

Asterisco Paraná - Manual do Expert

Este manual pretende documentar o funcionamento da plataforma de Telefonia IP “asteriscoParaná”, sob a ótica de desenvolvimento. A utilização do sistema por administradores e usuários está descrita nos manuais respectivos.

Tópicos abordados:

- O que vai em banco e o que vai em arquivos de configuração?
- Documentação do mini-framework da interface
- ClasseAsterisk, ClasseParametro , ClasseConexão e demais
- AGIfuncionalidades
- Funcionamento Audioconferencias
- Grupo Chefe Secretária
- Alarme e Rechamada Automática
- Webservice (click2call, auth, ...)
- Empacotamento, Divisão dos arquivos
- Esquema de segurança (ACL dos controles)
- healthCheck.sh
-
- Agentes (CC)
-
- Provisionamento STP
-
- Powertools
-
- ...?

O que vai em banco e o que vai em arquivos de configuração?

As configurações do asterisk feitas pela sistema asteriscoParaná estão divididas em 3 categorias:

- Arquivos de configuração (.conf em /etc/asterisk)
- Banco de dados
- astDB

Arquivos de configuração (.conf em /etc/asterisk)

Para alterações menos frequentes o sistema escreve diretamente nos arquivos de configuração do Asterisk. Dados como configurações de troncos, classes de musica de espera, entre outros. A seguir estão listados os arquivos de configuração do asterisk.

Os arquivo escritos pela interface são customizáveis através de arquivos com a extensão .base (que estão na pasta /etc/asterisk/telip/base). A escrita do arquivo é feita da seguinte forma: O conteúdo do arquivo base correspondente é carregado e utilizado no início da escrita do arquivo, logo após os dados dinâmicos são adicionados. Por exemplo, o arquivo sip.conf: Ele será formando pelo conteúdo do arquivo /etc/asterisk/telip/base/sip.base e logo em seguida serão adicionadas, pela interface, as contas SIP dos troncos configurados.

Todos os arquivo que são gerados pela interface tem o seguinte cabeçalho:

```
; Arquivo gerado em dd/mm/yyyy-hh:mm:ss pelo sistema Asterisco Parana
; ---- Nao edite diretamente ----
;
```

Listagem dos arquivos de configuração do asterisk

- asterisk.conf:

Arquivo padrão - Não editado pelo sistema
Define as pastas do sistema

- cdr_manager.conf:

Arquivo padrão - Não editado pelo sistema
Mantem desligado o envio do CDR pelo AMI

- cdr_pgsql.conf:

Arquivo de configuração do banco Postgres a ser utilizado para armazenar os CDRs
Se o banco estiver em um servidor remoto o admin do sistema deverá inserir manualmente as informações necessárias neste arquivo.

- cdr.conf:

Arquivo padrão - Não editado pelo sistema

- codecs.conf:

Arquivo padrão - Não editado pelo sistema

- extconfig.conf:

Arquivo padrão - Não editado pelo sistema
Neste arquivo estão listadas as tabelas da base de dados Postgres a serem utilizadas

- extensions.conf:

Arquivo Não editado pelo sistema
Plano de discagem principal. Todas as funcionalidades discadas do sistema estão definidas neste arquivo. O extensions.conf não é editado pela interface, porem nele estão diversos 'includes' de outros arquivos, que estão localizados na sub-pasta 'telip', os quais são escritos pela interface para que o plano de discagem reflita as configurações nela efetuadas. Evite fazer qualquer alteração neste arquivo, pois uma alteração mal executada pode quebrar o funcionamento do PABX, além de que estas alterações serão sobrescritas quando o PABX for atualizado.

- features.conf:

Arquivo Não editado pelo sistema
Configura os códigos de transferência, captura a funcionalidades adicionais (conferencia a 3 e gravação sob demanda)

- http.conf:

Arquivo Não editado pelo sistema
Mantem o gerenciamento via web desligado

- iax.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de troncos IAX
BASE: /etc/asterisk/telip/base/iax.base
Neste arquivo estão listados os troncos IAX definidos na interface

- iaxprov.conf:

Arquivo Não editado pelo sistema
Funcionalidade não utilizada. Arquivo presenta para evitar WRNING na console do asterisk

- indications.conf:

Arquivo Não editado pelo sistema
Define os tons de sinalização do padrão brasileiro a serem utilizados pelo sistema

- khomp.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de troncos com placas Khomp

BASE: /etc/asterisk/telip/base/khomp.base

- logger.conf:

Arquivo Não editado pelo sistema

Define os tipos de log. Utilizado para ativar manualmente o log de Debug na console

- manager.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de telefonistas

BASE: /etc/asterisk/telip/base/manager.base

Este arquivo é o ponto principal de configuração dos dados de login AMI a ser utilizado pelo sistema

- meetme.conf:

Arquivo EDITADO pelo sistema. Re-escrito de 15 em 15 minutos para habilitar as salas de conferencia a 3 agendadas

No final do arquivo estão as salas 81 a 88, acessadas pelos códigos *81 a *88. O script de CRON a seguir:

```
0-45/15 * * * * root /usr/lib/telefonaiip/arquivos/scripts/cronAudioConf.php
```

é executado de 15 em 15 minutos e habilita/desabilita as salas de conferencia neste arquivo baseado no agendamentos de audioconferências cadastrados pelo sistema

- modules.conf:

Arquivo Não editado pelo sistema

Na sua configuração inicial certifica que o módulo res_config_pgsql seja carregado logo no inicio da inicialização do asterisk, para que as configurações feitas no Banco de Dados estejam presentes o quanto antes.

- musiconhold.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos grupos de música em espera

BASE: /etc/asterisk/telip/base/musiconhold.base

Todas as musicas de espera carregadas através da interface são salvas na pasta /var/lib/asterisk/moh, que é a pasta da classe 'default' de música de espera. Quando é feito um agrupamento de musica de espera uma sub-pasta com o nome do agrupamento é criada na pasta base, e links simbólicos são feitos apontando para os arquivos selecionados (que estão na pasta base)

- queues.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações em telefonistas e grupo chefe-secretária

BASE: /etc/asterisk/telip/base/queue.base

O arquivo base DEVE terminar com as configurações da fila da telefonista, visto que ao escrever o arquivo a interface irá adicionar logo após o conteúdo base os membros da fila telefonista (caso configurado para isto). Após os dados de telefonistas são criadas as filas de atendimento chefe-secretária da seguinte forma:

Os agrupamentos de secretárias são analisados e uma fila é criada para cada agrupamento distinto, com o nome poolSecretXX (onde o XX é incremental). Desta forma não são criadas filas distintas com as mesmas secretárias

- res_pgsql.conf:

Arquivo de configuração do banco Postgres a ser utilizado para armazenar as configurações. Este arquivo é o ponto principal de configuração do banco de dados a ser utilizado. O sistema asteriscoParaná vem buscar nele as credenciais para conexão com o banco de dados.

Se o banco estiver em um servidor remoto o admin do sistema deverá inserir manualmente as informações necessárias neste arquivo.

- res_snmp.conf:

Arquivo Não editado pelo sistema

- rtp.conf:

Arquivo Não editado pelo sistema

Define o range de portas RTP a serem utilizadas. Configuração importante no caso de configuração avançada de NAT

- say.conf:

Arquivo Não editado pelo sistema

- sip.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de troncos SIP

BASE: /etc/asterisk/telip/base/sip.base

Neste arquivo estão listados os troncos SIP definidos na interface e as PAs de atendimento (do módulo de CallCenter)

- voicemail.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de usuários

BASE: /etc/asterisk/telip/base/voicemail.base

Uma entrada de correio de voz é criada para cada usuário configurado

- zapata.conf:

Arquivo EDITADO pelo sistema. Re-escrito quando houverem alterações nos dados de troncos com placas Digium/similares

BASE: /etc/asterisk/telip/base/zapata.base

Configurações no banco de dados

Alterações frequentes no sistema estão configuradas no banco de dados (e o acesso a este configurado no arquivo /etc/asterisk/res_config_pgsql.conf). As tabelas criadas no banco de dados representam os arquivos de configuração. Por exemplo a linha a seguir, do sip.conf:

allow=ulaw

Estaria representada na tabela "sip", na coluna "allow" com o valor "ulaw".

Configurações no astDB

O astDB é um mini-banco-de-dados interno do asterisk, de acesso muito rápido (cacheado em memória), que pode ser lido através no plano de discagem, pela CLI, pelo AMI e por AGI. É um banco de estrutura muito simples, na forma de chave-valor. Para efeito de organização as chaves podem ser nomeadas para criarem famílias. Por exemplo temos a família "plano" onde estão as variáveis de controle do plano de discagem.

Ele serve para dar dinamismo ao plano de discagem para que não seja preciso rescreve-lo a cada alteração na interface. Por exemplo o ramal e o status do serviço noturno, quando alterados

pelo interface, são atualizados no astDB, para que a mudança reflita no plano de discagem. A seguir estão listadas todas as entradas utilizadas no astDB:

plano/svcnot_ativo - Serviço noturno ativo - 0 ou 1

plano/svcnot_ramal - Ramal do serviço noturno. Ex.: SIP/1201

ramaisAtivos/\${RAMAL} - ramais que estão cadastrados - 1 = ativo

naoperturbe/\${RAMAL} - não perturbe ativado para o ramal - 0 ou 1

mobilidade/\${RAMAL} - se o ramal esta em mobilidade - 0 ou 1

ramalNA/\${RAMAL_X} - Ramal para redir a ligação de não atender o ramal X

ramalOcupado/\${RAMAL_X} - Ramal para redir a ligação de o o ramal X

semcorreiodevoz/\${RAMAL} - Ramal NAO tem correio de voz - 0 ou 1

rediscagem/\${RAMAL} - variável interna do DP - para rechamada automática

conf3/sala - Sala atual para conferencia a 3

Documentação do mini-framework da interface

A interface web do sistema asteriscoParaná é desenvolvida no modelo MVC, com classes Visao* para controle de apresentação, classes Controle* para as regras de negócio de classes bdFacil para persistência. O Fluxo de navegação no sistema esta representado nos 2 diagramas a seguir:

Diagrama de sequência da Interface (montagem da tela):

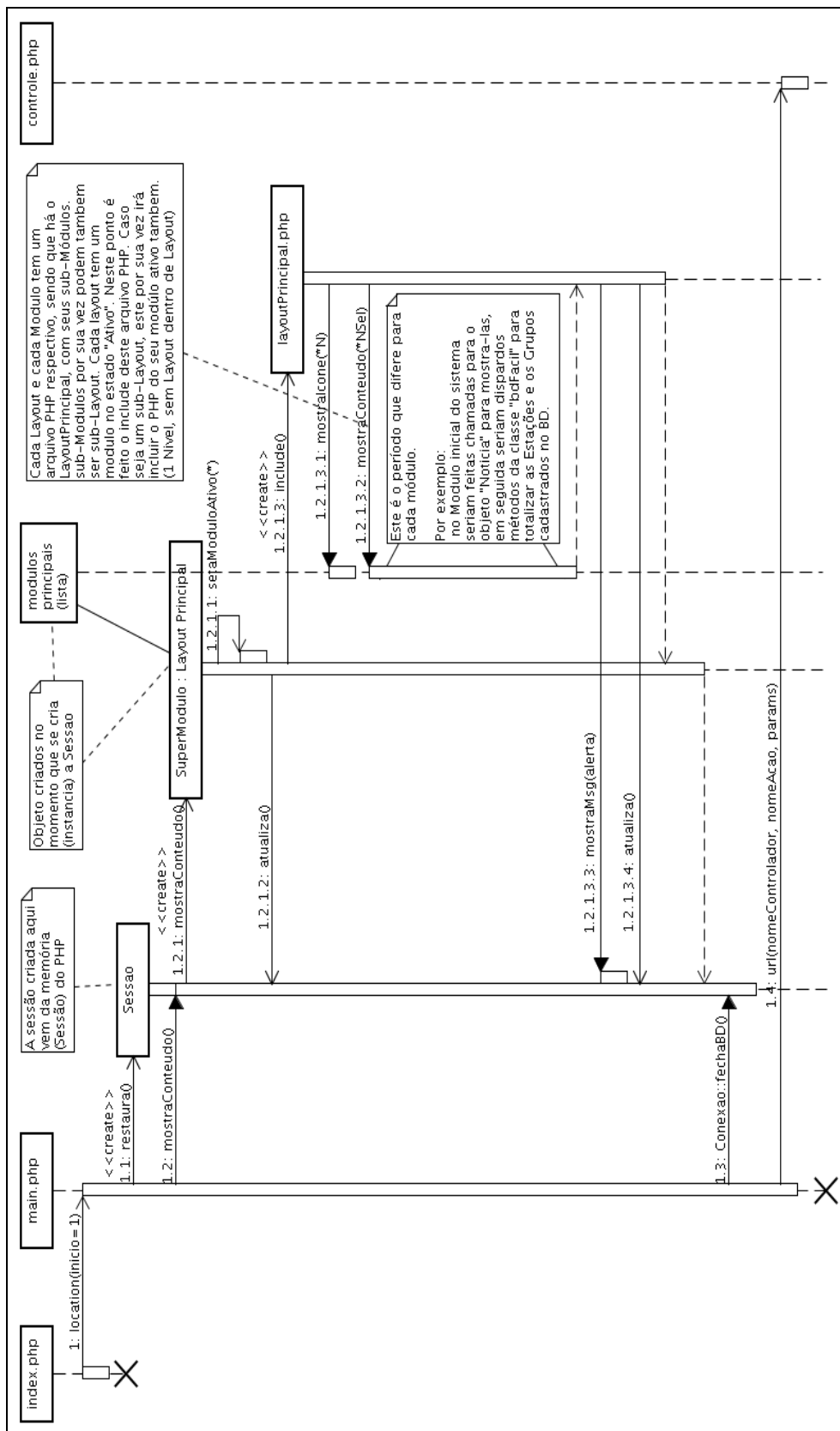
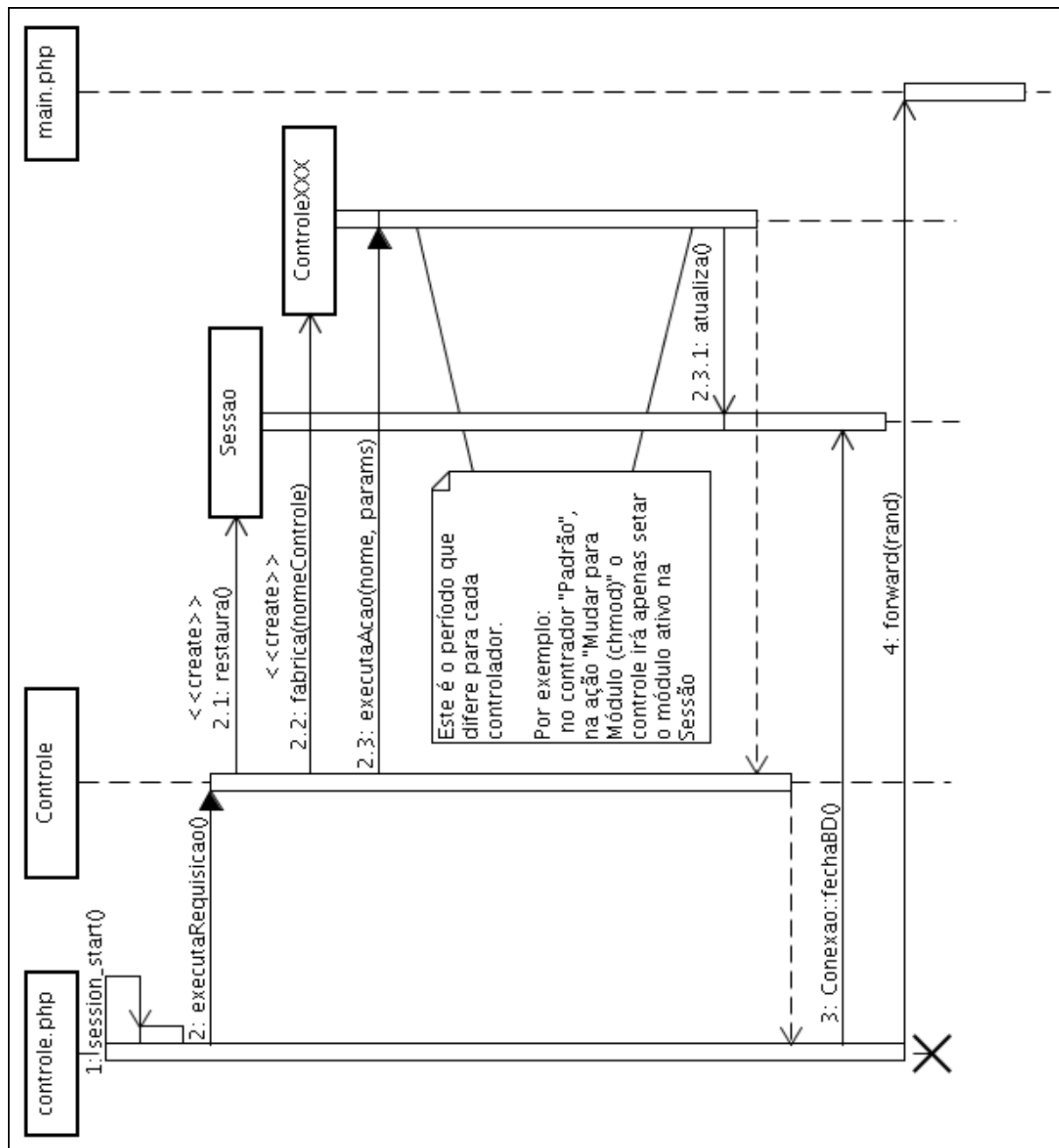


Diagrama de sequência do Controle (execução de requisições/comandos):



ClasseAsterisk, ClasseParametro , ClasseConexão+bdFacil e demais

A **classeAsterisk** centraliza todas as funções genéricas (estáticas) do sistema para interação com o asterisk. É nela que estão as funções que escrevem os arquivos de configuração do asterisk (métodos `escreve*`). Estes métodos tem a função de coletar dados de diversas fontes (primariamente do banco de dados) e escrever os arquivos de acordo com estes dados.

Outro método muito utilizado desta classe é o `Asterisk::executaComando`, que recebe um comando CLI do asterisk (ex.: “`dialplan reload`”) e o executa através do AMI (para performance é muito superior ao método “`asterisk -rx 'comando'`”), opcionalmente devolvendo o retorno do asterisk.

A **classeParametro** representa a tabela parâmetro do banco de dados e é utilizado por todo o sistema. Os seus métodos “`get`” e “`set`” leem e atualizam a respectiva tabela. A grande maioria destes parâmetros está listado na tela “Sistema->Configuração” para edição.

As **classeConexão** e **classebdFacil** são a camada de persistência do sistema. A **classeConexao** trata das funções de baixo nível para acesso ao banco (conexão e execução de queries, basicamente). Já a **classebdFacil** faz o “meio-de-campo” mais alto nível, funcionando da seguinte maneira:

Todos os objetos persistentes do sistema tem uma respectiva classe que estende a **bdFacil**. Este objeto é nomeado segundo o nome da sua respectiva tabela, e tem definidos atributos correspondentes as colunas da mesma. Desta forma é possível utilizar os métodos “`salvar`”, “`atualizar`”, “`remover`”, e o próprio construtor da classe, quando informado um id numérico na sua execução, já se encarrega de trazer os dados do objeto do banco de dados. Vejamos um exemplo:

```
Estrutura da tabela custo, que representa os centros de custos do sistema:
CREATE TABLE custo (
  id          serial          NOT NULL,
  nome       character varying(128) NOT NULL,
  sigla      character varying(16)  NOT NULL,
  pai        character varying(16)  NOT NULL,
  codigo     integer,
  idchefe    integer
)

Respectiva classeCusto.php:
<?php
/**
 * Representa os Centros de Custos da organização
 */
class Custo extends bdFacil {
    var $nome, $sigla, $codigo,$classificacao, $pai, $idchefe;
}
?>
```

Perceba que a coluna **id** não está definida na **classeCusto.php**. Isto por que **\$id** já está definido na **classebdFacil**, o que nos indica que esta é uma coluna obrigatória para qualquer classe que se deseje utilizar com este esquema.

Desta forma já é possível utilizar os seguintes métodos definidos na **bdFacil**:

`begin()`, `commit()`, `rollback()`, `carregar()`, `salvar()`, `atualizar()`, `remover()`,
Entre outros...

AGIfuncionalidades

AGIs são formas de estender a “inteligência” do plano de discagem. São scripts desenvolvidos em qualquer linguagem que interagem com o asterisk através da entrada e da saída padrão. Estes scripts devem estar, preferencialmente, na pasta `/var/lib/asterisk/agi-bin/`, sendo chamados no `extensions.conf` da seguinte forma (por exemplo):

```
exten => 1234,1,AGI(funcionalidades, p1, p2, p3)
```

A linha acima irá invocar o script `/var/lib/asterisk/agi-bin/funcionalidades` (que deve ter permissão de execução), passando os parâmetros `p1`, `p2` e `p3`. Feito isso o controle do plano de discagem é passado para o respectivo script.

Na plataforma asteriscoParaná todas as funcionalidades que necessariamente são implementadas via AGI estão em um único script, o `AGIfuncionalidade.php`. Este script fica na pasta do projeto (`/usr/lib/telefonaiip/arquivos/scripts/`), sendo que um link simbólico chamado apenas “funcionalidades” é criado na pasta `agi-bin` apontando para ele.

Este script recebe em **p1** a ação que se deseja que ele execute, sendo os demais parâmetros dependentes da ação. A seguir uma lista das ações atualmente encontradas no script:

- `checaCadeado` - verifica o cadeado eletrônico do ramal
- `movilidade` - configura a mobilidade através do ***60**
- `cadeado` - configura o cadeado eletrônico do ramal através do ***62**
- `do-not-disturb` - configura o não-perturbe do ramal através do ***64**
- `chefe-secretaria` - ativa/desativa o grupo chefe-secretária do ramal através do ***65**
- `prePago` - verifica saldo e calcula tempo máximo para chamadas de usuário pré-pago

Funcionamento Audioconferencias

O sistema asteriscoParaná conta com a funcionalidade de agendamento de audioconferências (AC). Para funcionar existe um “cron job” (script agendado para rodar periodicamente), que roda de 15 em 15 minutos. Este script verifica se existem audioconferencias prestes a iniciar, prestes a terminar, ou a 15 minutos do seu fim.

Ao agendar uma AC o sistema verifica a disponibilidade de salas (8 ao todo) e faz a reserva de uma delas, e em seguida são disparados emails para todos os participantes selecionados, informado a data/hora, o número da sala e a senha para entrada. Quando esta AC deve ser iniciada a respectiva sala é adicionada a configuração do asterisk pelo script cron, e são disparadas chamadas para todos os envolvidos, encaminhando-os para a sala de conferência.

15 minutos antes do término do período agendado o sistema irá disparar um anúncio informando o término da AC em 15 minutos. Passado este tempo o script irá desativar a sala da conferência, derrubando os eventuais canais que ainda estejam em pé.

Grupo Chefe Secretária

O grupo chefe secretaria funciona interceptando chamadas para usuário marcados como “chefe” e as encaminhando para uma fila de atendimento composta por uma ou mais secretárias.

A variável “*filasecretaria*”, armazenada no *astDB*, indica se o usuário chamado é chefe, informando qual o nome da fila chefe-secretária a ser utilizada. Se esta variável for vazia ou nula (não existir para o ramal em questão) significa que este usuário não tem esta funcionalidade ativa.

Quando um grupo chefe-secretária é criado as respectivas filas são re-criadas, levando em consideração todos os grupos definidos no sistema. Desta forma se houverem grupos onde as

secretárias são as mesmas para chefes distintos só será criada uma fila, que será utilizada para ambos os usuários chefes.

A nomenclatura das filas segue o padrão: *poolSecretX*, onde X é um número incremental.

Re-chamada Automática

A re-chamada automática é implementada no plano de discagem e utiliza *callfiles* para originar chamadas. Quando uma ligação interna (ramal a ramal) dá ocupado ou não atende é apresentada ao usuário uma mini URA que possibilita ativar a re-chamada (“Tecle 1 para ativar a re-chamada automática, ou aguarde..”).

Ao ativar a funcionalidade uma entrada no *astDB* é salva para indicar que o respectivo ramal tem uma re-chamada pendente. Assim que este ramal efetuar um desligamento no PABX esta entrada no *astDB* é checada, e se presente um arquivo “*callfile*” é escrito e acionado (movido para /var/spool/asterisk/outgoing), disparando uma chamada para o usuário que solicitou a re-chamada, e ao atender esta ligação o sistema efetua a chamada para o alvo.

Webservice (click2call, auth, ...)

Alguns serviços da plataforma estão disponíveis através de um *webservice*, são eles: listagem, configuração de parâmetros e autenticação de usuário, disparo de chamadas (click2call) e o agendamento e disparo de audioconferencias.

O acesso a este *webservice* é controlado por uma lista de IPs permitidos, e é feito através do acionamento de um URL para um script PHP. Através dos parâmetros passados o *script* a ação desejada é executada. Para verificar a lista de ações e os respectivos parâmetros consulte o próprio *script* em *telefoniaip/www/servicos/voip.php*).

Esquema de segurança (ACL dos controles)

Todas as telas do sistema estão “amarradas” aos respectivos controles que elas executam. Seguem um exemplo do arquivo que configura um módulo do sistema (**usuario.desc**):

```
ordem: 100
nome: Usuários
func01:listaUsuarios|Listar Usuários|usuarioLista|Usuario@delete-Usuario@tocar-
Usuario@pesquisaUsuario
func02:addUsuario|Adicionar Usuário|usuarioAdd|Usuario@novo
func03:editUsuario|Editar Usuário|usuarioAdd|Usuario@edit-Usuario@expirarSenha-Usuario@senhaPadrao
func04:grupos|Grupos de Captura|usuarioGrupo|Usuario@delGrupo-Usuario@addGrupoCaptura
func05:desvios|Desvios de Ramais|usuarioDesvios|Usuario@addDesvio-Usuario@delDesvio
```

Neste arquivo estão listadas todas as funcionalidades do módulo “Usuários” do sistema. As linhas que nos interessam para este tema são as iniciadas por “funcXX”. Nelas estão definidos vários parâmetros para a função, e observando no final de cada linha existem entradas como “[Usuario@delete](#)”. Desta forma sabemos que a função “**listaUsuarios**” tem acesso as funções “**delete**”, “**tocar**” e “**pesquisaUsuario**” do controle **Usuario**, e **nada mais**.

Desta forma se o usuário logado esta na tela de listagem de usuários e tenta acessar uma função de outra tela do sistema (maliciosamente) o controle irá negar acesso a outros controles.

Sempre que se deseja inserir alguma funcionalidade nova no sistema os arquivos de descrição dos módulos afetados devem ser atualizados para permitirem o acesso a nova função.

Alarme

Empacotamento, Divisão dos arquivos

Para facilitar a atualização do sistema através de pacotes Debian este foi quebrado em diversos pacotes:

- asterisco-parana**: meta-pacote se arquivos que depende dos pacotes abaixo.
- asterisco-parana-tema**: arquivos específicos do tema asteriscoparana
- pabxip-sistema**: arquivos PHP da interface web
- pabxip-sistema-data**: arquivos PNG da interface web
- pabxip-asterisk-gsd**: binários do asterisk personalizados para o sistema
- pabxip-sistema-confs**: arquivos de configuração do asterisk personalizado
- pabxip-sistema-cc**: módulo de call-center
- pabxip-bd**: dump SQL do banco de dados
- pabxip-sons**: sons base do asterisk
- pabxip-sons-sistema**: sons personalizados para o asteriscoParaná
- pabxip-cacti**: versão personalizada do cacti para monitoramento do PABX
- pabxip-ha**: arquivos para prover alta-disponibilidade para o PABX

Desta forma é possível liberar pacotes menores que atualizam apenas parte do sistema. Um pacote especial aqui é o pabxip-bd, que verifica a existência de uma base de dados (configurada em `/etc/asterisk/res_pgsql.conf`), e caso haja uma ele executa um script que procura e corrige alterações na estrutura da base de dados.

healthCheck.sh

Este script tem a finalidade de assegurar que todos os arquivos e permissões estão corretos para o funcionamento do PABX. É invocado na instalação do pacote e pode ser rodado a qualquer momento.

Agentes CC

O aplicativos dos agentes é totalmente web, desenvolvido em PHP e utilizando a tecnologia AJAX (Comet) para dar dinamismo. Devido a natureza extremamente dinâmica do sistema deve ser utilizado o servidor web 'lighttpd' para servir o sistema aos clientes, pois este foi desenvolvido com leveza e eficiência em mente.

O aplicativo cliente tem seus dados atualizados a cada evento novo que ocorra no PABX. Nesta atualização o servidor envia uma grande string contendo os dados do agente e também de todas as filas que este cliente pertence. Esta string é tratada por javascript no navegador e os dados dela apresentados na interface para o agente.

Quando o agente deseja executar alguma ação este aciona um script PHP por AJAX mandando os dados da ação a ser executados. Este script PHP por sua vez insere ação em uma fila de comandos para que o controleCC a execute.

Provisionamento STP

Powertools