

**Documentação**

**Duplicidade nos registros na base online - GCC**

Histórico

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Versão | Data Versão | Descrição | Responsável | Aprovador | Data de Aprovação |
| 1.0 | 13/02/2014 | Criação do Documento | Daniel Bastos |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Informações Gerais

|  |
| --- |
| SOLICITANTE / DEPARTAMENTO |
| **Solicitante:** Norberto Brás |
| **Número do Projeto: CT-65238-W8K4** |
| **Serviço / Aplicação:** GCC |
| SERVIDORES E SCRIPTS ENVOLVIDOS NO PROCESSO |
|  |
| **Ip de Acesso / Host:**  brsphclgcc004 – 10.101.102.101 |
| **Descrição da Máquina:** Servidor Aplicação Web do GCC |
| **Scripts:** RegDuplGCC.sh |
| **Scheduled tasks:** A ser inserido na CRON da maquina 10.101.102.101  **#Verificar duplicidade de dados na base do GCC**  **0 6 \* \* \* /app/scripts/RegDuplGCC.sh** |
|  |
| Descrever de forma sucinta o funcionamento da monitoração e/ou rotina. |
| O script faz uma varredura todo dia às 6hs em uma base de dados remota (SPHDBADM02 – 10.101.104.203) atrás de duplicidades que possam existir nos registros online. São caracterizados dois tipos de buscas nas duplicidades, o que possam estar em Pré-Pagos e os que possam estar em Pós-Pagos.  Se alguma das duas buscas retornarem qualquer tipo de valor é exibido um alarme no Zabbix que dispara um e-mail a central de serviço relatando o erro encontrado e que procedimento seguir. |
| Caso a monitoração envie e-mail de notificação preencher as informações abaixo e anexar a este documento o email padrão. |
| **Notificação VISUAL Tela:** SIM  **Notificação via E-MAIL:** SIM  **Script de envio:** N/A  **Escalonados no email:** N/A  **Para: SUPORTE GCC <suporte.gcc@cleartech.com.br>**  **Cc:** N/A |
|  |
|  |
| **TRIGGERS e ITENS ZABBIX** |
| Itens  Duplicidade Pos-pago  Duplicidade Pre-pago  Triggers  {brsphclgcc004: chk.files.pos.last(0)}#0  {brsphclgcc004: chk.files.pre.last(0)}#0 |
|  |
| COMENTÁRIOS OU INFORMAÇÕES ADICIONAIS |
| **N/A** |

1. Objetivo do Documento

O documento descreve o funcionamento do processo de validação de registros duplicados na base do GCC, que é executado diariamente às 6hrs.

1. Contexto
   1. **– Como funciona o processo de “select” para o banco de dados remoto –Exadata -GCC**

O processo busca registros duplicados no banco de dados Exadata. Por não ter acesso as máquinas Exadatas é utilizado o programa “sqplus” junto com um “select” que faz uma conexão remota no schema pretendido.

1. Detalhes das Atividades dos Fluxos de Processos

**4.1 – Conexão remota e extração das informações.**

Path script criado:

* */app/scripts/RegDuplGCC.sh*

Comandos para conexão remota e extração de registro PÓS-PAGO

* *export ORACLE\_BASE=/usr/lib/oracle/11.2; echo "select sum (QTD) from ( select CTPCNTERMINAL, CTPNROTEL, count (\*) QTD from CCTERMINALPOS group by CTPCNTERMINAL, CTPNROTEL having count (\*) > 1);" | /app/instantclient\_11\_2/sqlplus -s sch\_novogcc/novogcc@//10.101.104.203:1521/gccp | grep "[0-9]" | awk -F" " '{print $1}' > /tmp/pospago*

Comandos para conexão remota e extração de registro PRÉ-PAGO

* *export ORACLE\_BASE=/usr/lib/oracle/11.2; echo "select sum (QTD) from (select APRNUMDDD, APRNUMTEL, count (\*) QTD from CCARQCADPRE group by APRNUMDDD, APRNUMTEL having count (\*) > 1);" | /app/instantclient\_11\_2/sqlplus -s sch\_novogcc/novogcc@//10.101.104.203:1521/gccp | grep "[0-9]" | awk -F" " '{print $1}' > /tmp/prepago*

**4.3 – Monitoração do Zabbix.**

A monitoração no Zabbix é baseada em valores que são colhidos na maquina onde o agente esta instalado. São as *Keys* criadas que aguardam retorno de valor, podendo serem de dois tipos, “0” sendo erro e “1” como sucesso. Quando houver retorno “0” será exibido um alarme com nome do erro no monitor do Zabbix e automaticamente será disparado um e-mail para central de serviço. O e-mail contém procedimentos para esse tipo de alarme.

Arquivo de configuração agent Zabbix:

* */etc/zabbix\_agentd.conf*

Criação da k*ey* no Zabbix-agent que verifica duplicidade de arquivos pre-pago:

* *UserParameter=chk.files.pre*

Criação da k*ey* no Zabbix-agent que verifica duplicidade de arquivos pos-pago:

* *UserParameter=chk.files.pos*

Comando indexado junto a key para valida duplicidade de arquivo pre-pago:

* *cat /tmp/prepago | wc -l*

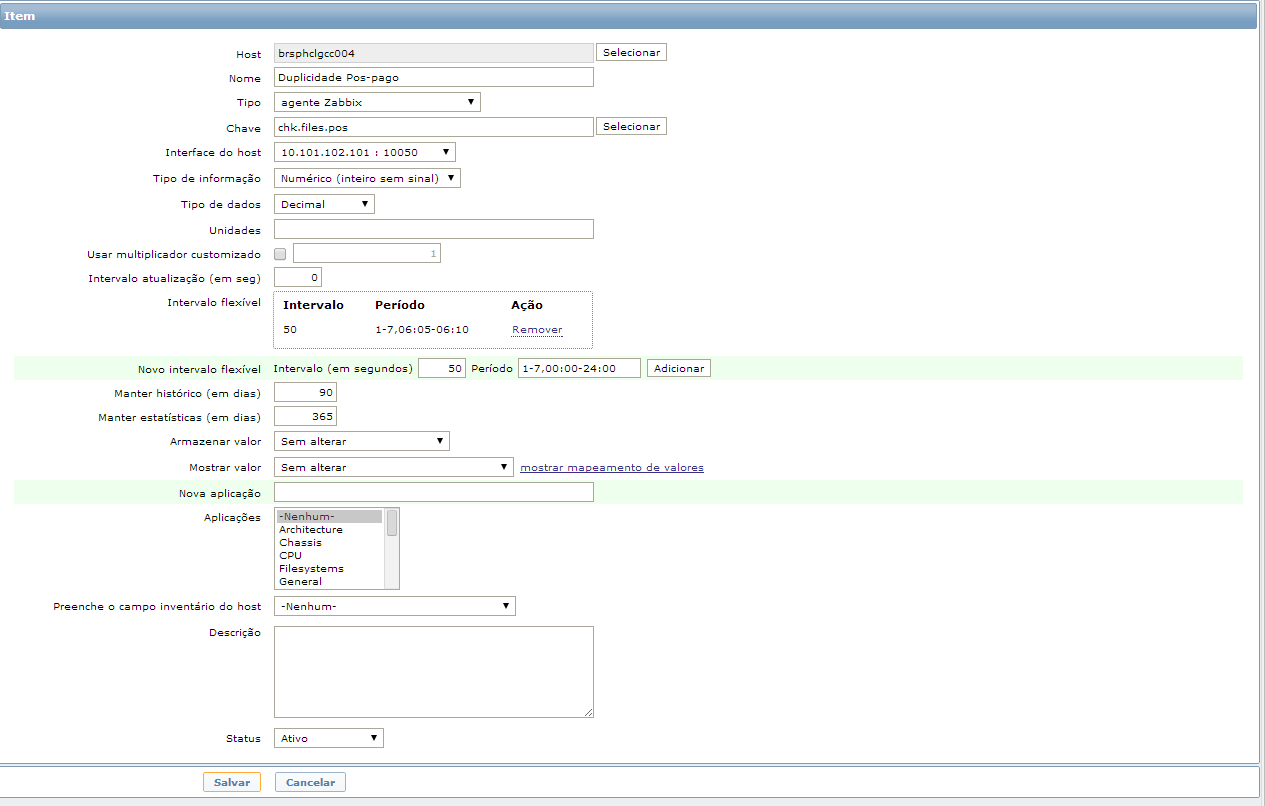
Comando indexado junto a key que valida duplicidade de arquivo pos-pago:

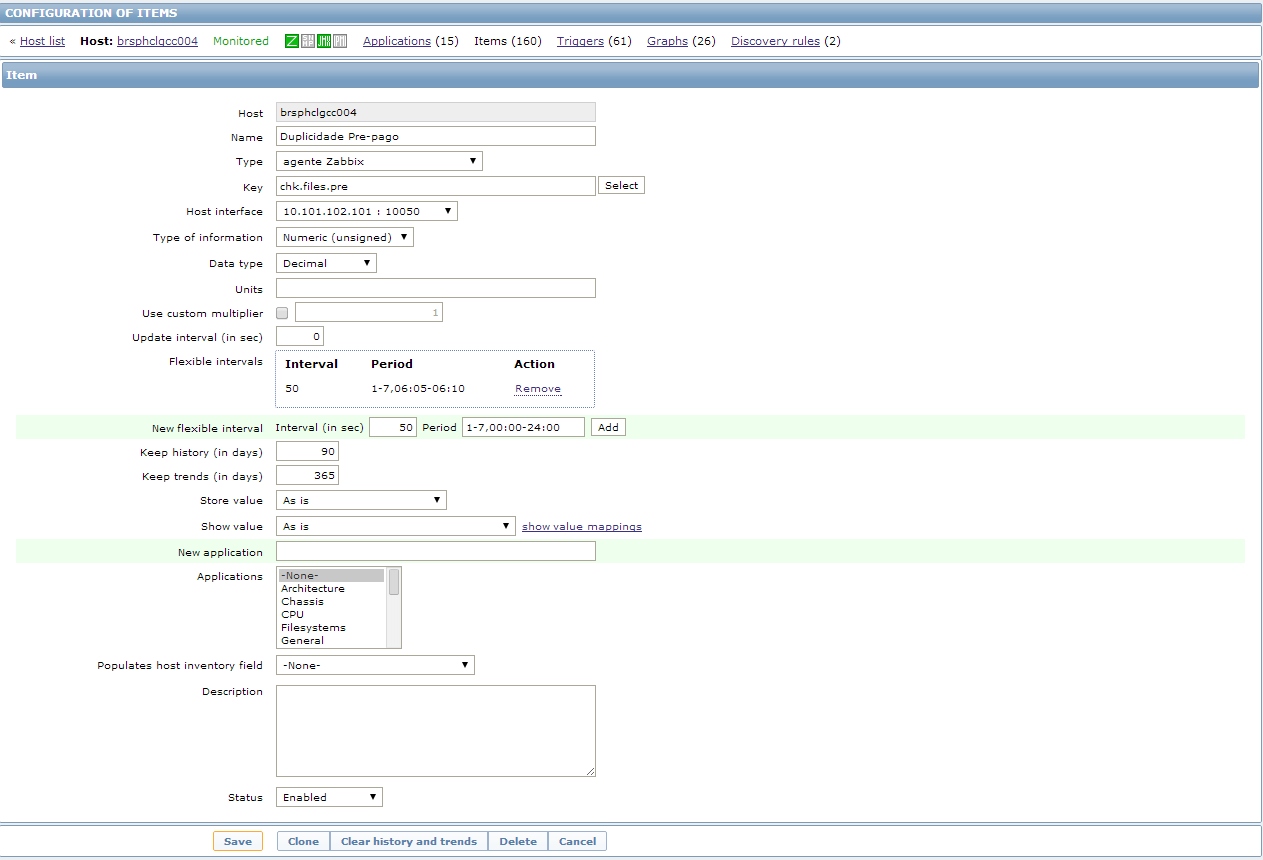
* *cat /tmp/pospago | wc -l*

\*Como o escopo desta documentação não é o Zabbix não será abordado assuntos como, criação de *ITENS, TRIGGERS* e *ACTIONS*.

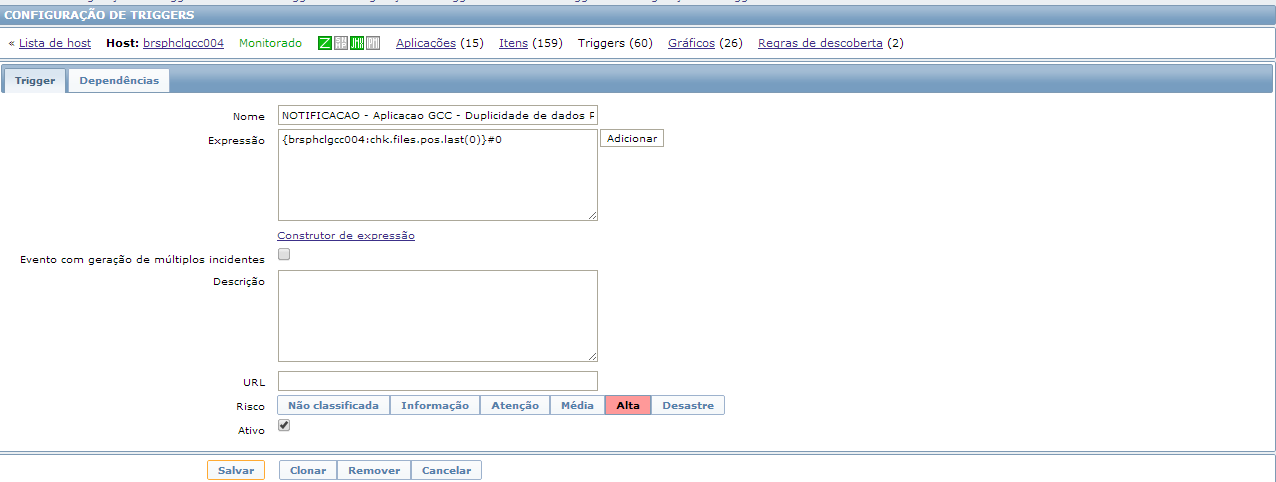
**4.4 – Itens , triggers , actions configurados no Zabbix.**

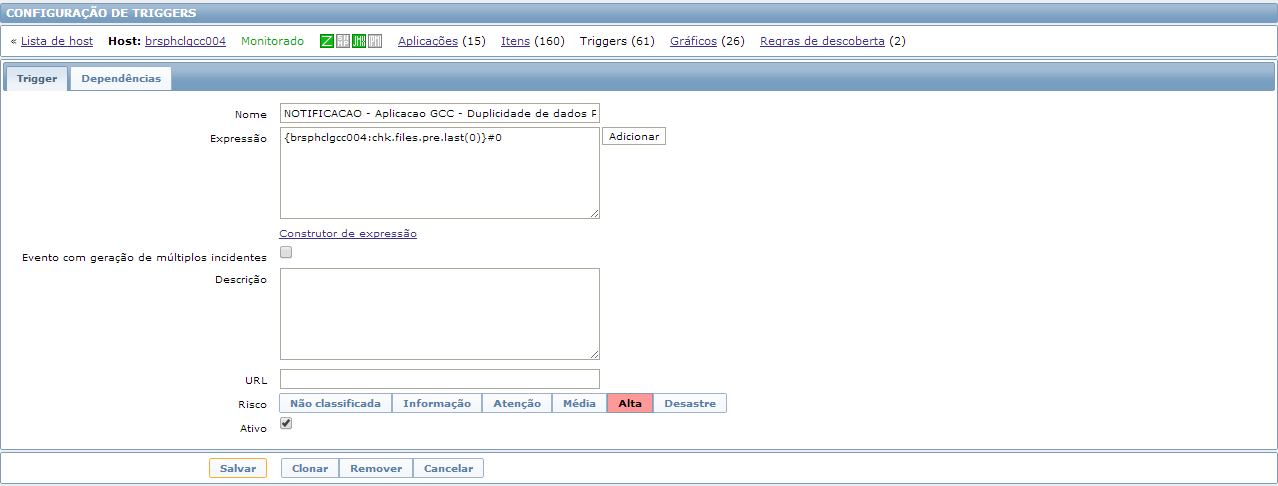
**Itens:**

****

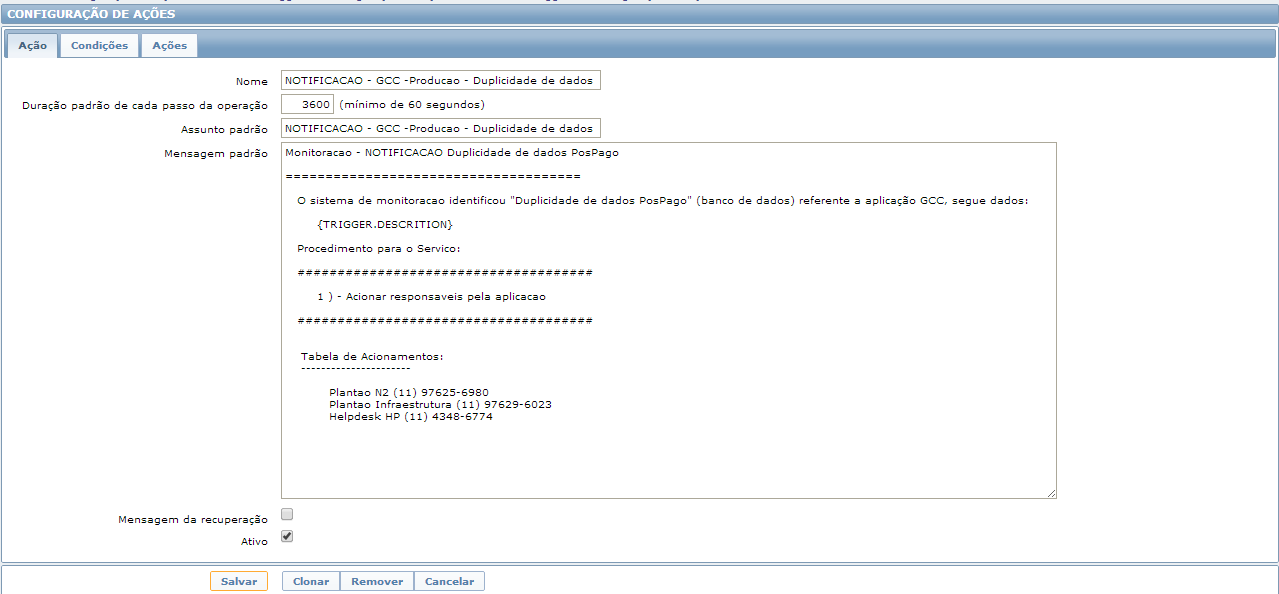


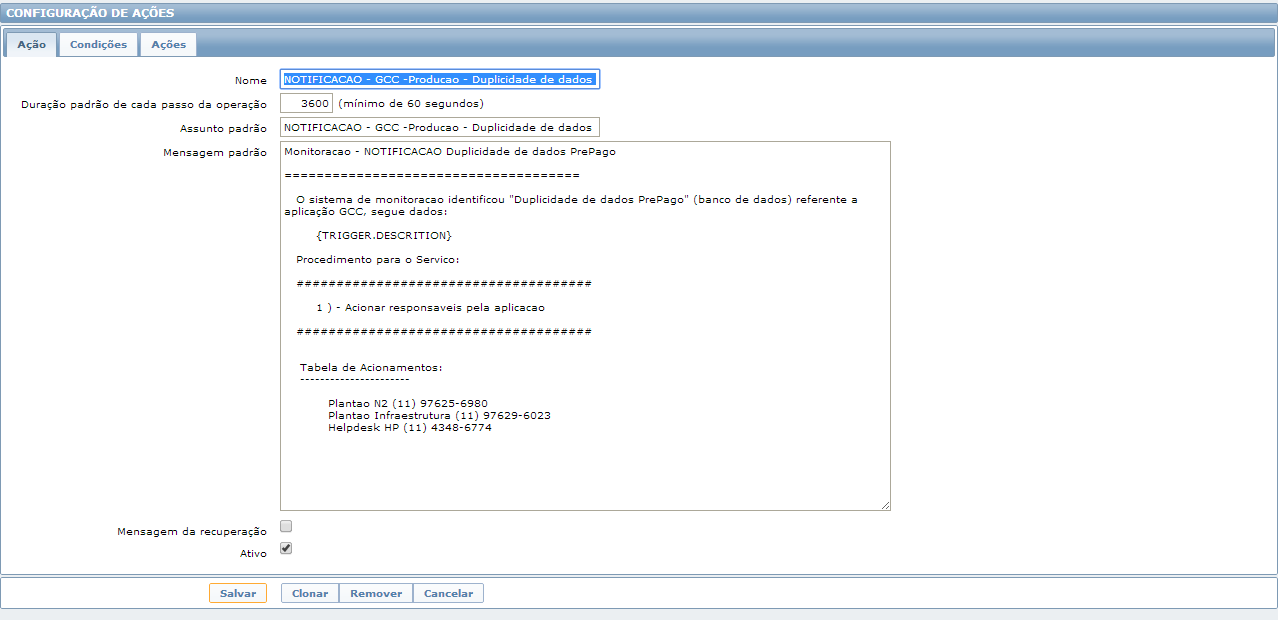
**TRIGGERS:**





**ACTIONS:**





1. Anexos

