API de Integração STOMP

Versão 1.3.1 (14/04/2012)

Vonix Av. Dr. Paulo Machado, 489 Campo Grande MS 79021-300 **T** 67 33180700 **F** 67 33180701 contato@vonix.com.br www.vonix.com.br

Índice

Visão Geral 1

Tipo da Integração entre Sistemas 1

Tipos de mensagens utilizadas 2

Mensagens utilizadas entre Plataforma Vonix e clientes do Sistema integrante 2

Mensagens do tipo “action” (Sistema integrante -> Plataforma Vonix) 2

Mensagens do tipo “event” (Plataforma Vonix -> Sistema integrante) 2

Comunicação e fluxo de entrega das mensagens 2

Clientes 2

Alimentador do discador 2

Dicionário de dados das mensagens utilizadas 1

Mensagens do tipo “action” (Sistema integrante -> Plataforma Vonix) 1

Action: dial 1

Action: transfer 1

Action: hangup 2

Action: pause 2

Action: unpause 3

Action: login 3

Action: logoff 3

Action: tag 4

Action: get_agent_status 4

Action: dialerfeed 5

Mensagens do tipo “event” (Plataforma Vonix -> Sistema integrante) 5

Event: dial 5

Event: dialconnect 6

Event: dialfailure 7

Event: hangup 7

Event: called 8

Event: calledsuccess 8

Event: calledfailure 9

Event: pause 9

Event: unpause 10

Event: login 10

Event: logoff 10

Event: agent_status 11

Event: dialerresult 11

Aplicação aos Casos de Uso 1

Exemplo 1. Agente discando através do softphone 1

Exemplo 2. Discagem a partir do sistema integrante (“discagem por click”) 3

Exemplo 3. Entrada de chamada receptiva 5

Exemplo 4. Discagem através do discador automático 7

Anexo 1: Tabelas 1

Tabela 1 - Motivos de Pausa 1

Tabela 2 - Códigos de Causas de Término de Chamada 1

Tabela 3 - Estados dos Agentes 2

Visão Geral

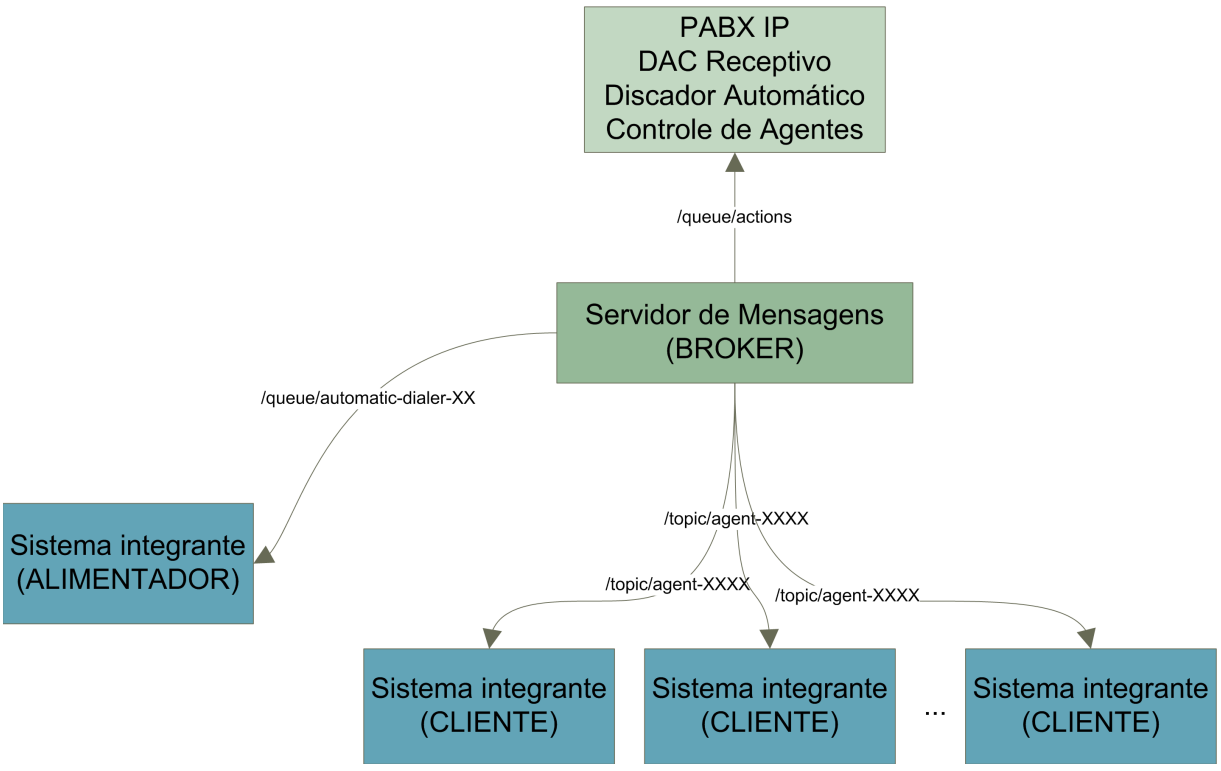
Tipo da Integração entre Sistemas

A integração dos sistemas se dá através de um serviço de messaging, utilizando-se um broker (servidor de mensagens), que se comunica através do protocolo STOMP (http://stomp.codehaus.org/Stomp/) para o transporte das mensagens. O conteúdo destas mensagens será em formato XML especificado aqui. Serão utilizadas filas e tópicos para o controle de entrega das mensagens.

Cada "**cliente**" do software integrante (pode ser um programa de desktop ou um browser no caso de sistema via web) fará uma única conexão ao "**broker**" (servidor de mensagens) para envio e recepção das mensagens da integração.

No caso de uso de discador automático (preditivo, serial, fonação, ura reversa, etc) na operação, deve existir um programa rodando no servidor do sistema integrante que alimentará o discador com a lista de clientes a serem discados de forma automática (independente do agente). Ele será chamado de "**alimentador**". Esta alimentação, assim como qualquer outra comunicação entre os sistemas, se dará através de mensagens, sendo que cada mensagem conterá um contato para discagem, cada contato podendo conter um ou mais telefones.

O envio das mensagens com os contatos pode ser sequencial, em blocos com várias mensagens juntas ou da forma que o sistema integrante entender ser melhor. O discador automático então, seguindo todas as regras de completamento (número de troncos para cada campanha, regras de custo, horário, balanceamento de tráfego, etc), fará as discagens em paralelo. Para todo contato alimentado ao discador, uma mensagem de retorno será enviada para o programa alimentador com o estado final do contato (conseguiu contato, não conseguiu contato, ou conseguiu contato, mas não tinha agentes para atendê-lo no momento, etc). Além disso, quando um dos números do contato atender a chamada, o Plataforma Vonix enviará a mensagem de entrada de chamada ao cliente do agente que está recebendo a chamada, para que o sistema integrante abra a ficha do cliente naquele momento, por exemplo.



Tipos de mensagens utilizadas

Serão usados dois tipos de mensagens.

O primeiro tipo, chamado de "**event**", são mensagens disparadas nos eventos específicos pela Plataforma Vonix para o sistema integrante, seja para consumo no cliente ou no alimentador (ex: agente iniciou uma discagem, entrou uma chamada receptiva para o agente, agente entrou em pausa, a chamada foi completada, um contato enviado ao discador foi concluído, etc).

O segundo são as mensagens chamadas de "**action**", que são publicadas pelo sistema integrante para a Plataforma Vonix, pedindo a execução de alguma ação (discar para algum número, pausar o agente, bloquear o agente a receber chamadas, alimentar um contato ao discador preditivo, etc).

Mensagens utilizadas entre Plataforma Vonix e clientes do Sistema integrante

Mensagens do tipo “action” (Sistema integrante -> Plataforma Vonix)

As mensagens do tipo action são publicadas pelo sistema integrante em determinados momentos, para requisitar uma ação da Plataforma Vonix. Estas mensagens sempre serão destinadas a fila de nome “`/queue/actions`”, vindas de qualquer cliente conectado ou do alimentador. A Plataforma Vonix é o único assinante desta fila e processa a mensagem assim que recebida.

Cada mensagem do tipo “action” deve vir com uma string identificadora única, citada abaixo, na especificação do pacote.

Mensagens do tipo “event” (Plataforma Vonix -> Sistema integrante)

As mensagens do tipo event são publicadas pela Plataforma Vonix para o sistema integrante na ocorrência de determinados eventos que ocorrerem, como discagem, desligamento, completamento, etc.

Estas mensagens são enviadas em tópicos exclusivos para cada agente, no caso de serem eventos para aquele agente (nome do tópico é “`/topic/agent-MATRICULA`", ou na fila de resposta do alimentador, no caso de ser uma resposta a ele (nome do tópico é “`/queue/automtic-dialer-IDFEEDER`”.

Comunicação e fluxo de entrega das mensagens

Clientes

Os clientes do sistema integrante se conectarão ao broker assim que o atendente se logar no sistema integrante e estiver pronto para receber ou originar chamadas. O cliente consome o tópico chamado "`/topic/agent-XXXX`", sendo `XXXX` a matrícula do agente junto a Plataforma Vonix (ex: no caso do agente 1005, todos os eventos relacionados com ele serão enviados para `/topic/agent-1005`). É neste tópico que o cliente receberá os eventos pertinentes a ele.

O Plataforma Vonix disparará mensagens do tipo “**event**” neste tópico para todo evento que tem algum interesse ao sistema integrante.

Para pedidos de ações do cliente, o sistema integrante publicará mensagens de ações para o sistema de telefonia através de uma fila única chamada "`/queue/actions`". Assim que a Plataforma Vonix receber as mensagens do tipo "**action**", executará a ação. E os eventos gerados por esta ação serão enviados no tópico do agente.

O cliente utilizará a mesma conexão para receber e enviar mensagens, não sendo necessário qualquer outro procedimento.

Alimentador do discador

O alimentador do discador se conectará ao broker e ficará conectado durante todo o período. O alimentador consome a fila chamada “`/queue/automatic-dialer-IDFEEDER`”, onde `IDFEEDER` é um identificador único para aquele alimentador. Pode ser qualquer string, com letras e números. Esta string identifica o alimentador para o caso de mais de um

alimentador jogando contatos para o discador. É nesta fila que ele receberá os retornos de todos os contatos enviados, seja ele qual for (sucesso, falha, abandono, etc). Para cada contato enviado para discagem, o alimentador necessariamente enviará uma mensagem de retorno. A regra é sempre para cada pacote enviado com um contato, um pacote de retorno deste contato com as tentativas de discagem (independente de quantos forem os telefones daquele contato, o resultado virá em um único pacote, com todos os telefones discados).

Para envio dos contatos, o alimentador publicará as mensagens na fila "`/queue/actions`", com um contato por mensagem.

Dicionário de dados das mensagens utilizadas

Mensagens do tipo “action” (Sistema integrante -> Plataforma Vonix)

Action: dial
Pedido de inicio de uma chamada direta para determinado agente logado.

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que está iniciando a chamada
to: número de destino da chamada no formato DDD+NUMERO, ex: 1130136400
name: nome do cliente destino da chamada
queue: ID da fila que sairá a chamada, caso em múltiplas filas (pode ser passado em branco para fila padrão do agente)
billing_group_id: numero identificador do centro de custo da fila desta chamada

Exemplo:

```
<action name="dial">
  <id>7839</id>
  <agent>1003</agent>
  <to>1133180758</to>
  <name>Joao Jose da Silva</name>
  <queue></queue>
  <billing_group_id>2</billing_group_id>
</action>
```

Action: transfer
Pedido para transferir uma chamada em andamento para determinado destino

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente

agent: matricula do agente que está com a chamada
call_id: id da chamada em curso para transferência
to: número de destino para transferência da chamada

Exemplo:

```
<action name="transfer">
  <id>7839</id>
  <agent>1003</agent>
  <call_id>336681213.1311</call_id>
  <to>0755</to>
</action>
```

Action: hangup

Pedido para desligar uma chamada em andamento

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que está com a chamada
call_id: id da chamada em curso para transferência

Exemplo:

```
<action name="hangup">
  <id>7839</id>
  <agent>1003</agent>
  <call_id>336681213.1311</call_id>
</action>
```

Action: pause

Pedido de pausa para determinado agente logado.

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que deve entrar em pausa
reason: numero identificador da razão da pausa (vide tabela 1)

Exemplo:

```
<action name="pause">
  <id>7839</id>
```

```
  <agent>1003</agent>
  <reason>3</reason>
</action>
```

Action: unpause
Pedido de saída de pausa para determinado agente logado.

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que deve ser tirado de pausa

Exemplo:
```
<action name="unpause">
  <id>7839</id>
  <agent>1003</agent>
</action>
```

Action: login
Pedido de deslogar determinado agente logado.

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que deve ser deslogado
mac_address: MAC address da estação onde está o VonixPhone (separados por vírgula caso múltiplos)

Exemplo:
```
<action name="login">
  <id>7839</id>
  <agent>1003</agent>
  <mac_address>00:12:34:56:77:21</mac_address>
</action>
```

Action: logoff
Pedido de deslogar determinado agente logado.

Remetente:

Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que deve ser deslogado

Exemplo:
```
<action name="logoff">
  <id>7839</id>
  <agent>1003</agent>
</action>
```

Action: tag
Vincular uma tag a uma chamada (etiquetar a chamada).

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
call_id: número identificador da chamada, é o ID único que identifica a chamada em todas as mensagens da integração (atributo id que vem em todas as mensagens, como called, calledsuccess, dial, hangup, etc)
tag: uma string, ou seja, palavra ou frase que identifique a chamada. Esta é a etiqueta em si.

Exemplo:
```
<action name="tag">
  <id>7839</id>
  <call_id>336681213.1311</call_id>
  <tag>Acordo</tag>
</action>
```

Action: get_agent_status
Pedido de travamento do agente para início de chamada agendada

Remetente:
Cliente do sistema integrante

Para onde é enviado:
/queue/actions

Tags:
id: número identificador do pacote gerado pelo remetente
agent: matricula do agente que está iniciando a chamada
queue: ID da fila para status, caso de agente em múltiplas filas (pode ser passado em branco para fila padrão do agente)

Exemplo:

```
<action name="get_agent_status">
  <id>7839</id>
  <agent>1003</agent>
  <queue></queue>
</action>
```

Action: dialerfeed

Alimentação do discador com um cliente a ser discado preditivamente. Cada pacote é um contato, com vários telefones.

Remetente:

Programa alimentador do discador preditivo

Para onde é enviado:

/queue/actions

Tags:

id: número identificador do pacote gerado pelo remetente
to: números de destino da chamada. Os números devem estar dentro de uma tag “number”, com atributos id, que é o número identificador do número telefônico; type, que é o tipo da chamada (textual) e priority, que é a prioridade de tentativa entre os números (quanto menor a o número, maior a prioridade de discagem dele). Dentro da tag “number” deve vir o número telefônico no formato DDD+NUMERO, ex: 1130136400, com 10 dígitos.
contact_name: nome do contato deste pacote
queue: nome da fila de trabalho
billing_group_id: numero identificador do centro de custo da fila desta chamada

Exemplo:

```
<action name="dialerfeed">
  <id>799506</id>
  <to>
    <number id="1" type="RESIDENCIAL" priority="1">1144418628</number>
    <number id="2" type="CELULAR" priority="2">1174530234</number>
  </to>
  <contact_name>Frederico Mao Nervosa</contact_name>
  <queue>cobranca</queue>
  <billing_group_id>1</billing_group_id>
</action>
```

Mensagens do tipo “event” (Plataforma Vonix -> Sistema integrante)

Event: dial

Mensagem enviada assim que o agente inicia uma chamada ativa. Este evento é disparado em qualquer discagem iniciada pelo agente, diretamente pelo ramal (softphone) ou discando pelo sistema integrante (clique de discagem automática).

Remetente:

Plataforma Vonix

Para onde é enviado:

/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
agent: matricula do agente que está iniciando a chamada
queue: ID da fila desta chamada
from: ramal que está iniciando a chamada (onde se encontra o agente)
to: número de destino da chamada no formato DDD+NUMERO, ex: 1130136400
callfilename: nome do arquivo de gravação desta chamada
action_id: esta tag existe somente no caso do evento ter sido gerado por um action do sistema. É o numero indentificador do pacote action de pedido do evento
contact_name: esta tag existe somente no caso do evento ter sido gerado por um action do sistema. É o nome do cliente de destino que consta no action de pedido do evento

Exemplo:
```
<event name="dial">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <agent>1003</agent>
  <queue>carrefour</queue>
  <from>0758</from>
  <to>6733180758</to>
  <callfilename>336681213.1311.WAV</callfilename>
  <action_id>46875</action_id>
  <contact_name>Joao Alves da Silva</contact_name>
</event>
```

Event: dialconnect
Mensagem enviada assim que a chamada originada pelo agente (diretamente por discagem de clique no sistema ou manualmente pelo softphone) é atendida.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)

Exemplo:
```
<event name="dialconnect">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

Event: dialfailure
Mensagem enviada no caso de não-completamento da chamada originada pelo agente (diretamente por discagem de clique no sistema ou manualmente pelo softphone) pelo agente.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
cause: número identificador da causa de insucesso da chamada (vide tabela 2)
cause-txt: descrição da causa de insucesso da chamada

Exemplo:
```
<event name="dialfailure">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <cause>1</cause>
  <cause-txt>Numero vago</cause-txt>
</event>
```

Event: hangup
Mensagem enviada assim que um agente desliga uma chamada em curso (qualquer tipo de chamada).

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
cause: número identificador da causa de insucesso da chamada (vide tabela 2)
cause-txt: descrição da causa de insucesso da chamada
agent: matricula do agente que está iniciando a chamada

Exemplo:
```
<event name="hangup">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <cause>16</cause>
  <cause-txt>Desconexão normal</cause-txt>
  <agent>1003</agent>
</event>
```

Event: called

Mensagem enviada assim que o agente receber uma chamada (externa receptiva ou chamada vinda do discador preditivo).

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
queue: nome da fila de trabalho
from: ramal que está iniciando a chamada (onde se encontra o agente)
to: número de destino da chamada no formato DDD+NUMERO, ex: 1130136400
agent: matricula do agente que está iniciando a chamada
callfilename: nome do arquivo de gravação desta chamada
action_id: esta tag existe somente no caso do evento ter sido gerado por um action do sistema. É o numero indentificador do pacote action de pedido do evento
data: esta tag existe somente no caso da chamada ter passado por atendimento automático de URA anteriormente, e o cliente ter digitado informações durante este auto-atendimento. Contém os dados internos do sistema digitados pelo usuário anteriormente, em atendimento automático (URA), como id do cliente, matricula, opções que passou, etc.

Exemplo:

```
<event name="called">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <queue>suporte</queue>
  <from>33180758</from>
  <to>0758</to>
  <agent>1003</agent>
  <callfilename>336681213.1311.WAV</callfilename>
  <action_id>46875</action_id>
  <data>
    <clientid>177954810</clientid>
  </data>
</event>
```

Event: calledsuccess

Mensagem enviada assim que o agente atender a chamada (externa receptiva ou chamada vinda do discador preditivo) oferecida para ele.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente

date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)

Exemplo:

```
<event name='calledsuccess'>
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

Event: calledfailure

Mensagem enviada caso o agente não atenda a chamada (externa receptiva ou chamada vinda do discador preditivo) oferecida para ele. Enviado no momento do oferecimento para o próximo agente ou no abandono por parte de quem está ligando.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)

Exemplo:

```
<event name='calledfailure'>
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

Event: pause

Mensagem enviada assim que o agente entra em pausa.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
reason: código de motivo da pausa

Exemplo:

```
<event name="pause">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <reason>2</reason>
</event>
```

Event: unpause
Mensagem enviada assim que o agente sai de pausa (volta a fila de trabalho).

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
reason: código de motivo da pausa
pause_seconds: tempo em segundos que o agente permaneceu em pausa

Exemplo:
```
<event name="unpause">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <reason>1003</reason>
  <pause_seconds>365</pause_seconds>
</event>
```

Event: login
Mensagem enviada assim que o agente se loga no sistema.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
location: número do ramal em que o agente se logou

Exemplo:
```
<event name="login">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <location>0710</location>
</event>
```

Event: logoff
Mensagem enviada assim que o agente se desloga do sistema.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
id: número identificador do pacote gerado pelo remetente
date: data do envio do pacote em formato (DD/MM/AAAA HH:MM:SS)
location: número do ramal em que o agente estava logado
duration: tempo em segundos que o agente permaneceu logado

Exemplo:

```
<event name="logoff">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <location>0710</location>
  <duration>3492</duration>
</event>
```

Event: agent_status
Mensagem enviada como resposta ao pedido de status do agente. É enviado assim que o sistema recebe um pedido de status através do pacote `get_agent_status`.

Remetente:
Plataforma Vonix

Para onde é enviado:
/topic/agent-XXXX

Tags:
action_id: número identificador do pacote gerado pelo remetente
status: estado do agente (conforme tabela 3, Anexo 1)
location: número do ramal em que o agente está logado

Exemplo:

```
<event name=“agent_status">
  <action_id>PEDIDO1234</action_id>
  <status>ONLINE</status>
  <location>0710</location>
</event>
```

Event: dialerresult
Mensagem enviada como retorno de cada pacote enviado ao discador pelo pacote “dialerfeed”. Cada pacote gerará um retorno.

Remetente:
Plataforma Vonix

Para onde é enviado:
/queue/automatic-dialer

Tags:

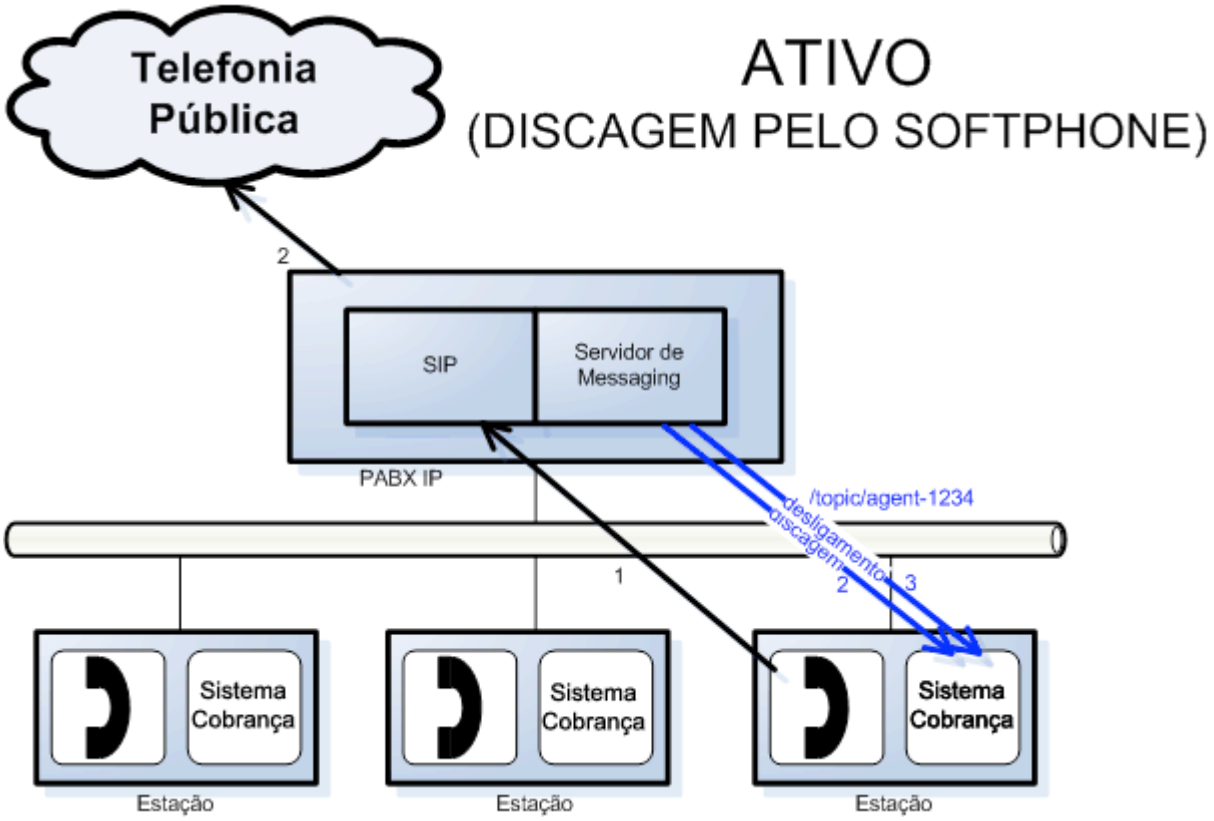
id: número identificador do pacote gerado pelo remetente
queue: nome da fila de trabalho
result: resultado das tentativas de discagem. O atributo status pode vir com “success” no caso de de ter conseguido atendimento humano com um dos telefones, “ivrsuccess” no caso de nenhum telefone ter sido atendido por um humano, mas o sistema ter deixado recado na caixa postal de um ou mais telefones, ou “failure” no caso de não ter conseguido contato em nenhum dos telefones do cliente. O retorno se dará número a número dos tentados, com o número identificador da causa dentro da tag “number” identificada (vide tabela 2)

Exemplo:

```
<event name='dialerresult'>
  <id>7839</id>
  <queue>suporte</queue>
  <result status="success"> (**)
    <number id="1001" dialed_at="16/04/2012 14:41:46" hangup_at="16/04/2012 14:41:48">19</number>
    <number id="1002" duration="5" dialed_at="17/04/2012 14:19:24" hangup_at="17/04/2012 14:19:36"
answered_at="17/04/2012 14:19:32">16</number>
    <number id="1003">0</number>
  </result>
</event>
```

Aplicação aos Casos de Uso

Exemplo 1. Agente discando através do softphone



Assim que o agente efetuar uma chamada, o Plataforma Vonix vai mandar uma mensagem para "/topic/agent-XXXX" para que o sistema integrante (se desejar) registre esta chamada, com os dados: ID da chamada, data da chamada, ramal de origem, número de destino, nome do arquivo de gravação da chamada, agente que está fazendo a chamada e nome da fila de trabalho do agente.

Segue modelo da mensagem abaixo:

```
<event name="dial">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <from>0758</from>
  <to>33180758</to>
  <callfilename>336681213.1311.WAV</callfilename>
  <agent>1003</agent>
  <queue>suporte</queue>
</event>
```

Quando (e se) a chamada for atendida, será enviada a seguinte mensagem:

```
<event name="dialconnect">
  <id>336681213.1311</id>
  <date>23/08/2007 18:44:14</date>
</event>
```

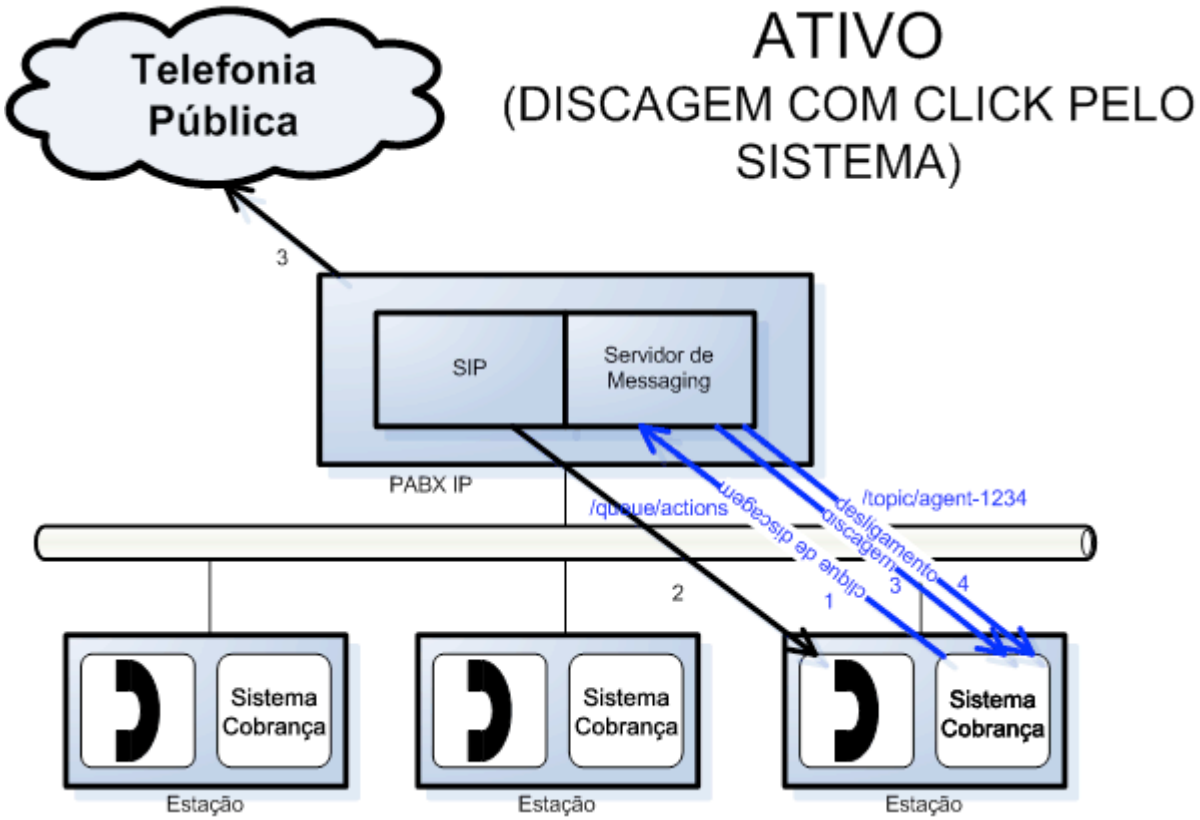
Quando essa chamada terminar, será enviada a mensagem com os dados do encerramento da chamada: ID da chamada, data do encerramento, causa do encerramento (código e descritivo) e agente. Segue modelo da mensagem abaixo:

```
<event name="hangup">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <cause>16</cause>
  <cause-description>Normal</cause-description>
  <agent>1003</agent>
</event>
```

Caso a chamada não seja atendida, será enviada a seguinte mensagem:

```
<event name="dialfailure">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:27</date>
  <cause>1</cause>
  <cause-txt>Numero vago</cause-txt>
</event>
```

Exemplo 2. Discagem a partir do sistema integrante (“discagem por click”)



Este cenário é a possibilidade de originar chamadas clicando dentro do sistema integrante.

Assim que o agente clicar no número de telefone dentro do sistema, este enviará uma mensagem para "/queue/actions", com o seguinte formato (agente e número a ser discado):

```
<action name="dial">
  <id>783913</id>
  <agent>1003</agent>
  <to>6733180758</to>
  <name>Joao Jose da Silva</name>
  <queue>carrefour</queue>
  <billing_group_id>2</billing_group_id>
</action>
```

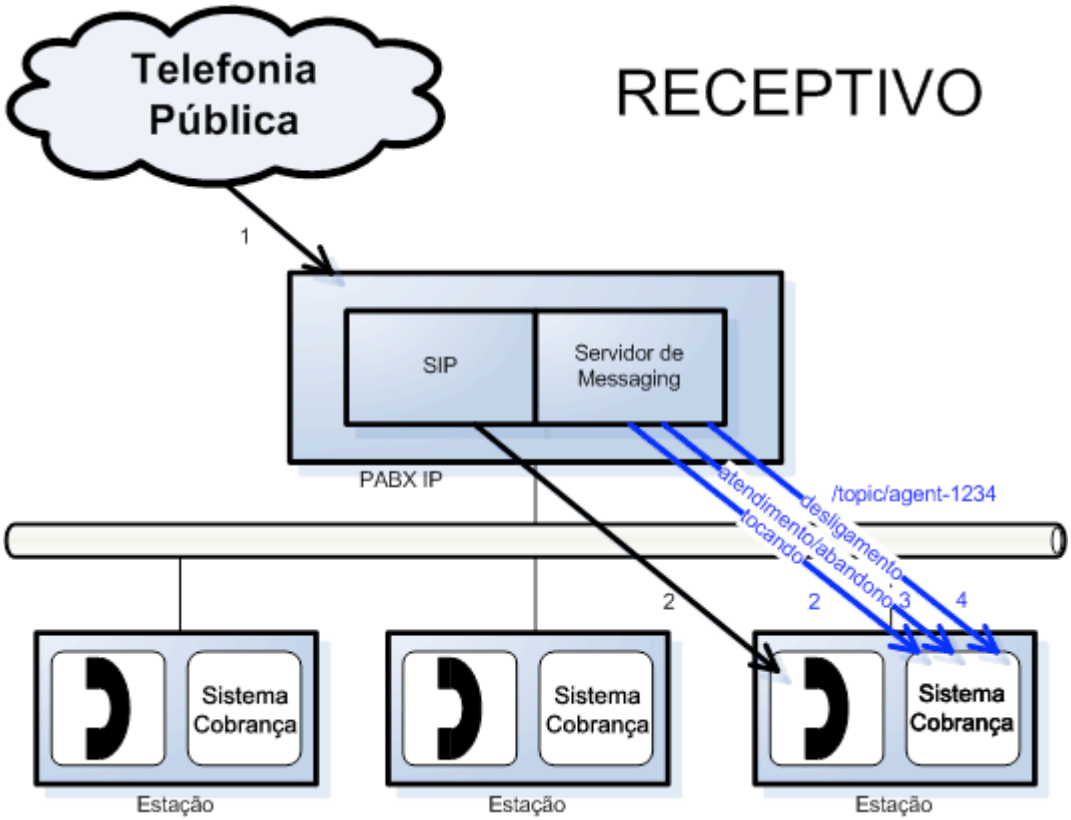
O Plataforma Vonix então vai discar automaticamente para o número pedido. Daí pra frente, segue-se o mesmo processo. O Plataforma Vonix enviará a mesma mensagem de discagem para "/topic/agent-XXXX" com os dados da chamada, como em todo os casos que o agente discar algum número. Segue modelo da mensagem abaixo:

```
<event name="dial">
  <id>336681215.1321</id>
  <action_id>783913</action_id>
  <date>23/08/2007 18:45:23</date>
  <from>0758</from>
  <to>6733180758</to>
  <callfilename>336681215.1321.WAV</callfilename>
```

```
  <agent>1003</agent>
  <queue>carrefour</queue>
</event>
```

Além disso, todas as outras mensagens do fluxo seguirão da mesma forma do primeiro caso, citado acima (eventos “dialconnect” e depois “hangup” no caso de completamento ou “dialfailure” no caso de não-completamento).

Exemplo 3. Entrada de chamada receptiva



Assim que o Plataforma Vonix oferecer uma chamada recebida que está na fila de espera para um agente disponível, será enviada uma mensagem "event" com o nome de "called", para que o sistema possa abrir a tela do cliente de antemão (ou tomar qualquer outra ação). A mensagem será enviada para "/topic/agent-XXXX" com os dados: ID da chamada, data da chamada, ramal de origem, número de destino, nome do arquivo de gravação da chamada, agente que está fazendo a chamada e nome da fila de trabalho do agente.

Segue modelo da mensagem abaixo:

```
<event name="called">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <from>33180758</from>
  <to>0758</to>
  <callfilename>336681213.1311.WAV</callfilename>
  <agent>1003</agent>
  <queue>suporte</queue>
</event>
```

O Plataforma Vonix então mandará uma mensagem de atendimento ou não-atendimento em seguida. Isto porque o agente pode atender ou não a chamada (ele pode ter passado a chamada para o próximo agente, ele pode não ter atendido em até 15 segundos e o PABX passou para o proximo automaticamente, o cliente pode ter desistido antes dele ter tempo hábil para atender, etc) São duas as mensagens que podem ser enviadas neste momento.

Isto serve, por exemplo, para que o sistema integrante feche a ficha do cliente no caso do agente não atender a chamada. Além disso, pode-se criar historico na interação que a chamada passou por mais de um agente.

No caso de atendimento:

```
<event name='calledsuccess'>
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

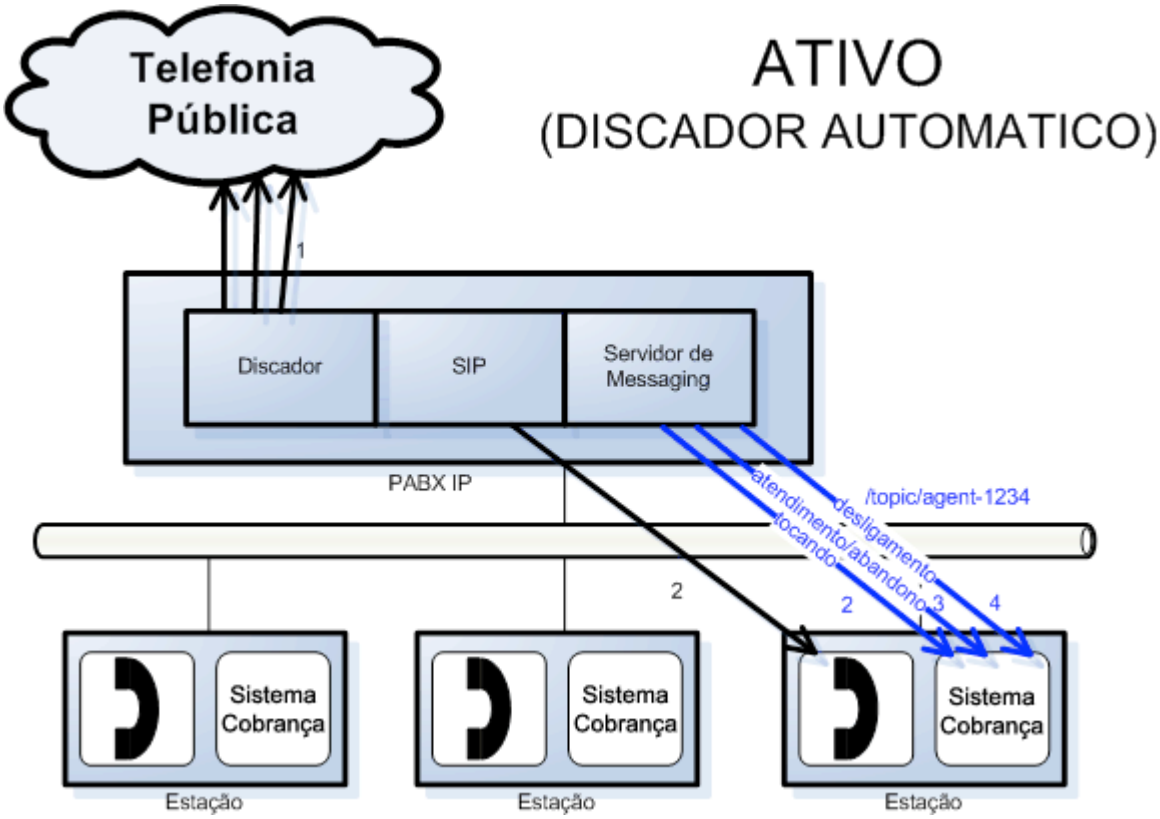
No caso de não-atendimento:

```
<event name='calledfailure'>
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

Além disso, assim que essa chamada terminar, o Plataforma Vonix mandará a mesma mensagem de finalização da chamada, como nos casos acima:

```
<event name="hangup">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <cause>16</cause>
  <cause-txt>normal-blablabla</cause-txt>
  <agent>1003</agent>
</event>
```

Exemplo 4. Discagem através do discador automático



Um alimentador instalado no servidor enviará através de mensagens na fila “/queue/actions” os contatos para discagem dentro dos pacotes do tipo “dialerfeed”. O discador automático então iniciará o trabalho na fila solicitada seguindo todos os critérios daquela fila específica.

Assim que uma chamada discada pelo discador automático for atendida, ela será automaticamente encaminhada ao agente da vez, baseado nos critérios de distribuição da fila.

Segue-se então o mesmo fluxo do cenário receptivo. O Plataforma Vonix enviará uma mensagem do tipo “called” ao agente que recebeu o oferecimento desta chamada do discador. Mesmo sendo uma chamada de discador, o agente pode não atender, gerando uma mensagem do tipo “calledfailure”. Caso atenda, o evento “calledsuccess” será enviado, e ao final do atendimento, o evento “hangup”.

O que diferencia a chamada de ser receptiva ou discada pelo discador é o fato da chamada originada por pedido do discador trazer a tag <action_id> com o id do pacote “dialerfeed”são critérios como a fila de trabalho, se você mandou a chamada de antemão ou não, etc.

Além disso, uma mensagem de retorno das tentativas de um cliente, seja ela de sucesso ou erro, serão enviados no tópico “/queue/automatic-dialer”, a ser monitorado pelo alimentador do discador. Esta mensagem será a cada cliente discado, indepedentemente do fluxo de atendimento.

Segue um exemplo de fluxo de uma operação de discagem preditiva:

O sistema integrante gera pacotes de alimentação com clientes para serem discados, em uma ou mais filas.

```
<action name="dialerfeed">
  <id>799506</id>
  <to>
```

```
    <number id="1" type="RESIDENCIAL" priority="1">1144418628</number>
    <number id="2" type="CELULAR" priority="2">1174530234</number>
  </to>
  <contact_name>Frederico Mao Nervosa</contact_name>
  <queue>cobranca</queue>
  <billing_group_id>1</billing_group_id>
</action>
```

O discador inicia o trabalho.

Caso algum dos números do cliente for atendido, ela é encaminhada a um agente da fila. Gera-se então uma mensagem para o cliente que está recebendo a chamada (no "/topic/agent-XXXX").

```
<event name="called">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <from>33180758</from>
  <to>0758</to>
  <callfilename>336681213.1311.WAV</callfilename>
  <agent>1003</agent>
  <queue>suporte</queue>
</event>
```

O agente atende a chamada (em discagem preditiva, o agente sempre fica em modo auto-atendimento, atendendo todas as chamadas imediatamente). Gera-se o evento "calledsuccess".

```
<event name='calledsuccess'>
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
</event>
```

O atendimento ocorre.

Assim que a chamada é desligada, gera-se normalmente o evento "hangup" no tópico do agente (enviado para o cliente). Neste momento, gera-se o pacote "dialerresult" com o retorno do contato do cliente para o alimentador do discador.

Para o cliente em que o agente está trabalhando:

```
<event name="hangup">
  <id>336681213.1311</id>
  <date>23/08/2007 18:43:24</date>
  <cause>16</cause>
  <cause-txt>normal-blablabla</cause-txt>
  <agent>1003</agent>
</event>
```

Para o alimentador:

```
<event name='dialerresult'>
  <id>7839</id>
  <result status=”success”>
    <number id="1">16</number>
```

```
  </result>
</event>
```

No caso de não-atendimento de nenhum dos números passados no pacote dialerfeed, o único evento que ocorre é uma mensagem “dialerresult” para o alimentador, já que, neste caso, não houve envolvimento de nenhum agente na tentativa da chamada.

```
<event name='dialerresult'>
  <id>7839</id>
  <result status=”failure”>
    <number id="1">18</number>
    <number id="2">1</number>
    <number id="3">18</number>
  </result>
</event>
```

Anexo 1: Tabelas

Tabela 1 - Motivos de Pausa

ID	Motivo
1	Avaliação
2	Treinamento
3	Lanche
4	Toalete
5	Ginástica Laboral
6	Erro no Sistema

Tabela 2 - Códigos de Causas de Término de Chamada

ID	Causa de Desligamento
0	Cancelado
1	Número Vago
2	Sem Rota Operadora
3	Sem Rota Destino
4	Número com Problema
5	Inclusão DDD Errada
6	Canal Inválido
8	Cancelado Operadora
16	Atendida (desligamento normal)
17	Ocupado
18	Não Responde
19	Não Responde
20	Fora de Área
21	Chamada Rejeitada pela Operadora
22	Número Mudou
23	A Cobrar Rejeitado
26	Término Inesperado
27	Destino com Defeito
28	Formato Inválido
29	Serviço Indisponível
31	Desconectado da Rede
34	Sem Canal Disponível
38	Falha de Rede
41	Falha Temporária
43	Negado pela Rede
53	Destino Bloqueado
55	Destino Bloqueado
91	DDD Inválido
102	Tempo Esgotado

127	Falha de Interconexão

Tabela 3 - Estados dos Agentes

Estado	Descrição
ONLINE	Agente logado e disponível
OFFLINE	Agente deslogado
PAUSED	Agente em pausa
UNAVAILABLE	Agente indisponível (problema no ramal)
RINGING	Agente em oferecimento de chamada
ONTHEPHONE	Agente ao telefone