

Outubro 2011 | rev. 01



# TOTVS

**MANUAL**

**AGENDADOR DE TAREFAS LOGIX**

**Elaboração:**

**Rubens Dos Santos Filho**

Framework | [rubens.santos@totvs.com.br](mailto:rubens.santos@totvs.com.br)

## Objetivo

Este documento tem como objetivo relatar a funcionalidade de agendamento de tarefas no produto Logix, que realiza a execução de processos em modo *background* (sem a interação com o usuário), onde são executados em datas e intervalos definidos pelo usuário.

Os processos acionados pelo agendador são chamados de gatilhos e cada um possui tarefas a serem executadas pelo mesmo. Tais gatilhos são cadastrados através do programa **JOB0003 – Agendador de Tarefas** e o monitoramento da execução destes pode ser feito através do programa **JOB0005 – Monitor de Tarefas**.

## Funcionamento

Uma vez iniciado o serviço TOTVS Tec com o agendador de tarefas ligado, imediatamente é acionado um escalonador (*scheduler*) que verifica ininterruptamente se há algum gatilho a ser executado conforme período e intervalos definidos pelo usuário. Estes gatilhos são executados em uma nova linha de execução (*thread*), para não interferir nos gatilhos que ainda estão executando.



## Pré-requisitos

Para executar o agendador de tarefas Logix, é necessário a utilização da tecnologia TOTVS Tec e estar no mínimo com a versão 10.02 do Logix. Esta ferramenta trabalha nos servidores e banco de dados homologados pela TOTVS.

**É extremamente aconselhável criar um ambiente TOTVS Tec separado dos outros ambientes para execução do agendador de tarefas.**

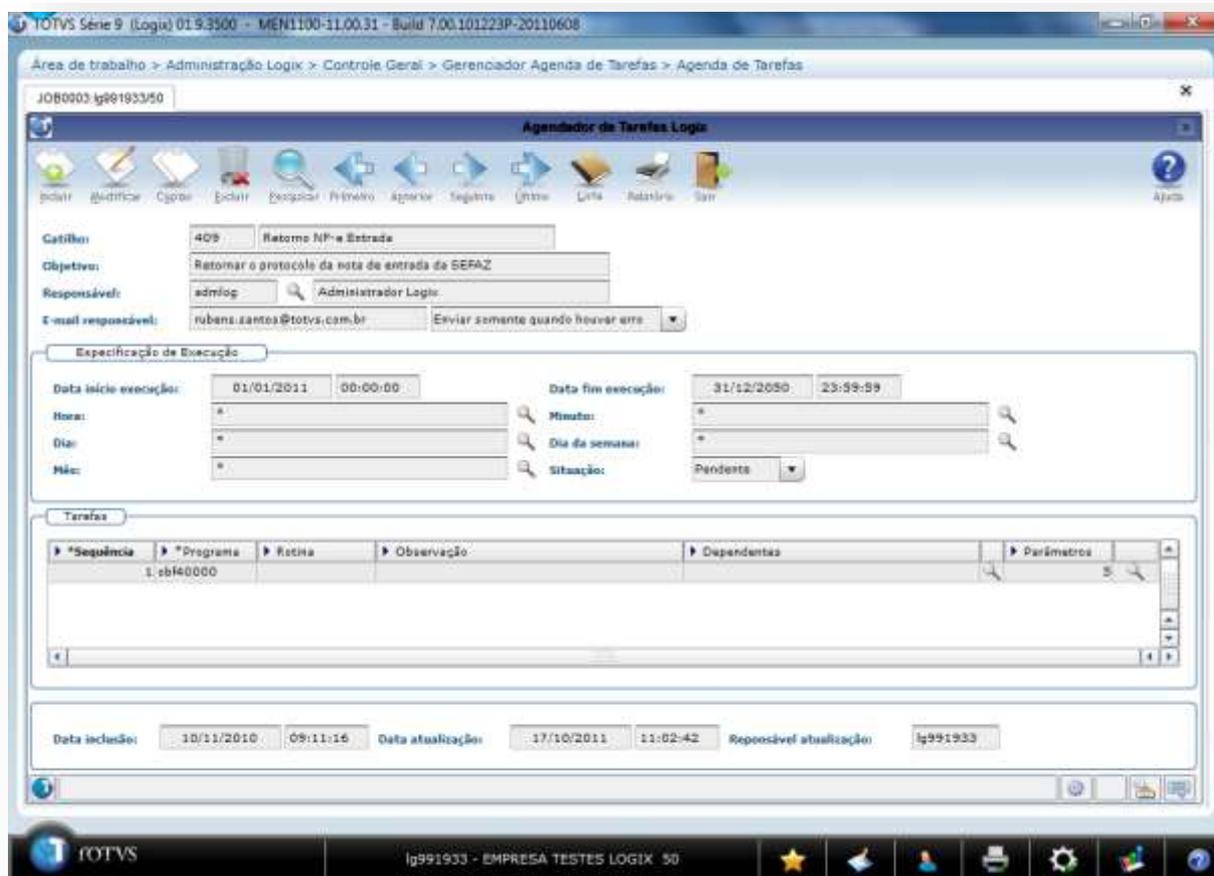


## JOB0003 – Agendador de Tarefas Logix (Cadastro de Gatilhos)

No **JOB0003** são cadastrados os gatilhos a serem executados pelo agendador de tarefas do Logix e suas especificações. Este programa é subdividido em quatro partes:

- Informações do gatilho;
- Especificação de execução;
- Tarefas;
- Informações de atualizações.

Nas informações do gatilho, são cadastradas informações básicas como: nome e descrição do gatilho, responsável pelo gatilho, e-mail do responsável e quando o sistema deverá enviar e-mails para o responsável.



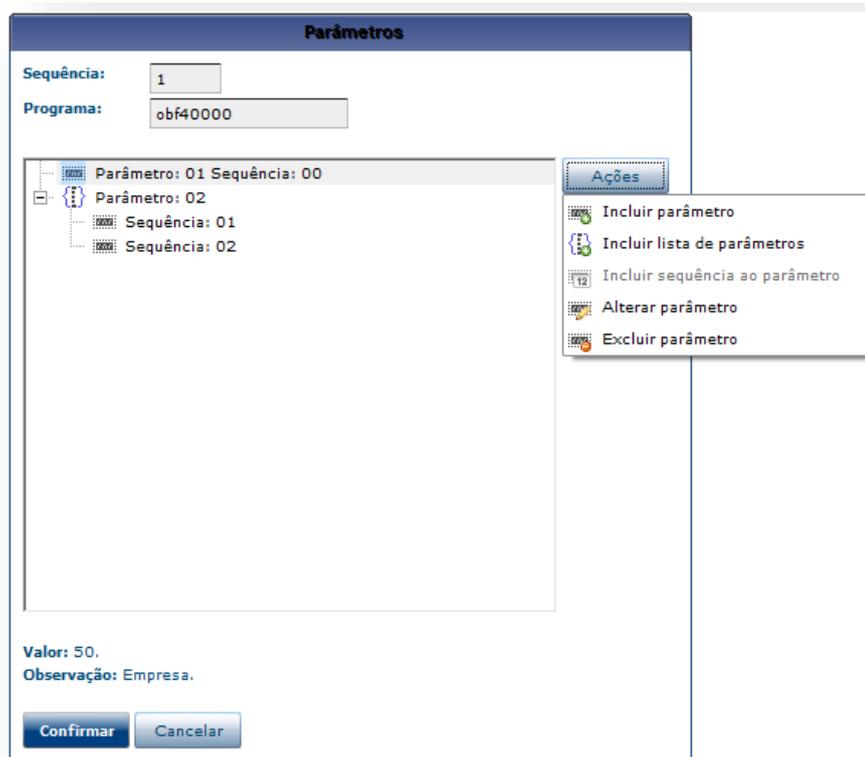
Na especificação de execução, são informados o período e os intervalos de execução do gatilho. Também é possível verificar a situação do gatilho, se o mesmo está pendente para execução, em andamento ou inativo (lembrando que para que um gatilho possa ser modificado, o mesmo não pode estar em andamento).

Os períodos de execução são informados através do campo de data inicial e final e os intervalos são definidos por hora, minuto, dia, dia da semana e mês. Estas informações são selecionadas a partir da janela abaixo, ao clicar no botão de lupa do respectivo campo do qual o usuário deseja informar:



No cadastro das tarefas, são informadas as tarefas (processos) que serão executadas para este gatilho e possibilita também as definições de:

- **Rotina;**
- **Observações;**
- **Dependentes:** define se a execução da tarefa depende do término com sucesso da tarefa executada anteriormente;
- **Parâmetros (simples e lista):** os parâmetros simples aceitam apenas um valor e as listas aceitam diversos valores, seqüenciados conforme são adicionados na mesma. Estes parâmetros são informados na tela abaixo:





## Configuração

Depois de cadastrados os gatilhos é necessário configurar o agendador de tarefas para a execução dos mesmos. Esta configuração consiste na criação de seções de execução dentro do arquivo de informações do servidor de aplicação (TotvsAppServer.ini) localizado dentro da pasta do servidor TOTVS Tec. As seções que deverão ser incluídas, bem como suas chaves e valores de execução, deverão estar como a seguir:

```
[OnStart]
Jobs=4glJob

[4glJob]
Environment=<ambiente Logix>
Main=JobStarter

[LogixScheduler]
ScheduleEnable=1
JobLogEnable=1
JobLogPath=C:\totvs\logix\bin\appserverj\log\
LoopLoadJob=33
VerifyJobInterval=55
JobMonitorAutoRefresh=5
; Informações utilizadas somente quando se faz uso de mais de um servidor
; (LoadBalance) para execução dos gatilhos agendados.
EnableJobBalance=0
JobServers=slave1,slave2
; Informações destinadas à validação de licença.
LicenseCompanyId=50
LicenseUserId=admlog

[slave1]
Type=tcPIP
Server=lnap07
Port=3301
Connections=50

[slave2]
Type=tcPIP
Server=lnap07
Port=3302
Connections=50
```

Seção	Descrição
[OnStart]	Utilizado para iniciar automaticamente uma seção definida no arquivo de configuração quando o servidor for iniciado.
Jobs	Indica qual a seção JOB configurado no arquivo de configuração será executada.
[4gJob]	Configuração da seção JOB. Nela será definida qual o ambiente será utilizado e qual o nome do programa responsável por executar o agendador de tarefas (vide arquivo de configuração).
Environment	Indica em qual o ambiente <b>JOB</b> configurado no arquivo de configuração será executado o agendador de tarefas.
Main	Nome do programa (AdvPL) que executará o agendador de tarefas.
[LogixScheduler]	Seção de configuração do agendador de tarefas Logix.
ScheduleEnable	Habilita/desabilita o escalonador do agendador de tarefas.
JobLogEnable	Habilita/desabilita a geração do arquivo de <b>log</b> de execução para cada gatilho.
JobLogPath	Caminho do servidor onde serão gerados os arquivos de <b>log</b> dos gatilhos.
LoopLoadJob	Quantidade de execuções a serem realizadas até a próxima atualização da lista de execuções ( <i>Select</i> em banco de dados). <b>Obs.:</b> o valor padrão mínimo para não sobrecarregar o servidor TOTVS Tec é 33.
VerifyJobInterval	Tempo em segundos entre cada execução de gatilho. <b>Obs.:</b> o valor padrão mínimo para não sobrecarregar o servidor TOTVS Tec é 55.
JobMonitorAutoRefresh	Tempo em segundos para a atualização automática das informações de gatilhos em execução no monitor de tarefas do agendador ( <b>JOB0005</b> ). <b>Obs.:</b> o valor padrão mínimo para não sobrecarregar o servidor TOTVS Tec é 5.
EnableJobBalance	Habilita/desabilita <i>LoadBalance</i> <sup>1</sup> para os gatilhos a serem processados. <b>Obs.:</b> não se deve incluir a seção <b>[ServerNetwork]</b> no servidor do agendador de tarefas, porém é necessário a definição dos servidores escravos no mesmo.
JobServers	Servidores escravos ( <i>slaves</i> ) utilizados para o <i>LoadBalance</i> <sup>1</sup> do agendador de tarefas.
LicenseCompanyId	Código da empresa que será utilizado para o consumo de licença TOTVS.
LicenseUserId	Código do usuário que será utilizado para o consumo de licença TOTVS.

Caso seja utilizado *LoadBalance* para o agendador de tarefas, é possível utilizar os mesmos servidores escravos dos outros servidores de aplicação TOTVS Tec.



<sup>1</sup> Para questões de otimização de recursos de infra-estrutura, recomenda-se utilizar o conceito de *LoadBalance* para distribuição de carga entre servidores Totvstec, que podem estar no mesma máquina ou em máquinas diferentes. Este conceito utiliza-se do paradigma mestre-escravo onde será criado um “servidor mestre” que irá usufruir dos recursos de seus “servidores escravos”. Neste caso, o servidor mestre distribui os processos entre os servidores escravos.

Para cada servidor de aplicação é necessário ter uma cópia do diretório *appserver*. Assim, sugere-se a seguinte estrutura de diretórios para o *LoadBalance*:

- c:\totvs\logix\bin\
- appserver0 → servidor master
- appserver1 → servidor slave 1
- appserver2 → servidor slave 2
- appserverN → servidor slave N

**Nota:** Mais detalhes sobre a configuração de LOAD BALANCE, vide manual de instalação TOTVS Tec.

## Desenvolvimento

Para que seja possível um programa do Logix ser executado através do agendador de tarefas, o mesmo deve ser adaptado conforme os padrões de desenvolvimento para os programas de execução no agendador de tarefas:

1. O processo definido na tarefa do gatilho deve ser desenvolvido no **4GL** com a palavra **\_job** após o nome da função e esta deve sempre receber como parâmetro o nome da rotina (mesmo que não informada no cadastro);
2. O parâmetro “rotina” possibilita a definição de várias rotinas **JOB** em um único arquivo de programa;
3. A função **JOB** deverá obrigatoriamente retornar uma situação, sendo o zero indicador que não houve erros ou diferente de zero caso tenha ocorrido algum erro. Esse retorno define o que será gravado no registro de execução, se o gatilho foi executado com sucesso ou não;
4. Caso tenha a necessidade de utilizar o usuário ou empresa para processamento do programa **JOB**, estes devem ser cadastrados como parâmetro para a tarefa e a função **JOB** deve ser preparada para receber estes parâmetros;
5. Os parâmetros informados para o processo devem ser recuperados dentro da função **JOB** através da função **JOB\_get\_parametro\_gatilho\_tarefa**;
6. O **LOG0010** não fará retorno destes parâmetros quando executado via **JOB**, apenas fará a conexão com o banco de dados e a carga das variáveis de ambiente.

Veja um exemplo de código fonte 4GL preparado para JOB.



```

DATABASE logix

GLOBALS
  DEFINE p_user          LIKE usuarios.cod_usuario
  DEFINE p_cod_empresa  LIKE empresa.cod_empresa
END GLOBALS

#-----#
FUNCTION sssnnnn_job(l_rotina)
#-----#
  DEFINE l_msg          CHAR(100)
  DEFINE l_rotina      CHAR(006)

  IF LOG_initAPP("[CHAVE_SISTEMA_LOGIX]") > 0 THEN
    # Falha na execução do JOB (sem permissão de executar o processo).
    RETURN 1
  END IF

  CASE l_rotina
    WHEN "[NOME_ROTINA_AGENDADOR]"

      # Recupera o parâmetro (simples) 1 de sequencia 0 da tarefa
      # (código da empresa).
      CALL JOB_get_parametro_gatilho_tarefa(1,0)
        RETURNING status, p_cod_empresa

      IF p_cod_empresa IS NULL OR p_cod_empresa = " " THEN
        LET l_msg = "Parâmetro 1 de sequencia 0 (código da empresa) ",
          "não repassado para a tarefa agendada."

        CALL LOG_consoleError(l_msg)
        RETURN 1 # Falha na execução do JOB.
      END IF

      # Quando empresa é recebida como parâmetro no gatilho JOB, deve-se
      # acionar a função abaixo para garantir que todo processamento a
      # partir deste ponto considere esta empresa para processamento.
      # Se a rotina processar dados para várias empresas e
      # conseqüentemente acionar outras aplicações via FGLRUN, deve-se
      # sempre acionar a função abaixo toda vez que precisar considera
      # esta empresa como geral como se fosse o valor atual da
      # variável global P_COD_EMPRESA.
      CALL JOB_setCompany(p_cod_empresa)

      # Recupera o parâmetro (lista) 2 de sequencia 1 da tarefa
      # (código do usuário).
      CALL JOB_get_parametro_gatilho_tarefa(1,0)
        RETURNING status, p_user

      IF p_user IS NULL OR p_user = " " THEN
        LET l_msg = "Parâmetro 2 de sequencia 1 (código do usuário) ",
          "não repassado para a tarefa agendada."

        CALL LOG_consoleError(l_msg)
        RETURN 1 # Falha na execução do JOB.
      END IF

      CALL sssnnnn_processamento()
    END CASE

  RETURN 0 # JOB executado com sucesso.
END FUNCTION

```