

# Ferramenta para desenvolvimento de Planos de Discagem no Asterisk

*Universidade Federal de Santa Catarina*

*Ciências da Computação*

Richard Hobold da Rosa

[hobold@inf.ufsc.br](mailto:hobold@inf.ufsc.br)

## **Abstract**

With the emerging of VoIP technology many solutions appears to implant the voice system, many of these being open solutions. The Asterisk appears in this process, as a PABX open source, replacing the proprietary solutions with a reduced cost. Along with the Asterisk, some tools are created to facilitate the configuration and dial plan creation process. However the majority of these tools doesn't presents a graphic interface, or are paid solutions.

In this work we present a graphic tool that can generate dial plans for the Asterisk system. Using a programming language with portability the user will be able to build dial plans in any operation system without the need to know the Asterisk system.

Neste trabalho apresentamos uma ferramenta gráfica que possa gerar planos de discagem para o sistema Asterisk. Utilizando uma linguagem de programação que permita portabilidade o usuário poderá criar planos de discagem em qualquer sistema operacional sem precisar conhecer o funcionamento do sistema Asterisk.

## **Resumo**

Com o surgimento da tecnologia VoIP surgem várias soluções para implantação do sistema de voz, sendo muitas destas soluções abertas. O Asterisk aparece neste processo, como uma PABX *open source*, substituindo as soluções proprietárias com um custo reduzido. E com o Asterisk surgem algumas ferramentas para facilitar o processo

de configuração do sistema e para criação de planos de discagem. No entanto a maioria destas ferramentas não apresenta uma interface gráfica que facilite o uso do Asterisk, ou são soluções pagas.

Neste trabalho apresentamos a criação de uma ferramenta gráfica que possa gerar planos de discagem para o sistema Asterisk. Utilizando uma linguagem de programação que permita portabilidade o usuário poderá criar planos de discagem em qualquer sistema operacional independente do conhecimento sobre o funcionamento do sistema Asterisk.

## **1. Introdução**

Com o crescimento das redes baseadas em IP (Internet Protocol) de comutação de pacotes, verificou-se a possibilidade de usar a infra-estrutura dessas redes para transmissão de voz. Assim surge o VoIP (Voice over Internet Protocol), uma tecnologia que torna capaz a transmissão de voz sobre uma rede IP com um padrão aceitável de qualidade de serviço (QoS) com um custo muito menor. Com o crescente aumento do uso do VoIP novas soluções surgiram para todos os tipos de usuários, algumas ainda com altos custos de implantação, outras com custos reduzidos como o Asterisk (Asterisk, 2007).

O Asterisk é um software livre que implementa as funcionalidades de uma PABX (Private Automatic Branch Exchange), podendo servir apenas como PABX IP ou mesmo de forma híbrida integrando VoIP e a rede telefônica convencional. Apesar de diversas comunidades e sites dedicados ao Asterisk com muitas informações e tutoriais, sua instalação e principalmente configuração não é sempre simples. O núcleo de um sistema Asterisk está na configuração de seu plano de discagem (Dialplan), por que define como o Asterisk irá gerenciar todas as ligações. A configuração de um plano de discagem pode ser muitas vezes pouco intuitiva, e quando muito extensa pode se tornar bastante complexa e confusa ao trabalhar com arquivo em modo texto. Existem algumas ferramentas gráficas que podem auxiliar a utilização do Asterisk neste processo, porem são ferramentas comerciais. Outras soluções gratuitas ou software livre ajudam na configuração do Asterisk, mas não facilitam em muito a criação de um plano de discagem.

Baseado nesta dificuldade de configuração, e inexistência de uma solução livre,

este trabalho propõe uma ferramenta gráfica para criação de planos de discagem do Asterisk.

## 2. Asterisk

O Asterisk (ASTERISK, 2006) é um *software* de código aberto que implementa uma central telefônica. Ele permite que clientes conectados a ele façam ligações uns para os outros e permite também que estes clientes façam ligações para telefones conectados em outras centrais, como por exemplo, uma central pública de telefonia. Seu nome vem do símbolo, \*, que nos sistemas Unix e DOS representa um “coringa” significando qualquer caractere, qualquer arquivo.

Desenvolvido sob a GPL (*General Public License*), o Asterisk foi criado por Mark Spencer da empresa Digium. Spencer foi o primeiro mantenedor do Asterisk, sendo que dezenas de outros programadores contribuíram para o desenvolvimento de novas funções e características.

Desenvolvido para o sistema operacional Linux, atualmente o Asterisk pode ser executado também no FreeBSD, OpenBSD, Mac OS X, Sun Solaris e Microsoft Windows. Está atualmente na versão 1.4.

O *software* básico do Asterisk inclui muitas características encontradas até então somente em sistemas telefônicos caros e proprietários. Características como correio de voz, respostas interativas, distribuição automática de chamadas e conferência em chamadas são encontradas no Asterisk. É possível ainda criar novas funcionalidades escrevendo *scripts* na linguagem do Asterisk, adicionando módulos escritos em linguagem C, além de outras formas de customização.

O Asterisk suporta muitos protocolos de voz sobre *IP* (*VoIP*), incluindo o SIP e o H.323. Ele pode trabalhar com a maioria dos telefones SIP, podendo atuar como um registrador ou um *gateway* entre o universo *IP* e o universo de telefonia convencional. Desenvolvedores do Asterisk criaram um novo protocolo, o IAX, para melhorar o entroncamento entre servidores Asterisk.

### 2.1. Plano de Discagem

Todo o sistema Asterisk é controlado pelo plano de discagem, através do arquivo

extension.conf usando a sintaxe já mencionada. Neste arquivo o administrador do sistema especifica que ações serão tomadas quando uma ligação for recebida. O plano de discagem é dividido em quatro partes: contextos, extensões, prioridades e aplicações.

## Contextos

O plano de discagem é dividido em seções chamadas de contextos. Contextos são nada mais que grupos de extensões. Extensões definidas em contextos diferentes não interagem, a não ser que especificado do contrário. Eles são identificados colocando o nome do contexto entre colchetes ([]), assim um contexto nomeado chamadas seria: **[chamadas]**. Todas as instruções colocadas depois da definição do contexto farão parte deste contexto até o próximo contexto ser declarado.

## Extensões

Dentro de cada contexto estão definidas uma ou mais extensões. Uma extensão é uma instrução executada pelo Asterisk, disparada por alguma ligação recebida ou número sendo discado no canal. As extensões especificam o que acontece com cada ligação à medida que elas avançam pelo plano de discagem.

```
exten => 100,1,Answer( )
```

O nome da extensão é 100, com prioridade 1 e a aplicação é Answer().

## Prioridade

Cada extensão pode ter vários passos, que chamamos de prioridades. Cada prioridade é numerada sequencialmente, começando em 1. Cada prioridade especifica uma aplicação, como no exemplo a seguir:

```
exten => 100,1,Answer( )
```

```
exten => 100,2,Hangup( )
```

Neste exemplo quando o plano atinge a extensão 100, a prioridade 1 atenderia a chamada e depois a prioridade 2 desligaria a chamada. As prioridades precisam ser numeradas corretamente, senão não funcionarão corretamente.

## Aplicações

Cada aplicação executa uma determinada ação sobre o canal atual, como tocar um som, receber tom de discagem, ou desligar a chamada. Nos exemplos de extensões e prioridades foram apresentadas duas aplicações: Answer() e Hangup(). Algumas outras aplicações requerem o uso de argumentos, como a aplicação Dial()

**exten => 100,1,Dial(SIP/Rose,20)**

A aplicação é utilizada para ligar para o cliente Rose usando um canal SIP, por no máximo 20 segundos.

## 3. Ferramentas Existentes

Já existem algumas ferramentas disponíveis para criação de planos de discagem e configuração do Asterisk. No entanto uma única ferramenta utiliza algum modelo gráfico para construção do plano de discagem, e esta não é um software livre. Existem outras gratuitas ou livres, porem não facilita muito o trabalho dos usuários.

O Visual Dialplan(VDIALPAN) é um completo editor gráfico de planos de discagem feita pela Apstel (Visual Dialplan, 2007). Ela é compatível com o Asterisk 1.4 e permite configurar URAs, gerir chamadas que entram e saem facilmente em um tempo mínimo usando diagramas para definir o fluxo de discagem. No entanto, é uma ferramenta paga que vai contra a utilização do sistema Asterisk que visa reduzir custos.

O Dialplanner(LANVIK, 2005) é uma ferramenta web (Dialplanner, 2005) que fornece facilidades para criação de planos de discagem. Você pode criar contextos e extensões, e selecionar os comandos apropriados a serem adicionados a partir de uma lista. Cada comando pode conter argumentos que devem ser preenchidos pelo usuário, da mesma forma que faria com o arquivo em texto. O plano de discagem vai sendo mostrado em forma de arvore na janela ao lado direito do programa, e necessita que o usuário tenha conhecimento da sintaxe do plano de discagem.

O ACTOS(ACTOS) é utilizado para gerenciar vários parâmetros de configuração de um sistema Asterisk. Os arquivos podem ser carregados e modificados

diretamente do servidor onde o Asterisk está instalado. Existem varias opções no ACTOS, para visualização de conexões de clientes, registro de clientes, parâmetros de configuração, edição de variáveis globais alem de gerenciamento de salas de conferencia. Também possui opções para criação de planos de discagem, porem assim como o Dialplanner não apresenta uma interface gráfica para criar os planos e ainda exige grande conhecimento do plano de discagem.

## **4. Ferramentas de Apoio**

Apesar de também existirem muitos tutoriais e informação sobre o assunto, os planos de discagem exigem um conhecimento das aplicações e sintaxe usadas no Asterisk. Já que dificilmente encontraremos um plano igual, ou mesmo parecido ao que pretendemos implantar nos veremos obrigados a estudar os vários comandos existentes. Uma ferramenta que torne transparente todos os aspectos de conhecimento de sintaxe, comandos e até lógica do plano de discagem, facilitaria este aspecto de maior complexidade do Asterisk.

A ferramenta tem como princípio ser desenvolvida como alternativa às ferramentas proprietárias existentes, portanto deve ser código livre. Deve ter características de usabilidade, como facilidade de aprendizado, facilidade de memorização e baixa taxa de erros, permitindo assim que o usuário não conheça plenamente o funcionamento do plano de discagem do Asterisk.

Outro aspecto importante é a portabilidade: a ferramenta deve ser independente de arquitetura. Apesar da maioria dos servidores Asterisk serem implementados em sistemas Unix, o usuário deve poder criar seu plano de discagem sem necessidade de conhecimentos específicos de determinada arquitetura.

A ferramenta deve permitir a criação dinâmica de um plano de discagem através de uma interface gráfica. O plano de discagem deverá poder ser salvo em um formato padrão para poder ser aberto mais tarde, ou usado de outra forma. O programa deve permitir que o plano de discagem criado na interface gráfica possa ser exportado para uma configuração válida para ser usado em um Sistema Asterisk.

### **4.1. Java/Swing**

A linguagem de programação utilizada neste projeto foi o Java(JAVA, 2007). Esta linguagem foi desenvolvida pela *Sun Microsystems* (JAVA, 2007), com o objetivo principal de ser independente da plataforma, isto é, que um programa feito em Java funcione sem necessidade de alterações, incluindo recompilação, em qualquer tipo de computador ou sistema. Isto garantirá a portabilidade da ferramenta criada.

Apresenta uma biblioteca de classes extensa, que auxiliam o programador nas diversas tarefas de implementação de programas, sendo uma destas classes o Swing. O Swing faz parte integrante da linguagem de programação Java, do que é conhecido por *Java Foundation Classes* (JFC). JFC é um conjunto de classes, disponíveis para os programadores Java, que servem para facilitar o processo de criação de aplicações com interfaces gráficas.

Existem vários aplicativos que usam Java/Swing, utilizando fluxogramas ou outros modelos de representação gráfica para diferentes problemas. Algumas destes aplicativos são livres e podem servir de base para o desenvolvimento da ferramenta proposta.

## **4.2. Sorria!**

O Sorria! (LEITE, 2000) seria inicialmente como base para todo editor. Todo seu sistema de criação de apresentações SMIL é muito parecido com os requisitos do editor de plano de discagem. Já possuí um sistema para salvar as apresentações em formato XML que pode ser modificado para se adequar ao funcionamento do editor.

Contudo, o Sorria! não poderia atender todas as necessidades do editor. No Sorria! os caminhos (arestas) não podem ser manipuladas diretamente, o que dificultaria a criação do editor. Acabou-se por usar a estrutura gráfica do Sorria! e usar outra ferramenta de apoio para manipular a manipulação do plano de discagem.

## **4.3. JGraph**

O JGraph(JGRAPH, 2008) acabou sendo escolhido por ser uma das opções mais completas. Com ele você pode criar qualquer tipo de grafo, podendo ser usado para descrever diagramas de processos, fluxo de trabalhos, tráfego, redes de computadores, etc.

No editor, o JGraph foi utilizado junto com o Sorria! para fornecer toda interface gráfica do projeto. Com o JGraph passamos a ter tanto caminhos (arestas), quanto símbolos (vértices) editáveis.

#### **4.4. Ferramenta Gráfica de Criação de URAs para Asterisk**

Esta ferramenta, desenvolvida por Guilherme Vieira(VIEIRA, 2007), tinha como objetivo poder criar URAs através de uma interface gráfica. O trabalho acabou servindo como base para geração dos arquivos gerados pela ferramenta gráfica. Neste trabalho a ferramenta armazena as URAs criadas em formato XML, processo semelhante usado neste trabalho.

### **5. Ferramenta Desenvolvida**

A interface do usuário foi constituída com o uso do Sorria! junto com o JGraph. Assim foi possível criar de forma rápida um editor eficiente para construção do plano de discagem. O uso da ferramenta é bem simples e direto, onde o usuário só precisa escolher a sequência que seu plano siga.

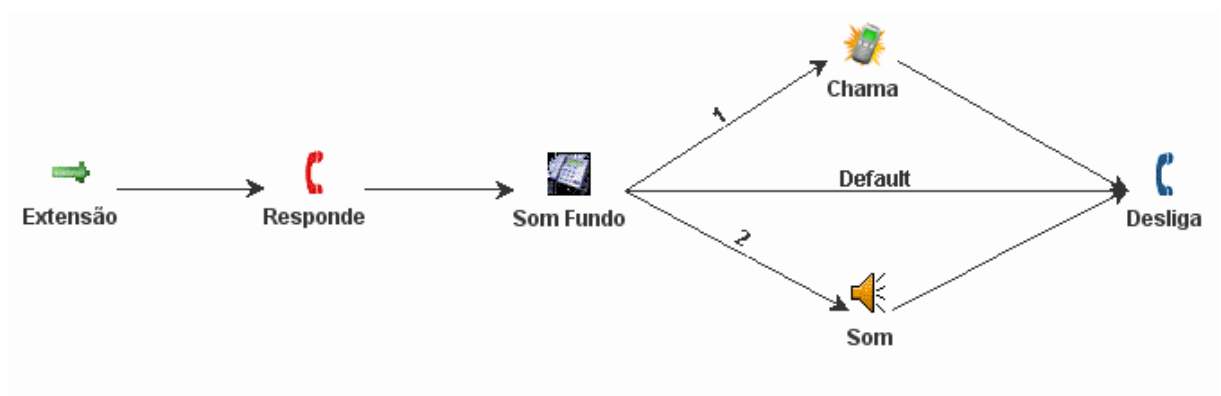
O usuário pode escolher os tipos de ações que serão executadas nas abas de edição, que apresentam um grupo de ações, mantendo ações correlatas na mesma aba. Assim a aba Chamada guarda todas as ações referentes ao gerenciamento de chamadas: responder, desligar, chamar um canal, tempo de espera, etc.

Depois de escolhido a ação o usuário pode clicar na área de criação onde pode editar cada ação separadamente. Pode adicionar os caminhos (arestas) que as ações tomarão gerando uma sequência de discagem. Na figura 15 podemos ver o aspecto geral do editor.

As ações representam os comandos que são usados no plano de discagem, cada uma possuindo características próprias, devendo ser editadas unicamente. Cada sequência deve começar por um elemento inicial, representado pela ação Extensão, e terminar por uma ação final que termine a chamada, representado pela ação Desliga. Entre estas ações existem um grande número de possibilidades que estão a escolha do usuário.



Abaixo, um exemplo simples de uma sequência válida em um plano de discagem que representa uma URA (Unidade de Resposta Audível). Ele começa com a ação e Extensão, a chamada é então atendida pela ação Responde e logo depois a ação Som Fundo é executada. A ação Som Fundo toca um som enquanto espera por uma entrada de dígitos do usuário na linha. Digitando “1” a ação Chama liga para um determinado canal, e depois de finalizado termina a chamada. Digitando “2” executa a ação Som que toca um arquivo de som para o usuário na linha e logo depois termina a chamada.



Exemplo de plano de discagem

## 5.1. Persistência em XML

Começamos com dados específicos pelo cabeçalho que guarda o nome do arquivo:

```

<asterisk>
<head>
  <metadata name="title" content="Simples URA" />
</head>

```

No campo do documento XML descrevemos a estrutura do plano de discagem. O primeiro elemento a ser convertida, a ação Extensão, é de fácil conversão, já que assim como no modelo gráfico servirá como elemento inicial. A ação extensão possui um atributo de identificação que corresponde ao atributo “nome” no arquivo XML:

```

<body>
  <ext nome="s" left="68" top="26" id="0" children="1">
  </ext>

```

Os atributos “left” e “top” guardam a posição do elemento no editor gráfico, enquanto o atributo “children” guarda o “id” das ações que podem ser atingidas a partir da ação atual. A seguir vem à ação Responde que atende a linha que só precisa do

atributo “children” para apontar quais seus sucessores:

```
<body>
  <ext nome="s" left="68" top="26" id="0" children="1">
    <resp left="88" top="26" id="1" children="2">
  </ext>
```

Na sequencia a ação Som Fundo que além dos atributos “top”, “left”, “children” possui o atributo “arquivo” que contem o nome do arquivo a ser tocado. No atributo “cvalues” armazena o valor de cada aresta que representa o número que será digitado:

```
<body>
  <ext nome="s" left="68" top="26" id="0" children="1">
    <resp left="88" top="26" id="1" children="2">
      <fundo left="108" top="26" id="2" children="3 4 5"
arquivo="arquivo som" cvalues="1 2 Default">
    </ext>
```

Depois a ação Chama com o atributo “valor” que guarda o recurso a ser chamado. Finalmente a ação Som com o atributo “arquivo” que informa o arquivo a ser tocado e a ação Desliga que não possui nenhum atributo diferenciado:

```
<body>
  <ext nome="s" left="68" top="26" id="0" children="1">
    <resp left="88" top="26" id="1" children="2">
      <fundo left="108" top="26" id="2" children="3 4 5"
arquivo="arquivosom" cvalues="1 2 Default">
      <chama left="108" top="26" id="3" children="5"
valor="SIP/Teste" >
      <som left="108" top="26" id="4" children="5"
arquivo="arquivosom2">
      <desliga left="108" top="26" id="5">
    </ext>
  </body>
```

De modo semelhante, outras ações são convertidas em XML. Algumas ações possuem parâmetros extras que serão convertidos em atributos no documento convertido.

## 5.2. Exportação do Plano de Discagem

O editor converte o plano criado em um arquivo de texto que pode ser anexado ao plano de discagem original. Do mesmo modo como é convertida em XML, a plano cada ação é convertida em uma configuração válida para o Asterisk. A conversão começa usando as informações do cabeçalho para gerar o nome do contexto.

```

<asterisk>
<head>
  <metadata name="title" content="SimplesURA" />
</head>

```

Obtemos o título do plano feito, que no caso é “SimplesURA”. Convertendo para formato do Asterisk:

```
[SimplesURA]
```

O nome do contexto é importante caso outros planos sejam criados e possam referenciar este contexto. A seguir a ação Extensão:

```

<body>
  <ext nome="s" left="68" top="26" id="0" children="1">

```

No format Asterisk a ação Extensão apenas indica como será a sequência de ações a partir dela. No caso ela “reversa” a extensão s para esta sequência criada pelo usuário:

```

[SimplesURA]
exten => s, 1,

```

O terceiro elemento da tripla é o comando propriamente dito. Como a primeira ação é a Responde, o terceiro elemento da primeira linha é comando corresponde no Asterisk:

```

[SimplesURA]
exten => s, 1, Answer()

```

Agora a segunda ação, Som Fundo, que tem uma conversão um pouco mais complexa. Cada aresta que sai da ação Som Fundo representa uma extensão que deverá ser criada. O número que o criador do plano definiu nas arestas será o identificador da extensão.

```

[SimplesURA]
exten => s, 1, Answer()
exten => s, 2, Background(arquivosom)

```

Depois de adicionado o comando relativo ao Som Fundo, cada aresta recebe sua extensão correspondente. Para a aresta 1 que possui a ação Chama uma extensão com o identificador 1 é adicionado seguido pela ação Desliga:

```

[SimplesURA]
exten => s, 1, Answer()

```

```
exten => s, 2, Background(arquivosom)
exten => 1, 1, Dial(SIP/Teste)
exten => 1, 2, Hangup()
```

O mesmo é feito para a aresta com o número 2 :

```
[SimlesURA]
exten => s, 1, Answer()
exten => s, 2, Background(arquivosom)
exten => 1, 1, Dial(SIP/Teste)
exten => 1, 2, Hangup()
exten => 2, 1, Playback(arquivosom2)
exten => 2, 2, Hangup()
```

A aresta nomeada “Default” é um caso especial que determina para onde segue a sequência em caso de erro na ação Som Fundo. Caso ela seja definida então a extensão *i* (*invalid*) e extensão *t* (timeout) são construídas usando a sequência a partir da aresta default, que no caso vai para uma ação Desliga. A configuração resultante:

```
[SimlesURA]
exten => s, 1, Answer()
exten => s, 2, Background(arquivosom)
exten => 1, 1, Dial(SIP/Teste)
exten => 1, 2, Hangup()
exten => 2, 1, Playback(arquivosom2)
exten => 2, 2, Hangup()
exten => i, 1, Hangup()
exten => t, 1, Hangup()
```

## 6. Conclusão e Trabalhos Futuros

A ferramenta desenvolvida apresenta um estágio muito inicial de desenvolvimento. Optou-se por reutilizar código de outra ferramenta gráfica, o Sorria!, para facilitar a criação do código do programa. No entanto isto tornou a compreensão do código um tanto complexa, o que dificultaria a extensão do programa em trabalhos futuros.

Ela também abrange uma gama muito pequena de comandos suportados pelo plano de discagem Asterisk, mas cumpre o seu papel ao demonstrar a viabilidade de produzir uma ferramenta open source para criar planos de discagem.

Para tornar a ferramenta mais robusta e facilitar a extensão seria razoável recriar a interface básica para que não dependa mais do código do Sorria!. Para permitir que a ferramenta possua mais ações para criar planos de discagem mais completos. Acrescentar módulos que permitam a configuração de outros aspectos do sistema

Asterisk. E também fornecer arquivos de ajuda mais detalhados para facilitar ainda mais o uso da ferramenta.

## 7. Referências

[ASTERISK, 2006] **Asterisk**: An Open Source PBX and telephony toolkit. Disponível em: <<http://www.asterisk.org/>>. Acesso em: 16 nov. 2007.

[JAVA, 2007] Sun Microsystem. **Java Technology**. Disponível em: <<http://java.sun.com/>>. Acesso em: 12 de nov. de 2007.

[JGRAPH, 2008] JGraph. **JGraph and JGraph Layout Pro User Manual**. Disponível em: <http://www.jgraph.com/pub/jgraphmanual.pdf>. Acesso em: 15 abr. de 2008.

[LANVIK, 2005] Lanvik, I. S. B. **Dialplanner**. <<http://www.lanvik-icu.com/asterisk/dialplanner/index.php>>. Acesso em: 15 nov. 2007.

[LEITE, 2000] Leite, R. **Estudo e Desenvolvimento de um Sistema de Autoria para SMIL**. Monografia (Graduação em Ciências da Computação) – Universidade Federal de Santa Catarina, 2000.

[VIEIRA, 2007] Vieira, G. **Ferramenta para desenvolvimento de URA no Asterisk**. Monografia (Graduação em Ciências da Computação) – Universidade Federal de Santa Catarina, 2007. Disponível em: <http://projetos.inf.ufsc.br/>. Acesso em 16 de nov. 2007.

[VDIALPAN] Apstel. **Visual Dialplan**. Disponível em: <<http://www.apstel.com/>>. Acesso em: 15 nov. 2007.

[ACTOS] ACTOS. Asterisk Configuration Tool Open Source. Disponível em: <<http://www.derrier.com/pierre/code/actos.html>>. Acesso em: 16 nov. 2007.