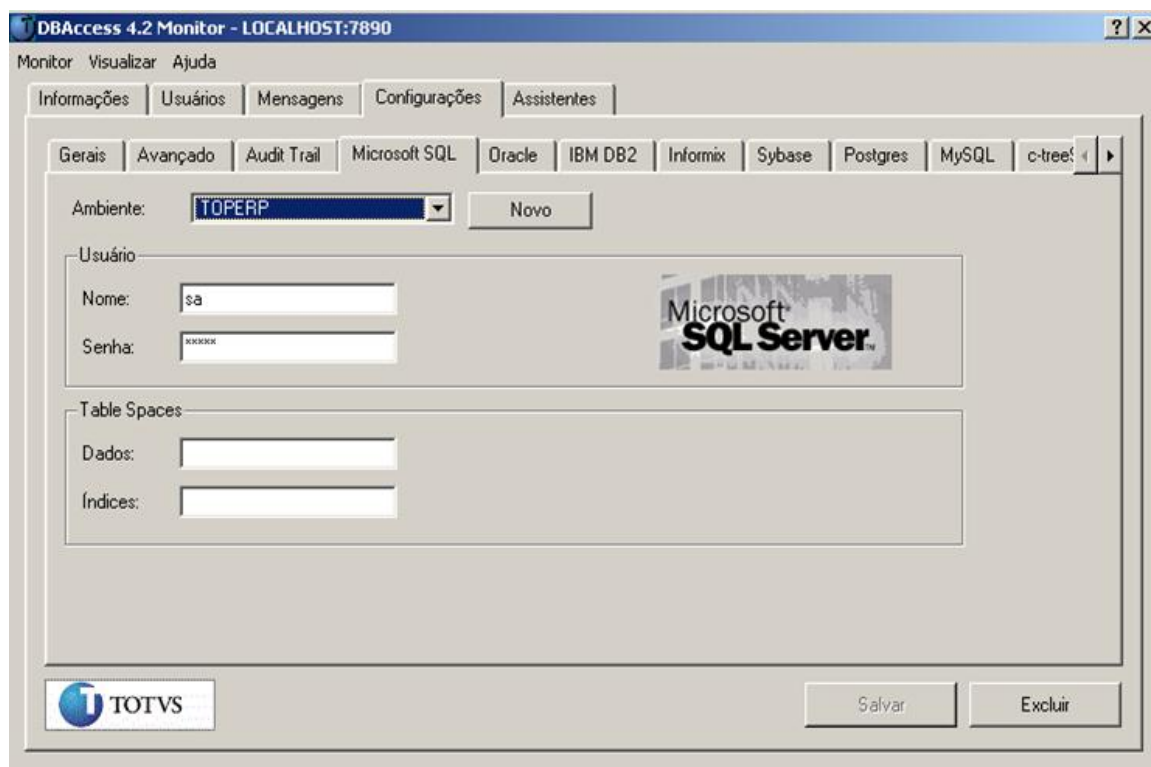


Neste exemplo, o banco de dados do ERP está no endereço: 172.16.104.249.



## Configuração do DBAccess:

Através do DBAccess Monitor





Desta forma, o DBAcces na base local terá acesso direto à base de dados do ERP.

Com isso, no ambiente TAF, é necessário configurar os parâmetros abaixo (Via Configurador):

<b>Parâmetro</b>	MV_TAFTDB
<b>Conteúdo (exemplo)</b>	MSSQL
<b>Descrição</b>	Indicar <b>Top Data Base</b> da base de dados da tabela TAFST1 (em domínio do ERP) para integração.

<b>Parâmetro</b>	MV_TAFTALI
<b>Conteúdo (exemplo)</b>	TOPERP
<b>Descrição</b>	Indicar <b>Top Alias</b> da base de dados da tabela TAFST1 (em domínio do ERP) para integração.

**! Importante**

Antes de qualquer integração deverá informada, no cadastro de estabelecimentos TAF, a empresa/estabelecimento.

Esse cadastro serve para relacionar a identificação da Empresa/Filial do ERP com a Empresa Filial no TAF.

Exemplo:

Ao ter registros na TAFST1, com o campo TAFFIL preenchido com a filial do ERP: "001".

	TAFFIL	TAFCDMSG	TAFSEQ	TAFTPREG	TAFKEY	TAFMSG
1	001	2	001	S-1000	2013101702:40:13S-1000 00010001	0x3C657674496E666F456D7072656761646F723E3C766572...

Através da rotina de Complemento de Estabelecimento (*mais informações na especificação técnica REQ002.S1000 – Empregador*), deve-se cadastrar previamente o relacionamento desta filial "001" com a filial correspondente na base TAF.

<b>Filial TAF*</b>	<b>Cod Filial*</b>
D MG 01	001

Caso a integração contemple múltiplos *softwares*, através deste mesmo cadastro estará disponibilizada no TAF.



Outras Filiais	
Cod.Filial	
001A	
001B	

### Especificação da Tabela Compartilhada:

- **Layout Shared Table (TAFST1/TAFST2)**

As informações transferidas para esta relação (tabela) será de responsabilidade da aplicação através de sua rotina de extração de movimento. Após devidamente populada, todo o restante do processamento é de controle do TAF, inclusive a remoção dos registros já processados, tornando-a volátil. Ela estará localizada na mesma instância da aplicação (ERPs) e será criada automaticamente pelo TAF no momento da implantação.

#### Definições

Esta relação se chamará **TAFST1 / TAFST2** e terá 7 atributos. Suas duplas terão como finalidade principal armazenar as mensagens, seja ela XML ou TXT (texto no *layout* TOTVS).

#### **TAFIL ( CHAR(10) ) NOT NULL**

Representa o código da empresa/filial vinculada ao TAF no Cadastro de Complemento do Estabelecimento. Esta informação qual estabelecimento o movimento pertence.

#### **TAFCODMSG ( CHAR(1) ) NOT NULL**

Representa o código da mensagem objeto da comunicação.

1, para mensagem no formato de arquivo texto (*layout* Totvs).

2, para mensagem no formato XML.

3, para mensagem NFe no formato XML (notas integradas via TSS).

#### **TAFSEQ ( CHAR(3) ) NOT NULL**

Representa sempre uma ordem crescente para processamento das informações. Principalmente nos casos de mesmos tipos de registros (TP\_REG) caso o tamanho total da mensagem ultrapasse 1 MB de limite do campo (MSG), especificado abaixo.

#### **TAFTPREG ( CHAR(10) ) NULL**

Representa o tipo de registro, este domínio da relação é **opcional/(OBRIGATÓRIO)** e poderá ser utilizado para quando a mensagem for segregada por registro, no caso do e-Social.

#### **TAFKEY ( CHAR(100) ) NOT NULL**

Representa uma chave de identidade da mensagem para controle do ERP em futuras consultas, atualizações, etc. Esta identidade (*key*) tem o objetivo de evitar duplicidades de informações caso a dupla ainda não tenha sido processada pelo TAF e uma nova inserção esteja sendo efetuada para a mesma chave FILIAL + COD\_MSG + SEQ + TP\_REG .



## TAFMSG ( BLOB ) NOT NULL

Representa a mensagem objeto da comunicação, conforme padrão definido pelo comitê e **limitado a 1 MB**. (VERIFICAR BANCOS).

Observação:

Um **BLOB (Binary Large Object - Grande Objeto Binário)** pode conter qualquer informação binária como imagens, vídeos, áudio, bem como qualquer outro tipo de multimídia ou dados em geral.

## TAFSTATUS ( CHAR(1) ) NOT NULL

Representa o status de processamento das informações por dupla.

**0**, indica que as informações estão sendo processadas pela aplicação (ERP).

**1**, indica que as informações estão disponíveis para que a integração a processe, ou seja, está disponível para o TAF processá-la.

**2**, indica que as informações estão sendo processadas.

**3**, indica que as informações já foram processadas e estão disponíveis no ambiente do TAF.

**7**, indica que houve erro na exclusão dos registros (somente para arquivo texto COD\_MSG = 1).

**8**, indica que a filial do registro não está cadastrada no sistema.

**7**, indica erro de estrutura no registro (numero de pipes menor que o esperado).

## TAFIDTHR ( CHAR(10) ) NOT NULL

Representa a identificação da *Thread* que processará o registro. É utilizada para um processamento com múltiplas *threads*.

## UNIQUE KEY

FILIAL + COD\_MSG + SEQ + TP\_REG + KEY

## PRIMARY KEY

FILIAL + COD\_MSG + SEQ + TP\_REG + KEY

FILIAL	COD_MSG	SEQ	TP_REG	TAFKEY	MSG	STATUS	TAFIDTHR
001	1	001	T001	20120701155000	T001 001 TESTE@TESTE.COM.BR ...	1	
001	1	001	T007	20120701155000	T007 000022333 ADFASDFSADSA...	1	
001	1	002	T007	20120701155000	T007 000044444 ADFASDFSADSA...	1	
001	1	003	T007	20120701155000	T007 000055555 ADFASDFSADSA...	0	
001	2	001	S1200	20120701155000	<A><B>SDFSDF</B><X>SAFASDFS...	1	
001	2	002	S1200	20120701155000	<A><B>RRRDFF</B><X>SAFASDFS...	0	

Neste exemplo, foi utilizado como KEY a data e hora de geração das duplas no formato (AAAAMMDDHHMMSS), porém fica a critério da aplicação o controle deste domínio.

## ! Importante

Considera-se informação apta para a integração todas àquelas que estiverem com status igual a "1".



Para exclusão dos registos com COD\_MSG = 1 (arquivo texto) deve-se substituir o Tipo de Registro (TP\_REG) pelo código T999:

FILIAL	COD_MSG	SEQ	TP_REG	TAFKEY	MSG	STATUS	TAFIDTHRD
001	1	001	T999	20120701155000	T001 001 TESTE@TESTE.COM.BR ...	1	
001	1	001	T999	20120701155000	T007 000022333 ADFASDFSADSA...	1	
001	1	002	T999	20120701155000	T007 000044444 ADFASDFSADSA...	1	

## ! Importante

O sistema se baseia na chave da tabela Pai para realizar a exclusão.

**A criação a manutenção da estrutura desta tabela TAFST1 será de responsabilidade do produto TAF.**

### Chave dos registos TAF

Registro	Chave única do Layout E-Social	Tam.Chave
S-1000	Filial + Filial de Origem das informações	8
S-1010	Filial + CodRubrica + iniValidade + fimValidade	46
'S-1020	Filial + CodLotacao + iniValidade + fimValidade	46
S-1030	Filial + CodCargo + iniValidade + fimValidade	46
S-1040	Filial + CodFuncao + iniValidade + fimValidade	46
S-1050	Filial + CodJornada + iniValidade + fimValidade	46
S-1060	Filial + TpInscricao + nrInscricao + iniValidade + fimValidade	31
S-1070	Filial + TpProcesso + nrProcesso + iniValidade + fimValidade	37
S-1080	Filial + CnpjOpPortuario + iniValidade + fimValidade	30
S-1100	Filial + indApuracao+perApuracao	7
S-1200	Filial + indApuracao+perApuracao+cpfTrab	19
S-1300	Filial + indApuracao+perApuracao	7
S-1310	Filial + indApuracao+perApuracao	7
S-1320	Filial + indApuracao+perApuracao	7
S-1330	Filial + indApuracao+perApuracao	7
S-1340	Filial + indApuracao+perApuracao	7
S-1350	Filial + indApuracao+perApuracao	7
S-1360	Filial + indApuracao+perApuracao	7
S-1370	Filial + indApuracao+perApuracao	7

S-1380	Filial + indApuracao+perApuracao	7
S-1390	Filial + indApuracao+perApuracao	7
S-1399	Filial + indApuracao+perApuracao	7
S-1400	Filial + indApuracao+perApuracao	7
S-1800	Filial + dtApuracao	8
S-2100	Filial + Matricula+cpfTrab	41
S-2200	Filial + Matricula+cpfTrab	41
S-2220	Filial + cpfTrab+dtAlteracao	19
S-2240	Filial + Matricula+cpfTrab+dtAlteracao	49
S-2260	Filial + cpfTrab+dtAcidente+horaAcidente	24
S-2280	Filial + Matricula+cpfTrab+dtAso	49
S-2320	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2325	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2330	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2340	Filial + Matricula+cpfTrab+dtIniEstabilidade	49
S-2345	Filial + Matricula+cpfTrab+dtFimEstabilidade	49
S-2360	Filial + Matricula+cpfTrab+dtIniCondicao+tpCondicao	51
S-2365	Filial + Matricula+cpfTrab+tpCondicao+dtFimCondicao	51
S-2400	Filial + Matricula+cpfTrab+dtAvisoPrevio	49
S-2405	Filial + Matricula+cpfTrab+dtCancAvisoPrevio	49
S-2600	Filial + Matricula+cpfTrab	40
S-2620	Filial + cpfTrab+dtAlteracao	19
S-2680	Filial + cpfTrab+dtTermino	19
S-2800	Filial + Matricula+cpfTrab+motivoDesligamento+dtDesligamento+indPagtoAPI	52
S-2820	Filial + Matricula+cpfTrab+tpReintegracao+nrProclJud	48
S-2900	Filial + tpEvento+cpfTrab	17

Para todos os registros do E-Social a chave única das tabelas é a mesma indicada no *Layout* do Fisco, com exceção do registro S-1000 - Informações do Empregador, onde a chave única é a filial de origem das informações.





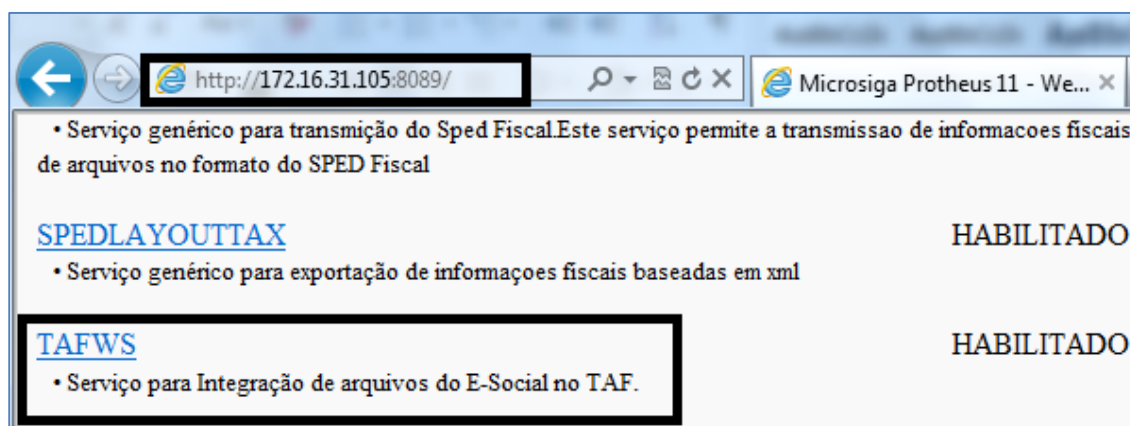
## 5. Integração – Webservice

A integração utilizando Webservice realiza a gravação dos arquivos diretamente na tabela ST2 dispensando a utilização da conexão banco a banco. Para a utilização deste serviço é necessário configurar um server para trabalhar como servidor web, sendo este preferencialmente separado do server do TAF criando-se um serviço novo. No serviço é necessário incluir as informações do servidor web no arquivo de configuração (appserver.ini) conforme **EXEMPLO** abaixo:

```
[HTTP]
Enable=1
Port=8089
Path=C:\Base_Oficial\TAF_Data\web
instances=40,100
SESSIONTIMEOUT=60000
```

```
[172.16.31.105:8089]
ENABLE=1
PATH=C:\Base_Oficial\TAF_Data\web\ws
ENVIRONMENT=TAF_CLOUD
RESPONSEJOB=JOB_WS
INSTANCENAME=WS
DEFAULTPAGE=wsindex.apw
```

Após a configuração do appserver.ini o servidor web vai estar disponível na Url configurada, no caso do exemplo acima 172.16.31.105:8089, digitar essa Url na barra de endereços de um navegador e verificar se o serviço **TAFWS** está disponível.



O Método para gravação dos arquivos é o **GRAVAST2** que possui as seguintes propriedades:

**Processo:** Assíncrono

**Função** – Método de recebimento dos XMLs do E-Social para gravação direta na tabela ST2.

**Url** - http://HOST:PORTA/TAFWS.apw



## Parâmetros de entrada:

```
<soapenv:Body>
<loc:GRAVAST2>
<loc:ESOCIAL>
<loc:REGESOCIAL>
<!--Zero or more repetitions:-->
<loc:REGISTRO>
<loc:CODFIL>STRING</loc:CODFIL>
<loc:CODMSG>STRING</loc:CODMSG>
<loc:SEQ>STRING</loc:SEQ>
<loc:TAFKEY>STRING</loc:TAFKEY>
<loc:TIPOREG>STRING</loc:TIPOREG>
<loc:XMLESOC>BASE64BINARY</loc:XMLESOC>
</loc:REGISTRO>
</loc:REGESOCIAL>
</loc:ESOCIAL>
</loc:GRAVAST2>
</soapenv:Body>
```

Parâmetro	Tipo	Tam	Pai	Observação
GRAVAST2	Estrutura	-	-	
ESOCIAL	Estrutura		GRAVAST2	
REGESOCIAL	Estrutura		ESOCIAL	
REGISTRO	Estrutura		REGESOCIAL	
CODFIL	Alfanumérico	10	REGESOCIAL	Filial do TAF que corresponde a filial do ERP.
CODMSG	Alfanumérico	01	REGESOCIAL	Representa o código da mensagem objeto da comunicação. (1-TXT,2-XML,3-Msg Nfe XML)
SEQ	Alfanumérico	03	REGESOCIAL	Sequencia do arquivo
TAFKEY	Alfanumérico	100	REGESOCIAL	Chave para identificação do Registro conforme <a href="#">Regra TAF</a>
TIPOREG	Alfanumérico	10	REGESOCIAL	Código do evento e-social
XMLESOC	BASE64Binary		REGESOCIAL	XML do evento, conforme Manual de Orientação do e-Social.

## Parâmetros de saída:

```
<soap:Body>
<GRAVAST2RESULT>
<REGRETORNO>
<RETORNO>
<MENSAGEM>STRING</MENSAGEM>
<RETORNO>BOOLEAN</RETORNO>
```



```
<TAFKEY>STRING</TAFKEY>
</RETORNO>
</REGRETORNO>
</GRAVAST2RESULT>
</soap:Body>
```

Parâmetro	Tipo	Tam	Pai	Observação
GRAVAST2RESULT	Estrutura	-	-	
REGRETORNO	Estrutura		GRAVAST2RESULT	
RETORNO	Estrutura		ESOCIAL	
MENSAGEM	Estrutura		REGESOCIAL	Mensagem de retorno.
RETORNO	Boolean		REGESOCIAL	Retorno lógico da requisição
TAFKEY	Alfanumérico	100	REGESOCIAL	Chave para identificação do Registro conforme <a href="#">Regra TAF</a>

### Parâmetros do sistema:

X6_VAR	MV_TAFVXML
X6_TIPO	Lógico
X6_DESCRIC	Determina se deve ser efetuado a validação dos Xml's na Integração dos registros para tabela TAFST2
X6_CONTEUD	.F.

X6_VAR	MV_TAFVLAY
X6_TIPO	Lógico
X6_DESCRIC	Determina se deve ser efetuado a validação dos Layouts na Integração dos registros para tabela TAFST2
X6_CONTEUD	.T.

## 6. Integração - nativa

Para integração nativa é utilizado o próprio MVC para gravações diretas nas tabelas TAF, ficando como opcional a utilização da conexão entre banco de dados e utilização das tabelas compartilhadas (ST – *Shared Table*). Isso será feito quando o Protheus RH instalado for a partir da versão 11.00, ou seja, o TAF estará automaticamente no mesmo ambiente (Binário, Dicionário e Repositório de Dados).

Tabelas TAF:

Tabela	Requisito	Descrição	Fonte PRW
C87	REQ001.01	Cadastro de Categoria de Trabalhadores	TAFA212
C88	REQ001.02	Cadastro de Grau de Exposição a Agentes Nocivos	TAFA213
C89	REQ001.03	Cadastro de Natureza das Rubricas da folha de pagamento	TAFA214
C8A	REQ001.04	Cadastro de Códigos e Alíquotas de Fpas - Terceiro	TAFA215
C8C	REQ001.06	Cadastro de classificação de Serv. Ret. Contrib. Previd.	TAFA217
C94	REQ001.07	Cadastro dos Riscos Operacionais Específicos	TAFA240
C95	REQ001.07	Cadastro dos Riscos Operacionais Específicos	TAFA241
C96	REQ001.07	Cadastro dos Riscos Operacionais Específicos	TAFA242
C97	REQ001.07	Cadastro dos Riscos Operacionais Específicos	TAFA243
C98	REQ001.07	Cadastro dos Riscos Operacionais Específicos	TAFA244
C8D	REQ001.08	Classificação Tributária	TAFA219
C8E	REQ001.09	Cadastro de Tipos de Arquivo da e-Social	TAFA220
C8F	REQ001.10	Cadastro de Tipos de Lotação	TAFA221
C8G	REQ001.11	Cadastro de Códigos x Siglas x Alíquotas de Outras Entidades e Fundos	TAFA222
C8H	REQ001.12	Cadastro de tipos de Contribuição	TAFA223
C8I	REQ001.13	Cadastro de Parte do Corpo Atingida	TAFA224
C8J	REQ001.14	Cadastro de Agente Causador do Acidente do Trabalho	TAFA225
C8K	REQ001.15	Cadastro de Agente Causador-Sit. Ger. Doença Prof.	TAFA226
C8L	REQ001.16	Cadastro de Situação Geradora do Acidente de Trabalho	TAFA227
C8M	REQ001.17	Cadastro de Descrição da Natureza da Lesão	TAFA228
C8N	REQ001.18	Cadastro de Motivos de Afastamento	TAFA229
C8O	REQ001.19	Cadastro de Motivos de Desligamento	TAFA230
C06	REQ001.20	Cadastro de Tipos de Logradouro	TAFA006
C8P	REQ001.21	Cadastro de Natureza Jurídica do Contribuinte	TAFA231
C8T	REQ001.22	Cadastro de códigos de incidência tributária da rubrica para a Previdência Social	TAFA233
C8U	REQ001.23	Cadastro de Códigos de incidência tributária da rubrica para o IRRF	TAFA234
C8Z	REQ001.24	Cadastro do Código Brasileiro de Ocupação	TAFA237
C8S	REQ001.25	Cadastro de Indicativo de Decisão	TAFA247
CR9	REQ002	Complemento do cadastro de Complemento de Empresa - C1E	TAFA050
C1E	REQ002	S-1000 - Informações do Empregador	TAFA050

# Integração TAF



C8R	REQ003	S-1010 - Tabelas de Rubricas	TAFA232
C99	REQ004	S-1020 - Tabela de Lotação / Departamento	TAFA246
C8V	REQ005	S-1030 - Tabela de Cargos	TAFA235
C8X	REQ006	S-1040 - Tabela de Funções	TAFA236
C90	REQ007	S-1050 - Tabela de Horários / Turnos de Trabalho	TAFA238
C92	REQ008	S-1060 - Cadastro de Estabelecimento	TAFA253
C93	REQ009	S-1070 - Tabela de Processos	TAFA245
C8W	REQ010	S-1080 - Tabela de Operadores Portuários	TAFA248
C9V	REQ011	S-2100,S-2200,S-2220,S-2240 - Cadastro de Funcionários	TAFA256
C9Y	REQ011	S-2100,S-2200,S-2220,S-2240 - Cadastro de Funcionários	TAFA256
C9Z	REQ011	S-2100,S-2200,S-2220,S-2240 - Cadastro de Funcionários	TAFA256
CMH	REQ011.1	Cadastro de Grau de Instrução	TAFA216
CMI	REQ011.2	Cadastro de Tipo de Dependente	TAFA218
CM0	REQ013	S-2260 - Comunicação de Acidente de Trabalho	TAFA257
CM1	REQ013	S-2260 - Comunicação de Acidente de Trabalho	TAFA257
CM2	REQ013	S-2260 - Comunicação de Acidente de Trabalho	TAFA257
CM3	REQ013	S-2260 - Comunicação de Acidente de Trabalho	TAFA257
CM7	REQ013	S-2260 - Comunicação de Acidente de Trabalho / Cadastro de Médico	TAFA262
C8B	REQ014	S-2280 - Atestado de Saúde Ocupacional	TAFA258
C9W	REQ014	S-2280 - Atestado de Saúde Ocupacional	TAFA258
C9X	REQ014	S-2280 - Atestado de Saúde Ocupacional	TAFA258
CM6	REQ016	S-2320 - Afastamento Temporário	TAFA261
CMM	REQ016.1	Classificação Internacional de Doenças	TAFA270
CM5	REQ017.01	S-2340 - Estabilidade – Início	TAFA260
CRH	REQ017.02	S-2345 - Estabilidade – Término	TAFA287
CM9	REQ018	S-2360,S-2365 - Condição diferenciada de trabalho	TAFA264
CMA	REQ018	S-2360,S-2365 - Condição diferenciada de trabalho	TAFA264
CMB	REQ018	S-2360,S-2365 - Condição diferenciada de trabalho	TAFA264
CM8	REQ019	S-2400 - Aviso Prévio e S2405 - Cancelamento Aviso Prévio	TAFA263
CMK	REQ022	S-2600,S-2620,S-2680 - Trabalhador sem vínculo	TAFA256
CML	REQ022	S-2600,S-2620,S-2680 - Trabalhador sem vínculo	TAFA273
CMD	REQ023	S-2800 - Desligamento	TAFA266
CME	REQ023	S-2800 - Desligamento	TAFA266
CMF	REQ023	S-2820 - Reintegração por determinação Judicial	TAFA267
CMJ	REQ024	S2900 - Exclusão de Evento Enviado Indevidamente	TAFA269
C8Y	REQ025	S-1100 - E-Social Mensal - Abertura	TAFA249
C91	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9J	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9K	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9L	REQ026	S-1200 - Folha de Pagamento	TAFA250



C9M	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9N	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9O	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9P	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9Q	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9R	REQ026	S-1200 - Folha de Pagamento	TAFA250
CRN	REQ026	S-1200 - Folha de Pagamento	TAFA250
C9U	REQ026	S-1200 - Folha de Pagamento	TAFA250
CUS	REQ026	S-1200 - Folha de Pagamento	TAFA250
CUT	REQ026	S-1200 - Folha de Pagamento	TAFA250
CMN	REQ027	S-1310 - Mensal – Outras Informações - Serv. Tomados (Cessão de Mão de Obra)	TAFA271
CMO	REQ027	S-1310 - Mensal – Outras Informações - Serv. Tomados (Cessão de Mão de Obra)	TAFA271
CMP	REQ027	S-1310 - Mensal – Outras Informações - Serv. Tomados (Cessão de Mão de Obra)	TAFA271
CMQ	REQ027	S-1310 - Mensal – Outras Informações - Serv. Tomados (Cessão de Mão de Obra)	TAFA271
CMR	REQ031	S-1350 - Mensal – Outras Informações - Aquisição de Produção	TAFA272
CMS	REQ031	S-1350 - Mensal – Outras Informações - Aquisição de Produção	TAFA272
CMT	REQ031	S-1350 - Mensal – Outras Informações - Aquisição de Produção	TAFA272
CMU	REQ031	S-1350 - Mensal – Outras Informações - Aquisição de Produção	TAFA272
CMV	REQ031	S-1350 - Mensal – Outras Informações - Aquisição de Produção	TAFA272
C8Q	REQ032	S-1360 - Comercialização da Produção	TAFA252
C9C	REQ032	S-1360 - Comercialização da Produção	TAFA252
C9D	REQ032	S-1360 - Comercialização da Produção	TAFA252
C9E	REQ032	S-1360 - Comercialização da Produção	TAFA252
C9B	REQ033	S-1370 - Rec. Recebidos ou Repassados para Clube de Futebol	TAFA255
C9G	REQ033	S-1370 - Rec. Recebidos ou Repassados para Clube de Futebol	TAFA255
C9H	REQ033	S-1370 - Rec. Recebidos ou Repassados para Clube de Futebol	TAFA255
C9I	REQ033	S-1370 - Rec. Recebidos ou Repassados para Clube de Futebol	TAFA255
CMX	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CMZ	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR0	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR1	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR2	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR3	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR3	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR4	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR5	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR6	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR7	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CR8	REQ034	S-1400 - Mensal – Bases, Retenção, Deduções e Contribuições	TAFA274
CRS	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298



CRT	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUG	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUH	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUI	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUJ	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUK	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
CUL	REQ003.003129	S-1800 - Espetáculo Desportivo	TAFA298
C9A	REQ051	Cadastro de Varas	TAFA251
C9F	REQ052	Cadastro de Responsável pelas Informações	TAFA254

## Eventos:

S-1000 – Informações do Empregador					
<b>Rotina:</b>	TAF050.prw				
<b>Model MVC:</b>	MODEL_C1E				
<b>Validação</b>	TAF050VLD()				
<b>Geração XML</b>	TAF050XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt ()</b>
	01	eSocial	-		
	02	evtInfoEmpregador	-		
	03	id	-		
	04	versao	-		
	05	ideEvento	-		
	06	tpAmb	-		
	07	procEmi	-		
	08	indSegmento	-		
	09	verProc	-		
	10	ideEmpregador	-		
	11	tpInscricao	-		
	12	nrInscricao	-		
	13	infoEmpregador	-		
	14	Inclusão / alteração	-		
	15	idePeriodo	-		
	16	iniValidade	MODEL_C1E	C1E_DTINI	
	17	fimValidade	MODEL_C1E	C1E_DTFIN	
	18	infoCadastro	-		
	19	nomeRazao	MODEL_C1E	C1E_NOME	
	20	classTrib	MODEL_C1E	C1E_CLAFIS	"classTrib"
	21	natJuridica	MODEL_C1E	C1E_NATJUR	"natJuridica"
	22	cnaePreponderante	MODEL_C1E	C1E_CNAE	
	23	indCooperativa	MODEL_C1E	C1E_INCOOP	
	24	indConstrutora	MODEL_C1E	C1E_INCONS	
	25	aliqRat	-		
	26	aliqRat	MODEL_C1E	C1E_ALIRAT	
27	fap	MODEL_C1E	C1E_FAP		



28	aliqRatAjustada	MODEL_C1E	C1E_AJURAT	
29	procAdmJudRat	-		
30	tpProcesso	MODEL_C1E	C1E_PRFRAT	"tpProcesso","nrProcesso"
31	nrProcesso	-	-	
32	procAdmJudFap	-		
33	tpProcesso	MODEL_C1E	C1E_PRFFAP	"tpProcesso","nrProcesso"
34	nrProcesso	-	-	
35	dadosIsencao	-		
36	siglaMin	MODEL_C1E	C1E_SIGMIN	
37	nrCertificado	MODEL_C1E	C1E_NRCERT	
38	dtEmissaoCertificado	MODEL_C1E	C1E_DTEMCE	
39	dtVenctoCertificado	MODEL_C1E	C1E_DTVCCCE	
40	nrProtRenovacao	MODEL_C1E	C1E_NRPRRE	
41	dtProtRenovacao	MODEL_C1E	C1E_DTPRRE	
42	dtDou	MODEL_C1E	C1E_DTDYOU	
43	pagDou	MODEL_C1E	C1E_PAGDOU	
44	contato	-		
45	nomeContato	MODEL_C1E	C1E_NOMCNT	
46	cpfContato	MODEL_C1E	C1E_CPF CNT	
47	foneFixo	MODEL_C1E	C1E_FONCNT	
48	foneCelular	MODEL_C1E	C1E_CELCNT	
49	fax	MODEL_C1E	C1E_FAXCNT	
50	email	MODEL_C1E	C1E_EMAIL	
51	infoOrgInternacional	-		
52	indAcordoisencaoMulta	MODEL_C1E	C1E_ISEMUL	
53	softwareHouse	-		
54	cnpjSoftwareHouse	MODEL_CRM	CRM_CNPJ	
55	nomeRazao	MODEL_CRM	CRM_NOME	
56	nomeContato	MODEL_CRM	CRM_CONTAT	
57	telefone	MODEL_CRM	CRM_FONE	
58	codMunicipio	MODEL_CRM	CRM_MUN	"uf", "codMunicipio"
59	uf	MODEL_CRM	CRM_UF	uf
60	email	MODEL_CRM	CRM_MAIL	
61	infoComplementares	-		
62	indSocioOstensivo	MODEL_C1E	C1E_SOCOST	
63	indSituacaoEspecial	MODEL_C1E	C1E_SITESP	

## Exemplo:

```
Begin Transaction
```

```
Aadd( aChave, { "C", "C1E_CODFIL", "001", .T. } )
```

```
//0AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
```

```
//3Função para validar se a operação desejada pode ser realizada3
```

```
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÙ
```

```
If FTafVldOpe( 'C1E', 7, @nOpc, cFilEv, @aIncons, aChave, @oModel, 'TAF050', cCmpsNoUpd, 2 )
```

```
//0AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
```

```
//3Quando se tratar de uma Exclusão direta, é preciso apenas realizar 3
```

```
//3o Commit(), não sendo necessária nenhuma manutenção nas informações3
```

```
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÙ
```

```
If nOpc <> 5
```





```

oModel:LoadValue( "MODEL_C1E", "C1E FILTAF", C1E->C1E FILTAF )
oModel:LoadValue( "MODEL_C1E", "C1E CODFIL", C1E->C1E CODFIL )

oModel:LoadValue( "MODEL_C1E", "C1E_FILIAL", xFilial( "C1E" ) )
oModel:LoadValue( "MODEL_C1E", "C1E_VERSAO", xFunGetVer() )

oModel:LoadValue( "MODEL_C1E", "C1E DTINI", "012013" )
oModel:LoadValue( "MODEL_C1E", "C1E DTFIN", "012013" ) )
oModel:LoadValue( "MODEL_C1E", "C1E_NOME", "Nome da Empresa" )
oModel:LoadValue( "MODEL_C1E", "C1E_CNAE", "CNAE01" )
oModel:LoadValue( "MODEL_C1E", "C1E_INCOOP", "1" )

//³Quando se tratar de um código de Tabela do TAF, ou seja, se refere³
//³a um ID, deve-se utilizar a função FGetIdInt() para que o ID seja ³
//³retornado
oModel:LoadValue( "MODEL_C1E", "C1E_BANCO", FGetIdInt( "banco", "033" ) )

EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAA¿
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAU
FWFormCommit( oModel )

oModel:DeActivate()

EndIf

End Transaction
    
```

## S-1010 – Tabela de Rubricas

<b>Rotina:</b>	TAF232.prw
<b>Model MVC:</b>	MODEL_C8R



Validação	TAF232VLD()				
Geração XML	TAF232XML()				
Regras	#	TAG	Model	Campo	FGetIDInt()
	01	eSocial	-		
02	evtTabRubrica	-			
03	Id	-			
04	versao	-			
05	ideEvento	-			
06	tpAmb	-			
07	procEmi	-			
08	indSegmento	-			
09	verProc	-			
10	ideEmpregador	-			
11	tpInscricao	-			
12	nrInscricao	-			
13	infoRubrica	-			
14	Inclusão / alteração	-			
15	ideRubrica	-			
16	codRubrica	MODEL_C8R		C8R_CODRUB	
17	iniValidade	MODEL_C8R		C8R_DTINI	
18	fimValidade	MODEL_C8R		C8R_DTFIN	
19	dadosRubrica	-			
20	descRubrica	MODEL_C8R		C8R_DESRUB	
21	natRubrica	MODEL_C8R		C8R_NATRUB	natRubrica
22	indProvDesc	MODEL_C8R		C8R_INDTRB	
23	codIncidCP	MODEL_C8R		C8R_CINTPS	codIncidCP
24	codIncidIRRF	MODEL_C8R		C8R_CINTIR	codIncidIRRF
25	codIncidFGTS	MODEL_C8R		C8R_CINTFG	
26	codIncidSIND	MODEL_C8R		C8R_CINTSL	
27	repDSR	MODEL_C8R		C8R_REPDSR	
28	repDecTerceiro	MODEL_C8R		C8R_REPDTE	
29	repFerias	MODEL_C8R		C8R_REPFER	
30	repRescisao	MODEL_C8R		C8R_REPREC	
31	fatorRubrica	MODEL_C8R		C8R_FATRUB	
32	ideProcessoCP	-			
33	tpProcesso	MODEL_C8R		C8R_PROCCP	tpProcesso,nrProcesso
34	nrProcesso	MODEL_C8R		C8R_PROCCP	tpProcesso,nrProcesso
35	extDecisao	MODEL_C8R		C8R_PROCCP	tpProcesso,nrProcesso
36	ideProcessoIRRF	-			
37	tpProcesso	MODEL_C8R		C8R_PROCI	tpProcesso,nrProcesso
38	nrProcesso	MODEL_C8R		C8R_PROCI	tpProcesso,nrProcesso
39	ideProcessoFGTS	-			
40	tpProcesso	MODEL_C8R		C8R_PROCFG	tpProcesso,nrProcesso
41	nrProcesso	MODEL_C8R		C8R_PROCFG	tpProcesso,nrProcesso
42	ideProcessoSIND	-			
43	tpProcesso	MODEL_C8R		C8R_PROCCS	tpProcesso,nrProcesso
44	nrProcesso	MODEL_C8R		C8R_PROCCS	tpProcesso,nrProcesso

### Exemplo:

```
Aadd( aChave, { "C", "C8R_CODRUB", "00000001", .T. } )
Aadd( aChave, { "C", "C8R_DTINI", "112013", .T. } )
Aadd( aChave, { "C", "C8R_DTFIN", "122013", .T. } )
```



## S-1020 – Tabela de Lotações

<b>Rotina:</b>	TAF246.prw				
<b>Model MVC:</b>	MODEL_C99				
<b>Validação</b>	TAF246VLD()				
<b>Geração XML</b>	TAF246XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtTabLotacao	-		
	02	versao	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	tpAmb	-		
	07	procEmi	-		
	08	indSegmento	-		
	09	verProc	-		
	10	ideEmpregador	-		
	11	tpInscricao	-		
	12	nrInscricao	-		
	13	infoLotacao	-		
	14	Inclusão / alteração	-		
	15	ideLotacao	-		
	16	codLotacao	MODEL_C99	C99_CODIGO	
	17	iniValidade	MODEL_C99	C99_DTINI	
	18	fimValidade	MODEL_C99	C99_DTFIN	
	19	dadosLotacao	-		
	20	descLotacao	MODEL_C99	C99_DESCRI	
	21	tpLotacao	MODEL_C99	C99_TPLOT	tpLotacao
	22	tpInscEstab	MODEL_C99	C99_TPINES	
	23	nrInscEstab	MODEL_C99	C99_NRINES	
	24	Endereço	-		
	25	tpLogradouro	MODEL_C99	C99_TPLOGR	tpLogradouro
	26	descLogradouro	MODEL_C99	C99_DESLOG	
	27	nrLogradouro	MODEL_C99	C99_NUMLOG	
	28	Complemento	MODEL_C99	C99_COMLOG	
	29	Bairro	MODEL_C99	C99_BAIRRO	
	30	CEP	MODEL_C99	C99_CEP	
	31	codMunicipio	MODEL_C99	C99_CODMUN	Uf, codMunicipio
	32	Uf	MODEL_C99	"C99_UF"	Uf
	33	fpasLotacao	-		
	34	Fpas	MODEL_C99	C99_FPAS	Fpas,codTerceiros
	35	codTerceiros	MODEL_C99	C99_FPAS	Fpas,codTerceiros
	36	infoEmprParcial	-		
	37	tpInscContratante	MODEL_C99	C99_TPINCT	
	38	nrInscContratante	MODEL_C99	C99_NRINCT	
	39	tpInscProprietario	MODEL_C99	C99_TPINPR	
40	nrInscProprietario	MODEL_C99	C99_NRINPR		







## S-1040 – Tabela de Funções

<b>Rotina:</b>	TAF236.prw				
<b>Model MVC:</b>	MODEL_C8X				
<b>Validação</b>	TAF236VLD()				
<b>Geração XML</b>	TAF236XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtTabFuncao	-		
	03	Id	-		
	04	versao	-		
	05	ideEvento	-		
	06	tpAmb	-		
	07	procEmi	-		
	08	indSegmento	-		
	09	verProc	-		
	10	ideEmpregador	-		
	11	tpInscricao	-		
	12	nrInscricao	-		
	13	infoFuncao	-		
	14	Inclusão / alteração	-		
	15	ideFuncao	-		
	16	codFuncao	MODEL_C8X	C8X_CODIGO	
	17	iniValidade	MODEL_C8X	C8X_DTINI	
	18	fimValidade	MODEL_C8X	C8X_DTFIN	
	19	dadosFuncao	-		
20	descFuncao	MODEL_C8X	C8X_DESCRI		

### Exemplo:

Begin Transaction

```

Aadd( aChave, { "C", "C8X_CODIGO", "Codigo", .T. } )
Aadd( aChave, { "C", "C8X_DTINI", "112013", .T. } )
Aadd( aChave, { "C", "C8X_DTFIN", "122013", .T. } )

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
If FTafVldOpe( 'C8X', 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, 'TAF236', cCmpsNoUpd )

    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
    //³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
    //³o Commit(), não sendo necessária nenhuma manutenção nas informações³
    //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
    If nOpc <> 5
        oModel:LoadValue( "MODEL_C8X", "C8X_CODIGO", "Codigo" )
        oModel:LoadValue( "MODEL_C8X", "C8X_DESCRI", "Descrição" )

    EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
    
```



```
FWFormCommit ( oModel )
```

```
oModel:DeActivate ()
```

```
EndIf
```

```
End Transaction
```











```
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
If nOpc <> 5
    oModel:LoadValue( "MODEL_C92", "C92_TPINSC", "1" )
    oModel:LoadValue( "MODEL_C92", "C92_CNAE", "123456" )
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAç
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÙ
FWFormCommit( oModel )

oModel:DeActivate()

EndIf

End Transaction
```



## S-1070 – Tabela de Processos

<b>Rotina:</b>	TAF245.prw				
<b>Model MVC:</b>	MODEL_C93				
<b>Validação</b>	TAF245VLD()				
<b>Geração XML</b>	TAF245XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtTabProcesso	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	tpAmb	-		
	07	procEmi	-		
	08	indSegmento	-		
	09	verProc	-		
	10	ideEmpregador	-		
	11	tpInscricao	-		
	12	nrInscricao	-		
	13	infoProcesso	-		
	14	Inclusão / alteração	-		
	15	ideProcesso	-		
	16	tpProcesso	MODEL_C93	C93_TPPROC	
	17	nrProcesso	MODEL_C93	C93_NRPROC	
	18	iniValidade	MODEL_C93	C93_DTINI	
	19	fimValidade	MODEL_C93	C93_DTFIN	
	20	dadosProcesso	-		
	21	indDecisao	MODEL_C93	C93_INDDEC	indDecisao
	22	dtDecisao	MODEL_C93	C93_DTDEC	
	23	indDeposito	MODEL_C93	C93_INDDEP	
	24	dadosProcJud	-		
	25	ufVara	MODEL_C93	C93_IDVARA	idVara
	26	codMunicipio	MODEL_C93	C93_IDVARA	idVara
	27	idVara	MODEL_C93	C93_IDVARA	idVara
28	indAutoria	MODEL_C93	C93_INDAUT		

### Exemplo:

```

Begin Transaction

  Aadd( aChave, { "C", "C93_TPPROC", "1", .T. } )
  Aadd( aChave, { "C", "C93_NRPROC", "15000259872013", .T. } )
  Aadd( aChave, { "C", "C93_DTINI", "072013", .T. } )
  Aadd( aChave, { "C", "C93_DTFIN", "102013", .T. } )

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÙ
    
```



```
If FTafVldOpe( 'C93', 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, 'TAF245', cCmpsNoUpd, 1 )  
  
//ÚÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁ;  
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³  
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³  
//ÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÛ  
  
If nOpc <> 5  
    oModel:LoadValue( "MODEL C93 ", "C93 FILIAL", C93->C93 FILIAL)  
    oModel:LoadValue( "MODEL C93 ", "C93 INDEP", "S" )  
  
EndIf  
  
//ÚÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁ;  
//³Efetiva a operação desejada³  
//ÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÛ  
FWFormCommit( oModel )  
  
    oModel:DeActivate()  
  
EndIf  
End Transaction
```





EndIf

```
//ÚÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀ;  
//³Efetiva a operação desejada³  
//ÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÁÀÙ
```

```
FWFormCommit( oModel )
```

```
oModel:DeActivate()
```

EndIf

End Transaction





## S-1100 – eSocial Mensal – Abertura

<b>Rotina:</b>	TAF249				
<b>Model MVC:</b>	MODEL_C8Y				
<b>Validação</b>	TAF249VLD()				
<b>Geração XML</b>	TAF249XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpAbertura	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao			
	07	nrRecibo			
	08	indApuracao	MODEL_C8Y	C8Y_INDAPU	
	09	perApuracao	MODEL_C8Y	C8Y_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	ideRespInformacao	-		
	18	nomeResponsavel			
	19	cpfResponsavel	MODEL_C8Y	C8Y_RESPON	cpfResponsavel
	20	Telefone	-	-	
	21	Fax	-	-	
	22	Email	-	-	
	23	infoSubstPatronal	-		
	24	indSubstPatronal	MODEL_C8Y	C8Y_INDSPT	
	25	percAliqPatronal	MODEL_C8Y	C8Y_ALQPTN	
	26	recExpServicos	-		
	27	recBrutaExp12m	MODEL_C8Y	C8Y_RBRVEN	
	28	recBrutaTot12m	MODEL_C8Y	C8Y_RBRTOT	
	29	vlrTribRecBruta	MODEL_C8Y	C8Y_VLIMPC	
	30	qtdMesesAtiv	MODEL_C8Y	C8Y_QTDMES	
	31	percReducaoLei11774	MODEL_C8Y	C8Y_PERCRZ	
	32	recAtivConcomitante	-		
	33	recBrutaTotMes	MODEL_C8Y	C8Y_RBTMES	
	34	recBrutaTotAno	MODEL_C8Y	C8Y_RBTANO	
	35	recBrutaAtssMes	MODEL_C8Y	C8Y_RBTATM	
	36	recBrutaAtssAno	MODEL_C8Y	C8Y_RBTATA	
	37	fatorMes	MODEL_C8Y	C8Y_FATMES	
38	Fator13	MODEL_C8Y	C8Y_FATDTS		



### Exemplo:

```
Aadd( aChave, { "C", "C8Y_INDAPU", "1", .T. } )  
Aadd( aChave, { "C", "C8Y_PERAPU", "102013", .T. } )
```

### Begin Transaction

```
If FTafVldOpe( "C8Y", 2, @nOpc, cFilev, @aIncons, aChave, @oModel, "Tafa249", cCmpsNoUpd )  
/*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;:  
 ³Carrega array com os campos De/Para de gravação das informações ³  
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU*/  
aRules := TAF249Rul(cCabec)  
oModel:LoadValue( "MODEL_C8Y", "C8Y FILIAL", C8Y->C8Y FILIAL )  
oModel:LoadValue( "MODEL_C8Y", "C8Y_PERAPU", "102013")  
/*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;:  
 ³Efetiva a operação desejada³  
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU*/  
FWFormCommit( oModel )  
lRet := .T.  
oModel:DeActivate()  
  
EndIf
```

### End Transaction



## S-1200 – eSocial Mensal – Remuneração do Trabalhador

<b>Rotina:</b>	TAF250				
<b>Model MVC:</b>	MODEL_C91				
<b>Validação</b>	TAF250VLD()				
<b>Geração XML</b>	TAF250XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpRemuneracao	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao			
	07	nrRecibo			
	08	indApuracao	MODEL_C91	C91_INDAPU	
	09	perApuracao	MODEL_C91	C91_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tplInscricao	-		
	16	nrInscricao	-		
	17	ideTrabalhador	-		
	18	cpfTrab	MODEL_C91	C91_TRABAL	cpfTrab
	19	nisTrab	-		
	20	qtdDepSF			
	21	qtdDepIRRF			
	22	infoMultiplosVinculos	-		
	23	indMV	MODEL_C91	C91_INDMVI	
	24	remOutrasEmpresas			
	25	tplInscricao			
	26	nrInscricao			
	27	vlrRemuneracao			
	28	infoSimplesAtivConcomitante			
	29	indSimples	MODEL_C91	C91_INDSIM	
	30	infoComplementares			
	31	nomeTrab			
	32	dtNascto			
	33	codCBO			
	34	infoTrabAvulso	-		
	35	cnpjSindicato	MODEL_C91	C91_CNPJSI	
	36	fPasSindicato	MODEL_C91	C91_FPASSI	"fpas","codTerceiros"
	37	codTerceiros	-	-	
	38	procJudTrabalhador	-	-	
	39	tpTributo	MODEL_CRN	CRN_TPTRIB	
	40	nrProcJud	MODEL_CRN	CRN_IDPROC	
41	infoPeriodoApuracao	-			



42	ideEstabLotacao	MODEL_C9K	C9K_ESTABE	
43	tplInscricao	MODEL_C9K	C9K_LOTACA	
44	nrInscricao			
45	codLotacal			
46	remunPeriodoApuracao			
47	Matricula	MODEL_C9L	C9L_MATRIC	
48	codCateg	MODEL_C9L	C9L_CODCAT	codCateg
49	bcCP	MODEL_C9L	C9L_BCPREV	
50	bciRRF	MODEL_C9L	C9L_BCIRRF	
51	bcFGTS	MODEL_C9L	C9L_BCFGTS	
52	descCP	MODEL_C9L	C9L_VDESPR	
53	vlrProventos	MODEL_C9L	C9L_PROVEN	
54	vlrDescontos	MODEL_C9L	C9L_DESCON	
55	vlrLiquido	MODEL_C9L	C9L_VLRLIQ	
56	itensRemun			
57	codRubrica	MODEL_C9M	C9M_CODRUB	codRubrica
58	qtdRubrica	MODEL_C9M	C9M_QTDRUB	
59	vlrRubrica	MODEL_C9M	C9M_VLRRUB	
60	infoAgenteNocivo			
61	grauExp	MODEL_C9L	C9L_GRAEXP	
62	infoRescisao			
63	nrReciboDeslig	MODEL_C9L	C9L_NRECDE	
64	infoPeriodoAnterior			
65	ideAcordoDissidio			
66	dtAcordo	MODEL_C9N	C9N_DTACOR	
67	tpAcordo	MODEL_C9N	C9N_TPACOR	
68	idePeriodo			
69	perReferencia	MODEL_C9O	C9O_PERREF	
70	ideEstabLotacao			
71	tplInscricao	MODEL_C9P	C9P_ESTABE	tplInscricao,nrInscricao
72	nrInscricao	-	-	
73	codLotacao	MODEL_C9P	C9P_LOTACA	codLotacao
74	remunPeriodoAnterior			
75	Matricula	MODEL_C9Q	C9Q_MATRIC	
76	codCateg	MODEL_C9Q	C9Q_CODCAT	codCateg
77	bcCP	MODEL_C9Q	C9Q_BCPREV	
78	bciRRF	MODEL_C9Q	C9Q_BCIRRF	
79	bcFGTS	MODEL_C9Q	C9Q_BCFGTS	
80	descCP	MODEL_C9Q	C9Q_VDESPR	
81	vlrProventos	MODEL_C9Q	C9Q_PROVEN	
82	vlrDescontos	MODEL_C9Q	C9Q_DESCON	
83	vlrLiquido	MODEL_C9Q	C9Q_VLRLIQ	
84	itensRemun			
85	codRubrica	MODEL_C9R	C9R_CODRUB	codRubrica
86	qtdRubrica	MODEL_C9R	C9R_QTDRUB	
87	vlrRubrica	MODEL_C9R	C9R_VLRRUB	



```

oModel:LoadValue( "MODEL C9K", "C9K LOTACA", FGetIdInt( "codLotacao", "", "01")
)
    nC9L:= 1
    cC9LPath := cC9KPath+"/remunPeriodoApuracao[" + CVALTOCHAR(nC9L) + "]"

    If nOpc == 4 .And. oDados.XPathHasNode( cC9LPath )
        For nJ := 1 to oModel:GetModel( 'MODEL_C9L' ):Length()
            oModel:GetModel( 'MODEL_C9L' ):GoLine(nJ)
            oModel:GetModel( 'MODEL_C9L' ):DeleteLine()
        Next nJ
    EndIf

    nC9L := 1
    While oDados.XPathHasNode( cC9LPath)

        oModel:GetModel( 'MODEL_C9L' ):LVALID := .T.

        If nOpc == 4 .Or. nC9L > 1
            oModel:GetModel( 'MODEL_C9L' ):AddLine()
        EndIf

        oModel:LoadValue( "MODEL_C9L", "C9L_VLRLIQ", "10")
        oModel:LoadValue( "MODEL_C9L", "C9L_NRECDE", "0123456789")
        nC9M:= 1
        cC9MPath := cC9LPath+"/itensRemun[" + CVALTOCHAR(nC9M) + "]"

        If nOpc == 4 .And. oDados.XPathHasNode( cC9MPath )
            For nJ := 1 to oModel:GetModel( 'MODEL_C9M' ):Length()
                oModel:GetModel( 'MODEL_C9M' ):GoLine(nJ)
                oModel:GetModel( 'MODEL_C9M' ):DeleteLine()
            Next nJ
        EndIf

        nC9M := 1
        While oDados.XPathHasNode( cC9MPath)
            oModel:GetModel( 'MODEL_C9M' ):LVALID := .T.

            If nOpc == 4 .Or. nC9M > 1
                oModel:GetModel( 'MODEL_C9M' ):AddLine()
            EndIf

            oModel:LoadValue( "MODEL_C9M", "C9M_QTDRUB" , "2")
            nC9M++
            cC9MPath := cC9LPath+"/itensRemun[" + CVALTOCHAR(nC9M) + "]"
        EndDo
        nC9L++
        cC9LPath := cC9KPath+"/remunPeriodoApuracao[" + CVALTOCHAR(nC9L) + "]"
    EndDo
    nC9K++
    cC9KPath := "/evtFpRemuneracao/infEvento/infoPeriodoApuracao/ideEstabLotacao["
+ CVALTOCHAR(nC9K) + "]"
    EndDo
    nC9N := 1
    cC9NPath := "/evtFpRemuneracao/infEvento/infoPeriodoAnterior/ideAcordoDissidio[" +
CVALTOCHAR(nC9N) + "]"

    If nOpc == 4 .And. oDados.XPathHasNode( cC9NPath )
        For nJ := 1 to oModel:GetModel( 'MODEL_C9N' ):Length()
            oModel:GetModel( 'MODEL_C9N' ):GoLine(nJ)
            oModel:GetModel( 'MODEL_C9N' ):DeleteLine()
        Next nJ
    EndIf

    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
    //³Roda o XML parseado para gravar as novas informações no GRID³
    //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÙ
    nC9N := 1
    While oDados.XPathHasNode( cC9NPath)
        oModel:GetModel( 'MODEL_C9N' ):LVALID := .T.

        If nOpc == 4 .Or. nC9N > 1

```



```

oModel:GetModel( 'MODEL_C9N' ):AddLine()

EndIf

oModel:LoadValue( "MODEL_C9N", "C9N_TPACOR" , "1")
nC90:= 1
cC90Path := cC9NPath + "/idePeriodo[" + CVALTOCHAR(nC90) + "]"

If nOpc == 4 .And. oDados.XPathHasNode( cC90Path )
  For nJ := 1 to oModel:GetModel( 'MODEL_C90' ):Length()
    oModel:GetModel( 'MODEL_C90' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_C90' ):DeleteLine()
  Next nJ
EndIf

nC90 := 1
While oDados.XPathHasNode(cC90Path)

  oModel:GetModel( 'MODEL_C90' ):LVALID      := .T.

  If nOpc == 4 .Or. nC90 > 1
    oModel:GetModel( 'MODEL_C90' ):AddLine()
  EndIf

  oModel:LoadValue( "MODEL_C90", "C90_PERREF", "102013")
  nC9P:= 1
  cC9PPath := cC90Path + "/ideEstabLotacao[" + CVALTOCHAR(nC9P) + "]"

  If nOpc == 4 .And. oDados.XPathHasNode( cC9PPath )
    For nJ := 1 to oModel:GetModel( 'MODEL_C9P' ):Length()
      oModel:GetModel( 'MODEL_C9P' ):GoLine(nJ)
      oModel:GetModel( 'MODEL_C9P' ):DeleteLine()
    Next nJ
  EndIf

  nC9P := 1
  While oDados.XPathHasNode(cC9PPath)
    oModel:GetModel( 'MODEL_C9P' ):LVALID      := .T.

    If nOpc == 4 .Or. nC9P > 1
      oModel:GetModel( 'MODEL_C9P' ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL_C9P", "C9P_ESTABE", FGetIdInt(
"tpInscricao", "nrInscricao", "1" , "0235789" ) )
    nC9Q:= 1
    cC9QPath := cC9PPath+ "/remunPeriodoAnterior[" +
CVALTOCHAR(nC9Q) + "]"

    If nOpc == 4 .And. oDados.XPathHasNode( cC9QPath )
      For nJ := 1 to oModel:GetModel( 'MODEL_C9Q' ):Length()
        oModel:GetModel( 'MODEL_C9Q' ):GoLine(nJ)
        oModel:GetModel( 'MODEL_C9Q' ):DeleteLine()
      Next nJ
    EndIf

    nC9Q := 1
    While oDados.XPathHasNode(cC9QPath)
      oModel:GetModel( 'MODEL_C9Q' ):LVALID      := .T.

      If nOpc == 4 .Or. nC9Q > 1
        oModel:GetModel( 'MODEL_C9Q' ):AddLine()
      EndIf

      oModel:LoadValue( "MODEL_C9Q", "C9Q_DESCON" , "1")
      nC9R:= 1
      cC9RPath := cC9QPath + "/itensRemun[" + CVALTOCHAR(nC9R)
+ "]"

      If nOpc == 4 .And. oDados.XPathHasNode( cC9QPath )
        For nJ := 1 to oModel:GetModel( 'MODEL_C9R'
):Length()

```



```

) :DeleteLine ()
oModel:GetModel ( 'MODEL C9R' ) :GoLine (nJ)
oModel:GetModel ( 'MODEL C9R'

Next nJ
EndIf

nC9R := 1

While oDados:XPathHasNode (cC9RPath)
oModel:GetModel ( 'MODEL_C9R' ) :LVALID :=

If nOpc == 4 .Or. nC9R > 1
oModel:GetModel ( 'MODEL C9R' ) :AddLine ()
EndIf
oModel:LoadValue ( "MODEL C9R", "C9R QTDRUB", "5")

nC9R++
cC9RPath := cC9QPath + "/itensRemun[" +

EndDo
nC9Q++
cC9QPath := cC9PPath + "/remunPeriodoAnterior[" +

EndDo
nC9P++
cC9PPath := cC9OPath + "/itensRemun[" + CVALTOCHAR (nC9P) + "]"

EndDo
nC9O++
cC9OPath := cC9NPath + "/remunPeriodoApuracao[" + CVALTOCHAR (nC9O) + "]"

EndDo
nC9N++
cC9NPath := "/evtFpRemuneracao/infEvento/infoPeriodoApuracao/ideEstabLotacao["

+ CVALTOCHAR (nC9N) + "]"
EndDo
nC9S := 1
cC9SPath := "/evtFpRemuneracao/infEvento/infoPagtosEfetuados[" + CVALTOCHAR (nC9S) + "]"

If nOpc == 4 .And. oDados:XPathHasNode ( cC9SPath )
For nJ := 1 to oModel:GetModel ( 'MODEL_C9S' ) :Length ()
oModel:GetModel ( 'MODEL C9S' ) :GoLine (nJ)
oModel:GetModel ( 'MODEL C9S' ) :DeleteLine ()
Next nJ
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
nC9U := 1
cC9UPath := "/evtFpRemuneracao/infEvento/totRemuneracao[" + CVALTOCHAR (nC9U) + "]"

If nOpc == 4 .And. oDados:XPathHasNode ( cC9UPath )
For nJ := 1 to oModel:GetModel ( 'MODEL_C9U' ) :Length ()
oModel:GetModel ( 'MODEL C9U' ) :GoLine (nJ)
oModel:GetModel ( 'MODEL_C9U' ) :DeleteLine ()
Next nJ
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Roda o XML parseado para gravar as novas informações no GRID³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
nC9U := 1
While oDados:XPathHasNode (cC9UPath)

oModel:GetModel ( 'MODEL C9U' ) :LVALID := .T.

If nOpc == 4 .Or. nC9U > 1
oModel:GetModel ( 'MODEL_C9U' ) :AddLine ()
EndIf
oModel:LoadValue ( "MODEL C9U", "C9U VDESPR", "1")

```





```

nC9U++
cC9UPath := "/evtFpRemuneracao/infEvento/totRemuneracao[" + CVALTOCHAR(nC9U) +
"]"

EndDo

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
nC9J := 1
cC9JPath :=
"evtFpRemuneracao/infEvento/ideTrabalhador/infoMultiplosVinculos/remunOutrasEmpresas[" + CVALTOCHAR(nC9J) +
"]"

If nOpc == 4 .And. oDados.XPathHasNode( cC9JPath )
For nJ := 1 to oModel:GetModel( 'MODEL_C9J' ):Length()
oModel:GetModel( 'MODEL_C9J' ):GoLine(nJ)
oModel:GetModel( 'MODEL_C9J' ):DeleteLine()
Next nJ
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Roda o XML parseado para gravar as novas informações no GRID³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
nC9J := 1
While oDados.XPathHasNode(cC9JPath)

oModel:GetModel( 'MODEL_C9J' ):LVALID := .T.

If nOpc == 4 .Or. nC9J > 1
oModel:GetModel( 'MODEL_C9J' ):AddLine()
EndIf
oModel:LoadValue( "MODEL_C9J", "C9J_VLREMU", "500")

nC9J++
cC9JPath :=
"evtFpRemuneracao/infEvento/ideTrabalhador/infoMultiplosVinculos/remunOutrasEmpresas[" + CVALTOCHAR(nC9J) +
"]"

EndDo

EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction

```



## S-1300 – Eventos Periódicos – Pagamentos Diversos

<b>Rotina:</b>	TAF285				
<b>Model MVC:</b>	MODEL_CRW; MODEL_CRX; MODEL_CRY; MODEL_CU2; MODEL_CRZ; MODEL_CU3; MODEL_CU4; MODEL_CU5; MODEL_CU6; MODEL_CU7; MODEL_CU8; MODEL_CU9;				
<b>Validação</b>	TAF285VLD()				
<b>Geração XML</b>	TAF285XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtPagtosDiversos	-		
	03	Id	-		
	04	versao	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	IndApuracao	MODEL_CRW	CRW_INDAPU	
	09	perApuracao	MODEL_CRW	CRW_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	ideBeneficiario	-		
	18	codRendimento	MODEL_CRW	CRW_IDREND	codRendimento
	19	tpInscBeneficiario	MODEL_CRW	CRW_TPINSC	
	20	nrInscBeneficiario	MODEL_CRW	CRW_NRINSC	
	21	dadosNaoResidente	-		
	22	dadosPessoais	-		
	23	nomeBeneficiario	MODEL_CRW	CRW_NOMEBE	
	24	dadosEndereco	-		
	25	paisResidencia	MODEL_CRW	CRW_IDPAIS	paisResidencia
	26	descLogradouro	MODEL_CRW	CRW_LOGRAD	
	27	nrLogradouro	MODEL_CRW	CRW_NUMERO	
	28	Complemento	MODEL_CRW	CRW_COMPLE	
	29	Bairro	MODEL_CRW	CRW_BAIRRO	
	30	nomeCidade	MODEL_CRW	CRW_CODMU N	
	31	codPostal	MODEL_CRW	CRW_CEP	
	32	dadosFiscais	-		
	33	indNIF	MODEL_CRW	CRW_TPNIF	
	34	nifBeneficiario	MODEL_CRW	CRW_NIF	
	35	relFontePagadora	MODEL_CRW	CRW_RELFON	relFontePagadora
	36	dadosMolestiaGrave	-		
	37	dtLaudo	MODEL_CRW	CRW_DTLAUD	
	38	infoPagto	-		
	39	pagtoResidente	-		
	40	pagtoBPF	-		
	41	dtPagto	MODEL_CRX	CRX_DATA	
	42	indSuspExigibilidade	MODEL_CRX	CRX_EXISUS	
	43	indDecTerceiro	MODEL_CRX	CRX_DECTER	
44	vlrRendTributavel	MODEL_CRX	CRX_VLRTR		

# Integração TAF



45	vlrIRRF	MODEL_CRX	CRX_VLRIRR	
46	detDeducao	-		
47	indTpDeducao	MODEL_CRY	CRY_TPDEDU	
48	vlrDeducao	MODEL_CRY	CRY_VLRDED	
49	rendIsento	-		
50	tplsencao	MODEL_CU2	CU2_TPISEN	tplsencao
51	vlrIsento	MODEL_CU2	CU2_VLRISE	
52	detOutros	-		
53	descRendimento	MODEL_CU2	CU2_REFER	
54	detCompetencia	-		
55	indPerReferencia	MODEL_CRZ	CRZ_PERREF	
56	perRefPagto	MODEL_CRZ	CRZ_REFER	
57	vlrRendTributavel	MODEL_CRZ	CRZ_RENTRI	
58	compJudicial	-		
59	vlrCompAnoCalendario	MODEL_CRX	CRX_VLRCAC	
60	vlrcompAnosAnteriores	MODEL_CRX	CRX_VLRCCA	
61	depJudicial	-		
62	vlrDepJudicial	MODEL_CRX	CRX_VLRDJU	
63	infoRRA	-		
64	tpProcRRA	MODEL_CU3	CU3_IDPROC	tpProcRRA, nrProcRRA
65	nrProcRRA	-	-	-
66	natRRA	MODEL_CU3	CU3_NATRRA	
67	qtdMesesRRA	MODEL_CU3	CU3_QTDMES	
68	despProcJudicial	-		
69	vlrDespCustas	MODEL_CU3	CU3_VLRCUS	
70	vlrDespAdvogados	MODEL_CU3	CU3_VLRADV	
71	ideAdvogado	-		
72	tpInscAdvogado	MODEL_CU4	CU4_IDADV	tpInscAdvogado, nrInscAdvogado
73	nrInscAdvogado	-	-	-
74	vlrAdvogado	MODEL_CU4	CU4_VLRADV	
75	infoProcJudicial	-		
76	nrProcJud	MODEL_CU5	CU5_IDPROC	"" , "nrProcJud"
77	indOrigemRecursos	MODEL_CU5	CU5_ORIREC	
78	despProcJudicial	-		
79	vlrDespCustas	MODEL_CU5	CU5_VLRCUS	
80	vlrDespAdvogados	MODEL_CU5	CU5_VLRADV	
81	ideAdvogado	-		
82	tpInscAdvogado	MODEL_CU6	CU6_IDADV	"tpInscAdvogado", "nrInscAdvogado"
83	nrInscAdvogado	-	-	-
84	vlrAdvogado	MODEL_CU6	CU6_VLRADV	
85	origemRecursos	-		
86	cnpjOrigemRecursos	MODEL_CU5	CU5_CNPJOR	
87	pagtoBPJ	-		
88	dtPagto	MODEL_CU7	CU7_DTPGTO	
89	vlrRendTributavel	MODEL_CU7	CU7_VLRPGT	
90	vlrIRRF	MODEL_CU7	CU7_VLRIRR	
91	pagtoNaoResidente	-		
92	dtPagto	MODEL_CU8	CU8_DTPGTO	
93	tpRendimento	MODEL_CU8	CU8_IDTPRE	tpRendimento
94	formaTributacao	MODEL_CU8	CU8_IDFTRI	formaTributacao
95	vlrPago	MODEL_CU8	CU8_VLRPGT	
96	vlrIRRF	MODEL_CU8	CU8_VLRIRR	
97	infoPlanoSaudeColetivo	-		
98	cnpjOperadora	MODEL_CRW	CRW_CNPJPS	
99	vlrPagoTitular	MODEL_CRW	CRW_VLRTIT	
100	depPlano	-		





```

//³INICIO CRY- <detDeducao>      ³
nCRY := 1
cCRYPPath := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detDeducao[" + cValToChar(nCRY) + "]"

//Deleta as linhas existentes se for alteraca
If nOpc == 4 .And. oDados:XPathHasNode( cCRYPPath )
  For nCRY := 1 to oModel:GetModel( 'MODEL_CRY' ):Length()
    oModel:GetModel( 'MODEL_CRY' ):GoLine(nCRY)
    oModel:GetModel( 'MODEL_CRY' ):DeleteLine()
  Next nCRY
EndIf

//Alimenta o modelo CRY
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detDeducao[" + cValToChar(nCRY) + "]" )
  cCRYPPath := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detDeducao[" + cValToChar(nCRY) + "]"
  If nOpc == 4 .Or. nCRY > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CRY" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CRY" ):AddLine()
  EndIf

  oModel:LoadValue( "MODEL_CRY", "CRY_VLRDED", FTafGetVal( cCRYPPath + "/vlrDeducao", "N", .F., @alIncons, .F. ) )
  nCRY ++

EndDo

//³INICIO CRZ - <rendIsento>      ³
nCRZ := 1
cCRZPath := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detCompetencia[" + cValToChar(nCRZ) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCRZPath )
  For nCRZ := 1 to oModel:GetModel( 'MODEL_CRZ' ):Length()
    oModel:GetModel( 'MODEL_CRZ' ):GoLine(nCRZ)
    oModel:GetModel( 'MODEL_CRZ' ):DeleteLine()
  Next nCRZ
EndIf

//Alimenta o modelo CRZ
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detCompetencia[" + cValToChar(nCRZ) + "]" )
  cCRZPath := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/detCompetencia[" + cValToChar(nCRZ) + "]"
  If nOpc == 4 .Or. nCRZ > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CRZ" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CRZ" ):AddLine()
  EndIf

  oModel:LoadValue( "MODEL_CRZ", "CRZ_REFER", FTafGetVal( cCRZPath + "/perRefPagto", "C", .F., @alIncons, .F. ) )
  nCRZ ++

EndDo

//³INICIO CU2- <rendIsento>³
nCU2 := 1
cCU2Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/rendIsento[" + cValToChar(nCU2) + "]"

//Deleta as linhas existentes se for alteraca
If nOpc == 4 .And. oDados:XPathHasNode( cCU2Path )
  For nCU2 := 1 to oModel:GetModel( 'MODEL_CU2' ):Length()
    oModel:GetModel( 'MODEL_CU2' ):GoLine(nCU2)
    oModel:GetModel( 'MODEL_CU2' ):DeleteLine()
  Next nCU2

```

```

Endif

//Alimenta o modelo CU2
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/rendIsento[" + cValToChar(nCU2) + "]" )
  cCU2Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/rendIsento[" + cValToChar(nCU2) + "]"
  If nOpc == 4 .Or. nCU2 > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CU2" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CU2" ):AddLine()
  Endif

  oModel:LoadValue( "MODEL_CU2", "CU2_VLRRISE", FTafGetVal( cCU2Path + "/vrlriSento", "N", .F., @alncons, .F. ) )
  nCU2 ++

EndDo

//³INICIO CU3 - <infoRRA>³
nCU3 := 1
cCU3Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoRRA[" + cValToChar(nCU3) + "]"

//Deleta as linhas existentes se for alteracao
if nOpc == 4 .And. oDados:XPathHasNode( cCU3Path )
  For nCU3 := 1 to oModel:GetModel( 'MODEL_CU3' ):Length()
    oModel:GetModel( 'MODEL_CU3' ):GoLine(nCU3)
    oModel:GetModel( 'MODEL_CU3' ):DeleteLine()
  Next nCU3
Endif

//Alimenta o modelo CU3
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoRRA[" + cValToChar(nCU3) + "]" )
  cCU3Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoRRA[" + cValToChar(nCU3) + "]"
  If nOpc == 4 .Or. nCU3 > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CU3" ):LVALID := .T.
    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CU3" ):AddLine()
  Endif

  nGetId := FGetIdInt("tpProcRRA","nrProcRRA", cCU3Path + "/tpProcRRA", cCU3Path + "/nrProcRRA",.T.)

  oModel:LoadValue( "MODEL_CU3", "CU3_IDPROC", nGetId)
  oModel:LoadValue( "MODEL_CU3", "CU3_NATRRRA", FTafGetVal( cCU3Path + "/natRRA", "C", .F., @alncons, .F. ) )

  //³INICIO CU4 - <ideAdvogado> ³
  nCU4 := 1
  cCU4Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoRRA/despProJudicial/ideAdvogado[" +
cValToChar(nCU4) + "]"

  //Deleta as linhas existentes se for alteracao
  If nOpc == 4 .And. oDados:XPathHasNode( cCU4Path )
    For nCU4 := 1 to oModel:GetModel( 'MODEL_CU4' ):Length()
      oModel:GetModel( 'MODEL_CU4' ):GoLine(nJ)
      oModel:GetModel( 'MODEL_CU4' ):DeleteLine()
    Next nCU4
  Endif

  //Alimenta o modelo CU4
  While oDados:XPathHasNode( cCabec +
"/infoPagto/pagtoResidente/pagtoBPF/infoRRA/despProJudicial/ideAdvogado[" + cValToChar(nCU4) + "]" )
    cCU4Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoRRA/despProJudicial/ideAdvogado[" +
cValToChar(nCU4) + "]"

    If nOpc == 4 .Or. nCU4 > 1
      // Informa que a linha está válida para que seja incluída uma nova linha
      oModel:GetModel( "MODEL_CU4" ):LVALID := .T.
    Endif
  Endwhile

```



```

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CU4" ):AddLine()

Endif
nGetId := FGetIdInt("tplnscAdvogado", "nrInscAdvogado", cCU4Path + "/tplnscAdvogado", cCU4Path +
"/nrInscAdvogado", .T.)

oModel:LoadValue( "MODEL_CU4", "CU4_IDADV", nGetId)
oModel:LoadValue( "MODEL_CU4", "CU4_VLRADV", FTafGetVal( cCU4Path + "/vlrAdvogado", "N", .F.,
@alIncons, .F. ))

nCU4 ++

EndDo

nCU3 ++

EndDo

//³INICIO CU5 - <infoProJudicial> ³
nCU5 := 1
cCU5Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial[" + cValToChar(nCU5) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCU5Path )
For nCU5 := 1 to oModel:GetModel( 'MODEL_CU5' ):Length()
oModel:GetModel( 'MODEL_CU5' ):GoLine(nCU5)
oModel:GetModel( 'MODEL_CU5' ):DeleteLine()
Next nCU5
Endif

//Alimenta o modelo CU5
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial[" + cValToChar(nCU5) + "]" )
cCU5Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial[" + cValToChar(nCU5) + "]"
If nOpc == 4 .Or. nCU5 > 1
// Informa que a linha está válida para que seja incluída uma nova linha
oModel:GetModel( "MODEL_CU5" ):LVALID := .T.
// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CU5" ):AddLine()
Endif

nGetId := FTafGetVal(FGetIdInt( "", "nrProJud", cCU5Path + "/nrProJud" ), "C", .T., @alIncons, .F. )
oModel:LoadValue( "MODEL_CU5", "CU5_IDPROC", nGetId )

//³INICIO CU6 - <ideAdvogado> ³
nCU6 := 1
cCU6Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial/despProJudicial/ideAdvogado[" +
cValToChar(nCU6) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCU6Path )
For nCU6 := 1 to oModel:GetModel( 'MODEL_CU6' ):Length()
oModel:GetModel( 'MODEL_CU6' ):GoLine(nCU6)
oModel:GetModel( 'MODEL_CU6' ):DeleteLine()
Next nCU6
Endif

//Alimenta o modelo CU6
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial[" +
cValToChar(nCU5) + "]/despProJudicial/ideAdvogado[" + cValToChar(nCU6) + "]" )
cCU6Path := cCabec +
"/infoPagto/pagtoResidente/pagtoBPF/infoProJudicial/despProJudicial/ideAdvogado[" + cValToChar(nCU6) + "]"
If nOpc == 4 .Or. nCU6 > 1
// Informa que a linha está válida para que seja incluída uma nova linha
oModel:GetModel( "MODEL_CU6" ):LVALID := .T.

// Inclui uma linha a cada volta do laço

```

```

oModel:GetModel( "MODEL_CU6" ):AddLine()
EndIf
nGetId := FGetIdInt("tplnscAdvogado","nrInscAdvogado", cCU6Path + "/tplnscAdvogado", cCU6Path +
"/nrInscAdvogado",.T.)

oModel:LoadValue( "MODEL_CU6", "CU6_IDADV", nGetId)
nCU6 ++

EndDo
nCU5 ++

EndDo
nCRX ++

EndDo
//³INICIO CU7 - <pagtoBPJ>      ³
nCU7 := 1
cCU7Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPJ[" + cValToChar(nCU7) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCU7Path )
  For nCU7 := 1 to oModel:GetModel( 'MODEL_CU7' ):Length()
    oModel:GetModel( 'MODEL_CU7' ):GoLine(nCU7)
    oModel:GetModel( 'MODEL_CU7' ):DeleteLine()
  Next nCU7
EndIf

//Alimenta o modelo CU7
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoResidente/pagtoBPJ[" + cValToChar(nCU7) + "]" )
  cCU7Path := cCabec + "/infoPagto/pagtoResidente/pagtoBPJ[" + cValToChar(nCU7) + "]"
  If nOpc == 4 .Or. nCU7 > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CU7" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CU7" ):AddLine()
  EndIf

  oModel:LoadValue( "MODEL_CU7", "CU7_DTPGTO", FTafGetVal( cCU7Path + "/dtPagto", "D", .F., @alncons, .F. ) )
  oModel:LoadValue( "MODEL_CU7", "CU7_VLRPGT", FTafGetVal( cCU7Path + "/vlrRendTributavel", "N", .F., @alncons, .F. ) )
  oModel:LoadValue( "MODEL_CU7", "CU7_VLRIRR", FTafGetVal( cCU7Path + "/vlrIRR", "N", .F., @alncons, .F. ) )
  nCU7 ++

EndDo
//³INICIO CU8 - <pagtoNaoResidente> ³
nCU8 := 1
cCU8Path := cCabec + "/infoPagto/pagtoNaoResidente[" + cValToChar(nCU8) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCU8Path )
  For nCU8 := 1 to oModel:GetModel( 'MODEL_CU8' ):Length()
    oModel:GetModel( 'MODEL_CU8' ):GoLine(nCU8)
    oModel:GetModel( 'MODEL_CU8' ):DeleteLine()
  Next nCU8
EndIf

//Alimenta o modelo CU8
While oDados:XPathHasNode( cCabec + "/infoPagto/pagtoNaoResidente[" + cValToChar(nCU8) + "]" )
  cCU8Path := cCabec + "/infoPagto/pagtoNaoResidente[" + cValToChar(nCU8) + "]"
  If nOpc == 4 .Or. nCU8 > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CU8" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CU8" ):AddLine()
  EndIf

```





```

oModel:LoadValue( "MODEL_CU8", "CU8_DTPGTO", FTafGetVal( cCU8Path + "/dtPagto", "D", .F., @alncons, .F. ) )
oModel:LoadValue( "MODEL_CU8", "CU8_IDTPRE", FTafGetVal( FGetIdInt( "tpRendimento", "", cCU8Path +
"/tpRendimento"), "C", .T., @alncons, .F. ) )
oModel:LoadValue( "MODEL_CU8", "CU8_IDFTRI", FTafGetVal( FGetIdInt( "formaTributacao", "", cCU8Path +
"/formaTributacao"), "C", .T., @alncons, .F. ) )
oModel:LoadValue( "MODEL_CU8", "CU8_VLRPGT", FTafGetVal( cCU8Path + "/vlrPago", "N", .F., @alncons, .F. ) )
oModel:LoadValue( "MODEL_CU8", "CU8_VLRIRR", FTafGetVal( cCU8Path + "/vlrIRRF", "N", .F., @alncons, .F. ) )
nCU8 ++

EndDo

//³INICIO CU9 - <depPlano>      3
nCU9 := 1
cCU9Path := cCabec + "/infoPlanoSaudeColetivo/depPlano[" + cValToChar(nCU9) + "]"

//Deleta as linhas existentes se for alteracao
If nOpc == 4 .And. oDados:XPathHasNode( cCU9Path )
  For nCU9 := 1 to oModel:GetModel( 'MODEL_CU9' ):Length()
    oModel:GetModel( 'MODEL_CU9' ):GoLine(nCU9)
    oModel:GetModel( 'MODEL_CU9' ):DeleteLine()
  Next nCU9
EndIf

//Alimenta o modelo CU9
While oDados:XPathHasNode( cCabec + "/infoPlanoSaudeColetivo/depPlano[" + cValToChar(nCU9) + "]" )
  cCU9Path := cCabec + "/infoPlanoSaudeColetivo/depPlano[" + cValToChar(nCU9) + "]"
  If nOpc == 4 .Or. nCU9 > 1
    // Informa que a linha está válida para que seja incluída uma nova linha
    oModel:GetModel( "MODEL_CU9" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CU9" ):AddLine()
  EndIf
  oModel:LoadValue( "MODEL_CU9", "CU9_CPFDEP", FTafGetVal( cCU9Path + "/cpfDep", "C", .F., @alncons, .F. ) )
  oModel:LoadValue( "MODEL_CU9", "CU9_DTNASC", FTafGetVal( cCU9Path + "/dtNasctoDep", "D", .F., @alncons, .F. ) )
  oModel:LoadValue( "MODEL_CU9", "CU9_NOMDEP", FTafGetVal( cCU9Path + "/nomeDep", "C", .F., @alncons, .F. ) )
  oModel:LoadValue( "MODEL_CU9", "CU9_TPDEP", FGetIdInt( "relDependencia", "", cCU9Path + "/relDependencia"), "C", .T.,
@alncons, .F. )

  oModel:LoadValue( "MODEL_CU9", "CU9_VLRPGT", FTafGetVal( cCU9Path + "/vlrPagoDep", "N", .F., @alncons, .F. ) )
  nCU9 ++

EndDo

EndIf
FWFormCommit( oModel )
IRet := .T.

oModel:DeActivate()

EndIf

End Transaction

```



### S-1310 – eSocial Mensal – Outras Informações – Serv. Tomados (Cessão de Mão de Obra)

<b>Rotina:</b>	TAF271				
<b>Model MVC:</b>	MODEL_CMN; MODEL_CMO; MODEL_CMP; MODEL_CMQ;				
<b>Validação</b>	TAF271VLD()				
<b>Geração XML</b>	TAF271XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpServTomados	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CMN	CMN_INDAPU	
	09	perApuracao	MODEL_CMN	CMN_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoServTomados	-		
	18	ideEstabelecimento	-		
	19	tpInscricao	MODEL_CMO	CMO_TPINSC	
	20	nrInscricao	MODEL_CMO	CMO_INSCES	
	21	idePrestadorServicos	-		
	22	cnpjPrestador	MODEL_CMP	CMP_INSCCO	
	23	vlrBruto	MODEL_CMP	CMP_VLBRUT	
	24	vlrBaseRetencao	MODEL_CMP	CMP_BCRETE	
	25	vlrRetencao	MODEL_CMP	CMP_VLRETE	
	26	vlrAdicional	MODEL_CMP	CMP_VLADIC	
	27	vlrNaoRetido	MODEL_CMP	CMP_VLNRET	
	28	nfsTerceiros	-		
	29	Serie	MODEL_CMQ	CMQ_SERIE	
	30	numDocto	MODEL_CMQ	CMQ_NUMDOC	
	31	dtEmissaoNF	MODEL_CMQ	CMQ_DTEMIS	
	32	indObra	MODEL_CMQ	CMQ_INDOBR	
	33	nrCno	MODEL_CMQ	CMQ_NRCNO	
	34	vlrBaseRetencao	MODEL_CMQ	CMQ_BCRETE	
	35	vlrBruto	MODEL_CMQ	CMQ_VLBRUT	
	36	vlrRetencaoApurada	MODEL_CMQ	CMQ_VLAPUR	
	37	vlrRetencaoSubEmpreitada	MODEL_CMQ	CMQ_VLSUBE	
	38	vlrRetencao	MODEL_CMQ	CMQ_VLRETE	
	39	vlrAdicional	MODEL_CMQ	CMQ_VLADIC	
	40	itensNfs	-		
	41	nrItem	MODEL_CRO	CRO_NRITEM	
42	tpServico	MODEL_CRO	CRO_TPSERV		



```

If nOpc == 4 .And. oDados.XPathHasNode( cCROPath )
  For nJ := 1 to oModel:GetModel( 'MODEL_CRO' ):Length()
    oModel:GetModel( 'MODEL_CRO' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_CRO' ):DeleteLine()
  Next nJ
EndIf

While oDados.XPathHasNode( cCROPath )

  If nOpc == 4 .Or. nCRO > 1
    // Informa que a linha está para que seja incluída
    oModel:GetModel( "MODEL_CRO" ):LINVALID := .T.
    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CRO" ):AddLine()
  EndIf

  oModel:LoadValue( "MODEL_CRO", "CRO_NRITEM", FTafGetVal( cCROPath +
"/nrItem", "C", .F., @aIncons, .T. ) )
  oModel:LoadValue( "MODEL_CRO", "CRO_TPSESV", FTafGetVal(
FGetIdInt( "tpServico", , cCROPath + "/tpServico" ), "C", .T., @aIncons, .T. ) )
  oModel:LoadValue( "MODEL_CRO", "CRO_VLRBRU", FTafGetVal(
cCROPath + "/vlrBruto", "N", .F., @aIncons, .T. ) )
  oModel:LoadValue( "MODEL_CRO", "CRO_VBSRET", FTafGetVal(
cCROPath + "/vlrBaseRetencao", "N", .F., @aIncons, .T. ) )

  nCRO++
  cCROPath := cCMQPath + "/itensNfs[" + AllTrim(Str(nCRO))
+ "]"

  EndDo // Fim - CRO

  // CMP
  nCMP := 1
  cCMPPath := cCMOPath + "/idePrestadorServicos[" + AllTrim(Str(nCMP)) + "]"

  If nOpc == 4 .And. oDados.XPathHasNode( cCMPPath )
    For nJ := 1 to oModel:GetModel( 'MODEL_CMP' ):Length()
      oModel:GetModel( 'MODEL_CMP' ):GoLine(nJ)
      oModel:GetModel( 'MODEL_CMP' ):DeleteLine()
    Next nJ
  EndIf

  While oDados.XPathHasNode( cCMPPath )

    If nOpc == 4 .Or. nCMP > 1
      // Informa que a linha está para que seja incluída uma nova
      oModel:GetModel( "MODEL_CMP" ):LINVALID := .T.

      // Inclui uma linha a cada volta do laço
      oModel:GetModel( "MODEL_CMP" ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL_CMP", "CMP_VLADIC", "350")
    oModel:LoadValue( "MODEL_CMP", "CMP_VLNRET", "200")

    // CMQ
    nCMQ := 1
    cCMQPath := cCMPPath + "/nfsTerceiros[" + AllTrim(Str(nCMQ)) + "]"

    If nOpc == 4 .And. oDados.XPathHasNode( cCMQPath )
      For nJ := 1 to oModel:GetModel( 'MODEL_CMQ' ):Length()
        oModel:GetModel( 'MODEL_CMQ' ):GoLine(nJ)
        oModel:GetModel( 'MODEL_CMQ' ):DeleteLine()
      Next nJ
    EndIf

    While oDados.XPathHasNode( cCMQPath )

      If nOpc == 4 .Or. nCMQ > 1
        // Informa que a linha está para que seja incluída uma
        oModel:GetModel( "MODEL_CMQ" ):LINVALID := .T.

```

uma nova linha

linha

nova linha



```

        // Inclui uma linha a cada volta do laço
        oModel:GetModel( "MODEL_CMQ" ):AddLine()
    EndIf
    oModel:LoadValue( "MODEL_CMQ", "CMQ_VLBRUT", "350")
    oModel:LoadValue( "MODEL_CMQ", "CMQ_VLAPUR", "200")

    If oDados.XPathHasNode( cCMQPath + "/servPrestCondEspeciais" )
        oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV15", "100")
        oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV20", "100")
        oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV25", "100")
    EndIf

    If oDados.XPathHasNode( cCMQPath + "/infoProcJudicial" )
        oModel:LoadValue( "MODEL_CMQ", "CMQ_VLNRET", "100")
    EndIf

    If oDados.XPathHasNode( cCMQPath + "/proprietarioCNO" )
        oModel:LoadValue( "MODEL_CMQ", "CMQ_TPINSC", "1")
        oModel:LoadValue( "MODEL_CMQ", "CMQ_INSCPR",
"5681000114")
    EndIf

    nCMQ++
    cCMQPath := cCMPPath + "/nfsTerceiros[" + AllTrim(Str(nCMQ)) +
"]"

    EndDo // Fim - CMQ

    nCMP++
    cCMPPath := cCMOPath + "/idePrestadorServicos[" + AllTrim(Str(nCMP)) +
"]"

    EndDo // Fim - CMP

    nCMO++
    cCMOPath := cInfEvento + "/infoServTomados/ideEstabelecimento[" +
AllTrim(Str(nCMO)) + "]"
    EndDo // Fim - CMO
    EndIf

    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
    //³Efetiva a operacao desejada³
    //ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
    FWFormCommit( oModel )
    lRet := .T.

    oModel:DeActivate()

    EndIf
End Transaction

```



### S-1320 – eSocial Mensal – Outras Informações – Serv. Prestados (Cessão de Mão de Obra)

<b>Rotina:</b>	TAF288				
<b>Model MVC:</b>	MODEL_CMN; MODEL_CMO; MODEL_CMP; MODEL_CMQ;				
<b>Validação</b>	TAF288VLD()				
<b>Geração XML</b>	TAF288XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpServPrestados	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CMN	CMN_INDAPU	
	09	perApuracao	MODEL_CMN	CMN_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoServPrestados	-		
	18	ideEstabPrestador	-		
	19	tpInscricao	MODEL_CMO	CMO_TPINSC	
	20	nrInscricao	MODEL_CMO	CMO_INSCES	
	21	ideContratante	-		
	22	tpInscContratante	MODEL_CMP	CMP_TPINSC	
	23	nrInscContratante	MODEL_CMP	CMP_INSCCO	
	24	vlrBruto	MODEL_CMP	CMP_VLBRUT	
	25	vlrServicos	MODEL_CMP	CMP_VLSERV	
	26	vlrBaseRetencao	MODEL_CMP	CMP_BCRETE	
	27	vlrRetencao	MODEL_CMP	CMP_VLRETE	
	28	vlrAdicional	MODEL_CMP	CMP_VLADIC	
	29	vlrNaoRetido	MODEL_CMP	CMP_VLNRET	
	30	nfsEmitidas	-		
	31	Serie	MODEL_CMQ	CMQ_SERIE	
	32	numDocto	MODEL_CMQ	CMQ_NUMDOC	
	33	dtEmissaoNF	MODEL_CMQ	CMQ_DTEMIS	
	34	indObra	MODEL_CMQ	CMQ_INDOBR	
	35	nrCno	MODEL_CMQ	CMQ_NRCNO	
	36	vlrBruto	MODEL_CMQ	CMQ_VLBRUT	
	37	vlrServicos	MODEL_CMQ	CMQ_VLSERV	
	38	vlrDeducoes	MODEL_CMQ	CMQ_VLDEDU	
	39	vlrBaseRetencao	MODEL_CMQ	CMQ_BCRETE	
	40	vlrRetencaoApurada	MODEL_CMQ	CMQ_VLAPUR	
	41	vlrRetencaoSubEmpreitada	MODEL_CMQ	CMQ_VLSUBE	
42	vlrRetencao	MODEL_CMQ	CMQ_VLRETE		



```

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL CMO" ):AddLine()
EndIf

// Grava dados no model
oModel:LoadValue( "MODEL CMO", "CMO TPINSC", "1" )
oModel:LoadValue( "MODEL CMO", "CMO INSCES", "102505781460" )

// CMP
nCMP := 1
cCMPPath := cCMOPath + "/ideContratante[" + AllTrim(Str(nCMP)) + "]"

If nOpc == 4 .And. oDados.XPathHasNode( cCMPPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CMP' ):Length()
oModel:GetModel( 'MODEL_CMP' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CMP' ):DeleteLine()
Next nJ
EndIf

While oDados.XPathHasNode( cCMPPath )

If nOpc == 4 .Or. nCMP > 1
// Informa que a linha está para que seja incluída uma nova
oModel:GetModel( "MODEL_CMP" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CMP" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CMP", "CMP VLADIC", "300" )
oModel:LoadValue( "MODEL_CMP", "CMP_VLNRET", "500" )

// CMQ
nCMQ := 1
cCMQPath := cCMPPath + "/nfsEmitidas[" + AllTrim(Str(nCMQ)) + "]"

If nOpc == 4 .And. oDados.XPathHasNode( cCMQPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CMQ' ):Length()
oModel:GetModel( 'MODEL_CMQ' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CMQ' ):DeleteLine()
Next nJ
EndIf

While oDados.XPathHasNode( cCMQPath )

If nOpc == 4 .Or. nCMQ > 1
// Informa que a linha está para que seja incluída uma
oModel:GetModel( "MODEL_CMQ" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CMQ" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CMQ", "CMQ_VLRETE", "250" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLADIC", "400" )

If oDados.XPathHasNode( cCMQPath + "/servPrestCondEspeciais" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV15", "200" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV20", "200" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV25", "200" )
EndIf

If oDados.XPathHasNode( cCMQPath + "/infoProcJudicial" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLNRET", "250" )
EndIf

If oDados.XPathHasNode( cCMQPath + "/servSubempreitados" )
oModel:LoadValue( "MODEL_CMQ", "CMQ TPINSC", "1" )
oModel:LoadValue( "MODEL_CMQ", "CMQ INSCPR",
"5544778800112")

```





```

EndIf

If oDados.XPathHasNode( cCMQPath + "/proprietarioCNO" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_TPINSC", "1")
oModel:LoadValue( "MODEL_CMQ", "CMQ_INSCPR",
"87887782320")

EndIf

nCMQ++
cCMQPath := cCMPPath + "/nfsEmitidas[" + AllTrim(Str(nCMQ)) +
"]"

EndDo // Fim - CMQ

nCMP++
cCMPPath := cCMOPath + "/ideContratante[" + AllTrim(Str(nCMP)) + "]"
EndDo // Fim - CMP

nCMO++
cCMOPath := cInfEvento + "/infoServPrestados/ideEstabPrestador[" +
AllTrim(Str(nCMO)) + "]"
EndDo // Fim - CMO
EndIf

FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction

```

### S-1330 – eSocial Mensal – Outras Informações – Serv. Tomados de Cooperativa de Trabalho

<b>Rotina:</b>	TAF289				
<b>Model MVC:</b>	MODEL_CMN; MODEL_CMO; MODEL_CMQ				
<b>Validação</b>	TAF289VLD()				
<b>Geração XML</b>	TAF289XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpServTomadosCoop	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao			
	07	nrRecibo			
	08	indApuracao	MODEL_CMN	CMN_INDAPU	
	09	perApuracao	MODEL_CMN	CMN_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tplInscricao	-		
	16	nrInscricao	-		
	17	infoServTomados			
	18	ideEstabelecimento			
	19	tplInscricao	MODEL_CMO	CMO_TPINSC	
	20	nrInscricao	MODEL_CMO	CMO_INSCES	
	21	nfsTerceiros			
	22	cnpjCooperativa	MODEL_CMQ	CMQ_CNPJCO	
	23	Serie	MODEL_CMQ	CMQ_SERIE	
	24	numDocto	MODEL_CMQ	CMQ_NUMDOC	
	25	dtEmissaoNF	MODEL_CMQ	CMQ_DTEMIS	
	26	dtPagtoNF	MODEL_CMQ	CMQ_DTPGTO	
	27	indObra	MODEL_CMQ	CMQ_INDOBR	
	28	nrCno	MODEL_CMQ	CMQ_NRCNO	
	29	vlrBruto	MODEL_CMQ	CMQ_VLBRUT	
	30	vlrMatEquip	MODEL_CMQ	CMQ_VLMATE	
	31	vlrServicos	MODEL_CMQ	CMQ_VLSERV	
	32	vlrDeducoes	MODEL_CMQ	CMQ_VLDEDU	
	33	vlrBaseCoop	MODEL_CMQ	CMQ_BCCOOP	
	34	servCoopCondEspeciais	-		
	35	vlrServicos15	MODEL_CMQ	CMQ_VLSV15	
	36	vlrServicos20	MODEL_CMQ	CMQ_VLSV20	
	37	vlrServicos25	MODEL_CMQ	CMQ_VLSV25	
	38	servPrestAtivConcomitante			
	39	vlrBaseCoop	MODEL_CMQ	CMQ_BCCONC	
	40	vlrBaseCoop15	MODEL_CMQ	CMQ_BCCO15	
	41	vlrBaseCoop20	MODEL_CMQ	CMQ_BCCO20	
42	vlrBaseCoop25	MODEL_CMQ	CMQ_BCCO25		



43	proprietarioCNO	-		
44	tplnscProprietario	MODEL_CMQ	CMQ_TPINSC	
45	nrlnscProprietario	MODEL_CMQ	CMQ_INSCP	
46	totBaseCoop	-		
47	indIncidencia	MODEL_CMO	CMO_INDINC	
48	vlrBaseCoop	MODEL_CMO	CMO_BCTOT	
49	vlrBaseCoop15	MODEL_CMO	CMO_BCCP15	
50	vlrBaseCoop20	MODEL_CMO	CMO_BCCP20	
51	vlrBaseCoop25	MODEL_CMO	CMO_BCCP25	

## Exemplo:

```

cXml := xFunLerXML()

oDados := TXmlManager():New()
oDados.Parse(cXml)

Add( aChave, { "C", "CMN NOMEVE", "S1330", .T. } )
Add( aChave, { "C", "CMN INDAPU", "1", .T. } )
Add( aChave, { "C", "CMN PERAPU", "102014", .T. } )

Begin Transaction
  If FTafVldOpe( 'CMN', 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, 'TAF289', cCmpsNoUpd )
    If nOpc <> 5
      oModel.LoadValue( "MODEL CMN", "CMN FILIAL", CMN->CMN FILIAL )
      oModel.LoadValue( 'MODEL CMN', 'CMN NOMEVE', "S1330" )

      // CMO
      nCMO := 1
      cCMOPath := cInfEvento + "/infoServTomados/ideEstabelecimento[" + AllTrim(Str(nCMO)) +
"]"

      If nOpc == 4 .And. oDados.XPathHasNode( cCMOPath )
        For nJ := 1 to oModel.GetModel( 'MODEL_CMO' ):Length()
          oModel.GetModel( 'MODEL_CMO' ):GoLine(nJ)
          oModel.GetModel( 'MODEL_CMO' ):DeleteLine()
        Next nJ
      EndIf

      While oDados.XPathHasNode( cCMOPath )

        If nOpc == 4 .Or. nCMO > 1
          // Informa que a linha está válida para que seja incluída uma nova linha
          oModel.GetModel( "MODEL_CMO" ):LVALID := .T.
          // Inclui uma linha a cada volta do laço
          oModel.GetModel( "MODEL_CMO" ):AddLine()
        EndIf

        // Grava dados no model
        oModel.LoadValue( "MODEL_CMO", "CMO_INDINC", "1 " )

        // CMQ
        nCMQ := 1
        cCMQPath := cCMOPath + "/nfsTerceiros[" + AllTrim(Str(nCMQ)) + "]"

        If nOpc == 4 .And. oDados.XPathHasNode( cCMQPath )
          For nJ := 1 to oModel.GetModel( 'MODEL_CMQ' ):Length()
            oModel.GetModel( 'MODEL_CMQ' ):GoLine(nJ)
            oModel.GetModel( 'MODEL_CMQ' ):DeleteLine()
          Next nJ
        EndIf

        While oDados.XPathHasNode( cCMQPath )

```



```

linha
        If nOpc == 4 .Or. nCMQ > 1
            // Informa que a linha está para que seja incluída uma nova
            oModel:GetModel( "MODEL_CMQ" ):LVALID := .T.

            // Inclui uma linha a cada volta do laço
            oModel:GetModel( "MODEL_CMQ" ):AddLine()

            EndIf
            oModel:LoadValue( "MODEL_CMQ", "CMQ_NUMDOC", "1504")
            nCMQ++
            cCMQPath := cCMOPath + "/nfsTerceiros[" + AllTrim(Str(nCMQ)) + "]"
        EndDo // Fim - CMQ

        nCMO++
        cCMOPath := cInfEvento + "/infoServTomados/ideEstabelecimento[" +
AllTrim(Str(nCMO)) + "]"
        EndDo // Fim - CMO
    EndIf

    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAç
    //³Efetiva a operação desejada³
    //AAAAAAAAAAAAAAAAAAAAAAAAAAAAU
    FWFormCommit( oModel )
    lRet := .T.

    oModel:DeActivate()

EndIf
End Transaction

```



### S-1340 – eSocial Mensal – Outras Informações – Serv. Prestados pela Cooperativa de Trabalho

<b>Rotina:</b>	TAF290				
<b>Model MVC:</b>	MODEL_CMN; MODEL_CMO; MODEL_CMP; MODEL_CMQ				
<b>Validação</b>	TAF290VLD()				
<b>Geração XML</b>	TAF290XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpServPrestadoCoop	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CMN	CMN_INDAPU	
	09	perApuracao	MODEL_CMN	CMN_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoServPrestados	-		
	18	ideEstabPrestador	-		
	19	tpInscricao	MODEL_CMO	CMO_TPINSC	
	20	nrInscricao	MODEL_CMO	CMO_INSCES	
	21	ideContratante	-		
	22	tpInscContratante	MODEL_CMP	CMP_TPINSC	
	23	nrInscContratante	MODEL_CMP	CMP_INSCCO	
	24	vlrBruto	MODEL_CMP	CMP_VLBRUT	
	25	vlrBaseCoop	MODEL_CMP	CMP_BCCOOP	
	26	nfsEmitidas	-		
	27	Serie	MODEL_CMQ	CMQ_SERIE	
	28	numDocto	MODEL_CMQ	CMQ_NUMDOC	
	29	dtEmissaoNF	MODEL_CMQ	CMQ_DTEMIS	
	30	dtPagtoNF	MODEL_CMQ	CMQ_DTPGTO	
	31	indObra	MODEL_CMQ	CMQ_INDOBR	
	32	nrCno	MODEL_CMQ	CMQ_NRCNO	
	33	vlrBruto	MODEL_CMQ	CMQ_VLBRUT	
	34	vlrMatEquip	MODEL_CMQ	CMQ_VLMATE	
	35	vlrServicos	MODEL_CMQ	CMQ_VLSERV	
	36	vlrDeducoes	MODEL_CMQ	CMQ_VLDEDU	
	37	vlrBaseCoop	MODEL_CMQ	CMQ_BCCOOP	
	38	servCoopCondEspeciais	-		
	39	vlrServicos15	MODEL_CMQ	CMQ_VLSV15	
	40	vlrServicos20	MODEL_CMQ	CMQ_VLSV20	
	41	vlrServicos25	MODEL_CMQ	CMQ_VLSV25	
	42	proprietarioCNO	-		
	43	tpInscProprietario	MODEL_CMQ	CMQ_TPINSC	
44	nrInscProprietario	MODEL_CMQ	CMQ_INSCPR		



## Exemplo:

```

cXml := xFunLerXML()

oDados := TXmlManager():New()
oDados.Parse(cXml)

Aadd( aChave, { "C", "CMN_NOMEVE", "S1340", .T. } )
Aadd( aChave, { "C", "CMN_INDAPU", "1", .T. } )
Aadd( aChave, { "C", "CMN_PERAPU", "102013", .T. } )

Begin Transaction
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If FTafVldOpe( 'CMN', 2, @nOpc, cFilEv, @aIncons, aChave, @oModel, 'TAFA290', cCmpsNoUpd )

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If nOpc <> 5
    oModel:LoadValue( "MODEL_CMN", "CMN_FILIAL", CMN->CMN_FILIAL )
    oModel:LoadValue( 'MODEL_CMN', 'CMN_NOMEVE', "S1340" )

    // CMO
    nCMO := 1
    cCMOPath := cInfEvento + "/infoServPrestados/ideEstabPrestador[" + AllTrim(Str(nCMO)) +
"]"

    If nOpc == 4 .And. oDados.XPathHasNode( cCMOPath )
        For nJ := 1 to oModel:GetModel( 'MODEL_CMO' ):Length()
            oModel:GetModel( 'MODEL_CMO' ):GoLine(nJ)
            oModel:GetModel( 'MODEL_CMO' ):DeleteLine()
        Next nJ
    EndIf

    While oDados.XPathHasNode( cCMOPath )

        If nOpc == 4 .Or. nCMO > 1
            // Informa que a linha está válida para que seja incluída uma nova linha
            oModel:GetModel( "MODEL_CMO" ):LVALID := .T.

            // Inclui uma linha a cada volta do laço
            oModel:GetModel( "MODEL_CMO" ):AddLine()

        EndIf

        // Grava dados no model
        oModel:LoadValue( "MODEL_CMO", "CMO_TPINSC" )
        oModel:LoadValue( "MODEL_CMO", "CMO_INSCES", "665487991" )

        // CMP
        nCMP := 1
        cCMPPath := cCMOPath + "/ideContratante[" + AllTrim(Str(nCMP)) + "]"

        If nOpc == 4 .And. oDados.XPathHasNode( cCMPPath )
            For nJ := 1 to oModel:GetModel( 'MODEL_CMP' ):Length()
                oModel:GetModel( 'MODEL_CMP' ):GoLine(nJ)
                oModel:GetModel( 'MODEL_CMP' ):DeleteLine()
            Next nJ
        EndIf

        While oDados.XPathHasNode( cCMPPath )

            If nOpc == 4 .Or. nCMP > 1
                // Informa que a linha está para que seja incluída uma nova
                oModel:GetModel( "MODEL_CMP" ):LVALID := .T.

                // Inclui uma linha a cada volta do laço

```

linha



Este documento é de propriedade da TOTVS. Todos os direitos reservados. ©

Este documento é de propriedade da TOTVS. Todos os direitos reservados. ©

Este documento é de propriedade da TOTVS. Todos os direitos reservados. ©

```

oModel:GetModel( "MODEL_CMP" ):AddLine()
EndIf
oModel:LoadValue( "MODEL_CMP", "CMP_BCCOOP", "500")

// CMQ
nCMQ := 1
cCMQPath := cCMPPath + "/nfsEmitidas[" + AllTrim(Str(nCMQ)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCMQPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CMQ' ):Length()
oModel:GetModel( 'MODEL_CMQ' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CMQ' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCMQPath )

If nOpc == 4 .Or. nCMQ > 1
// Informa que a linha está para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CMQ" ):LINVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CMQ" ):AddLine()
EndIf
oModel:LoadValue( "MODEL_CMQ", "CMQ_BCCOOP", "305")

If oDados:XPathHasNode( cCMQPath + "/servCoopCondEspeciais" )
oModel:LoadValue( "MODEL_CMQ", "CMQ_VLSV25", "200")
EndIf

nCMQ++
cCMQPath := cCMPPath + "/nfsEmitidas[" + AllTrim(Str(nCMQ)) + "]"
"]"

EndDo // Fim - CMQ

nCMP++
cCMPPath := cCMOPath + "/ideContratante[" + AllTrim(Str(nCMP)) + "]"
EndDo // Fim - CMP

nCMO++
cCMOPath := cInfEvento + "/infoServPrestados/ideEstabPrestador[" +
AllTrim(Str(nCMO)) + "]"
EndDo // Fim - CMO
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAU
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction
    
```

### S-1350 – eSocial Mensal – Outras Informações – Aquisição de Produção

<b>Rotina:</b>	TAF272				
<b>Model MVC:</b>	MODEL_CMR				
<b>Validação</b>	TAF272VLD()				
<b>Geração XML</b>	TAF272XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpAquisProducao	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CMR	CMR_INDAPU	
	09	perApuracao	MODEL_CMR	CMR_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoAquisProducao	-		
	18	ideEstabAdquirente	-		
	19	tpInscricao	MODEL_CMS	CMS_TPINSC	
	20	nrInscricao	MODEL_CMS	CMS_INSCES	
	21	tipoAquisicao	-		
	22	indAquisicao	MODEL_CMT	CMT_INDAQU	
	23	vlrTotalAquisicao	MODEL_CMT	CMT_VLAQUI	
	24	ideProdutor	-		
	25	tpInscricao	MODEL_CMU	CMU_TPINSC	
	26	nrInscricao	MODEL_CMU	CMU_INSCPR	
	27	vlrBruto	MODEL_CMU	CMU_VLBRUT	
	28	vlrContribDescPR	MODEL_CMU	CMU_VLCONT	
	29	vlrGilratDescPR	MODEL_CMU	CMU_VLGILR	
	30	vlrSenarDescPR	MODEL_CMU	CMU_VLSENA	
	31	notasFiscais	-		
	32	Serie	MODEL_CMV	CMV_SERIE	
	33	numDocto	MODEL_CMV	CMV_NUMDOC	
	34	dtEmissaoNF	MODEL_CMV	CMV_DTEMIS	
	35	vlrBruto	MODEL_CMV	CMV_VLBRUT	
	36	vlrContribDescPR	MODEL_CMV	CMV_VLCONT	
	37	vlrGilratDescPR	MODEL_CMV	CMV_VLGILR	
	38	vlrSenarDescPR	MODEL_CMV	CMV_VLSENA	
	39	infoProcJudicial	-		
40	nrProcJud	MODEL_CMU	CMU_NRPROC		





```

oModel:GetModel( "MODEL_CMT" ):LINVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CMT" ):AddLine()

EndIf

// Grava dados no model
oModel:LoadValue( "MODEL_CMT", "CMT_VLAQUI", "1200")

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAç
//³INICIO CMU - <ideProdutor>³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAU
nCMU := 1
cCMUPath := cCMTPath + "/ideProdutor[" + AllTrim(Str(nCMU)) + "]"

//Deleta as linhas existentes se for alteraca
If nOpc == 4 .And. oDados:XPathHasNode( cCMUPath )
  For nJ := 1 to oModel:GetModel( 'MODEL_CMU' ):Length()
    oModel:GetModel( 'MODEL_CMU' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_CMU' ):DeleteLine()
  Next nJ
EndIf

While oDados:XPathHasNode( cCMUPath )

  If nOpc == 4 .Or. nCMU > 1
    // Informa que a linha está válida para que seja incluída
    oModel:GetModel( "MODEL_CMU" ):LINVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CMU" ):AddLine()

  EndIf

  // Grava dados no model
  oModel:LoadValue( "MODEL_CMU", "CMU_VLSENA", "1250")

  //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAç
  //³INICIO CMV - <notasFiscais>³
  //AAAAAAAAAAAAAAAAAAAAAAAAAAAAU

  nCMV := 1
  cCMVPath := cCMUPath + "/notasFiscais[" + AllTrim(Str(nCMV)) + "]"

  //Deleta as linhas existentes se for alteração
  If nOpc == 4 .And. oDados:XPathHasNode( cCMVPath )
    For nJ := 1 to oModel:GetModel( 'MODEL_CMV' ):Length()
      oModel:GetModel( 'MODEL_CMV' ):GoLine(nJ)
      oModel:GetModel( 'MODEL_CMV' ):DeleteLine()
    Next nJ
  EndIf

  While oDados:XPathHasNode( cCMVPath )

    If nOpc == 4 .Or. nCMV > 1
      // Informa que a linha está válida para que seja
      oModel:GetModel( "MODEL_CMV" ):LINVALID := .T.

      // Inclui uma linha a cada volta do laço
      oModel:GetModel( "MODEL_CMV" ):AddLine()

    EndIf

    // Grava dados no model
    oModel:LoadValue( "MODEL_CMV", "CMV_VLSENA", "500")

    nCMV++
    cCMVPath := cCMUPath + "/notasFiscais[" +

AllTrim(Str(nCMV)) + "]"

EndDo

```

uma nova linha

incluída uma nova linha





## S-1360 – eSocial Mensal – Outras Informações – Comercialização da Produção

<b>Rotina:</b>	TAF252				
<b>Model MVC:</b>	MODEL_C8Q; MODEL_C9C; MODEL_C9D; MODEL_C9E				
<b>Validação</b>	TAF252VLD()				
<b>Geração XML</b>	TAF252XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpComercProducao	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_C8Q	C8Q_INDAPU	
	09	perApuracao	MODEL_C8Q	C8Q_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoComercProducao	-		
	18	ideEstabelecimento	-		
	19	tpInscricao	MODEL_C9C	C9C_ESTABE	tpEstabelecimento
	20	nrInscricao	-	-	
	21	tipoComercializacao	-		
	22	indComercializacao	MODEL_C9D	C9D_INDCOM	
	23	vlrTotalComercializacao	MODEL_C9D	C9D_VLTCOM	
	24	ideAdquirente	-		
	25	tpInscricao	MODEL_C9E	C9E_TPINCA	
	26	nrInscricao	MODEL_C9E	C9E_NRINCA	
	27	vlrComercializacao	MODEL_C9E	C9E_VBCPRD	
28	vlrRetidoProdRural	MODEL_C9E	C9E_VRETPR		

### Exemplo:

```

cXml := xFunLerXML()

oDados := TXmlManager():New()
oDados.Parse(cXml)

Aadd( aChave, { "C", "C8Q_INDAPU", "1", .T. } )
Aadd( aChave, { "C", "C8Q_PERAPU", "102013", .T. } )

Begin Transaction
    If FTafVldOpe( "C8Q", 2, @nOpc, cFilEv, @aIncons, aChave, @oModel, "TAF252", cCmpsNoUpd )
        oModel.LoadValue( "MODEL_C8Q", "C8Q_FILIAL", C8Q->C8Q_FILIAL )
    
```



```

If nOpc == 4
  For nJ := 1 to oModel:GetModel( 'MODEL_C9C' ):Length()
    oModel:GetModel( 'MODEL_C9C' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_C9C' ):DeleteLine()
  Next nJ
EndIf

/*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
³Roda o XML parseado para gravar as novas informações no GRID³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
nJ := 1
While oDados:XPathHasNode(cCabec + "/infoComercProducao/ideEstabelecimento[" + CVALTOCHAR(nJ)
+ "]" )

  If nOpc == 4 .Or. nJ > 1
    oModel:GetModel( 'MODEL_C9C' ):LVALID := .T.
    oModel:GetModel( 'MODEL_C9C' ):AddLine()
  EndIf
  oModel:LoadValue( "MODEL_C9C", "C9C_ESTABE", FGetIdInt(
"tpEstabelecimento"
, "" , cTpInsc,cNrInsc,.F.))

  /* ³ Tipo de Comercializacao ³*/
  /*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
  If nOpc == 4
    For nX:= 1 to oModel:GetModel( 'MODEL_C9D' ):Length()
      oModel:GetModel( 'MODEL_C9D' ):GoLine(nX)
      oModel:GetModel( 'MODEL_C9D' ):DeleteLine()
    Next nX
  EndIf
  /*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
³Roda o XML parseado para gravar as novas informações no GRID³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
  nX := 1
  While oDados:XPathHasNode(cCabec + "/infoComercProducao/ideEstabelecimento[" +
CVALTOCHAR(nJ) + "]/tipoComrcializacao[" + CVALTOCHAR(nX) + "]" )

    If nOpc == 4 .Or. nX > 1
      oModel:GetModel( 'MODEL_C9D' ):LVALID := .T.
      oModel:GetModel( 'MODEL_C9D' ):AddLine()
    EndIf
    oModel:LoadValue( "MODEL_C9D", "C9D_INDCOM", "1")
    // ³ Adquirente da Comercialização ³
    /*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/

    If nOpc == 4
      For nI := 1 to oModel:GetModel( 'MODEL_C9E' ):Length()
        oModel:GetModel( 'MODEL_C9E' ):GoLine(nI)
        oModel:GetModel( 'MODEL_C9E' ):DeleteLine()
      Next nI
    EndIf

    /*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
³Roda o XML parseado para gravar as novas informações no GRID³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
    nI := 1
    While oDados:XPathHasNode(cCabec + "/infoComercProducao/ideEstabelecimento[" +
CVALTOCHAR(nJ) + "]/tipoComercializacao[" + CVALTOCHAR(nX) + "]/ideAdquirente[" + CVALTOCHAR(nI) + "]" )

      If nOpc == 4 .Or. nI > 1
        oModel:GetModel( 'MODEL_C9E' ):LVALID := .T.
        oModel:GetModel( 'MODEL_C9E' ):AddLine()
      EndIf
      oModel:LoadValue( "MODEL_C9E", "C9E_VBCPRD", "10")
      nI++
    EndDo
    nX++
  EndDo
  nJ++

```



```
EndDo
/*ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAÿ
³Efetiva a operação desejada³
ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
FWFormCommit( oModel )
lRet := .T.
oModel:DeActivate ()

EndIf
End Transaction
```



### S-1370 – eSocial Mensal – Outras Informações – Rec. Recebidos ou Repassados p/ Clube de Futebol

<b>Rotina:</b>	TAF255				
<b>Model MVC:</b>	MODEL_C9B; MODEL_C9G; MODEL_C9H; MODEL_C9I;				
<b>Validação</b>	TAF255VLD()				
<b>Geração XML</b>	TAF255XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpAssocDesportiva	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_C9B	C9B_INDAPU	
	09	perApuracao	MODEL_C9B	C9B_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
	16	nrInscricao	-		
	17	infoRecursoRecebido	-		
	18	recursosRecebidos	-		
	19	cnjEmpOrigemRecurso	MODEL_C9G	C9G_CNPJOR	
	20	tpRepasso	MODEL_C9G	C9G_TPREPO	
	21	dtRepasso	MODEL_C9G	C9G_DTREPA	
	22	vlrRepasso	MODEL_C9G	C9G_VLREPA	
	23	infoRecursoRepassado	-		
	24	ideEstabelecimento	-		
	25	tpInscricao	MODEL_C9H	C9H_ESTABE	tpEstabelecimento nrEstabelecimento
	26	nrInscricao	-	-	
	27	vlrTotalRepasses	MODEL_C9H	C9H_VTOTRE	
	28	recursosRepassados	-		
	29	cnjAssocDesportiva	MODEL_C9I	C9I_CNPJAD	
	30	tpRepasso	MODEL_C9I	C9I_TPREPA	
	31	dtRepasso	MODEL_C9I	C9I_DTREPA	
32	vlrRepasso	MODEL_C9I	C9I_VLREPA		

#### Exemplo:

```
cXml := xFunLerXML()
oDados := TXmlManager().New()
oDados.Parse(cXml)
```



```

Aadd( aChave, { "C", "C9B INDAPU", "1 ", .T. } )
Aadd( aChave, { "C", "C9B PERAPU", "122013 ", .T. } )

Begin Transaction
//UAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
If FTafVldOpe( 'C9B', 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, 'TAF255', cCmpsNoUpd )

    If nOpc <> 5
        oModel:LoadValue( "MODEL_C9B", "C9B_FILIAL", C9B->C9B_FILIAL )
            // C9G
        nC9G := 1
        cC9GPath := cInfEvento + "/infoRecursoRecebido/recursosRecebidos[" + AllTrim(Str(nC9G))
+ "]"

        If nOpc == 4 .And. oDados:XPathHasNode( cC9GPath )
            For nJ := 1 to oModel:GetModel( 'MODEL_C9G' ):Length()
                oModel:GetModel( 'MODEL_C9G' ):GoLine(nJ)
                oModel:GetModel( 'MODEL_C9G' ):DeleteLine()
            Next nJ
        EndIf

        While oDados:XPathHasNode( cC9GPath )

            If nOpc == 4 .Or. nC9G > 1
                // Informa que a linha está para que seja incluída uma nova linha
                oModel:GetModel( "MODEL_C9G" ):LVALID := .T.

                // Inclui uma linha a cada volta do laço
                oModel:GetModel( "MODEL_C9G" ):AddLine()
            EndIf
            oModel:LoadValue( "MODEL_C9G", "C9G_VLREPA", "10" )

            nC9G++
            cC9GPath := cInfEvento + "/infoRecursoRecebido/recursosRecebidos[" +
AllTrim(Str(nC9G)) + "]"
            EndDo // Fim - C9G

            // C9H
            nC9H := 1
            cC9HPath := cInfEvento + "/infoRecursoRepassado/ideEstabelecimento[" +
AllTrim(Str(nC9H)) + "]"

            If nOpc == 4 .And. oDados:XPathHasNode( cC9HPath )
                For nJ := 1 to oModel:GetModel( 'MODEL_C9H' ):Length()
                    oModel:GetModel( 'MODEL_C9H' ):GoLine(nJ)
                    oModel:GetModel( 'MODEL_C9H' ):DeleteLine()
                Next nJ
            EndIf

            While oDados:XPathHasNode( cC9HPath )

                If nOpc == 4 .Or. nC9H > 1
                    // Informa que a linha está válida para que seja incluída uma nova linha
                    oModel:GetModel( "MODEL_C9H" ):LVALID := .T.

                    // Inclui uma linha a cada volta do laço
                    oModel:GetModel( "MODEL_C9H" ):AddLine()
                EndIf
                oModel:LoadValue( "MODEL_C9H", "C9H_ESTABE", "110" )

                // C9I
                nC9I := 1
                cC9IPath := cC9HPath + "/recursosRepassados[" + AllTrim(Str(nC9I)) + "]"

                If nOpc == 4 .And. oDados:XPathHasNode( cC9IPath )
                    For nJ := 1 to oModel:GetModel( 'MODEL_C9I' ):Length()
                        oModel:GetModel( 'MODEL_C9I' ):GoLine(nJ)
                        oModel:GetModel( 'MODEL_C9I' ):DeleteLine()
                    Next nJ
                EndIf
            EndWhile

        EndWhile
    EndIf
EndTransaction
    
```





```

                                Next nJ
                                EndIf

                                While oDados:XPathHasNode( cC9IPath )

                                    If nOpc == 4 .Or. nC9I > 1

                                        // Informa que a linha está válida para que seja incluída uma
nova linha
                                        oModel:GetModel( "MODEL_C9I" ):LVALID := .T.

                                        // Inclui uma linha a cada volta do laço
                                        oModel:GetModel( "MODEL_C9I" ):AddLine()

                                    EndIf

                                    oModel:LoadValue( "MODEL_C9I", "C9I_TPREPA", "1")
                                    nC9I++
                                    cC9IPath := cC9HPath + "/recursosRepassados[" + AllTrim(Str(nC9I)) + "]"
                                EndDo // Fim - C9I

                                nC9H++
                                cC9HPath := cInfEvento + "/infoRecursoRepassado/ideEstabelecimento[" +
AllTrim(Str(nC9H)) + "]"
                                EndDo // Fim - C9H
                                EndIf

                                //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
                                //³Efetiva a operação desejada³
                                //AAAAAAAAAAAAAAAAAAAAAAAAAAU
                                FWFormCommit( oModel )
                                lRet := .T.

                                oModel:DeActivate()

                                EndIf
                            End Transaction
```

## S-1380 – eSocial Mensal – Informações Complementares Desoneração

<b>Rotina:</b>	TAF239				
<b>Model MVC:</b>	MODEL_CRR				
<b>Validação</b>	TAF239VLD()				
<b>Geração XML</b>	TAF239XML()				
<b>Regras</b>	#	TAG	Model	Campo	FGetIDInt()
	01	eSocial	-		
	02	evtFpInfoDesoneracao	-		
	03	Id	-		
	04	Versão	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CRR	CRR_INDAPU	
	09	perApuracao	MODEL_CRR	CRR_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		
16	nrInscricao	-			



17	recExpServicos	-		
18	recBrutaExp12m	MODEL_CRR	CRR_VLREXP	
19	recBrutaTot12m	MODEL_CRR	CRR_VLRTOT	
20	vlrTribRecBruta	MODEL_CRR	CRR_VLTRIB	
21	qtdMesesAtiv	MODEL_CRR	CRR_QTDMES	
22	percReducaoLei11774	MODEL_CRR	CRR_VLREDU	
23	infoSubstPatronal			
24	indSubstPatronal	MODEL_CRR	CRR_INDSUB	
25	percAliqPatronal	MODEL_CRR	CRR_ALIQPA	

## Exemplo:

```
Aadd( aChave, { "C", "CRR INDAPU", cCabec + "/ideEvento/indApuracao", .F. } )
Aadd( aChave, { "C", "CRR PERAPU", cCabec + "/ideEvento/perApuracao", .F. } )
```

```
Begin Transaction
```

```
    //Função para validar se a operação desejada pode ser realizada
    If FTafVldOpe( "CRR", nIndChv, @nOpc, cFilEv, @aIncons, aChave, @oModel, "TAF303",
cCmpsNoUpd, nIndIDVer, .F.)
        If nOpc <> 5
            oModel:LoadValue( "MODEL_CRR", "CRR_FILIAL", CRR->CRR_FILIAL )
            oModel:LoadValue( "MODEL_CRR", "CRR_VLREXP", "1" )
```

```
EndIf
```

```
FWFormCommit( oModel )
lRet := .T.
oModel:DeActivate()
EndIf
End Transaction
```

## S-1390 – eSocial Periódicos – Receitas de Atividades Concomitantes

<b>Rotina:</b>	TAF292				
<b>Model MVC:</b>	MODEL_				
<b>Validação</b>	TAF292VLD()				
<b>Geração XML</b>	TAF292XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtFpRecAtivConcomitante	-		
	03	Id	-		
	04	Versão	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CRV	CRV_INDAPU	
	09	perApuracao	MODEL_CRV	CRV_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
15	tplInscricao	-			



16	nrInscricao	-		
17	recAtivConcomitante	-		
18	recBrutaTotMes	MODEL_CRV	CRV_TOTMES	
19	recBrutaTotAno	MODEL_CRV	CRV_TOTANO	
20	recBrutaAtssMes	MODEL_CRV	CRV_ASSMES	
21	recBrutaAtssAno	MODEL_CRV	CRV_ASSANO	
22	fatorMes	MODEL_CVR	CRV_FATMES	
23	fator13	MODEL_CRV	CRV_FAT13	

## Exemplo

```
Aadd( aChave, { "C", "CRV_INDAPU", cCabec + "/ideEvento/indApuracao", .F. } )
Aadd( aChave, { "C", "CRV_PERAPU", cCabec + "/ideEvento/perApuracao", .F. } )
```

**Begin** Transaction

```
    //Função para validar se a operação desejada pode ser realizada
    If FTafVldOpe( "CRV", nIndChv, @nOpc, cFilEv, @aIncons, aChave, @oModel, "TAF3292",
cCmpsNoUpd, nIndIDVer, .F.)
        If nOpc <> 5
            oModel:LoadValue( "MODEL_CRV", "CRV_FILIAL", CRV->CRV_FILIAL )
            oModel:LoadValue( "MODEL_CRV", "CRV_TOTMES", "1" )
        EndIf
        FWFormCommit( oModel )
        lRet := .T.
        oModel:DeActivate()
    EndIf
```

**End** Transaction

## S-1399 – eSocial Mensal – Fechamento

<b>Rotina:</b>	TAF303				
<b>Model MVC:</b>	MODEL_CUO				
<b>Validação</b>	TAF303VLD()				
<b>Geração XML</b>	TAF303XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtFpFechamento	-		
	03	Id	-		
	04	Versão	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CUO	CUO_INDAPU	
	09	perApuracao	MODEL_CUO	CUO_PERAPU	
	10	tpAmb	-		
	11	procEmi	-		
	12	indSegmento	-		
	13	verProc	-		
	14	ideEmpregador	-		
	15	tpInscricao	-		



16	nrlInscricao	-		
17	InfoFechamento	-		
18	evtRemuneracao	MODEL_CUO	CUO_REMUN	
19	evtPagamentosDiversos	MODEL_CUO	CUO_PAGDIV	
20	evtServTomadosCMO	MODEL_CUO	CUO_SRVTCM	
21	evtServPrestadosCMO	MODEL_CUO	CUO_SRVPCM	
22	evtServTomadosCOOP	MODEL_CUO	CUO_SRVTCO	
23	evtServPrestadosCOOP	MODEL_CUO	CUO_SRVPCO	
24	evtAquisProdRural	MODEL_CUO	CUO_AQUPRU	
25	evtComercProducao	MODEL_CUO	CUO_COMPRO	
26	evtRepasseClubeFutebol	MODEL_CUO	CUO_REPACF	
27	evtInfoDesoneracao	MODEL_CUO	CUO_INFDES	
28	evtReceitaAtivConcomitantes	MODEL_CUO	CUO_RECCON	

## Exemplo

```
Aadd( aChave, { "C", "CUO_INDAPU", cCabec + "/ideEvento/indApuracao", .F. } )
Aadd( aChave, { "C", "CUO_PERAPU", cCabec + "/ideEvento/perApuracao", .F. } )
```

```
Begin Transaction
```

```
    //Função para validar se a operação desejada pode ser realizada
    If FTafVldOpe( "CUO", nIndChv, @nOpc, cFileEv, @aIncons, aChave, @oModel, "TAF303",
cCmpsNoUpd, nIndIDVer, .F.)
        If nOpc <> 5
            oModel:LoadValue( "MODEL_CUO", "CUO_FILIAL", CUO->CUO_FILIAL )
            oModel:LoadValue( "MODEL_CUO", "CUO_REMUN", "1" )
        EndIf
        FWFormCommit( oModel )
        lRet := .T.
        oModel:DeActivate()
    EndIf
```

```
End Transaction
```

## S-1400 – eSocial Mensal – Bases, Retenção, Deduções e Contribuições

<b>Rotina:</b>	TAF274				
<b>Model MVC:</b>	MODEL_CMX; MODEL_CMZ; MODEL_CR0; MODEL_CR1; MODEL_CR2; MODEL_CR3; MODEL_CR4; MODEL_CR5; MODEL_CR6; MODEL_CR7; MODEL_CR8;				
<b>Validação</b>	TAF274VLD()				
<b>Geração XML</b>	TAF274XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtFpBasesContrib	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	indApuracao	MODEL_CMX	CMX_INDAPU	
	09	perApuracao	MODEL_CMX	CMX_PERAPU	
	10	tpAmb	-		
11	procEmi	-			

12	indSegmento	-		
13	verProc	-		
14	ideEmpregador	-		
15	tpInscricao	-		
16	nrInscricao	-		
17	ideConteudo	-		
18	indConteudo	MODEL_CMX	CMX_INDCON	
19	infoBasesContrib	-		
20	ideEstabelecimento	-		
21	tpInscricao	MODEL_CMZ	CMZ_TPINSC	
22	nrInscricao	MODEL_CMZ	CMZ_INSCES	
23	basesPorCategoria			
24	indIncidencia	MODEL_CR0	CR0_INDINC	
25	codCateg	MODEL_CR0	CR0_CODCAT	catTrabalhador
26	vlrBcCP	MODEL_CR0	CR0_BASECP	
27	vlrBcCP15	MODEL_CR0	CR0_VLBC15	
28	vlrBcCP20	MODEL_CR0	CR0_VLBC20	
29	vlrBcCP25	MODEL_CR0	CR0_VLBC25	
30	vlrDescCP	MODEL_CR0	CR0_VLDESC	
31	vlrSalFam	MODEL_CR0	CR0_VLSALF	
32	vlrSalMat	MODEL_CR0	CR0_VLSALM	
33	basesPorOperPortuario	-		
34	cnpjOpPortuario	MODEL_CR1	CR1_CNPJOP	
35	vlrBcOpPortuario	MODEL_CR1	CR1_VLBCOP	
36	basesPorFpas	-		
37	Fpas	MODEL_CR2	CR2_FPAS	"fpas","codTerceiros"
38	codTerceiros			
39	bcTerceiros	MODEL_CR2	CR2_BCTERC	
40	contribPrevSocial	-		
41	indTpContrib	MODEL_CR3	CR3_INDTP	tpContrib
42	vlrContribuicao	MODEL_CR3	CR3_VLCONT	
43	contribOutrasEntidades	-		
44	Fpas	MODEL_CR4	CR4_FPAS	"fpas","codTerceiros"
45	codTerceiros			
46	vlrTotTerceiros	MODEL_CR4	CR4_VLTOT	
47	contribTerceiro	-		
48	codTerceiro	MODEL_CR5	CR5_TERCEI	"fpas","codTerceiros"
49	aliqTerceiro	MODEL_CR5	CR5_ALIQTE	
50	vlrTerceiro	MODEL_CR5	CR5_BASECP	
51	infoCessaoMaoObra	-		
52	retencaoEfetuada	-		
53	idePrestadorObra	-		
54	cnpjPrestador	MODEL_CR6	CR6_CNPJPR	
55	indObra	MODEL_CR6	CR6_INDOBR	
56	nrCno	MODEL_CR6	CR6_NRCNO	
57	vlrRetencao	MODEL_CR6	CR6_VLRETE	
58	vlrAdicional	MODEL_CR6	CR6_VLADIC	
59	vlrNaoRetido	MODEL_CR6	CR6_VLNRET	
60	proprietarioCNO	-		
61	tpInscProprietario	MODEL_CR6	CR6_TPINSC	
62	nrInscProprietario	MODEL_CR6	CR6_NRINSC	





```

oModel:GetModel( "MODEL_CMZ" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CMZ" ):AddLine()

EndIf

// Grava dados no model
oModel:LoadValue( "MODEL_CMZ", "CMZ TPINSC", "1" )
oModel:LoadValue( "MODEL_CMZ", "CMZ INSCES", "0123456789" )

// CR0
If nOpc == 4 .And. lCR0
For nJ := 1 to oModel:GetModel( 'MODEL_CR0' ):Length()
oModel:GetModel( 'MODEL_CR0' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CR0' ):DeleteLine()
Next nJ
EndIf

While !AliasCRO->EOF

If nOpc == 4 .Or. nCR0 > 1

// Informa que a linha está válida para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CR0" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CR0" ):AddLine()

EndIf
oModel:LoadValue( "MODEL_CR0", "CR0_VLSALF", "123,")
oModel:LoadValue( "MODEL_CR0", "CR0_VLSALM", "325")

nCR0++
EndDo // Fim - CR0

// CR1
If nOpc == 4 .And. oDados:XPathHasNode( cCR1Path )
For nJ := 1 to oModel:GetModel( 'MODEL_CR1' ):Length()
oModel:GetModel( 'MODEL_CR1' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CR1' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCR1Path )

If nOpc == 4 .Or. nCR1 > 1
// Informa que a linha está válida para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CR1" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL_CR1" ):AddLine()

EndIf

oModel:LoadValue( "MODEL_CR1", "CR1_CNPJOP", "00123000122")
oModel:LoadValue( "MODEL_CR1", "CR1_VLB COP", "123456", "N", .F., @aIncons, .T. )

EndDo // Fim - CR1

// CR2
If nOpc == 4 .And. oDados:XPathHasNode( cCR2Path )
For nJ := 1 to oModel:GetModel( 'MODEL_CR2' ):Length()
oModel:GetModel( 'MODEL_CR2' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CR2' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCR2Path )

```



```

If nOpc == 4 .Or. nCR2 > 1
    // Informa que a linha está válida para que seja incluída uma
nova linha
    oModel:GetModel( "MODEL_CR2" ):LVALID := .T.

    // Inclui uma linha a cada volta do laço
    oModel:GetModel( "MODEL_CR2" ):AddLine()
EndIf
    oModel:LoadValue( "MODEL_CR2", "CR2_FPAS", FTafGetVal( nGetId,
"C", .T., @aIncons, .T. ) )
    oModel:LoadValue( "MODEL_CR2", "CR2_BCTERC", FTafGetVal(
cCR2Path + "/bcTerceiros", "N", .F., @aIncons, .T. ) )

EndDo // Fim - CR2

// CR3

If nOpc == 4 .And. oDados:XPathHasNode( cCR3Path )
For nJ := 1 to oModel:GetModel( 'MODEL_CR3' ):Length()
    oModel:GetModel( 'MODEL_CR3' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_CR3' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCR3Path )

    If nOpc == 4 .Or. nCR3 > 1
nova linha
        // Informa que a linha está válida para que seja incluída uma

        oModel:GetModel( "MODEL_CR3" ):LVALID := .T.

        // Inclui uma linha a cada volta do laço
        oModel:GetModel( "MODEL_CR3" ):AddLine()
EndIf

        oModel:LoadValue( "MODEL_CR3", "CR3_INDTP", FTafGetVal( nGetId,
"C", .T., @aIncons, .T. ) )
        oModel:LoadValue( "MODEL_CR3", "CR3_VLCONT", FTafGetVal(
cCR3Path + "/vlrContribuicao", "N", .F., @aIncons, .T. ) )

EndDo // Fim - CR3

// CR4

If nOpc == 4 .And. oDados:XPathHasNode( cCR4Path )
For nJ := 1 to oModel:GetModel( 'MODEL_CR4' ):Length()
    oModel:GetModel( 'MODEL_CR4' ):GoLine(nJ)
    oModel:GetModel( 'MODEL_CR4' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCR4Path )

    If nOpc == 4 .Or. nCR4 > 1
linha
        // Informa que a linha está para que seja incluída uma nova

        oModel:GetModel( "MODEL_CR4" ):LVALID := .T.

        // Inclui uma linha a cada volta do laço
        oModel:GetModel( "MODEL_CR4" ):AddLine()
EndIf

        oModel:LoadValue( "MODEL_CR4", "CR4_FPAS", FTafGetVal( nGetId,
"C", .T., @aIncons, .T. ) )
        oModel:LoadValue( "MODEL_CR4", "CR4_VLTOT", FTafGetVal( cCR4Path
+ "/vlrTotTerceiros", "N", .F., @aIncons, .T. ) )

        // CR5

If nOpc == 4 .And. oDados:XPathHasNode( cCR5Path )
For nJ := 1 to oModel:GetModel( 'MODEL_CR5' ):Length()
    oModel:GetModel( 'MODEL_CR5' ):GoLine(nJ)

```





```

                                oModel:GetModel( 'MODEL CR5' ):DeleteLine()
                            Next nJ
                        EndIf

                        While oDados:XPathHasNode( cCR5Path )

                            If nOpc == 4 .Or. nCR5 > 1
                                // Informa que a linha está para que seja incluída uma
                                // nova linha
                                oModel:GetModel( "MODEL_CR5" ):LVALID := .T.

                                // Inclui uma linha a cada volta do laço
                                oModel:GetModel( "MODEL CR5" ):AddLine()

                            EndIf

                                oModel:LoadValue( "MODEL_CR5", "CR5_TERCEI", FTafGetVal(
cCR5Path + "/codTerceiro", "C", .F., @aIncons, .T. ) )
                                oModel:LoadValue( "MODEL_CR5", "CR5_ALIQTE", FTafGetVal(
cCR5Path + "/aliqTerceiro", "N", .F., @aIncons, .T. ) )
                                oModel:LoadValue( "MODEL_CR5", "CR5_BASECP", FTafGetVal(
cCR5Path + "/vlrTerceiro", "N", .F., @aIncons, .T. ) )
                            EndDo // Fim - CR5

                                nCR4++
                                cCR4Path := cCMZPath + "/contribOutrasEntidades[" + AllTrim(Str(nCR4)) +
"]"

                            EndDo // Fim - CR4

                                nCMZ++
                                cCMZPath := cInfEvento + "/infoBasesContrib/ideEstabelecimento[" +
AllTrim(Str(nCMZ)) + "]"
                            EndDo

                                // CR6
                                nCR6 := 1
                                cCR6Path := cInfEvento + "/infoCessaoMaoObra/retencaoEfetuada/idePrestadorObra[" +
AllTrim(Str(nCR6)) + "]"

                                If nOpc == 4 .And. oDados:XPathHasNode( cCR6Path )
                                    For nJ := 1 to oModel:GetModel( 'MODEL_CR6' ):Length()
                                        oModel:GetModel( 'MODEL CR6' ):GoLine(nJ)
                                        oModel:GetModel( 'MODEL CR6' ):DeleteLine()
                                    Next nJ
                                EndIf

                                While oDados:XPathHasNode( cCR6Path )

                                    If nOpc == 4 .Or. nCR6 > 1
                                        // Informa que a linha está para que seja incluída uma nova linha
                                        oModel:GetModel( "MODEL_CR6" ):LVALID := .T.

                                        // Inclui uma linha a cada volta do laço
                                        oModel:GetModel( "MODEL CR6" ):AddLine()

                                    EndIf

                                    nGetId := FGetIdInt( "fpas", "codTerceiros", cCR6Path + "/fpas", cCR6Path +
"/codTerceiros" )

                                    oModel:LoadValue( "MODEL CR6", "CR6 CNPJPR", FTafGetVal( cCR6Path +
"/cnpjPrestador", "C", .F., @aIncons, .T. ) )
                                    oModel:LoadValue( "MODEL_CR6", "CR6_INDOBR", FTafGetVal( cCR6Path + "/indObra",
"C", .F., @aIncons, .T. ) )
                                    oModel:LoadValue( "MODEL CR6", "CR6 NRCNO", FTafGetVal( cCR6Path + "/nrCno",
"C", .F., @aIncons, .T. ) )
                                    oModel:LoadValue( "MODEL CR6", "CR6 VLRETE", FTafGetVal( cCR6Path +
"/vlrRetencao", "N", .F., @aIncons, .T. ) )
                                    oModel:LoadValue( "MODEL_CR6", "CR6 VLADIC", FTafGetVal( cCR6Path +
"/vlrAdicional", "N", .F., @aIncons, .T. ) )
                                    oModel:LoadValue( "MODEL CR6", "CR6 VLNRET", FTafGetVal( cCR6Path +
"/vlrNaoRetido", "N", .F., @aIncons, .T. ) )

```

```

        If oDados:XPathHasNode( cCR6Path + "/proprietarioCNO[1]" )
            oModel:LoadValue( "MODEL_CR6", "CR6_TPINSC", FTafGetVal( cCR6Path +
"/proprietarioCNO[1]/tpInscProprietario", "C", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR6", "CR6_NRINSC", FTafGetVal( cCR6Path +
"/proprietarioCNO[1]/nrInscProprietario", "C", .F., @aIncons, .T. ) )
        EndIf

        nCR6++
        cCR6Path := cInfEvento +
"/infoCessaoMaoObra/retencaoEfetuada/idePrestadorObra[" + AllTrim(Str(nCR6)) + "]"
        EndDo // Fim - CR6

        // CR7
        nCR7 := 1
        cCR7Path := cInfEvento + "/infoCessaoMaoObra/retencaoSofrida/ideContratanteObra[" +
AllTrim(Str(nCR7)) + "]"

        If nOpc == 4 .And. oDados:XPathHasNode( cCR7Path )
            For nJ := 1 to oModel:GetModel( 'MODEL_CR7' ):Length()
                oModel:GetModel( 'MODEL_CR7' ):GoLine(nJ)
                oModel:GetModel( 'MODEL_CR7' ):DeleteLine()
            Next nJ
        EndIf

        While oDados:XPathHasNode( cCR7Path )

            If nOpc == 4 .Or. nCR7 > 1
                // Informa que a linha está para que seja incluída uma nova linha
                oModel:GetModel( "MODEL_CR7" ):LVALID := .T.

                // Inclui uma linha a cada volta do laço
                oModel:GetModel( "MODEL_CR7" ):AddLine()
            EndIf

            nGetId := FGetIdInt( "fpas", "codTerceiros", cCR7Path + "/fpas", cCR7Path +
"/codTerceiros" )

            oModel:LoadValue( "MODEL_CR7", "CR7_TPINOBR", FTafGetVal( cCR7Path +
"/tpInscContratante", "C", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_INSCCO", FTafGetVal( cCR7Path +
"/nrInscContratante", "C", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_INDOBR", FTafGetVal( cCR7Path + "/indObra",
"C", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_NRCNO", FTafGetVal( cCR7Path + "/nrCno",
"C", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_VLRETE", FTafGetVal( cCR7Path +
"/vlrRetencao", "N", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_VLADIC", FTafGetVal( cCR7Path +
"/vlrAdicional", "N", .F., @aIncons, .T. ) )
            oModel:LoadValue( "MODEL_CR7", "CR7_VLNRET", FTafGetVal( cCR7Path +
"/vlrNaoRetido", "N", .F., @aIncons, .T. ) )

            If oDados:XPathHasNode( cCR7Path + "/proprietarioCNO[1]" )
                oModel:LoadValue( "MODEL_CR7", "CR7_TPINSC", FTafGetVal( cCR7Path +
"/proprietarioCNO[1]/tpInscProprietario", "C", .F., @aIncons, .T. ) )
                oModel:LoadValue( "MODEL_CR7", "CR7_NRINSC", FTafGetVal( cCR7Path +
"/proprietarioCNO[1]/nrInscProprietario", "C", .F., @aIncons, .T. ) )
            EndIf

            nCR7++
            cCR7Path := cInfEvento +
"/infoCessaoMaoObra/retencaoSofrida/ideContratanteObra[" + AllTrim(Str(nCR7)) + "]"
            EndDo // Fim - CR7

            // CR8
            nCR8 := 1
            cCR8Path := cInfEvento + "/infoCessaoMaoObra/remunPorContratante[" + AllTrim(Str(nCR8))
+ "]"

            If nOpc == 4 .And. oDados:XPathHasNode( cCR8Path )
                For nJ := 1 to oModel:GetModel( 'MODEL_CR8' ):Length()

```



```

oModel:GetModel( 'MODEL CR8' ):GoLine(nJ)
oModel:GetModel( 'MODEL CR8' ):DeleteLine()

Next nJ
EndIf

While oDados:XPathHasNode( cCR8Path )

If nOpc == 4 .Or. nCR8 > 1
// Informa que a linha está para que seja incluída uma nova linha
oModel:GetModel( "MODEL_CR8" ):LVALID := .T.

// Inclui uma linha a cada volta do laço
oModel:GetModel( "MODEL CR8" ):AddLine()
EndIf

nGetId := FGetIdInt( "catTrabalhador", , cCR8Path + "/codCateg" )

oModel:LoadValue( "MODEL_CR8", "CR8_TPINSC", FTafGetVal( cCR8Path +
"/tpInscContratante", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CR8", "CR8_INSCCO", FTafGetVal( cCR8Path +
"/nrInscContratante", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CR8", "CR8_NRCNO", FTafGetVal( cCR8Path + "/nrCno",
"C", .F., @aIncons, .F. ) )
oModel:LoadValue( "MODEL CR8", "CR8 CODCAT", FTafGetVal( nGetId, "C", .T.,
@aIncons, .T. ) )
oModel:LoadValue( "MODEL_CR8", "CR8_BCCP", FTafGetVal( cCR8Path + "/bcCP", "N",
.F., @aIncons, .T. ) )

If oDados:XPathHasNode( cCR8Path + "/proprietarioCNO[1]" )
oModel:LoadValue( "MODEL CR8", "CR8_TPINSP", FTafGetVal( cCR8Path +
"/proprietarioCNO[1]/tpInscProprietario", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CR8", "CR8_NRINSC", FTafGetVal( cCR8Path +
"/proprietarioCNO[1]/nrInscProprietario", "C", .F., @aIncons, .T. ) )
EndIf

nCR8++
cCR8Path := cInfEvento + "/infoCessaoMaoObra/remunPorContratante[" +
AllTrim(Str(nCR8)) + "]"
EndDo // Fim - CR8
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction

```

### S-1800 – eSocial Mensal – Espetáculo Desportivo

<b>Rotina:</b>	TAF298				
<b>Model MVC:</b>	MODEL_CRS; MODEL_CRT; MODEL_CUG; MODEL_CUH; MODEL_CUI; MODEL_CUJ; MODEL_CUK; MODEL_CUL;				
<b>Validação</b>	TAF298VLD()				
<b>Geração XML</b>	TAF298XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtEspDesportivo	-		
	03	Id	-		
	04	Versão	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	dtApuracao	MODEL_CRS	CRS_DTAPUR	
	09	tpAmb	-		
	10	procEmi	-		
	11	indSegmento	-		
	12	verProc	-		
	13	ideEmpregador	-		
	14	tpInscricao	-		
	15	nrInscricao	-		
	16	ideEstabelecimento	-		
	17	tpInscricao	MODEL_CRT	CRT_TPINSC	
	18	nrInscricao	MODEL_CRT	CRT_NRINSC	
	19	boletim	-		
	20	nrBoletim	MODEL_CUG	CUG_NBOLET	
	21	tpCompeticao	MODEL_CUG	CUG_TCOMPE	
	22	categEvento	MODEL_CUG	CUG_CTEVEN	
	23	nomeCompeticao	MODEL_CUG	CUG_NOMCOM	
	24	cnpjMandante	MODEL_CUG	CUG_CNPJMA	
	25	cnpjMandante	MODEL_CUG	CUG_CNPJVI	
	26	nomeVisitante	MODEL_CUG	CUG_NOMVIS	
	27	pracaDesportiva	MODEL_CUG	CUG_PRADES	
	28	codMunicipio	MODEL_CUG	CUG_CODMUN	"uf", "codMunicipio"
	29	uf	MODEL_CUG	CUG_UF	uf
	30	dtRealizacao	MODEL_CUG	CUG_DTREAL	
	31	hrRealizacao	MODEL_CUG	CUG_HRINIC	
	32	qtdePagantes	MODEL_CUG	CUG_QTDPAG	
	33	qtdeNaoPagantes	MODEL_CUG	CUG_QTDNOP	
	34	infoIngressos	-		
	35	cnpjResponsavel	MODEL_CUH	CUH_CNPJRE	
	36	tpInscricao	MODEL_CUH	CUH_TINSCR	
	37	nrInscricao	MODEL_CUH	CUH_NUMINS	
	38	qtdIngressos	MODEL_CUH	CUH_QTDING	
	39	receitaIngressos	-		
	40	tpIngresso	MODEL_CUI	CUI_TPINGR	
	41	tpPreco	MODEL_CUI	CUI_TPRECO	





```

// Informa que a linha está válida para que seja incluída uma nova linha
oModel:GetModel( "MODEL CRT" ):LVALID := .T.
oModel:GetModel( "MODEL CRT" ):AddLine()

EndIf

// Grava dados no model
oModel:LoadValue( "MODEL CRT", "CRT TPINSC", FTafGetVal( cCRTPath + "/tpInscricao", "C", .F.,
@aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT NRINSC", FTafGetVal( cCRTPath + "/nrInscricao", "C", .F.,
@aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT_CODCON", FTafGetVal( FGetIdInt( "tpContrib", "", ;
cCRTPath +
"/contribPrevSocial/indTpContrib"), "C", .T., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT VLCONT", FTafGetVal( cCRTPath +
"/contribPrevSocial/vlrContribuicao", "N", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT_VLRBRU", FTafGetVal( cCRTPath +
"/receitaTotal/vlrReceitaTotal", "N", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT_VLCLUB", FTafGetVal( cCRTPath +
"/receitaTotal/vlrReceitaClubes", "N", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CRT", "CRT_VLPARC", FTafGetVal( cCRTPath +
"/receitaTotal/vlrRetidoParcelamento", "N", .F., @aIncons, .T. ) )

// CUG
nCUG := 1
cCUGPath := cCRTPath + "/boletim[" + AllTrim(Str(nCUG)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCUGPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CUG' ):Length()
oModel:GetModel( 'MODEL_CUG' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CUG' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCUGPath )

If nOpc == 4 .Or. nCUG > 1
// Informa que a linha está para que seja incluída uma nova
linha
oModel:GetModel( "MODEL_CUG" ):LVALID := .T.
oModel:GetModel( "MODEL_CUG" ):AddLine()

EndIf

oModel:LoadValue( "MODEL_CUG", "CUG_NBOLET", FTafGetVal( cCUGPath + "/nrBoletim", "C",
.F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG TCOMPE", FTafGetVal( cCUGPath + "/tpCompeticao",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG CTEVEN", FTafGetVal( cCUGPath + "/categEvento",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG_NOMCOM", FTafGetVal( cCUGPath + "/nomeCompeticao",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG CNPJMA", FTafGetVal( cCUGPath + "/cnpjMandante",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG CNPJVI", FTafGetVal( cCUGPath + "/cnpjVisitante",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG_NOMVIS", FTafGetVal( cCUGPath + "/nomeVisitante",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG PRADES", FTafGetVal( cCUGPath +
"/pracaDesportiva", "C", .F., @aIncons, .T. ) )

oModel:LoadValue( "MODEL_CUG", "CUG_CODMUN", FTafGetVal( FGetIdInt(
"uf", "codMunicipio", ;
cCUGPath + "/uf", ;
cCUGPath + "/codMunicipio"), "C", .T., @aIncons, .F. ) )
oModel:LoadValue( "MODEL_CUG", "CUG_UF", FTafGetVal( FGetIdInt( "uf"
, "", ;
cCUGPath + "/uf" )
, "C", .T., @aIncons, .F. ) )

```



```

oModel:LoadValue( "MODEL CUG", "CUG DTREAL", FTafGetVal( cCUGPath + "/dtRealizacao",
"D", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CUG", "CUG HRINIC", FTafGetVal( cCUGPath + "/hrRealizacao",
"C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUG", "CUG_QTDPAG", FTafGetVal( cCUGPath + "/qtdePagantes",
"N", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL CUG", "CUG QTDNOP", FTafGetVal( cCUGPath + "/qtdeNaoPagantes",
"N", .F., @aIncons, .T. ) )

// CUH
nCUIH := 1
cCUIHPath := cCUGPath + "/infoIngressos[" + AllTrim(Str(nCUIH)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCUIHPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CUH' ):Length()
oModel:GetModel( 'MODEL_CUH' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CUH' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCUIHPath )

If nOpc == 4 .Or. nCUIH > 1
// Informa que a linha está para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CUH" ):LINVALID := .T.
oModel:GetModel( "MODEL_CUH" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CUH", "CUH CNPJRE", FTafGetVal(
cCUIHPath + "/cnpjResponsavel", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUH", "CUH TINSCR", FTafGetVal(
cCUIHPath + "/tpInscricao", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUH", "CUH NUMINS", FTafGetVal(
cCUIHPath + "/nrInscricao", "C", .F., @aIncons, .F. ) )
oModel:LoadValue( "MODEL_CUH", "CUH QTDDING", FTafGetVal(
cCUIHPath + "/qtdeIngressos", "N", .F., @aIncons, .T. ) )

nCUIH++
cCUIHPath := cCUGPath + "/infoIngressos[" + AllTrim(Str(nCUIH)) + "]"

EndDo // Fim - CUH

// CUI
nCUI := 1
cCUIPath := cCUGPath + "/receitaIngressos[" + AllTrim(Str(nCUI)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCUIPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CUI' ):Length()
oModel:GetModel( 'MODEL_CUI' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CUI' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCUIPath )

If nOpc == 4 .Or. nCUI > 1
// Informa que a linha está para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CUI" ):LINVALID := .T.
oModel:GetModel( "MODEL_CUI" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CUI", "CUI TPINGR", FTafGetVal(
cCUIPath + "/tpIngresso", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUI", "CUI TPRECO", FTafGetVal(
cCUIPath + "/tpPreco", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUI", "CUI QTDIVA", FTafGetVal(
cCUIPath + "/qtdeIngressosVenda", "N", .F., @aIncons, .F. ) )
oModel:LoadValue( "MODEL_CUI", "CUI QTDIVS", FTafGetVal(
cCUIPath + "/qtdeIngressosVendidos", "N", .F., @aIncons, .T. ) )

```



```

oModel:LoadValue( "MODEL_CUI", "CUI_QTDIDV", FTafGetVal(
cCUIPath + "/qtdeIngressosDevolvidos", "N", .F., @aIncons, .F. ) )
oModel:LoadValue( "MODEL_CUI", "CUI_PRCIND", FTafGetVal(
cCUIPath + "/precoIndividual", "N", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUI", "CUI_TOTARC", FTafGetVal(
cCUIPath + "/vlrTotalArrecadado", "N", .F., @aIncons, .F. ) )

nCUI++
cCUIPath := cCUGPath + "/receitaIngressos[" + AllTrim(Str(nCUI))
+ "]"

EndDo // Fim - CUI

// CUJ
nCUI := 1
cCUJPath := cCUGPath + "/outrasReceitas[" + AllTrim(Str(nCUI)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCUJPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CUJ' ):Length()
oModel:GetModel( 'MODEL_CUJ' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CUJ' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCUJPath )

If nOpc == 4 .Or. nCUI > 1
// Informa que a linha está para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CUJ" ):LINVALID := .T.
oModel:GetModel( "MODEL_CUJ" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CUJ", "CUJ_TPPEC", FTafGetVal( cCUJPath
+ "/tpReceita", "C", .F., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUJ", "CUJ_VLRREC", FTafGetVal(
cCUJPath + "/vlrReceita", "N", .F., @aIncons, .T. ) )

nCUI++
cCUJPath := cCUGPath + "/outrasReceitas[" + AllTrim(Str(nCUI)) +
"]"

EndDo // Fim - CUJ

// CUK
nCUI := 1
cCUKPath := cCUGPath + "/despesas[" + AllTrim(Str(nCUI)) + "]"

If nOpc == 4 .And. oDados:XPathHasNode( cCUKPath )
For nJ := 1 to oModel:GetModel( 'MODEL_CUK' ):Length()
oModel:GetModel( 'MODEL_CUK' ):GoLine(nJ)
oModel:GetModel( 'MODEL_CUK' ):DeleteLine()
Next nJ
EndIf

While oDados:XPathHasNode( cCUKPath )

If nOpc == 4 .Or. nCUI > 1
// Informa que a linha está para que seja incluída uma
nova linha
oModel:GetModel( "MODEL_CUK" ):LINVALID := .T.
oModel:GetModel( "MODEL_CUK" ):AddLine()
EndIf

oModel:LoadValue( "MODEL_CUK", "CUK_TPDESP", FTafGetVal(
FGetIdInt( "tpDespesa", "", ;
cCUKPath +
"/tpDespesa"), "C", .T., @aIncons, .T. ) )
oModel:LoadValue( "MODEL_CUK", "CUK_VLRDES", FTafGetVal(
cCUKPath + "/vlrDespesa", "N", .F., @aIncons, .T. ) )

// CUL
nCUI := 1

```





```

cCULPath := cCUKPath + "/detMaoObra[" + AllTrim(Str(nCUL)) + "]"

If nOpc == 4 .And. oDados.XPathHasNode( cCULPath )
    For nJ := 1 to oModel:GetModel( 'MODEL_CUL' ):Length()
        oModel:GetModel( 'MODEL_CUL' ):GoLine(nJ)
        oModel:GetModel( 'MODEL_CUL' ):DeleteLine()
    Next nJ
EndIf

While oDados.XPathHasNode( cCULPath )

    If nOpc == 4 .Or. nCUL > 1
        // Informa que a linha está para que seja incluída
        oModel:GetModel( "MODEL_CUL" ):LINVALID := .T.
        oModel:GetModel( "MODEL_CUL" ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL_CUL", "CUL_CPFTRB", FTafGetVal(
cCULPath + "/cpfTrab", "C", .F., @aIncons, .T. ) )
    oModel:LoadValue( "MODEL_CUL", "CUL_VLRREM", FTafGetVal(
cCULPath + "/vlrRemuneracao", "N", .F., @aIncons, .T. ) )
    oModel:LoadValue( "MODEL_CUL", "CUL_VLDESC", FTafGetVal(
cCULPath + "/vlrDescCP", "N", .F., @aIncons, .T. ) )

    nCUL++
    cCULPath := cCUKPath + "/detMaoObra[" +
AllTrim(Str(nCUL)) + "]"

    EndDo // Fim - CUL

    nCUK++
    cCUKPath := cCUGPath + "/despesas[" + AllTrim(Str(nCUK)) + "]"
    EndDo // Fim - CUK

    nCUG++
    cCUGPath := cCRTPath + "/boletim[" + AllTrim(Str(nCUG)) + "]"
    EndDo // Fim - CUG

    nCRT++
    cCRTPath := "/eSocial/evtEspDesportivo/ideEstabelecimento[" +
AllTrim(Str(nCRT)) + "]"
    EndDo // Fim - CRT
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction
    
```



## S-2100 – Evento Cadastramento Inicial do Vínculo

<b>Rotina:</b>	TAF256.prw				
<b>Model MVC:</b>	MODEL_C9V				
<b>Validação</b>	TAF256VLD()				
<b>Geração XML</b>	TAF256XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtCadInicial	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tplInscricao	-		
	14	nrInscricao	-		
	15	Trabalhador	-		
	16	cpfTrab	MODEL_C9V	C9V_CPF	
	17	nisTrab	MODEL_C9V	C9V_NIS	
	18	nomeTrab	MODEL_C9V	C9V_NOME	
	19	Sexo	MODEL_C9V	C9V_SEXO	
	20	racaCor	MODEL_C9V	C9V_RCCOR	
	21	estadoCivil	MODEL_C9V	C9V_ESTCIV	
	22	grauInstrucao	MODEL_C9V	C9V_GRINST	
	23	Nascimento	-		
	24	dtNascto	MODEL_C9V	C9V_DTNASC	
	25	codMunicipio	MODEL_C9V	C9V_CODMUN	Uf, codMunicipio
	26	Uf	MODEL_C9V	C9V_CODUF	Uf
	27	paisNascto	MODEL_C9V	C9V_CODPAI	paisNascto
	28	paisNacionalidade	MODEL_C9V	C9V_PAINAC	paisNacionalidade
	29	nomeMae	MODEL_C9V	C9V_NOMMAE	
	30	nomePai	MODEL_C9V	C9V_NOMPAI	
	31	Documentos	-		
	32	CTPS	-		
	33	nrCtps	MODEL_C9V	C9V_NRCTPS	
	34	serieCtps	MODEL_C9V	C9V_SERCTP	
	35	ufCtps	MODEL_C9V	C9V_UFCTPS	ufCtps
	36	RIC	-		
	37	nrRic	MODEL_C9V	C9V_NRRIC	
	38	orgaoEmissor	MODEL_C9V	C9V_OREMRI	
	39	dtExpedicao	MODEL_C9V	C9V_DTEXRI	
	40	RG	-		
	41	nrRg	MODEL_C9V	C9V_NRRG	
	42	orgaoEmissor	MODEL_C9V	C9V_OREMRG	
43	dtExpedicao	MODEL_C9V	C9V_DTEMRG		

44	RNE	-		
45	nrRne	MODEL_C9V	C9V_NRRNE	
46	orgaoEmissor	MODEL_C9V	C9V_OREMRN	
47	dtExpedicao	MODEL_C9V	C9V_DTEMRN	
48	OC	-		
49	nrOc	MODEL_C9V	C9V_NUMOC	
50	orgaoEmissor	MODEL_C9V	C9V_OREMOC	
51	dtExpedicao	MODEL_C9V	C9V_DTEXOC	
52	dtValidade	MODEL_C9V	C9V_DTVLOC	
53	CNH	-		
54	nrCnh	MODEL_C9V	C9V_NRCNH	
55	orgaoEmissor	MODEL_C9V	C9V_OREMCN	
56	dtExpedicao	MODEL_C9V	C9V_DTEXCN	
57	dtValidade	MODEL_C9V	C9V_DTVLCN	
58	Endereço	-		
59	Brasil	-		
60	tpLogradouro	MODEL_C9V	C9V_TPLOGR	tpLogradouro
61	descLogradouro	MODEL_C9V	C9V_LOGRAD	
62	nrLogradouro	MODEL_C9V	C9V_NRLOG	
63	Complemento	MODEL_C9V	C9V_COMLOG	
64	Bairro	MODEL_C9V	C9V_BAIRRO	
65	CEP	MODEL_C9V	C9V_CEP	
66	codMunicipio	MODEL_C9V	C9V_MUN	uf, codMunicipio
67	Uf	MODEL_C9V	C9V_UF	uf
68	Exterior	-		
69	paisResidencia	MODEL_C9V	C9V_PAIS	paisResidencia
70	descLogradouro	MODEL_C9V	C9V_LOGRAD	
71	nrLogradouro	MODEL_C9V	C9V_NRLOG	
72	Complemento	MODEL_C9V	C9V_COMLOG	
73	Bairro	MODEL_C9V	C9V_BAIRRO	
74	nomeCidade	MODEL_C9V	C9V_NOMCID	
75	codPostal	MODEL_C9V	C9V_CEP	
76	infoCasaPropria	-		
77	residenciaPropria	MODEL_C9V	C9V_RESPRP	
78	recursoFGTS	MODEL_C9V	C9V_RECFGT	
79	trabEstrangeiro	-		
80	dtChegada	MODEL_C9V	C9V_DTCHEG	
81	dtNaturalizacao	MODEL_C9V	C9V_DTNATU	
82	casadoBr	MODEL_C9V	C9V_CASBRA	
83	filhosBr	MODEL_C9V	C9V_FILBRA	
84	infBancarias	-		
85	Banco	MODEL_C9V	C9V_BANCO	
86	Agencia	MODEL_C9V	C9V_AGENC	
87	tpContaBancaria	MODEL_C9V	C9V_TPCTBA	
88	nrContaBancaria	MODEL_C9V	C9V_NRCTBA	
89	infoDeficiencia	-		
90	defMotora	MODEL_C9V	C9V_DEFMOT	
91	defVisual	MODEL_C9V	C9V_DEFVIS	
92	defAuditiva	MODEL_C9V	C9V_DEFAUD	
93	Reabilitado	MODEL_C9V	C9V_REABIL	
94	Observação	MODEL_C9V	C9V_OBSDEF	

95	Dependente	-		
96	tpDep	MODEL_C9Y	C9Y_TPDEP	
97	nomeDep	MODEL_C9Y	C9Y_NOMDEP	
98	dtNascto	MODEL_C9Y	C9Y_DTNASC	
99	cpfDep	MODEL_C9Y	C9Y_CPFDEP	
100	deplRRF	MODEL_C9Y	C9Y_DEPIRF	
101	depSF	MODEL_C9Y	C9Y_DEPSFA	
102	contato	-		
103	fonePrincipal	MODEL_C9V	C9V_FONPRC	
104	foneAlternativo	MODEL_C9V	C9V_FONALT	
105	emailPrincipal	MODEL_C9V	C9V_EMAILP	
106	emailAlternativo	MODEL_C9V	C9V_EMAILA	
107	Vinculo	-		
108	Matricula	MODEL_C9V	C9V_MATRIC	
109	dtAdmissao	MODEL_C9V	C9V_DTADMI	
110	tpAdmissao	MODEL_C9V	C9V_TPADMI	
111	indAdmissao	MODEL_C9V	C9V_INDADM	
112	indPrimeiroEmprego	MODEL_C9V	C9V_INDPRE	
113	Contrato	-		
114	tpRegimeTrab	MODEL_C9V	C9V_TPREGT	
115	tpRegimePrev	MODEL_C9V	C9V_TPREGP	
116	natAtividade	MODEL_C9V	C9V_NATATV	
117	codCateg	MODEL_C9V	C9V_CODCAT	codCateg
118	codCargo	MODEL_C9V	C9V_CODCGO	codCargo
119	codFuncao	MODEL_C9V	C9V_CODFUN	codFuncao
120	Remuneração	-		
121	vlrSalFixo	MODEL_C9V	C9V_VLSLFX	
122	unidSalFixo	MODEL_C9V	C9V_UNSLFX	
123	vlrSalVariavel	MODEL_C9V	C9V_VLSLVR	
124	unidSalVariavel	MODEL_C9V	C9V_UNSLVR	
125	Duração	-		
126	tpContrato	MODEL_C9V	C9V_TPCONT	
127	dtTermino	MODEL_C9V	C9V_DTTERM	
128	localTrabalho	-		
129	tpInscricao	MODEL_C9V	C9V_TPINSC	
130	nrInscricao	MODEL_C9V	C9V_NRINSC	
131	codLotacao	MODEL_C9V	C9V_CODLOT	codLotacao
132	descComplementar	MODEL_C9V	C9V_COMLOT	
133	Jornada	-		
134	jornadaPadrao	-		
135	qtdHoras	MODEL_C9V	C9V_QTDHJP	
136	codJornadaSegSexta	MODEL_C9V	C9V_CJORSS	codJornadaSegSexta
137	codJornadaSab	MODEL_C9V	C9V_CJORSA	codJornadaSab
138	jornadaTurnoFixo	-		
139	qtdHoras	MODEL_C9V	C9V_QTDHJF	
140	codJornadaSeg	MODEL_C9V	C9V_JORSEG	codJornadaSeg
141	codJornadaTer	MODEL_C9V	C9V_JORTER	codJornadaTer
142	codJornadaQua	MODEL_C9V	C9V_JORQUA	codJornadaQua
143	codJornadaQui	MODEL_C9V	C9V_JORQUI	codJornadaQui
144	codJornadaSex	MODEL_C9V	C9V_JORSEX	codJornadaSex
145	codJornadaSab	MODEL_C9V	C9V_JORSAB	codJornadaSab





```

//ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If nOpc <> 5

    oModel:LoadValue( "MODEL_C9V", "C9V_FILIAL", C9V->C9V_FILIAL )
    oModel:LoadValue( "MODEL_C9V", "C9V_CPF", "37400956876" )
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
If nOpc == 4 .and. lExisteC9Y
    For nJ := 1 to oModel:GetModel( "MODEL_C9Y" ):Length()
        oModel:GetModel( "MODEL_C9Y" ):GoLine(nJ)
        oModel:GetModel( "MODEL_C9Y" ):DeleteLine()
    Next nJ
EndIf

//ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Roda o XML parseado para gravar as novas informações no GRID³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
For nJ := 1 to nX
    If nOpc == 4 .or. nJ > 1
        oModel:GetModel( "MODEL_C9Y" ):lValid:= .T.
        oModel:GetModel( "MODEL_C9Y" ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL_C9Y", "C9Y_IDDEP", StrZero(nJ,6) )
    oModel:LoadValue( "MODEL_C9Y", "C9Y_NOMDEP", "Leandro")

Next nJ
EndIf

//ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()
EndIf
End Transaction

```

## S-2200 – Evento Admissão

<b>Rotina:</b>	TAF278.prw				
<b>Model MVC:</b>	MODEL_C9V				
<b>Validação</b>	TAF278VLD()				
<b>Geração XML</b>	TAF278XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetInt()</b>
	01	evtAdmissao	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
09	procEmi	-			



10	indSegmento	-		
11	verProc	-		
12	ideEmpregador	-		
13	tplInscricao	-		
14	nrInscricao	-		
15	Trabalhador	-		
16	cpfTrab	MODEL_C9V	C9V_CPF	
17	nisTrab	MODEL_C9V	C9V_NIS	
18	nomeTrab	MODEL_C9V	C9V_NOME	
19	Sexo	MODEL_C9V	C9V_SEXO	
20	racaCor	MODEL_C9V	C9V_RCCOR	
21	estadoCivil	MODEL_C9V	C9V_ESTCIV	
22	graulInstrucao	MODEL_C9V	C9V_GRINST	
23	Nascimento	-		
24	dtNascto	MODEL_C9V	C9V_DTNASC	
25	codMunicipio	MODEL_C9V	C9V_CODMUN	Uf, codMunicipio
26	Uf	MODEL_C9V	C9V_CODUF	Uf
27	paisNascto	MODEL_C9V	C9V_CODPAI	paisNascto
28	paisNacionalidade	MODEL_C9V	C9V_PAINAC	paisNacionalidade
29	nomeMae	MODEL_C9V	C9V_NOMMAE	
30	nomePai	MODEL_C9V	C9V_NOMPAI	
31	Documentos	-		
32	CTPS	-		
33	nrCtps	MODEL_C9V	C9V_NRCTPS	
34	serieCtps	MODEL_C9V	C9V_SERCTP	
35	ufCtps	MODEL_C9V	C9V_UFCTPS	ufCtps
36	RIC	-		
37	nrRic	MODEL_C9V	C9V_NRRIC	
38	orgaoEmissor	MODEL_C9V	C9V_OREMRI	
39	dtExpedicao	MODEL_C9V	C9V_DTEXRI	
40	RG	-		
41	nrRg	MODEL_C9V	C9V_NRRG	
42	orgaoEmissor	MODEL_C9V	C9V_OREMRG	
43	dtExpedicao	MODEL_C9V	C9V_DTEMRG	
44	RNE	-		
45	nrRne	MODEL_C9V	C9V_NRRNE	
46	orgaoEmissor	MODEL_C9V	C9V_OREMRN	
47	dtExpedicao	MODEL_C9V	C9V_DTEMRN	
48	OC	-		
49	nrOc	MODEL_C9V	C9V_NUMOC	
50	orgaoEmissor	MODEL_C9V	C9V_OREMOC	
51	dtExpedicao	MODEL_C9V	C9V_DTEXOC	
52	dtValidade	MODEL_C9V	C9V_DTVLOC	
53	CNH	-		
54	nrCnh	MODEL_C9V	C9V_NRCNH	
55	orgaoEmissor	MODEL_C9V	C9V_OREMCN	
56	dtExpedicao	MODEL_C9V	C9V_DTEXCN	
57	dtValidade	MODEL_C9V	C9V_DTVLCN	
58	Endereço	-		
59	Brasil	-		
60	tpLogradouro	MODEL_C9V	C9V_TPLOGR	tpLogradouro



61	descLogradouro	MODEL_C9V	C9V_LOGRAD	
62	nrLogradouro	MODEL_C9V	C9V_NRLOG	
63	Complemento	MODEL_C9V	C9V_COMLOG	
64	Bairro	MODEL_C9V	C9V_BAIRRO	
65	CEP	MODEL_C9V	C9V_CEP	
66	codMunicipio	MODEL_C9V	C9V_MUN	uf, codMunicipio
67	Uf	MODEL_C9V	C9V_UF	uf
68	Exterior	-		
69	paisResidencia	MODEL_C9V	C9V_PAIS	paisResidencia
70	descLogradouro	MODEL_C9V	C9V_LOGRAD	
71	nrLogradouro	MODEL_C9V	C9V_NRLOG	
72	Complemento	MODEL_C9V	C9V_COMLOG	
73	Bairro	MODEL_C9V	C9V_BAIRRO	
74	nomeCidade	MODEL_C9V	C9V_NOMCID	
75	codPostal	MODEL_C9V	C9V_CEP	
76	infoCasaPropria	-		
77	residenciaPropria	MODEL_C9V	C9V_RESPRP	
78	recursoFGTS	MODEL_C9V	C9V_RECFTG	
79	trabEstrangeiro	-		
80	dtChegada	MODEL_C9V	C9V_DTCHEG	
81	dtNaturalizacao	MODEL_C9V	C9V_DTNATU	
82	casadoBr	MODEL_C9V	C9V_CASBRA	
83	filhosBr	MODEL_C9V	C9V_FILBRA	
84	infBancarias	-		
85	Banco	MODEL_C9V	C9V_BANCO	
86	Agencia	MODEL_C9V	C9V_AGENC	
87	tpContaBancaria	MODEL_C9V	C9V_TPCTBA	
88	nrContaBancaria	MODEL_C9V	C9V_NRCTBA	
89	infoDeficiencia	-		
90	defMotora	MODEL_C9V	C9V_DEFMOT	
91	defVisual	MODEL_C9V	C9V_DEFVIS	
92	defAuditiva	MODEL_C9V	C9V_DEFAUD	
93	Reabilitado	MODEL_C9V	C9V_REABIL	
94	Observação	MODEL_C9V	C9V_OBSDEF	
95	Dependente	-		
96	tpDep	MODEL_C9Y	C9Y_TPDEP	
97	nomeDep	MODEL_C9Y	C9Y_NOMDEP	
98	dtNascto	MODEL_C9Y	C9Y_DTNASC	
99	cpfDep	MODEL_C9Y	C9Y_CPFDEP	
100	depIRRF	MODEL_C9Y	C9Y_DEPIRF	
101	depSF	MODEL_C9Y	C9Y_DEPSFA	
102	contato	-		
103	fonePrincipal	MODEL_C9V	C9V_FONPRC	
104	foneAlternativo	MODEL_C9V	C9V_FONALT	
105	emailPrincipal	MODEL_C9V	C9V_EMAILP	
106	emailAlternativo	MODEL_C9V	C9V_EMAILA	
107	Vinculo	-		
108	Matricula	MODEL_C9V	C9V_MATRIC	
109	dtAdmissao	MODEL_C9V	C9V_DTADMI	
110	tpAdmissao	MODEL_C9V	C9V_TPADMI	
111	indAdmissao	MODEL_C9V	C9V_INDADM	





112	indPrimeiroEmprego	MODEL_C9V	C9V_INDPRE	
113	Contrato	-		
114	tpRegimeTrab	MODEL_C9V	C9V_TPREGT	
115	tpRegimePrev	MODEL_C9V	C9V_TPREGP	
116	natAtividade	MODEL_C9V	C9V_NATATV	
117	codCateg	MODEL_C9V	C9V_CODCAT	codCateg
118	codCargo	MODEL_C9V	C9V_CODCGO	codCargo
119	codFuncao	MODEL_C9V	C9V_CODFUN	codFuncao
120	Remuneração	-		
121	vlrSalFixo	MODEL_C9V	C9V_VLSLFX	
122	unidSalFixo	MODEL_C9V	C9V_UNSLFX	
123	vlrSalVariavel	MODEL_C9V	C9V_VLSLVR	
124	unidSalVariavel	MODEL_C9V	C9V_UNSLVR	
125	Duração	-		
126	tpContrato	MODEL_C9V	C9V_TPCONT	
127	dtTermino	MODEL_C9V	C9V_DTTERM	
128	localTrabalho	-		
129	tpInscricao	MODEL_C9V	C9V_TPINSC	
130	nrlInscricao	MODEL_C9V	C9V_NRINSC	
131	codLotacao	MODEL_C9V	C9V_CODLOT	codLotacao
132	descComplementar	MODEL_C9V	C9V_COMLOT	
133	Jornada	-		
134	jornadaPadrao	-		
135	qtdHoras	MODEL_C9V	C9V_QTDHJP	
136	codJornadaSegSexta	MODEL_C9V	C9V_CJORSS	codJornadaSegSexta
137	codJornadaSab	MODEL_C9V	C9V_CJORSA	codJornadaSab
138	jornadaTurnoFixo	-		
139	qtdHoras	MODEL_C9V	C9V_QTDHJF	
140	codJornadaSeg	MODEL_C9V	C9V_JORSEG	codJornadaSeg
141	codJornadaTer	MODEL_C9V	C9V_JORTER	codJornadaTer
142	codJornadaQua	MODEL_C9V	C9V_JORQUA	codJornadaQua
143	codJornadaQui	MODEL_C9V	C9V_JORQUI	codJornadaQui
144	codJornadaSex	MODEL_C9V	C9V_JORSEX	codJornadaSex
145	codJornadaSab	MODEL_C9V	C9V_JORSAB	codJornadaSab
146	codJornadaDom	MODEL_C9V	C9V_JORDOM	codJornadaDom
147	jornadaTurnoFlexivel	-		
148	Turno	MODEL_C9V	C9V_CODTUR	
149	codJornada	MODEL_C9V	C9V_CODJTF	codJornada
150	jornadaEspecial	-		
151	tpEscala	MODEL_C9V	C9V_TPEJES	
152	qtdHoras	MODEL_C9V	C9V_QTDHJE	
153	codJornada	MODEL_C9V	C9V_CODJES	codJornada
154	filiacaoSindical	-		
155	cnpjSindTrabalhador	MODEL_C9V	C9V_CNPJSD	
156	alvaraJudicial	-		
157	nrProcJud	MODEL_C9V	C9V_ALVJUD	nrProcJud
158	FGTS	-		
159	optanteFGTS	MODEL_C9V	C9V_FGTS	
160	dtOpcaoFGTS	MODEL_C9V	C9V_DTFGTS	
161	sucessaoVinc	-		
162	cnpjEmpregadorAnterior	MODEL_C9V	C9V_CNPJEA	





```
oModel:LoadValue( "MODEL C9Y", "C9Y NOMDEP", "Leandro")

    Next nJ
EndIf

//ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAÿ
//³Efetiva a operação desejada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()
EndIf
End Transaction
```



## S-2220 – Alteração de Dados Cadastrais

<b>Rotina:</b>	TAF275				
<b>Model MVC:</b>	MODEL_CRA				
<b>Validação</b>	TAF275VLD()				
<b>Geração XML</b>	TAF275XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtAltCadastral	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	Alteração	-		
	16	dtAlteracao			
	17	dadosTrabalhador			
	18	cpfTrab	MODEL_CRA	CRA_CPF	
	19	nisTrab	MODEL_CRA	CRA_NIS	
	20	nomeTrab	MODEL_CRA	CRA_NOME	
	21	Sexo	MODEL_CRA	CRA_SEXO	
	22	racaCor	MODEL_CRA	CRA_RCCOR	
	23	estadoCivil	MODEL_CRA	CRA_ESTCIV	
	24	graulInstrucao	MODEL_CRA	CRA_GRINST	
	25	Nascimento	-		
	26	dtNascto	MODEL_CRA	CRA_DTNASC	
	27	codMunicipio	MODEL_CRA	CRA_CODMUN	Uf, codMunicipio
	28	Uf	MODEL_CRA	CRA_CODUF	Uf
	29	paisNascto	MODEL_CRA	CRA_CODPAI	paisNascto
	30	paisNacionalidade	MODEL_CRA	CRA_PAINAC	paisNacionalidade
	31	nomeMae	MODEL_CRA	CRA_NOMMAE	
	32	nomePai	MODEL_CRA	CRA_NOMPAI	
	33	Documentos	-		
	34	CTPS	-		
	35	nrCtps	MODEL_CRA	CRA_NRCTPS	
	36	serieCtps	MODEL_CRA	CRA_SERCTP	
	37	ufCtps	MODEL_CRA	CRA_UFCTPS	ufCtps
	38	RIC	-		
	39	nrRic	MODEL_CRA	CRA_NRRIC	
	40	orgaoEmissor	MODEL_CRA	CRA_OREMRI	
	41	dtExpedicao	MODEL_CRA	CRA_DTEXRI	
	42	RG	-		
43	nrRg	MODEL_CRA	CRA_NRRG		

44	orgaoEmissor	MODEL_CRA	CRA_OREMRG	
45	dtExpedicao	MODEL_CRA	CRA_DTEMRG	
46	RNE	-		
47	nrRne	MODEL_CRA	CRA_NRRNE	
48	orgaoEmissor	MODEL_CRA	CRA_OREMRN	
49	dtExpedicao	MODEL_CRA	CRA_DTEMRN	
50	OC	-		
51	nrOc	MODEL_CRA	CRA_NUMOC	
52	orgaoEmissor	MODEL_CRA	CRA_OREMOC	
53	dtExpedicao	MODEL_CRA	CRA_DTEXOC	
54	dtValidade	MODEL_CRA	CRA_DTVLOC	
55	CNH	-		
56	nrCnh	MODEL_CRA	CRA_NRCNH	
57	orgaoEmissor	MODEL_CRA	CRA_OREMCN	
58	dtExpedicao	MODEL_CRA	CRA_DTEXCN	
59	dtValidade	MODEL_CRA	CRA_DTVLCN	
60	Endereço	-		
61	Brasil	-		
62	tpLogradouro	MODEL_CRA	CRA_TPLOGR	tpLogradouro
63	descLogradouro	MODEL_CRA	CRA_LOGRAD	
64	nrLogradouro	MODEL_CRA	CRA_NRLOG	
65	Complemento	MODEL_CRA	CRA_COMLOG	
66	Bairro	MODEL_CRA	CRA_BAIRRO	
67	CEP	MODEL_CRA	CRA_CEP	
68	codMunicipio	MODEL_CRA	CRA_MUN	uf, codMunicipio
69	Uf	MODEL_CRA	CRA_UF	uf
70	Exterior	-		
71	paisResidencia	MODEL_CRA	CRA_PAIS	paisResidencia
72	descLogradouro	MODEL_CRA	CRA_LOGRAD	
73	nrLogradouro	MODEL_CRA	CRA_NRLOG	
74	Complemento	MODEL_CRA	CRA_COMLOG	
75	Bairro	MODEL_CRA	CRA_BAIRRO	
76	nomeCidade	MODEL_CRA	CRA_NOMCID	
77	codPostal	MODEL_CRA	CRA_CEP	
78	infoCasaPropria	-		
79	residenciaPropria	MODEL_CRA	CRA_RESRPP	
80	recursoFGTS	MODEL_CRA	CRA_RECFGT	
81	trabEstrangeiro	-		
82	dtChegada	MODEL_CRA	CRA_DTCHEG	
83	dtNaturalizacao	MODEL_CRA	CRA_DTNATU	
84	casadoBr	MODEL_CRA	CRA_CASBRA	
85	filhosBr	MODEL_CRA	CRA_FILBRA	
86	infBancarias	-		
87	Banco	MODEL_CRA	CRA_BANCO	
88	Agencia	MODEL_CRA	CRA_AGENC	
89	tpContaBancaria	MODEL_CRA	CRA_TPCTBA	
90	nrContaBancaria	MODEL_CRA	CRA_NRCTBA	
91	infoDeficiencia	-		
92	defMotora	MODEL_CRA	CRA_DEFMOT	
93	defVisual	MODEL_CRA	CRA_DEFVIS	
94	defAuditiva	MODEL_CRA	CRA_DEFAUD	





```
oModel:DeActivate()  
EndIf  
End Transaction
```

## S-2240 – Alteração Contratual

<b>Rotina:</b>	TAF276				
<b>Model MVC:</b>	MODEL_CRB				
<b>Validação</b>	TAF276VLD()				
<b>Geração XML</b>	TAF276XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtAltContratual	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CRB		
	17	nisTrab	MODEL_CRB		
	18	Matricula	MODEL_CRB		
	19	altContratual	-		
	20	dtAlteracao	MODEL_CRB	CRB_DTALTC	
	21	Contrato	-		
	22	tpRegimeTrab	MODEL_CRB	CRB_TPREGT	
	23	tpRegimePrev	MODEL_CRB	CRB_TPREGP	
	24	natAtividade	MODEL_CRB	CRB_NATATV	
	25	codCateg	MODEL_CRB	CRB_CODCAT	codCateg
	26	codCargo	MODEL_CRB	CRB_CODCGO	codCargo
	27	codFuncao	MODEL_CRB	CRB_CODFUN	codFuncao
	28	Remuneração	-		
	29	vlrSalFixo	MODEL_CRB	CRB_VLSLFX	
	30	unidSalFixo	MODEL_CRB	CRB_UNSLFX	
	31	vlrSalVariavel	MODEL_CRB	CRB_VLSLVR	
	32	unidSalVariavel	MODEL_CRB	CRB_UNSLVR	
	33	Duração	-		
	34	tpContrato	MODEL_CRB	CRB_TPCONT	
	35	dtTermino	MODEL_CRB	CRB_DTTERM	
	36	localTrabalho	-		
	37	tpInscricao	MODEL_CRB	CRB_TPINSC	
	38	nrInscricao	MODEL_CRB	CRB_NRINSC	
	39	codLotacao	MODEL_CRB	CRB_CODLOT	codLotacao
	40	descComplementar	MODEL_CRB	CRB_COMLOT	
	41	Jornada	-		
	42	jornadaPadrao	-		
43	qtdHoras	MODEL_CRB	CRB_QTDHJP		







## S-2260 – Comunicação de Acidente de Trabalho

<b>Rotina:</b>	TAF257				
<b>Model MVC:</b>	MODEL_CM0				
<b>Validação</b>	TAF257VLD()				
<b>Geração XML</b>	TAF257XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtCAT	-		
	03	Id	-		
	04	Versão	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tplInscricao	-		
	14	nrInscricao	-		
	15	ideTrabalhador	-		
	16	cpfTrab	MODEL_CM0	CM0_TRABAL	cpfTrab
	17	nisTrab	MODEL_CM0		
	18	Cat	-		
	19	dtAcidente	MODEL_CM0	CM0_DTACID	
	20	horaAcidente	MODEL_CM0	CM0_HRACID	
	21	horasTrabAntesAcidente	MODEL_CM0	CM0_HRTRAB	
	22	tpAcidente	MODEL_CM0	CM0_TPACID	
	23	tpCat	MODEL_CM0	CM0_TPCAT	
	24	indCatParcial	MODEL_CM0	CM0_INDPAR	
	25	indCatObito	MODEL_CM0	CM0_INDOBI	
	26	indComunicPolicia	MODEL_CM0	CM0_COMPOL	
	27	codSitGeradora	MODEL_CM0	CM0_CODSIT	
	28	localAcidente	-		
	29	tpLocal	MODEL_CM0	CM0_TPLOC	
	30	descLocal	MODEL_CM0	CM0_DESLOC	
	31	descLogradouro	MODEL_CM0	CM0_DESLOG	
	32	nrLogradouro	MODEL_CM0	CM0_NRLOG	
	33	codMunicipio	MODEL_CM0	CM0_CODMUN	Uf,codMunicipio
	34	Uf	MODEL_CM0	CM0_UF	uf
	35	cnpjLocalAcidente	MODEL_CM0	CM0_CNPJLO	
	36	parteAtingida	-		
	37	codParteAtingida	MODEL_CM1	CM1_CODPAR	codParteAtingida
	38	agenteCausador	-		
	39	codAgenteCausador	MODEL_CM2	CM2_CODAGE	codAgenteCausador
	40	Testemunha	-		
	41	nomeTestemunha	MODEL_CM3	CM3_NOMTES	
	42	descLogradouro	MODEL_CM3	CM3_DESLOG	
43	nrLogradouro	MODEL_CM3	CM3_NRLOG		





```

        If nOpc == 4 .Or. nJ > 1
            oModel:GetModel( 'MODEL CM1' ):AddLine()
        EndIf

oModel:LoadValue( "MODEL_CM1", "CM1_CODPAR", "00001")
    EndDo

//³Agente Causador³
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU

If nOpc == 4 .And. lAgCausador
    For nJ := 1 to oModel:GetModel( 'MODEL_CM2' ):Length()
        oModel:GetModel( 'MODEL CM2' ):GoLine(nJ)
        oModel:GetModel( 'MODEL CM2' ):DeleteLine()
    Next nJ
EndIf
    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Roda o XML parseado para gravar as novas informações no GRID³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU
For nJ := 1 to nAgCausador
    oModel:GetModel( 'MODEL_CM2' ):LVALID := .T.

    If nOpc == 4 .Or. nJ > 1
        oModel:GetModel( 'MODEL_CM2' ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL_CM2", "CM2_CODAGE", "0001" )
Next nJ
//ÚAAAAAAAAAAAAAAAAAAA;
//³Testemunhas
//³
//AAAAAAAAAAAAAAAAAAA
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU
    If nOpc == 4 .And. lTestemun
        For nJ := 1 to oModel:GetModel( 'MODEL_CM3' ):Length()
            oModel:GetModel( 'MODEL_CM3' ):GoLine(nJ)
            oModel:GetModel( 'MODEL_CM3' ):DeleteLine()
        Next nJ
    EndIf
    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Roda o XML parseado para gravar as novas informações no GRID³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU
For nJ := 1 to nTestemun
    oModel:GetModel( 'MODEL CM3' ):LVALID := .T.
    If nOpc == 4 .Or. nJ > 1
        oModel:GetModel( 'MODEL CM3' ):AddLine()
    EndIf

    oModel:LoadValue( "MODEL CM3", "CM3 NOMTES" , "Gabriela")
    oModel:LoadValue( "MODEL CM3", "CM3 FONE" , "955441100" )
Next nJ

EndIf
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
FWFormCommit( oModel )
lRet := .T.
oModel:DeActivate()

EndIf
End Transaction

```



## S-2280 – Atestado de Saúde Ocupacional

<b>Rotina:</b>	TAF258.PRW				
<b>Model MVC:</b>	MODEL_C8B; MODEL_C9W; MODEL_C9X; MODEL_CRP;				
<b>Validação</b>	TAF258VLD()				
<b>Geração XML</b>	TAF258XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtASO	-		
	03	Versao	-		
	04	Id	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_C8B	C8B_FUNC	cpfTrab
	17	nisTrab	-		
	18	Matricula	-		
	19	aso	-		
	20	dtAso	MODEL_C8B	C8B_DTASO	
	21	tpAso	MODEL_C8B	C8B_TPASO	
	22	resultadoAso	MODEL_C8B	C8B_RESULT	
	23	exame	-		
	24	dtExame	MODEL_C9W	C9W_DTEXAM	
	25	descExame	MODEL_C9W	C9W_DESEXM	
	26	Risco	-		
	27	codAgente	MODEL_C9X	C9X_CODAGE	codAgente
	28	monitBiologica	-		
	29	resultMonitoracao	-		
	30	codAgenteQuimico	MODEL_CRP	CRP_IDMBIO	
	31	matBiologico	-	-	
	32	codAnalise	-	-	
	33	expExcessiva	MODEL_CRP	CRP_EXPEXC	
	34	ordemExame	MODEL_CRP	CRP_ORDEXA	
	35	indResultado	MODEL_CRP	CRP_INDRES	
	36	dtIniMonitoracao	MODEL_CRP	CRP_DINIMO	
	37	dtFimMonitoracao	MODEL_CRP	CRP_DFIMMO	
	38	Medico	-		
	39	nomeMedico	-	-	
	40	foneContato	-	-	
	41	CRM	-	-	
	42	nrCRM	MODEL_C8B	C8B_CODMED	nrOC
43	ufCRM	-	-		

**Exemplo:**

```
AAdd( aChave, { "C", "C8B_FUNC", FGetIdInt( "cpfTrab", "", "37400956876" ), .T. } )
aAdd( aChave, { "D", "C8B_DTASO", "2013-11-07", .t. } )
```

**Begin** Transaction

```
  If FTafVldOpe( "C8B", 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, "TAFA258", cCmpsNoUpd )
```

```
    oModel:LoadValue( "MODEL C8B", "C8B FILIAL", C8B->C8B FILIAL )
    oModel:LoadValue( "MODEL C8B", "C8B DTASO", "2013-11-07" ) )
```

```
  // ³ Risco ³
```

```
  If nOpc == 4
```

```
    For nJ := 1 to oModel:GetModel( 'MODEL_C9X' ):Length()
```

```
      oModel:GetModel( 'MODEL_C9X' ):GoLine(nJ)
```

```
      oModel:GetModel( 'MODEL_C9X' ):DeleteLine()
```

```
    Next nJ
```

```
  EndIf
```

```
  For nJ := 1 to nRiscos
```

```
    oModel:GetModel( 'MODEL_C9X' ):LVALID := .T.
```

```
    If nOpc == 4 .Or. nJ > 1
```

```
      oModel:GetModel( 'MODEL_C9X' ):AddLine()
```

```
    EndIf
```

```
    oModel:LoadValue( "MODEL_C9X", "C9X CODAGE", "001"
```

```
  Next nJ
```

```
  //Exame
```

```
  If nOpc == 4
```

```
    For nJ := 1 to oModel:GetModel( 'MODEL_C9W' ):Length()
```

```
      oModel:GetModel( 'MODEL_C9W' ):GoLine(nJ)
```

```
      oModel:GetModel( 'MODEL_C9W' ):DeleteLine()
```

```
    Next nJ
```

```
  EndIf
```

```
  For nJ := 1 to nExames
```

```
    oModel:GetModel( 'MODEL_C9W' ):LVALID := .T.
```

```
    If nOpc == 4 .Or. nJ > 1
```

```
      oModel:GetModel( 'MODEL_C9W' ):AddLine()
```

```
    EndIf
```

```
    oModel:LoadValue( "MODEL_C9W", "C9W_DTEXAM", "2013-10-10"
```

```
  Next nJ
```

```
  For nJ := 1 to nExames
```

```
    oModel:GetModel( 'MODEL_C9W' ):LVALID := .T.
```

```
    If nOpc == 4 .Or. nJ > 1
```

```
      oModel:GetModel( 'MODEL_C9W' ):AddLine()
```

```
    EndIf
```

```
  cIdCUA := FGetIdInt( "codAgenteQuimico", "", "01"
```

```
  cIdCUM := FGetIdInt( "codAnalise", "", "1.1"
```

```
  cMatBio:= "1"
```

```
  ("CUQ")->(DBSetOrder(3))
```

```
  If ("CUQ")->(MsSeek(xFilial("CUQ")+ cIdCUA + cMatBio + cIdCUM ))
```

```
    oModel:LoadValue( "MODEL_CRP", "CRP_IDMBIO", CUQ->CUQ_ID )
```

```
  EndIf
```

```
  oModel:LoadValue( "MODEL_CRP", "CRP_EXPEXC", "1"
```

```
  oModel:LoadValue( "MODEL_CRP", "CRP_ORDEXA ", "1"
```

```
  cIdCUA := ""
```

```
  cIdCUM := ""
```

```
  cMatBio:= ""
```

```
  Next nJ
```

```
  FWFormCommit( oModel )
```

```
  lRet := .T.
```

```
  oModel:DeActivate()
```

```
  EndIf
```

```
End Transaction
```



## S-2320 – Afastamento Temporário

<b>Rotina:</b>	TAF261.PRW				
<b>Model MVC:</b>	MODEL_CM6				
<b>Validação</b>	TAF261VLD()				
<b>Geração XML</b>	TAF261XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtAfastTemp	-		
	03	id	-		
	04	versao	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CM6	CM6_FUNC	Matricula, cpfTrab
	17	nisTrab	-		
	18	matricula	MODEL_CM6		
	19	infoAfastamento	-		
	20	dtAfastamento	MODEL_CM6	CM6_DTAFAS	
	21	codMotAfastamento	MODEL_CM6	CM6_MOTVAF	
	22	tpAcidenteTransito	MODEL_CM6	CM6_TPACID	
	23	observação	MODEL_CM6	CM6_OBSERV	
	24	infoAtestado	-		
	25	codCID	MODEL_CM6	CM6_CODCID	codCID
	26	qtdDiasAfastamento	MODEL_CM6	CM6_DIASAF	
	27	emitente	-		
	28	nomeEmitente	MODEL_CM6	CM6_MEDICO	nrCRM
	29	nrOC	-		
	30	ufOC	-		
	31	infoCessao	-		
	32	cnpjCessionario	MODEL_CM6	CM6_CNPJCE	
	33	infoOnus	MODEL_CM6	CM6_INFOCE	
	34	infoMandadoSindical	-		
	35	cnpjSindicato	MODEL_CM6	CM6_CNPJSD	
36	infoOnusRemuneracao	MODEL_CM6	CM6_INFOSD		



## Exemplo:

```

Add( aChave, {"C", "CM6_FUNC" , FGetIdInt( "cpfTrab", , "37400956876" ) , .T.} )
Add( aChave, {"C", "CM6_MOTVAF", FGetIdInt( "codMotAfastamento", "", "0001" ) , .T. } )

Begin Transaction
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If FTafVldOpe( 'CM6', 2, @nOpc, cFilEv, @aIncons, aChave, @oModel, 'TAFA261', cCmpsNoUpd )

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If nOpc <> 5

oModel:LoadValue( "MODEL_CM6", "CM6_FUNC", CM6->CM6_FUNC)
oModel:LoadValue( "MODEL_CM6", "CM6_DTAFAS", CM6->CM6_DTAFAS)
oModel:LoadValue( "MODEL_CM6", "CM6_MOTVAF", CM6->CM6_MOTVAF)
oModel:LoadValue( "MODEL_CM6", "CM6_FUNC" , FGetIdInt( "cpfTrab",
"37400956876", "MAT0001" ) )

Next

EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate ()

EndIf
End Transaction

```







```
//³Efetiva a operação desejada³  
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAU  
FWFormCommit( oModel )  
lRet := .T.  
  
oModel:DeActivate()  
  
EndIf  
End Transaction
```





```

For nCRJ := 1 to
If (nCRJ > 1) .Or. (nOpc == 4)
    // é necessário atribuir LVALID para que permita o addline();
    oModel:GetModel( "MODEL_CRJ" ):LVALID := .T.

    oModel:GetModel( "MODEL_CRJ" ):AddLine()

EndIf
//Começa a salvar as informações;
oModel:LoadValue( "MODEL_CRJ", "CRJ_CODAGE", FGetIdInt( "codAgente", "00001"))
    Next nCRJ

EndIf

//ÛAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction
    
```

## S-2340 – Estabilidade – Início

<b>Rotina:</b>	TAF260.PRW				
<b>Model MVC:</b>	MODEL_CM5				
<b>Validação</b>	TAF260VLD()				
<b>Geração XML</b>	TAF260XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtEstabInicio	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CM5	CM5_TRABAL	cpfTrab,matricula
	17	nisTrab	-		
	18	matricula	-		
	19	infoEstabInicio	-		
	20	dtIniEstabilidade	MODEL_CM5	CM5_DTAFAS	
	21	codMotivoEstabilidade	MODEL_CM5	CM5_CODEST	
22	observacao	MODEL_CM5	CM5_OBSERV		







## S-2360 – Condição Diferenciada de Trabalho – Início

<b>Rotina:</b>	TAF264.PRW				
<b>Model MVC:</b>	MODEL_CM9; MODEL_CMA; MODEL_CMB				
<b>Validação</b>	TAF264VLD()				
<b>Geração XML</b>	TAF264XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtCDTInicio	-		
	03	id	-		
	04	versao	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CM9	CM9_FUNC	cpfTrab, matricula
	17	nisTrab	-		
	18	matricula	-		
	19	infoCDTInicio	-		
	20	dtIniCondicao	MODEL_CM9	CM9_DTINI	
	21	tpCondicao	MODEL_CM9	CM9_TPCOND	tpCondicao
	22	fatoresRisco	-		
	23	codAgente	MODEL_CMA	CMA_CODAGE	codAgente
	24	intensidConcentracao	MODEL_CMA	CMA_INTCON	
	25	tecMedicacao	MODEL_CMA	CMA_TECMED	
	26	utilizacaoEPC	MODEL_CMA	CMA_UTLEPC	
	27	utilizacaoEPI	MODEL_CMA	CMA_UTLEPI	
	28	epi	-		
	29	caEPI	MODEL_CMB	CMB_CAEPi	
	30	requisitosEPI	-		
	31	medProtecao	MODEL_CM9	CM9_MEDPRT	
	32	condFuncionamento	MODEL_CM9	CM9_CNDFUN	
	33	prazoValidade	MODEL_CM9	CM9_PRZVLD	
	34	periodicTroca	MODEL_CM9	CM9_PERTRC	
35	Higienização	MODEL_CM9	CM9_HIGIEN		

### Exemplo:

```

Aadd( aChave, { "C", "CM9_NOMEVE", 'S2360', .T. } )
Aadd( aChave, { "C", "CM9_FUNC", FGetIdInt( "cpfTrab", "matricula", "37400956876", "MAT00001" ), .T. } )
Aadd( aChave, { "D", "CM9_DTINI", "2013-10-10", .F. } )
Aadd( aChave, { "C", "CM9_TPCOND", "1", .F. } )
nIndice := 2
    
```



```

Begin Transaction
//Função para validar se a operação desejada pode ser realizada
  If FTafVldOpe( 'CM9', nIndice, @nOpc, cFilEv, @aIncons, aChave, @oModel, 'TAFa264', cCmpsNoUpd,1 )

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  If nOpc <> 5
    oModel:LoadValue( "MODEL_CM9", "CM9_FILIAL", CM9->CM9_FILIAL )
    //Grava o tipo de registro, indicando se é condição diferenciada de trabalho - início /
    término.
    oModel:LoadValue( "MODEL_CM9", "CM9_NOMEVE", 'S2360' )
    oModel:LoadValue( "MODEL_CM9", "CM9_TPCOND", "1" )
    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    //³Quando se trata de uma alteração deleta-se todas as linhas do Grid³
    //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    If (nOpc == 4)
      For nJ := 1 to oModel:GetModel( 'MODEL_CMA' ):Length()
        oModel:GetModel( 'MODEL_CMA' ):GoLine(nJ)
        oModel:GetModel( 'MODEL_CMA' ):DeleteLine()
      Next nJ
    EndIf

    For nJ := 1 to nCMA
      If (nJ > 1) .Or. (nOpc == 4)
        // é necessário atribuir LVALID para que permita o addline();
        oModel:GetModel( "MODEL_CMA" ):LVALID := .T.

        oModel:GetModel( "MODEL_CMA" ):AddLine()
      EndIf
      //Começa a salvar as informações;
      oModel:LoadValue( "MODEL_CMA", "CMA_UTLEPI", "S")
      // CMB
      If nOpc == 4
        For nI := 1 to oModel:GetModel( 'MODEL_CMB' ):Length()
          oModel:GetModel( 'MODEL_CMB' ):GoLine(nJ)
          oModel:GetModel( 'MODEL_CMB' ):DeleteLine()
        Next nI
      EndIf

      For nI := 1 to nCMB
        If (nCMB > 1) .Or. (nOpc == 4)
          oModel:GetModel( "MODEL_CMB" ):LVALID := .T.
          oModel:GetModel( "MODEL_CMB" ):AddLine()
        EndIf
        oModel:LoadValue( "MODEL_CMB", "CMB_CAEPI", "000001" )
      Next nI
    Next nJ

    EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU
    FWFormCommit( oModel )
    lRet := .T.

    oModel:DeActivate()
  EndIf
End Transaction

```









## S-2400 – Aviso Prévio

<b>Rotina:</b>	TAF263.PRW				
<b>Model MVC:</b>	MODEL_CM8				
<b>Validação</b>	TAF263VLD()				
<b>Geração XML</b>	TAF263XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtAvisoPrevio	-		
	03	id	-		
	04	versao	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CM8	CM8_TRABAL	Matricula,cpfTrab
	17	nisTrab	-	-	
	18	Matricula	-	-	
	19	infoAvisoPrevio	-		
	20	dtAvisoPrevio	MODEL_CM8	CM8_DTAVIS	
	21	dtProjAfastamento	MODEL_CM8	CM8_DTAFAS	
	22	tpAvisoPrevio	MODEL_CM8	CM8_TPAVIS	
23	Observação	MODEL_CM8	CM8_OBSERV		

### Exemplo:

```

Aadd( aChave, { "C", "CM8_TRABAL"      , FGetIdInt( "cpfTrab", "", "37400956876" ), .T. } )
aAdd( aChave, { "D", "CM8_DTAVIS"     , "2013-10-10", .T. } )

Begin Transaction
  If FTafVldOpe( "CM8", 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, "TAF263", cCmpsNoUpd )
    oModel:LoadValue( "MODEL_CM8", "CM8_FILIAL", CM8->CM8_FILIAL )
    oModel:LoadValue( "MODEL_CM8", "CM8_DTAVIS", "2013-10-10" )

    /*ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
      ³Efetiva a operação desejada³
      ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ*/
    FWFormCommit( oModel )
    lRet := .T.
    oModel:DeActivate ()
  EndIf
End Transaction

```



## S-2405 – Cancelamento do Aviso Prévio

<b>Rotina:</b>	TAF286.PRW				
<b>Model MVC:</b>	MODEL_CRG				
<b>Validação</b>	TAF286VLD()				
<b>Geração XML</b>	TAF286XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtCancAvisoPrevio	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CRG	CRG_TRABAL	dtCancAvisoPrevio
	17	nisTrab	-	-	
	18	Matricula	-	-	
	19	infoCancAvisoPrevio	-		
	20	dtCancAvisoPrevio	MODEL_CRG	CRG_DTCANC	
	21	Observação	MODEL_CRG	CRF_OBSRET	
22	motivoCancAvisoPrevio	MODEL_CRG	CRG_MOTCAN		

### Exemplo:

```

Aadd( aChave, { 'C', 'CRG_TRABAL', FGetIdInt( "cpfTrab", "", "37400956876" ), .T. } )
aAdd( aChave, { 'C', 'CRG_IDAVIS', FGetIdInt( "dtCancAvisoPrevio", "", cIdFun,,.F.), .T. } )

Begin Transaction
  If FTafVldOpe( 'CRG', 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, 'TAF286', cCmpsNoUpd )
    oModel:LoadValue( "MODEL_CRG", "CRG_FILIAL", CRG->CRG_FILIAL )
    oModel:LoadValue( "MODEL_CRG", "CRG_OBSERV", "Observacao..")
    *Efetiva a operação desejada*
    FWFormCommit( oModel )
    lRet := .T.
    oModel:DeActivate()
  EndIf
End Transaction
    
```



## S-2600 – Trabalhador Sem Vínculo de Emprego - Início

<b>Rotina:</b>	TAF279.PRW				
<b>Model MVC:</b>	MODEL_C9V				
<b>Validação</b>	TAF279VLD()				
<b>Geração XML</b>	TAF279XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtTSVInicio	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	Trabalhador	-		
	16	cpfTrab	MODEL_C9V	C9V_CPF	
	17	nisTrab	MODEL_C9V	C9V_NIS	
	18	nomeTrab	MODEL_C9V	C9V_NOME	
	19	sexo	MODEL_C9V	C9V_SEXO	
	20	racaCor	MODEL_C9V	C9V_RCCOR	
	21	estadoCivil	MODEL_C9V	C9V_ESTCIV	
	22	grauInstrucao	MODEL_C9V	C9V_GRINST	
	23	Nascimento	-		
	24	dtNascto	MODEL_C9V	C9V_DTNASC	
	25	codMunicipio	MODEL_C9V	C9V_CODMUN	
	26	Uf	MODEL_C9V	C9V_CODUF	
	27	paisNascto	MODEL_C9V	C9V_CODPAI	
	28	paisNacionalidade	MODEL_C9V	C9V_PAINAC	
	29	nomeMae	MODEL_C9V	C9V_NOMMAE	
	30	nomePai	MODEL_C9V	C9V_NOMPAI	
	31	Documentos	-		
	32	CTPS	-		
	33	nrCtps	MODEL_C9V	C9V_NRCTPS	
	34	serieCtps	MODEL_C9V	C9V_SERCTP	
	35	ufCtps	MODEL_C9V	C9V_UFCTPS	
	36	RIC	-		
	37	nrRIC	MODEL_C9V	C9V_NRRIC	
	38	orgaoEmissor	MODEL_C9V	C9V_OREMRI	
	39	dtExpedicao	MODEL_C9V	C9V_DTEXRI	
	40	RG	-		
	41	nrRg	MODEL_C9V	C9V_NRRG	
	42	orgaoEmissor	MODEL_C9V	C9V_OREMRG	
43	dtExpedicao	MODEL_C9V	C9V_DTEMRG		



44	RNE	-	
45	nrRne	MODEL_C9V	C9V_NRRNE
46	orgaoEmissor	MODEL_C9V	C9V_OREMRN
47	dtExpedicao	MODEL_C9V	C9V_DTEMRN
48	OC	-	
49	nrOc	MODEL_C9V	C9V_NUMOC
50	orgaoEmissor	MODEL_C9V	C9V_OREMOC
51	dtExpedicao	MODEL_C9V	C9V_DTEXOC
52	dtValidade	MODEL_C9V	C9V_DTVLOC
53	CNH	-	
54	nrCnh	MODEL_C9V	C9V_NRCNH
55	orgaoEmissor	MODEL_C9V	C9V_OREMCN
56	dtExpedicao	MODEL_C9V	C9V_DTEXCN
57	dtValidade	MODEL_C9V	C9V_DTVLCN
58	Endereço	-	
59	Brasil	-	
60	tpLogradouro	MODEL_C9V	C9V_TPLOGR
61	descLogradouro	MODEL_C9V	C9V_LOGRAD
62	nrLogradouro	MODEL_C9V	C9V_NRLOG
63	Complemento	MODEL_C9V	C9V_COMLOG
64	Bairro	MODEL_C9V	C9V_BAIRRO
65	CEP	MODEL_C9V	C9V_CEP
66	codMunicipio	MODEL_C9V	C9V_MUN
67	Uf	MODEL_C9V	C9V_UF
68	Exterior	-	
69	paisResidencia	MODEL_C9V	C9V_PAIS
70	descLogradouro	MODEL_C9V	C9V_LOGRAD
71	nrLogradouro	MODEL_C9V	C9V_NRLOG
72	Complemento	MODEL_C9V	C9V_COMLOG
73	Bairro	MODEL_C9V	C9V_BAIRRO
74	nomeCidade	MODEL_C9V	C9V_NOMCID
75	codPostal	MODEL_C9V	C9V_CEP
76	infoCasaPropria	-	
77	residenciaPropria	MODEL_C9V	C9V_RESRPP
78	recursoFGTS	MODEL_C9V	C9V_RECFTG
79	trabEstrangeiro	-	
80	dtChegada	MODEL_C9V	C9V_DTCHEG
81	dtNaturalizacao	MODEL_C9V	C9V_DTNATU
82	casadoBr	MODEL_C9V	C9V_CASBRA
83	filhosBr	MODEL_C9V	C9V_FILBRA
84	infBancarias	-	
85	Banco	MODEL_C9V	C9V_BANCO
86	Agencia	MODEL_C9V	C9V_AGENC
87	tpContaBancaria	MODEL_C9V	C9V_TPCTBA
88	nrContaBancaria	MODEL_C9V	C9V_NRCTBA
89	infoDeficiencia	-	
90	defMotora	MODEL_C9V	C9V_DEFMOT
91	defVisual	MODEL_C9V	C9V_DEFVIS
92	defAuditiva	MODEL_C9V	C9V_DEFAUD
93	Reabilitado	MODEL_C9V	C9V_REABIL
94	Observação	MODEL_C9V	C9V_OBSDEF



95	Dependente	-		
96	tpDep	MODEL_C9Y	C9Y_TPDEP	
97	nomeDep	MODEL_C9Y	C9Y_NOMDEP	
98	dtNascto	MODEL_C9Y	C9Y_DTNASC	
99	cpfDep	MODEL_C9Y	C9Y_CPFDEP	
100	deplRRF	MODEL_C9Y	C9Y_DEPIRF	
101	depSF	MODEL_C9Y	C9Y_DEPSFA	
102	contato	-		
103	fonePrincipal	MODEL_C9V	C9V_FONPRC	
104	foneAlternativo	MODEL_C9V	C9V_FONALT	
105	emailPrincipal	MODEL_C9V	C9V_EMAILP	
106	emailAlternativo	MODEL_C9V	C9V_EMAILA	
107	infoTSVInicio	-		
108	trabalhadorAvulso	-		
109	dtInicio	MODEL_C9V	C9V_DTINAV	
110	codCateg	MODEL_C9V	C9V_CATAV	codCateg
111	FGTS	-		
112	optanteFGTS	MODEL_C9V	C9V_OPFGAV	
113	dtOpcaoFGTS	MODEL_C9V	C9V_DTFGAV	
114	contribIndividual	-		
115	codCateg	MODEL_C9V	C9V_CATCI	codCateg
116	dtInicio	MODEL_C9V	C9V_DTINCI	
117	codCargo	MODEL_C9V	C9V_CARCI	codCargo
118	codFuncao	MODEL_C9V	C9V_FUNCI	codFuncao
119	FGTS	-		
120	optanteFGTS	MODEL_C9V	C9V_OPFGCI	
121	dtOpcaoFGTS	MODEL_C9V	C9V_DTFGCI	
122	servPubIndConselho	-		
123	dtInicio	MODEL_C9V	C9V_DTINSP	
124	codCateg	MODEL_C9V	C9V_CATSP	codCateg
125	codCargo	MODEL_C9V	C9V_CARSP	codCargo
126	codFuncao	MODEL_C9V	C9V_FUNSP	codFuncao
127	dirigenteSindical	-		
128	dtInicio	MODEL_C9V	C9V_DTINDS	
129	codCateg	MODEL_C9V	C9V_CATDS	codCateg
130	-	MODEL_C9V		
131	categOrigem	MODEL_C9V	C9V_CATODS	categOrigem
132	cnpjOrigem	MODEL_C9V	C9V_CNPJDS	
133	dtAdmissaoOrigem	MODEL_C9V	C9V_DTADDS	
134	matricOrigem	MODEL_C9V	C9V_MATODS	
135	FGTS	-		
136	optanteFGTS	MODEL_C9V	C9V_OPFGDS	
137	dtOpcaoFGTS	MODEL_C9V	C9V_DTFGDS	
138	Estagiário	-		
139	dtInicio	MODEL_C9V	C9V_DTINES	
140	codCateg	MODEL_C9V	C9V_CATES	codCateg
141	natEstagio	MODEL_C9V	C9V_NATEES	
142	nivEstagio	MODEL_C9V	C9V_NIVEES	
143	areaAtuacao	MODEL_C9V	C9V_AREAES	
144	nrApolice	MODEL_C9V	C9V_NRAPES	
145	vlrBolsa	MODEL_C9V	C9V_VLBLES	







```

                                oModel:GetModel( "MODEL C9Y" ):AddLine()
                EndIf
                oModel:LoadValue( "MODEL_C9Y", "C9Y_IDDEP" , StrZero(nJ,6) )
                oModel:LoadValue( "MODEL_C9Y", "C9Y_NOMDEP", "Fernanda")
            Next nJ
        EndIf

        //ÚÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁ;
        //³Efetiva a operação desejada³
        //ÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÁÙ
        FWFormCommit( oModel )
        lRet := .T.

        oModel:DeActivate()
    EndIf
End Transaction
    
```



## S-2620 – Trabalhador Sem Vínculo de Emprego – Alteração Contratual

<b>Rotina:</b>	TAF277.PRW				
<b>Model MVC:</b>	MODEL_CRC				
<b>Validação</b>	TAF277VLD()				
<b>Geração XML</b>	TAF277XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtTSVAltContratual	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	Trabalhador	-		
	16	cpfTrab	MODEL_CRC	CRC_CODTRA	Matricula,cpfTrab
	17	nisTrab	-	-	
	18	dtAlteracao	MODEL_CRC	CRC_DALTSV	
	19	contribIndividual	-		
	20	codCateg	MODEL_CRC	CRC_CATCI	codCateg
	21	codCargo	MODEL_CRC	CRC_CARCI	codCargo
	22	codFuncao	MODEL_CRC	CRC_FUNCI	codFuncao
	23	servPublndConselho			
	24	codCargo	MODEL_CRC	CRC_CARSP	codCargo
	25	codFuncao	MODEL_CRC	CRC_FUNSP	codFuncao
	26	Estagiário	-		
	27	natEstagio	MODEL_CRC	CRC_NATEES	
	28	nivEstagio	MODEL_CRC	CRC_NIVEES	
	29	areaAtuacao	MODEL_CRC	CRC_AREAES	
	30	nrApolice	MODEL_CRC	CRC_NRAPES	
	31	vlrBolsa	MODEL_CRC	CRC_VLBLES	
	32	dtPrevistaTermino	MODEL_CRC	CRC_DTTEES	
	33	instEnsino	-		
	34	cnpjInstRazao	MODEL_CRC	CRC_CNPEES	
	35	nomeRazao	MODEL_CRC	CRC_NOMEES	
	36	descLogradouro	MODEL_CRC	CRC_LOGEES	
	37	nrLogradouro	MODEL_CRC	CRC_NLGEES	
	38	Bairro	MODEL_CRC	CRC_BAREES	
	39	CEP	MODEL_CRC	CRC_CEPES	
	40	codMunicipio	MODEL_CRC	CRC_MUNES	Uf,codMunicipio
	41	Uf	MODEL_CRC	CRC_UFES	uf
	42	agelIntegracao	-		
43	cnpjAgenteIntegracao	MODEL_CRC	CRC_CNPAES		





## S-2680 – Trabalhador Sem Vínculo de Emprego – Término

<b>Rotina:</b>	TAF280.PRW				
<b>Model MVC:</b>	MODEL_CRD, MODEL_CMK				
<b>Validação</b>	TAF280VLD()				
<b>Geração XML</b>	TAF280XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtTSVTermino	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	Trabalhador	-		
	16	cpfTrab	MODEL_CRD	CRD_CODTRA	Matricula,cpfTrab
	17	nisTrab	-	-	
	18	trabalhadorAvulso	-		
	19	dtTermino	MODEL_CRD	CRD_DTERAV	
	20	contribIndividual			
	21	codCateg	MODEL_CRD	CRD_CATCI	codCateg
	22	dtTermino	MODEL_CRD	CRD_DTERCI	
	23	tpDesligDiretorNaoEmpregado	MODEL_CRD	CRD_TPDDCI	
	24	verbasRescisorias	-		
	25	bcCP	MODEL_CRD	CRD_BCCPCI	
	26	bcIRRF	MODEL_CRD	CRD_BCIRCI	
	27	bcFGTS	MODEL_CRD	CRD_BCFGCI	
	28	bcFGTSVerbasIndeniz	MODEL_CRD	CRD_BCVICI	
	29	bcFgtsMesAnt	MODEL_CRD	CRD_BCMACI	
	30	descCP	MODEL_CRD	CRD_VDCPCI	
	31	vlrProventos	MODEL_CRD	CRD_VPRVCI	
	32	vlrDescontos	MODEL_CRD	CRD_VDESCI	
	33	vlrLiquido	MODEL_CRD	CRD_VLIQCI	
	34	itensRemuneracao	-		
	35	codRubrica	MODEL_CMK	CMK_CODRUB	codRubrica
	36	qtdRubrica	MODEL_CMK	CMK_QTDRUB	
	37	vlrRubrica	MODEL_CMK	CMK_VLRRUB	
	38	servPubIndConselho	-		
	39	dtTermino	MODEL_CRD	CRD_DTTESP	
	40	dirigenteSindical	-		
	41	dtTermino	MODEL_CRD	CRD_DTTEDS	
	42	Estagiário	-		
43	dtTermino	MODEL_CRD	CRD_DTTEES		

**Exemplo:**

```

aAdd( aChave, {"C", "CRD_CODTRA", FGetIdInt( "cpfTrab" , "", "37400956876" ), .T. } )

Begin Transaction
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If FTafVldOpe( "CRD", 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, "TAFA280", cCmpsNoUpd )
    //³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
    //³o Commit(), não sendo necessária nenhuma manutenção nas informações³
    If nOpc <> 5

        oModel:LoadValue( "MODEL_CRD", "CRD_FILIAL", CRD->CRD_FILIAL )
        oModel:LoadValue( "MODEL_CRD", "CRD_DTRESP", "2013-10-10" )
        //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
        //³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
        //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
        If nOpc == 4 .and. lItensRem
            For nJ := 1 to oModel:GetModel( "MODEL_CMK" ):Length()
                oModel:GetModel( "MODEL_CMK" ):GoLine(nJ)
                oModel:GetModel( "MODEL_CMK" ):DeleteLine()
            Next nJ
        EndIf

        //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
        //³Roda o XML parseado para gravar as novas informações no GRID³
        //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
        For nJ := 1 to nCMK
            If nOpc == 4 .or. nJ > 1
                oModel:GetModel( "MODEL_CMK" ):lValid:= .T.
                oModel:GetModel( "MODEL_CMK" ):AddLine()
            EndIf
            oModel:LoadValue( "MODEL_CMK", "CMK_QTDRUB", "15")
        Next nJ
    EndIf

    //ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
    //³Efetiva a operação desejada³
    //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
    FWFormCommit( oModel )
    lRet := .T.

    oModel:DeActivate()

EndIf
End Transaction

```



## S-2800 - Desligamento

<b>Rotina:</b>	TAF266.PRW				
<b>Model MVC:</b>	MODEL_CMD				
<b>Validação</b>	TAF266VLD()				
<b>Geração XML</b>	TAF266XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	eSocial	-		
	02	evtDesligamento	-		
	03	id	-		
	03	versao	-		
	04	ideEvento	-		
	05	indRetificacao	-		
	07	nrRecibo	-		
	08	tpAmb	-		
	09	procEmi	-		
	10	indSegmento	-		
	11	verProc	-		
	12	ideEmpregador	-		
	13	tpInscricao	-		
	14	nrInscricao	-		
	15	ideVinculo	-		
	16	cpfTrab	MODEL_CMD	CMD_FUNC	cpfTrab, matricula
	17	nisTrab	-	-	
	18	matricula	-	-	
	19	infoDesligamento	-		
	20	motivoDesligamento	MODEL_CMD	CMD_MOTDES	motivoDesligamento
	21	dtDesligamento	MODEL_CMD	CMD_DTDESL	
	22	indPagtoAPI	MODEL_CMD	CMD_INDPAG	
	23	dtTerminoAPI	MODEL_CMD	CMD_TERAPI	
	24	nrAtestadoObito	MODEL_CMD	CMD_NRATES	
	25	nrProcTrabalhista	MODEL_CMD	CMD_NRPROC	
	26	bcFgtsMesAnt	MODEL_CMD	CMD_FGTSME	
	27	observacao	MODEL_CMD	CMD_OBSERV	
	28	sucessaoVinculo	-		
	29	cnpjSucessora	MODEL_CMD	CMD_CNPJSU	
	30	verbasRescisorias	-		
	31	bcCP	MODEL_CMD	CMD_BCCP	
	32	bcIRRF	MODEL_CMD	CMD_BCIRRF	
	33	bcFGTS	MODEL_CMD	CMD_BCFGTS	
	34	bcFGTSVerbasIndeniz	MODEL_CMD	CMD_BCVERB	
	35	descCP	MODEL_CMD	CMD_DESCCP	
	36	vlrProventos	MODEL_CMD	CMD_VLPROV	
	37	vlrDescontos	MODEL_CMD	CMD_VLDESC	
	38	vlrLiquido	MODEL_CMD	CMD_VLLIQU	
	39	itemRemuneracao	-		
	40	codRubrica	MODEL_CME	CME_CODRUB	codRubrica
	41	qtdRubrica	MODEL_CME	CME_QTDRUB	
	42	vlrRubrica	MODEL_CME	CME_VLRRUB	
43	infoAgenteNocivo	-			



44	grauExp	MODEL_CMD	CMD_GRAUEX	
45	aso	MODEL_CMD	CMD_CODASO	
46	dtAso	MODEL_CMD	CMD_CODASO	dtAso
47	exame	-		
48	dtExame	-		
49	descExame	-		
50	medico	-		
51	nomeMedico	-		
52	foneContato	-		
53	crm	-		
54	nrCRM	-		
55	ufCRM	-		

### Exemplo:

```

aAdd( aChave, { "C", "CMD_FUNC" , FGetIdInt( "cpfTrab" , "", "37400956876" ), .T. } )
aAdd( aChave, { "C", "CMD MOTDES", "0001" , .T. } )
aAdd( aChave, { "D", "CMD_DTDESL", "2013-11-05" , .T. } )
aAdd( aChave, { "C", "CMD_INDPAG", "1" , .T. } )

Begin Transaction
//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Função para validar se a operação desejada pode ser realizada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If FTafVldOpe( "CMD", 2, @nOpc, cFilev, @aIncons, aChave, @oModel, "Tafa266", cCmpsNoUpd )
//³Quando se tratar de uma Exclusão direta, é preciso apenas realizar ³
//³o Commit(), não sendo necessária nenhuma manutenção nas informações³
If nOpc <> 5

oModel:LoadValue( "MODEL_CMD", "CMD_FILIAL", CMD->CMD_FILIAL )
oModel:LoadValue( "MODEL_CMD", "CMD_DTDESL", "2013-11-05")

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Quando se trata de uma alteração, deleta-se todas as linhas do Grid³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
If nOpc == 4 .and. lItenRem
For nJ := 1 to oModel:GetModel( "MODEL_CME" ):Length()
oModel:GetModel( "MODEL_CME" ):GoLine(nJ)
oModel:GetModel( "MODEL_CME" ):DeleteLine()
Next nJ
EndIf

For nJ := 1 to nCME
If nOpc == 4 .or. nJ > 1
oModel:GetModel( "MODEL_CME" ):lValid:= .T.
oModel:GetModel( "MODEL_CME" ):AddLine()
EndIf
oModel:LoadValue( "MODEL_CME", "CME_QTDTRUB", "01")
Next nJ
EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;
//³Efetiva a operação desejada³
//ÀAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAÛ
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()
EndIf
End Transaction

```







```
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU
For nLI := 1 To Len( aRules )

    oModel:LoadValue( "MODEL_CRH", aRules[ nLI, 01 ], FTafGetVal( aRules[
nLI, 02 ], aRules[nLI, 03], aRules[nLI, 04], @aIncons, .F. ) )
    Next

EndIf

//ÚAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAç
//³Efetiva a operação desejada³
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU
FWFormCommit( oModel )
lRet := .T.

oModel:DeActivate()

EndIf
End Transaction
```



## S-2900 – Exclusão de Eventos

<b>Rotina:</b>	TAF269.PRW				
<b>Model MVC:</b>	MODEL_CMJ				
<b>Validação</b>	TAF269VLD()				
<b>Geração XML</b>	TAF269XML()				
<b>Regras</b>	<b>#</b>	<b>TAG</b>	<b>Model</b>	<b>Campo</b>	<b>FGetIDInt()</b>
	01	evtExclusao	-		
	02	Versão	-		
	03	infEvento	-		
	04	idEvento	-		
	05	ideEvento	-		
	06	tpAmb	-		
	07	procEmi	-		
	08	indSegmento	-		
	09	verProc	-		
	10	ideEmpregador	-		
	11	tpInscricao	-		
	12	nrInscricao	-		
	13	infoExclusao	-		
	14	tpEvento	MODEL_CMJ	CMJ_TPEVEN	
	15	nrReciboEvento	MODEL_CMJ	CMJ_NRRECI	
	16	cpfTrab	MODEL_CMJ	CMJ_TRABAL	Matricula,cpfTrab
17	nisTrab	-	-		

### Exemplo:

```

Aadd( aChave, { "C", "CMJ TPEVEN", "1", .F. } )
aAdd( aChave, { "C", "CMJ_TRABAL", FGetIDInt("cpfTrab", "", "37400956876") , .T. } )

Begin Transaction
  If FTafVldOpe( "CMJ", 2, @nOpc, cFileEv, @aIncons, aChave, @oModel, "TAF269", cCmpsNoUpd )
    oModel:LoadValue( "MODEL_CMJ", "CMJ TPEVEN", "1")
    oModel:LoadValue( "MODEL_CMJ", "CMJ_NRRECI ", "12165165465")
    oModel:LoadValue( "MODEL_CMJ", "CMJ_FILIAL", CMJ->CMJ_FILIAL )
  EndIf
End Transaction
    
```

## FUNÇÕES DE UTILIZAÇÃO GENÉRICA

### 1.1 FGetIdInt (cFieldLay1,cFieldLay2,cInfo1,cInfo2, lIdChave, aInfComp)

#### Descrição da função:

Função responsável por retornar os IDs para gravação nas tabelas do TAF durante a integração.

#### Parâmetros:

Ordem	Parâmetro	Tipo	Descrição
01	cFieldLay1	Caracter	Primeiro nome do Campo do Layout do eSocial a que se refere à informação
02	cFieldLay2	Caracter	Segundo nome do Campo do Layout do eSocial a que se refere à informação
03	cInfo1	Caracter	Informação referente ao primeiro campo informado (chave)
04	cInfo2	Caracter	Informação referente ao segundo campo informado (chave)
05	lIdChave	Lógico	Indica se os valores indicados são TAGs ou valores já para execução do Seek ( .T. = Tag , .F. = Valor )
06	aInfComp	Array	Array com os campos a mais da chave que devem ser incluídos na tabela auxiliar

#### Lista de Opções:

Minemônimo1	Minemônimo2	Descrição	Exemplo
cpfTrab	codMotivoAnterior	Afastamento Temporário (CM6)	FGetIdInt("cpfTrab", "", "78887020230", .F.)
IniEst		Estabilidade Inicial (CM5)	FGetIdInt("IniEst", "", "000001", .F.)
cpfBeneficiario		Beneficiário (C9Z)	FGetIdInt("cpfBeneficiario", "", "78887020230", .F.)
tpProcesso	NrProcesso   nrProcJud	Processo referenciado E-Social (Tabela C93)	FGetIdInt("tpProcesso", "nrProcesso", "J", "123456")
tpInscricao	nrInscricao	Estabelecimento (C92)	FGetIdInt("tpInscricao", "nrInscricao", "F", "66610634327", .F.)
classTrib		Classificação Tributária (C8D)	FGetIdInt("classTrib", "", "123", .F.)
codAgenteQuimico		"CUA"	FGetIdInt("codAgenteQuimico", "", "123", .F.)
codAnalise		"CUM"	FGetIdInt("codAnalise", "", "123", .F.)
codAgente		Código Agente (Tabela C98)	FGetIdInt("classTrib", "", "123", .F.)
tpDespesa		"CUV"	FGetIdInt("tpDespesa", "", "123", .F.)
codMotivoEstabilidade		CUR	FGetIdInt("codMotivoEstabilidade", "", "123", .F.)
cpfTrab		Trabalhador (C9V)	FGetIdInt("cpfTrab", "", "37400956872", .F.)
banco		Bancos (C1V)	FGetIdInt("banco", "", "123", .F.)

catTrabalhador   cod Categ   categOrigem		Categoria de Trabalhador (C87)	FGetIdInt("catTrabalhador", "", "1", "", .F.)
codCargo		Cargos (C8V)	FGetIdInt("codCargo", "", "01", "", .F.)
codFuncao		Funções (C8X)	FGetIdInt("codFuncao", "", "01", "", .F.)
codLotacao		Lotações (C99)	FGetIdInt("codLotacao", "", "01", "", .F.)
codJornadaSegSexta   codJornadaSab   codJornadaSeg   codJornadaTer   codJornadaQua   codJornadaQui   codJornadaSex   codJornadaSab   codJornadaDom   codJornada		Horários/Turnos de Trabalho (C90)	FGetIdInt("codJornadaSegSexta", "", "01", "", .F.)
dtAso		Atestado de Saúde Ocupacional (C8B)	FGetIdInt("dtAso", "", "000001", "01/01/2013", .F.)
tpDep		Tipo de Dependente (CMI)	FGetIdInt("tpDep", "", "01", "", .F.)
tpDesligDiretorNaoEmpregado		Mot deslig. Diretor sem Vínculo (CML)	FGetIdInt("tpDesligDiretorNaoEmpregado", "", "01", "", .F.)
codRubrica		Rubricas (C8R)	FGetIdInt("codRubrica", "", "01", "", .F.)
motivoDesligamento		Motivos de Desligamento (C8O)	FGetIdInt("motivoDesligamento", "", "01", "", .F.)
grauExp		Grau Exposição Agentes Nocivos (C88)	FGetIdInt("grauExp", "", "01", "", .F.)
tpContrib		Tipo de Contribuição (C8H)	FGetIdInt("tpContrib", "", "01", "", .F.)
uf	codMunicipio	Municípios (C07)	FGetIdInt("uf", "codMunicipio", "SP", "00001", .F.)
uf   ufCtps		Unidades da Federação (C09)	FGetIdInt("ufCtps", "", "SP", "", .F.)
paisNascto   paisNacionalidade   paisResidencia		Países Banco Central/SISCO MEX (C08)	FGetIdInt("paisNascto", "", "00001", "", .F.)
codCBO		Código Brasileiro de Ocupação (C8Z)	FGetIdInt("codCBO", "", "01", "", .F.)
fpas	codTerceiros	Cod. Aliq. FPAS / Terceiros (C8A)	FGetIdInt("fpas", "codTerceiros", "01", "001", .F.)

tpLogradouro		Tipo de Logradouro (C06)	FGetIdInt("tpLogradouro", "", "01", "", .F.)
tpLotacao		Tipo de Lotação (C8F)	FGetIdInt("tpLotacao", "", "01", "", .F.)
idVara		Cadastro de Varas (C9A)	FGetIdInt("idVara", "", "000001", "", .F.)
indDecisao		Indicativo de Decisão (C8S)	FGetIdInt("indDecisao", "", "01", "", .F.)
natRubrica		Natureza da Rubrica (C89)	FGetIdInt("natRubrica", "", "01", "", .F.)
codIncidCP		Incidenc. Trib. Prev. Soc. (C8T)	FGetIdInt("codIncidCP", "", "01", "", .F.)
codIncidIRRF		Incidenc. Trib. IRRF. (C8U)	FGetIdInt("codIncidIRRF", "", "01", "", .F.)
codParteAtingida		Parte Atingida (C8I)	FGetIdInt("codParteAtingida", "", "01", "", .F.)
codAgenteCausador		Agente causador (C8J)	FGetIdInt("codAgenteCausador", "", "01", "", .F.)
nrOC		Médico (C87)	FGetIdInt("nrOC", "", "SP12345", "", .F.)
codSitGeradora		Situação Geradora (C8J)	FGetIdInt("codSitGeradora", "", "01", "", .F.)
codMotAfastamento		Motivo Acidente (C8N)	FGetIdInt("codMotAfastamento", "", "01", "", .F.)
codCID		Motivo Acidente (C8N)	FGetIdInt("codCID", "", "01", "", .F.)
nomeMedico		Nome do médico (CM7)	FGetIdInt("nomeMedico", "", "Carlos Eduardo", "", .F.)
grauInstrucao		Grau de Instrucao (CMH)	FGetIdInt("grauInstrucao", "", "01", "", .F.)
tpEvento		C8E	FGetIdInt("tpEvento", "", "123", "", .F.)
codCID		Motivo Acidente (C8N)	FGetIdInt("codCID", "", "01", "", .F.)
tpEstabelecimento		Tabela de Estabelecimentos (C92)	FGetIdInt("tpEstabelecimento", "J", "66228823000187", "", .F.)
tpServico		Tipo servico E-Social (C8C)	FGetIdInt("tpServico", "", "01", "", .F.)
dtCancAvisoPrevio		Aviso Prévio (CM8)	FGetIdInt("dtCancAvisoPrevio", "", "000001", "", .F.)
cpfResponsavel		Responsável Informação (C9F)	FGetIdInt("cpfResponsavel", "", "78887020230", "", .F.)
natJuridica		C8P	FGetIdInt("natJuridica", "", "123", "", .F.)
tpCondicao		CUN	FGetIdInt("tpCondicao", "", "123", "", .F.)
tplsencao		CMY	FGetIdInt("tplsencao", "", "123", "", .F.)

tpRendimento		CUC	FGetIdInt("tpRendimento", "", "123", "", .F.)
formaTributacao		CUD	FGetIdInt("formaTributacao", "", "123", "", .F.)
paisResidencia		C08	FGetIdInt("paisResidencia", "", "123", "", .F.)
codRendimento		CUF	FGetIdInt("codRendimento", "", "123", "", .F.)
relFontePagadora		"CUB"	FGetIdInt("relFontePagadora", "", "123", "", .F.)
relDependencia		CMI	FGetIdInt("relDependencia", "", "123", "", .F.)
tpProcRRA	nrProcRRA	C93	FGetIdInt("tpProcRRA", "", "123", "", .F.)
tpInscAdvogado	nrInscAdvogado	CMW	FGetIdInt("tpInscAdvogado", nrInscAdvogado, "1", "32148", "", .F.)
nrProcJud		C93	FGetIdInt("nrProcJud", "", "123", "", .F.)
cpfTrab	matricula		FGetIdInt("cpfTrab", "matricula", "37400956871", "001", .F.)

**Retorno:**

Ordem	Retorno	Tipo	Descrição
01	cId	Caracter	Retorna o ID do registro. Caso não encontre, retorna a string "ERR" + Código passado por parâmetro

**1.2 FTafVldOpe** (*cAlias, nInd, nOpc, cFilEv, alIncons, aChave, oModel, cNomModel, cCmpsNoUpd, nIndIdVer, lRuleEvCad, aNewData, cMatTrab*)

**Descrição da função:**

Função que valida se a operação desejada pode ser realizada. Retornando os campos obrigatórios de controle da tabela (Filial, ID, Versão Anterior)

**Parâmetros:**

Ordem	Parâmetro	Tipo	Descrição
01	cAlias	Caracter	Alias a ser gravado
02	nInd	Numérico	Índice de busca da chave única do registro ( De acordo com o Layout do E-Social. )
03	nOpc	Numérico	Operação a ser realizada: 3 - Inclusão 4 - Alteração 5 – Exclusão Esta variável é modificada pela função TAFRegStat(). Podendo ser: 4 – Alteração – Registros não transmitidos 5 – Exclusão – Registros não transmitidos 6 – Alteração – Registros já transmitidos 7 – Exclusão – Registros já transmitidos 9 – Contém inconsistências.
04	cFilEv	Caracter	Filial de Origem do ERP, sempre deve ser informada.
05	alIncons	Array	Array para gravação de inconsistências encontradas
06	aChave	Array	Chave de busca do registro na tabela, sendo que:  Posição 1 do Array: Tipo da informação  Posição 2 do Array: Nome do Campo da tabela  Posição 3 do Array: Valor referente ao campo informado na posição 2.



			Posição 4 do Array: .T.
07	oModel	Nil	Objeto a ser criado para manter as informações
08	cNomModel	Caracter	Nome do Fonte referente ao Model
09	cCmpsNoUpd	Caracter	Nome dos campos que não devem ser replicados no novo registro no caso de uma exclusão de um registro já transmitido ao RET.
10	nIndIdVer	Númerico	Índice referente à chave Filial+ID+Versao da tabela.
11	IRuleEvCad	Lógico	Indica se o registro se refere aos eventos cadastrais
12	aNewData	Array	Conteudo da tag novaValidade para operacoes da alteracao de eventos cadastrais
13	cMatTrab	Caracter	Matrícula do funcionário, quando não for evento de funcionário ignorar esse parâmetro.

### Retorno:

Ordem	Retorno	Tipo	Descrição
01	Ivalido	Lógico	Indica se a operação desejada pode ser realizada ou não.
<b>Atenção para os parâmetros que são passados com @referência. Necessário para que sejam alimentadas através da função.</b>			

### 1.3 FTafGetFil (cSeekFil,alIncons)

#### Descrição da função:

Função que tem por finalidade retornar a filial para onde as informações de integração serão importadas.

#### Parâmetros:

Ordem	Parâmetro	Tipo	Descrição
01	cSeekFil	Caracter	Filial a ser verificada a existência
02	alIncons	Array	Array contendo as inconsistências encontradas.

### Retorno:

Ordem	Retorno	Tipo	Descrição
01	cFilTAF	Caracter	Indica a filial para onde as informações de integração serão importadas.

### 1.4 xFunGetVer()

#### Descrição da função:

Função responsável por retornar a versão do registro no TAF.

**Retorno:**

Ordem	Retorno	Tipo	Descrição
01	cVersao	Caracter	Retorno da versão do registro, sendo AAAAMMDDHHMMSS  A – Ano M – Mês D – Dia H – Hora M – Minuto S – Segundo

**Importante:**

Para todos os registros do E-Social a chave única das tabelas é a mesma indicada no *Layout* do Fisco, com exceção do registro S-1000 - Informações do Empregador onde a chave única é a filial de origem das informações, conforme abaixo:

Registro	Chave única do <i>Layout</i> E-Social	Tam.Chave
S-1000	Filial +Filial de Origem das informações	8
S-1010	Filial + CodRubrica + iniValidade + fimValidade	46
S-1020	Filial + CodLotacao + iniValidade + fimValidade	46
S-1030	Filial + CodCargo + iniValidade + fimValidade	46
S-1040	Filial + CodFuncao + iniValidade + fimValidade	46
S-1050	Filial + CodJornada + iniValidade + fimValidade	46
S-1060	Filial + TpInscricao + nrInscricao + iniValidade + fimValidade	31
S-1070	Filial + TpProcesso + nrProcesso + iniValidade + fimValidade	37
S-1080	Filial + CnpjOpPortuario + iniValidade + fimValidade	30
S-1100	Filial + indApuracao+perApuracao	7
S-1200	Filial + indApuracao+perApuracao+cpfTrab	19
S-1300	Filial + indApuracao+perApuracao	7
S-1310	Filial + indApuracao+perApuracao	7
S-1320	Filial + indApuracao+perApuracao	7
S-1330	Filial + indApuracao+perApuracao	7
S-1340	Filial + indApuracao+perApuracao	7
S-1350	Filial + indApuracao+perApuracao	7





S-1360	Filial + indApuracao+perApuracao	7
S-1370	Filial + indApuracao+perApuracao	7
S-1380	Filial + indApuracao+perApuracao	7
S-1390	Filial + indApuracao+perApuracao	7
S-1399	Filial + indApuracao+perApuracao	7
S-1400	Filial + indApuracao+perApuracao	7
S-1800	Filial + dtApuracao	10
S-2100	Filial + Matricula+cpfTrab	41
S-2200	Filial + Matricula+cpfTrab	41
S-2220	Filial + cpfTrab+dtAlteracao	19
S-2240	Filial + Matricula+cpfTrab+dtAlteracao	49
S-2260	Filial + cpfTrab+dtAcidente+horaAcidente	24
S-2280	Filial + Matricula+cpfTrab+dtAso	49
S-2320	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2325	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2330	Filial + Matricula+cpfTrab+codMotAfastamento	43
S-2340	Filial + Matricula+cpfTrab+dtIniEstabilidade	49
S-2345	Filial + Matricula+cpfTrab+dtFimEstabilidade	49
S-2360	Filial + Matricula+cpfTrab+dtIniCondicao+tpCondicao	51
S-2365	Filial + Matricula+cpfTrab+tpCondicao+dtFimCondicao	51
S-2400	Filial + Matricula+cpfTrab+dtAvisoPrevio	49
S-2405	Filial + Matricula+cpfTrab+dtCancAvisoPrevio	49
S-2600	Filial + Matricula+cpfTrab	40
S-2620	Filial + cpfTrab+dtAlteracao	19
S-2680	Filial + cpfTrab+dtTermino	19
S-2800	Filial + Matricula+cpfTrab+motivoDesligamento+dtDesligamento+indPagtoAPI	52
S-2820	Filial + Matricula+cpfTrab+tpReintegracao+nrProcJud	48
S-2900	Filial + tpEvento+cpfTrab	17

**O controle de transação é de responsabilidade do ERP de origem. Sempre antes de iniciar o processo de integração é imprescindível que o controle esteja ativo para evitar inconsistências na base do TAF.**

### 1.5 xFunRetPrt (cTpEvento,cTafKey)

#### Descrição da função:

Esta função tem como objetivo retornar o número do protocolo gerado ao RET e o status do registro solicitado dentro do TAF.

#### Parâmetros:

Ordem	Parâmetro	Tipo	Descrição
01	cTpEvento	Caracter	Tipo do evento, por exemplo, "S-2260"
02	cTafKey	Caracter	A chave de busca na tabela do Evento conforme informado neste manual.

#### Retorno:

Ordem	Retorno	Tipo	Descrição
01	aRet	Array	O retorno desta função é um Array com a estrutura conforme abaixo: aRet[ 01 ] – Indica se a chave informada foi encontrada aRet[ 02, 01 ] – Indica o número do Protocolo encontrado aRet[ 02, 02 ] – Indica o status do registro encontrado

### 1.6 TafPerOpen ( cFilAnt, cMes, cAno )

#### Descrição da função:

Esta função tem como objetivo retornar se existe período em aberto de acordo com os parâmetros informados.

#### Parâmetros:

Ordem	Parâmetro	Tipo	Descrição
01	cFilAnt	Caracter	Filial do TAF onde será realizada a consulta.
02	cMes	Caracter	Mês de referência do período a ser consultado ( Formato MM )
03	cAno	Caracter	Ano de referência do período a ser consultado ( Formato AAAA )

#### Retorno:

Ordem	Retorno	Tipo	Descrição
01	aRet	Array	Caso o retorno do array seja vazio significa que não existe período em aberto para os dados informados nos parâmetros da função, pode ser que exista o período em aberto no evento S-1300 porém o mesmo período já foi encerrado pelo evento S-1399.  No caso de existir períodos em aberto o retorno do Array será realizado conforme abaixo:  aRet[ 01 ] – Indica qual o tipo do período que foi encontrado, podendo ser: 1 = Folha de Pagamento Mensal 2=Folha de Decimo Terceiro Salario



			<p>aRet[ 02 ] – Indica o Status do período encontrado, podendo ser:</p> <ul style="list-style-type: none"> <li>0=Reg.Valido</li> <li>1=Reg.Invalido</li> <li>2=Reg.Transmitido</li> <li>3=Reg.Transmitido com inconsistência</li> <li>4=Reg.Transmitido valido</li> <li>9=Em Processamento</li> </ul>
--	--	--	---

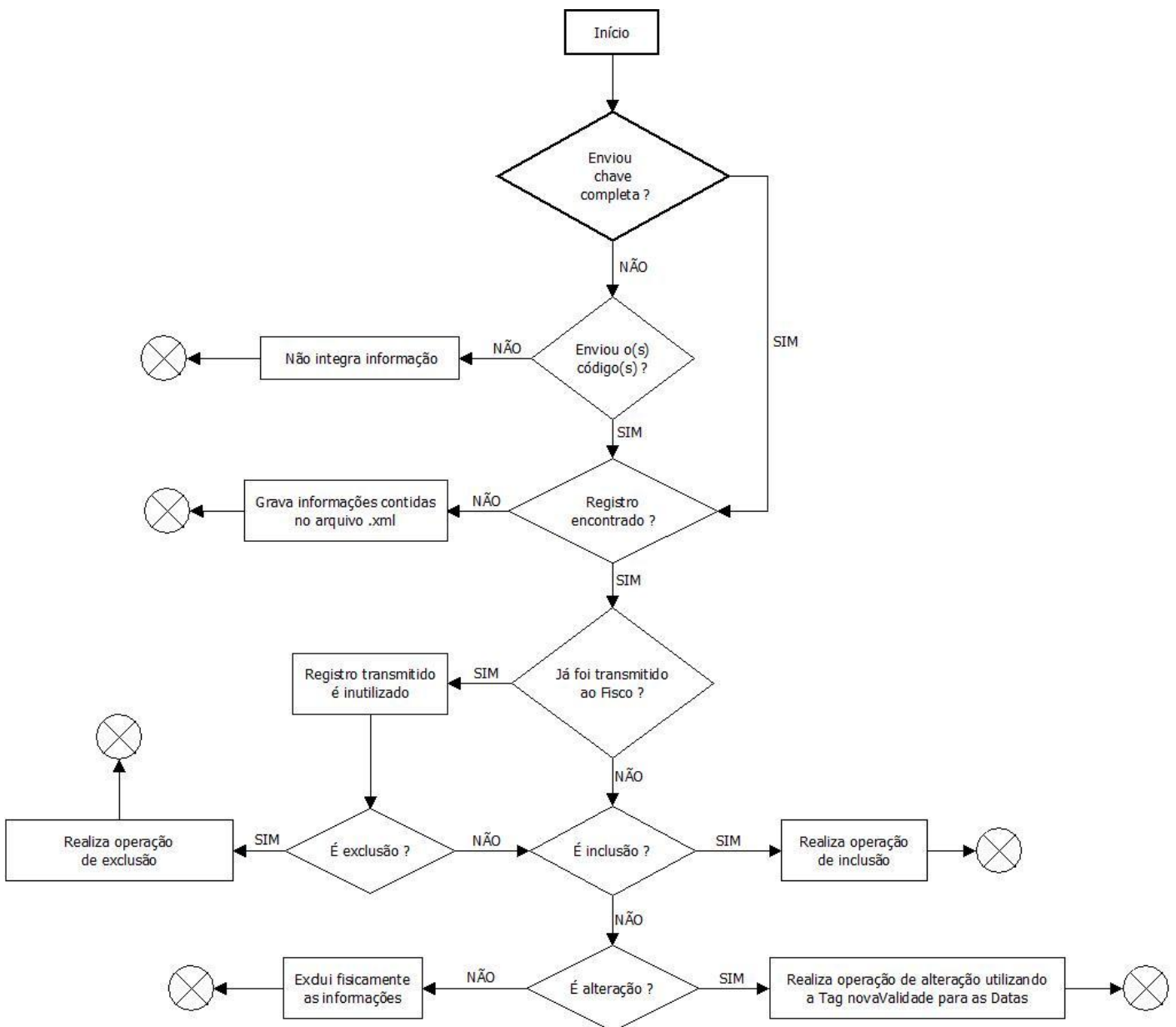
## 7. Controle de Versão

O grupo TOTVS representado por suas marcas irá administrar as demandas de evolução dos *layouts* e demais ajustes, acordando junto aos solicitantes o prazo de liberação de *release*.

Todas as evoluções programadas deverão ser discutidas e aprovadas pelas marcas antes do início do desenvolvimento e somente serão desenvolvidas em caso de concordância das marcas e alinhamento com as diretrizes definidas pelo Comitê de Integração TOTVS.

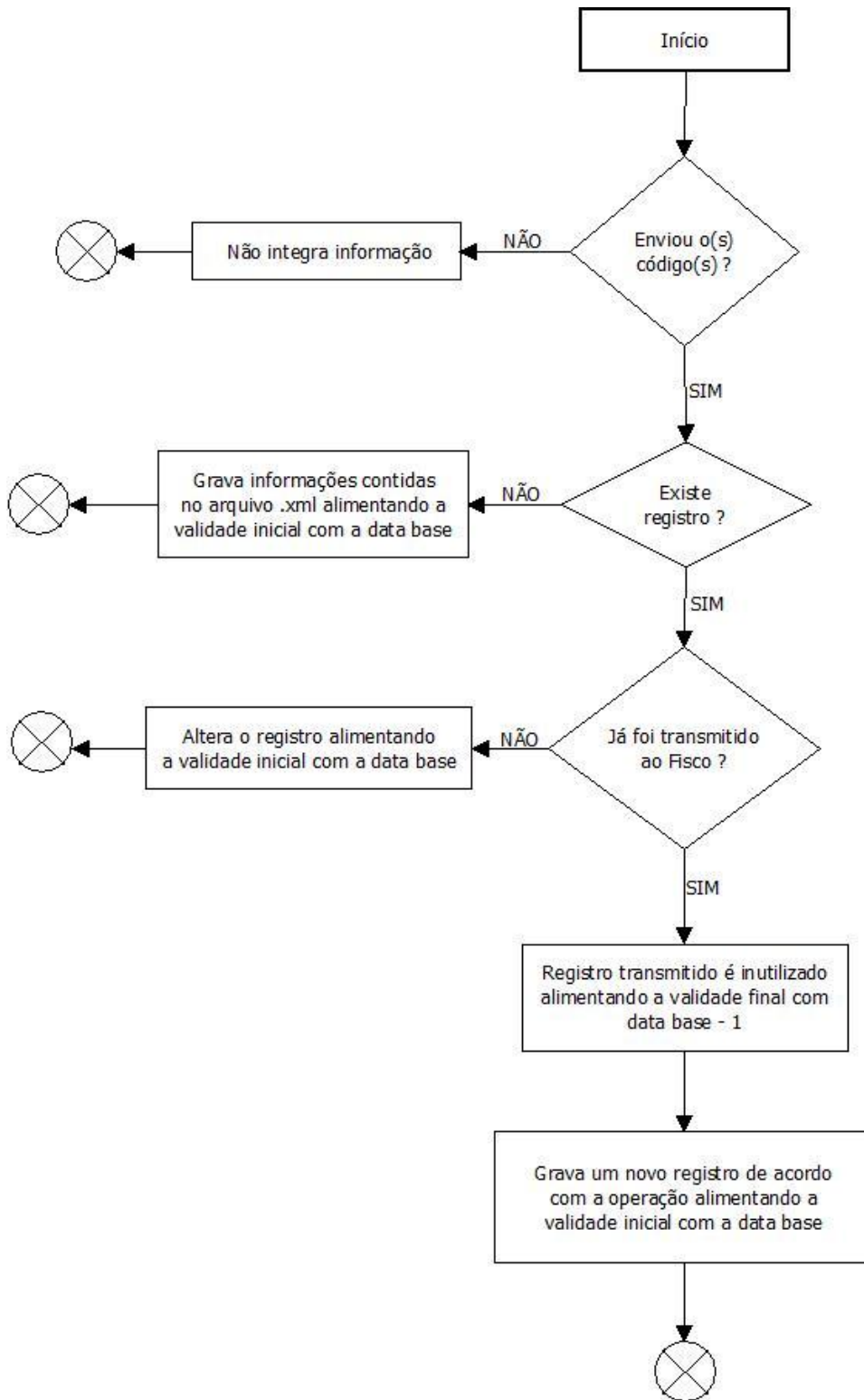
## 8. Fluxo das Informações

- Integração de eventos Tabela (S-1010 a S-1080) utilizando conceito de controle sobre as datas de validade.





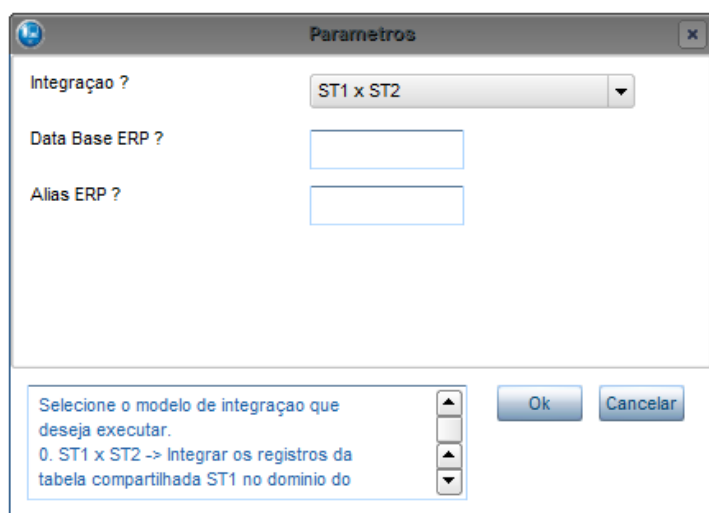
- **Integração de eventos Tabela (S-1010 à S-1080) não utilizando conceito de controle sobre as datas de validade.**



## 9. Como fazer

- **Integração Manual:**

Para executar a integração manual acesse a opção de menu: **Miscelânea/Esocial/Integração Esocial**.



Os parâmetros “Data Base ERP?” e “Alias ERP?” referem-se apenas à integração “ST1 x ST2”, indicando a localização da tabela TAFST1 de domínio do ERP.

- **Integração via Agendamento (Schedule).**

Primeiramente, é preciso configurar o arquivo appserver.ini como segue:

**[OnStart]**

jobs=FWSCHDQUEUE,FWSCHDMANAG\_XXX  
RefreshRate=NNN

**[FWSCHDMANAG]**

Main=FWSCHDMANAG  
Environment=XXX

**[FWSCHDQUEUE]**

Main=FWSCHDQUEUE  
Environment=XXX

**[FWSCHDMANAG\_XXX]**

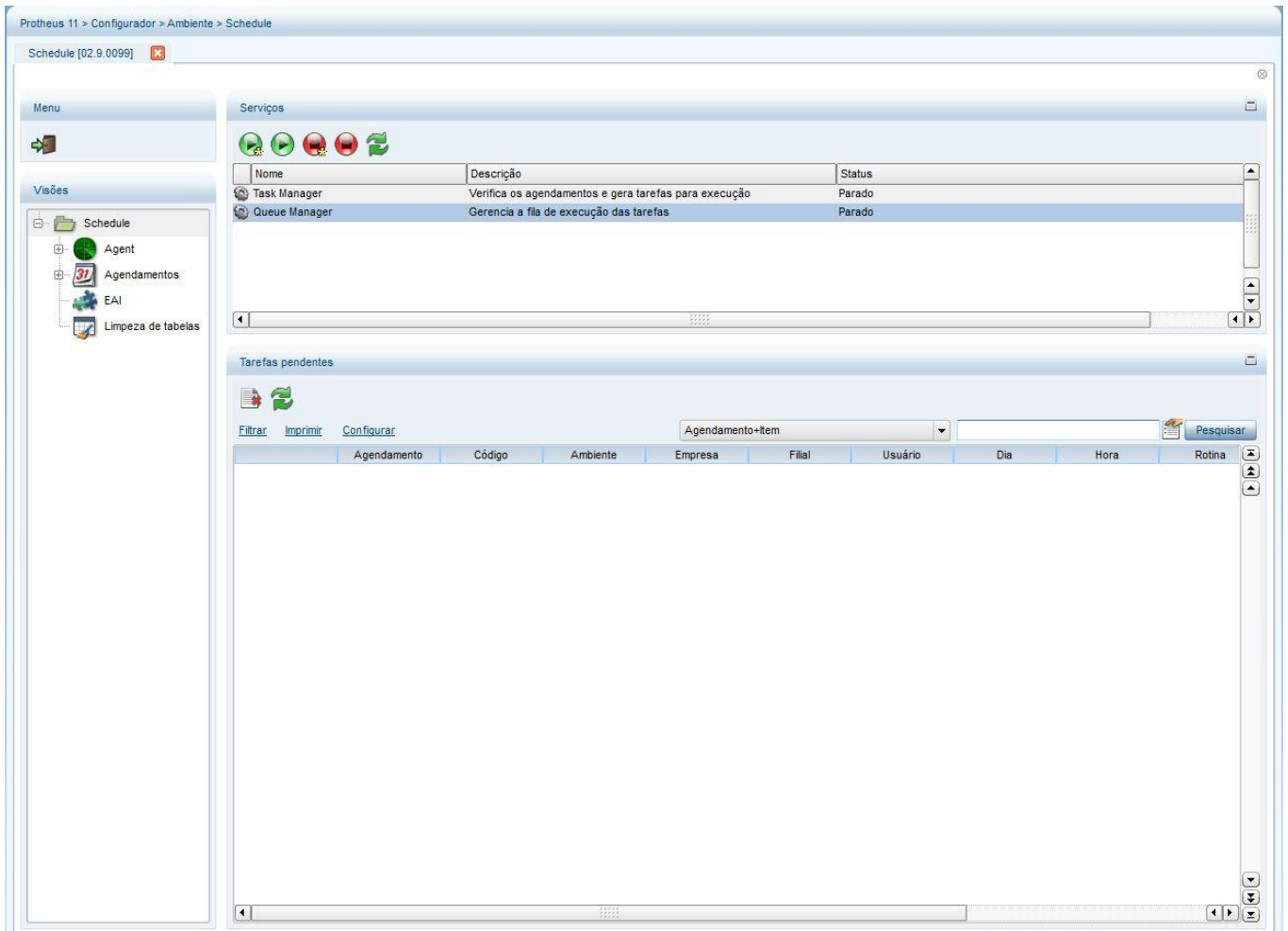
Main=FWSCHDMANAG  
Environment=XXX



Onde:

- **XXX** é nome do ambiente.
- **NNN** é o intervalo de tempo (em segundos) de atualização dos dados sobre a execução do(s) *job(s)*.

Em seguida, deve-se configurar o *Schedule* no Configurator (SIGACFG) acessando a opção de menu **Ambiente/Schedule/Schedule**.



O primeiro passo é incluir um Agente:





Protheus 11 > Configurador > Ambiente > Schedule

TOTVS Série T Manufatura (Microsiga) 02.9.0099

Menu

Visões

- Schedule
  - Agent
  - Cadastro
  - Monitor
  - Agendamentos
    - Cadastro
    - Monitor
  - EAI
  - Limpeza de tabelas

Browse

Código  Buscar

Código	Nome	IP	Porta	Ambiente	Empresa	Nome da Empres	Nº de Threads	Ultimo dia	Ultima hora
01	AGENT 01	localhost	1401	TAF_SQL	T1	Grupo TOTVS 1	1	18/11/2013	14:11

Detalhes

Código\*  Nome\*

IP\*  Porta\*

Ambiente\*  Empresa\*  Nome da Empresa

Nº de Threads  Habilitado

Em seguida, é preciso criar um Agendamento para cada rotina (*job*) que será executada:

Protheus 11 > Configurador > Ambiente > Schedule

TOTVS Série T Manufatura (Microsiga) 02.9.0099

Menu

Visões

- Schedule
  - Agent
    - Cadastro
    - Monitor
  - Agendamentos
    - Cadastro
    - Monitor
  - EAI
    - Limpeza de tabelas

Browse

Imprimir Configurar

Código

Código	Usuário	Nome do Usuário	Rotina	Data	Hora	Ambiente	Empresa/Filial	Módulo	Nome do Módulo	Descrição
000001	000000	Administrador	TAFANTEG	20131118	08:00	TAF_SQL	T1/D MG 01;	84	Produto Fiscal único	Escalonamento de integração ST1 x ST2.
000002	000000	Administrador	TAFANTEG	20131119	00:00	TAF_SQL	T1/D MG 01;	84	Produto Fiscal único	Escalonamento de integração ST2 x TAF.
000003	000000	Administrador	TAFANTEG	20131119	00:00	TAF_SQL	T1/D MG 01;	84	Produto Fiscal único	Escalonamento de Validação.
000004	000000	Administrador	TAFANTEG	20131119	00:00	TAF_SQL	T1/D MG 01;	84	Produto Fiscal único	Escalonamento de integração com TSS.

Detalhes

Código\* 000001    Usuário\* 000000    Nome do Usuário Administrador    Rotina\* TAFANTEG    Parâmetros  
 Data\* 18/11/2013    Hora 08:00    Ambiente TAF\_SQL    Empresa/Filial\* T1/D MG 01;  
 Módulo\* 84    Nome do Módulo Produto Fiscal unico    Habilitada     Descrição Escalonamento de integração ST1 x ST2.

**Obs.:** preencher o campo Rotina com "TAFANTEG(par1, par2, par3)", onde:

- **par1** é o código do *job* a ser executado.
- **par2** é o TOP ALIAS para a base do ERP.
- **par3** é o Database do ERP.