

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Convergência de Mídias em um Ambiente VoIP

JONATAN HARTMANN MATSCHULAT

Florianópolis

2010 / 2

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO**

Jonatan Hartmann Matschulat

CONVERGÊNCIA DE MÍDIAS EM UM AMBIENTE VOIP

Trabalho de conclusão de curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharelado em Sistemas de Informação.

Orientador: Prof. Dr. Carlos Becker Westphall

Banca Examinadora:

Prof. Carlos Becker Westphall, Dr.

Orientador

Universidade Federal de Santa Catarina

Prof^ª. Carla Merkle Westphall, Dra.

Universidade Federal de Santa Catarina

Edison Tadeu Lopes Melo

Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço primeiramente aos amigos e colegas da SeTIC que colaboraram muito no desenvolvimento deste trabalho, provendo a infraestrutura necessária e me guiando com paciência e dedicação.

De forma especial, agradeço também à minha família que está sempre presente, me acompanhando e aconselhando em diversos momentos.

Estendo meus cumprimentos ao orientador Prof. Carlos Westphall e membros da banca, que se propuseram a avaliar e me guiar na realização deste trabalho.

Procure bons conselhos e você terá sucesso; não entre na batalha sem antes fazer planos.

Provérbios 20:18

RESUMO

A telefonia IP está sendo usada cada vez mais no cotidiano das empresas e residências. Um dos principais fatores que contribui para isto é a expansão da rede mundial de computadores, a Internet, que proporciona o canal para transporte dos dados VoIP. De acordo com a *International Communication Union* (União Internacional de Telecomunicações) e a *Internet World Stats*, em 2009, o número de usuários da Internet atingiu a marca de 1.8 bilhões, o que representa aproximadamente 26% da população mundial. Em 2000, este número era cerca de 400% menor.

A telefonia convencional pouco a pouco vem sendo substituída pela baseada no protocolo IP. Isto se deve principalmente ao fato de podermos utilizar uma rede única para transferência de dados, eliminando a necessidade de uso de circuitos dedicados baseados na tecnologia de comutação de pacotes para a transferência de dados de voz, como é o caso da PSTN - *Public Switched Telephone Network* (Rede pública de telefonia comutada). Podemos ver esta mudança também na Universidade Federal de Santa Catarina, que vem implantando desde 2007 uma estrutura de telefonia IP para alunos, professores e servidores.

É interessante que na implantação de um ambiente de telefonia IP, aproveitemos as vantagens que ele tem a oferecer, como por exemplo, a implementação de novas aplicações e serviços baseados nos recursos que a Internet proporciona. Esta foi a principal motivação para o desenvolvimento deste trabalho. Foram estudadas as principais ferramentas e protocolos envolvidos em uma rede de telefonia IP, seguindo pelo levantamento de necessidades e possíveis soluções para o ambiente VoIP da UFSC e por fim, o desenvolvimento de uma aplicação voltada à convergência de dados em um ambiente de telefonia IP.

Palavras-chave: VoIP, Telefonia IP, Convergência de dados, SIP.

LISTA DE FIGURAS

Figura 1 - Exemplo de sessão SIP.....	16
Figura 2 - SIP Invite.....	16
Figura 3 - Backbone RNP.....	22
Figura 4 - Descrição funcional da PBX-IP.....	24
Figura 5 - Utilização da classe TemplateHandler.....	27
Figura 6 - Arquitetura do portal do usuário.....	28
Figura 7 - Portal do usuário: Lista de contatos.....	29
Figura 8 - Portal do usuário: Webphone Java.....	31
Figura 9 - Funcionamento do Click2Call.....	32
Figura 10 - Exemplo de mensagem HTTP Push.....	35
Figura 11 - Portal do usuário: Correio de voz.....	36
Figura 12 - SIP Notify para Correio de Voz.....	37
Figura 13 - Trigger de contabilização da PBX-IP.....	37
Figura 14 - Portal do Usuário: Configurações.....	38
Figura 15 - Portal do Usuário: Definição de senha.....	40

LISTA DE TABELAS

Tabela 1 - Propriedades dos codecs	20
Tabela 2 - Componentes da PBX-IP	23

LISTA DE ABREVIATURAS E SIGLAS

CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata)

DOM – *Document Object Model* (Modelo de Objetos de Documentos)

HTML – *Hypertext Markup Language* (Linguagem de Marcação Hipertexto)

HTTP – *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertextos)

IAX – *Inter-Asterisk eXchange* (Troca/comunicação entre Asterisk)

IP – *Internet Protocol* (Protocolo de Internet)

JSON – *Javascript Object Notation* (Notação de Objetos Javascript)

PBX – *Private Branch eXchange* (Troca/comunicação de Ramais Privados)

PHP – *PHP: Hypertext Preprocessor* (Pré-processador de Hipertexto)

PSTN – *Public Switched Telephone Network* (Rede pública de telefonia comutada)

RNP – Rede Nacional de Ensino e Pesquisa

RTP – *Real-time Transfer Protocol* (Protocolo de Transporte em Tempo-real)

SIP – *Session Initiation Protocol* (Protocolo de Iniciação de Sessão)

SMTP - Simple Mail Transfer Protocol (Protocolo de Transferência Simples de Correio eletrônico)

UFSC – Universidade Federal de Santa Catarina

VoIP – *Voice over IP* (Voz sobre IP)

WAN – *Wide Area Network* (Rede de longa distância)

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 Objetivo Geral.....	13
1.1.2 Objetivos Específicos	13
2 VOIP	14
2.1 TELEFONIA IP	14
2.2 PROTOCOLOS	14
2.2.1 Protocolos de sinalização.....	15
2.2.2 Protocolos de mídia	18
2.3 SERVIDORES	18
2.3.1 Asterisk	19
2.3.2 OpenSER	19
2.4 CODECS	20
3 TELEFONIA IP - UFSC	20
3.1 FONE@RNP	21
3.2.1 PBX-IP	23
4 CONVERGÊNCIA DE MÍDIAS EM UM AMBIENTE VOIP	25
5 DESENVOLVIMENTO DA SOLUÇÃO	27
5.1 Lista de contatos	28
5.2 Webphone	30
5.3 Mensagens	33
5.4 Correio de voz.....	35
5.5 Fatura	37
5.6 Configurações	38
5.7 Definição de senha.....	39
6 TRABALHOS FUTUROS E CONCLUSÕES	40
6.1 Trabalhos futuros	40
6.2 Conclusões	41
REFERÊNCIAS	43
APÊNDICES	47

1 INTRODUÇÃO

Quando observamos o iminente crescimento da Internet e de todas as tecnologias que a envolvem, podemos claramente perceber a tendência na convergência de dados e aplicações. Em telecomunicações, um dos principais conceitos relacionados à convergência, integração e interoperabilidade é o de Comunicações Unificadas. Segundo PLEASANT (2009), Comunicações Unificadas são comunicações integradas a fim de otimizar o processo de negócio. Atualmente utilizamos diversas formas para nos comunicar, como por exemplo o telefone, *email*, IM (*Instant Message*), SMS (*Short Message Service*) video-conferência, redes sociais, entre outros. Estes meios de comunicação podem ser integrados em uma única aplicação com o objetivo de proporcionar maior interatividade, agilidade e conveniência para seus usuários.

Há algum tempo a telefonia está deixando de ser uma forma de apenas efetuar e receber chamadas e está introduzindo diversos recursos conjuntos, tais como identificação de números de origem, redirecionamento de chamadas, serviço de secretária eletrônica e fax. Todos estes recursos citados estão disponíveis mesmo na telefonia convencional, porém com certas limitações. Na telefonia IP, temos maior flexibilidade e ferramentas disponíveis que possibilitam o desenvolvimento de diferentes aplicações capazes de integrar serviços e equipamentos, como por exemplo, video-conferências, envio de recados para *email* e *softphones*¹. Neste trabalho serão apresentadas detalhadamente algumas soluções visando atender às necessidades levantadas na rede de telefonia IP da UFSC.

O aproveitamento da rede de computadores (WAN – *Wide Area Network*) para o transporte dos pacotes de voz da telefonia VoIP favorece sua implantação uma vez que praticamente não há necessidade de utilização de canais de transporte exclusivos para este fim. Além disto, a redução de custos é facilmente percebida em chamadas de longa distância, principalmente entre instituições interconectadas via IP, onde o custo é de uma chamada local quando a instituição de destino completa ligações para a rede pública de telefonia, ou nulo, caso a chamada seja destinada à um ramal dentro da instituição. Todos estes fatores contribuíram e contribuem para a substituição da telefonia convencional pela telefonia IP.

¹ Softphone: Software projetado para realizar e receber chamadas VoIP

Podemos observar esta mudança na própria Universidade Federal de Santa Catarina, a qual adquiriu entre os anos de 2009 e 2010, cerca de 650 telefones IP e já se planeja adquirir mais 1000 aparelhos no próximo ano, conforme a demanda.

Outra motivação para a implantação da telefonia IP na UFSC foi o fone@RNP, que será apresentado no decorrer deste trabalho.

1.1 OBJETIVOS

Este trabalho visa estudar e aplicar soluções para a convergência de mídias, seja em formato de texto, áudio, imagem ou vídeo, em uma estrutura de telefonia IP.

1.1.1 Objetivo Geral

O objetivo deste trabalho é estudar o ambiente VoIP da Universidade Federal de Santa Catarina, analisando e descrevendo as tecnologias, ferramentas e protocolos envolvidos, para que com as informações obtidas, seja possível identificar novas soluções para a rede que empreguem o conceito de convergência de mídias.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho que podem ser listados são:

1. Estudar as tecnologias utilizadas em um ambiente VoIP;
2. Explorar e entender a rede de telefonia IP da UFSC;
3. Identificar necessidades gerais e específicas do ambiente;
4. Modelar e descrever possíveis soluções encontradas;
5. Definir os requisitos para a aplicação a ser implementada;
6. Desenvolver uma aplicação para convergência de mídias de acordo com o escopo definido, de forma a atender os objetivos identificados.

2 VOIP

VoIP - *Voice over IP* (Voz sobre IP) é o termo dado à tecnologia de transmissão de dados de voz através de uma rede de comunicação baseada em IP.

Uma definição mais técnica, proposta por AMARAL, da Cisco Certified (2008), diz que VoIP é uma propagação do tráfego de voz, seja ele em modo *half-duplex* ou *full-duplex*, sobre um canal de rede IP.

De acordo com MEGGELEN, MADSEN e SMITH (2007), VoIP não é apenas um simples método de obter chamadas de longa distância gratuitas. Sua principal vantagem é a possibilidade da transmissão de voz não ser nada mais do que outra aplicação na rede de dados. MEGGELEN, MADSEN e SMITH (2007) tratam isto como uma grande revolução, inevitável e que traz consigo importantes ferramentas *open-source*, como o PBX Asterisk.

2.1 TELEFONIA IP

Trata-se simplesmente do VoIP agregado de recursos, onde há a presença de *hardphones* ou *softphones*. Assim, os telefones terão IP, recursos aprimorados e poderão apresentar interatividade com aplicações dos servidores da rede. Os demais produtos que farão parte da rede de telefonia IP também se comunicarão via IP, por exemplo: um FAX, uma vez conectado à um ATA (*Analog Telephone Adapter*) (AMARAL, Italo. Cisco Certified, 2008).

2.2 PROTOCOLOS

Os protocolos descrevem as mensagens e regras para a comunicação dos dados. No VoIP, os protocolos são responsáveis pela sinalização e transporte dos dados de voz entre as partes comunicantes.

2.2.1 Protocolos de sinalização

DINAU e VAZ (2006) explicam que um protocolo de sinalização para VoIP deve especificar a codificação da voz, configuração das chamadas, transporte de dados, modo de autenticação, métodos utilizados na comunicação, cabeçalho, endereçamento e a sintaxe da mensagem. Sinalizar significa que a informação da chamada é carregada através dos limites da rede. Os protocolos de sinalização utilizados na telefonia IP são utilizados para definir as opções de serviço, como a solicitação do início da comunicação, a confirmação ou término do pedido.

SIP

JOHNSTON (2009) define o SIP – *Session Initiation Protocol* como um protocolo de sinalização, presença e mensagem instantânea, desenvolvido para estabelecer, modificar e encerrar sessões multimídia; requisitar e entregar presença; e enviar e receber mensagens instantâneas.

SINNREICH (2004) explica que o protocolo SIP se assemelha aos protocolos HTTP 1.1 e SMTP. Isto representa uma revolução nas comunicações uma vez que abandona os antigos modelos de sinalização e controle desenvolvidos para a telefonia convencional, em lugar de protocolos baseados em Internet e web.

Segundo ROSENBERG (2002), o SIP é baseado em cinco abordagens para o estabelecimento e término de comunicações multimídia:

- Localização do usuário: Determinar o sistema a ser utilizado para a comunicação;
- Disponibilidade do usuário: Verificar o estado de disponibilidade do usuário para o qual está se tentando estabelecer a comunicação;
- Capacidades do usuário: Identificação da mídia e dos parâmetros a serem utilizados;
- Estabelecimento de sessão: Solicitação para estabelecimento da comunicação com o destino;

- Gerenciamento de sessão: Transferência, cancelamento e modificação de sessões.

ROSENBERG (2002) ainda explica que o SIP é baseado em um modelo de transação como o HTTP, no qual há uma requisição e uma resposta. Cada transação consiste em uma requisição feita por um cliente, que é recebida e processada pelo servidor, e que por fim, retorna a resposta ao cliente. No exemplo demonstrado na figura 1, a transação inicia quando o telefone da Alice envia uma mensagem de INVITE destinada ao telefone do Bob. O INVITE é um método SIP utilizado para solicitar ao servidor (Bob) a estabelecer uma sessão com o cliente (Alice). A requisição de INVITE contém uma série de parâmetros em seu cabeçalho (*header*), que servem para prover informações adicionais a respeito da mensagem, como por exemplo: identificador da chamada, endereço do telefone de origem, destino e do servidor VoIP, chaves para autenticação ou tipo de sessão que se pretende estabelecer.

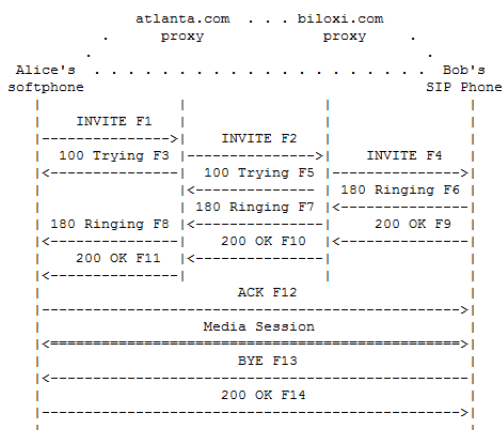


Figura 1 - Exemplo de sessão SIP

Fonte: ROSENBERG, RFC 3261, 2002

O INVITE representado pelo identificador F1 na figura 1, pode ser exemplificado conforme a figura 2.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhd
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

Figura 2 - SIP Invite

Fonte: ROSENBERG, RFC 3261, 2002

As normas RFC 3261, 3265, 3515 e 3428 definem os seguintes métodos de requisição do protocolo SIP:

- REGISTER: Utilizado para o ramal se registrar na central, indicando seu endereço IP atual;
- INVITE: Utilizado para estabelecer uma sessão entre ramais;
- ACK: Confirma o recebimento de uma mensagem;
- CANCEL: Cancela uma requisição pendente, como por exemplo um convite para estabelecer uma chamada;
- BYE: Encerra uma sessão estabelecida (chamada em andamento);
- OPTIONS: Utilizado para obter informações de um ramal, tais como as capacidades do telefone IP;
- SUBSCRIBE: Inscreve-se em um evento ou notificação;
- REFER: Utilizado para transferência de chamadas;
- MESSAGE: Envio de mensagem instantânea;

As mensagens de resposta seguem o mesmo modelo que o protocolo HTTP. Os tipos definidos pela RFC 3261 são os seguintes:

- Temporário (1xx): Requisição recebida e sendo processada;
- Sucesso (2xx): Requisição recebida e processada corretamente;
- Redirecionamento (3xx): Requisição deverá ser redirecionada a outro ponto para que a requisição seja completada com sucesso. Erro no cliente (4xx): Requisição com problemas de sintaxe, parâmetros inválidos, etc;
- Erro no servidor (5xx): Ocorreu um erro ao processar a requisição no servidor;
- Falha global (6xx): A requisição falhou em todas as tentativas.

H.323

O H.323 é um dos protocolos mais maduros, robustos e utilizados nos equipamentos que utilizam VoIP por ser um dos mais antigos e definido especificamente com a finalidade de comunicações multimídia. Ele é o protocolo padrão utilizado em gateways de voz sobre IP das marcas Cisco, Polycom e Siemens, por exemplo, além de estar presente na maioria das centrais telefônicas que possibilitam VoIP (GROSS, 2009).

Aos poucos o H.323 vem sendo substituído por outros protocolos mais novos, como o SIP e o IAX.

IAX

Segundo MONTORO e CASILARI (2009), o IAX - *Inter Asterisk eXchange*, é um novo protocolo desenvolvido para o servidor Asterisk, que reúne as funcionalidades de sinalização de chamadas (H.323/SIP) e o transporte de dados multimídia (RTP). Uma das principais vantagens do IAX é a capacidade de conexões *Trunk*, que divide o *overhead* do protocolo com todas as chamadas entre dois nós IAX. Este recurso reduz significativamente o uso de banda.

2.2.2 Protocolos de mídia

Os protocolos de mídia são utilizados para o transporte dos dados fim-a-fim, isto é, a transferência da voz codificada entre as duas ou mais partes envolvidas em uma chamada VoIP.

RTP

De acordo com GROSS, 2009 *apud* SCHULZRINNE, H. et al:

O RTP provê funções de transporte de rede fim-a-fim ajustadas para aplicações transmitindo dados em tempo-real, como áudio, vídeo ou dados simulados, sobre serviços de rede *multicast* ou *unicast*. O RTP não contempla reserva de recursos e não garante qualidade de serviço para serviços em tempo-real.

2.3 SERVIDORES

Servidores são sistemas de computação que fornecem serviços a uma rede de computadores clientes ou a outros servidores.

2.3.1 Asterisk

De acordo com QADEER e IMRAN (2008), PBX – *Private Branch Exchange* é uma rede de telefonia privada utilizada nas organizações a fim de permitir a comunicação por voz através de aparelhos telefônicos ligados a um circuito dedicado. As centrais PBX geralmente possuem grandes instalações para permitir conexão e gerenciamento de vários ramais. Os sistemas PBX podem ser programáveis e customizados para atender as demandas específicas do negócio.

O Asterisk, por sua vez, é uma solução completa de PBX, escrita na linguagem C para ser executada em máquinas com o sistema operacional baseado em Linux. Implementa diversos protocolos para VoIP, podendo interoperar com praticamente todos os equipamentos de telefonia utilizando placas de baixo custo. A comunicação de dados de voz provida pelo Asterisk é flexível, de baixa manutenção e relativamente barata. Efetua a tradução de codecs, PBX *switching* e implementa diversas aplicações como o correio de voz, salas de conferência e IVR. (MAHLER, 2005)

2.3.2 OpenSER

OpenSER é uma implementação *open source* de um servidor SIP, que possibilita o roteamento de chamadas SIP além de outras funcionalidades em nível de aplicação, tal como o serviço de mensagens instantâneas (IM). É um componente essencial em uma solução VoIP baseada em SIP, podendo ser customizado para atender eficientemente as necessidades do ambiente.

Por motivos de conflitos com registros de marcas, em 2008 o projeto foi renomeado para OpenSIPS. Segundo o site oficial do servidor, as principais aplicações do OpenSIPS são:

- Servidor de registro SIP;
- Roteamento (*Proxy*) SIP – LCR, Roteamento dinâmico, plano de discagem;
- Servidor de redirecionamento SIP;

- Agente de presença (*presence*) SIP;
- Servidor de mensagens instantâneas (IM);
- *Gateway* de SIP para SMS (bidirecional);
- Balanceamento de carga SIP;
- Interface para *gateways* e Asterisk;
- Servidor de aplicação SIP;

2.4 CODECS

Codecs são responsáveis pela codificação e decodificação de um sinal ou fluxo de dados digitais. Sua utilização no VoIP se faz necessária para compressão dos sinais digitalizados, fazendo com que haja melhor aproveitamento da banda disponível.

A tabela 1 demonstra as propriedades dos codecs envolvidos na comunicação VoIP, tais como o *bit rate*², *frame size*³ e *data size per frame*,⁴ que indicam a qualidade do áudio e o total de banda utilizado.

Codec	Bit rate	Frame size	Data size per frame
g.711 (A Law)	64 kbps	20 ms	160 bytes
g.726	32 kbps	20 ms	80 bytes
Gsm	13 kbps	20 ms	33 bytes
lpc10	2.4 kbps	22.5 ms	7 bytes

Tabela 1 - Propriedades dos codecs

Fonte: *A Comparative Study of VoIP Standards with Asterisk*, 2009

3 TELEFONIA IP - UFSC

A Rede Nacional de Ensino e Pesquisa é uma organização social criada pelo Ministério da Ciência e Tecnologia (MCT) em 1989, com o objetivo de implantar uma

² Bit rate: Quantidade de bits processados por unidade de tempo

³ Frame size: Duração de cada amostragem

⁴ Data size per frame: Tamanho dos dados transferidos em cada frame

infraestrutura de rede (*backbone*) para a comunidade acadêmica. Foi a primeira rede de acesso à internet no Brasil e atualmente integra mais de 600 instituições de ensino e pesquisa, provendo aplicações e serviços de alta qualidade. Possui um Ponto de Presença (PoP) em cada estado do país e mais um no Distrito Federal. Em Santa Catarina, o PoP-SC está localizado na Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação (SeTIC) - antigo NPD - da UFSC.

3.1 FONE@RNP

fone@RNP é o nome do serviço que permite o encaminhamento de chamadas, isto é, a interconexão das instituições clientes da rede VoIP (Voz sobre IP) da RNP (REDE NACIONAL DE ENSINO E PESQUISA, 2009).

O projeto fone@RNP foi uma das principais motivações para a implantação de um ambiente de telefonia IP na UFSC. Através dele, tornou-se possível a comunicação gratuita entre diversas instituições de ensino e demais clientes ligados à RNP, além de oportunidades de estudo, expansão do VoIP e desenvolvimento de novas soluções.

Atualmente existem 97 instituições usuárias do serviço fone@RNP, sendo a UFSC uma delas.

A figura 3 ilustra de forma simples a estrutura do fone@RNP, onde as instituições cliente como a UFSC estão ligadas aos Pontos de Presença da RNP em cada estado do país, que por sua vez, estão ligados ao *backbone* da RNP. Desta forma, todas as instituições envolvidas tem a sua disposição a infraestrutura necessária para implantação de um ambiente VoIP, que permite a realização de chamadas sem custos entre elas.

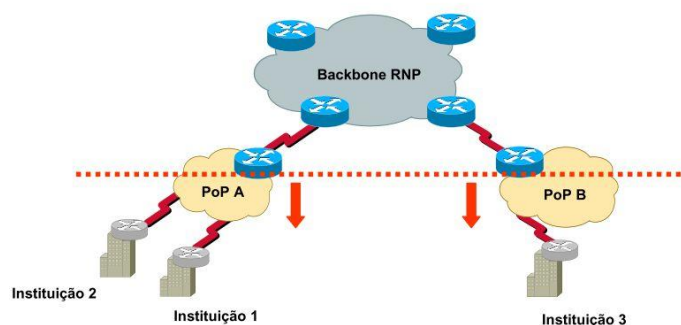


Figura 3 - Backbone RNP

Fonte: Serviço fone@RNP – Descrição geral

3.2 IMPLANTAÇÃO DO VOIP NA UFSC

A primeira forma de implantação do fone@RNP na UFSC foi através do VoIP@UFSC, um serviço de telefonia IP para os alunos, professores e servidores da universidade. Integrado ao sistema RAS (Serviço de Acesso Remoto), possui atualmente mais de 5.000 usuários cadastrados.

As vantagens observadas com a implantação e utilização do VoIP@UFSC somadas a necessidade de evoluir a rede de telefonia da universidade, levaram a criação do VoIP Corporativo, um ambiente de telefonia IP voltado aos reitores, diretores, coordenadores, professores, secretários e demais servidores da UFSC, especialmente aos que já possuíam um ramal da telefonia convencional.

Atualmente o serviço conta com mais de 650 ramais cadastrados, totalizando aproximadamente 180 mil chamadas efetuadas desde o ano de 2008. O objetivo é substituir completamente a infraestrutura de telefonia convencional, proporcionando aos usuários serviços e aplicações de alta qualidade, além de diminuição dos gastos por parte da universidade.

3.2.1 PBX-IP

PBX-IP é o nome dado ao sistema de telefonia IP corporativa, desenvolvido e mantido pela Universidade Federal de Santa Catarina em parceria com a RNP e PoP-SC. É uma solução completa para VoIP adotada também pelo Ministério da Saúde.

A tabela 2 descreve os principais componentes de software da PBX-IP:

Software	Função
Linux Debian	Sistema operacional para a central VoIP
OpenSER	Proxy SIP responsável pelo roteamento das chamadas SIP.
OpenLDAP	Repositório que contem os ramais, unidades, departamentos, telefones e demais dados relacionados ao ambiente.
MySQL	Base de Dados para armazenamento dos usuários ativos no ambiente, administradores da interface web e registro de chamadas efetuadas e perdidas.
Asterisk	Gateway SIP/PBX e serviços adicionais.
MediaProxy	Solução de roteamento de Áudio utilizada para contornar NAT.
Apache	Servidor Web utilizado no ambiente de administração e gerência do ambiente.
PHP	Linguagem de programação da interface de administração da PBX-IP.
NTP	Sincronismo de tempo para que seja possível correlacionar as chamadas.
DNS	DNS Resolver local.
PBX-IP Web	Interface web de gerência da PBX-IP.

Tabela 2 - Componentes da PBX-IP

Os ramais IP fazem o registro no SIP Proxy (OpenSER), que utiliza o diretório LDAP para autenticar os usuários e verificar suas permissões de chamadas, utilizando o banco de dados MySQL para fins de contabilização e controles internos. Quando uma chamada é destinada para a PSTN (telefonia convencional) ou PBX (telefonia convencional interna), o OpenSER encaminha esta chamada para o Asterisk, conectado à PBX, que por sua vez está

conectado com a telefonia convencional (PSTN). O Asterisk utiliza a base MySQL para contabilização das chamadas efetuadas e perdidas e provimento de alguns serviços adicionais, como o correio de voz (secretária eletrônica). As chamadas destinadas aos ramais da PBX-IP fazem o caminho inverso, tornando transparente para ambos os usuários.

Outro componente importante presente na PBX-IP é a interface de administração e gerência, onde o administrador ou operador da central IP pode interagir com a configuração do ambiente, podendo criar novos ramais e organizá-los por unidades e departamentos, alterar parâmetros da telefonia IP, visualizar a contabilização local e *logs* da central IP, acompanhar o status do sistema além de controlar a PBX-IP através de uma interface web amigável e intuitiva.

Na figura 4 estão representados os principais componentes da PBX-IP juntamente com seus relacionamentos.

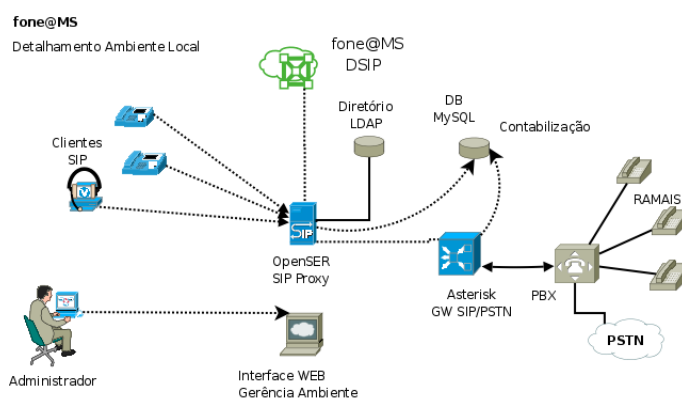


Figura 4 - Descrição funcional da PBX-IP

Fonte: Projeto fone@MS: Telefonia IP para o Ministério da Saúde, 2009

O ambiente foi concebido para trabalhar de forma redundante. Sendo assim, 2 PBX-IPs fazem a redundância em modo ativo/passivo compartilhando o mesmo endereço IP e tendo tanto a base de dados MySQL quanto o diretório LDAP replicados. Os demais componentes estão presentes em ambas as máquinas, podendo trabalhar em redundância ou de forma isolada.

4 CONVERGÊNCIA DE MÍDIAS EM UM AMBIENTE VOIP

Durante o estudo da telefonia IP da UFSC, foi possível identificar diversas oportunidades de desenvolvimento para melhoria do ambiente, tanto do ponto de vista do usuário quanto do administrador. A tendência observada em outras soluções de telecomunicação, como a CISCO Unified Computing (CISCO, 2010), Lync Server da Microsoft (UNIFIED, 2010), Aastra MX-One™ (MX-ONE, 2010), OpenScrape US Server da Siemens (INTEGRATION, 2010), IBM Lotus Sametime (IBM, 2010), Avaya Aura AS5300 (AVAYA, 2010), entre outras, indicam que a integração entre serviços de mensagem instantânea, compartilhamento, redes sociais e outros é algo indispensável para melhoria da produtividade e interatividade dos usuários. Todas estas soluções parecem caminhar para o conceito de Comunicações Unificadas (UC), que segundo MCCHARLES (2010), é uma arquitetura de tecnologia de comunicação que automatiza e unifica várias formas e dispositivos de comunicação humana em um contexto. O objetivo principal da UC é otimizar processos de negócios e melhorar a comunicação humana, reduzindo a latência e a dependência em mídias e dispositivos e específicos.

Na PBX-IP, uma das principais necessidades identificadas foi oferecer serviços que possibilitem a integração e convergência de mídias. Atualmente os usuários dispõem apenas de recursos básicos da telefonia, como o recebimento, efetuação e transferência de chamadas. Existe também o serviço de correio de voz, porém com acesso único através do telefone, onde a navegação é difícil.

Com base nisto, foram levantadas sugestões de aplicações avançadas, que aumentem a flexibilidade, interatividade e integração nos serviços de comunicação.

Diversas propostas de desenvolvimento e demandas foram identificadas, as quais estão listadas abaixo:

- Correio de voz via web: Interface para ouvir, excluir e pesquisar mensagens deixadas na secretária eletrônica;
- Chat: Cliente para envio e recebimento de mensagens instantâneas entre os ramais do ambiente de telefonia IP, utilizando o protocolo XMPP (Jabber);

- Presença: Sistema para o usuário dizer onde ele se encontra disponível (*email*, *chat*, telefone, celular, etc);
- Integração entre FAX e *email*: Possibilidade de envio de um FAX diretamente para uma conta de *email* e vice-versa;
- Central de mensagens: Sistema para envio de mensagens para um telefone IP ou grupo de ramais;
- Fatura *online*: Possibilidade de visualizar a relação de chamadas efetuadas e recebidas através de uma interface web;
- Configurações de serviços: Possibilidade de ativar e desativar serviços adicionais como o correio de voz, hora certa, envio do saldo de crédito após cada ligação efetuada, entre outros, através de um gerenciador via web;
- Ligações particulares: Sistema para informar as ligações pessoais efetuadas a partir do ramal corporativo;
- Visualização de créditos via web;
- Discar um número para receber seu saldo de créditos;
- Lista de contatos: Criação de uma agenda de contatos por ramal;
- Sala de conferência: Possibilidade do usuário criar salas para conferências utilizando o Asterisk;
- Webphone: Desenvolvimento de um cliente SIP via web para realização de chamadas.

Após o levantamento das soluções, o próximo passo foi analisar quais seriam desenvolvidas, levando em conta suas respectivas complexidades, utilidades e a relação com o tema do trabalho.

Tendo em vista os conceitos de convergência em VoIP e Comunicações Unificadas, a solução definida neste projeto foi o desenvolvimento de uma plataforma web que agregaria diversos serviços para o usuário final. Desta forma os recursos oferecidos pela telefonia poderiam ser acessados não apenas a partir de telefone IP ou *softphone*, mas também através de um portal na internet.

5 DESENVOLVIMENTO DA SOLUÇÃO

Neste trabalho foi desenvolvido um portal web específico aos usuários do serviço de telefonia IP corporativa da UFSC. O portal do usuário nada mais é que um local onde o usuário do ramal pode ter acesso a serviços complementares no ambiente VoIP, através de um navegador web.

A linguagem de programação escolhida para desenvolvimento do lado servidor foi o PHP, por ser a linguagem atualmente adotada em outras aplicações da PBX-IP. A interface web, acessada pelo cliente, foi desenvolvida utilizando HTML para a estrutura e apresentação das páginas, Javascript para requisições AJAX, manipulação do DOM, efeitos visuais e CSS para o estilo das páginas. A biblioteca jQuery foi adotada para a utilização do Javascript e também como base para outros *plugins*.

A estrutura de arquivos foi dividida em duas partes. Uma contendo as classes, bibliotecas, arquivos de configuração, *templates* e *scripts* pertencentes à lógica de negócio, utilizados na parte do servidor de aplicação e outra com os arquivos relacionados à visualização e interface pública, como imagens, *javascripts* e páginas HTML.

A fim de permitir a fácil inclusão de novos serviços, o portal foi desenvolvido de forma modular, eliminando a necessidade de modificação de *layouts*, controle de sessões, conexão ao banco de dados e diretório LDAP. Para a criação de uma nova página, basta utilizar a classe TemplateHandler, a qual trata a inclusão dos arquivos necessários para a montagem da estrutura de apresentação (templates HTML, CSS e javascripts). O exemplo da figura 5 mostra a forma de utilização desta classe.

```
<?php
require_once(dirname(__FILE__) . '/../application/include/TemplateHandler.php');

$template = new TemplateHandler();
//Arquivos CSS a serem utilizados na página:
$template->set_css_files(array(
    'painel.css',
    'jplayer.blue.monday.css'
));
//Arquivos Javascript a serem utilizados na página:
$template->set_js_files(array(
    'jquery.jplayer.min.js',
    'jquery.form.js'
));
//Arquivo contendo o conteúdo principal:
$template->set_content_file('painel/painel.php');
//ID da seção (página) para ser destacado no menu:
$template->set_active_nav_item(ID_PAINEL);
//Somente visível para usuário autenticados:
$template->set_authenticated_only(true);
//Exibir a página:
$template->show();
?>
```

Figura 5 - Utilização da classe TemplateHandler

Apesar de herdar diversas características do gerenciador web da PBX-IP, o portal foi projetado para ser portátil a outros ambientes de telefonia IP, sendo necessário apenas adaptações no nível de dados (estrutura do banco de dados e LDAP).

A arquitetura de funcionamento do Portal do Usuário (figura 6) consiste basicamente em quatro elementos principais:

- Usuário, utilizando um navegador web, faz requisições HTTP ao servidor;
- Servidor Web, que atende as requisições do usuário, realiza as consultas nos bancos de dados (LDAP e MySQL) quando necessário, processa e retorna os dados ao navegador web;
- Diretório LDAP, onde estão armazenados os dados dos ramais VoIP e convencionais, unidades e departamentos da instituição, dentre outras estruturas que fazem parte da PBX-IP;
- Banco de dados MySQL, responsável pelo armazenamento dos dados dos usuários do portal, registros de contabilização de chamadas efetuadas e perdidas, etc.

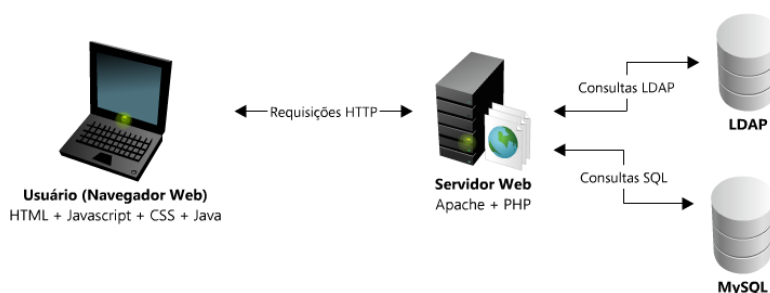


Figura 6 - Arquitetura do portal do usuário

Tendo a plataforma já estabelecida, a próxima etapa foi o planejamento dos serviços a serem agregados ao portal do usuário.

5.1 Lista de contatos

O gerenciamento da agenda de contatos disponibilizada pelos aparelhos telefônicos é complexo devido à interface com o usuário. Os telefones Polycom IP 331, IP 430 e IP 630, oferecidos aos usuários dos ramais na UFSC, possuem apenas um teclado numérico, o tamanho e resolução da tela são extremamente limitados e não existe a possibilidade de

importar ou exportar os contatos cadastrados. Em função disso, grande parte dos usuários não utiliza ou utiliza muito pouco este recurso no aparelho, o que levou à necessidade de desenvolvimento de uma nova agenda de contatos.

A nova agenda (figura 7), disponível no Portal do Usuário, oferece uma interface intuitiva para o cadastramento, edição, exclusão e localização de contatos. Um módulo apenas para o gerenciamento foi desenvolvido, no qual o usuário pode buscar dentre os ramais já cadastrados na PBX-IP e adicioná-los em sua lista pessoal. Ao digitar parte do nome da pessoa, ramal ou *email*, o sistema faz uma pesquisa dinâmica, utilizando AJAX, no diretório LDAP, retornando os ramais IP e convencionais que atendam os critérios da busca e disponibiliza esses ramais para o usuário realizar a escolha. Os contatos de cada ramal são armazenados no banco de dados MySQL.

Além do módulo para gerenciamento da agenda, uma lista de contatos é disponibilizada abaixo do *menu* principal, em todas as seções do portal. Desta forma o usuário pode facilmente procurar por um contato e ter acesso a informações relativas a ele, como seu tipo (ramal IP, ramal PBX, telefone externo ou celular), *email* do usuário, departamento e unidade onde está vinculado. Além de informações sobre o ramal, ao passar o mouse sobre um contato da lista, o usuário pode realizar ações específicas, como o envio de *email* ou mensagem para o ramal (SMS), realização de chamada e abertura de *chat* (ainda não disponível).

A lista de contatos é integrada ao *webphone*, onde o usuário tem a possibilidade de efetuar uma chamada para um contato específico a partir de seu registro na lista.

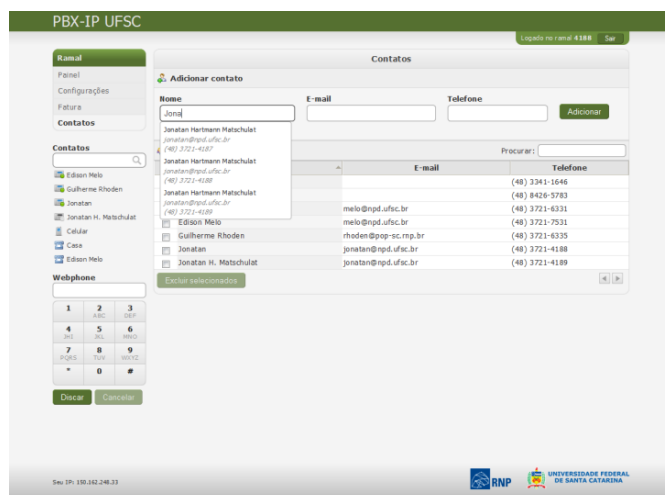


Figura 7 - Portal do usuário: Lista de contatos

5.2 Webphone

Muitas vezes o usuário se encontra longe do seu telefone IP, o que acaba impossibilitando-o de realizar e receber chamadas. Em outros casos, mesmo estando próximo do aparelho, a localização de um contato torna-se difícil devido às limitações de interface, conforme comentado anteriormente. Pensando nisso, surgiu a oportunidade de inclusão de um telefone virtual que possa ser utilizado no próprio computador do usuário, sem que haja necessidade de instalação de um *softphone*. Este telefone virtual, chamado de *webphone*, poderia ser integrado à agenda de contatos com o objetivo de facilitar a realização de chamadas.

Após uma pesquisa de soluções existentes no mercado e do estudo das mensagens de sinalização utilizando o protocolo SIP, foram levantadas as três seguintes alternativas: Java Applet, Click2Call e Integração com *Softphone*.

Java Applet

A alternativa mais completa encontrada foi uma solução disponibilizada em Java Applet, desenvolvida pela empresa Mizutech. Por ser implementada em Java, sua utilização não depende de uma plataforma específica, podendo ser executada em qualquer navegador que tenha o Java instalado, independente do sistema operacional. Como os navegadores atualmente não oferecem meios para utilização do microfone do usuário, a utilização do Java Applet ou similares como Flash e ActiveX se faz necessária.

Segundo a Mizutech, soluções em Flash não possuem os *codecs* comumente utilizados em VoIP, como o g.711, gsm, speex, g.726 ou g.729, tendo apenas um *codec* proprietário. Isto faz com que seja necessária a utilização de um servidor de mídia para realizar a conversão, o que pode resultar em perda da qualidade dos dados de voz.

Webphones desenvolvidos em ActiveX rodam apenas em sistemas Windows e podem oferecer riscos de segurança. De acordo com a Mizutech, o ActiveX é uma tecnologia ultrapassada e, por padrão, desabilitada na maioria dos navegadores.

O *webphone* desenvolvido pela Mizutech, além de ser fácil de usar e independente de plataforma, possui suporte para diversos *codecs* e métodos SIP, agindo como um *softphone* completo compatível com servidores VoIP baseados no protocolo SIP. Outra vantagem observada foi a disponibilização de uma API em Javascript, o que facilita muito sua customização e integração com outros elementos de uma aplicação web.

A versão para testes do *webphone* possui limitações quanto a duração da chamada (90 segundos), ao número de instâncias rodando ao mesmo tempo (10), quantidade de chamadas efetuadas por sessão (10) e ao tempo de execução (20 minutos). As licenças do produto custam entre US\$ 400,00 e US\$ 3.900,00.

No portal do usuário, foi implementado um *plugin* em Javascript (figura 8) para realizar as chamadas na API do *webphone*, tornando-o integrado à lista de contatos e *layout* da aplicação.

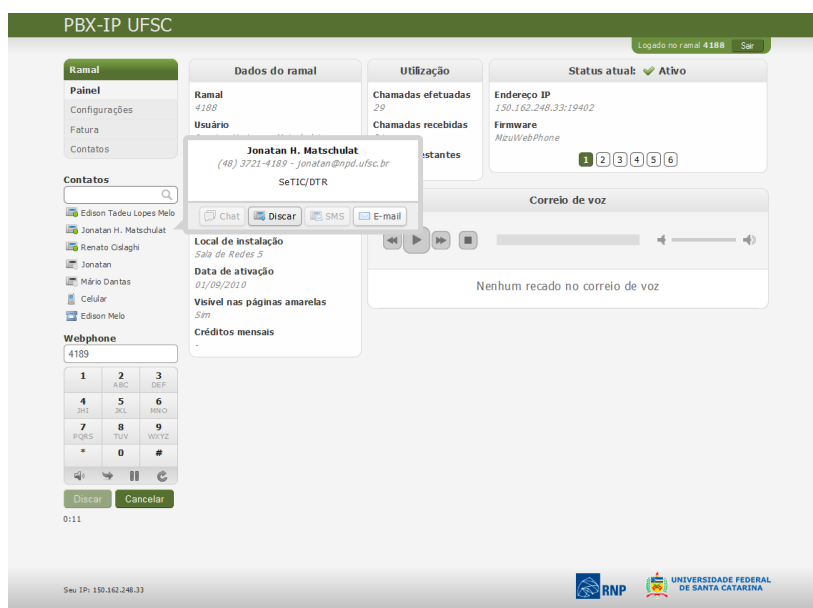


Figura 8 - Portal do usuário: Webphone Java

Click2Call

Diferente da solução em Java Applet, o Click2Call (Clique para Discar) suporta apenas a efetuação de chamadas, sem o transporte dos dados de voz, ou seja, somente

sinalização SIP. Sua principal vantagem é a independência de plataformas externas como o Java, rodando diretamente no servidor web.

A biblioteca php-sip, desenvolvida por Chris Maciejewski, foi utilizada na implementação desta aplicação. Lançada em 2009, a classe php-sip oferece recursos para o envio e recebimento de mensagens SIP, funcionando como um *User Agent* (MACIEJEWSKI, 2009).

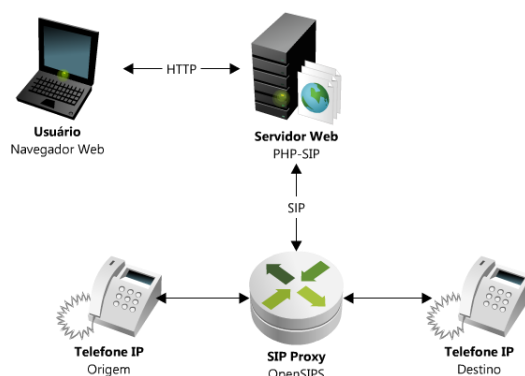


Figura 9 - Funcionamento do Click2Call

O funcionamento do Click2Call (figura 9) pode ser resumido da seguinte forma:

1. Usuário informa o número de destino e submete o formulário ao clicar no botão Discar;
2. Servidor web envia a mensagem de INVITE para o ramal do usuário que está efetuando a chamada;
3. Ao aceitar o INVITE, atendendo o telefone, o servidor web imediatamente envia uma mensagem de REFER para efetuar a transferência da chamada para o número de destino;
4. Servidor web encerra a chamada enviando uma mensagem de BYE ao ramal de origem;
5. Ao receber o REFER do servidor web, o ramal de origem envia um INVITE para o número de destino.

Para permitir o cancelamento da chamada antes de sua efetuação, ou seja, enquanto o telefone está tocando, foi necessário realizar ajustes na biblioteca php-sip e na sua forma de

execução. Por funcionar em um modelo cliente-servidor onde cada requisição possui instâncias independentes, ao fazer o cancelamento da chamada, é preciso recuperar os dados da requisição feita anteriormente (INVITE), como o ID (CallID) e sequência (Cseq) da chamada. Para isto, foi necessário armazenar estas informações em um arquivo temporário para que pudessem ser recuperadas e incorporadas na mensagem de CANCEL. Além disto, para evitar conflitos na abertura de conexões via *socket*, foi necessário criar processos a serem rodados em segundo plano, em cada requisição (discagem e cancelamento), simulando um comportamento *multithreading*.

Integração com Softphone

A terceira opção oferecida ao usuário que possui um *softphone* instalado na máquina é a integração através dos *links sip*. A maior parte dos navegadores e *softphones* oferece suporte para URLs do tipo sip:// ou sip:, composto pelo número de destino e o endereço do servidor VoIP, como por exemplo: sip://4188@adm.voip.ufsc.br. Desta forma, ao clicar sobre um link deste tipo, o *softphone* instalado no computador do usuário é acionado, efetuando a chamada para o respectivo número.

Assim como o Click2Call, este recurso funciona apenas para efetuar chamadas, podendo ser integrado à outras aplicações do portal, como a agenda, a fim de facilitar o encontro e discagem para contatos.

5.3 Mensagens

O serviço de mensagens tem por objetivo oferecer uma alternativa para envio de pequenas mensagens, funcionando como o SMS – *Short Message Service*. O serviço pode ser utilizado para divulgação de avisos, alertas ou recados para grupos de ramais ou um ramal específico. As mensagens enviadas devem ser apresentadas na tela dos aparelhos telefônicos IP.

Para o desenvolvimento desta aplicação, foi necessário estudar os telefones IPs, mais especificamente os modelos do fabricante Polycom, disponíveis aos usuários dos ramais na universidade. A Polycom oferece em seu website informações sobre o desenvolvimento de aplicativos em seus telefones IP. Para os modelos Polycom® SoundPoint® são disponibilizados guias de desenvolvimento, SDK e um pacote de aplicações para demonstração. A UFSC possui os modelos desktop Polycom IP 331, IP 430 e IP 630.

Para ter acesso aos guias e SDK – *Software Development Kit* foi necessário preencher um formulário informando quais os motivos do uso. Tal SDK inclui um simulador, uma ferramenta para testes da API web e local, em XML e exemplos de aplicações.

Um recurso bastante interessante oferecido nos modelos SoundPoint IP 330/320, 430, 501, 550, 560, 600, 601, 650, e 670 é o Microbrowser (Micronavegador). O Microbrowser funciona como qualquer outro navegador web, que faz requisições em um servidor web, processa e imprime os resultados. A maior diferença é que ele possui limitações quanto ao conjunto de tags XHTML e processamento gráfico. O navegador do telefone também não oferece suporte a Javascript. Com isso, as aplicações web, desenvolvidas em XHTML, são acessíveis nos telefones que possuem suporte ao recurso.

O funcionamento do microbrowser pode ser descrito da seguinte maneira:

- 1) Download e armazenamento do conteúdo XHTML na memória do telefone;
- 2) Análise da resposta obtida (XHTML) e renderização das *tags* para o *display* do telefone;
- 3) Impressão dos resultados.

A aparência das páginas XHTML é muito simples e varia de acordo com o modelo de telefone (tamanho do *display*, cores, etc). A resolução das telas variam entre 102x22 pixels e 320x160 pixels.

Outro recurso suportado pelos telefones IPs é o HTTP Push, o qual permite o envio de uma mensagem ou URL a ser apresentada no aparelho. Através do simulador incluído no SDK, é possível testar tal recurso.

O formato de uma requisição via HTTP Push segue o exemplo da figura 10:

```
<PolycomIPPhone>
  <Data priority="critical">
    <html>
      <title>Mensagem de: 4188</title>
      <body>Teste</body>
    </html>
  </Data>
</PolycomIPPhone>
```

Figura 10 - Exemplo de mensagem HTTP Push

Para envio da mensagem é necessário que o telefone tenha a opção habilitada. Além disso, a requisição deve ser autenticada através do HTTP Digest. O envio deve ser feito diretamente ao IP do aparelho, da seguinte maneira: `http://150.162.248.43/push`. A biblioteca `cURL`⁵ para PHP foi utilizada para a transmissão das mensagens, facilitando o gerenciamento de autenticação, cabeçalhos HTTP e tratamento de erros.

5.4 Correio de voz

Outro serviço cuja usabilidade é prejudicada devido aos recursos limitados do telefone IP, é o correio de voz. Para ter acesso às mensagens deixadas na caixa é necessário discar para um número específico, digitar a senha de acesso e navegar pelos botões do teclado, utilizando DTMF - *Dual-Tone Multi-Frequency*. Este processo é vagaroso e portanto muitos usuários dificilmente verificam os recados deixados em sua caixa de mensagens.

A fim de melhorar o uso da secretária eletrônica (correio de voz), foi desenvolvida uma interface web (figura 11) na qual o usuário pode ouvir e apagar os recados de forma rápida e simples. O *plugin* para jQuery⁶ utilizado para reprodução das mensagens foi o `jPlayer`⁷, o qual aproveita os novos recursos de áudio providos pelo e para os navegadores que ainda não o suportam, uma alternativa em Flash. As mensagens do correio de voz são gravadas no banco de dados MySQL em campos do tipo BLOB - *Binary Large Object*, no formato de áudio wav. Porém, como o *plugin* `jPlayer` não suporta arquivos de áudio em tal formato, foi necessário desenvolver um *script* para conversão sob-demanda das mensagens

⁵ `cURL`: Biblioteca que auxilia na comunicação entre diversos tipos de servidores e protocolos

⁶ `jQuery`: Biblioteca projetada para simplificar o uso do Javascript em diferentes navegadores

⁷ `jPlayer`: *Plugin* para jQuery para reprodução de áudio no navegador

para que pudessem ser reproduzidas. Este script faz uso do LAME para codificar o áudio mp3 e o ffmpeg para fazer a conversão do wav para mp3.

O funcionamento do script de conversão do áudio das mensagens no correio de voz acontece da seguinte maneira:

1. Usuário requisita o áudio para o servidor através da interface web (jPlayer);
2. Servidor consulta no banco de dados MySQL, obtendo o binário (BLOB) referente ao áudio no formato wav;
3. Servidor armazena o conteúdo binário em um arquivo temporário (temp.wav);
4. Servidor executa o comando `ffmpeg -i <arquivo.wav> <arquivo.mp3>` no terminal para efetuar a conversão do áudio;
5. Servidor lê o arquivo mp3 gerado e retorna o conteúdo para o usuário, utilizando o cabeçalho HTTP referente ao tipo de arquivo (Content-Type: audio/mpeg);
6. jPlayer reproduz o áudio para o usuário.

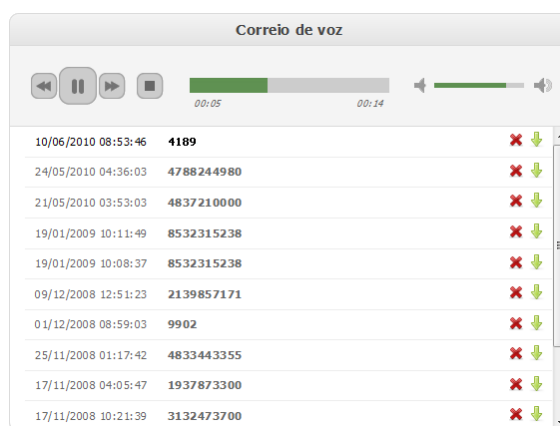


Figura 11 - Portal do usuário: Correio de voz

Além de reproduzir os recados do correio de voz, o sistema é integrado à central VoIP, que notifica o telefone IP do usuário quando um recado é lido. Para isto, após a reprodução do áudio ou exclusão de uma mensagem, uma consulta é feita no banco de dados MySQL para obter a quantidade de recados lidos e não lidos na caixa de mensagens do usuário. Estes dados são enviados ao ramal através de uma mensagem SIP NOTIFY (figura 12) que é tratada e apresentada ao usuário pelo telefone IP.

```

NOTIFY sip:4188@adm.voip.ufsc.br SIP/2.0.
Via: SIP/2.0/UDP 150.162.2.20:5173;rport;branch=z9hg4bk286270.
From: <sip:4188@150.162.2.20>;tag=84808.
To: <sip:4188@adm.voip.ufsc.br>.
Call-ID: d1e58a5060f1751deb631160d0ec98ba@150.162.2.20.
CSeq: 20 NOTIFY.
Contact: <sip:4188@150.162.2.20:5173>.
Content-Type: application/simple-message-summary.
Max-Forwards: 70.
User-Agent: PHP SIP.
Event: message-summary.
Content-Length: 43.
.
Messages-waiting: no
voicemail: 0/8 (0/0)

```

Figura 12 - SIP NOTIFY para Correio de Voz

5.5 Fatura

O módulo de fatura permite ao usuário gerar relatórios da utilização de seu ramal. A interface permite a seleção do período no qual as chamadas foram efetuadas e recebidas para que com base nestes valores, o sistema faça a geração da fatura no formato PDF.

Os registros de eventos SIP são armazenados no banco de dados através do OpenSIPs (OpenSER). Para contabilizar as chamadas completadas e perdidas foi necessário criar um *trigger* (gatilho) no MySQL, que consiste em consultar os parâmetros das mensagens de INVITE e BYE de uma determinada chamada, armazenando o resultado em uma nova tabela. A figura 12 representa a sintaxe utilizada para a criação do *trigger* responsável pela contabilização.

```

1  DELIMITER $$
2
3  USE `voip`$$
4
5  DROP TRIGGER /*!50032 IF EXISTS */ `voipacc`$$
6
7  CREATE
8  /*!50017 DEFINER = 'root'@'localhost' */
9  TRIGGER `voipacc` AFTER INSERT ON `acc`
10 FOR EACH ROW BEGIN
11   IF NEW.method = 'BYE' THEN
12     INSERT INTO completed_calls (
13       SELECT DISTINCT
14         a.time AS _datetime,
15         a.from_number AS _from,
16         a.to_number AS _to,
17         TIMEDIFF(b.time, a.time) AS _duration
18       FROM acc a,
19            acc b
20       WHERE a.method = 'INVITE'
21             AND b.method = 'BYE'
22             AND a.callid = b.callid
23             AND a.callid = NEW.callid);
24   END IF;
25 END;
26 $$
27
28 DELIMITER ;

```

Figura 13 - Trigger de contabilização da PBX-IP

Essa *trigger* foi criada para outros propósitos na PBX-IP e, portanto, nesta aplicação foi preciso apenas obter os dados já consolidados nas tabelas de chamadas completadas e perdidas.

Embora a interface de gerência web da PBX-IP já ofereça serviços de geração de relatórios, tanto gerenciais quanto operacionais, optou-se por desacoplar este recurso no portal do usuário, ou seja, não importar as classes e bibliotecas desenvolvidas no gerenciador PBX-IP. Ao invés disso, o código foi adaptado e inserido diretamente no portal do usuário, permitindo sua utilização em outros ambientes VoIP ou máquinas distintas.

5.6 Configurações

Outra necessidade identificada foi a de oferecer ao usuário a possibilidade de atualizar seus dados pessoais ou ainda ativar/desativar serviços avançados da telefonia IP, como o Correio de Voz, Hora Certa, Data Certa, entre outros.

The image shows a web-based configuration interface for a PBX-IP system. The header indicates the user is logged in as 'ramal 4188'. The left sidebar contains a navigation menu with options: 'Ramal', 'Configurações', 'Fatura', 'Contatos', 'Contatos' (with a search bar), and 'Webphone'. The main content area is titled 'Configurações' and is divided into several sections:

- Seus dados:** Fields for 'E-mail' (jonatan@npd.ufsc.br), 'Local' (Sala de Redes 5), and 'Senha' (with a placeholder 'Digite a nova senha aqui').
- Serviços:** A list of services with checkboxes:
 - Correio de Voz - Secretaria Eletronica - Senha: [placeholder]
 - Correio de Voz - Anexar recado para o e-mail
 - Correio de Voz - Deixar recado para o Ramal
 - Hora Certa
 - Data Certa
 - Teste de ECO para verificar audio e latência
- Webphone:** Radio buttons for selecting the webphone type:
 - Java (Efetua e recebe ligações, utilizando o microfone e som do seu computador)
 - Click2Call (Apenas efetua ligações a partir de seu ramal)
 - Softphone (Apenas efetua ligações a partir do softphone instalado em seu computador)
- Número máximo de contatos na lista lateral:** A dropdown menu set to '15'.

At the bottom, there are 'Discar', 'Cancelar', and 'Salvar' buttons. The footer contains the text 'Seu IP: 190.162.246.33' and logos for 'RNP' and 'UNIVERSIDADE FEDERAL DE SANTA CATARINA'.

Figura 14 - Portal do Usuário: Configurações

A tela de configuração desenvolvida (figura 14), além das opções de ajustes de serviços e dados pessoais, oferece a possibilidade de configuração de opções do próprio portal do usuário, como por exemplo o tipo do *webphone* desejado: Java, Click2Call ou Softphone.

5.7 Definição de senha

O sistema de definição de senha foi projetado para permitir a ativação das contas dos ramais que ainda não possuem uma senha para acesso ao Portal do Usuário, ou para aqueles que esqueceram sua senha.

Para realizar a definição ou redefinição de senha, o usuário primeiramente deverá informar seu ramal na página “Esqueci minha senha”. Ao confirmar o número do ramal, um código de verificação será gerado e a central de definição de senha entrará em contato automaticamente, utilizando o mesmo princípio do Click2Call. Na URA (Unidade de Resposta Audível) o usuário deverá informar o código gerado anteriormente através do teclado numérico do telefone. Assim que validado, um *email* será enviado para a conta do usuário contendo um segundo código de verificação, que deverá ser inserido na interface web para completar o processo de definição de senha. O tempo limite para efetuar a definição é de 30 minutos. Após este período o usuário deverá reiniciar o processo.

Cada etapa do processo corresponde a um status que é armazenado em uma tabela do banco de dados e segue os seguintes valores:

1. Informou o ramal;
2. Atendeu a chamada;
3. Informou o código na central de redefinição (URA);
4. Verificou o email;
5. Informou o código recebido no email;
6. Cadastrou a nova senha de acesso ao portal.

À medida que o usuário realiza as ações solicitadas, o status é atualizado e apresentado ao usuário, conforme demonstra a figura 15.

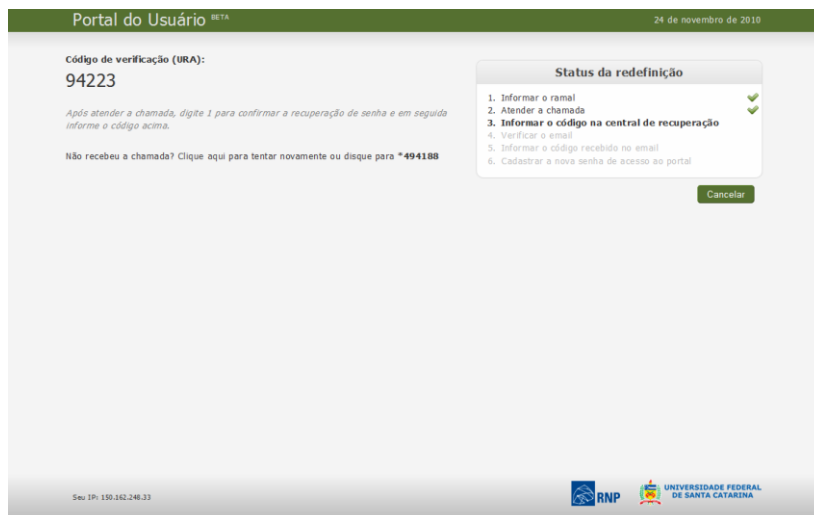


Figura 15 - Portal do Usuário: Definição de senha

A Unidade de Resposta Audível foi programada no Asterisk através do arquivo “extensions.conf”, utilizando comandos como o “Background” para reprodução do áudio contendo as instruções de procedimento, “Shell” para a chamada do *script* Perl responsável pela verificação do código e envio do *email* e “WaitExten” para aguardar enquanto o usuário digita as opções ou o código de verificação.

Para verificar a leitura do *email* foi inserida uma pequena imagem junto ao corpo da mensagem utilizando HTML que quando requisitada ao servidor web, é passado um parâmetro com o número do ramal o qual é utilizado para a atualização do status do processo no MySQL. Após verificar e processar a atualização, o servidor gera a imagem utilizando a biblioteca GD para PHP e em seguida retorna ao cliente de *email* do usuário.

Este serviço é mais um exemplo de integração possível entre aplicações web, *email* e um ambiente VoIP.

6 TRABALHOS FUTUROS E CONCLUSÕES

6.1 Trabalhos futuros

Tendo uma plataforma para desenvolvimento estabelecida juntamente com algumas aplicações, a expansão torna-se mais fácil. Com base nas possíveis soluções e demandas

levantadas anteriormente, as seguintes possibilidades poderão ser exploradas futuramente, a fim de oferecer novas facilidades ao usuário da telefonia IP da UFSC e provendo maior convergência de mídias.

- Correio de voz: Informar o nome do contato que deixou a mensagem no correio de voz, caso esteja na base de dados de ramais ou de contatos do usuário;
- Telefone: Ao receber uma ligação de um contato cadastrado na lista, informar seu nome no visor do telefone e *webphone*;
- Microbrowser Telefone IP: Serviços adicionais disponíveis através de uma aplicação web, acessível pelo navegador do telefone IP, como por exemplo uma aplicação para a Central de Mensagens;
- *Webphone*: Desenvolvimento de um telefone virtual próprio, possivelmente em Java Applet;
- Conferência: Desenvolvimento de um sistema para gerenciar salas para conferência via Asterisk;
- *Chat*: Sistema de mensagem instantânea utilizando SIP;
- Presence: Integração com diferentes tipos de meios de comunicação, como o *chat*, *email* ou telefone;
- Fax2Mail: Possibilidade de envio de FAX para um *email* e vice-versa ou ainda, envio de um FAX a partir de uma interface web;
- Lista de contatos: Integração da lista de contatos do portal web com a dos telefones IPs;
- Central de Mensagens: Desenvolvimento de uma solução baseada em SIP (SIP MESSAGE), podendo ser compatível com outros aparelhos telefônicos além dos modelos da Polycom Soundpoint IP.

6.2 Conclusões

Neste projeto foi possível estudar os principais componentes envolvidos em um ambiente VoIP, desde os protocolos utilizados até as aplicações de alto nível, oferecidas aos usuário final.

Com base no estudo feito, diversas necessidades foram identificadas, as quais foram mapeadas em possíveis soluções para prover maior convergência e integração entre mídias e aplicações.

O portal do usuário foi desenvolvido com o intuito de agregar diferentes serviços, tal como o telefone virtual (*webphone*), correio de voz, fatura, entre outros, em um único local, acessível através de um navegador web. Tais aplicações proporcionam maior facilidade ao usuário, o que conseqüentemente leva ao aumento de produtividade e atinge os principais objetivos relacionados ao conceito de Comunicação Unificadas, que está intimamente ligado à convergência de mídias.

O planejamento e implantação do portal do usuário abriu portas para o desenvolvimento de novas aplicações, que podem ser facilmente integradas aos serviços já existentes através da adição de novos módulos. Portanto, a continuidade deste trabalho é viável e pode melhorar o ambiente de telefonia IP da Universidade Federal de Santa Catarina.

REFERÊNCIAS

AVAYA Aura. Disponível em <www.avaya.com/usa/product/avaya-aura/> Acesso em 24 de novembro de 2010.

AMARAL, Italo. *VoIP vs Telefonia IP*. Disponível em <<http://blog.ccna.com.br/2008/05/14/voip-vs-telefonip/>> Acesso em 9 de julho de 2010.

CISCO Unified Computing: Unify, Simplify, Amplify. Disponível em <www.cisco.com/web/go/unifiedcomputing/> Acesso em 24 de novembro de 2010.

DINAU, Priscilla; VAZ, Igor. *SIP*. Disponível em <http://www.gta.ufrj.br/grad/06_1/sip/index.html> Acesso em 23 de novembro de 2010.
Universidade Federal do Rio de Janeiro. Engenharia de Computação. UFRJ, 2006.

GROSS, Fábio Danieleski. *Implementando uma Solução VoIP Baseada em Asterisk com Alta-disponibilidade e Balanceamento de Carga*. Florianópolis, SC: Universidade Federal de Santa Catarina, 2009.

IBM Lotus Sametime. Disponível em <www.ibm.com/lotus/sametime/> Acesso em 24 de novembro de 2010.

INTEGRATION Services. Disponível em <<http://www.siemens-enterprise.com/us/products/integration-solutions.aspx>> Acesso em 24 de novembro de 2010.

INTERNATIONAL TELECOMUNICATION UNION. *World Internet Users and Population Stats 2009*. Disponível em <<http://www.itu.int/ITU-D/ict/statistics/>> Acesso em 7 de julho de 2010.

JOHNSTON, Alan B. *SIP, Understanding the Session Initiation Protocol, Third edition*. Norwood, MA: Artech House 2009.

MACIEJEWSKI, Chris. *Click to Call with PHP-SIP*. Disponível em <<http://level7systems.co.uk/en/blog/Click+to+Call+with+PHP-SIP>> Acesso em 23 de novembro de 2010.

MAHLER, Paul. *VoIP Telephony with Asterisk*. Signate, 2005.

MCCHARLES, Rick. *Definition of Unified Communications, Its Features and Benefits*. Disponível em <<http://www.ric.ca/blog/2008/04/definition-of-unified-communic.html>> Acesso em 3 de outubro de 2010.

MEGGELEN, Jim Van; MADSEN, Leif; SMITH, Jared. *Asterisk, The future of telephony, Second Edition*. Sebastopol, CA: O'Reilly Media, Inc, 2007.

MIZUTECH, Mizu Webphone. *A java applet internet phone*. Disponível em <http://www.mizu-voip.com/Portals/0/Files/Mizu_WebPhone.pdf> Acesso em 17 de outubro de 2010.

MONTORO Pablo; CASILARI Eduardo. *A Comparative Study of VoIP Standards with Asterisk*. Spain: University of Malaga, 2009

MX-ONE Telephone System – The open path do UC and Mobility. Disponível em <<http://www.aastra.com/mx-one-telephony-system.htm>> Acesso em 24 de novembro de 2010.

OPENSIPS. *The new breed of communication engine*. Disponível em <<http://www.opensips.org>> Acesso em 14 de outubro de 2010.

PLEASANT, Blair. *UC NOT Convergence on Steroids!* Disponível em <<http://www.ucstrategies.com/detail.aspx?id=1774>> Acesso em 22 de novembro de 2010.

QADEER, Mohammed; IMRAN, Ale. *Asterisk Voice Exchange: An Alternative to Conventional EPBX*. iccee, pp.652-656, 2008 International Conference on Computer and Electrical Engineering, 2008.

REDE NACIONAL DE ENSINO E PESQUISA. *fone@RNP*. Disponível em <<http://www.rnp.br/voip/>> Acesso em 8 de julho de 2010.

REDE NACIONAL DE ENSINO E PESQUISA. *Serviço fone@RNP – Descrição geral*. Disponível em <http://www.rnp.br/_arquivo/voip/fone_rnp.pdf> Acesso em 8 de julho de 2010.

RNP, Escola Superior de Redes. *Projeto fone@MS: Telefonia IP para o Ministério da Saúde*, 2009.

ROSENBERG, et. al. *Standards Track - RFC 3261, SIP: Session Initiation Protocol*, 2002. Disponível em <<http://www.ietf.org/rfc/rfc3261.txt>> Acesso em 22 de novembro de 2010.

SALVADORI, Ivan Luiz. *Implementação de uma rede de voz sobre IP, aplicando propriedades de qualidade de serviço na rede e na interface com o usuário*. 2008. TCC. Universidade Federal de Santa Catarina. Ciências da Computação.

SEARCHUNIFIEDCOMMUNICATIONS.COM, *Unified Communications, Cutting Through The Hype*. Cap. 1, 2008.

SINNREICH, Henry. *SIP: Redefining IP Voice Communications*. Disponível em <<http://www.tmcnet.com/it/0104/0104F-MCI.htm>> Acesso em 10 de outubro de 2010.

UNIFIED Communications. Disponível em <www.microsoft.com/uc/> Acesso em 24 de novembro de 2010.

APÊNDICES

Código-fonte da implementação deste trabalho.

usuario/application/config/constants.php

```
<?php
//IDs:
define('ID_PAINEL', 1);
define('ID_CONFIGURACOES', 2);
define('ID_FATURA', 4);
define('ID_CONTATOS', 8);

//Template:
define('DEFAULT_TITLE', 'PBX-IP UFSC');
define('DEFAULT_GLOBAL_FILE', 'global.php');
define('DEFAULT_HEADER_FILE', 'header.php');
define('DEFAULT_BODY_FILE', 'body.php');
define('DEFAULT_NAVIGATION_FILE', 'navigation.php');
define('DEFAULT_FOOTER_FILE', 'footer.php');
define('DEFAULT_NAVIGATION_FILE', 'navigation.php');
define('DEFAULT_FORBIDDEN_FILE', 'forbidden.php');
define('DEFAULT_ERROR_FILE', 'error.php');

//Paths:
define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../'));
define('PUBLIC_PATH', realpath(dirname(__FILE__) . '/../public/'));

//URIs:
define('URI_RAIZ', 'http://' . htmlentities($_SERVER['SERVER_NAME']) . '/usuario/');
define('URI_CSS', URI_RAIZ . '_css/');
define('URI_JS', URI_RAIZ . '_js/');
define('URI_IMAGENS', URI_RAIZ . '_img/');

define('URI_LOGIN', URI_RAIZ . 'login/');
define('URI_SMS', URI_RAIZ . 'sms/');
define('URI_REDEFINICAO', URI_RAIZ . 'redefinicao/');

define('URI_PAINEL', URI_RAIZ . 'painel/');
define('URI_CONFIGURACOES', URI_RAIZ . 'configuracoes/');
define('URI_FATURA', URI_RAIZ . 'fatura/');
define('URI_CONTATOS', URI_RAIZ . 'contatos/');

//Log:
define('LOG_VERBOSITY_NONE', 1);
define('LOG_VERBOSITY_NORMAL', 2);
define('LOG_VERBOSITY_DEBUG', 3);
define('LOG_VERBOSITY_DEFAULT', LOG_VERBOSITY_DEBUG);

//Outros:
define('DDD', 48);
define('PREFIXO', 3721);
define('REALM', 'adm.voip.ufsc.br');
define('SERVER_IP', '150.162.2.20');
define('PUSH_USERNAME', 'USUARIO');
define('PUSH_PASSWORD', 'SENHA');
```

```
define('PUSH_PASSWORD_ALT', 'SENHA_ALTERNATIVA');
define('URA_RESET_PW', '*49XXXX');
```

```
usuario/application/config/general.php
```

```
<?php
require_once(dirname(__FILE__) . '/constants.php');
require_once(dirname(__FILE__) . '/../include/functions.php');

global $config;

/* MySQL Settings */
$config['mysql_host'] = 'localhost';           //MySQL location (IP/Domain)
$config['mysql_user'] = 'USUARIO';           //MySQL user
$config['mysql_password'] = 'SENHA';         //MySQL password
$config['mysql_database'] = 'BASE';          //MySQL database

/* MySQL Midia Server Settings */
$config['mysql_midia_host'] = '150.162.2.26'; //MySQL location (IP/Domain)
$config['mysql_midia_user'] = 'USUARIO';     //MySQL user
$config['mysql_midia_password'] = 'SENHA';   //MySQL password
$config['mysql_midia_database'] = 'BASE';    //MySQL database

/* Logs Settings */
$config['log_verbosity'] = LOG_VERBOSITY_DEBUG; //Log verbosity level
$config['log_enabled'] = false;               //Log errors into DB table / E-mail
$config['log_email'] = 'contato@jonatan.com.br'; //E-mail address to send the error logs (when log
DB insert fails)

/* Polycom User Agents that supports Push messages */
$config['supported_user_agents'] = array(
    'PolycomSoundPointIP-SPIP_330',
    'PolycomSoundPointIP-SPIP_331',
    'PolycomSoundPointIP-SPIP_320',
    'PolycomSoundPointIP-SPIP_430',
    'PolycomSoundPointIP-SPIP_501',
    'PolycomSoundPointIP-SPIP_550',
    'PolycomSoundPointIP-SPIP_560',
    'PolycomSoundPointIP-SPIP_600',
    'PolycomSoundPointIP-SPIP_650',
    'PolycomSoundPointIP-SPIP_670'
);

/* Error/Exception Handling */
set_exception_handler('handle_exception'); //Define the function to handle exceptions
set_error_handler('handle_error');       //Define the function to handle errors
error_reporting(E_ALL);                   //Sets which PHP errors are reported
//error_reporting(0);                     //Sets which PHP errors are reported
ini_set('display_errors', '1');          //Turn error displaying on or off

date_default_timezone_set('America/Sao_Paulo');
?>
```

```
usuario/application/controllers/ControllerContacts.php
```

```
<?php
require_once(dirname(__FILE__) . '/../include/DB.php');
require_once(dirname(__FILE__) . '/../include/LDAP.php');
```

```

require_once(dirname(__FILE__) . '/../include/DataTables.php');
require_once(dirname(__FILE__) . '/../include/Auth.php');
require_once(dirname(__FILE__) . '/../include/FirePHPCore/fb.php');

class ControllerContacts extends DataTables {

    protected $DB;
    protected $LDAP;
    private static $instance;

    /**
     * Obtém a única instância da classe, restringindo-a somente a um objeto (Singleton)
     *
     * @see http://en.wikipedia.org/wiki/Singleton_pattern
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new self();
        }
        return self::$instance;
    }

    /**
     * Construtor privado, chamado a partir da função getInstance()
     */
    private function __construct() {
        $this->DB = DB::getInstance();
        $this->LDAP = LDAP::getInstance();
        $this->DB->connect();
    }

    /**
     * Previne cópia do objeto
     */
    private function __clone() { }

    /**
     * Pesquisa e retorna os registros de contatos do banco de dados
     *
     * @param integer $id ID do contato (opcional)
     * @param array $fields Campos (colunas) da tabela a serem retornados
     * @param integer $start Deslocamento do índice de início dos resultados, útil para paginação
     (opcional)
     * @param integer $limit Quantidade de resultados a serem retornados a partir do deslocamento
     (opcional)
     * @param integer $ramal Ramal do usuário. Utilizado para obter os contatos apenas deste ramal
     * @param integer $nome Parte do nome do usuário. Utilizado nos campos de auto completar (opcional)
     * @return array Resultado da pesquisa
     */
    public function get($id = '', $fields = array(), $start = 0, $limit = 0, $ramal = '', $nome = '') {
        $columns = '*';
        $nome = str_replace(' ', '%', $nome);
        if ($fields) {
            //Junta os campos/colunas para a consulta SQL
            $columns = implode(',', $fields);
        }
        //Prepara a consulta SQL
        $query = "SELECT $columns FROM contatos";
    }
}

```

```

//Caso o ID tenha sido especificado, retornará apenas os dados referentes a aquele ID
if ($id) {
    $query.= sprintf(" WHERE id='%s' LIMIT 1", mysql_real_escape_string($id));
}
//Senão, caso o ramal tenha sido especificado, retornará apenas os registros daquele ramal
elseif ($ramal) {
    $query.= sprintf(" WHERE ramal_usuario='%s'", mysql_real_escape_string(utf8_decode($ramal)));
    //Caso o nome tenha sido especificado, retornará apenas os autores que contém o termo em
seu nome
        if ($nome) {
            $query.= sprintf(" AND nome LIKE '%%%s%%'",
mysql_real_escape_string(utf8_decode($nome)));
        }
    }
//Senão, caso o nome tenha sido especificado, retornará apenas os autores que contém o termo em seu
nome
elseif ($nome) {
    $query.= sprintf(" WHERE nome LIKE '%%%s%%'", mysql_real_escape_string(utf8_decode($nome)));
}
//Verifica se os parâmetros para limite/paginação foram passados
if (!$id && is_numeric($start) && $start >= 0 && is_numeric($limit) && $limit > 0) {
    $query.= sprintf(" LIMIT %u,%u",
        mysql_real_escape_string($start),
        mysql_real_escape_string($limit));
}
try {
    $result_sql = $this->DB->query($query);
}
catch (Exception $e) {
    Logger::log($e->getMessage() . " (Query: $query)", __FILE__);
    return false;
}
//Retorna os resultados em forma de matriz
return $this->DB->parse_result($result_sql);
}

/**
 * Pesquisa e retorna os registros de ramais (SIP E PBX) do banco de dados
 *
 * @param integer $cn Ramal (opcional)
 * @param array $fields Campos (colunas) da tabela a serem retornados
 * @param integer $start Deslocamento do índice de início dos resultados, útil para paginação
(opcional)
 * @param integer $limit Quantidade de resultados a serem retornados a partir do deslocamento
(opcional)
 * @param integer $term Termo de busca. Utilizado nos campos de auto completar (opcional)
 * @return array Resultado da pesquisa
 */
public function get_ramais($id = '', $fields = array(), $start = 0, $limit = null, $term = '') {
    $ramais_sip = $this->get_ramais_sip($id, $fields, $start, $limit, $term);
    $ramais_pbx = $this->get_ramais_pbx($id, $fields, $start, $limit, $term);
    return array_merge($ramais_sip, $ramais_pbx);
}

/**
 * Pesquisa e retorna os registros de ramais do banco de dados
 *

```



```

* @param integer $cn Ramal (opcional)
* @param array $fields Campos (colunas) da tabela a serem retornados
* @param integer $start Deslocamento do índice de início dos resultados, útil para paginação
(opcional)
* @param integer $limit Quantidade de resultados a serem retornados a partir do deslocamento
(opcional)
* @param integer $term Termo de busca. Utilizado nos campos de auto completar (opcional)
* @return array Resultado da pesquisa
*/
public function get_ramais_sip($id = '', $fields = array(), $start = 0, $limit = null, $term = '') {

    $dn = 'ou=sip';

    //Caso o ID tenha sido especificado, retornará apenas os dados referentes a aquele ID
    if ($id) {
        $filter = "(&(cn=$id)(yellowPages=1))";
    }
    //Senão, caso o contact tenha sido especificado, retornará apenas os registros que contém o termo
no campo
    elseif ($term) {
        $filter = "(&(|(cn=$term*)(contact=$term*)(mail=$term*)))(yellowPages=1))";
    }
    try {
        $response = $this->LDAP->search($dn, $filter, $fields, false);
        $response = $this->LDAP->sort($response, $fields[0], null, null, $limit, $start);
    }
    catch (Exception $e) {
        return false;
    }
    //Retorna os resultados em forma de matriz
    return $this->LDAP->parse_result($response);
}

/**
 * Pesquisa e retorna os registros de ramais do banco de dados
 */
* @param integer $cn Ramal (opcional)
* @param array $fields Campos (colunas) da tabela a serem retornados
* @param integer $start Deslocamento do índice de início dos resultados, útil para paginação
(opcional)
* @param integer $limit Quantidade de resultados a serem retornados a partir do deslocamento
(opcional)
* @param integer $term Termo de busca. Utilizado nos campos de auto completar (opcional)
* @return array Resultado da pesquisa
*/
public function get_ramais_pbx($id = '', $fields = array(), $start = 0, $limit = null, $term = '') {

    $dn = 'ou=ramaisConvencionais';

    //Caso o ID tenha sido especificado, retornará apenas os dados referentes a aquele ID
    if ($id) {
        $filter = "(&(cn=$id)(yellowPages=1))";
    }
    //Senão, caso o contact tenha sido especificado, retornará apenas os registros que contém o termo
no campo
    elseif ($term) {
        $filter = "(&(|(cn=$term*)(contact=$term*)(mail=$term*)))(yellowPages=1))";
    }
}

```

```

    try {
        $response = $this->LDAP->search($dn, $filter, $fields, false);
        $response = $this->LDAP->sort($response, $fields[0], null, null, $limit, $start);
    }
    catch (Exception $e) {
        return false;
    }
    //Retorna os resultados em forma de matriz
    return $this->LDAP->parse_result($response);
}

/**
 * Adiciona o contato no banco de dados
 *
 * @return string Resultado da inserção no formato JSON
 */
public function add($nome, $email, $telefone) {

    if (!Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }

    //Validação
    $invalid_fields = $this->validate_parameters($nome, $email, $telefone);
    if ($invalid_fields) {
        return json_encode(array('error' => $invalid_fields));
    }

    //Inserção no Banco de Dados (exemplo: $fields['nome_da_coluna'] = 'valor;')
    $fields = array();
    $fields['ramal_usuario'] = $_SESSION['ramal'];
    $fields['nome'] = $nome;
    $fields['email'] = $email;
    $fields['telefone'] = $telefone;

    try {
        $this->DB->insert('contatos', $fields, false);
    }
    catch (Exception $e) {
        return json_encode(array('error' => 'Erro ao inserir no banco de dados'));
    }

    //Nenhum erro ocorreu. Retorna nulo.
    return json_encode(array('error' => null));
}

/**
 * Atualiza o contato no banco de dados
 *
 * @return string Resultado da atualização no formato JSON
 */
public function update($id, $nome, $email, $telefone) {

    if (!Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }

    //Validação

```

```

    if (!$id) {
        return json_encode(array('error' => 'ID não especificado'));
    }
    $invalid_fields = $this->validate_parameters($nome, $email, $telefone, $id);
    if ($invalid_fields) {
        return json_encode(array('error' => $invalid_fields));
    }

    //Atualização no Banco de Dados (exemplo: $fields['nome_da_coluna'] = 'valor');
    $fields = array();
    $fields['nome'] = $nome;
    $fields['email'] = $email;
    $fields['telefone'] = $telefone;

    $where = "id = '" . mysql_real_escape_string($id) . "'";

    try {
        $this->DB->update('contatos', $fields, $where);
    }
    catch (Exception $e) {
        return json_encode(array('error' => 'Erro ao atualizar no banco de dados'));
    }

    return json_encode(array('error' => null));
}

/**
 * Remove os registros do banco de dados
 *
 * @param array $ids IDs dos registros a serem excluídos
 * @return string Resultado no formato JSON
 */
public function del($ids) {

    if (!$Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }
    if (!is_array($ids) || !$ids) {
        return json_encode(array('error' => 'IDs inválidos'));
    }

    //Exclui do banco de dados
    $ids_str = implode(",", $ids);
    $query = sprintf("DELETE FROM contatos
                     WHERE id IN ('%s'",
                               $ids_str);

    try {
        $this->DB->query($query);
    }
    catch (Exception $e) {
        return json_encode(array('error' => 'Erro ao excluir no banco de dados'));
    }

    //Nenhum erro ocorreu. Retorna nulo.
    return json_encode(array('error' => null));
}

protected function fnBuildWhereClause($aColumns, $sSearch) {

```

```

if (!Auth::check()) {
    return json_encode(array('error' => 'Acesso negado'));
}
$where = 'WHERE ramal_usuario=' . $_SESSION['ramal'];
if (is_array($columns) && $search != '') {
    $search = preg_replace('/\s\s+/', ' ', $search); //Remove extra spaces
    $search = str_replace(' ', '%', $search); //Replace space with %
    $where = ' AND ';
    array_shift($columns);
    foreach ($columns as $column) {
        $where .= $column . " LIKE '%" . mysql_real_escape_string(utf8_decode($search)) . "%' OR
";
    }
    $where = substr($where, 0, -4);
}
return $where;
}

/**
 * Valida os campos (parâmetros) do contato
 *
 * @return array Campos que não passaram no teste de validação
 */
private function validate_parameters($nome, $email, $telefone, $id = null) {
    $invalid_fields = array();
    if (!$nome) {
        $invalid_fields['nome'] = 'O nome não pode ser vazio';
    }
    if ($email && !validate_email($email)) {
        $invalid_fields['email'] = 'E-mail inválido';
    }
    if (!$telefone) {
        $invalid_fields['telefone'] = 'Informe o telefone';
    }
    else {
        if ($id) {
            $query = sprintf("SELECT COUNT(*) AS total
                FROM contatos
                WHERE telefone='%s'
                AND ramal_usuario='%s'
                AND id != %d",
                mysql_real_escape_string($telefone),
                mysql_real_escape_string($_SESSION['ramal']),
                mysql_real_escape_string($id));
        }
        else {
            $query = sprintf("SELECT COUNT(*) AS total
                FROM contatos
                WHERE telefone='%s'
                AND ramal_usuario='%s'",
                mysql_real_escape_string($telefone),
                mysql_real_escape_string($_SESSION['ramal']));
        }
        $result = $this->DB->query($query);
        if (mysql_result($result, 0, 'total') > 0) {
            $invalid_fields['telefone'] = 'Telefone já está na lista';
        }
    }
}

```

```

        return $invalid_fields;
    }
}

```

usuario/application/controllers/ControllerSMS.php

```

<?php
require_once(dirname(__FILE__) . '/../include/DB.php');
require_once(dirname(__FILE__) . '/../include/Auth.php');
require_once(dirname(__FILE__) . '/../include/DataTables.php');
require_once(dirname(__FILE__) . '/../include/FirePHPCore/fb.php');

class ControllerSMS extends DataTables {

    protected $DB;
    protected $curl;
    protected $sent_messages;
    private static $instance;

    /**
     * Obtém a única instância da classe, restringindo-a somente a um objeto (Singleton)
     *
     * @see http://en.wikipedia.org/wiki/Singleton_pattern
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new self();
        }
        return self::$instance;
    }

    /**
     * Construtor privado, chamado a partir da função getInstance()
     */
    private function __construct() {
        $this->DB = DB::getInstance();
        $this->DB->connect();
    }

    /**
     * Previne cópia do objeto
     */
    private function __clone() { }

    /**
     * Pesquisa e retorna os registros de mensagens do banco de dados
     *
     * @param integer $id ID da mensagem (opcional)
     * @param array $fields Campos (colunas) da tabela a serem retornados
     * @param integer $start Deslocamento do índice de início dos resultados, útil para paginação
     (opcional)
     * @param integer $limit Quantidade de resultados a serem retornados a partir do deslocamento
     (opcional)
     * @return array Resultado da pesquisa
     */
    public function get($id = '', $fields = array(), $start = 0, $limit = 0, $ramal = '') {

        if (!Auth::check()) {

```

```

        return false;
    }

    $columns = '*';
    if ($fields) {
        //Junta os campos/colunas para a consulta SQL
        $columns = implode('`,`', $fields);
    }

    //Prepara a consulta SQL
    $query = "SELECT `{$columns}` FROM messages";

    //Caso o ID tenha sido especificado, retornará apenas os dados referentes a aquele ID
    if ($id) {
        $query.= sprintf(" WHERE id='%s' LIMIT 1", mysql_real_escape_string($id));
    }
    elseif ($ramal) {
        $query.= sprintf(" WHERE `to`='%s'", mysql_real_escape_string($ramal));
    }
    //Verifica se os parâmetros para limite/paginação foram passados
    elseif (!($id && is_numeric($start) && $start >= 0 && is_numeric($limit) && $limit > 0) {
        $query.= sprintf(" LIMIT %u,%u",
            mysql_real_escape_string($start),
            mysql_real_escape_string($limit));
    }

    try {
        $result_sql = $this->DB->query($query);
    }
    catch (Exception $e) {
        //
        Logger::log($e->getMessage() . " (Query: $query)", __FILE__);
        return false;
    }
    //Retorna os resultados em forma de matriz
    return $this->DB->parse_result($result_sql);
}

public function send($ramal, $mensagem) {

    global $config;

    if (!Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }

    if (!$ramal) {
        return json_encode(array('error' => 'Ramal não especificado'));
    }

    if (!$mensagem) {
        return json_encode(array('error' => 'Mensagem não especificada'));
    }

    $query = sprintf('SELECT contact, user_agent
        FROM location
        WHERE username = %d',
        mysql_real_escape_string($ramal));

```

```

$response = $this->DB->query($query);

if (mysql_num_rows($response) == 0) {
    return json_encode(array('error' => 'Ramal não encontrado'));
}

$this->sent_messages = 0;

//Inserção no Banco de Dados (exemplo: $fields['nome_da_coluna'] = 'valor');
$fields = array();
$fields['from'] = $_SESSION['ramal'];
$fields['to'] = $ramal;
$fields['content'] = $mensagem;

try {
    $this->DB->insert('messages', $fields, false);
}
catch (Exception $e) {
    return json_encode(array('error' => 'Erro ao inserir no banco de dados'));
}

$mensagem = nl2br("<html><title>Mensagem de: $ramal</title><body>$mensagem</body></html>");
$mensagem = sprintf('<PolycomIPPhone><Data priority="critical">%s</Data></PolycomIPPhone>',
$mensagem);

$this->curl = curl_init();

curl_setopt($this->curl, CURLOPT_POST, 1);
curl_setopt($this->curl, CURLOPT_FOLLOWLOCATION, 0);
curl_setopt($this->curl, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($this->curl, CURLOPT_POSTFIELDS, $mensagem);
curl_setopt($this->curl, CURLOPT_HTTPAUTH, CURLAUTH_DIGEST);
curl_setopt($this->curl, CURLOPT_USERPWD, PUSH_USERNAME . ':' . PUSH_PASSWORD);

while ($row = mysql_fetch_array($response)) {
    $contact = $row['contact'];
    $contact = explode('@', $contact);
    $contact = explode(':', $contact[1]);
    $contact = explode(':', $contact[0]);
    $phone_ip = $contact[0];
    $user_agent = $row['user_agent'];

    $supported = false;
    foreach ($config['supported_user_agents'] as $supported_user_agent) {
        if (strstr($user_agent, $supported_user_agent) !== false) {
            $supported = true;
            break;
        }
    }

    if ($supported && $phone_ip) {
        $url = "http://$phone_ip/push";
        curl_setopt($this->curl, CURLOPT_URL, $url);
        $this->send_push();
    }
}

curl_close($this->curl);

```

```

        return json_encode(array('error' => null, 'sent' => $this->sent_messages));
    }

/**
 * Remove os registros do banco de dados
 *
 * @param array $sids IDs dos registros a serem excluidos
 * @return string Resultado no formato JSON
 */
public function del($sids) {

    if (!Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }
    if (!is_array($sids) || !$sids) {
        return json_encode(array('error' => 'IDs inválidos'));
    }

    //Exclui do banco de dados
    $sids_str = implode(",", $sids);
    $query = sprintf("DELETE FROM messages
                     WHERE id IN ('%s')",
                     $sids_str);

    try {
        $this->DB->query($query);
    }
    catch (Exception $e) {
        // Logger::log($e->getMessage() . " (Query: $query)", __FILE__);
        return json_encode(array('error' => 'Erro ao excluir no banco de dados'));
    }

    //Nenhum erro ocorreu. Retorna nulo.
    return json_encode(array('error' => null));
}

private function send_push($retry = true) {
    $result = curl_exec($this->curl);
    if (!curl_errno($this->curl)) {
        $info = curl_getinfo($this->curl);
        if ($info['http_code'] == 401 && $retry) {
            curl_setopt($this->curl, CURLOPT_USERPWD, PUSH_USERNAME . ':' . PUSH_PASSWORD_ALT);
            $this->send_push(false);
        }
        if ($info['http_code'] == 200) {
            $this->sent_messages++;
        }
    }
}

protected function fnBuildWhereClause($aColumns, $sSearch) {
    if (!Auth::check()) {
        return json_encode(array('error' => 'Acesso negado'));
    }
    $sWhere= 'WHERE `to`=' . $_SESSION['ramal'];
    if (is_array($aColumns) && $sSearch != '') {
        $sSearch = preg_replace('/\s\s+/', ' ', $sSearch); //Remove extra spaces
        $sSearch = str_replace(' ', '%', $sSearch); //Replace space with %
    }
}

```



```

        $sWhere = ' AND ';
        array_shift($aColumns);
        foreach ($aColumns as $aColumn) {
            $sWhere.= "`$aColumn` LIKE '%" . mysql_real_escape_string(utf8_decode($sSearch)) . "%' OR
";
        }
        $sWhere = substr($sWhere, 0, -4);
    }
    return $sWhere;
}

protected function fnParseResult($rResult) {
    $aData = array();
    $iIndex = 0;
    while ($aRow = mysql_fetch_array($rResult)) {
        $aData[$iIndex] = array();
        $sId = array_shift($aRow);
        $aData[$iIndex][] = '<input type="checkbox" name="ids[]" value="'.$sId.'" />';
        array_shift($aRow);
        $aKeys = array_keys($aRow);
        foreach ($aKeys as $sKey) {
            if (!is_numeric($sKey)) {
                if ($aRow[$sKey] === null) {
                    $aRow[$sKey] = '';
                }
                if ($sKey == 'date') {
                    $aRow[$sKey] = date('d/m/Y H:i:s', strtotime($aRow[$sKey]));
                }
                $aData[$iIndex][] = utf8_encode($aRow[$sKey]);
            }
        }
        $iIndex++;
    }
    return $aData;
}
}

```

usuario/application/include/php-sip/PhpSIP.class.php

```

<?php
/**
 * (c) 2007-2009 Chris Maciejewski
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files
 * (the "Software"), to deal in the Software without restriction,
 * including without limitation the rights to use, copy, modify, merge,
 * publish, distribute, sublicense, and/or sell copies of the Software,
 * and to permit persons to whom the Software is furnished to do so,
 * subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

```

```
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
* DEALINGS IN THE SOFTWARE.
*
*/

/**
 * PHP SIP UAC class
 *
 * @ingroup API
 * @author Chris Maciejewski <chris@wima.co.uk>
 *
 * @version SVN: $Id: PhpSIP.class.php 24 2009-11-25 11:39:57Z level7systems $
 */
require_once 'PhpSIP.Exception.php';

class PhpSIP
{
    private $debug = false;

    /**
     * Min port
     */
    private $min_port = 5065;

    /**
     * Max port
     */
    private $max_port = 5265;

    /**
     * Final Response timer (in seconds)
     */
    private $fr_timer = 20;

    /**
     * Lock file
     */
    private $lock_file;

    /**
     * Allowed methods array
     */
    private $allowed_methods = array(
        "CANCEL", "NOTIFY", "INVITE", "BYE", "REFER", "OPTIONS", "SUBSCRIBE", "MESSAGE"
    );

    /**
     * Dialog established
     */
    private $dialog = false;

    /**
     * The opened socket we listen for incoming SIP messages
     */
    private $socket;

    /**
```

```
* Source IP address
*/
private $src_ip;

/**
 * Source IP address
 */
private $user_agent = 'PHP SIP';

/**
 * CSeq
 */
private $cseq = 20;

/**
 * NC
 */
private $nc = 0;

/**
 * Source port
 */
private $src_port;

/**
 * Call ID
 */
private $call_id;

/**
 * Contact
 */
private $contact;

/**
 * Request URI
 */
private $uri;

/**
 * Request host
 */
private $host;

/**
 * Request port
 */
private $port = 5060;

/**
 * Outbound SIP proxy
 */
private $proxy;

/**
 * Method
 */
```

```
private $method;

/**
 * Auth username
 */
private $username;

/**
 * Auth password
 */
private $password;

/**
 * MD5 checksum password: md5(username:realm:password)
 */
private $password_md5;

/**
 * To
 */
private $to;

/**
 * To tag
 */
private $to_tag;

/**
 * From
 */
private $from;

/**
 * From User
 */
private $from_user;

/**
 * From tag
 */
private $from_tag;

/**
 * Via tag
 */
private $via;

/**
 * Content type
 */
private $content_type;

/**
 * Body
 */
private $body;

/**
```

```

* Received Response
*/
private $response; // whole response body
private $res_code;
private $res_contact;
private $res_cseq_method;
private $res_cseq_number;

/**
 * Received Request
 */
private $req_method;
private $req_cseq_method;
private $req_cseq_number;
private $req_contact;

/**
 * Authentication
 */
private $auth;

/**
 * Routes
 */
private $routes = array();

/**
 * Request vias
 */
private $request_via = array();

/**
 * Additional headers
 */
private $extra_headers = array();

/**
 * Constructor
 *
 * @param $src_ip Ip address to bind (optional)
 */
public function __construct($src_ip = null, $fr_timer = null)
{
    if (!function_exists('socket_create'))
    {
        throw new PhpSIPException("socket_create() function missing.");
    }

    if (!$src_ip)
    {
        // running in a web server
        if (isset($_SERVER['SERVER_ADDR']))
        {
            $src_ip = $_SERVER['SERVER_ADDR'];
        }
        // running from command line
        else
        {

```

```

        $addr = gethostbyname(gethostname());

        if (!is_array($addr) || !isset($addr[0]) || substr($addr[0],0,3) == '127')
        {
            throw new PhpSIPException("Failed to obtain IP address to bind. Please set bind address
manually.");
        }

        $src_ip = $addr[0];
    }

    if ($fr_timer && is_integer($fr_timer))
    {
        $this->fr_timer = $fr_timer;
    }

    $this->src_ip = $src_ip;

    $this->lock_file = rtrim(sys_get_temp_dir(), DIRECTORY_SEPARATOR).DIRECTORY_SEPARATOR.'phpSIP.lock';

    $this->session_file =
rtrim(sys_get_temp_dir(), DIRECTORY_SEPARATOR).DIRECTORY_SEPARATOR.'phpSIP.session';

    $this->createSocket();
}

/**
 * Destructor
 */
public function __destruct()
{
    $this->closeSocket();
    @unlink($this->session_file);
}

/**
 * Sets debuggin ON/OFF
 *
 * @param bool $status
 */
public function setDebug($status = false)
{
    $this->debug = $status;
}

/**
 * Gets src IP
 *
 * @return string
 */
public function getSrcIp()
{
    return $this->src_ip;
}

/**
 * Gets port number to bind

```

```

*/
private function getPort()
{
    if ($this->min_port > $this->max_port)
    {
        throw new PhpSIPException ("Min port is bigger than max port.");
    }

    // waiting until file will be locked for writing
    // (1000 milliseconds as timeout)
    $fp = @fopen($this->lock_file, 'a+b');

    if (!$fp)
    {
        throw new PhpSIPException ("Failed to open lock file " . $this->lock_file);
    }

    $startTime = microtime();

    do
    {
        $scanWrite = flock($fp, LOCK_EX);
        // If lock not obtained sleep for 0 - 100 milliseconds,
        // to avoid collision and CPU load
        if (!$scanWrite) usleep(round(rand(0, 100)*1000));
    } while ((!$scanWrite)and((microtime()-$startTime) < 1000));

    if (!$scanWrite)
    {
        throw new PhpSIPException ("Failed to lock a file in 1000 ms.");
    }

    //file was locked
    $size = filesize($this->lock_file);
    if ($size)
    {
        $contents = fread($fp, $size);
        $pids = explode(",",$contents);
    }
    else
    {
        $pids = false;
    }

    ftruncate($fp, 0);

    // we are the first one to run, initialize "PID" => "port number" array
    if (!$pids)
    {
        if (!fwrite($fp, $this->min_port))
        {
            throw new PhpSIPException("Fail to write data to a lock file.");
        }

        $this->src_port = $this->min_port;
    }

    // there are other programs running now

```

```

else
{
    // check if there are any empty ports left
    if (count($pids) >= ($this->max_port - $this->min_port))
    {
        throw new PhpSIPException("No more ports left to bind.");
    }

    asort($pids, SORT_NUMERIC);

    $prev = current($pids);

    if ($prev > $this->min_port)
    {
        $src_port = $this->min_port;
    }
    else
    {
        foreach ($pids as $port)
        {
            if (($port - $prev) > 1)
            {
                $src_port = $prev + 1;
                break;
            }

            $prev = $port;
        }

        if (($prev + 1) >= $this->max_port)
        {
            throw new PhpSIPException("No more ports left to bind. We shouldn't be here!");
        }

        $src_port = $prev + 1;
    }

    if (in_array($src_port, $pids))
    {
        throw new PhpSIPException("Fail to obtain free port number.");
    }

    $pids[] = $src_port;

    if (!fwrite($fp, implode(", ", $pids)))
    {
        throw new PhpSIPException("Failed to write data to lock file.");
    }

    if ($src_port < $this->min_port || $src_port > $this->max_port)
    {
        $src_port = rand($this->min_port, $this->max_port);
    }

    $this->src_port = $src_port;
}

if (!fclose($fp))

```



```

    {
        throw new PhpSIPException("Failed to close lock_file");
    }
}

/**
 * Releases port
 */
public function releasePort()
{
    // waiting until file will be locked for writing
    // (1000 milliseconds as timeout)
    $fp = fopen($this->lock_file, 'r+b');

    if (!$fp)
    {
        throw new PhpSIPException("Can't open lock file.");
    }

    $startTime = microtime();

    do
    {
        $scanWrite = flock($fp, LOCK_EX);
        // If lock not obtained sleep for 0 - 100 milliseconds,
        // to avoid collision and CPU load
        if (!$scanWrite) usleep(round(rand(0, 100)*1000));
    } while ((!$scanWrite)and((microtime()-$startTime) < 1000));

    if (!$scanWrite)
    {
        throw new PhpSIPException("Failed to lock a file in 1000 ms.");
    }

    clearstatcache();

    $size = filesize($this->lock_file);
    $content = fread($fp,$size);

    //file was locked
    $pids = explode(",", $content);

    $key = array_search($this->src_port, $pids);

    unset($pids[$key]);

    if (count($pids) === 0)
    {
        if (!fclose($fp))
        {
            throw new PhpSIPException("Failed to close lock_file");
        }

        if (!unlink($this->lock_file))
        {
            throw new PhpSIPException("Failed to delete lock_file.");
        }
    }
}

```

```

    }
}
else
{
    ftruncate($fp, 0);

    if (!fwrite($fp, implode("",$pids)))
    {
        throw new PhpSIPException("Failed to save data in lock_file");
    }

    if (!fclose($fp))
    {
        throw new PhpSIPException("Failed to close lock_file");
    }
}
}

/**
 * Adds additional header
 *
 * @param string $header
 */
public function addHeader($header)
{
    $this->extra_headers[] = $header;
}

/**
 * Sets From header
 *
 * @param string $from
 */
public function setFrom($from)
{
    if (preg_match('/<.*>/', $from))
    {
        $this->from = $from;
    }
    else
    {
        $this->from = '<'.$from.'>';
    }

    $m = array();
    if (!preg_match('/sip:(.*)@/i', $this->from, $m))
    {
        throw new PhpSIPException('Failed to parse From username.');
```

```

public function setMethod($method)
{
    if (!in_array($method,$this->allowed_methods))
    {
        throw new PhpSIPException('Invalid method.');
```

```
    }

    $this->method = $method;
```

```
    if ($method == 'INVITE')
    {
        $body = "v=0\r\n";
        $body.= "o=4187 0 0 IN IP4 ".$this->src_ip."\r\n";
        $body.= "s=click2dial call\r\n";
        $body.= "c=IN IP4 ".$this->src_ip."\r\n";
        $body.= "t=0 0\r\n";
        $body.= "m=audio 8000 RTP/AVP 0 18 97 98\r\n";
        $body.= "a=rtpmap:0 PCMU/8000\r\n";
        $body.= "a=rtpmap:18 G729/8000\r\n";
        $body.= "a=rtpmap:97 ilbc/8000\r\n";
        $body.= "a=rtpmap:98 speex/8000\r\n";
```

```
        $this->body = $body;
```

```
        $this->setContentType (null);
```

```
    }
```

```
    if ($method == 'REFER')
```

```
    {
        $this->setBody('');
```

```
    if ($method == 'CANCEL')
```

```
    {
        $this->setBody('');
        $this->setContentType (null);
    }
```

```
    if ($method == 'MESSAGE')
```

```
    {
        $this->setContentType (null);
    }
```

```
    }
```

```
/**
```

```
 * Sets SIP Proxy
```

```
 *
```

```
 * @param $proxy
```

```
 */
```

```
public function setProxy($proxy)
```

```
{
```

```
    $this->proxy = $proxy;
```

```
}
```

```
/**
```

```
 * Sets request URI
```

```
 *
```

```
 * @param string $uri
```

```

*/
public function setUri($uri)
{
    if (strpos($uri, 'sip:') === false)
    {
        throw new PhpSIPException("Only sip: URI supported.");
    }

    $this->uri = $uri;
    $this->to = '<'.$uri.>';

    if ($this->proxy)
    {
        if (strpos($this->proxy, ':'))
        {
            $temp = explode(":", $this->proxy);

            $this->host = $temp[0];
            $this->port = $temp[1];
        }
        else
        {
            $this->host = $this->proxy;
        }
    }
    else
    {
        $url = str_replace("sip:", "sip://", $uri);

        if (!$url = @parse_url($url))
        {
            throw new PhpSIPException("Failed to parse URI.");
        }

        $this->host = $url['host'];

        if (isset($url['port']))
        {
            $this->port = $url['port'];
        }
    }
}

/**
 * Sets username
 *
 * @param string $username
 */
public function setUsername($username)
{
    $this->username = $username;
    $this->session_file.= '.' . $username;
    $this->restoreSession();
}

/**
 * Sets User Agent
 *

```

```
* @param string $user_agent
*/
public function setUserAgent($user_agent)
{
    $this->user_agent = $user_agent;
}

/**
 * Sets password
 *
 * @param string $password
 */
public function setPassword($password)
{
    $this->password = $password;
}

/**
 * Sets md5 password
 *
 * @param string $password
 */
public function setPasswordMd5($password)
{
    $this->password_md5 = $password;
}

/**
 * Sends SIP request
 *
 * @return string Reply
 */
public function send()
{
    if (!$this->from)
    {
        throw new PhpSIPException('Missing From.');
```

```

    $_SESSION['via'] = $this->getVia();
    $_SESSION['from_tag'] = $this->getFromTag();
    $_SESSION['call_id'] = $this->getCallId();
    $_SESSION['cseq'] = $this->getCSeq();
    $this->saveSession();
}

$this->sendData($data);

$this->readResponse();

if ($this->method == 'CANCEL' && $this->res_code == '200')
{
    $i = 0;
    while (substr($this->res_code,0,1) != '4' && $i < 2)
    {
        $this->readResponse();
        $i++;
    }
}

if (substr($this->res_code,0,1) == '1')
{
    $i = 0;
    while (substr($this->res_code,0,1) == '1' && $i < 4)
    {
        $this->readResponse();
        $i++;
    }
}

if ($this->res_code == '407')
{
    $this->cseq++;

    $this->auth();

    $data = $this->formatRequest();

    $this->sendData($data);

    $this->readResponse();
}

if ($this->res_code == '401')
{
    $this->cseq++;

    $this->authWWW();

    if ($this->method == 'INVITE')
    {
        //$SESSION['response'] = $this->response;
        $_SESSION['nc'] = $this->nc;
        $this->saveSession();
    }

    $data = $this->formatRequest();
}

```

```

        $this->sendData($data);

        $this->readResponse();
    }

    if (substr($this->res_code,0,1) == '1')
    {
        $i = 0;
        while (substr($this->res_code,0,1) == '1' && $i < 4)
        {
            $this->readResponse();
            $i++;
        }
    }

    $this->extra_headers = array();
    $this->cseq++;

    return $this->res_code;
}

/**
 * Sends data
 */
private function sendData($data)
{
    usleep(10000);

    if (!@socket_sendto($this->socket, $data, strlen($data), 0, $this->host, $this->port))
    {
        $err_no = socket_last_error($this->socket);
        throw new PhpSIPException("Failed to send data. ".socket_strerror($err_no));
    }

    if ($this->debug)
    {
        $temp = explode("\r\n", $data);

        echo "--> ".$temp[0]. " (" .date('h:i:s').")\n";
    }
}

/**
 * Listen for request
 *
 * @todo This needs to be improved
 */
public function listen($method)
{
    $i = 0;
    while ($this->req_method != $method)
    {
        $this->readResponse();

        $i++;

        if ($i > 5)

```

```

    {
        throw new PhpSIPException("Unexpected request ".$this->req_method."received.");
    }
}

/**
 * Reads response
 */
private function readResponse()
{
    $from = "";
    $port = 0;

    if (!@socket_recvfrom($this->socket, $this->response, 10000, 0, $from, $port))
    {
        $this->res_code = "No final response in $this->fr_timer seconds.";
        return $this->res_code;
    }

    if ($this->debug)
    {
        $temp = explode("\r\n", $this->response);

        echo "<-- ".$temp[0]. " (" .date('h:i:s').")\n";
    }

    // Response
    $result = array();
    if (preg_match('/^SIP\/2\.0 ([0-9]{3})/', $this->response, $result))
    {
        $this->res_code = trim($result[1]);

        $res_class = substr($this->res_code, 0, 1);
        if ($res_class == '1' || $res_class == '2')
        {
            $this->dialog = true;
        }

        $this->parseResponse();
    }
    // Request
    else
    {
        $this->parseRequest();
    }
}

/**
 * Parse Response
 */
private function parseResponse()
{
    // To tag
    $result = array();
    if (preg_match('/^To: .*;tag=(.*)$/im', $this->response, $result))
    {
        $this->to_tag = trim($result[1]);
    }
}

```



```

}

// Route
$result = array();
if (preg_match_all('/^Record-Route: (.*)$/im',$this->response,$result))
{
    foreach ($result[1] as $route)
    {
        if (!in_array(trim($route),$this->routes))
        {
            $this->routes[] = trim($route);
        }
    }
}

// Request via
$result = array();
$this->request_via = array();
if (preg_match_all('/^Via: (.*)$/im',$this->response,$result))
{
    foreach ($result[1] as $via)
    {
        $this->request_via[] = trim($via);
    }
}

// Response contact
$result = array();
if (preg_match('/^Contact:.*<(.)>/im',$this->response,$result))
{
    $this->res_contact = trim($result[1]);

    $semicolon = strpos($this->res_contact,";");

    if ($semicolon !== false)
    {
        $this->res_contact = substr($this->res_contact,0,$semicolon);
    }
}

// Response CSeq method
$result = array();
if (preg_match('/^CSeq: [0-9]+ (.*)$/im',$this->response,$result))
{
    $this->res_cseq_method = trim($result[1]);
}

// ACK 2XX-6XX - only invites - RFC3261 17.1.2.1
if ($this->res_cseq_method == 'INVITE' && in_array(substr($this->res_code,0,1),array('2','3','4','5','6')))
{
    $this->ack();
}

return $this->res_code;
}

/**

```

```

* Parse Request
*/
private function parseRequest()
{
    $temp = explode("\r\n",$this->response);
    $temp = explode(" ",$temp[0]);
    $this->req_method = trim($temp[0]);

    // Route
    $result = array();
    if (preg_match_all('/^Record-Route: (.*)$/im',$this->response,$result))
    {
        foreach ($result[1] as $route)
        {
            if (!in_array(trim($route),$this->routes))
            {
                $this->routes[] = trim($route);
            }
        }
    }

    // Request via
    $result = array();
    $this->request_via = array();
    if (preg_match_all('/^Via: (.*)$/im',$this->response,$result))
    {
        foreach ($result[1] as $via)
        {
            $this->request_via[] = trim($via);
        }
    }

    // Method contact
    $result = array();
    if (preg_match('/^Contact: <(.*)>/im',$this->response,$result))
    {
        $this->req_contact = trim($result[1]);

        $semicolon = strpos($this->res_contact,";");

        if ($semicolon !== false)
        {
            $this->res_contact = substr($this->res_contact,0,$semicolon);
        }
    }

    // Response CSeq method
    if (preg_match('/^CSeq: [0-9]+ (.*)$/im',$this->response,$result))
    {
        $this->req_cseq_method = trim($result[1]);
    }

    // Response CSeq number
    if (preg_match('/^CSeq: ([0-9]+) .*/im',$this->response,$result))
    {
        $this->req_cseq_number = trim($result[1]);
    }
}

```

```

/**
 * Send Response
 *
 * @param int $code    Response code
 * @param string $text  Response text
 */
public function reply($code,$text)
{
    $r = 'SIP/2.0 ' . $code . ' ' . $text . "\r\n";
    // Via
    foreach ($this->request_via as $via)
    {
        $r.= 'Via: ' . $via . "\r\n";
    }
    // From
    $r.= 'From: ' . $this->from . ';tag=' . $this->to_tag . "\r\n";
    // To
    $r.= 'To: ' . $this->to . ';tag=' . $this->from_tag . "\r\n";
    // Call-ID
    $r.= 'Call-ID: ' . $this->call_id . "\r\n";
    //CSeq
    $r.= 'CSeq: ' . $this->req_cseq_number . ' ' . $this->req_cseq_method . "\r\n";
    // Max-Forwards
    $r.= 'Max-Forwards: 70' . "\r\n";
    // User-Agent
    $r.= 'User-Agent: ' . $this->user_agent . "\r\n";
    // Content-Length
    $r.= 'Content-Length: 0' . "\r\n";
    $r.= "\r\n";

    $this->sendData($r);
}

/**
 * ACK
 */
private function ack()
{
    if ($this->res_cseq_method == 'INVITE' && $this->res_code == '200')
    {
        $a = 'ACK ' . $this->res_contact . ' SIP/2.0' . "\r\n";
    }
    else
    {
        $a = 'ACK ' . $this->uri . ' SIP/2.0' . "\r\n";
    }
    // Via
    $a.= 'Via: ' . $this->via . "\r\n";
    // Route
    if ($this->routes)
    {
        foreach ($this->routes as $route)
        {
            $a.= 'Route: ' . $route . "\r\n";
        }
    }
    // From

```

```

if (!$this->from_tag) $this->setFromTag();
$a.= 'From: '.$this->from.';tag='.$this->from_tag."\r\n";
// To
if ($this->to_tag)
    $a.= 'To: '.$this->to.';tag='.$this->to_tag."\r\n";
else
    $a.= 'To: '.$this->to."\r\n";
// Call-ID
if (!$this->call_id) $this->setCallId();
$a.= 'Call-ID: '.$this->call_id."\r\n";
//CSeq
$a.= 'CSeq: '.$this->cseq.' ACK'."\r\n";
// Authentication
if ($this->res_code == '200' && $this->auth)
{
    $a.= 'Proxy-Authorization: '.$this->auth."\r\n";
}
// Max-Forwards
$a.= 'Max-Forwards: 70'."\r\n";
// User-Agent
$a.= 'User-Agent: '.$this->user_agent."\r\n";
// Content-Length
$a.= 'Content-Length: 0'."\r\n";
$a.= "\r\n";

$this->sendData($a);
}

/**
 * Formats SIP request
 *
 * @return string
 */
private function formatRequest()
{
    if (in_array($this->method,array('BYE','REFER','SUBSCRIBE')))
    {
        $r = $this->method.' '.$this->res_contact.' SIP/2.0'."\r\n";
    }
    else
    {
        $r = $this->method.' '.$this->uri.' SIP/2.0'."\r\n";
    }
    // Via
    if ($this->method != 'CANCEL')
    {
        $this->setVia();
    }
    $r.= 'Via: '.$this->via."\r\n";
    // Route
    if ($this->method != 'CANCEL' && $this->routes)
    {
        foreach ($this->routes as $route)
        {
            $r.= 'Route: '.$route."\r\n";
        }
    }
    // From

```

```

if (!$this->from_tag) $this->setFromTag();
$r.= 'From: '.$this->from.';tag='.$this->from_tag."\r\n";
// To
if (!in_array($this->method,array("INVITE","CANCEL","NOTIFY")) && $this->to_tag)
    $r.= 'To: '.$this->to.';tag='.$this->to_tag."\r\n";
else
    $r.= 'To: '.$this->to."\r\n";
// Authentication
if ($this->auth)
{
    $r.= $this->auth."\r\n";
    $this->auth = null;
}
// Call-ID
if (!$this->call_id) $this->setCallId();
$r.= 'Call-ID: '.$this->call_id."\r\n";
//CSeq
if ($this->method == 'CANCEL')
{
    $this->cseq--;
}
$r.= 'CSeq: '.$this->cseq.' '.$this->method."\r\n";
// Contact
if ($this->method != 'MESSAGE')
{
    $r.= 'Contact: <sip:'.$this->from_user.'@'.$this->src_ip.':'.$this->src_port.>'. "\r\n";
}
// Allow
if ($this->method == 'INVITE')
{
    $allow = implode(' ', $this->allowed_methods);
    $r.= 'Allow: '.$allow."\r\n";
}
// Content-Type
if ($this->content_type)
{
    $r.= 'Content-Type: '.$this->content_type."\r\n";
}
// Max-Forwards
$r.= 'Max-Forwards: 70'. "\r\n";
// User-Agent
$r.= 'User-Agent: '.$this->user_agent."\r\n";
// Additional header
foreach ($this->extra_headers as $header)
{
    $r.= $header."\r\n";
}
// Content-Length
$r.= 'Content-Length: '.strlen($this->body). "\r\n";
$r.= "\r\n";
$r.= $this->body;

return $r;
}

/**
 * Sets body
 */

```

```

public function setBody($body)
{
    $this->body = $body;
}

/**
 * Sets Content Type
 */
public function setContentType($content_type = null)
{
    if ($content_type !== null)
    {
        $this->content_type = $content_type;
    }
    else
    {
        switch ($this->method)
        {
            case 'INVITE':
                $this->content_type = 'application/sdp';
                break;
            case 'MESSAGE':
                $this->content_type = 'text/html; charset=utf-8';
                break;
            default:
                $this->content_type = null;
        }
    }
}

/**
 * Sets Via header
 */
public function setVia($via = null)
{
    if ($via)
    {
        $this->via = $via;
    }
    else
    {
        $rand = rand(100000,999999);
        $this->via = 'SIP/2.0/UDP '.$this->src_ip.':'.$this->src_port.';rport;branch=z9hG4bK'.$rand;
    }
}

/**
 * Gets Via header
 */
public function getVia()
{
    return $this->via;
}

/**
 * Sets from tag
 */
public function setFromTag($tag = null)

```

```
{
    if ($tag)
    {
        $this->from_tag = $tag;
    }
    else
    {
        $this->from_tag = rand(10000,99999);
    }
}

/**
 * Gets from tag
 */
public function getFromTag()
{
    return $this->from_tag;
}

/**
 * Sets call id
 */
public function setCallId($id = null)
{
    if ($id)
    {
        $this->call_id = $id;
    }
    else
    {
        $this->call_id = md5(uniqid()).'@'.$this->src_ip;
    }
}

/**
 * Gets call sequence number
 */
public function getCSeq()
{
    return $this->cseq;
}

/**
 * Sets call sequence number
 */
public function setCSeq($cseq)
{
    $this->cseq = $cseq;
}

/**
 * Gets call id
 */
public function getCallId()
{
    return $this->call_id;
}
```

```

/**
 * Sets response
 *
 public function setResponse($response)
 {
     $this->response = $response;
 }
 */

/**
 * Sets NC
 */
public function setNC($nc)
{
    $this->nc = $nc;
}

/**
 * Gets value of the header from the previous request
 *
 * @param string $name Header name
 *
 * @return string or false
 */
public function getHeader($name)
{
    if (preg_match('/^\.$name.': (.*)$/m', $this->response, $result))
    {
        return trim($result[1]);
    }
    else
    {
        return false;
    }
}

/**
 * Calculates Digest authentication response
 */
private function auth()
{
    if (!$this->username)
    {
        throw new PhpSIPException("Missing username");
    }

    if (!$this->password && !$this->password_md5)
    {
        throw new PhpSIPException("Missing password");
    }

    // realm
    $result = array();
    if (!preg_match('/^Proxy-Authenticate: .* realm="(.)"/imU', $this->response, $result))
    {
        throw new PhpSIPException("Can't find realm in proxy-auth");
    }
}

```



```

}

$realm = $result[1];

// nonce
$result = array();
if (!preg_match('/^Proxy-Authenticate: .* nonce="(.)"/imU',$this->response, $result))
{
    throw new PhpSIPException("Can't find nonce in proxy-auth");
}

$nonce = $result[1];

if ($this->password_md5)
{
    $sha1 = $this->password_md5;
}
else {
    $sha1 = md5($this->username.':'.$realm.':'.$this->password);
}

$sha2 = md5($this->method.':'.$this->uri);

$res = md5($sha1.':'.$nonce.':'.$sha2);

$this->auth = 'Proxy-Authorization: Digest username="'.$this->username.'"', realm="'.$realm.'"',
nonce="'.$nonce.'"', uri="'.$this->uri.'"', response="'.$res.'"', algorithm=MD5';
}

/**
 * Calculates WWW authorization response
 *
 */
public function authWWW()
{
    if (!$this->username)
    {
        throw new PhpSIPException("Missing auth username");
    }

    if (!$this->password && !$this->password_md5)
    {
        throw new PhpSIPException("Missing auth password");
    }

    $qop_present = false;
    if (strpos($this->response,'qop=') !== false)
    {
        $qop_present = true;

        // we can only do qop="auth"
        if (strpos($this->response,'qop="auth"') === false)
        {
            throw new PhpSIPException('Only qop="auth" digest authentication supported.');
```

```

$result = array();
if (!preg_match('/^WWW-Authenticate: .* realm="(.)"/imU', $this->response, $result))
{
    throw new PhpSIPException("Can't find realm in www-auth");
}

$realm = $result[1];

// nonce
$result = array();
if (!preg_match('/^WWW-Authenticate: .* nonce="(.)"/imU', $this->response, $result))
{
    throw new PhpSIPException("Can't find nonce in www-auth");
}

$nonce = $result[1];

if ($this->password_md5)
{
    $sha1 = $this->password_md5;
}
else
{
    $sha1 = md5($this->username.':'.$realm.':'.$this->password);
}

$sha2 = md5($this->method.':'.$this->uri);

if ($qop_present)
{
    $cnonce = md5(time());

    // nc
    $this->nc = sprintf('%08s', intval($this->nc)+1);

    $res = md5($sha1.':'.$nonce.':'.$this->nc.':'.$cnonce.':auth:'.$sha2);
//    $res = md5($sha1.':'.$nonce.':00000001:'.$cnonce.':auth:'.$sha2);
}
else
{
    $res = md5($sha1.':'.$nonce.':'.$sha2);
}

$this->auth = 'Authorization: Digest username="'.$this->username.'"', realm="'.$realm.'"',
nonce="'.$nonce.'"', uri="'.$this->uri.'"', response="'.$res.'"', algorithm=MD5';

if ($qop_present)
{
    $this->auth.= ', qop="auth", nc="'.$this->nc.'"', cnonce="'.$cnonce.'";
//    $this->auth.= ', qop="auth", nc="00000001", cnonce="'.$cnonce.'";
}
}

/**
 * Create network socket
 *
 * @return bool True on success
 */

```

```

private function createSocket()
{
    $this->getPort();

    if (!$this->src_ip)
    {
        throw new PhpSIPException("Source IP not defined.");
    }

    if (!$this->socket = @socket_create(AF_INET, SOCK_DGRAM, SOL_UDP))
    {
        $err_no = socket_last_error($this->socket);
        throw new PhpSIPException (socket_strerror($err_no));
    }

    if (@socket_bind($this->socket, $this->src_ip, $this->src_port))
    {
        $err_no = socket_last_error($this->socket);
        throw new PhpSIPException ("Failed to bind ".$this->src_ip.":".$this->src_port."
.socket_strerror($err_no));
    }

    if (@socket_set_option($this->socket, SOL_SOCKET, SO_RCVTIMEO, array("sec"=>$this-
>fr_timer,"usec"=>0)))
    {
        $err_no = socket_last_error($this->socket);
        throw new PhpSIPException (socket_strerror($err_no));
    }

    if (@socket_set_option($this->socket, SOL_SOCKET, SO_SNDTIMEO, array("sec"=>5,"usec"=>0)))
    {
        $err_no = socket_last_error($this->socket);
        throw new PhpSIPException (socket_strerror($err_no));
    }
}

/**
 * Close the connection
 *
 * @return bool True on success
 */
private function closeSocket()
{
    $this->releasePort();
    if (is_resource($this->socket))
    {
        socket_close($this->socket);
    }
}

private function saveSession()
{
    if (isset($_SESSION))
    {
        $fp = @fopen($this->session_file, 'w');
        if (!$fp)
        {
            throw new PhpSIPException ("Failed to open session file " . $this->session_file);
        }
    }
}

```

```

    }

    fwrite($fp, serialize($_SESSION));
    fclose($fp);
}

}

public function restoreSession()
{
    if (file_exists($this->session_file))
    {
        $session = file_get_contents($this->session_file);
        if (!isset($_SESSION))
        {
            session_start();
        }
        if ($session)
        {
            $_SESSION = array_merge($_SESSION, unserialize($session));
        }
    }
}

/**
 * Resets callid, to/from tags etc.
 *
 */
public function newCall()
{
    $this->cseq = 20;
    $this->call_id = null;
    $this->to_tag = null;;
    $this->from_tag = null;;

    /**
     * Body
     */
    $this->body = null;

    /**
     * Received Response
     */
    $this->response = null;
    $this->res_code = null;
    $this->res_contact = null;
    $this->res_cseq_method = null;
    $this->res_cseq_number = null;

    /**
     * Received Request
     */
    $this->req_method = null;
    $this->req_cseq_method = null;
    $this->req_cseq_number = null;
    $this->req_contact = null;

    $this->routes = array();
    $this->request_via = array();
}

```

```

}
}
?>

```

usuario/application/include/Auth.php

```

<?php
require_once(dirname(__FILE__) . '/../config/general.php');
require_once(dirname(__FILE__) . '/LDAP.php');
require_once(dirname(__FILE__) . '/DB.php');

class Auth {

    public static function login($username, $password) {

        if (!$username || !$password) {
            exit(json_encode(array('error' => 'Informe o ramal e senha')));
        }

        $LDAP = LDAP::getInstance();
        $DB = DB::getInstance();
        $DB->connect();

        $filter = "(&(cn=$username)(senhaWeb=$password))";
        $fields = array('permission', 'userPassword');
        $dn = "ou=sip";

        try {
            $resposta = $LDAP->search($dn, $filter, $fields);
        }
        catch (Exception $e) {
            Logger::log($e->getMessage(), __FILE__);
            return json_encode(array('error' => 'Ocorreu um erro'));
        }

        if ($resposta['count'] == 0) {
            return json_encode(array('error' => 'Dados não conferem'));
        }
        @session_start();

        for ($i=0; $i<$resposta['count']; $i++) {
            $_SESSION['codigo_permissao'] = $resposta[$i]['permission'][0];
            $_SESSION['senha'] = $resposta[$i]['userpassword'][0];
        }
        $_SESSION['logged_in'] = true;
        $_SESSION['ramal'] = $username;

        $query = sprintf('SELECT * FROM usuarios_configuracao
            WHERE ramal = %d',
                mysql_real_escape_string($username));

        $result = $DB->query($query);

        $_SESSION['webphone'] = 'click2call';
        $_SESSION['numero_max_contatos'] = 10;

```

```

while ($row = mysql_fetch_array($result)) {
    $_SESSION['webphone'] = $row['webphone'];
    $_SESSION['numero_max_contatos'] = $row['numero_max_contatos'];
}

return json_encode(array('error' => null, 'name' => $username));
}

public static function logout() {
    if (self::check()) {
        session_unset();
        session_destroy();
    }
    return json_encode(array('error' => null));
}

public static function check() {
    if (!isset($_SESSION)) {
        session_start();
    }
    if (isset($_SESSION['logged_in'])) {
        return true;
    }
    return false;
}
}
}

```

usuario/application/include/DataTables.php

```

<?php
require_once(dirname(__FILE__) . '/FirePHPCore/fb.php');

abstract class DataTables {

    public function getTableData($aParams) {
        $sQuery = $this->fnBuildQuery($aParams);
        try {
            $rResult = $this->DB->query($sQuery);
            $sQuery = 'SELECT FOUND_ROWS()';
            $rResultFilterTotal = $this->DB->query($sQuery);
            $aResultFilterTotal = mysql_fetch_array($rResultFilterTotal);
            $iFilteredTotal = $aResultFilterTotal[0];
        }
        catch (Exception $e) {
            return false;
        }
        /* Output parameters */
        $iEcho = isset($aParams['sEcho']) ? $aParams['sEcho'] : 0;
        $iTotal = $this->count($aParams['sTable']);
        $sColumns = $aParams['sColumns'];
        $aaData = $this->fnParseResult($rResult);
        /* Output */
        $sOutput = $this->fnBuildOutput($iEcho, $iTotal, $iFilteredTotal, $sColumns, $aaData);
        return json_encode($sOutput);
    }

    protected function fnBuildQuery($aParams) {
        /* Query statements */
    }
}

```

```

$aClauses = $this->fnBuildQueryClauses($aParams);

$sColumns = str_replace(',', '`,`', $aParams['sColumns']);
$sColumns = "`$sColumns`";

/* Query */
$query = "SELECT SQL_CALC_FOUND_ROWS $sColumns
        FROM {$aParams['sTable']}
        {$aClauses['sWhere']}
        {$aClauses['sOrder']}
        {$aClauses['sLimit']}";
return $query;
}

protected function fnBuildQueryClauses($aParams) {

    $aParams['sSearch'] = isset($aParams['sSearch']) ? $aParams['sSearch'] : '';
    $aParams['iDisplayStart'] = isset($aParams['iDisplayStart']) ? $aParams['iDisplayStart'] : 0;
    $aParams['iDisplayLength'] = isset($aParams['iDisplayLength']) ? $aParams['iDisplayLength'] : 1000;
    $aParams['sColumns'] = isset($aParams['sColumns']) ? $aParams['sColumns'] : '';
    $aColumns = explode(',', $aParams['sColumns']);

    /* Filtering */
    $sWhere = $this->fnBuildWhereClause($aColumns, $aParams['sSearch']);

    /* Ordering */
    $sOrder = $this->fnBuildOrderClause($aColumns, $aParams);

    /* Paging */
    $sLimit = $this->fnBuildLimitClause($aParams['iDisplayStart'], $aParams['iDisplayLength']);

    return array('sWhere' => $sWhere, 'sOrder' => $sOrder, 'sLimit' => $sLimit);
}

protected function fnBuildWhereClause($aColumns, $sSearch) {
    $sWhere = '';
    if (is_array($aColumns) && $sSearch != '') {
        $sSearch = preg_replace('/\s\s+/', ' ', $sSearch); //Remove extra spaces
        $sSearch = str_replace(' ', '%', $sSearch); //Replace space with %
        $sWhere = 'WHERE ';
        array_shift($aColumns);
        foreach ($aColumns as $aColumn) {
            $sWhere .= "`$aColumn` LIKE '%" . mysql_real_escape_string(utf8_decode($sSearch)) . "%' OR ";
        }
        $sWhere = substr($sWhere, 0, -4);
    }
    return $sWhere;
}

protected function fnBuildOrderClause($aColumns, $aParams) {
    $sOrder = '';
    if (isset($aParams['iSortCol_0']) && $aParams['iSortCol_0'] != '0') {
        $sOrder = 'ORDER BY ';
        for ($i=0; $i<$aParams['iSortingCols']; $i++) {
            $sOrder .= $aColumns[$aParams['iSortCol_'.$i]] . ' ' .
mysql_real_escape_string($aParams['sSortDir_'.$i]) . ', ';
        }
    }
}

```

```

    $sOrder = substr($sOrder, 0, -2);
}
return $sOrder;
}

protected function fnBuildLimitClause($iDisplayStart, $iDisplayLength) {
    $sLimit = '';
    if ($iDisplayStart >= 0 && $iDisplayLength >= 0) {
        $sLimit = 'LIMIT ' . intval($iDisplayStart) . ', ' .
            intval($iDisplayLength);
    }
    return $sLimit;
}

protected function fnBuildOutput($iEcho, $iTotal, $iFilteredTotal, $sColumns, $aaData) {
    /* Array Output */
    $sOutput = array();
    $sOutput['iEcho'] = intval($iEcho);
    $sOutput['iTotalRecords'] = intval($iTotal);
    $sOutput['iTotalDisplayRecords'] = intval($iFilteredTotal);
    $sOutput['sColumns'] = $sColumns;
    $sOutput['aaData'] = $aaData;
    return $sOutput;
}

protected function fnParseResult($rResult) {
    $aaData = array();
    $iIndex = 0;
    while ($aRow = mysql_fetch_array($rResult)) {
        $aaData[$iIndex] = array();
        $sId = array_shift($aRow);
        $aaData[$iIndex][] = '<input type="checkbox" name="ids[]" value="'. $sId. '" />';
        array_shift($aRow);
        $aKeys = array_keys($aRow);
        foreach ($aKeys as $sKey) {
            if (!is_numeric($sKey)) {
                if ($aRow[$sKey] === null) {
                    $aRow[$sKey] = '';
                }
                $aaData[$iIndex][] = utf8_encode($aRow[$sKey]);
            }
        }
        $iIndex++;
    }
    return $aaData;
}

protected function count($table) {
    $query = sprintf("SELECT COUNT(*) as num
        FROM %s",
            mysql_real_escape_string($table));

    try {
        $result = $this->DB->query($query);
        $total_rows = mysql_result($result, 0, 'num');
    }
    catch (Exception $e) {
        return 0;
    }
}

```



```

        return $total_rows;
    }
}

```

usuario/application/include/DB.php

```

<?php
require_once(dirname(__FILE__) . '/../config/general.php');

class DB {
    private $connection;
    private $host;
    private $user;
    private $password;
    private $database;
    private static $instance;

    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new DB();
        }
        return self::$instance;
    }

    private function __construct() {
        global $config;
        $this->host = $config['mysql_host'];
        $this->user = $config['mysql_user'];
        $this->password = $config['mysql_password'];
        $this->database = $config['mysql_database'];
    }

    public function __destruct() {
        $this->disconnect();
    }

    public function connect() {
        if (!$this->connection = mysql_connect($this->host, $this->user, $this->password)) {
            throw new Exception('MySQL Connect: ' . mysql_error());
        }
        if (!mysql_select_db($this->database, $this->connection)) {
            throw new Exception('MySQL Select DB: ' . mysql_error());
        }
    }

    public function connect_to_midia() {
        global $config;
        $this->disconnect();
        $this->host = $config['mysql_midia_host'];
        $this->user = $config['mysql_midia_user'];
        $this->password = $config['mysql_midia_password'];
        $this->database = $config['mysql_midia_database'];
        $this->connect();
    }

    public function connect_to_local() {
        global $config;
        $this->disconnect();
    }
}

```

```

$this->host = $config['mysql_host'];
$this->user = $config['mysql_user'];
$this->password = $config['mysql_password'];
$this->database = $config['mysql_database'];
$this->connect();
}

public function disconnect() {
    if (is_resource($this->connection)) {
        mysql_close($this->connection);
    }
}

public function query($query) {
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if (!$result = mysql_query($query, $this->connection)) {
        throw new Exception('MySQL Query: ' . mysql_error());
    }
    return $result;
}

public function insert($table, $fields, $nulls = true) {
    if (!$table) {
        throw new InvalidArgumentException('The first argument ($table) must not be empty');
    }
    if (!is_array($fields)) {
        throw new InvalidArgumentException('The second argument ($fields) must be an array
(dictionary)');
    }
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if (!$nulls) {
        $fields = array_filter($fields, 'is_not_empty');
    }
    if (!$fields) {
        return true;
    }
    $query = "INSERT INTO $table (`" . implode('`,`', array_keys($fields)) . "`) VALUES(";
    foreach ($fields as $field) {
        if (get_magic_quotes_gpc()) {
            $field = stripslashes($field);
        }
        if (is_empty($field)) {
            $query.= "NULL,";
        }
        else {
            $query.= "'" . mysql_real_escape_string(utf8_decode($field)) . "',";
        }
    }
    $query = substr($query, 0, -1) . ')';
    return $this->query($query);
}

public function update($table, $fields, $where, $nulls = true) {
    if (!$table) {

```

```

        throw new InvalidArgumentException('The first argument ($table) must not be empty');
    }
    if (!is_array($fields)) {
        throw new InvalidArgumentException('The second argument ($fields) must be an array
(dictionary)');
    }
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if (!$nulls) {
        $fields = array_filter($fields, 'is_not_empty');
    }
    if (!$fields) {
        return true;
    }
    $query = "UPDATE $table SET ";
    foreach ($fields as $column=>$value) {
        if (get_magic_quotes_gpc()) {
            $value = stripslashes($value);
        }
        if (is_empty($value)) {
            $query.= "`$column` = NULL,";
        }
        else {
            $query.= "`$column` = '".mysql_real_escape_string(utf8_decode($value))."',";
        }
    }
    $query = substr($query, 0, -1);
    if ($where) {
        $query.= " WHERE $where";
    }
    return $this->query($query);
}

public function get_last_id() {
    return mysql_insert_id($this->connection);
}

public function parse_result($result_sql) {
    $result = array();
    $index = 0;
    //Percorre todos os resultados da consulta
    while ($row = mysql_fetch_array($result_sql)) {
        //Obtém os índices do registro da tabela (nomes das colunas)
        $keys = array_keys($row);
        foreach ($keys as $key) {
            //Ignora os índices numéricos para evitar duplicação (obtém apenas os nomes das colunas)
            if (!is_numeric($key)) {
                //Armazena o valor no array do resultado
                $result[$index][$key] = replace_quotes(utf8_encode($row[$key]));
            }
        }
        $index++;
    }
    //Retorna o resultado em forma de matriz
    return $result;
}
}

```

```

usuario/application/include/functions.php
<?php
/**
 * Handles the exception details according to the log verbosity settings
 *
 * @param $exception
 */
function handle_exception($exception) {
    global $config;
    try {
        if ($config['log_verbosity'] == LOG_VERBOSITY_DEBUG) {
            print '<b>Error:</b> ' . $exception->getMessage();
            $trace = $exception->getTrace();
            if ($trace) {
                $trace = $trace[0];
                print '<br /><b>File:</b> ' . $exception->getFile();
                if (isset($trace['file']) && isset($trace['line'])) {
                    print '<br /><b>Called on:</b> ' . $trace['file'] . ' (Line ' . $trace['line'] .
                ')';
            }
            if (isset($trace['class'])) {
                print '<br /><b>Class:</b> ' . $trace['class'];
            }
            if (isset($trace['function'])) {
                print '<br /><b>Function:</b> ' . $trace['function'];
            }
            if (isset($trace['args'])) {
                print '<br /><b>Arguments:</b> ' . print_r($trace['args'], true);
            }
        }
        print '<br /><b>Stack trace:</b> ' . $exception->getTraceAsString();
    }
    else {
        error_redirect();
    }
    if ($config['log_enabled']) {
        Logger::log($exception->getMessage(), $exception->getFile());
    }
}
catch (Exception $e) {
    print get_class($e) . " thrown within the exception handler. Message: \"$e->getMessage().\" on line
    \"$e->getLine()";
}
}

/**
 * Generates an exception from an error
 * @param $errno
 * @param $errstr
 * @param $errfile
 * @param $errline
 */
function handle_error($errno, $errstr, $errfile, $errline) {
    if (0 == error_reporting()) {
        return;
    }
    throw new Exception($errstr, $errno);
}

```

```

}

/**
 * Sends the output buffer and turn on output buffering again
 */
function flush_buffers() {
    ob_end_flush();
    ob_flush();
    flush();
    ob_start();
}

/**
 * Redirects the user to a custom error page
 */
function error_redirect() {
    //header("HTTP/1.0 500 Internal Server Error");
    if (!headers_sent()) {
        header('Status: 500');
        //header("Location: /index.php");
        ob_clean();
        require_once(dirname(__FILE__) . '/TemplateHandler.php');
        $template = new TemplateHandler();
        try {
            $template->set_body_file(DEFAULT_ERROR_FILE);
            $template->set_title('nKey - Erro no servidor');
            $template->show();
        }
        catch (Exception $e) {
            print 'Ocorreu um erro inesperado. Por favor, tente novamente mais tarde.';
        }
    }
    else {
        print 'Ocorreu um erro inesperado. Por favor, tente novamente mais tarde.';
    }
}

/**
 * Cut string to n symbols and add delim but do not break words.
 *
 * @param $string - String we are operating with
 * @param $limit - Character count to cut to
 * @param $break - Where to stop truncating. Default: '.'
 * @param $pad - Delimiter. Default: '...'
 */
function text_trim($string, $limit, $break = '.', $pad = ' (...)') {
    // return with no change if string is shorter than $limit
    if (strlen($string) <= $limit) {
        return $string;
    }
    // is $break present between $limit and the end of the string?
    if (($breakpoint = strpos($string, $break, $limit)) !== false) {
        if ($breakpoint < strlen($string) - 1) {
            $string = substr($string, 0, $breakpoint) . $pad;
        }
    }
    return restore_tags($string);
}

```

```

/**
 * Attempts to close any tags that have been left open in a HTML string.
 * Please Note: This function will restore only HTML tags, opened and closed in a valid order, with no
attributes
 *
 * @param $input - HTML String
 */
function restore_tags($input) {
    $opened = array();
    // loop through opened and closed tags in order
    if (preg_match_all("</(\?[a-z]+)>?/i", $input, $matches)) {
        foreach ($matches[1] as $tag) {
            if (preg_match("/^[a-z]+$/i", $tag, $regs)) {
                // a tag has been opened
                if (strtolower($regs[0]) != 'br') {
                    $opened[] = $regs[0];
                }
            }
            elseif (preg_match("/^\\[a-z]+\$/i", $tag, $regs)) {
                // a tag has been closed
                $keys = array_keys($opened, $regs[1]);
                unset($opened[array_pop($keys)]);
            }
        }
    }
    // close tags that are still open
    if ($opened) {
        $tagstoclose = array_reverse($opened);
        foreach ($tagstoclose as $tag) {
            $input .= "</$tag>";
        }
    }
    return $input;
}

/**
 * Replaces double (") and sigle (') quotes from a string to their HTML symbol
 *
 * @param $str - Input string
 */
function replace_quotes($str) {
    $str = str_replace('"', '&quot;', $str);
    $str = str_replace("'", '&#39;', $str);
    return $str;
}

/**
 * Checks if the date is valid
 *
 * @param $date - The date in format DD-MM-YYYY
 */
function validate_date($date) {
    //match the format of the date
    if (preg_match (" /^[0-9]{2}\/([0-9]{2})\/([0-9]{4})$/", $date, $parts)) {
        //check if the date is valid of not
        if (checkdate($parts[2], $parts[1], $parts[3])) {

```

```

        return true;
    }
}
return false;
}

/**
 * Checks if the time is valid
 *
 * @param $time - The date in format HH:MM or HH:MM:SS
 */
function validate_time($time) {
    if (preg_match ("/^[0-1]\d|2[0-3]:([0-5]\d):([0-5]\d)?$/", $time)) {
        return true;
    }
    return false;
}

/**
 * Validate an email address.
 * Returns true if the email address has the email address format and the domain exists.
 *
 * @param $email - Email address (raw input)
 */
function validate_email($email) {
    $isValid = true;
    $atIndex = strrpos($email, '@');
    if (is_bool($atIndex) && !$atIndex) {
        $isValid = false;
    } else {
        $domain = substr($email, $atIndex+1);
        $local = substr($email, 0, $atIndex);
        $localLen = strlen($local);
        $domainLen = strlen($domain);

        if ($localLen < 1 || $localLen > 64) {
            // local part length exceeded
            $isValid = false;
        } else if ($domainLen < 1 || $domainLen > 255) {
            // domain part length exceeded
            $isValid = false;
        } else if ($local[0] == '.' || $local[$localLen-1] == '.') {
            // local part starts or ends with '.'
            $isValid = false;
        } else if (preg_match('/\\\\.\\.\\./', $local)) {
            // local part has two consecutive dots
            $isValid = false;
        } else if (!preg_match('/^[A-Za-z0-9\\-\\.]+$/', $domain)) {
            // character not valid in domain part
            $isValid = false;
        } else if (preg_match('/\\\\.\\.\\./', $domain)) {
            // domain part has two consecutive dots
            $isValid = false;
        } else if (!preg_match('/^(\\\\\\\\.|[A-Za-z0-9!#%&`_=@\\/$!'+?^{}|~.-])+$/',
str_replace("\\\\", "", $local))) {
            // character not valid in local part unless
            // local part is quoted
            if (!preg_match('/^"(\\\\\\\\|"[^"]")+$/', str_replace("\\\\", "", $local))) {

```

```

        $isValid = false;
    }
}
if ($isValid && !(checkdnsrr($domain, 'MX') || checkdnsrr($domain, 'A'))) {
    // domain not found in DNS
    $isValid = false;
}
}
return $isValid;
}

/**
 * Detects if the user browser is IE
 */
function is_internet_explorer() {
    if (isset($_SERVER['HTTP_USER_AGENT']) && (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== false)) {
        return true;
    }
    else {
        return false;
    }
}

/**
 * Generates a pagination controller
 *
 * @param $page - Current page
 * @param $totalitems - Number of items to paginate
 * @param $limit - Items per page
 * @param $adjacents -
 * @param $file - URL to be called by the javascript function
 */
function create_pagination($page = 1, $totalitems, $limit = 4, $adjacents = 1, $file = '') {
    //defaults
    if (!$adjacents) $adjacents = 1;
    if (!$limit) $limit = 15;
    if (!$page) $page = 1;

    //other vars
    $prev = $page - 1; //previous page is page - 1
    $next = $page + 1; //next page is page + 1
    $lastpage = ceil($totalitems / $limit); //lastpage is = total items / items per page,
rounded up.
    $lpm1 = $lastpage - 1; //last page minus 1

    /*
    Now we apply our rules and draw the pagination object.
    We're actually saving the code to a variable in case we want to draw it more than once.
    */
    $pagination = "";
    if ($lastpage > 1) {
        $pagination .= "<div class=\"pagination-wrapper\">";
        $pagination .= "<ul class=\"pagination\">";

        //previous button
        if ($page > 1)
            $pagination .= "<li class=\"previous\"><a href=\"$file?page=$prev\"><<</a></li>";
    }

```



```

else
    $pagination .= "<li class=\"previous-off\"><</li>";

//pages
if ($lastpage < 7 + ($adjacents * 2)) { //not enough pages to bother breaking it up
    for ($counter = 1; $counter <= $lastpage; $counter++) {
        if ($counter == $page)
            $pagination .= "<li class=\"active_page\">$counter</li>";
        else
            $pagination .= "<li><a href=\"\$file?page=$counter\">$counter</a></li>";
    }
}
elseif ($lastpage >= 7 + ($adjacents * 2)) { //enough pages to hide some
    //close to beginning; only hide later pages
    if ($page < 1 + ($adjacents * 3)) {
        for ($counter = 1; $counter < 4 + ($adjacents * 2); $counter++) {
            if ($counter == $page)
                $pagination .= "<li class=\"active_page\">$counter</li>";
            else
                $pagination .= "<li><a href=\"\$file?page=$counter\">$counter</a></li>";
        }
        $pagination .= "...";
        $pagination .= "<li><a href=\"\$file?page=$lpml\">$lpml</a></li>";
        $pagination .= "<li><a href=\"\$file?page=$lastpage\">$lastpage</a></li>";
    }
    //in middle; hide some front and some back
    elseif ($lastpage - ($adjacents * 2) > $page && $page > ($adjacents * 2)) {
        $pagination .= "<li><a href=\"\$file?page=1\">1</a></li>";
        $pagination .= "<li><a href=\"\$file?page=2\">2</a></li>";
        $pagination .= "...";
        for ($counter = $page - $adjacents; $counter <= $page + $adjacents; $counter++) {
            if ($counter == $page)
                $pagination .= "<li class=\"active_page\">$counter</li>";
            else
                $pagination .= "<li><a href=\"\$file?page=$counter\">$counter</a></li>";
        }
        $pagination .= "<li><a href=\"\$file?page=$lpml\">$lpml</a></li>";
        $pagination .= "<li><a href=\"\$file?page=$lastpage\">$lastpage</a></li>";
    }
    //close to end; only hide early pages
    else {
        $pagination .= "<li><a href=\"\$file?page=1\">1</a></li>";
        $pagination .= "<li><a href=\"\$file?page=2\">2</a></li>";
        $pagination .= "...";
        for ($counter = $lastpage - (1 + ($adjacents * 3)); $counter <= $lastpage; $counter++) {
            if ($counter == $page)
                $pagination .= "<li class=\"active_page\">$counter</li>";
            else
                $pagination .= "<li><a href=\"\$file?page=$counter\">$counter</a></li>";
        }
    }
}

//next button
if ($page < $counter - 1)
    $pagination .= "<li class=\"next\"><a href=\"\$file?page=$next\">>></a></li>";
else
    $pagination .= "<li class=\"next-off\">>></li>";

```

```

        $pagination .= "</ul></div>\n";
    }
    return $pagination;
}

/**
 * Creates a thumbnail of a image
 *
 * @param $file - Path to the original image file
 * @param $width - Width in pixels
 * @param $output - Whether output the image or save it
 * @param $targetPath - Path+Filename to save the thumbnail
 */
function create_thumbnail($file, $width = 100, $output = false, $targetPath = '') {
    $imgSize = getimagesize($file);
    $fileParts = pathinfo($file);

    switch(strtolower($fileParts['extension'])) {
        case 'jpg':
        case 'jpeg':
            $image = imagecreatefromjpeg($file);
            break;
        case 'png':
            $image = imagecreatefrompng($file);
            break;
        case 'gif':
            $image = imagecreatefromgif($file);
            break;
        default:
            return false;
    }

    $height = $imgSize[1]/$imgSize[0]*$width; //This maintains proportions

    $src_w = $imgSize[0];
    $src_h = $imgSize[1];

    $picture = imagecreatetruecolor($width, $height);
    imagealphablending($picture, false);
    imagesavealpha($picture, true);

    $bool = imagecopyresampled($picture, $image, 0, 0, 0, 0, $width, $height, $src_w, $src_h);

    if ($bool) {
        if (!$targetPath) {
            $targetPath = $fileParts['dirname'] . '/' . $fileParts['filename'] . '_thumb.' .
                $fileParts['extension'];
        }
        switch(strtolower($fileParts['extension'])) {
            case 'jpg':
            case 'jpeg':
                if (!headers_sent() && $output) {
                    header('Content-Type: image/jpeg');
                    imagejpeg($picture);
                }
            else {
                imagejpeg($picture, $targetPath, 80);
            }
        }
    }
}

```

```

        break;
    case 'png':
        if (!headers_sent() && $output) {
            header('Content-Type: image/png');
            imagepng($picture);
        }
        else {
            imagepng($picture, $targetPath);
        }
        break;
    case 'gif':
        if (!headers_sent() && $output) {
            header('Content-Type: image/gif');
            imagegif($picture);
        }
        else {
            imagegif($picture, $targetPath);
        }
        break;
    }
} else {
    return false;
}
imagedestroy($picture);
imagedestroy($image);
return $targetPath;
}

/**
 * Retrieve the mime type of a file according to Apache mime.types
 *
 * @param string $filename - Name of the file to detect the mime type
 * @param string $mimePath - Path to mime.types file
 */
function get_mime_type($filename, $mimePath = '/etc') {
    $fileext = substr(strrchr($filename, '.'), 1);
    if (empty($fileext)) return (false);
    $regex = "/^([\w\+\.\/])\s+(\w+)\s+(\$fileext\s)/i";
    $lines = @file("$mimePath/mime.types");
    foreach($lines as $line) {
        if (substr($line, 0, 1) == '#') continue; // skip comments
        $line = rtrim($line) . " ";
        if (!preg_match($regex, $line, $matches)) continue; // no match to the extension
        return ($matches[1]);
    }
    return (false); // no match at all
}

/**
 * Generate a hash from a file
 *
 * @param $filename - Path (location) of the file
 */
function file_hash($filename) {
    $readsize = 16 * 1024; //16Kb
    if ($filename && $handle = @fopen($filename, 'r')) {
        $contents = fread($handle, $readsize); //Reads 16kb from the beginning of the file
        fseek($handle, -$readsize, SEEK_END); //Sets the position to 16kb before the end-of-file
    }
}

```

```

        $contents.= fread($handle, $readsize); //Reads the last 16kb from the file
        fclose($handle);
        return md5($contents);
    }
    return false;
}

/**
 * Delete a file or recursively delete a directory
 *
 * @param string $str - Path to file or directory
 * @param boolean $remDir - Remove the directories or only the files
 */
function recursiveDelete($str, $remDir = true){
    if (is_file($str)){
        return @unlink($str);
    }
    elseif (is_dir($str)){
        $scan = glob(rtrim($str, '/') . '/*');
        foreach ($scan as $index=>$path){
            recursiveDelete($path);
        }
        if ($remDir) {
            return @rmdir($str);
        }
        else {
            return true;
        }
    }
}

/**
 * Replace special characters from a filename
 *
 * @param string $filename - Input filename
 */
function clean_filename($filename) {
    $filename = strtolower($filename);
    $ext_point = stripos($filename, ".");
    if ($ext_point === false) {
        return false;
    }
    $ext = substr($filename, $ext_point, strlen($filename));
    $filename = substr($filename, 0, $ext_point);
    $filename = preg_replace('/\s/', '_', $filename);
    return clean_string($filename).$ext;
}

/**
 * Replace or remove special characters from a string and convert them to lowercase
 *
 * @param $string - Input string
 */
function clean_string($string) {
    $string = strtolower(normalize($string));
    return preg_replace('/[^\sa-z0-9-]/', '', $string);
}

```

```

/**
 * Converts a ISO Date format to DD Month
 *
 * @param $date - Input date (YYYY-MM-DD [HH:MM:SS])
 */
function clean_date($date) {
    $date = explode(' ', $date);
    $date = $date[0];
    $date = explode('-', $date);
    $month = '';
    switch ($date[1]) {
        case "01": $month = "Jan"; break;
        case "02": $month = "Fev"; break;
        case "03": $month = "Mar"; break;
        case "04": $month = "Abr"; break;
        case "05": $month = "Mai"; break;
        case "06": $month = "Jun"; break;
        case "07": $month = "Jul"; break;
        case "08": $month = "Ago"; break;
        case "09": $month = "Set"; break;
        case "10": $month = "Out"; break;
        case "11": $month = "Nov"; break;
        case "12": $month = "Dez"; break;
    }
    $day = $date[2];
    return $day . ' ' . $month;
}

/**
 * Gets the month name
 *
 * @param $month - Month number
 */
function get_month_name($month) {
    switch (intval($month)) {
        case 1: $month = "janeiro"; break;
        case 2: $month = "fevereiro"; break;
        case 3: $month = "março"; break;
        case 4: $month = "abril"; break;
        case 5: $month = "maio"; break;
        case 6: $month = "junho"; break;
        case 7: $month = "julho"; break;
        case 8: $month = "agosto"; break;
        case 9: $month = "setembro"; break;
        case 10: $month = "outubro"; break;
        case 11: $month = "novembro"; break;
        case 12: $month = "dezembro"; break;
    }
    return $month;
}

/**
 * Replace not usual characters of a string
 *
 * @param $string - Input string
 */
function normalize($string) {
    $stable = array(

```

```

'S'=>'S', 'š'=>'s', 'Đ'=>'Dj', 'đ'=>'dj', 'Ž'=>'Z', 'ž'=>'z', 'Č'=>'C', 'č'=>'c', 'Ć'=>'C',
'ć'=>'c',
'À'=>'A', 'Á'=>'A', 'Â'=>'A', 'Ã'=>'A', 'Ä'=>'A', 'Å'=>'A', 'Æ'=>'A', 'Ç'=>'C', 'È'=>'E', 'É'=>'E',
'Ê'=>'E', 'Ë'=>'E', 'Ì'=>'I', 'Í'=>'I', 'Î'=>'I', 'Ï'=>'I', 'Ñ'=>'N', 'Ò'=>'O', 'Ó'=>'O', 'Ô'=>'O',
'Õ'=>'O', 'Ö'=>'O', 'Ø'=>'O', 'Ù'=>'U', 'Ú'=>'U', 'Û'=>'U', 'Ü'=>'U', 'Ý'=>'Y', 'ß'=>'B',
'ß'=>'Ss',
'à'=>'a', 'á'=>'a', 'â'=>'a', 'ã'=>'a', 'ä'=>'a', 'å'=>'a', 'æ'=>'a', 'ç'=>'c', 'è'=>'e', 'é'=>'e',
'ê'=>'e', 'ë'=>'e', 'ì'=>'i', 'í'=>'i', 'î'=>'i', 'ï'=>'i', 'ð'=>'o', 'ñ'=>'n', 'ò'=>'o', 'ó'=>'o',
'ô'=>'o', 'õ'=>'o', 'ö'=>'o', 'ø'=>'o', 'ù'=>'u', 'ú'=>'u', 'û'=>'u', 'ý'=>'y', 'ÿ'=>'y', 'þ'=>'b',
'ÿ'=>'y', 'Ŕ'=>'R', 'ŕ'=>'r',
);
return strstr($string, $table);
}

/**
 * Generate SEO friendly URLs
 *
 * @param $str - String to be cleaned/normalized
 */
function generate_url($str) {
    $str = str_replace(' ', '-', $str);
    return urlencode(clean_string($str));
}

/**
 * Gets the description of an file upload error
 *
 * @see http://www.php.net/manual/en/features.file-upload.errors.php
 * @param $error_code - Code from $_FILES['file']['error']
 */
function file_upload_error_message($error_code) {
    switch ($error_code) {
        case UPLOAD_ERR_INI_SIZE:
            return 'O tamanho do arquivo excedeu o limite do servidor';
        case UPLOAD_ERR_FORM_SIZE:
            return 'O tamanho do arquivo excedeu o limite';
        case UPLOAD_ERR_PARTIAL:
            return 'O arquivo foi parcialmente enviado';
        case UPLOAD_ERR_NO_FILE:
            return 'Nenhum arquivo foi enviado';
        case UPLOAD_ERR_NO_TMP_DIR:
            return 'Diretório temporário não encontrado';
        case UPLOAD_ERR_CANT_WRITE:
            return 'Erro ao salvar o arquivo no servidor';
        case UPLOAD_ERR_EXTENSION:
            return 'Uma extensão do PHP previniu o arquivo de ser salvo no servidor';
        default:
            return 'Erro de envio de arquivo desconhecido';
    }
}

/**
 * Compares array1 against array2 and returns the difference.
 * This function handles multi-dimensional arrays or objects through serialization
 *
 * @param $ar1 - The array to compare from
 * @param $ar2 - An array to compare against
 */

```

```

function array_diff_no_cast(&$ar1, &$ar2) {
    $diff = Array();
    foreach ($ar1 as $key => $vall) {
        if (array_search($vall, $ar2) === false) {
            $diff[$key] = $vall;
        }
    }
    return $diff;
}

/**
 * Checks if the argument is null or empty
 *
 * @param $value - The input value
 */
function is_empty($value) {
    return (is_null($value) || $value == '');
}

/**
 * Checks if the argument is not null or empty
 *
 * @param $value - The input value
 */
function is_not_empty($value) {
    return !is_empty($value);
}

/**
 * Sends an email
 *
 * @param object $DB - Database helper instance to search for the SMTP server
 * @param string $to - Destination e-mail
 * @param string $subject - Subject of the message
 * @param string $body - Content of the message
 * @return boolean - Success/Fail
 */
function send_mail($DB, $to, $subject, $body) {

    require_once(dirname(__FILE__) . '/PHPMailer/class.phpmailer.php');

    $sql = $DB->query("SELECT value
                                FROM voip.configurador
                                WHERE field_name='smtp'
                                OR field_name='smtp_from'");

    $smtp = mysql_result($sql,0);
    $smtp_from = mysql_result($sql,1);

    $mail = new PHPMailer();

    $mail->IsSMTP();
    $mail->Host = $smtp;
    $mail->SMTPAuth = false;

    $mail->From = $smtp_from;
    $mail->FromName = 'PBX-IP UFSC';
}

```

```

$mail->AddAddress($to);

$mail->Subject = $subject;
$mail->Body = $body;
$mail->CharSet = 'utf-8';

if (!$mail->Send()) {
    return $mail->ErrorInfo;
}
else {
    return true;
}
}

/**
 * Run a background process
 *
 * @param string $Command - Command to be executed
 * @param int $Priority - Level of priority
 * @return int - Process ID
 */
function run_in_background($Command, $Priority = 0) {
    if ($Priority) {
        $PID = shell_exec("nohup nice -n $Priority $Command 2> /dev/null & echo $!");
    }
    else {
        $PID = shell_exec("nohup $Command > /dev/null 2> /dev/null & echo $!");
        //$PID = shell_exec("nohup $Command 2> /dev/null & echo $!");
    }
    return($PID);
}

/**
 * Check if the process is running
 *
 * @param $PID - Process ID
 * @return boolean - Yes/No
 */
function is_process_running($PID) {
    exec("ps $PID", $ProcessState);
    return(count($ProcessState) >= 2);
}

```

```

usuario/application/include/LDAP.php

```

```

<?php
require_once(dirname(__FILE__) . '/DB.php');

class LDAP {
    private $connection;
    private $host;
    private $base_dn;
    private $user;
    private $password;
    public $realm;
    private $auth;
    private static $instance;
}

```



```

public static function getInstance() {
    if (!self::$instance) {
        self::$instance = new LDAP();
    }
    return self::$instance;
}

private function __construct() {
    $DB = DB::getInstance();
    $sql_configurador = "SELECT value FROM voip.configurador
        WHERE field_name = 'voip_REALM' || field_name = 'IP_master' ||
        field_name = 'IP_slave' || field_name = 'ldap_dn' ||
        field_name = 'ldap_voip_password' || field_name = 'ldap_user'
        ORDER BY field_name";

    $result = $DB->query($sql_configurador);

    $host_master = mysql_result($result,0); //IP da maquina master do ldap.
    $host_slave = mysql_result($result,1); //IP da maquina slave do ldap.
    $base_dn = mysql_result($result,2); //Rept do ldap.
    $user = mysql_result($result,3); //Usuário do ldap.
    $password = mysql_result($result,4); //Senha do ldap.
    $realm = mysql_result($result,5); //Realm.

    $this->host = $host_master;
    $this->base_dn = $base_dn;
    $this->user = $user;
    $this->password = $password;
    $this->realm = $realm;

    try {
        $this->connect();
    }
    catch (Exception $e) {
        $this->host = $host_slave;
        $this->connect();
    }
}

public function __destruct() {
    $this->disconnect();
}

public function set_host($host) {
    $this->host = $host;
}

public function connect() {
    if (!$this->connection = ldap_connect($this->host)) {
        throw new Exception("LDAP Connect: " . ldap_error($this->connection));
    }
    ldap_set_option($this->connection, LDAP_OPT_PROTOCOL_VERSION, 3);
    $bind = "cn=" . $this->user . "," . $this->base_dn;
    if (!$this->auth = ldap_bind($this->connection, $bind, $this->password)) {
        throw new Exception("LDAP Bind: " . ldap_error($this->connection));
    }
}
}

```

```

public function disconnect() {
    if (is_resource($this->connection)) {
        ldap_close($this->connection);
    }
}

public function search($dn, $filter, $attributes = array(), $getEntries = true, $oneLevelSearch =
false) {
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if ($this->auth) {
        $dn = $this->configure_dn($dn);
        if ($oneLevelSearch) {
            if (!$result = @ldap_list($this->connection, $dn, $filter, $attributes))
            {
                throw new Exception("LDAP List: " . ldap_error($this-
>connection));
            }
        }
        else {
            if (!$result = @ldap_search($this->connection, $dn, $filter,
$attributes)) {
                throw new Exception("LDAP Search: " . ldap_error($this-
>connection));
            }
        }
        if ($getEntries) {
            if (!$entries = @ldap_get_entries($this->connection, $result)) {
                throw new Exception("LDAP Get Entries: " . ldap_error($this-
>connection));
            }
            return $entries;
        }
        else {
            return $result;
        }
    }
    else {
        throw new Exception("LDAP Search: Não possui autorização para realizar esta
operação");
    }
}

public function add($dn, $entry = array()) {
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if ($this->auth) {
        $dn = $this->configure_dn($dn);
        if (!@ldap_add($this->connection, $dn, $entry)) {
            throw new Exception("LDAP Add: " . ldap_error($this->connection));
        }
        return true;
    }
    else {
        throw new Exception("LDAP Add: Não possui autorização para realizar esta
operação");
    }
}

```

```

    }
}

public function update($dn, $entry = array()) {
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if ($this->auth) {
        $dn = $this->configure_dn($dn);
        if (!@ldap_modify($this->connection, $dn, $entry)) {
            throw new Exception("LDAP Modify: " . ldap_error($this->connection));
        }
        return true;
    }
    else {
        throw new Exception("LDAP Modify: Não possui autorização para realizar esta
operação");
    }
}

public function update_delete($dn, $entry = array()) {
    if(!is_resource($this->connection)) {
        $this->connect();
    }
    if ($this->auth) {
        $dn = $this->configure_dn($dn);
        if (!@ldap_mod_del($this->connection, $dn, $entry)) {
            throw new Exception("LDAP Modify Delete: " . ldap_error($this->connection));
        }
        return true;
    }
    else {
        throw new Exception("LDAP Modify: Não possui autorização para realizar esta operação");
    }
}

public function delete($dn, $recursive = false) {
    if (!is_resource($this->connection)) {
        $this->connect();
    }
    if ($this->auth) {
        if ($recursive == false) {
            $dn = $this->configure_dn($dn);
            if (!@ldap_delete($this->connection, $dn)) {
                throw new Exception("LDAP Delete: " . ldap_error($this-
>connection));
            }
        }
        else {
            //searching for sub entries
            $result = $this->search($dn, "(objectClass=*)", array(""), true, true);
            for ($i=0; $i<$result['count']; $i++){
                //deleting recursively sub entries
                $this->delete($result[$i]['dn'], $recursive);
            }
            $dn = $this->configure_dn($dn);
            if (!@ldap_delete($this->connection, $dn)) {
                throw new Exception("LDAP Delete: " . ldap_error($this-

```

```

>connection));
        }
    }
    return true;
}
else {
    throw new Exception("LDAP Delete: Não possui autorização para realizar esta
operação");
}
}

/**
 * Use http://php.net/ldap_count_entries to calculate the min and max pages values before call this
function. The first page value is 0 (zero).
 */
 * Order a search in ascending and descending order.
 *
 * @param $searchResult resource from ldap_search()
 * @param $field string of attribute to order
 * @param $order string "asc" or "desc"
 * @param $page integer page number, first is 0 zero
 * @param $perPage integer entries per page
 * @return string[]
 */
public function sort($searchResult, $field, $order = 'asc', $page = null, $perPage = null, $start =
0) {
    $totalEntries = ldap_count_entries($this->connection, $searchResult);
    if ($perPage === null) {
        # fetch all in one page
        $end = $totalEntries - 1;
    }
    else {
        # calculate range of page
        $start = (ceil($totalEntries/$perPage)-1) * $perPage;
        $end = $start + $perPage - 1;
        if ($order == 'desc') {
            # revert range
            $start = $totalEntries - 1 - $end;
            $end = $start + $perPage - 1;
        }
    }
    # fetch entries
    ldap_sort($this->connection, $searchResult, $field);
    $list = array();
    $list['count'] = $totalEntries;
    for ($i = 0, $entry = ldap_first_entry($this->connection, $searchResult);
        $i <= $end && is_resource($entry);
        $i++, $entry = ldap_next_entry($this->connection, $entry)) {
        if ($i >= $start) {
            array_push($list, ldap_get_attributes($this->connection, $entry));
        }
    }
    # if order is desc revert page's entries
    return $order == 'desc' ? array_reverse($list) : $list;
}

private function configure_dn($dn) {
    //$dn = str_replace(" ", "", $dn);
    if (strstr($dn, $this->base_dn) === false) {

```

```

        $dn = $dn . ',' . $this->base_dn;
    }
    return $dn;
}

public function parse_result($response) {
    $result = array();
    $index = 0;
    //Percorre todos os resultados da consulta
    foreach ($response as $item) {
        if (is_array($item)) {
            //Obtém os índices do registro da tabela (nomes das colunas)
            $keys = array_keys($item);
            foreach ($keys as $key) {
                if (is_array($item[$key])) {
                    //Armazena o valor no array do resultado
                    $result[$index][$key] = replace_quotes($item[$key][0]);
                }
            }
            $index++;
        }
    }
    //Retorna o resultado em forma de matriz
    return $result;
}
}
?>

```

usuario/application/include/Log.php

```

<?php
class Log {

    private $data;
    private $idAdm;
    private $ip;
    private $sql;
    private $consulta;
    private $mysql;

    function Log($objMysql) {
        $this->data = date("Y-m-d H:i:s");
        if (isset($_SESSION["id_adm"]))
            $this->idAdm = $_SESSION["id_adm"];
        $this->ip = $_SERVER["REMOTE_ADDR"];
        $this->mysql = $objMysql;
    }

    function inserirLog( $log_acao, $log_descricao ){
        $this->acao = $log_acao;
        $this->descricao = $log_descricao;
        $this->sql = "INSERT INTO voip.log (data_log, idAdm_log, ip_log, acao_log, descricao_log)
VALUES ('$this->data', '$this->idAdm', '$this->ip', '$this->acao', '$this->descricao')";
        $this->consulta = $this->mysql->query($this->sql);
        if($this->consulta)
            return true;
        else

```

```

        return false;
    }

    function getIdAcao($nome_acao) {
        $this->sql = "SELECT id_log_acao FROM voip.log_acao where nome_log_acao =
_utf8'$nome_acao' LIMIT 1";
        $this->consulta = $this->mysql->query($this->sql);
        if($this->consulta)
            return @mysql_result($this->consulta,0);
        else
            exit("Atenção: Erro no MySQL. A operação foi cancelada!");
    }
}

```

usuario/application/include/TemplateHandler.php

```

<?php
require_once(dirname(__FILE__) . '/../config/general.php');
require_once(dirname(__FILE__) . '/Auth.php');

/**
 * Template functions
 * @author Jonatan
 *
 */
class TemplateHandler {

    private $global_file;
    private $header_file;
    private $body_file;
    private $footer_file;
    private $content_file;
    private $navigation_file;
    private $title;
    private $css_files;
    private $js_files;
    private $active_nav_item;
    private $authenticated_only;
    private $authentication_list;

    public function __construct() {
        $this->global_file = DEFAULT_GLOBAL_FILE;
        $this->header_file = DEFAULT_HEADER_FILE;
        $this->body_file = DEFAULT_BODY_FILE;
        $this->footer_file = DEFAULT_FOOTER_FILE;
        $this->navigation_file = DEFAULT_NAVIGATION_FILE;
        $this->title = DEFAULT_TITLE;
        $this->css_files = array();
        $this->js_files = array();
        $this->active_nav_item = 0;
        $this->authenticated_only = false;
        $this->authentication_list = array();
    }

    public function show() {
        if ($this->is_authenticated_only() && !Auth::check($this->authentication_list)) {
            $this->body_file = DEFAULT_FORBIDDEN_FILE;
        }
    }
}

```

```

    if ($this->template_file_exists($this->global_file)) {
        include(APPLICATION_PATH . '/template/' . $this->global_file);
    }
    else {
        throw new Exception('Global template not found.');
```

```
    }
}
```

```
public function show_header() {
    if ($this->template_file_exists($this->header_file)) {
        include (APPLICATION_PATH . '/template/' . $this->header_file);
    }
}
```

```
public function show_body() {
    if ($this->template_file_exists($this->body_file)) {
        include (APPLICATION_PATH . '/template/' . $this->body_file);
    }
}
```

```
public function show_footer() {
    if ($this->template_file_exists($this->footer_file)) {
        include (APPLICATION_PATH . '/template/' . $this->footer_file);
    }
}
```

```
public function show_navigation() {
    if ($this->template_file_exists($this->navigation_file)) {
        include (APPLICATION_PATH . '/template/' . $this->navigation_file);
    }
}
```

```
public function show_content() {
    if ($this->public_file_exists($this->left_content_file)) {
        include (PUBLIC_PATH . '/' . $this->left_content_file);
    }
}
```

```
public function import_css() {
    if ($this->css_files) {
        $internal_css_files = array();
        foreach ($this->css_files as $css_file) {
            if ($this->is_external($css_file)) {
                echo '<link rel="stylesheet" type="text/css" href="'. $css_file . '" />';
            }
            else {
                $internal_css_files[] = $css_file;
                echo '<link rel="stylesheet" type="text/css" href="'. URI_CSS . $css_file . '" />';
            }
        }
    }
}
```

```
public function import_js() {
    if ($this->js_files) {
        $internal_js_files = array();
        foreach ($this->js_files as $js_file) {
            if ($this->is_external($js_file)) {
```

```

        echo '<script type="text/javascript" src="' . $js_file . '"></script>';
    }
    else {
        $internal_js_files[] = $js_file;
        echo '<script type="text/javascript" src="' . URI_JS . $js_file . '"></script>';
    }
}
}
}

public function is_external($file) {
    if (substr($file, 0, 4) == 'http' || substr($file, 0, 3) == 'www') {
        return true;
    }
    return false;
}

private function template_file_exists($file) {
    if ($file) {
        return file_exists(APPLICATION_PATH . '/template/' . $file);
    }
    return false;
}

private function public_file_exists($file) {
    if ($file) {
        return file_exists(PUBLIC_PATH . '/' . $file);
    }
    return false;
}

public function get_header_file() {
    return $this->header_file;
}

public function set_header_file($header_file) {
    if ($this->template_file_exists($header_file)) {
        $this->header_file = $header_file;
    }
    else {
        throw new Exception('Header file does not exists.');
```



```
        return $this->footer_file;
    }

    public function set_footer_file($footer_file) {
        if ($this->template_file_exists($footer_file)) {
            $this->footer_file = $footer_file;
        }
        else {
            throw new Exception("Footer file ($footer_file) does not exists.");
        }
    }

    public function get_content_file() {
        return $this->left_content_file;
    }

    public function set_content_file($content_file) {
        if ($this->public_file_exists($content_file)) {
            $this->left_content_file = $content_file;
        }
        else {
            throw new Exception("Content file ($content_file) does not exists.");
        }
    }

    public function get_navigation_file() {
        return $this->navigation_file;
    }

    public function set_navigation_file($navigation_file) {
        if ($this->template_file_exists($navigation_file)) {
            $this->navigation_file = $navigation_file;
        }
        else {
            throw new Exception("Navigation file ($navigation_file) does not exists.");
        }
    }

    public function get_title() {
        return $this->title;
    }

    public function set_title($title) {
        $this->title = $title . ' - ' . DEFAULT_TITLE;
    }

    public function get_css_files() {
        return $this->css_files;
    }

    public function set_css_files($css_files) {
        if (is_array($css_files)) {
            $this->css_files = $css_files;
        }
        else {
            throw new Exception('CSS files must be an array.');
```

```
public function add_js_file($js_file) {
    $this->js_files[] = $js_file;
}

public function add_css_file($css_file) {
    $this->css_files[] = $css_file;
}

public function get_js_files() {
    return $this->js_files;
}

public function set_js_files($js_files) {
    if (is_array($js_files)) {
        $this->js_files = $js_files;
    }
    else {
        throw new Exception('JS files must be an array.');
```

```
usuario/application/scripts/click2call_cancel.php
```

```
<?php
require_once(dirname(__FILE__) . '/../config/general.php');
require_once(dirname(__FILE__) . '/../include/php-sip/PhpSIP.class.php');

function cancel($argv) {
    $from = isset($argv[1]) ? $argv[1] : '';
    $to = isset($argv[2]) ? $argv[2] : '';

    if (substr($to, -8, 4) == PREFIXO) {
        $to = substr($to, -4, 4);
    }

    $from_sip = "sip:$from@" . REALM;
    $to_sip = "sip:$to@" . REALM;

    $api = new PhpSIP(SERVER_IP, 5);
    $api->setDebug(false);

    $api->setUsername($from);

    $api->setMethod('CANCEL');
    $api->setFrom('sip:'. $from .'@'. $api->getSrcIp());
    $api->setUri($from_sip);

    $api->setVia($_SESSION['via']);
    $api->setFromTag($_SESSION['from_tag']);
    $api->setCallId($_SESSION['call_id']);
    $api->setCSeq($_SESSION['cseq']+1);

    $res = $api->send();
}

function write_log($text) {
    $fp = fopen(dirname(__FILE__) . '/error.log', 'a');
    if ($fp) {
        $text = date('Y-m-d h:i:s') . ' - CANCEL - ' . $text . "\n";
        fwrite($fp, $text);
        fclose($fp);
    }
}

try {
    cancel($argv);
}
catch (Exception $e) {
    write_log($e->getMessage());
    if (strstr($e->getMessage(), 'Failed to bind') !== false || strstr($e->getMessage(), 'Failed to open')
    !== false) {
        $lock_file = rtrim(sys_get_temp_dir(), DIRECTORY_SEPARATOR) . DIRECTORY_SEPARATOR . 'phpSIP.lock';
        @unlink($lock_file);
    }
    cancel($argv);
}
```

```
usuario/application/scripts/click2call_dial.php
```

```
<?php
```

```

require_once(dirname(__FILE__) . '/../config/general.php');
require_once(dirname(__FILE__) . '/../include/php-sip/PhpSIP.class.php');

function dial($argv) {
    $from = isset($argv[1]) ? $argv[1] : '';
    $to = isset($argv[2]) ? $argv[2] : '';
    $password = isset($argv[3]) ? $argv[3] : '';

    if (substr($to, -8, 4) == PREFIXO) {
        $to = substr($to, -4, 4);
    }

    $from_sip = "sip:$from@" . REALM;
    $to_sip = "sip:$to@" . REALM;

    $api = new PhpSIP(SERVER_IP);

    $api->setUsername($from);
    $api->setPasswordMd5($password);

    $api->addHeader('Subject: click2call');
    $api->setMethod('INVITE');
    $api->setFrom('sip:'.$from.'@'.$api->getSrcIp());
    $api->setUri($from_sip);

    $res = $api->send();

    if ($res == 200) {
        $api->setMethod('REFER');
        $api->addHeader('Refer-to: '.$to_sip);
        $api->addHeader('Referred-By: sip:'.$from.'@'.$api->getSrcIp());
        $api->send();

        $api->setMethod('BYE');
        $api->send();

        $api->listen('NOTIFY');
        $api->reply(481, 'Call Leg/Transaction Does Not Exist');
    }

    if (strstr($res, 'No final response') !== false) {
        $api->setMethod('CANCEL');
        $res = $api->send();
    }
}

function write_log($text) {
    $fp = fopen(dirname(__FILE__) . '/error.log', 'a');
    if ($fp) {
        $text = date('Y-m-d h:i:s') . ' - DIAL - ' . $text . "\n";
        fwrite($fp, $text);
        fclose($fp);
    }
}

try {
    dial($argv);
}

```

```

catch (Exception $e) {
    write_log($e->getMessage());
    if (strstr($e->getMessage(), 'Failed to bind') !== false || strstr($e->getMessage(), 'Failed to
open') !== false) {
        $lock_file =
rtrim(sys_get_temp_dir(), DIRECTORY_SEPARATOR).DIRECTORY_SEPARATOR.'phpSIP.lock';
        @unlink($lock_file);
    }
    dial($argv);
}

```

usuario/application/scripts/recuperacao_dial.php

```

<?php
require_once(dirname(__FILE__) . '/../config/general.php');
require_once(dirname(__FILE__) . '/../include/php-sip/PhpSIP.class.php');
require_once(dirname(__FILE__) . '/../include/DB.php');

function dial($argv) {
    $from = isset($argv[1]) ? $argv[1] : '';
    $to = isset($argv[2]) ? $argv[2] : '';
    $password = isset($argv[3]) ? $argv[3] : '';

    if (substr($to, -8, 4) == PREFIXO) {
        $to = substr($to, -4, 4);
    }

    $DB = DB::getInstance();
    $query = sprintf("SELECT `status`
                    FROM usuarios_recuperacao_senha
                    WHERE `ramal`='%s'",
                    $from);

    try {
        $response = $DB->query($query);
    }
    catch (Exception $e) {
        write_log($e->getMessage());
    }

    if (mysql_num_rows($response) == 0) {
        write_log('Status unknown');
        exit;
    }

    if (mysql_result($response, 'status') != 1) {
        write_log('Status != 1');
        exit;
    }

    $from_sip = "sip:$from@" . REALM;
    $to_sip = "sip:$to@" . REALM;

    $api = new PhpSIP(SERVER_IP);

    $api->setUsername($from);
    $api->setPasswordMd5($password);

    $api->addHeader('Subject: click2call');

```

```

    $api->setMethod('INVITE');
    $api->setFrom('sip:'.$from.'@'.$api->getSrcIp());
    $api->setUri($from_sip);

    $res = $api->send();

    if ($res == 200) {
        $api->setMethod('REFER');
        $api->addHeader('Refer-to: '.$to_sip);
        $api->addHeader('Referred-By: sip:'.$from.'@'.$api->getSrcIp());
        $api->send();

        $query = sprintf("UPDATE usuarios_recuperacao_senha
                        SET `status`=2
                        WHERE `ramal`='%s'",
                        $from);

        try {
            $DB->query($query);
        }
        catch (Exception $e) {
            write_log($e->getMessage());
        }

        $api->setMethod('BYE');
        $api->send();

        $api->listen('NOTIFY');
        $api->reply(481, 'Call Leg/Transaction Does Not Exist');
    }

    if (strstr($res, 'No final response') !== false) {
        $api->setMethod('CANCEL');
        $res = $api->send();
    }
}

function write_log($text) {
    $fp = fopen(dirname(__FILE__) . '/error.log', 'a');
    if ($fp) {
        $text = date('Y-m-d H:i:s') . ' - DIAL - ' . $text . "\n";
        fwrite($fp, $text);
        fclose($fp);
    }
}

try {
    dial($argv);
}
catch (Exception $e) {
    write_log($e->getMessage());
    if (strstr($e->getMessage(), 'Failed to bind') !== false || strstr($e->getMessage(), 'Failed to
open') !== false) {
        $lock_file =
rtrim(sys_get_temp_dir(), DIRECTORY_SEPARATOR).DIRECTORY_SEPARATOR.'phpSIP.lock';
        @unlink($lock_file);
    }
    dial($argv);
}

```

```

usuario/application/template/body.php
<div id="left_side">
    <?php $this->show_navigation(); ?>
</div>
<div id="right_side">
    <?php $this->show_content(); ?>
</div>
<div class="clear"></div>

```

```

usuario/application/template/footer.php
<div id="footer_left">
    <br />Seu IP: <?php echo $_SERVER["REMOTE_ADDR"]; ?></p>
</div>
<div id="footer_right">
    <a href="http://www.rnp.br">
        
    </a>
    <a href="http://www.ufsc.br" style="margin-left: 20px">
        
    </a>
</div>
<div class="clear"></div>

```

```

usuario/application/template/forbidden.php
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS ?>public.css" />

<div class="center">
    
    <h1>Você precisa estar logado para acessar esta página</h1>
    <em>Sua sessão pode ter expirado. Utilize o formulário de login abaixo para entrar novamente.</em>
</div>
<?php
require_once(PUBLIC_PATH . '/login/login.php');
?>

```

```

usuario/application/template/global.php
<?php
ob_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title><?php echo $this->get_title(); ?></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>global.css" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>webphone.css" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>colorbox.css" />
<!--[if IE]>
<style type="text/css">
.menu_options {
    border-bottom: 1px solid #DBDBDB;
}

```

```

</style>
<![endif]-->
<?php
$this->import_css();
?>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery.qtip-1.0.min.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery.colorbox-min.js"></script>
<?php
$this->import_js();
?>
<script type="text/javascript">
$(function() {
    $('input[title]').not('#search').each(function() {
        if ($(this).val() === '') {
            $(this).val($(this).attr('title')).addClass('placeholder');
        }
        if ($(this).val() === $(this).attr('title')) {
            $(this).addClass('placeholder');
        }
        $(this).focus(function() {
            if ($(this).val() === $(this).attr('title')) {
                $(this).val('').removeClass('placeholder');
            }
        });
        $(this).blur(function() {
            if ($(this).val() === '') {
                $(this).val($(this).attr('title')).addClass('placeholder');
            }
        });
    });
});
</script>
</head>
<body>
<div id="main">
    <div id="header">
        <div id="header_content">
            <?php $this->show_header(); ?>
        </div>
    </div>
    <div id="body">
        <?php $this->show_body(); ?>
    </div>
</div>
<div id="footer">
    <div id="footer_content">
        <?php $this->show_footer(); ?>
    </div>
</div>
</body>
</html>
<?php ob_end_flush(); ?>

```

```

usuario/application/template/header.php

```

```

<h1><a href="<?php echo URI_RAIZ; ?>">Portal do Usuário</a></h1>
<span>BETA</span>

```



```

<p><?php echo date('d') . ' de ' . get_month_name(date('m')) . ' de ' . date('Y'); ?></p>
<?php
require_once(dirname(__FILE__) . '/../include/Auth.php');
$auth = new Auth();
if ($auth->check()) {
?>
<div class="clear"></div>
<div id="logout">
    <form method="post" action="<?php echo URI_LOGIN; ?>?action=logout">
        Logado no ramal <b><?php echo $_SESSION['ramal']; ?></b>
        <input type="submit" value="Sair" />
    </form>
</div>
<?php
}
?>

```

usuario/application/template/navigation.php

```

<?php
require_once(dirname(__FILE__) . '/../include/Auth.php');
require_once(dirname(__FILE__) . '/../include/DB.php');
if (Auth::check()) {
    $DB = DB::getInstance();
    $DB->connect();
    echo '<div class="menu_box">
        <h4 class="menu_header">Ramal</h4>
        <ul class="menu_options">';

    //Painel
    if ($this->get_active_nav_item() == ID_PAINEL) {
        echo '<li class="active">';
    } else {
        echo '<li>';
    }
    echo '<a href="' . URI_PAINEL . '">Painel</a>';
    echo '</li>';

    //Configurações
    if ($this->get_active_nav_item() == ID_CONFIGURACOES) {
        echo '<li class="active">';
    } else {
        echo '<li>';
    }
    echo '<a href="' . URI_CONFIGURACOES . '">Configurações</a>';
    echo '</li>';

    //Fatura
    if ($this->get_active_nav_item() == ID_FATURA) {
        echo '<li class="active">';
    } else {
        echo '<li>';
    }
    echo '<a href="' . URI_FATURA . '">Fatura</a>';
    echo '</li>';

    //Contatos
    if ($this->get_active_nav_item() == ID_CONTATOS) {

```

```

        echo '<li class="active">';
    } else {
        echo '<li>';
    }
    echo '<a href="' . URI_CONTATOS . '">Contatos</a>';
    echo '</li>';

    echo '</ul></div>';
    ?>

<h4>Contatos</h4>
<form id="search_form" method="get" action="<?php echo URI_RAIZ; ?>usuario_contatos.php">
    <div id="search_container"><input type="text" id="search" name="search" tabindex="1"
accesskey="f" title="Procurar contatos" />
        <button title="Procurar na lista de contatos" type="submit" id="search_button">
            
        </button>
    </div>
</form>
<div id="contatos_container"></div>

<div id="webphone_container">
    <h4>Webphone</h4>
    <input type="text" id="phone_number" />
    <table id="numeric_pad" cellspacing="1">
        <tr>
            <td><a href="javascript:BtnValue('btn_1')" id="btn_1"><span>1</span>&nbsp;</a></td>
            <td><a href="javascript:BtnValue('btn_2')" id="btn_2"><span>2</span>ABC</a></td>
            <td><a href="javascript:BtnValue('btn_3')" id="btn_3"><span>3</span>DEF</a></td>
        </tr>
        <tr>
            <td><a href="javascript:BtnValue('btn_4')" id="btn_4"><span>4</span>JHI</a></td>
            <td><a href="javascript:BtnValue('btn_5')" id="btn_5"><span>5</span>JKL</a></td>
            <td><a href="javascript:BtnValue('btn_6')" id="btn_6"><span>6</span>MNO</a></td>
        </tr>
        <tr>
            <td><a href="javascript:BtnValue('btn_7')" id="btn_7"><span>7</span>PQRS</a></td>
            <td><a href="javascript:BtnValue('btn_8')" id="btn_8"><span>8</span>TUV</a></td>
            <td><a href="javascript:BtnValue('btn_9')" id="btn_9"><span>9</span>WXYZ</a></td>
        </tr>
        <tr>
            <td><a href="javascript:BtnValue('btn_10')" id="btn_10"><span>*</span>&nbsp;</a></td>
            <td><a href="javascript:BtnValue('btn_0')" id="btn_0"><span>0</span>&nbsp;</a></td>
            <td><a href="javascript:BtnValue('btn_11')" id="btn_11"><span>#</span>&nbsp;</a></td>
        </tr>
    </table>

    <div id="webphone_controls">
        <a href="javascript:voipAudioDevice()" id="speaker" title="Configuração de
Som"></a>
        <a href="javascript:voipTransfer()" id="btn_transfer" title="Transferir chamada"
class="inactive"></a>
        <a href="javascript:voipHold()" id="btn_hold" title="Colocar em espera"
class="inactive"></a>
        <a href="javascript:voipRedial()" id="btn_redial" title="Re-discar"
class="inactive"></a>
    </div>

```

```

<!--         <ul id="controls">-->
<!--         <li><a href="javascript:voipAudioDevice()" id="speaker">Volume</a></li>-->
<!--         <li><a href="javascript:voipTransfer()" id="btn_transfer">Trasferência</a></li>-->
<!--         <li><a href="javascript:voipHold()" id="btn_hold">Espera</a></li>-->
<!--         <li><a href="javascript:voipRedial()" id="btn_redial">Re-discar</a></li>-->
<!--         </ul>-->

        <div id="webphone_buttons">
            <div id="callbuttons">
                <input type="button" value="Discar" id="btn_call" onclick="voipCall()"
class="disabled" disabled="disabled" />
                <input type="button" value="Cancelar" id="btn_hangup"
onclick="voipHangup(<?php echo $_SESSION['ramal']; ?>)" class="disabled" disabled="disabled" />
            </div>
            <div id="acceptreject">
                <input type="button" value="Aceitar" id="btn_accept"
onclick="voipAccept()" />
                <input type="button" value="Rejeitar" id="btn_reject" onclick="voipReject()" />
            </div>
        </div>
        <div id="webphone_messages">
            <div id="webphone_status" onclick="voipStatusRegister()"></div>
            <div id="webphone_events"></div>
            <div id="webphone_dtmf"></div>
        </div>
</div>

<script type="text/javascript">
    var serverAddress = '<?php echo REALM; ?>'; // your VoIP server IP address or domain name

    $(function() {
        $('#search_form').ajaxForm(function(response) {
            $('#contatos_container').html(response);
        });

        $('#search').focus();
        $('#search').keyup(function(event) {
            if (event.keyCode == 8 && $(this).val() == '') {
                $('#search_form').submit();
            }
        });

        $("#contatos_container").load('<?php echo URI_RAIZ; ?>usuario_contatos.php');
        var refreshId = setInterval(function() {
            //$("#contatos_container").load('<?php echo URI_RAIZ; ?>usuario_contatos.php?randval='+
Math.random());
            $('#search_form').submit();
        }, 10000);
    });
</script>
    <?php
    if ($_SESSION['webphone'] == 'java') {
    ?>
    <script type="text/javascript" src="<?php echo URI_JS; ?>deployJava.js"></script>
    <script type="text/javascript"></script>
    <script type="text/javascript" src="<?php echo URI_JS; ?>webphone_java.js"></script>
    <script type="text/javascript">
    var attributes = {

```

```

    id: 'webphone',
    code: 'webphone.webphone.class',
    name: 'webphone',
    archive: '../webphone.jar',
    codebase: '.',
    width: 1,
    height: 1,
    MAYSCRIPT: true
};

var parameters = {
    serveraddress: '' + serverAddress + '',
    JAVA_CODEBASE: '.',
    mayscript: 'yes',
    scriptable: 'true',
    jsscriptevent: '2',
    autocfgsave: '0',
    loglevel: '0',
    hasincomingcall: 'false',
    webphonetojs: 'handleNotifications'
//     username: '<?php echo $_SESSION['ramal']; ?>',
//     md5: '<?php echo $_SESSION['senha']; ?>',
//     register: 'true'
};

deployJava.runApplet(attributes, parameters, '1.4');

$(function() {
    webphone_init('<?php echo $_SESSION['ramal']; ?>', '', '<?php echo $_SESSION['senha']; ?>');
});
</script>
<?php
}
elseif ($_SESSION['webphone'] == 'click2call') {
?>
<script type="text/javascript">
$(function() {
    <?php
    $query = sprintf('SELECT contact
        FROM voip.location
        WHERE username = %d',
        mysql_real_escape_string($_SESSION['ramal']));

    $response = $DB->query($query);

    if (mysql_num_rows($response) == 0) {
        ?>
        $('#webphone_status').html('<span class="warning">Seu ramal não está registrado na central
PBX</span>');
        <?php
        }
        else {
        ?>
        $('#btn_call').attr('disabled', false).removeClass('disabled');
        <?php
        }
        ?>
    });

```

```

</script>
<script type="text/javascript" src="<?php echo URI_JS; ?>webphone_click2call.js"></script>
<?php
}
else {
?>
<script type="text/javascript" src="<?php echo URI_JS; ?>webphone_sofphone.js"></script>
<?php
}
}
?>

```

usuario/public/_css/configuracoes.css

```

#right_side {
    padding: 0;
    width: 815px;
    min-height: 100px;
    background-color: #f7f7f7;
}

#right_side h3 {
    text-align: center;
    padding: 6px 0;
    height: 30px;
    border-radius-topleft: 10px;
    border-radius-topright: 10px;
    -moz-border-radius-topleft: 10px;
    -moz-border-radius-topright: 10px;
    -webkit-border-radius-topleft: 10px;
    -webkit-border-radius-topright: 10px;
    background: #f5f5f5;
    background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e0e0e0));
    background: -moz-linear-gradient(top, #f5f5f5, #e0e0e0);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e0e0e0');
    color: #444;
}

#right_side form {
    width: 795px;
    padding: 0 10px;
    margin-bottom: 10px;
}

#right_side form h4 {
    width: 801px;
    /*background: #eee url(../_img/add_user.png) 5px center no-repeat;*/
    background: #eee;
    padding: 7px;
    margin: -10px;
    border-top: 1px solid #ddd;
    border-bottom: 1px solid #ddd;
    margin-bottom: 5px;
}

#right_side form label {
    line-height: 2em;
}

```

```
usuario/public/_css/contatos.css

#right_side {
    padding: 0;
    width: 815px;
    min-height: 100px;
    background-color: transparent;
}

#right_side h3 {
    text-align: center;
    padding: 6px 0;
    height: 30px;
    border-radius-topleft: 10px;
    border-radius-topright: 10px;
    -moz-border-radius-topleft: 10px;
    -moz-border-radius-topright: 10px;
    -webkit-border-radius-topleft: 10px;
    -webkit-border-radius-topright: 10px;
    background: #f5f5f5;
    background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e0e0e0));
    background: -moz-linear-gradient(top, #f5f5f5, #e0e0e0);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e0e0e0');
    color: #444;
}

#adicionar_contato {
    width: 795px;
    background-color: #f5f5f5;
    padding: 0 10px;
    border-bottom: 2px solid #ddd;
    padding-bottom: 15px;
}

#adicionar_contato h4, #lista_contatos_container h4 {
    width: 780px;
    background: #eee url(../_img/add_user.png) 5px center no-repeat;
    padding: 7px 7px 7px 28px;
    margin: -10px;
    border-top: 1px solid #ddd;
    border-bottom: 1px solid #ddd;
    margin-bottom: 5px;
}

#lista_contatos_container h4 {
    margin: 0;
    background: #eee url(../_img/users.png) 5px center no-repeat;
    border-bottom: none;
}

#lista_contatos_container .dataTables_wrapper table {
    border: none;
}

#lista_contatos_container .dataTables_processing {
    margin-top: -32px;
}
```

```

#lista_contatos_container .dataTables_filter {
    margin-top: -27px;
    margin-right: 5px;
}

#lista_contatos_container .bottom {
    margin: 0 5px;
}

#lista_contatos_container .dataTables_info {
    padding: 0;
    color: #777;
    font-size: 85%;
}

#lista_contatos_container .dataTables_paginate {
    margin-right: 5px;
}

```

usuario/public/_css/global.css

```

/* Geral
*****/

* {
    margin: 0;
}

html, body {
    font-size: 0.9em;
    letter-spacing: 0.04em;
    line-height: 1.4em;
    font-family: Tahoma, "Lucida Sans Unicode", Verdana, Arial;
    background: #F4F4F4 url(../img/bg_top.png) top left repeat-x;
    color: #333;
    margin: 0;
    padding: 0;
    height: 100%;
}

img {
    border: none;
}

/* Headers
*****/

h1, h2, h3, h4, h5, h6 {
    margin: 0;
    font-weight: bold;
}

h1 {
    font-size: 1.97em;
    line-height: 1.6em;
    font-weight: normal;
}

```

```
h2 {
    font-size: 1.4em;
    line-height: 1.5em;
    font-weight: normal;
}

h3 {
    font-size: 1.1em;
    line-height: 1.4em;
}

h4 {
    font-size: 1em;
    line-height: 1.4em;
}

h5 {
    font-size: 0.9em;
    line-height: 1.3em;
}

h6 {
    font-size: 0.8em;
    line-height: 1em;
}

/* Texts
*****/

p {
    margin: 0;
    padding: 0.6em 0;
}

em {
    font-style: italic;
    font-size: 1em;
    color: #808080;
}

a {
    color: #444;
    text-decoration: none;
}

a:hover {
    color: #000;
    text-decoration: underline;
}

/* Forms
*****/

label {
    font-weight: bold;
}

label.normal {
```



```
        font-weight: normal;
    }

fieldset {
    padding: 0 1.4em 1.4em 1.4em;
    margin: 0 0 .7em 0;
    border: 1px solid #ccc;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
}

legend {
    font-weight: bold;
    font-size: 1.1em;
    margin-bottom: 5px;
}

input[type=text], input[type=password],
input.text, input.title,
textarea, select {
    background-color: #fff;
    border: 1px solid #999;
    color: #444;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
    padding: 2px 3px;
}

input[type=text]:focus, input[type=password]:focus,
input.text:focus, input.title:focus,
textarea:focus, select:focus {
    border-color: #222;
}

/*input[type=text],*/
input.text, input.title,
textarea, select, label {
    margin: 0.15em 0;
}

input.text, select.text, input.title, textarea {
    width: 200px; padding: 5px;
}

input.title {
    font-size: 1em;
}

input.placeholder {
    font-style: italic;
    color: #777;
}

textarea {
    width: 232px;
    height: 100px;
}

select.text {
    width: 208px;
```

```

}

input[type=checkbox], input[type=radio],
input.checkbox, input.radio {
    position: relative;
    top: .25em;
}

input[type=button], input[type=submit], button {
    background-color: #557030;
    color: white;
    padding: 3px 10px;
    /*margin-left: 10px;*/
    border-bottom: 1px solid #36471D;
    border-right: 1px solid #36471D;
    border-top: 1px solid #6C8F3D;
    border-left: 1px solid #6C8F3D;
    cursor: pointer;
    min-width: 40px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
}

input[type=button].disabled, input[type=submit].disabled, button.disabled {
    color: #ddd;
    opacity: 0.6;
    filter:alpha(opacity=60);
    -khtml-opacity: 0.6;
    -moz-opacity: 0.6;
    cursor: default;
}

input[type=text].disabled {
    background-color: #eee;
}

form.inline { }
form.inline p {
    margin: 0;
    padding: 4px 0;
    float: left;
    margin-right: 20px;
    min-height: 40px;
}

.form_table td {
    padding: 2px;
}

.form_table td:first-child, .form_table td.label {
    text-align: right;
    vertical-align: top;
    padding-top: 5px;
}

/* Containers
******/

```

```
#header {
    width: 100%;
    height: 35px;
    background-color: #557030;
    position: absolute;
    top: 0;
    left: 0;
    /*background-color: #165189;*/
}

#header_content {
    width: 990px;
    margin: 0 auto;
    padding: 4px 0;
    color: #E5E5E5;
}

#header_content h1 {
    line-height: 1;
    font-size: 1.85em;
    float: left;
    margin-right: 7px;
}

#header_content span {
    font-size: 80%;
}

#header_content p {
    float: right;
    margin-top: -4px;
}

#header_content ul {
    margin: 0;
    padding: 0;
    list-style-type: none;
    margin-top: 4px;
}

#header_content ul li {
    display: inline;
    padding: 2px 13px;
}

#header_content .active {
    background-color: #6C8F3D;
    /*background-color: #1E70B9;*/
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    font-weight: bold;
    color: #FFF;
}

#header_content .maquinas {
    float: left;
    margin-left: 35px;
}
```

```
        width: 300px;
    }

#header_content a {
    color: #E5E5E5;
    text-decoration: none;
}

#header_content .modulos {
    float: right;
}

#logout {
    float: right;
    background-color: #6C8F3D;
    margin-top: 1px;
    padding: 2px 10px;
    min-width: 175px;
    font-size: 85%;

    border-radius-bottomleft: 10px;
    border-radius-bottomright: 10px;
    -moz-border-radius-bottomleft: 10px;
    -moz-border-radius-bottomright: 10px;
    -webkit-border-radius-bottomleft: 10px;
    -webkit-border-radius-bottomright: 10px;
}

#logout input[type=submit] {
    padding: 1px 10px;
    float: right;
    font-size: 100%;
}

#main {
    width: 990px;
    margin: 0 auto -85px;
    min-height: 100%;
    height: auto !important;
    height: 100%;
}

#body {
    padding: 65px 0 80px 0;
}

#left_side {
    float: left;
    width: 160px;
}

#right_side {
    float: right;
    background-color: #FFFFFFE;
    border: 1px solid #DBDBDB;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    width: 790px;
}
```

```
        min-height: 547px;
        padding: 10px;
    }

#footer {
    background: url(../_img/bg_footer.png) bottom left repeat-x;
    border-bottom: 5px solid #557030;
    padding: 20px 0;
    width: 100%;
    height: 40px;
    clear: both;
}

#footer_content {
    width: 990px;
    margin: 0 auto;
}

#footer_left {
    float: left;
    width: 200px;
    font-size: 80%;
}

#footer_right {
    float: right;
    text-align: right;
}

/* Menu
*****/

.menu_box {
    background-color: #EBEBEB;
    margin-bottom: 20px;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
}

.menu_header {
    background: #557030 url(../_img/bg_menu_header.png) top left repeat-x;
    /*background-color: #165189;*/
    color: #EDEDDE;
    margin: 0;
    padding: 6px 8px;
    border-radius-topleft: 10px;
    border-radius-topright: 10px;
    -moz-border-radius-topleft: 10px;
    -moz-border-radius-topright: 10px;
    -webkit-border-radius-topleft: 10px;
    -webkit-border-radius-topright: 10px;
}

.menu_options {
    margin: 0;
    padding: 0;
    list-style-type: none;
}
```

```
}

.menu_options li {
  padding: 4px 8px;
  border-left: 1px solid #DBDBDB;
  border-right: 1px solid #DBDBDB;
  border-top: 1px solid #E5E5E5;
}

.menu_options li:last-child {
  border-bottom: 1px solid #DBDBDB;
  border-radius-bottomleft: 10px;
  border-radius-bottomright: 10px;
  -moz-border-radius-bottomleft: 10px;
  -moz-border-radius-bottomright: 10px;
  -webkit-border-radius-bottomleft: 10px;
  -webkit-border-radius-bottomright: 10px;
}

.menu_options a {
  color: #666;
  text-decoration: none;
  display: block;
}

#left_side .active a {
  font-weight: bold;
  color: #333;
}

.menu_options li:hover, #left_side .active {
  background-color: #F4F4F4;
}

/* Misc
*****/

.loading {
  padding: 10px;
  background: url(../_img/loading.gif) center center no-repeat;
}

.clear {
  clear: both;
  padding: 0px;
  margin: 0px;
  font-size: 1px;
  height: 0px;
}

.left {
  float: left;
}

.right {
  float: right;
}
```

```
.center {
    text-align: center;
}

.small {
    margin: 0;
    font-size: 10px;
    color: #B5B5B5;
}

.regular {
    font-size: 100%;
    text-decoration: none;
    font-weight: normal;
}

hr {
    border-top: 1px solid #E5E5E5;
    border-bottom: none;
    color: #fff;
    background-color: #fff;
    height: 1px;
}

.dashed_box {
    border: #E5E5E5 1px dashed;
    padding: 8px;
    margin: 10px 0;
    background-color: #FCFCFC;
}

.error_box {
    background-color: #FFF4F4;
    color: #8a1f11;
    padding: 5px;
    margin: 6px 0 3px 0;
    border: 2px solid #C41300;
}

.warning_box {
    background-color: #FFFCC;
    color: #000;
    padding: 5px;
    margin: 6px 0 3px 0;
    border: 2px solid #FDF053;
}

.success_box {
    background-color: #DDE9CD;
    color: #172F00;
    padding: 5px;
    margin: 6px 0 3px 0;
    border: 2px solid #6C8F3D;
}

input.warningBox {
    background-color: #FFF4F4;
    border: 2px solid #C41300;
}
```

```

}

.rounded {
    background-color: #FFFFFFE;
    border: 1px solid #DBBBDB;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    padding: 5px;
    margin-bottom: 10px;
}

.error {
    padding: 2px 2px 2px 22px;
    background: url(../_img/error.png) center left no-repeat;
}

.warning {
    padding: 2px 2px 2px 22px;
    background: url(../_img/warning.png) center left no-repeat;
}

.success {
    padding: 2px 2px 2px 22px;
    background: url(../_img/ok.png) center left no-repeat;
}

/* Tabelas
*****/

.tabela_lista {
    margin: -5px auto;
    border-collapse: separate;
    border-spacing: 2px;
    color: #333333;
    width: 100%;
}

.tabela_lista tr td {
    padding: 0 7px;
}

.tabela_lista thead {
    font-weight: bold;
    height: 30px;
    text-align: left;
}

.tabela_lista tbody div.dashed_box {
    display: none;
    color: #666;
    width: 100%;
}

#lista_ramais_atualizados thead {
    background-color: #ddd;
}

```



```
#lista_ramais_atualizados tr:nth-child(even) {
    background-color: #eee;
}

/* Paginação
*****/

.pagination-wrapper {
    margin: 10px 0 4px 0;
    text-align: center;
}

.pagination {
    margin: 0;
    padding: 0;
}

.pagination li {
    border: 0;
    padding: 0;
    list-style: none;
    margin: 0 2px;
    display: inline;
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
}

.pagination a {
    border: solid 1px #555;
    margin: 0;
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
}

.pagination .previous-off, .pagination .next-off {
    border: none;
    color: #888888;
    font-weight: bold;
    margin-right: 2px;
    padding: 2px 3px;
}

.pagination .active_page {
    background: #557030;
    color: #FFFFFF;
    font-weight: bold;
    padding: 3px 0px;
}

.pagination .active_page a:link, .pagination .active_page a:visited {
    color: #fff;
}

.pagination a:link, .pagination a:visited {
    color: #000;
    padding: 2px 5px;
}
```

```

    text-decoration: none;
}

.pagination .next a, .pagination .previous a {
    font-weight: bold;
    padding: 2px 3px;
    border: none;
}

.pagination .next a:hover, .pagination .previous a:hover {
    border: none;
}

.pagination a:hover {
    border: solid 1px #444
}

/* Menu de contatos
*****/

#contatos_container {
    position: relative;
}

#contatos ul {
    margin: 0;
    padding: 0;
    list-style: none;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    font-size: 85%;
}

#contatos ul li {
    margin: 0;
    padding: 3px 0;
}

#contatos ul li a {
    display: block;
}

#contatos ul li a:hover {
    text-decoration: none;
}

#contatos ul li.phone_available, .phone_available {
    background: url(../_img/phone_available.png) 3px center no-repeat;
    padding-left: 25px;
}

#contatos ul li.phone_busy, .phone_busy {
    background: url(../_img/phone_busy.png) 3px center no-repeat;
    padding-left: 25px;
}

#contatos ul li.phone_disabled, .phone_disabled {

```

```
background: url(../_img/phone_disabled.png) 3px center no-repeat;
padding-left: 25px;
}

#contatos ul li.phone_pbx, .phone_pbx {
background: url(../_img/phone_pbx.png) 3px center no-repeat;
padding-left: 25px;
}

#contatos ul li.phone_mobile, .phone_mobile {
background: url(../_img/phone_mobile.png) 3px center no-repeat;
padding-left: 25px;
}

#contatos ul li.phone_available:hover,
#contatos ul li.phone_busy:hover,
#contatos ul li.phone_disabled:hover,
#contatos ul li.phone_pbx:hover,
#contatos ul li.phone_mobile:hover {
background-color: #eee;
}

#search_form #search_container {
border: 1px solid #777;
background-color: #fff;
-moz-border-radius: 6px;
-webkit-border-radius: 6px;
border-radius: 6px;
}

#search_form #search_container:hover {
border: 1px solid #222;
}

#search_form #search {
background-color: transparent;
border: none;
padding: 4px 0px 3px 5px;
width: 123px;
margin: 0;
vertical-align: top;
}

#search_button {
border: none;
background-color: transparent;
cursor: pointer;
font-size: x-small;
margin: 3px 3px 0 0;
padding: 0;
min-width: 20px;
}

.opcoes_contato {
width: 100%;
background-color: #eee;
margin: 10px -5px -5px -5px;
```

```
        border-top: 3px solid #d3d3d3;
        padding: 5px
    }

.status_info {
    position: absolute;
    top: 5px;
    left: 5px;
    background-color: #eee;
    border: 2px solid #ccc;
    padding: 3px 3px 3px 25px;
    height: 20px;
    border-radius-bottomright: 7px;
    -moz-border-radius-bottomright: 7px;
    -webkit-border-radius-bottomright: 7px;
    white-space: nowrap;
}

.chat {
    background: url(../_img/chat.png) center left no-repeat;
    padding-left: 21px;
    margin: 0;
}

.dial {
    background: url(../_img/dial.png) center left no-repeat;
    padding-left: 21px;
    margin: 0;
}

.mail {
    background: url(../_img/mail.png) center left no-repeat;
    padding-left: 21px;
    margin: 0;
}

.sms {
    background: url(../_img/sms.png) center left no-repeat;
    padding-left: 21px;
    margin: 0;
}

.button {
    display: inline-block;
    zoom: 1; /* zoom and *display = ie7 hack for display:inline-block */
    *display: inline;
    vertical-align: baseline;
    outline: none;
    cursor: pointer;
    text-align: center;
    text-decoration: none;
    padding: .2em .5em .3em;
    -webkit-border-radius: .4em;
    -moz-border-radius: .4em;
    border-radius: .4em;
    -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.2);
    -moz-box-shadow: 0 1px 2px rgba(0,0,0,.2);
}
```

```

    box-shadow: 0 1px 2px rgba(0,0,0,.2);
}
.button:hover {
    text-decoration: none;
}
.button:active {
    position: relative;
    top: 1px;
}

.silver {
    color: #444;
    border: solid 1px #bbb;
    background: #fff;
    background: -webkit-gradient(linear, left top, left bottom, from(#fff), to(#f0f0f0));
    background: -moz-linear-gradient(top, #fff, #f0f0f0);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffff', endColorstr='#f0f0f0');
}
.silver:hover {
    border: solid 1px #999;
    background: #f5f5f5;
    background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e5e5e5));
    background: -moz-linear-gradient(top, #f5f5f5, #e5e5e5);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e5e5e5');
}
.silver:active {
    color: #999;
    background: -webkit-gradient(linear, left top, left bottom, from(#ededed), to(#fff));
    background: -moz-linear-gradient(top, #ededed, #fff);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ededed', endColorstr='#ffffff');
}

a.disabled {
    opacity: 0.5;
    -ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=50)";
    filter: alpha(opacity=50);
    color: #555;
}

a.disabled:hover {
    color: #555;
    border: solid 1px #bbb;
    background: #fff;
    background: -webkit-gradient(linear, left top, left bottom, from(#fff), to(#f0f0f0));
    background: -moz-linear-gradient(top, #fff, #f0f0f0);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffff', endColorstr='#f0f0f0');
    opacity: 0.5;
    -ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=50)";
    filter: alpha(opacity=50);
}

.qtip {
    top: 0;
    left: 0;
}

#sem_contatos {
    text-align: center;
}

```

```

margin-top: 10px;
line-height: 1.2em;
font-size: 85%;
}

```

```

usuario/public/_css/painel.css

```

```

#right_side {
    border: none;
    background-color: transparent;
    padding: 0;
    width: 815px;
    font-size: 90%;
}

.rounded h3, #correio_voz h3 {
    font-size: 1.2em;
    text-align: center;
    margin: -5px -5px 5px -5px;
    padding: 6px 0;
    border-radius-topleft: 10px;
    border-radius-topright: 10px;
    -moz-border-radius-topleft: 10px;
    -moz-border-radius-topright: 10px;
    -webkit-border-radius-topleft: 10px;
    -webkit-border-radius-topright: 10px;
    background: #f5f5f5;
    background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e4e4e4));
    background: -moz-linear-gradient(top, #f5f5f5, #e4e4e4);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e4e4e4');
    color: #444;
}

#correio_voz h3 {
    margin: 0;
    width: 560px;
    border: 1px solid #dbdbdb;
    border-bottom: none;
    font-size: 0.9em;
}

.rounded p {
    padding: 3px 2px;
}

.rounded strong {
    display: block;
}

.rounded table {
    width: 100%;
}

.rounded table th {
    text-align: left;
}

#status_left {

```

```
float: left;
width: 240px;
}

#status_right {
float: left;
width: 560px;
margin-left: 10px;
}

#dados_ramal {
width: 230px;
}

#utilizacao {
float: left;
width: 150px;
}

#status_atual {
float: left;
width: 378px;
margin-left: 8px;
}

#correio_voz, #mensagens {
float: left;
width: 550px;
}

#jplayer_playlist ul li {
position: relative;
}

.correio_voz_origem {
/*margin-left: 20px;*/
position: absolute;
left: 170px;
}

/* .correio_voz_duracao {
float: right;
}*/

.correio_voz_acoes {
float: right;
margin-right: 10px;
}

.correio_voz_acoes img {
cursor: pointer;
}

#mensagens {
margin-top: 10px;
position: relative;
}
```

```
#lista_mensagens_wrapper {
    margin: -6px;
}

#lista_mensagens_info {
    position: absolute;
    bottom: 5px;
    width: 100%;
    text-align: center;
}

#lista_mensagens_paginate {
    margin-right: 5px;
}

#enviar_sms {
    float: left;
    margin: 0 5px;
}

#discar {
    float: left;
    width: 230px;
}

#status_discagem {
    display: none;
    padding-top: 5px;
    margin-top: 5px;
}

#status_discagem iframe {
    width: 195px;
    height: 90px;
}

.avisos {
    text-align: center;
    font-size: 1.35em;
    color: #666;
    line-height: 1.6em;
}
```

usuario/public/_css/public.css

```
#right_side {
    border: none;
    background-color: transparent;
    width: 100%;
}

#login_wrapper {
    margin: 70px auto;
    width: 275px;
    padding: 15px;
    text-align: right;
}
```



```

#login_wrapper a {
    font-size: 85%;
    float: left;
    margin-left: 65px;
}

#login_wrapper #status {
    margin-top: 5px;
}

#login_wrapper .error_box , #login_wrapper .success_box {
    width: 197px;
    float: right;
    text-align: center;
}

```

usuario/public/_css/recuperacao.css

```

#status_redefinicao li {
    font-weight: bold;
}

#status_redefinicao li.disabled {
    color: #bbb;
    font-weight: normal;
}

#status_redefinicao li.completed {
    background: url(../_img/ok.png) center right no-repeat;
    padding-right: 22px;
    font-weight: normal;
}

#redefinicao_senha {
    width: 550px;
    float: left;
    margin-top: -10px;
}

#redefinicao_senha input[type=text] {
    width: 130px;
    margin-right: 10px;
}

#right_container {
    float: right;
    width: 400px;
}

#status_redefinicao {
    width: 400px;
}

#status_redefinicao h3 {
    font-size: 1.2em;
    text-align: center;
    margin: -5px -5px 5px -5px;
    padding: 6px 0;
    border-radius-topleft: 10px;
}

```

```

border-radius-topright: 10px;
-moz-border-radius-topleft: 10px;
-moz-border-radius-topright: 10px;
-webkit-border-radius-topleft: 10px;
-webkit-border-radius-topright: 10px;
background: #f5f5f5;
background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e4e4e4));
background: -moz-linear-gradient(top, #f5f5f5, #e4e4e4);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e4e4e4');
color: #444;
}

#status_redefinicao ol {
margin: 10px 0;
padding-left: 30px;
}

#status {
width: 430px;
float: right;
margin-top: 2px;
}

#cancelar {
float: right;
}

```

usuario/public/_css/webphone.css

```

#webphone_container {
margin-top: 10px;
}

#phone_number {
padding: 4px 5px;
width: 148px;
}

#numeric_pad {
border-radius: 7px;
-moz-border-radius: 7px;
-webkit-border-radius: 7px;
background-color: #ccc;
margin-top: 5px;
}

#numeric_pad td a {
display: block;
color: inherit;
padding: 3px;
}

#numeric_pad td a:hover {
text-decoration: none;
}

#numeric_pad tr:first-child td:first-child {
border-radius-topleft: 7px;
}

```

```

-moz-border-radius-topleft: 7px;
-webkit-border-radius-topleft: 7px;
}

#numeric_pad tr:first-child td:last-child {
border-radius-topright: 7px;
-moz-border-radius-topright: 7px;
-webkit-border-radius-topright: 7px;
}

#numeric_pad tr:last-child td:last-child {
border-radius-bottomright: 7px;
-moz-border-radius-bottomright: 7px;
-webkit-border-radius-bottomright: 7px;
}

#numeric_pad tr:last-child td:first-child {
border-radius-bottomleft: 7px;
-moz-border-radius-bottomleft: 7px;
-webkit-border-radius-bottomleft: 7px;
}

#numeric_pad tr:last-child td:last-child {
border-radius-bottomright: 7px;
-moz-border-radius-bottomright: 7px;
-webkit-border-radius-bottomright: 7px;
}

#numeric_pad td {
text-align: center;
vertical-align: top;
line-height: 1.3em;
font-size: .8em;
color: #999;
width: 52px;
height: 35px;
background: #f5f5f5;
background: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e4e4e4));
background: -moz-linear-gradient(top, #f5f5f5, #e4e4e4);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f5f5f5', endColorstr='#e4e4e4');
}

#numeric_pad td:hover {
background: #eee;
background: -webkit-gradient(linear, left top, left bottom, from(#eee), to(#ddd));
background: -moz-linear-gradient(top, #eee, #ddd);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#eee', endColorstr='#ddd');
}

#numeric_pad td span {
color: #222;
font-weight: bold;
display: block;
font-size: 1.3em;
}

#webphone_controls {
margin-top: -5px;
}

```

```

background: #bbb;
background: -webkit-gradient(linear, left top, left bottom, from(#d5d5d5), to(#e8e8e8));
background: -moz-linear-gradient(top, #d5d5d5, #e8e8e8);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#d5d5d5', endColorstr='#e8e8e8');
border: 1px solid #ccc;
height: 19px;
padding-top: 4px;
border-radius-bottomleft: 7px;
-moz-border-radius-bottomleft: 7px;
-webkit-border-radius-bottomleft: 7px;
border-radius-bottomright: 7px;
-moz-border-radius-bottomright: 7px;
-webkit-border-radius-bottomright: 7px;
text-align: center;
display: none;
}

#webphone_controls a {
padding: 9px 9px 4px 9px;
margin: 0;
border-radius: 7px;
-moz-border-radius: 7px;
-webkit-border-radius: 7px;
filter: progid:DXImageTransform.Microsoft.BasicImage(grayscale=1);
/* Element must "hasLayout"! */
zoom: 1;
}

#webphone_controls a.inactive {
opacity: 0.4;
}

#webphone_buttons {
margin-top: 7px;
}

#acceptreject {
display: none;
}

#webphone_messages {
height: 60px;
font-size: 85%;
}

#webphone_status {
margin-top: 5px;
}

#webphone_events, #webphone_dtmf {
margin-top: 4px;
}

#webphone_messages .success {
padding: 2px 2px 2px 20px;
background: url(../_img/bullet_success.png) center left no-repeat;
}

```

```

#webphone_messages .warning {
    padding: 2px 2px 2px 20px;
    background: url(../_img/bullet_warning.png) center left no-repeat;
}

#webphone_messages .error {
    padding: 2px 2px 2px 20px;
    background: url(../_img/bullet_error.png) center left no-repeat;
}

#webphone_messages .loading {
    padding: 2px 2px 2px 20px;
    background: url(../_img/bullet_loading.gif) center left no-repeat;
}

#webphone_events {
    display: none;
}

```

```

usuario/public/_js/jquery.loadTable.js

```

```

;(function($) {

    $.fn.loadTableOptions = [];

    $.fn.getTable = function(id) {
        var allOptions = $.fn.loadTableOptions;
        for (var i=0 ; i<allOptions.length; i++){
            if (allOptions[i].tableId == id){
                return allOptions[i].table;
            }
        }
        return null;
    };

    $.fn.getOptions = function(id) {
        var allOptions = $.fn.loadTableOptions;
        for (var i=0 ; i<allOptions.length; i++){
            if (allOptions[i].tableId == id){
                return allOptions[i];
            }
        }
        return null;
    };

    jQuery.fn.dataTableExt.oApi.fnSetFilteringDelay = function ( oSettings, iDelay ) {
        /*
        * Type:          Plugin for DataTables (www.datatables.net) JQuery plugin.
        * Name:          dataTableExt.oApi.fnSetFilteringDelay
        * Version:       2.2.1
        * Description:  Enables filtration delay for keeping the browser more
        *               responsive while searching for a longer keyword.
        * Inputs:        object:oSettings - dataTables settings object
        *               integer:iDelay - delay in miliseconds
        * Returns:       JQuery
        * Usage:         $('#example').dataTable().fnSetFilteringDelay(250);
        * Requires:     DataTables 1.6.0+
        *
        */
    };
}

```

```

* Author:      Zygimantas Berziunas (www.zygimantas.com) and Allan Jardine (v2)
* Created:     7/3/2009
* Language:    Javascript
* License:     GPL v2 or BSD 3 point style
* Contact:     zygimantas.berziunas /AT\ hotmail.com
*/
var
    _that = this,
    iDelay = (typeof iDelay == 'undefined') ? 250 : iDelay;

this.each( function ( i ) {
    $.fn.dataTableExt.iApiIndex = i;
    var
        $this = this,
        oTimerId = null,
        sPreviousSearch = null,
        anControl = $( 'input', _that.fnSettings().aanFeatures.f );

    anControl.unbind( 'keyup' ).bind( 'keyup', function() {
        var $$this = $this;

        if (sPreviousSearch === null || sPreviousSearch != anControl.val()) {
            window.clearTimeout(oTimerId);
            sPreviousSearch = anControl.val();
            oTimerId = window.setTimeout(function() {
                $.fn.dataTableExt.iApiIndex = i;
                _that.fnFilter( anControl.val() );
            }, iDelay);
        }
    });

    return this;
});

function trim(str) {
    str = str.replace(/^\s+/, '');
    for (var i = str.length - 1; i >= 0; i--) {
        if (/\s/.test(str.charAt(i))) {
            str = str.substring(0, i + 1);
            break;
        }
    }
    return str;
}

jQuery.fn.dataTableExt.oSort['date-euro-asc'] = function(a, b) {
    if (trim(a) != '') {
        var frDatea = trim(a).split(' ');
        var frTimea = frDatea[1].split(':');
        var frDatea2 = frDatea[0].split('/');
        var x = (frDatea2[2] + frDatea2[1] + frDatea2[0] + frTimea[0] + frTimea[1] +
frTimea[2]) * 1;
    } else {
        var x = 1000000000000000; // = 1'an 1000 ...
    }
}

```

```

        if (trim(b) != '') {
            var frDateb = trim(b).split(' ');
            var frTimeb = frDateb[1].split(':');
            frDateb = frDateb[0].split('/');
            var y = (frDateb[2] + frDateb[1] + frDateb[0] + frTimeb[0] + frTimeb[1] +
frTimeb[2]) * 1;
        } else {
            var y = 1000000000000000;
        }
        var z = ((x < y) ? -1 : ((x > y) ? 1 : 0));
        return z;
    };

    jQuery.fn.dataTableExt.oSort['date-euro-desc'] = function(a, b) {
        if (trim(a) != '') {
            var frDatea = trim(a).split(' ');
            var frTimea = frDatea[1].split(':');
            var frDatea2 = frDatea[0].split('/');
            var x = (frDatea2[2] + frDatea2[1] + frDatea2[0] + frTimea[0] + frTimea[1] +
frTimea[2]) * 1;
        } else {
            var x = 1000000000000000;
        }

        if (trim(b) != '') {
            var frDateb = trim(b).split(' ');
            var frTimeb = frDateb[1].split(':');
            frDateb = frDateb[0].split('/');
            var y = (frDateb[2] + frDateb[1] + frDateb[0] + frTimeb[0] + frTimeb[1] +
frTimeb[2]) * 1;
        } else {
            var y = 1000000000000000;
        }
        var z = ((x < y) ? 1 : ((x > y) ? -1 : 0));
        return z;
    };

    $.fn.loadTable = function(customOptions) {

        if (!customOptions) {
            customOptions = {};
        }

        var _options = $.fn.loadTableOptions;
        var me = this;

        function optionsClass(customOptions, me) {
            this.successMessage = (typeof customOptions.successMessage == 'undefined') ? '' :
customOptions.successMessage;
            this.editFormElement = (typeof customOptions.editFormElement == 'undefined') ?
null : customOptions.editFormElement;
            this.fileAddForm = (typeof customOptions.fileAddForm == 'undefined') ? '' :
customOptions.fileAddForm;
            this.fileEditForm = (typeof customOptions.fileEditForm == 'undefined') ? '' :
customOptions.fileEditForm;
            this.editCallback = (typeof customOptions.editCallback == 'undefined') ? null :
customOptions.editCallback;
            this.allowCreate = (typeof customOptions.allowCreate == 'undefined') ? true :

```

```

customOptions.allowCreate;
    this.allowDelete = (typeof customOptions.allowDelete == 'undefined') ? true :
customOptions.allowDelete;
    this.allowUpdate = (typeof customOptions.allowUpdate == 'undefined') ? true :
customOptions.allowUpdate;
    this.delConfirmation = (typeof customOptions.delConfirmation == 'undefined') ?
false : customOptions.delConfirmation;
    this.delConfirmationMsg = (typeof customOptions.delConfirmation == 'undefined') ?
'Tem certeza que deseja excluir o(s) registro(s) selecionados?' : customOptions.delConfirmationMsg;
    this.delSuccessMsg = (typeof customOptions.delSuccessMsg == 'undefined') ?
'Registro(s) excluído(s) com sucesso!' : customOptions.delSuccessMsg;
    this.boxHeight = (typeof customOptions.boxHeight == 'undefined') ? '' :
customOptions.boxHeight;
    this.boxWidth = (typeof customOptions.boxWidth == 'undefined') ? '' :
customOptions.boxWidth;

    //Custom table options:
    this.aoColumns = (typeof customOptions.aoColumns == 'undefined') ? [] :
customOptions.aoColumns;
    this.fnDrawCallback = (typeof customOptions.fnDrawCallback == 'undefined') ? null
: customOptions.fnDrawCallback;
    this.fnInitComplete = (typeof customOptions.fnInitComplete == 'undefined') ? null
: customOptions.fnInitComplete;
    this.sAjaxSource = (typeof customOptions.sAjaxSource == 'undefined') ? "" :
customOptions.sAjaxSource;
    this.sDom = (typeof customOptions.sDom == 'undefined') ?
'<"top"ifr>t<"bottom"p><"clear">' : customOptions.sDom;
    this.iDisplayLength = (typeof customOptions.iDisplayLength == 'undefined') ? 20 :
customOptions.iDisplayLength;
    this.aaSorting = (typeof customOptions.aaSorting == 'undefined') ? [[ 0, "asc" ]]
: customOptions.aaSorting;
    this.sPaginationType = (typeof customOptions.sPaginationType == 'undefined') ?
"full_numbers" : customOptions.sPaginationType;
    this.bServerSide = (typeof customOptions.bServerSide == 'undefined') ? true :
customOptions.bServerSide;

    this.formTableElement = me.parent();
    this.tableDiv = null;
    this.tableId = null;
    this.table = null;
    this.statusTimeout = null;
}

return this.each(function() {

    var options = new optionsClass(customOptions, me);
    options.tableId = this.getAttribute("id");
    _options.push(options);

    function resetAjaxForm() {
        if (options.delConfirmation && !confirm(options.delConfirmationMsg)) {
            return false;
        }
        options.formTableElement.find(".status_" + options.tableId).html('<span
class="loading"></span>');
    }
}

```



```

        function processDeleteError(XMLHttpRequest, textStatus, errorThrown) {
            options.formTableElement.find(".status_" +
options.tableId).html(textStatus+' '+XMLHttpRequest.responseText);
        }

        function processDeleteResponse(response, status) {
            if (response.error == null) {
                options.formTableElement.find(".status_" +
options.tableId).html('<span class="success">' + options.delSuccessMsg + '</span>');
                if (options.statusTimeout != null) {
                    clearTimeout(options.statusTimeout);
                }
                options.statusTimeout =
setTimeout("$.fn.getOptions('"+options.tableId+"').formTableElement.find('.status_" + options.tableId +
"').html(')', 8000);
            }
            else {
                options.formTableElement.find(".status_" +
options.tableId).html('<span class="error">'+response.error+'</span>');
            }
            $.fn.getTable(options.tableId).fnDraw();
            $("#" + options.tableId + " :checkbox:first").attr('checked', false);
            $('#del_button_' + options.tableId).attr('disabled','disabled');
            $('#del_button_' + options.tableId).addClass('disabled');
        }

        //Ajax form plugin: http://malsup.com/jquery/form/
        var optionsDelete = {
            beforeSend: resetAjaxForm, // pre-submit callback
            success: processDeleteResponse, // post-submit callback
            error: processDeleteError,
            dataType: 'json'
        };
        //É possível adicionar outras opções da função Ajax do JQuery
        (http://docs.jquery.com/Ajax/jquery.ajax)
    };

//
    alert(typeof(options.formTableElement));

    if (options.allowDelete == true) {
        if (options.formTableElement.is('form')) {
            options.formTableElement.ajaxForm(optionsDelete);
        }
        else {
            alert('A tabela deve estar dentro de um formulário');
        }
    }

    /* Get the rows which are currently selected */
    function fnGetSelected(oTableLocal) {
        var aReturn = new Array();
        var aTrs = oTableLocal.fnGetNodes();

        for (var i=0; i<aTrs.length; i++ ) {
            if ($(aTrs[i]).hasClass('row_selected') &&
$(aTrs[i]).is(":visible")) {
                aReturn.push(aTrs[i]);
            }
        }
    }

```

```

        return aReturn;
    }

    //Datatables plugin: http://www.datatables.net/
    var dataTableOptions = {
        "oLanguage": {
            "sProcessing": "Carregando...",
            "sLengthMenu": "Exibir _MENU_ resultados",
            "sZeroRecords": "Nenhum resultado encontrado",
            "sInfo": "Exibindo _START_ a _END_ de _TOTAL_ resultados",
            "sInfoEmpty": "Nenhum resultado",
            "sInfoFiltered": "(filtrados de _MAX_ resultados)",
            "sInfoPostFix": "",
            "sSearch": "Procurar:",
            "sUrl": "",
            "oPaginate": {
                "sFirst": "Primeira",
                "sPrevious": "Anterior",
                "sNext": "Próxima",
                "sLast": "Última"
            }
        },
        "sPaginationType": options.sPaginationType,
        "bAutoWidth": false,
        "fnDrawCallback": function() {
            bindEvents();
            if (options.fnDrawCallback) {
                options.fnDrawCallback();
            }
        },
        "bLengthChange": false,
        "bProcessing": true,
        "bServerSide": options.bServerSide,
        /*"fnServerData": function ( sSource, aoData, fnCallback ) {
            $.ajax( {
                "type": "POST",
                "url": sSource,
                "data": aoData,
                "success": function (msg) {
                    alert(msg);
                }
            } );
        }*/
        "aaSorting": options.aaSorting,
        "iDisplayLength": options.iDisplayLength,
        "sDom": options.sDom,
        "fnInitComplete": function() {

            var del_button = "<input id='del_button_"+options.tableId+"'
type='submit' value='Excluir selecionados' disabled='disabled' class='disabled' /> ";
            var add_button = "<input id='add_button_"+options.tableId+"'
type='button' value='Cadastrar' />";

            var toolbar_footer = "";

            if (options.allowDelete) {
                toolbar_footer+= del_button;
            }

            if (options.allowCreate) {

```

```

        toolbar_footer+= add_button;
    }

    options.formTableElement.find(".bottom").prepend(toolbar_footer +
"<span class='status_' + options.tableId + '></span>");

    if (options.allowCreate) {
        $('#add_button_' + options.tableId).click(function() {
            window.location = options.fileAddForm;
        });
    }
    if (options.allowDelete) {
        $("##" + options.tableId + "
:checkbox:first").click(function() {
            if ($(this).is(':checked')) {
                $("##" + options.tableId + "
:checkbox").attr("checked", true);
                tr").addClass('row_selected');
            }
            else {
                $("##" + options.tableId + "
:checkbox").attr("checked", false);
                tr").removeClass('row_selected');
            }
        });
    }

    if (options.fnInitComplete) {
        options.fnInitComplete();
    }
};

if (options.aoColumns.length) {
    $.extend(dataTableOptions, {
        "aoColumns": options.aoColumns
    });
}

if (options.sAjaxSource) {
    $.extend(dataTableOptions, {
        "sAjaxSource": options.sAjaxSource
    });
}

options.table = $('#' + options.tableId).dataTable(dataTableOptions);

$.fn.getTable(options.tableId).fnSetFilteringDelay(300);

function bindEvents() {
    if (options.allowDelete) {
        $("##" + options.tableId + " :checkbox").change(function(event) {
            if ($(this).is(':checked')){
                if ($("##" + options.tableId + "
:checkbox").size() > 1 || $(this).attr('name') != 'check_all') {

```

```

$(event.target.parentNode.parentNode).addClass('row_selected');
options.tableId).removeAttr("disabled");
options.tableId).removeClass('disabled');
}
}
else {
$(event.target.parentNode.parentNode).removeClass('row_selected');
options.tableId + " :checkbox:first").is(':checked')) {
:checkbox").attr("checked", false);
tr").removeClass('row_selected');
options.tableId).attr('disabled','disabled');
options.tableId).addClass('disabled');
});
}
if (options.allowUpdate) {
$("#" + options.tableId + " tbody tr").click(function(event) {
if ($(event.target).is(":not(:first-child)")) {
var id =
if (id) {
if (options.fileEditForm.search(/\?/)
var editUrl =
}
else {
var editUrl =
}
if (options.boxHeight &&
$(this).colorbox({
href: editUrl,
open: true,
height:
options.boxHeight,
width: options.boxWidth,
onClosed: function() {
//options.table.fnReloadAjax();
options.table.fnDraw();
}
});
}
else {

```



```

else {
    window.stop();
}
$('#btn_hangup').attr('disabled', true).addClass('disabled');
$('#btn_call').attr('disabled', false).removeClass('disabled');
$.get('../cancel_caller.php',
    {
        from: from,
        to: $('#phone_number').val()
    },
    function() {
        $('#btn_call').attr('disabled', false).removeClass('disabled');
    }
);
}

```

```

usuario/public/_js/webphone_java.js

```

```

/*
 * in case don't want to use login page set these variables with username and
 * password (alternatively to password, md5 checksum can be used) then in the
 * init() function, comment the page(1) line and uncomment the page(2) line
 */
var getUser = '';
var getPassword = '';
var getMd5Checksum = '';

var dtmfDelay = 8000; // how long(in ms) sent DTMF should be displayed after last DTMF sending
var cookieExpire = 365; // cookies Expire after 'value' days
var eventDisplayTime = 5000; // time(in ms) an event should be displayed

var applethandle = null;
var checkCall; // verify if in call
var checkDTMF; // verify if sending DTMF
var alertTimerId;
var eventAlertTimerId;
var hold;
var redial;
var DTMFcount;
var callLengthTimerId; // timer id for calculating call length
var boolCallLengthTimer;
var boolRingTimer;
var callStartTime;
var callLengthInSec;
var boolIncommingCall; // true if having incomming call
var tempUser;
var tempPassw;

function webphone_init(user, password, md5Checksum) { // function to initialize all variables, called only
on page load
    //initcheck();

    getUser = user;
    getPassword = password;
    getMd5Checksum = md5Checksum;

    //Register:
    onLogin();

```

```

initcheck();
setTimeout('applehandle.API_ServerInit(serverAddress)', 300);
setTimeout('noLoginRegister()', 1300);

checkCall = false;
checkDTMF = false;
hold = false;
redial = '';
DTMFcount = 0;
btnStatusControl('btn_hangup', 1, 'hangup');
boolCallLengthTimer = true;
boolRingTimer = true;
boolIncommingCall = false;
callLengthInSec = 0;

$('#phone_number').val('');

$('#webphone_controls').show();
$('#callbuttons').show();
$('#acceptreject').hide();
}

function onLogin() { // reset afferent variables when registering: voipRegiste(), voipStatusRegiste()
    hold = false;
    DTMFcount = 0;
    boolCallLengthTimer = true;
    boolRingTimer = true;
    boolIncommingCall = false;
    callLengthInSec = 0;
    $('#btn_call').removeClass('disabled');
    $('#btn_call').attr('disabled', false);
}

function onCallInit() {
    $('#btn_call').addClass('disabled');
    $('#btn_call').attr('disabled', true);
    $('#btn_hangup').removeClass('disabled');
    $('#btn_hangup').attr('disabled', false);
    $('#btn_transfer').removeClass('inactive');
    $('#btn_hold').removeClass('inactive');
}

function onCallConnected() {
}

function onCallFinished() { // reset afferent variables when call finished
    hold = false;
    DTMFcount = 0;
    boolCallLengthTimer = true;
    boolRingTimer = true;
    boolIncommingCall = false;
    callLengthInSec = 0;
    $('#btn_call').removeClass('disabled');
    $('#btn_call').attr('disabled', false);
    $('#btn_hangup').addClass('disabled');
    $('#btn_hangup').attr('disabled', true);
    $('#btn_transfer').addClass('inactive');
}

```

```

    $('#btn_hold').addClass('inactive');
}

function noLoginRegister() { // register without login, showing directly numpad

    initcheck();
    if (getMd5Checksum) {
        applethandle.API_SetCredentialsMD5(serverAddress, getUser, getMd5Checksum, "");
    }
    applethandle.API_Register(serverAddress, getUser, getPassword+getMd5Checksum);

    tempUser = getUser;
    if (getPassword) {
        tempPassw = getPassword;
    }
    else {
        tempPassw = getMd5Checksum;
    }
}

function btnStatusControl(id, control, btn) { // call, hangup buttons disabled effect
    if ((btn == 'call' && checkCall) || (btn == 'hangup' && !checkCall)) {
        $('#'+ id).attr('disabled', true);
        $('#'+ id).addClass('disabled');
    }
}

function BtnValue(id) { // retrieving values of num pad buttons and sending DTMF if in call
    var id_length = id.length;
    var value = id.charAt(id_length - 1);
    if (id == 'btn_10') {
        value = '*'
    }
    if (id == 'btn_11') {
        value = '#'
    }

    if (checkCall == true) {
        initcheck();
        applethandle.API_Dtmf(-2, "" + value + "");
        var dtmf = $('#webphone_dtmf');
        DTMFcount++;

        checkDTMF = true;
        if (dtmf.html() == '') {
            displayDTMF('DTMF <strong>' + value + '</strong> - ' + value, dtmfDelay);
        } else {
            var auxString = dtmf.html();
            if (DTMFcount > 9) {
                displayDTMF('DTMF <strong>' + value + '</strong> - ' + value, dtmfDelay);
                DTMFcount = 0;
            } else {
                auxString = auxString.replace(/DTMF <strong>\w/, 'DTMF <strong>' +
value);
                displayDTMF(auxString + value, dtmfDelay);
            }
        }
    }
} else {

```



```

        var number = $('#phone_number').val();
        $('#phone_number').val(number + value);
    }
}

function displayDTMF(strDTMF, delay) { // timer, displaying DTMF
    $('#webphone_dtmf').html(strDTMF);
    clearTimeout(alertTimerId);
    alertTimerId = setTimeout("hideDTMF()", delay);
}

function hideDTMF() {
    checkDTMF = false;
    $('#webphone_dtmf').html('');
}

function displayStatus(statustr) { // display status messages
    $('#webphone_status').html(statustr);
}

function displayEvent(evStr) { // display event messages
    $('#webphone_events').html(evStr).fadeIn();
    clearTimeout(eventAlertTimerId);
    eventAlertTimerId = setTimeout("hideEvent()", eventDisplayTime);
}

function hideEvent() { // hide event messages
    $('#webphone_events').fadeOut(function() {
        $(this).html('');
    });
}

function dispCallLength() { // display call length
    var currTime = new Date();
    currentTime = currTime.getTime();
    var callLengthSec = parseInt((currentTime - callStartTime) / 1000);
    callLengthInSec = callLengthSec;
    var Sec = callLengthSec % 60;
    var callLengthMin = parseInt(callLengthSec / 60);
    var Min = callLengthMin % 60;
    var Hour = parseInt(callLengthMin / 60);

    var strTime = '<span>';
    if (Hour > 0) {
        strTime += Hour + ':';
    }
    strTime += Min + ':';
    if (Sec < 10) {
        strTime += '0';
    }
    strTime += Sec;
    if (Min < 0) {
        Hour--;
        Min += 60;
    }
    if (hold == true) {
        strTime += ' (Em espera)';
    }
}

```

```

strTime += '</span>';
displayStatus(strTime);
callLengthTimerId = setTimeout("dispCallLength()", 500);
}

function initcheck() {
    if (applethandle == null) {
        displayStatus('<span class="loading">Inicializando...</span>');

        try {
            applethandle = document.getElementById('webphone');
        }
        catch (e) { }

        if (applethandle == null) {
            var applets = null;
            try {
                applets = document.applets;

                // Needed for FireFox
                if (applets.length == 0) {
                    applets = document.getElementsByTagName("object");
                }
                if (applets.length == 0) {
                    applets = document.getElementsByTagName("applet");
                }

                // Find the active applet object
                for (var i=0; i<applets.length; ++i) {
                    try {
                        if (typeof (applets[i].API_Call) != "undefined") {
                            applethandle = applets[i];
                            break;
                        }
                    }
                    catch (e) { }
                }
            }
            catch (e) { }

            if (applethandle == null) {
                try {
                    applethandle = document.applets[0];
                }
                catch (e) { }
            }

            if (applethandle == null) {
                displayStatus('Não foi possível encontrar o applet!');
            }
        }

        if (applethandle != null) {
            // See if we're using the old Java Plug-In and the JNLPAppletLauncher
            try {
                applethandle = applethandle.getSubApplet();
            } catch (e) {
                // Using new-style applet -- ignore
            }
        }
    }
}

```

```

    }
}

}

//function voipRegister() {
//    onLogin();
//    initcheck();
//    applethandle.API_ServerInit("" + serverAddress + "");
//    setTimeout("voipDelayedRegister();", 400);
//}

function voipStatusRegister() { // register if double click on status and not in call
    if ((checkCall == false) && (tempUser != '') && (tempUser != null) && (tempPassw != '') &&
(tempPassw != null)) {
        onLogin();
        initcheck();
        applethandle.API_Register("" + serverAddress + "", "" + tempUser + "", "" + tempPassw +
"");
    }
}

function voipUnregister() {
    initcheck();
    applethandle.API_Unregister();
    $('#btn_call').addClass('disabled');
    $('#btn_call').attr('disabled', true);
}

function voipCall(to_number) {
    initcheck();
    if (to_number) {
        $('#phone_number').val(to_number);
    }
    var boolCall = false;
    var calledNumber = $('#phone_number').val();
    if (calledNumber != '' && calledNumber != getUser) {
        boolCall = applethandle.API_Call(-1, "" + calledNumber + "");
        redial = calledNumber;
        $('#btn_redial').removeClass('inactive');
    }
    if (boolCall == false) {
        displayEvent('<span class="error">Erro ao efetuar a chamada.</span>');
    }
}

function voipHangup() {
    initcheck();
    var boolHangup = false;
    boolHangup = applethandle.API_Hangup(-1);
    $('#webphone_dtmf').text('');
    checkDTMF = false;

    boolIncommingCall = false;
}

function voipAccept() {
    initcheck();

```

```

var boolAccept = false;
boolAccept = applethandle.API_Accept(-1);
if (boolAccept == false) {
    displayEvent('<span class="error">Erro ao aceitar a chamada.</span>');
}
}

function voipReject() {
    initcheck();
    var boolReject = false;
    boolReject = applethandle.API_Reject(-1);
}

function voipHold() {
    initcheck();
    if (!checkCall) {
        displayEvent('<span class="warning">Nenhuma chamada ativa</span>');
    }
    else {
        var boolHold = applethandle.API_Hold(-1, hold);
        if (boolHold == true) {
            hold = !hold;
            if (hold == true) {
                displayEvent('<span class="loading">Chamada em espera</span>');
            } else {
                displayEvent('<span class="success">Chamada em andamento</span>');
            }
        } else {
            displayEvent('<span class="error">Erro ao colocar em espera</span>');
        }
    }
}

function voipRedial() {
    var boolRedial = false;
    if (!checkCall) {
        initcheck();
        if (redial != '') {
            boolRedial = applethandle.API_Call(-1, "" + redial + "");
            $('#phone_number').val(redial);
        }
    }
    if (boolRedial == false) {
        displayEvent('<span class="warning">Nenhuma chamada realizada</span>');
    }
}

function voipTransfer() {
    initcheck();
    if (!checkCall) {
        displayEvent('<span class="warning">Nenhuma chamada ativa</span>');
    }
    else {
        var boolTransfer = false;
        boolTransfer = applethandle.API_TransferDialog();
        if (boolTransfer == false) {
            displayEvent('<span class="error">Erro ao transferir</span>');
        }
    }
}

```

```

    }
}

function voipAudioDevice() {
    initcheck();
    applehandle.API_AudioDevice();
}

function handleNotifications(varr) {
    var eventNotify = '' + varr;
    var callButtons = $('#callbuttons');
    var acceptReject = $('#acceptreject');

    initcheck();
    var stat = applehandle.API_GetStatus(-2); // get general status
    var status = stat.replace(/\s+/g, '');

    var chanelStat = applehandle.API_GetStatus(1); // get line status
    var chanelStatus = chanelStat.replace(/\s+/g, '');

    var inCallIndex = status.lastIndexOf("(");
    var inCallStatus = status.slice(0, inCallIndex);

    if ((status == 'Startingcall') || (inCallStatus == 'InCall')) { // call buttons control
        checkCall = true;
        onCallInit();
        btnStatusControl('btn_call', 1, 'call');
        btnStatusControl('btn_hangup', 1, 'hangup');
    }
    if (status == 'CallFinished') {
        checkCall = false;
        onCallFinished();
        btnStatusControl('btn_call', 1, 'call');
        btnStatusControl('btn_hangup', 1, 'hangup');
    }
    if (status == 'Incoming...') {
        callButtons.hide();
        acceptReject.show();
    }
    if ((inCallStatus == 'InCall') || (status == 'CallFinished')) {
        callButtons.show();
        acceptReject.hide();
    }
    if ((inCallStatus == 'Speaking') || (chanelStatus == 'Ringing')) { // calculating call length
        if (chanelStatus == 'Ringing') {
            if (boolRingTimer == true) {
                clearTimeout(callLengthTimerId);
                var time = new Date();
                if (boolIncommingCall == true) {
                    callStartTime = time.getTime();
                    callStartTime = callStartTime - 2000; // add 2 sec to call
length, when incomming call
                } else {
                    callStartTime = time.getTime();
                }
                dispCallLength();
                boolRingTimer = false;
            }
        }
    }
}

```

```

    }
    if (inCallStatus == 'Speaking') {
        if (boolCallLengthTimer == true) {
            clearTimeout(callLengthTimerId);
            var time = new Date();
            if (boolIncommingCall == true) {
                callStartTime = time.getTime();
                callStartTime = callStartTime - 2000; // add 2 sec to call
length, when incomming call
            } else {
                callStartTime = time.getTime();
            }
            dispCallLength();
            boolCallLengthTimer = false;
        }
    }
} else {
    if (checkCall == false) {
        boolCallLengthTimer = true;
        boolRingTimer = true;
        clearTimeout(callLengthTimerId);
        var realStatus = '';
        if (status == 'RegisterFailed') {
            realStatus = '<span class="error">Erro ao registrar</span>';
        } else if (status == 'Ready.') {
            realStatus = '<span class="success">Pronto</span>';
        } else if (status == 'Register...' || status == 'Registering...') {
            realStatus = '<span class="loading">Registrando...</span>';
        } else if (status == 'Registered.') {
            realStatus = '<span class="success">Registrado</span>';
        } else if (status == 'CallFinished') {
            realStatus = '<span>Chamada encerrada!</span>';
        } else {
            realStatus = '<span>' + status + '</span>';
        }
        displayStatus(realStatus);
    }
}

var inCallingField = document.getElementById('phone_number'); // get and display incomming caller
if (chanelStatus == 'Ringing') {
    var incomming = eventNotify.slice(eventNotify.lastIndexOf(",") + 1, eventNotify.length);
    if (incomming == '2') {
        var inCallingNum = eventNotify.split(",", 4);
        inCallingNum = inCallingNum.toString();
        inCallingNum = inCallingNum.slice(inCallingNum.lastIndexOf(",") + 1,
inCallingNum.length);

        inCallingField.value = inCallingNum;
        redial = inCallingNum;
        $('#btn_redial').removeClass('inactive');
        boolIncommingCall = true;
        displayEvent('Chamada de: <span style="font-weight:bold;">' + inCallingNum +
'</span>');
    }
}

if (status == 'CallFinished') {
    if (callLengthInSec > 2) {
        inCallingField.value = '';
    }
}

```

```

        boolIncommingCall = false;
    }
    var eventName = eventNotify.slice(0, eventNotify.indexOf(",")); // displaying events
    var eventType = eventNotify.split(", ", 2);
    eventType = eventType.toString();
    eventType = eventType.slice(eventType.indexOf(",") + 1, eventType.length);
    var eventDestroy = eventNotify.match('EVENT,EVENT, destroying');

    if (eventName == 'EVENT') {
        var evnt = eventNotify.slice(eventNotify.indexOf(",") + 1, eventNotify.length);
        evnt = evnt.toString();
        evnt = evnt.slice(evnt.indexOf(",") + 1, evnt.length);
        evnt = evnt.substr(0, 1).toUpperCase() + evnt.substr(1);
        evnt = evnt.toString();
        if (evnt.length > 36) {
            evnt = evnt.slice(0, 36);
        }
        if ((eventType == 'EVENT') && (eventDestroy != 'EVENT,EVENT, destroying')) {
            //displayEvent(evnt);
        }
        if (eventType == 'WARNING') {
            displayEvent('<span class="warning">' + evnt + '</span>');
        }
        if (eventType == 'ERROR') {
            displayEvent('<span class="error">' + evnt + '</span>');
        }
    }
}

```

usuario/public/_js/webphone_sofphone.js

```

function BtnValue(id) {
    var id_length = id.length;
    var value = id.charAt(id_length - 1);
    if (id == 'btn_10') {
        value = '*'
    }
    if (id == 'btn_11') {
        value = '#'
    }
    var number = $('#phone_number').val();
    $('#phone_number').val(number + value);
}

function voipCall(to_number) {
    if (to_number) {
        $('#phone_number').val(to_number);
    }
    if ($('#phone_number').val()) {
        window.location = 'sip://' + $('#phone_number').val() + '@' + serverAddress;
    }
}

$(function() {
    $('#btn_call').attr('disabled', false).removeClass('disabled');
});

```

```

usuario/public/configuracoes/configuracoes.php
<?php
require_once(dirname(__FILE__) . '/../../application/include/Auth.php');
require_once(dirname(__FILE__) . '/../../application/include/DB.php');
require_once(dirname(__FILE__) . '/../../application/include/LDAP.php');
$DB = DB::getInstance();
$LDAP = LDAP::getInstance();

if (!Auth::check()) {
    exit('Acesso negado');
}
$filter = "(cn={$_SESSION['ramal']})";
$fields = array('mail', 'l', 'serviceCode');
$dn = "cn={$_SESSION['ramal']},ou=sip";

$result = $LDAP->search($dn, $filter, $fields);

for ($i=0; $i<$result['count']; $i++) {
    $email = $result[$i]['mail'][0];
    $local = $result[$i]['l'][0];
    $servicecode = $result[$i]['servicecode'][0];
}
?>
<h3>Configurações</h3>
<form id="atualizar_configuracoes" action="?action=update" method="post" class="inline">
    <h4>Seus dados</h4>
    <p><label for="email">E-mail</label><br />
    <input type="text" id="email" name="email" class="title" value="<?php echo $email; ?>" style="width:
230px" /></p>
    <p><label for="nome">Local</label><br />
    <input type="text" id="local" name="local" class="title" value="<?php echo $local; ?>" style="width:
230px" /></p>
    <p><label for="senhaWeb">Senha</label><br />
    <input type="text" id="senha_placeholder" name="senha_placeholder" title="Digite a nova senha aqui"
onfocus="showPassword('senha_placeholder', 'senhaWeb')" class="title" autocomplete="off" style="width:
230px" />
    <input type="password" id="senhaWeb" name="senhaWeb" style="display: none"
onblur="hidePassword('senha_placeholder', 'senhaWeb')" class="title" autocomplete="off" style="width:
230px" /></p>
    <div class="clear"></div>
    <br />
    <h4>Serviços</h4>
    <?php
    $filter = '(cn=*)';
    $fields = array('description', 'serviceCode');
    $dn = 'ou=servicos';

    $result = $LDAP->search($dn, $filter, $fields);

    for ($i=0; $i<$result['count']; $i++) {
        if ((int)$servicecode & (int)$result[$i]['servicecode'][0]) {
            $checked = "checked='checked'";
        }
        else {
            $checked = '';
        }
        if ((int)$result[$i]["servicecode"][0] & (int)1) {
            echo "<label class='regular'>";

```



```

        echo "<input id='service_code'.".$result[$i]["servicecode"][0]."' name='SERVICECODE['
type='checkbox' onchange='toggleCorreioVoz()' value='".$result[$i]["servicecode"][0]."' $checked /> ";
        echo $result[$i]["description"][0];
        echo "</label>";

        ?>

        - Senha: <input name="senha_voicemail" id="senha_voicemail" type="password" size="20"
maxlength="10" style="display: none; width: 175px" onblur="hidePassword('senha_voicemail_placeholder',
'senha_voicemail')" class="title" autocomplete="off" />
        <input type="text" id="senha_voicemail_placeholder" name="senha_voicemail_placeholder"
title="Digite a nova senha aqui" onfocus="showPassword('senha_voicemail_placeholder', 'senha_voicemail')"
style="width: 175px" class="title" autocomplete="off" />
        <?php
        echo "<br />";
        }
        else {
        echo "<label class='regular'>";
        echo "<input id='service_code'.".$result[$i]["servicecode"][0]."' name='SERVICECODE['
type='checkbox' value='".$result[$i]["servicecode"][0]."' $checked /> ";
        echo $result[$i]["description"][0];
        echo "</label><br />";
        }
        }
        ?>
        <br />
        <em>Para acessar seu correio de voz, disque <a href="javascript:voipCall('*45')">*45</a>. Para deixar
um recado para um ramal, disque <a href="javascript:voiceMessage('*44')">*44 + RAMAL</a>.</em>
        <br /><br />
        <h4>Portal</h4>
        <p><label for="tipo_webphone">Webphone</label><br />
        <label class="normal"><input name="tipo_webphone" type="radio" value="java" <?php if
($_SESSION['webphone'] == 'java') echo 'checked="checked"'; ?> /> Java</label>
        <em>(Efetua e recebe ligações, utilizando o microfone e som do seu computador - Beta)</em><br />
        <label class="normal"><input name="tipo_webphone" type="radio" value="click2call" <?php if
($_SESSION['webphone'] == 'click2call') echo 'checked="checked"'; ?> /> Click2Call</label>
        <em>(Apenas efetua ligações a partir de seu ramal)</em><br />
        <label class="normal"><input name="tipo_webphone" type="radio" value="softphone" <?php if
($_SESSION['webphone'] == 'softphone') echo 'checked="checked"'; ?> /> Softphone</label>
        <em>(Apenas efetua ligações a partir do softphone instalado em seu computador)</em></p>
        <div class="clear"></div>
        <p><label for="numero_max_contatos">Número máximo de contatos na lista lateral</label><br />
        <select id="numero_max_contatos" name="numero_max_contatos" style="width: 70px">
        <option <?php if ($_SESSION['numero_max_contatos'] == 5) echo 'selected="selected"'; ?>>5</option>
        <option <?php if ($_SESSION['numero_max_contatos'] == 10) echo 'selected="selected"';
?>>10</option>
        <option <?php if ($_SESSION['numero_max_contatos'] == 15) echo 'selected="selected"';
?>>15</option>
        <option <?php if ($_SESSION['numero_max_contatos'] == 20) echo 'selected="selected"';
?>>20</option>
        <option <?php if ($_SESSION['numero_max_contatos'] == 25) echo 'selected="selected"';
?>>25</option>
        </select></p>

        <div class="clear"></div>
        <p><br /></p><input type="submit" value="Salvar" /></p>
        <div class="clear"></div>
        <div id="status" style="margin-top: 10px"></div>
</form>
<script type="text/javascript">

```

```

function scrollTo(element, callback, offset) {
    var position = element.offset().top;
    if (offset) {
        position+= offset;
    }
    $('html, body').animate({
        scrollTop: position
    }, 1000, function() {
        if (callback) {
            callback();
        }
    });
}

function voiceMessage(number) {
    $('#phone_number').val(number).focus();
}

$(function() {

    /* Form submission (AJAX + JSON)
    *****/

    //Define the options (functions) to handle the submit and response
    var options = {
        beforeSubmit: function() {
            $('.error_box').css('visibility', 'hidden'); //Hide error messages (if exists)
            $('#status').html('<span class="loading"></span>'); //AJAX loading gif
        },
        success: function(response, status) {
            $('.error_box').remove(); //Remove error messages (if exists)
            //If no errors occurred, print the success message
            if (response && response.error == null) {
                $('#status').html('<span class="success">Configurações atualizadas com sucesso!</span>');
            }
            else {
                //Highlight invalid fields
                if (response && typeof(response.error.length) == "undefined") {
                    $('#status').html('<span class="error">Verifique o(s) campo(s) com
problema(s)</span>');
                    var focus = true;
                    $.each(response.error, function(i, val) {
                        $('#'+ i).after('<div class="error_box">'+val+'</div>');
                        if (focus) {
                            scrollTo($('#'+ i), function() {
                                $('#'+ i).focus();
                            }, -25);
                            focus = false;
                        }
                    });
                }
            }
            else {
                if (!response) {
                    response = {};
                    response.error = 'Ocorreu um erro inesperado';
                }
                //Print the error message
                $('#status').html('<span class="error">'+response.error+'</span>');
            }
        }
    };

```

```

    }
    },
    error: function(XMLHttpRequest, textStatus, errorThrown) {
        //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
        $('#status').html('<span class="error">'+textStatus+'<br
/>'+XMLHttpRequest.responseText+'</span>');
    },
    dataType: 'json'
};

$('#atualizar_configuracoes').ajaxForm(options);
toggleCorreioVoz();
});

function showPassword(placeholder, password) {
    $('#'+placeholder).hide();
    $('#'+password).show();
    $('#'+password).focus();
}

function hidePassword(placeholder, password) {
    if ($('#'+password).val() === '') {
        $('#'+password).hide();
        $('#'+placeholder).show();
        $('#'+placeholder).val($('#'+placeholder).attr('title')).addClass('placeholder');
    }
}

function toggleCorreioVoz() {
    if ($('#service_codel')[0].checked) {
        $('#service_code2').attr('disabled', false);
        $('#service_code4').attr('disabled', false);
        $('#senha_voicemail_placeholder').attr('disabled', false);
        $('#senha_voicemail_placeholder').removeClass('disabled');
    }
    else {
        $('#service_code2').attr('disabled', true);
        $('#service_code2')[0].checked = false;
        $('#service_code4').attr('disabled', true);
        $('#service_code4')[0].checked = false;
        $('#senha_voicemail_placeholder').attr('disabled', true);
        $('#senha_voicemail_placeholder').addClass('disabled');
        $('#senha_voicemail').val('');
        hidePassword('senha_voicemail_placeholder', 'senha_voicemail');
    }
}
</script>

```

```

usuario/public/configuracoes/index.php

```

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/DataTables.php');
require_once(dirname(__FILE__) . '/../../application/include/Auth.php');
require_once(dirname(__FILE__) . '/../../application/include/DB.php');
require_once(dirname(__FILE__) . '/../../application/include/LDAP.php');
require_once(dirname(__FILE__) . '/../../application/include/Log.php');

```

```

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    case 'update':

        if (!Auth::check()) {
            exit(json_encode(array('error' => 'Acesso negado')));
        }

        $LDAP = LDAP::getInstance();
        $DB = DB::getInstance();
        $DB->connect();

        //Seus dados
        $email = isset($_POST['email']) ? $_POST['email'] : '';
        $local = isset($_POST['local']) ? $_POST['local'] : '';
        $senhaWeb = isset($_POST['senhaWeb']) ? $_POST['senhaWeb'] : '';
        //Serviços
        $senha_voicemail = isset($_POST['senha_voicemail']) ? $_POST['senha_voicemail'] : '';
        $servicecode = isset($_POST['SERVICECODE']) && is_array($_POST['SERVICECODE']) ?
array_sum($_POST['SERVICECODE']) : 0;
        //Portal
        $webphone = isset($_POST['tipo_webphone']) ? $_POST['tipo_webphone'] : 'java';
        $numero_max_contatos = isset($_POST['numero_max_contatos']) ? $_POST['numero_max_contatos'] : 10;

        $ramal = $_SESSION['ramal'];

        $invalid_fields = array();

        /* Valida a senha */
        if (strlen($senhaWeb) < 4 && $senhaWeb != '') {
            $invalid_fields['senhaWeb'] = 'A senha (web) deve conter 4 caracteres ou mais';
        }

        /* Valida o e-mail */
        if (!validate_email($email)) {
            $invalid_fields['email'] = 'E-mail inválido';
        }

        if ($invalid_fields) {
            exit(json_encode(array('error' => $invalid_fields)));
        }

        /* Valida o código de serviço */
        if (!is_numeric($servicecode)) {
            exit(json_encode(array('error' => 'Código de serviço inválido')));
        }
        elseif ($servicecode & 2 || $servicecode & 4) {
            if (!$servicecode & 1) {
                exit(json_encode(array('error' => 'Código de serviço inválido')));
            }
        }

        /* Valida o tipo de webphone */
        if ($webphone != 'java' && $webphone != 'click2call' && $webphone != 'softphone') {
            exit(json_encode(array('error' => 'Webphone inválido')));
        }
    }
}

```

```

}

/* Valida o número máximo de contatos da lista lateral */
if ($numero_max_contatos < 5 || $numero_max_contatos > 25) {
    exit(json_encode(array('error' => 'Número máximo de contatos deve ser entre 5 e
25')));
}

/* Pesquisa os valores (atributos) antigos do ramal */
$filter = "(cn=$ramal)";
$dn = 'ou=sip';
$fields = array('l', 'mail', 'serviceCode', 'contact');
$result = $LDAP->search($dn, $filter, $fields);

$responsavel = isset($result[0]['contact'][0]) ? $result[0]['contact'][0] : '';
$local_antigo = isset($result[0]['l'][0]) ? $result[0]['l'][0] : '';
$email_antigo = isset($result[0]['mail'][0]) ? $result[0]['mail'][0] : '';
$servicecode_antigo = isset($result[0]['servicecode'][0]) ? $result[0]['servicecode'][0] :
'';

/* Pesquisa as configurações antigas do ramal */
$query = sprintf('SELECT * FROM usuarios_configuracao
                WHERE ramal = %d',
                mysql_real_escape_string($ramal));
$result_sql = $DB->query($query);

$webphone_antigo = '';
$numero_max_contatos_antigo = '';
while ($row = mysql_fetch_array($result_sql)) {
    $webphone_antigo = $row['webphone'];
$numero_max_contatos_antigo = $row['numero_max_contatos'];
}

/* Verifica os dados que foram alterados */
$arrayRamalInfo = array();
$str_atributos_alterados = '';
$alteracoes = false;

if ($local_antigo != $local) {
    $arrayRamalInfo['l'] = $local;
    $str_atributos_alterados.= "Local de instalação:\n Atual: $local\n";
    $str_atributos_alterados.= " Anterior: $local_antigo\n\n";
    $alteracoes = true;
}

if ($email_antigo != $email) {
    $arrayRamalInfo['mail'] = $email;
    $str_atributos_alterados.= "E-mail:\n Atual: $email\n";
    $str_atributos_alterados.= " Anterior: $email_antigo\n\n";
    $alteracoes = true;
}

if ($senhaWeb) {
    $arrayRamalInfo['senhaWeb'] = md5($senhaWeb);
    $str_atributos_alterados.= "Senha para acesso web:\n Atual: $senhaWeb\n\n";
    $alteracoes = true;
}

```

```

if ($senha_voicemail) {
    $str_tributos_alterados.= "Senha para secretária eletrônica:\n Atual: $senha_voicemail\n\n";
    $alteracoes = true;
}

if ($servicecode_antigo != $servicecode) {
    $arrayRamalInfo['serviceCode'] = $servicecode;
    $alteracoes = true;
}

if ($webphone_antigo != $webphone) {
    $str_tributos_alterados.= "Webphone:\n Atual: $webphone\n";
    $str_tributos_alterados.= " Anterior: $webphone_antigo\n\n";
    $alteracoes = true;
}

if ($numero_max_contatos_antigo != $numero_max_contatos) {
    $str_tributos_alterados.= "Número máximo de contatos na lista lateral:\n Atual:
$numero_max_contatos\n";
    $str_tributos_alterados.= " Anterior: $numero_max_contatos_antigo\n\n";
    $alteracoes = true;
}

/* Atualiza as configurações */
$sql_configuracoes = "REPLACE INTO voip.usuarios_configuracao
    SET ramal = '$ramal',
        webphone = '$webphone',
        numero_max_contatos = '$numero_max_contatos'";

try {
    $DB->query($sql_configuracoes);
    $_SESSION['webphone'] = $webphone;
    $_SESSION['numero_max_contatos'] = $numero_max_contatos;
}
catch (Exception $e) {
    exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
}

$DB->connect_to_midia();

/* Atualiza o registro da secretária eletrônica para este ramal */
$result_sql = $DB->query("SELECT COUNT(*) AS total
    FROM voip.voicemail
    WHERE uniqueid = '$ramal'");

$count = mysql_result($result_sql, 0, 'total');

if ($servicecode & 1 && $count == 0) {

    if ($senha_voicemail == '' || !(is_numeric($senha_voicemail) && strlen($senha_voicemail) >=
4 && strlen($senha_voicemail) <= 10)) {
        exit(json_encode(array('error' => 'A senha da secretária eletrônica deve ser numérica e
possuir de 4 a 10 caracteres.')));
    }

    if ($servicecode & 2) {
        $sql_voicemail = "INSERT INTO voip.voicemail (uniqueid, customer_id,
mailbox,password, fullname, email)
VALUES

```

```

('$ramal','$ramal','$ramal','$senha_voicemail','$responsavel','$email');
    }
    else {
        $sql_voicemail = "INSERT INTO voip.voicemail (uniqueid, customer_id,
mailbox, password, fullname, email)
                                VALUES
('$ramal','$ramal','$ramal','$senha_voicemail','$responsavel','')";
    }
    try {
        $DB->query($sql_voicemail);
    }
    catch (Exception $e) {
        exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
    }
    }
    else {
        if ($servicecode & 1) {

            if ($senha_voicemail && !(is_numeric($senha_voicemail) &&
strlen($senha_voicemail) >= 4 && strlen($senha_voicemail) <= 10)) {
                exit(json_encode(array('error' => 'A senha da secretária eletrônica deve ser numérica e
possuir de 4 a 10 caracteres.')));
            }

            $append = '';
            if ($senha_voicemail) {
                $append .= ", password = '$senha_voicemail'";
            }
            if ($servicecode & 2) {
                $append .= ", email = '$email'";
            }
            else {
                $append .= ", email = ''";
            }
            $sql_voicemail = "UPDATE voip.voicemail
                                SET uniqueid = '$ramal',
                                customer_id = '$ramal',
                                mailbox = '$ramal',
                                fullname = '$responsavel'
                                $append
                                WHERE uniqueid = '$ramal'";

            try {
                $DB->query($sql_voicemail);
            }
            catch (Exception $e) {
                exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
            }
            }
            else {
                $sql_voicemail = "DELETE FROM voip.voicemail
                                WHERE mailbox = '$ramal'";

                try {
                    $DB->query($sql_voicemail);
                }
                catch (Exception $e) {
                    exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
                }
            }
        }
    }
}

```

```

    }

    $DB->connect_to_local();

    # Se houver algum atributo do ramal alterado; o ramal será atualizado. #
    if ($arrayRamalInfo) {

        # LOG - Inicio #
        $logRamaisAtualizados = '';
        # Todos os atributos do $arrayRamalInfo (valores atualizados) juntamente com seus
valores anteriores (valoresAntigos) serão inseridos no $log descricao #
        foreach ($arrayRamalInfo as $key => $value) {
            $old_value = '';
            if (isset($result[0][strtolower($key)][0])) {
                $old_value = $result[0][strtolower($key)][0];
            }
            $logRamaisAtualizados .= $key . ': ' . $old_value . ' -> ' . $value .
'<br />';
        }
        $logRamaisAtualizados = substr_replace($logRamaisAtualizados, '',
(sizeof($logRamaisAtualizados) - 3));

        $log = new Log($DB);
        $log_acao = $log->getIdAcao('ALTERAÇÃO DE RAMAL');
        $log_descricao = "Ramal: $ramal - Atributos atualizados: $logRamaisAtualizados";

        if (!$log->inserirLog($log_acao, $log_descricao)) {
            exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
        }
        # LOG - Fim #
        # Atualiza o registro no LDAP #

        try {
            $dn = "cn=$ramal,ou=sip";
            $LDAP->update($dn, $arrayRamalInfo);
        }
        catch (Exception $e) {
            exit(json_encode(array('error' => 'Ocorreu um erro no banco de dados.')));
        }

        }

        elseif (!$alteracoes) {
            exit(json_encode(array('error' => 'Nenhum campo foi modificado')));
        }

    }

    if ($str_atributos_alterados) {
        $msg = "$responsavel,\n";
        $msg.= "Os dados de seu ramal ($ramal) na PBX-IP foram atualizados.\n";
        $msg.= "Veja abaixo os detalhes das alterações:\n\n";
        $msg.= $str_atributos_alterados;

        $msg.= "\nObservações:\n";
        $msg.= "- Este e-mail foi enviado de forma automática. Se tiver dúvidas entre com contato
através do ramal 6333.\n";
        $msg.= "- Acesse http://adm.voip.ufsc.br/ para visualizar a lista de ramais IP corporativos da
UFSC.";

        send_mail($DB, $email, "Atualização de Ramal ($ramal)", $msg);
    }

```



```

    }
    exit(json_encode(array('error' => null)));

    break;

default:
    require_once(dirname(__FILE__) . '/../../application/include/TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
        'configuracoes.css',
        'jquery-ui-1.8.2.custom.css'
    ));
    $template->set_js_files(array(
        'jquery.form.js',
        'jquery-ui-1.8.2.custom.min.js'
    ));
    $template->set_content_file('configuracoes/configuracoes.php');
    $template->set_active_nav_item(ID_CONFIGURACOES);
    $template->set_authenticated_only(true);
    $template->show();
}
?>

```

usuario/public/contatos/contatos.php

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/Auth.php');
if (!Auth::check()) {
    exit('Acesso negado');
}
?>
<h3>Contatos</h3>
<form id="adicionar_contato" action="?action=add" method="post" class="inline">
    <h4>Adicionar contato</h4>
    <p><label for="nome">Nome</label><br />
    <input type="text" id="nome" name="nome" class="title" style="width: 220px" /></p>
    <p><label for="email">E-mail</label><br />
    <input type="text" id="email" name="email" class="title" style="width: 210px" /></p>
    <p><label for="telefone">Telefone</label><br />
    <input type="text" id="telefone" name="telefone" class="title" style="width: 160px" /></p>
    <p><br /><input type="submit" value="Adicionar" /></p>
    <div class="clear"></div>
    <div id="status" style="margin-top: 10px"></div>
</form>
<form id="lista_contatos_container" action="?action=del" method="post">
    <h4>Lista de contatos</h4>
    <table id="lista_contatos" class="disable_first">
        <thead>
            <tr>
                <th><input type="checkbox" name="check_all" id="check_all" /></th>
                <th>Nome</th>
                <th>E-mail</th>
                <th>Telefone</th>
            </tr>
        </thead>
        <tbody>
            <tbody>
        </tbody>
    </table>

```

```

</form>
<script type="text/javascript">
function scrollTo(element, callback, offset) {
    var position = element.offset().top;
    if (offset) {
        position+= offset;
    }
    $('html, body').animate({
        scrollTop: position
    }, 1000, function() {
        if (callback) {
            callback();
        }
    });
}
$(function() {

    /* Contacts table
    *****/

    $('#lista_contatos').loadTable({
        allowCreate: false,
        sAjaxSource: '?action=getContacts',
        aoColumns: [
            { 'sWidth': '30px', 'sName': 'id', 'bSortable': false },
            { 'sName': 'nome' },
            { 'sName': 'email' },
            { 'sName': 'telefone' }
        ],
        aaSorting: [[1, 'asc']],
        sPaginationType: 'two_button',
        fileEditForm: 'editar.php',
        delConfirmation: true,
        sDom: '<"top"rf>t<"bottom"p><"clear">',
        boxWidth: '785px',
        boxHeight: '220px'
    });

    /* Form submission (AJAX + JSON)
    *****/

    //Define the options (functions) to handle the submit and response
    var options = {
        beforeSubmit: function() {
            $('.error_box').css('visibility', 'hidden'); //Hide error messages (if exists)
            $('#status').html('<span class="loading"></span>'); //AJAX loading gif
        },
        success: function(response, status) {
            $('.error_box').remove(); //Remove error messages (if exists)
            //If no errors ocurred, print the success message
            if (response && response.error == null) {
                $('#status').html('<span class="success">Contato cadastrado com sucesso!</span>');
                $.fn.getTable('lista_contatos').fnDraw();
            }
            else {
                //Highlight invalid fields
                if (response && typeof(response.error.length) == "undefined") {
                    $('#status').html('<span class="error">Verifique o(s) campo(s) com problema(s)</span>');
                }
            }
        }
    };
}

```

```

        var focus = true;
        $.each(response.error, function(i, val) {
            $('#' + i).after('<div class="error_box">'+val+'</div>');
            if (focus) {
                scrollTo($('#' + i), function() {
                    $('#' + i).focus();
                }, -25);
                focus = false;
            }
        });
    }
    else {
        if (!response) {
            response = {};
            response.error = 'Ocorreu um erro inesperado';
        }
        //Print the error message
        $('#status').html('<span class="error">'+response.error+'</span>');
    }
}
},
error: function(XMLHttpRequest, textStatus, errorThrown) {
    //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
    $('#status').html('<span class="error">'+textStatus+':<br
/>'+XMLHttpRequest.responseText+'</span>');
},
dataType: 'json'
};

$('#adicionar_contato').ajaxForm(options);

$('#telefone').mask('(99) 9999-9999');

/* Auto-complete
*****/

var cache = {};
var options = {
    minLength: 2,
    source: function(request, response) {
        if (request.term in cache) {
            if (typeof(cache[request.term][0]) != 'undefined' &&
                request.term == cache[request.term][0].value) {
            }
            response(cache[request.term]);
            return;
        }
    }
$.ajax({
    url: '?action=search',
    dataType: 'json',
    data: request,
    success: function(data) {
        cache[request.term] = data;
        if (typeof(cache[request.term][0]) != 'undefined' &&
            request.term == cache[request.term][0].value) {
        }
    }
}

```

```

        response(data);
    }
});
},
select: function(event, ui) {
    $('#nome').val(ui.item.contact);
    $('#email').val(ui.item.mail);
    $('#telefone').val(ui.item.cn);
    return false;
}
}

function renderItem(ul, item) {
    return $('<li></li>')
        .data('item.autocomplete', item)
        .append('<a>' + item.contact + '<br /><em>' + item.mail + '<br />' + item.cn + '</em></a>')
        .appendTo(ul);
}

$('#nome').autocomplete(options).data('autocomplete')._renderItem = renderItem;
$('#email').autocomplete(options).data('autocomplete')._renderItem = renderItem;
});
</script>

```

usuario/public/contatos/editar.php

```

<?php
require_once(dirname(__FILE__) . '/../application/controllers/ControllerContacts.php');
$controller = ControllerContacts::getInstance();

$id = isset($_REQUEST['id']) ? $_REQUEST['id'] : '';
if (!$id) {
    exit('ID não especificado');
}

$contato = $controller->get($id);
if (!$contato) {
    exit('Contato não localizado');
}
$contato = $contato[0];
?>

<form id="editar_contato" action="?action=update&id=<?php echo $id; ?>" method="post" class="inline">
    <fieldset>
        <legend>Editar contato</legend>
        <p><label for="nome">Nome</label><br />
        <input type="text" id="nome" name="nome" class="title" value="<?php echo $contato['nome']; ?>"
        style="width: 220px" /></p>
        <p><label for="email">E-mail</label><br />
        <input type="text" id="email" name="email" class="title" value="<?php echo $contato['email'];
        ?>" style="width: 210px" /></p>
        <p><label for="telefone">Telefone</label><br />
        <input type="text" id="telefone" name="telefone" class="title" value="<?php echo
        $contato['telefone']; ?>" style="width: 160px" /></p>
    </fieldset>
    <input type="submit" value="Atualizar" style="position: absolute; bottom: 0; left: 0; z-index: 9999" />
</form>

<script type="text/javascript">

```

```

var loaded = false;
$(document).bind('cbox_complete', function(){
  if (!loaded) {
    loaded = true;

    /* Form submission (AJAX + JSON)
    *****/

    //Define the options (functions) to handle the submit and response
    var options = {
      beforeSubmit: function() {
        $('#error_box').css('visibility', 'hidden'); //Hide error messages (if exists)
        $('#cboxTitle').html('<span class="loading"></span>'); //AJAX loading gif
      },
      success: function(response, status) {
        $('#error_box').remove(); //Remove error messages (if exists)
        //If no errors occurred, print the success message
        if (response && response.error == null) {
          $('#cboxTitle').html('<span class="success">Contato atualizado com sucesso!</span>');
          $.fn.getTable('lista_contatos').fnDraw();
        }
        else {
          //Highlight invalid fields
          if (response && typeof(response.error.length) == "undefined") {
            $('#cboxTitle').html('<span class="error">Verifique o(s) campo(s) com
problema(s)</span>');

            var focus = true;
            $.each(response.error, function(i, val) {
              $('#' + i).after('<div class="error_box">'+val+'</div>');
              if (focus) {
                $('#' + i).focus();
                focus = false;
              }
            });
            $.colorbox.resize();
          }
          else {
            if (!response) {
              response = {};
              response.error = 'Ocorreu um erro inesperado';
            }
            //Print the error message
            $('#cboxTitle').html('<span class="error">'+response.error+'</span>');
          }
        }
      },
      error: function(XMLHttpRequest, textStatus, errorThrown) {
        //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
        $('#cboxTitle').html('<span class="error">'+textStatus+':<br
/>'+XMLHttpRequest.responseText+'</span>');
      },
      dataType: 'json'
    };

    $('#editar_contato').ajaxForm(options);
  }
}

```

```

        $('#telefone').mask('(99) 9999-9999');
    }
});
</script>

```

```

usuario/public/contatos/index.php

```

```

<?php
require_once(dirname(__FILE__) . '/../../application/controllers/ControllerContacts.php');
require_once(dirname(__FILE__) . '/../../application/include/DataTables.php');

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    case 'add':
    case 'update':

        $nome = isset($_POST['nome']) ? $_POST['nome'] : '';
        $email = isset($_POST['email']) ? $_POST['email'] : '';
        $telefone = isset($_POST['telefone']) ? $_POST['telefone'] : '';
        $controller = ControllerContacts::getInstance();
        if ($action == 'add') {
            exit($controller->add($nome, $email, $telefone));
        }
        else {
            $id = isset($_REQUEST['id']) ? $_REQUEST['id'] : '';
            exit($controller->update($id, $nome, $email, $telefone));
        }
        break;

    case 'del':
        $ids = isset($_REQUEST['ids']) ? $_REQUEST['ids'] : '';
        $controller = ControllerContacts::getInstance();
        exit($controller->del($ids));
        break;

    case 'getContacts':
        //$ramal = isset($_REQUEST['ramal']) ? $_REQUEST['ramal'] : null;
        $_GET['sTable'] = 'contatos';
        $controller = ControllerContacts::getInstance();
        exit($controller->getTableData($_GET));
        break;

    case 'search':
        $controller = ControllerContacts::getInstance();
        $term = isset($_REQUEST['term']) ? $_REQUEST['term'] : '';
        $results = $controller->get_ramais(null, array('contact', 'mail', 'cn'), 0, 15, $term);
        $response = array();
        foreach ($results as $key=>$result) {
            $response[$key]['cn'] = '(' . $result['ddd'] . ') ' . $result['PREFIXO'] . '-' . $result['cn'];
            $response[$key]['mail'] = $result['mail'];
            $response[$key]['contact'] = $result['contact'];
        }
        exit(json_encode($response));
        break;
}

```

```

default:
    require_once(dirname(__FILE__) . '/../../application/include/TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
        'contatos.css',
        'jquery.dataTables.css',
        'jquery-ui-1.8.2.custom.css'
    ));
    $template->set_js_files(array(
        'jquery.form.js',
        'jquery.dataTables.min.js',
        'jquery.loadTable.js',
        'jquery.maskedinput-1.2.2.min.js',
        'jquery-ui-1.8.2.custom.min.js'
    ));
    $template->set_content_file('contatos/contatos.php');
    $template->set_active_nav_item(ID_CONTATOS);
    $template->set_authenticated_only(true);
    $template->show();
}
?>

```

usuario/public/contatos/info.php

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/DB.php');
require_once(dirname(__FILE__) . '/../../application/include/LDAP.php');

$id = isset($_REQUEST['id']) ? $_REQUEST['id'] : '';
$status = isset($_REQUEST['status']) ? $_REQUEST['status'] : '';
if (!$id) {
    exit('ID não especificado');
}

$db = DB::getInstance();
$db->connect();

$query = sprintf('SELECT nome, email, telefone
                FROM contatos
                WHERE id=%d',
                mysql_real_escape_string($id));

$response = $db->query($query);
while ($row = mysql_fetch_array($response)) {
    $nome = utf8_encode($row['nome']);
    $email = $row['email'];
    $telefone = $row['telefone'];
}

$voip = false;
if ($status == 0 || $status == 1 || $status == 2) { //Ramais PBX-IP
    $voip = true;
}

if ($voip) {
    $ldap = LDAP::getInstance();
}

```

```

$ramal = substr($telefone, -4);

$filtro = "(cn=$ramal)";
$camposResposta = array('ou','departamento');
$base_dn = 'ou=sip';
$resposta = $LDAP->search($base_dn, $filtro, $camposResposta);

if ($resposta['count'] == 0) {
    exit('Dados não disponíveis');
}

for ($i=0; $i<$resposta['count']; $i++) {
    $unidade = isset($resposta[$i]['ou'][0]) ? $resposta[$i]['ou'][0] : '';
    $departamento = isset($resposta[$i]['departamento'][0]) ? $resposta[$i]['departamento'][0]
: '';
}

$clean_telefone = preg_replace('/\D/', '', $telefone);
$clean_telefone = '0' . $clean_telefone;

?>
<h4><?php echo "$nome"; ?></h4>
<em>
<?php
echo $telefone;
if ($email) {
    echo " - $email";
}
?>
</em>
<?php
if (isset($unidade) && isset($departamento)) {
?>
<p>
<?php
echo $unidade;
if ($departamento) {
    echo "/"$departamento";
}
?>
</p>
<?php
}
?>
<div class="opcoes_contato">
    <?php
    if ($voip) {
        ?>
        <a href="#" class="button silver disabled"><span class="chat">Chat</span></a>
        <?php
        if ($status != 0 && $status != 2) {
            ?>
            <a href="javascript:voipCall('<?php echo $ramal; ?>')" class="button silver"><span
class="dial">Discar</span></a>
            <?php
            }
            else {

```



```

        ?>
        <a href="#" class="button silver disabled"><span class="dial">Discar</span></a>
        <?php
        }
        ?>

        <a href="#" class="button silver enviar_sms" id="sms_<?php echo $ramal; ?>"><span
class="sms">SMS</span></a>
        <?php
        }
        else {
            if ($status != 2) {
                ?>
                <a href="javascript:voipCall('<?php echo $clean_telefone; ?>')" class="button silver"><span
class="dial">Discar</span></a>
                <?php
                }
                else {
                    ?>
                    <a href="#" class="button silver disabled"><span class="dial">Discar</span></a>
                    <?php
                    }
                }
            }
            if ($email) {
                ?>
                <a href="mailto:<?php echo $email; ?>" class="button silver"><span class="mail">E-mail</span></a>
                <?php
            }
            else {
                ?>
                <a href="#" class="button silver disabled"><span class="mail">E-mail</span></a>
                <?php
            }
        }
    ?>
</div>
<script type="text/javascript">
$(function() {
    $('.enviar_sms').click(function() {
        var ramal = $(this).attr('id');
        ramal = ramal.slice(-4);
        $.colorbox({
            open: true,
            href: '<?php echo URI_SMS; ?>?ramal=' + ramal,
            width: '430px',
            onOpen: function() {
                $('#contatos ul li').qtip('hide');
            }
        });
    });
});
</script>

```

```

usuario/public/correio_voz/index.php

```

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/DB.php');
//require_once(dirname(__FILE__) . '/../../application/include/FirePHPCore/Fb.php');
require_once(dirname(__FILE__) . '/../../application/include/Auth.php');
require_once(APPLICATION_PATH . '/include/php-sip/PhpSIP.class.php');

```

```

if (!Auth::check()) {
    exit('Acesso negado');
}

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    case 'download':

        $id = isset($_REQUEST['id']) ? $_REQUEST['id'] : null;
        $format = isset($_REQUEST['format']) ? $_REQUEST['format'] : 'wav';
        if (!$id) {
            exit('Message ID not specified!');
        }
        if ($format != 'wav' && $format != 'mp3') {
            exit('Invalid audio format. Use wav ou mp3');
        }
        $DB = DB::getInstance();
        $DB->connect_to_midia();
        $query = sprintf('SELECT recording
                        FROM voicemessages
                        WHERE id=%d
                        LIMIT 1',
                        mysql_real_escape_string($id));
        $response = $DB->query($query);

        if (mysql_num_rows($response) > 0) {

            $query = sprintf("UPDATE voicemessages
                            SET dir=CONCAT(SUBSTRING_INDEX(dir, '/', 7), '/Old')
                            WHERE id=%d",
                            mysql_real_escape_string($id));
            $DB->query($query);

            if ($format == 'wav') {
                header('Content-Type: audio/x-wav');
                header("Content-Description: File Transfer");
                header("Content-Type: application/force-download");
                header("Content-Disposition: attachment; filename=$id.wav");
                echo mysql_result($response, 0, 'recording');
            }
            else {
                /*
                 * Requisitos para conversão de WAV para MP3:
                 * http://lame.sourceforge.net/
                 * http://www.ffmpeg.org/
                 */
                $filename_wav = dirname(__FILE__) . '/temp.wav';
                $filename_mp3 = dirname(__FILE__) . '/temp.mp3';
                if (!file_exists($filename_wav) || !is_writable($filename_wav)) {
                    if (!$handle = @fopen($filename_wav, 'w')) {
                        exit("Cannot open file ($filename_wav)");
                    }
                    if (@fwrite($handle, mysql_result($response, 0, 'recording')) === false) {
                        exit("Cannot write to file ($filename_wav)");
                    }
                }
            }
        }
    }
}

```

```

        fclose($handle);
    } else {
        exit("The file $filename_wav is not writable");
    }
    system("ffmpeg -i $filename_wav -y $filename_mp3");
    header('Content-Type: audio/mpeg');
    header('Content-Length: ' . urldecode(filesize($filename_mp3)));
    header("Content-Description: File Transfer");
    header("Content-Type: application/force-download");
    header("Content-Disposition: attachment; filename=$id.mp3");
    echo file_get_contents($filename_mp3);
    //unlink($filename_wav);
    //unlink($filename_mp3);
    }
}
break;

case 'delete':

    $id = isset($_REQUEST['id']) ? $_REQUEST['id'] : null;

    if (!$id) {
        exit('Message ID not specified');
    }

    $DB = DB::getInstance();
    $DB->connect_to_midia();
    $query = sprintf('DELETE FROM voicemessages
        WHERE id=%d',
        mysql_real_escape_string($id));
    $DB->query($query);

    echo json_encode(array('error'=>null));

    break;

default:

    exit(json_encode(array('error'=>'Invalid action')));
}

/* SIP NOTIFY */
try {

    $has_new = 'no';
    $new_messages = 0;
    $old_messages = 0;
    $urgent_messages = 0;

    $query = sprintf("SELECT
        COUNT(*) AS new
        FROM voicemessages
        WHERE SUBSTRING(dir, -5) = 'INBOX'
        AND mailboxuser = %d",
        mysql_real_escape_string($_SESSION['ramal']));

    $result = $DB->query($query);

```

```

$new_messages = mysql_result($result, 'new');

$query = sprintf("SELECT
                COUNT(*) AS new
                FROM voicemessages
                WHERE SUBSTRING(dir, -3) = 'Old'
                AND mailboxuser = %d",
                mysql_real_escape_string($_SESSION['ramal']));

$result = $DB->query($query);
$sold_messages = mysql_result($result, 'old');

if ($new_messages > 0) {
    $has_new = 'yes';
}

$from = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
$password = isset($_SESSION['senha']) ? $_SESSION['senha'] : '';

if ($from && $password) {

    $from_sip = "sip:$from@" . REALM;

    $api = new PhpSIP(SERVER_IP);

    $api->setUsername($from);
    $api->setPasswordMd5($password);

    $api->setMethod('NOTIFY');
    $api->setFrom('sip:'. $from .'@'. $api->getSrcIp());
    $api->setUri($from_sip);
    $api->setContentType('application/simple-message-summary');
    $api->addHeader('Event: message-summary');

    $sip_msg_content = "Messages-Waiting: $has_new\n";
    $sip_msg_content .= "Voicemail: $new_messages/$old_messages
($urgent_messages/$urgent_messages)\n\n";

    $api->setBody($sip_msg_content);

    $res = $api->send();

}

}
catch (Exception $e) {
//    exit($e->getMessage());
}
}

```

usuario/public/login/index.php

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/Auth.php');

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    case 'login':

```

```

    $ramal = isset($_POST['login_ramal']) ? $_POST['login_ramal'] : '';
    $senha = isset($_POST['login_senha']) ? $_POST['login_senha'] : '';

    exit(Auth::login($ramal, $senha));
    break;

case 'logout':

    Auth::logout();
    header('Location: ' . URI_RAIZ);
    exit();
    break;

default:

    if (Auth::check()) {
        header('Location: ' . URI_RAIZ);
        exit();
    }

    require_once(dirname(__FILE__) . '/../../application/include/TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
        'public.css'
    ));
    $template->set_js_files(array(
        'jquery.form.js'
    ));
    $template->set_content_file('login/login.php');
    $template->set_authenticated_only(false);
    $template->show();
}
?>

```

usuario/public/login/login.php

```

<div id="login_wrapper">
    <br />
    <form id="login_usuario" name="login_usuario" method="post" action="<?php echo URI_LOGIN .
'?'action=login'; ?>">
        <p><label>Ramal</label>
        <input type="text" id="login_ramal" name="login_ramal" class="title" /></p>
        <p><label>Senha</label>
        <input type="password" id="login_senha_fake" name="login_senha_fake" class="title" />
        <input type="hidden" id="login_senha" name="login_senha" /></p>
        <a href="<?php echo URI_REDEFINICAO; ?>" title="Clique aqui para redefinir sua senha ou ativar sua
conta">Esqueci minha senha</a>
        <input type="submit" id="login_submit" name="login_submit" value="Login" /></p>
        <div id="status"></div>
    </form>
</div>
<script type="text/javascript">
$(function() {
    $.getScript('<?php echo URI_JS; ?>jquery.md5.js', init);
});

function init() {

```

```

/* Form submission (AJAX + JSON)
*****/

//Define the options (functions) to handle the submit and response
var options = {
  beforeSend: function() {
    $('#error_box').css('visibility', 'hidden'); //Hide error messages (if exists)
    $('#status').html('<span class="loading"></span>'); //AJAX loading gif
  },
  beforeSend: function() {
    $('#login_senha_fake').attr('disabled', true);
    $('#login_senha').val($.md5($('#login_senha_fake').val()));
  },
  success: function(response, status) {
    $('#error_box').remove(); //Remove error messages (if exists)
    //If no errors occurred, print the success message
    if (response && response.error == null) {
      $('#status').html('<div class="success_box"><span
class="success">Autenticado!</span></div>');
      window.location.reload(true);
    }
    else {
      $('#login_senha_fake').attr('disabled', false);
      if (!response) {
        response = {};
        response.error = 'Ocorreu um erro inesperado';
      }
      //Print the error message
      $('#status').html('<div class="error_box"><span
class="error">'+response.error+'</span></div>');
    }
  },
  error: function(XMLHttpRequest, textStatus, errorThrown) {
    //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
    $('#status').html('<div class="error_box"><span class="error">'+textStatus+':<br
/>'+XMLHttpRequest.responseText+'</span></div>');
    $('#login_senha_fake').attr('disabled', false);
  },
  dataType: 'json'
};

$('#login_usuario').ajaxForm(options);
}
</script>

```

```

usuario/public/painel/index.php

```

```

<?php
require_once(dirname(__FILE__) . '/../../application/include/TemplateHandler.php');

$template = new TemplateHandler();
//Arquivos CSS a serem utilizados na página:
$template->set_css_files(array(
    'painel.css',
    'jplayer.blue.monday.css',
    'jquery.dataTables.css'
));

```

```
//Arquivos Javascript a serem utilizados na página:
$template->set_js_files(array(
    'jquery.jplayer.min.js',
    'jquery.form.js',
    'jquery.dataTables.min.js',
    'jquery.loadTable.js'
));

//Arquivo contendo o conteúdo principal:
$template->set_content_file('painel/painel.php');
//ID da seção (página) para ser destacado no menu:
$template->set_active_nav_item(ID_PAINEL);
//Somente visível para usuário autenticados:
$template->set_authenticated_only(true);
//Exibir a página:
$template->show();
?>
```

usuario/public/painel/painel.php

```
<?php
require_once(dirname(__FILE__) . '/../..//application/include/Auth.php');
require_once(dirname(__FILE__) . '/../..//application/include/DB.php');
require_once(dirname(__FILE__) . '/../..//application/include/LDAP.php');

if (!Auth::check()) {
    exit('Acesso negado');
}

$DB = DB::getInstance();
$LDAP = LDAP::getInstance();
?>

<div id="status_left">
    <div id="dados_ramal" class="rounded">
        <h3>Dados do ramal</h3>
        <?php
            $filtro = "(cn={$_SESSION['ramal']})";
            $camposResposta =
array('contact', 'mail', 'ou', 'departamento', 'l', 'activationDate', 'yellowpages',
            'permission', 'servicecode', 'creditos', 'creditosValorReposicao',
'userPassword');

            $base_dn = 'ou=sip';
            $resposta = $LDAP->search($base_dn, $filtro, $camposResposta);

            for ($i=0; $i<$resposta['count']; $i++) {
                $usuario = $resposta[$i]['contact'][0];
                $email = $resposta[$i]['mail'][0];
                $unidade = $resposta[$i]['ou'][0];
                $departamento = $resposta[$i]['departamento'][0];
                $local = $resposta[$i]['l'][0];
                $data_ativacao = $resposta[$i]['activationdate'][0];
                $paginas_amarelas = $resposta[$i]['yellowpages'][0];
                $permissoes = $resposta[$i]['permission'][0];
                $servicos = $resposta[$i]['servicecode'][0];
                $creditos = isset($resposta[$i]['creditos'][0]) ? $resposta[$i]['creditos'][0] : '';
                $creditos_recarga = $resposta[$i]['creditosvalorreposicao'][0];
                $senha = $resposta[$i]['userpassword'][0];
            }
        }
    }

```

```

?>
<p>
    <strong>Ramal</strong>
    <em><?php echo $_SESSION['ramal']; ?></em>
</p>
<p>
    <strong>Usuário</strong>
    <em><?php echo $usuario; ?></em>
</p>
<p>
    <strong>E-mail</strong>
    <em><?php echo $email; ?></em>
</p>
<p>
    <strong>Unidade</strong>
    <em><?php echo $unidade; ?></em>
</p>
    <strong>Departamento</strong>
    <em><?php echo $departamento; ?></em>
</p>
<p>
    <strong>Local de instalação</strong>
    <em><?php echo $local; ?></em>
</p>
<p>
    <strong>Data de ativação</strong>
    <em><?php echo $data_ativacao; ?></em>
</p>
<p>
    <strong>Visível nas páginas amarelas</strong>
    <em><?php echo ($paginas_amarelas) ? 'Sim' : 'Não'; ?></em>
</p>
<p>
    <strong>Créditos mensais</strong>
    <em>
        <?php
            if (doubleval($creditos_recarga)) {
                echo "R$$creditos_recarga";
            }
            else {
                echo '-';
            }
            ?>
        </em>
    </p>
</div>
</div>

<div id="status_right">
    <div id="utilizacao" class="rounded">
        <h3>Utilização</h3>
        <?php
            $query = sprintf('SELECT COUNT(*) as total
                FROM completed_calls
                WHERE _from = %d',
                    mysql_real_escape_string($_SESSION['ramal']));
            $response = $DB->query($query);

```



```

    $chamadas_efetuadas = mysql_result($response, 0, 'total');

    $query = sprintf('SELECT COUNT(*) as total
                    FROM completed_calls
                    WHERE _to = %d',
                    mysql_real_escape_string($_SESSION['ramal']));
    $response = $DB->query($query);
    $chamadas_recebidas = mysql_result($response, 0, 'total');
    ?>
<p>
    <strong>Chamadas efetuadas</strong>
    <em><?php echo $chamadas_efetuadas; ?></em>
</p>
<p>
    <strong>Chamadas recebidas</strong>
    <em><?php echo $chamadas_recebidas; ?></em>
</p>
<p>
    <strong>Créditos restantes</strong>
    <em>
        <?php
        if (doubleval($creditos)) {
            echo "R$$creditos";
        }
        else {
            echo '-';
        }
        ?>
    </em>
</p>
</div>

<div id="status_atual" class="rounded">
    <?php include(dirname(__FILE__) . '/status_atual.php'); ?>
</div>

<div id="correio_voz">

    <div id="jquery_jplayer"></div>

    <div class="jp-playlist-player">
        <h3>Correio de voz</h3>
        <div class="jp-interface">
            <ul class="jp-controls">
                <li><a href="#" id="jplayer_play" class="jp-play" tabindex="1">play</a></li>
                <li><a href="#" id="jplayer_pause" class="jp-pause"
tabindex="1">pause</a></li>
                <li><a href="#" id="jplayer_stop" class="jp-stop" tabindex="1">stop</a></li>
                <li><a href="#" id="jplayer_volume_min" class="jp-volume-min" tabindex="1">min
volume</a></li>
                <li><a href="#" id="jplayer_volume_max" class="jp-volume-max" tabindex="1">max
volume</a></li>
                <li><a href="#" id="jplayer_previous" class="jp-previous"
tabindex="1">previous</a></li>
                <li><a href="#" id="jplayer_next" class="jp-next" tabindex="1">next</a></li>
            </ul>
            <div class="jp-progress">

```

```

        <div id="jplayer_load_bar" class="jp-load-bar">
            <div id="jplayer_play_bar" class="jp-play-bar"></div>
        </div>
    </div>
    <div id="jplayer_volume_bar" class="jp-volume-bar">
        <div id="jplayer_volume_bar_value" class="jp-volume-bar-value"></div>
    </div>
    <div id="jplayer_play_time" class="jp-play-time"></div>
    <div id="jplayer_total_time" class="jp-total-time"></div>
</div>

<div id="jplayer_playlist" class="jp-playlist">
<?php

//TODO: Colocar negrito nas que não foram ouvidas ainda

$DB->connect_to_midia();
$query = sprintf('SELECT id,
                  SUBSTRING(dir, -5) AS dir,
                  callerid AS origem,
                  FROM_UNIXTIME(origtime, "%d/%m/%Y %H:%i:%s") AS data
                  FROM voicemessages
                  WHERE mailboxuser=%d
                  ORDER BY origtime DESC
                  LIMIT 100',
                  mysql_real_escape_string($_SESSION['ramal']));

try {
    $response = $DB->query($query);

    $playlist = array();
    $i = 0;
    if (mysql_num_rows($response) == 0) {
        echo '<p style="text-align: center">Nenhum recado no correio de voz</p>';
    }
    while ($row = mysql_fetch_array($response)) {
        if (preg_match('/<([\d]*)>/', $row['origem'], $parts)) {
            $row['origem'] = $parts[1];
        }
        elseif (preg_match('/\"([\w]*)\"/', $row['origem'], $parts)) {
            $row['origem'] = $parts[1];
        }
        $playlist[$i]['id'] = $row['id'];
        $playlist[$i]['name'] = $row['data'] .
            '<span class="correio_voz_origem">' . $row['origem'] .

'</span>';

        $playlist[$i]['actions'] = '<span class="correio_voz_acoes">' .
            '' .
            '' .
            '</span>';

            //'<span class="correio_voz_duracao">' . $row['duracao'] .

'</span>';

        $playlist[$i]['mp3'] = '../correio_voz/?action=download&id='.$row['id'].'&format=mp3';
        $playlist[$i]['old'] = true;

```

```

        if ($row['dir'] == 'INBOX') {
            $playlist[$i]['old'] = false;
        }
        $i++;
    }
    $playlist = json_encode($playlist);
}
catch (Exception $e) {
    echo '<p style="text-align: center"><span class="warning">Erro ao pesquisar no banco de
dados</span></p>';
    $playlist = '[]';
}

?>
<ul>
    <!-- The function displayPlayList() uses this unordered list -->
    <li></li>
</ul>
</div>
</div>
</div>

<div id="mensagens" class="rounded">
    <h3>Mensagens</h3>
    <form action="<?php echo URI_SMS; ?>?action=del">
        <table id="lista_mensagens" class="disable_first">
            <thead>
                <tr>
                    <th><input type="checkbox" name="check_all" id="check_all" /></th>
                    <th>Data de envio</th>
                    <th>Origem</th>
                    <th>Conteúdo</th>
                </tr>
            </thead>
            <tbody>
                <tbody>
            </tbody>
        </table>
    </form>
</div>
</div>

<div class="clear"></div>

<script type="text/javascript">
function registro(id) {
    $('#registros').each(function() {
        if ($(this).attr('id') == 'registro_' + id) {
            $(this).show();
        }
        else {
            $(this).hide();
        }
    });
    $('#status_atual .active_page').removeClass('active_page');
    $('#paginacao_' + id).addClass('active_page');
}

function excluir_recado(id) {
    $.get('../correiho_voz/?action=delete&id=' + id, function() {

```

```

        $('#listItem'+id).fadeOut();
    });
    return false;
    //window.location = '../correio_voz/?action=delete&id='+id;
}

function download_recado(id) {
    window.location = '../correio_voz/?action=download&id='+id+'&format=mp3';
}

$(function(){

    /* Mensagens
    *****/

    $('#lista_mensagens').loadTable({
        allowCreate: false,
        sAjaxSource: '<?php echo URI_SMS; ?>?action=get',
        aoColumns: [
            { 'sWidth': '30px', 'sName': 'id', 'bSortable': false },
            { 'sName': 'date', 'sWidth': '150px' },
            { 'sName': 'from' },
            { 'sName': 'content' }
        ],
        aaSorting: [[1,'desc']],
        sPaginationType: 'two_button',
        sDom: '<"top">t<"bottom"p>r<"clear">',
        iDisplayLength: 10,
        fileEditForm: '<?php echo URI_SMS; ?>?action=view',
        boxWidth: '785px',
        boxHeight: '220px',
        delSuccessMsg: 'Mensagens excluídas!',
        fnInitComplete: function() {
            $('#lista_mensagens_wrapper .bottom').prepend('<button id="enviar_sms">Enviar
mensagem</button> ');
            $('#enviar_sms').colorbox({
                href: '<?php echo URI_SMS; ?>',
                width: '430px'
            });
        }
    });

    /* Correio de voz
    *****/

    var playItem = 0;
    var myPlayList = <?php echo $playlist; ?>;

    // Local copy of jQuery selectors, for performance.
    var jpPlayTime = $("#jplayer_play_time");
    var jpTotalTime = $("#jplayer_total_time");

    $("#jquery_jplayer").jPlayer({
        ready: function() {
            displayPlayList();
            playListInit(false); // Parameter is a boolean for autoplay.
        },
        swfPath: "../_js"
    });
}

```

```

    })
    .jPlayer("onProgressChange", function(loadPercent, playedPercentRelative, playedPercentAbsolute,
playedTime, totalTime) {
        jpPlayTime.text($.jPlayer.convertTime(playedTime));
        jpTotalTime.text($.jPlayer.convertTime(totalTime));
    })
    .jPlayer("onSoundComplete", function() {
        //playListNext();
    });

$("#jplayer_previous").click(function() {
    playListPrev();
    $(this).blur();
    return false;
});

$("#jplayer_next").click(function() {
    playListNext();
    $(this).blur();
    return false;
});

$("#jplayer_play").click(function() {
    $('.jplayer_playlist_current').css('font-weight', 'normal');
});

function displayPlayList() {
    $("#jplayer_playlist ul").empty();
    for (i=0; i < myPlayList.length; i++) {
        var listItem = (i == myPlayList.length-1) ? "<li id='listItem'+myPlayList[i].id+'
class='jplayer_playlist_item_last'>" : "<li id='listItem'+myPlayList[i].id+'>";
        if (myPlayList[i].old) {
            listItem += "<a href='#' id='jplayer_playlist_item_"+i+"' tabindex='1'>"+
myPlayList[i].name + "</a>"+ myPlayList[i].actions + "<div class='clear'></div></li>";
        }
        else {
            listItem += "<a href='#' id='jplayer_playlist_item_"+i+"' tabindex='1' style='font-weight:
bold'>"+ myPlayList[i].name + "</a>"+ myPlayList[i].actions + "<div class='clear'></div></li>";
        }
    }
    $("#jplayer_playlist ul").append(listItem);
    $("#jplayer_playlist_item_"+i).data("index", i).click(function() {
        var index = $(this).data("index");
        if (playItem != index) {
            playListChange(index);
        } else {
            $(this).css('font-weight', 'normal');
            $("#jquery_jplayer").jPlayer("play");
        }
        $(this).blur();
        return false;
    });
}

function playListInit(autoplay) {
    if (autoplay) {
        playListChange(playItem);
    } else {

```

```

        playListConfig(playItem);
    }
}

function playListConfig(index) {

$("#jplayer_playlist_item_"+playItem).removeClass("jplayer_playlist_current").parent().removeClass("jplayer_playlist_current");

$("#jplayer_playlist_item_"+index).addClass("jplayer_playlist_current").parent().addClass("jplayer_playlist_current");
    playItem = index;
    $("#jquery_jplayer").jPlayer("setFile", myPlayList[playItem].mp3);
}

function playListChange(index) {
    playListConfig(index);
    $("#jplayer_playlist_item_"+index).css('font-weight', 'normal');
    $("#jquery_jplayer").jPlayer("play");
}

function playListNext() {
    var index = (playItem+1 < myPlayList.length) ? playItem+1 : 0;
    playListChange(index);
}

function playListPrev() {
    var index = (playItem-1 >= 0) ? playItem-1 : myPlayList.length-1;
    playListChange(index);
}

/* Status atual
*****

var refreshIdStatus = setInterval(function() {
    $("#status_atual").load('<?php echo URI_PAINEL; ?>status_atual.php?randval='+ Math.random());
}, 5000);

function secondsToTime(d) {
    d = Number(d);
    var h = Math.floor(d / 3600);
    var m = Math.floor(d % 3600 / 60);
    var s = Math.floor(d % 3600 % 60);
    return ((h > 0 ? (h < 10 ? "0" : "") + h + ":" : "00:") + (m > 0 ? (m < 10 ? "0" : "") + m + ":" :
"00:") + (s < 10 ? "0" : "") + s);
}

function timeToSeconds(time) {
    time = time.split(':');
    if (time.length == 3) {
        var h = Math.floor(time[0] * 3600);
        var m = Math.floor(time[1] * 60);
        var s = Math.floor(time[2]);
        return h+m+s;
    }
    return false;
}
}

```

```

var refreshIdTempoLigacao = setInterval(function() {
    if ($('#tempo_ligacao').text() {
        var secs = timeToSeconds($('#tempo_ligacao').text());
        if (secs) {
            secs++;
            $('#tempo_ligacao').text(secondsToTime(secs));
        }
    }
}, 1000);

$('#enviar_sms').ajaxForm();

});
</script>

```

usuario/public/painel/status_atual.php

```

<?php
require_once(dirname(__FILE__) . '/../application/include/Auth.php');
require_once(dirname(__FILE__) . '/../application/include/DB.php');

if (!Auth::check()) {
    exit('Acesso negado');
}

$DB = DB::getInstance();
$DB->connect();

$query = sprintf('SELECT contact, user_agent, cflags, callid
                FROM voip.location
                WHERE username = %d',
                mysql_real_escape_string($_SESSION['ramal']));

$response = $DB->query($query);

if (mysql_num_rows($response) == 0) {
    echo '<h3>Status atual: &nbsp;<span class="warning">Inativo</span></h3>';
    echo '<p class="aviso">Seu ramal não está registrado na central PBX</p>';
}
else {
    echo '<h3>Status atual: &nbsp;<span class="success">Ativo</span></h3>';

    $query = sprintf("SELECT from_uri, to_uri,
                    TIMEDIFF(NOW(), FROM_UNIXTIME(start_time)) AS tempo_ligacao
                    FROM voip.dialog
                    WHERE SUBSTR(from_uri, 5, 4) = %d
                    OR SUBSTR(SUBSTRING_INDEX(from_uri, '@', 1), -8) = %d
                    OR SUBSTR(to_uri, 5, 4) = %d
                    OR SUBSTR(SUBSTRING_INDEX(to_uri, '@', 1), -8) = %d
                    ORDER BY from_uri DESC",
                    mysql_real_escape_string($_SESSION['ramal']),
                    mysql_real_escape_string(PREFIXO . $_SESSION['ramal']),
                    mysql_real_escape_string($_SESSION['ramal']),
                    mysql_real_escape_string(PREFIXO . $_SESSION['ramal']));

    $response_dialog = $DB->query($query);
    $from_list = array(); //As vezes o from aparece duplicado no banco de dados

```

```

while ($row = mysql_fetch_array($response_dialog)) {
    $from_uri = $row['from_uri'];
    $to_uri = $row['to_uri'];
    $tempo_ligacao = $row['tempo_ligacao'];

    preg_match('/sip:([\d]+)@/', $from_uri, $from_parts);
    $from = $from_parts[1];

    preg_match('/sip:([\d]+)@/', $to_uri, $to_parts);
    $to = $to_parts[1];

    //Se o usuário estiver efetuado uma ligação:
    if ($from == $_SESSION['ramal'] || substr($from, -8) == PREFIXO . $_SESSION['ramal']) {
        echo "<p>Ligando para: <b>$to <span
id=\"tempo_ligacao\">$tempo_ligacao</span></b></p><hr />";
    }
    //Se o usuário estiver recebendo uma ligação:
    elseif (!isset($from_list[PREFIXO . $from]) && !isset($from_list[substr($from, -8)]) &&
        ($to == $_SESSION['ramal'] || substr($to, -8) == PREFIXO . $_SESSION['ramal'])) {
        echo "<p>Recebendo chamada de: <b>$from <span
id=\"tempo_ligacao\">$tempo_ligacao</span></b></p><hr />";
    }

    if (strlen($from) == 4) {
        $from_list[PREFIXO . $from] = PREFIXO . $from;
    }
    else {
        $from_list[substr($from, -8)] = substr($from, -8);
    }
}
$registros=0;
while ($row = mysql_fetch_array($response)) {
    $contact = $row['contact'];
    $contact = explode('@', $contact);
    $contact = explode(':', $contact[1]);
    $contact = $contact[0];
    $user_agent = $row['user_agent'];
    $nat = explode('@', $row['callid']);
    $nat = isset($nat[1]) ? $nat[1] : $nat[0];
    $cflags = $row['cflags'];

    $style = 'style="display: none"';
    if ($registros == 0) {
        $style = 'style="display: block"';
    }
    echo '<div class="registros" id="registro_'. $registros. '" '. $style. '>';
    echo '<p>';
    echo '<strong>Endereço IP</strong>';
    echo "<em>$contact</em>";
    echo '</p>';
    if (intval($cflags) > 0) {
        echo '<p>';
        echo '<strong>NAT</strong>';
        echo "<em>$nat ($cflags)</em>";
        echo '</p>';
    }
    echo '<p>';
    echo '<strong>Firmware</strong>';

```



```

        echo "<em>$user_agent</em>";
        echo '</p>';
        echo '</div>';
        $registros++;
    }
    echo '<div class="pagination-wrapper">';
    echo '<ul class="pagination">';
    for ($i=1; $i<$registros; $i++) {
        if ($i == 1) {
            echo '<li class="active_page" id="paginacao_'. $i .'>';
        }
        else {
            echo '<li id="paginacao_'. $i .'>';
        }
        echo '<a href="javascript:registro(.'. $i .' )">' . $i . '</a></li>';
    }
    echo '</ul></div>';
}
?>

```

usuario/public/redefinicao/index.php

```

<?php
require_once(dirname(__FILE__) . '/../..//application/include/Auth.php');
require_once(dirname(__FILE__) . '/../..//application/include/FirePHPCore/fb.php');
require_once(dirname(__FILE__) . '/../..//application/include/DB.php');
require_once(dirname(__FILE__) . '/../..//application/include/LDAP.php');

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    //Passo 1: Informar o ramal
    case '1':

        $DB = DB::getInstance();
        $LDAP = LDAP::getInstance();
        $DB->connect();

        $ramal = isset($_REQUEST['ramal']) ? $_REQUEST['ramal'] : '';

        if (!$ramal) {
            exit(json_encode(array('error' => 'Ramal não especificado')));
        }
        if (!preg_match('/[0-9]{4}/', $ramal)) {
            exit(json_encode(array('error' => 'Ramal inválido')));
        }

        $dn = 'ou=sip';
        $filter = "(cn=$ramal)";
        $fields = array('userpassword', 'mail');

        try {
            $response = $LDAP->search($dn, $filter, $fields);
        }
        catch (Exception $e) {
            exit(json_encode(array('error' => 'Erro no banco de dados. Tente novamente.')));
        }
    }
}

```

```

if ($response['count'] == 0) {
    exit(json_encode(array('error' => 'Ramal não cadastrado')));
}

session_start();

for ($i=0; $i<$response['count']; $i++) {
    $_SESSION['senha'] = $response[$i]['userpassword'][0];
    $_SESSION['email'] = $response[$i]['mail'][0];
}

$_SESSION['ramal'] = $ramal;
$_SESSION['codigo_ura'] = sprintf('%05d', mt_rand(0, 99999));
$_SESSION['codigo_email'] = sprintf('%05d', mt_rand(0, 99999));

$query = sprintf("REPLACE INTO usuarios_recuperacao_senha
                SET `ramal`='%s', `codigo_ura`='%s', `codigo_email`='%s',
                    `data_geracao`=NOW(), `status`=1, `email`='%s'",
                $_SESSION['ramal'],
                $_SESSION['codigo_ura'],
                $_SESSION['codigo_email'],
                $_SESSION['email']);

try {
    $DB->query($query);
}
catch (Exception $e) {
    exit(json_encode(array('error' => 'Erro no banco de dados. Tente novamente.')));
}

exit(json_encode(array('error' => null)));
break;

//Passo 3: Informar o código recebido no email
case '3':

    $DB = DB::getInstance();
    session_start();

    $codigo_email = isset($_REQUEST['codigo']) ? $_REQUEST['codigo'] : '';
    $ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
    $codigo_email_session = isset($_SESSION['codigo_email']) ? $_SESSION['codigo_email'] : '';

    if (!$ramal) {
        exit(json_encode(array('error' => 'Ramal não especificado')));
    }
    if (!$codigo_email) {
        exit(json_encode(array('error' => 'Informe o código')));
    }
    if (!preg_match('/[0-9]{5}/', $codigo_email)) {
        exit(json_encode(array('error' => 'Código inválido')));
    }

    if ($codigo_email_session != $codigo_email) {
        exit(json_encode(array('error' => 'Código não confere')));
    }

    $query = sprintf("SELECT `codigo_email`

```

```

        FROM usuarios_recuperacao_senha
        WHERE `ramal` = '%s'
            AND `codigo_email` = '%s'
            AND `status` >= 3
            AND `data_geracao` < DATE_ADD(NOW(), INTERVAL 30 MINUTE)",
        $ramal,
        $codigo_email);

    try {
        $result = $DB->query($query);
        if (mysql_num_rows($result) == 0) {
            exit(json_encode(array('error' => 'Código expirou. Por favor, reinicie o processo')));
        }
        $query = sprintf("UPDATE usuarios_recuperacao_senha
            SET `status` = 5
            WHERE `ramal` = '%s'",
            $ramal);
        $DB->query($query);
    }
    catch (Exception $e) {
        exit(json_encode(array('error' => 'Erro no banco de dados. Tente novamente.')));
    }

    exit(json_encode(array('error' => null)));
    break;

//Passo 4: Cadastrar a nova senha de acesso ao portal
case '4':

    $DB = DB::getInstance();
    $LDAP = LDAP::getInstance();
    session_start();

    $ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
    $senha = isset($_REQUEST['senha']) ? $_REQUEST['senha'] : '';
    $confirma_senha = isset($_REQUEST['confirma_senha']) ? $_REQUEST['confirma_senha'] : '';

    if (!$ramal) {
        exit(json_encode(array('error' => 'Ramal não especificado')));
    }
    if ($senha != $confirma_senha) {
        exit(json_encode(array('error' => 'Senhas não conferem')));
    }
    if (strlen($senha) < 4) {
        exit(json_encode(array('error' => 'A senha deve conter 4 caracteres ou mais')));
    }

    $query = sprintf("SELECT `status`
        FROM usuarios_recuperacao_senha
        WHERE `ramal` = '%s'
            AND `status` = 5
            AND `data_geracao` < DATE_ADD(NOW(), INTERVAL 30 MINUTE)",
        $ramal);

    try {
        $result = $DB->query($query);
        if (mysql_num_rows($result) == 0) {
            exit(json_encode(array('error' => 'Código expirou. Por favor, reinicie o processo')));
        }
    }

```

```

    }

    $arrayRamalInfo = array();
    $arrayRamalInfo['senhaWeb'] = md5($senha);

    $dn = "cn=$ramal,ou=sip";
    $LDAP->update($dn, $arrayRamalInfo);

    $query = sprintf("UPDATE usuarios_recuperacao_senha
                     SET `status` = 6
                     WHERE `ramal` = '%s'",
                     $ramal);
    $DB->query($query);

    Auth::login($ramal, md5($senha));
}
catch (Exception $e) {
    exit(json_encode(array('error' => 'Erro no banco de dados. Tente novamente.')));
}

exit(json_encode(array('error' => null)));
break;

//Imprime a imagem do email, usada para verificar a leitura
case 'get_image':

    session_start();

    $DB = DB::getInstance();
    $ramal = isset($_REQUEST['ramal']) ? $_REQUEST['ramal'] : '';

    if ($ramal) {

        $query = sprintf("UPDATE usuarios_recuperacao_senha
                         SET `status` = 4
                         WHERE `ramal` = '%s'
                         AND `status` = 3
                         AND `data_geracao` < DATE_ADD(NOW(), INTERVAL 30 MINUTE)",
                         $ramal);

        try {
            $DB->query($query);
        }
        catch (Exception $e) {
            exit($e->getMessage());
        }
    }

    header("Content-type: image/png");
    $im = @imagecreate(30, 10) or die("Cannot Initialize new GD image stream");
    $background_color = imagecolorallocate($im, 255, 255, 255);
    $text_color = imagecolorallocate($im, 0, 0, 0);
    imagestring($im, 1, 0, 0, "PBX-IP", $text_color);
    imagepng($im);
    imagedestroy($im);

    exit();
    break;

```

```

case 'cancel':

    @session_start();
    @session_unset();
    @session_destroy();
    //header('Location: ' . URI_REDEFINICAO);
    header('Location: ' . URI_RAIZ);
    exit();
    break;

default:

    if (Auth::check()) {
        header('Location: ' . URI_RAIZ);
        exit();
    }

    require_once(dirname(__FILE__) . '/../../application/include/TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
        'public.css',
        'recuperacao.css'
    ));
    $template->set_js_files(array(
        'jquery.form.js'
    ));
    $template->set_content_file('redefinicao/redefinicao.php');
    $template->set_authenticated_only(false);
    $template->show();
}
?>

```

usuario/public/redefinicao/passol.php

```

<form id="passo_1" action="?action=1" method="post">
    <fieldset>
        <legend>Informe seu ramal</legend>
        <br />
        <input type="text" name="ramal" id="ramal" class="title" maxlength="4" />
        <em>Preencha o número do seu ramal com 4 dígitos</em>
    </fieldset>
    <input type="submit" value="Avançar" />
    <div id="status"></div>
</form>

<script type="text/javascript">
$(function() {
    var options = {
        beforeSubmit: function() {
            $('#status').html('<span class="loading"></span>'); //AJAX loading gif
        },
        success: function(response, status) {
            //If no errors occurred, print the success message
            if (response && response.error == null) {
                $('#redefinicao_senha').load('passo_2.php');
            }
            else {
                if (!response) {

```

```

        response = {};
        response.error = 'Ocorreu um erro inesperado';
    }
    //Print the error message
    $('#status').html('<span class="error">'+response.error+'</span>');
}
},
error: function(XMLHttpRequest, textStatus, errorThrown) {
    //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
    $('#status').html('<span class="error">'+textStatus+':<br
/>'+XMLHttpRequest.responseText+'</span>');
},
dataType: 'json'
});

$('#passo_1').ajaxForm(options);
});
</script>

```

usuario/public/redefinicao/passos2.php

```

<?php
require_once(dirname(__FILE__) . '/../application/config/general.php');
require_once(dirname(__FILE__) . '/../application/include/DB.php');

session_start();

$codigo_ura = isset($_SESSION['codigo_ura']) ? $_SESSION['codigo_ura'] : '';
$ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
$ura_reset_pw = str_replace('XXXX', $ramal, URA_RESET_PW);
$senha = isset($_SESSION['senha']) ? $_SESSION['senha'] : '';

if (!$codigo_ura) {
    ?>
    <h4>O código da URA não foi gerado!</h4>
    <p><a href="<?php echo URI_REDEFINICAO; ?>?action=cancel">Clique aqui</a> para reiniciar o
processo.</p>
    <?php
        exit();
    }

if (!$ramal || !$senha) {
    ?>
    <h4>Seu ramal não foi especificado!</h4>
    <p><a href="<?php echo URI_REDEFINICAO; ?>?action=cancel">Clique aqui</a> para reiniciar o
processo.</p>
    <?php
        exit();
    }

$DB = DB::getInstance();
$query = sprintf("UPDATE usuarios_recuperacao_senha
                SET `status`=1
                WHERE `ramal`='%s'",
                $ramal);
try {
    $DB->query($query);
}

```

```

}
catch (Exception $e) {

}

$file = realpath(dirname(__FILE__) . '/../..../application/scripts/recuperacao_dial.php');
$args = "$ramal $ura_reset_pw $senha";

$cmd = "/usr/bin/php -f $file $args";

run_in_background($cmd);
?>

<h4>Código de verificação (URA):</h4>
<h1><?php echo $_SESSION['codigo_ura']; ?></h1>
<br />
<em>Após atender a chamada, digite 1 para confirmar a recuperação de senha e em seguida informe o código acima.</em>
<br />
<br />
<p>Não recebeu a chamada? <a href="<?php echo URI_REDEFINICAO; ?>">Clique aqui</a> para tentar novamente ou disque para <b><?php echo $ura_reset_pw; ?></b></p>

```

```

usuario/public/redefinicao/passo3.php
<?php
session_start();

$codigo_email = isset($_SESSION['codigo_email']) ? $_SESSION['codigo_email'] : '';
$ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
$email = isset($_SESSION['email']) ? $_SESSION['email'] : '';

if (!$codigo_email) {
?>
    <h4>O código do email não foi gerado!</h4>
    <p><a href="<?php echo URI_REDEFINICAO; ?>?action=cancel">Clique aqui</a> para reiniciar o processo.</p>
<?php
    exit();
}

if (!$ramal) {
?>
    <h4>Seu ramal não foi especificado!</h4>
    <p><a href="<?php echo URI_REDEFINICAO; ?>?action=cancel">Clique aqui</a> para reiniciar o processo.</p>
<?php
    exit();
}
?>

<form id="passo_3" action="<?php echo URI_REDEFINICAO; ?>?action=3" method="post">
    <fieldset>
        <legend>Código de verificação (Email)</legend>
        <br />
        <em>Preencha o código de 5 dígitos enviado ao seu email (<?php echo $email; ?>)</em>
        <p><input type="text" name="codigo" id="codigo" class="title" maxlength="5" /></p>
    </fieldset>

```

```

<input type="submit" value="Avançar" />
<div id="status"></div>
</form>

<script type="text/javascript">
$(function() {
    var options = {
        beforeSend: function() {
            $('#status').html('<span class="loading"></span>'); //AJAX loading gif
        },
        success: function(response, status) {
            //If no errors occurred, print the success message
            if (response && response.error == null) {
                $('#redefinicao_senha').load('passo_4.php');
            }
            else {
                if (!response) {
                    response = {};
                    response.error = 'Ocorreu um erro inesperado';
                }
                //Print the error message
                $('#status').html('<span class="error">'+response.error+'</span>');
            }
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            //Print the error message if the request fails (possible reasons: "timeout", "error",
            "notmodified" and "parsererror")
            $('#status').html('<span class="error">'+textStatus+':<br
            />'+XMLHttpRequest.responseText+'</span>');
        },
        dataType: 'json'
    };

    $('#passo_3').ajaxForm(options);
});
</script>

```

usuario/public/redefinicao/passo4.php

```

<?php
require_once(dirname(__FILE__) . '/../../application/config/general.php');
?>
<form id="passo_4" action="?action=4" method="post">
    <fieldset>
        <legend>Nova senha</legend>
        <br />
        <em>Insira a sua nova senha de acesso ao portal do usuário</em>
        <p style="float: left"><label for="senha">Senha</label><br />
        <input type="password" name="senha" id="senha" class="title" style="width: 170px"
        autocomplete="off" /></p>
        <p style="float: left; margin-left: 15px"><label for="confirma_senha">Confirme a senha</label><br
        />
        <input type="password" name="confirma_senha" id="confirma_senha" class="title" style="width: 170px"
        autocomplete="off" /></p>
    </fieldset>
    <input type="submit" value="Avançar" />
    <div id="status"></div>
</form>

```



```

<script type="text/javascript">
$(function() {
    var options = {
        beforeSend: function() {
            $('#status').html('<span class="loading"></span>'); //AJAX loading gif
        },
        success: function(response, status) {
            //If no errors occurred, print the success message
            if (response && response.error == null) {
                $('#status').html('<span class="success">Senha alterada com sucesso!</span>');
                window.location = '<?php echo URI_RAIZ; ?>';
            }
            else {
                if (!response) {
                    response = {};
                    response.error = 'Ocorreu um erro inesperado!';
                }
                //Print the error message
                $('#status').html('<span class="error">'+response.error+'</span>');
            }
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            //Print the error message if the request fails (possible reasons: "timeout", "error",
            "notmodified" and "parsererror")
            $('#status').html('<span class="error">'+textStatus+':<br
            />'+XMLHttpRequest.responseText+'</span>');
        },
        dataType: 'json'
    };

    $('#passo_4').ajaxForm(options);
});
</script>

```

usuario/public/redefinicao/redefinicao.php

```

<div id="redefinicao_senha"></div>

<div id="right_container">
    <div id="status_redefinicao" class="rounded"></div>
    <input type="button" value="Cancelar" id="cancelar" />
</div>

<?php
require_once(dirname(__FILE__) . '/../../application/include/DB.php');

$ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';

/*
 * Status
 * 0. Início
 * 1. Informou o ramal
 * 2. Atendeu a chamada
 * 3. Informou o código na central de recuperação
 * 4. Verificou o email
 * 5. Informou o código recebido no email
 * 6. Cadastrou a nova senha de acesso ao portal

```

```

*/

$status = 0;

if ($ramal) {
    $DB = DB::getInstance();
    $query = sprintf("SELECT `status`
                    FROM usuarios_recuperacao_senha
                    WHERE `ramal`='%s'",
                    $ramal);

    try {
        $response = $DB->query($query);
        $status = mysql_result($response, 'status');
    }
    catch (Exception $e) {

    }
}

switch ($status) {
    case 0:
        $passo = 1;
        break;

    case 1:
    case 2:
        $passo = 2;
        break;

    case 3:
    case 4:
        $passo = 3;
        break;

    case 5:
    case 6:
        $passo = 4;
        break;

    default:
        $passo = 1;
}

?>

<script type="text/javascript">
var passo = <?php echo $passo; ?>;
$(function() {
    $("#redefinicao_senha").load('passo_<?php echo $passo; ?>.php');
    $("#status_redefinicao").load('status_redefinicao.php');
    $("#cancelar").click(function() {
        window.location = '<?php echo URI_REDEFINICAO; ?>?action=cancel!';
    });
    var refreshId = setInterval(function() {
        $("#status_redefinicao").load('status_redefinicao.php?randval='+ Math.random());
    }, 3000);
});
</script>

```

```

usuario/public/redefinicao/status_redefinicao.php

```

```

<?php

```

```

require_once(dirname(__FILE__) . '/../../application/include/DB.php');

session_start();
$ramal = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
$status = 0;

if ($ramal) {
    $DB = DB::getInstance();
    $query = sprintf("SELECT `status`
                     FROM usuarios_recuperacao_senha
                     WHERE `ramal`='%s'",
                     $ramal);

    try {
        $response = $DB->query($query);
        $status = mysql_result($response, 'status');
    }
    catch (Exception $e) {

    }
}

/*
 * Status
 * 0. Início
 * 1. Informou o ramal
 * 2. Atendeu a chamada
 * 3. Informou o código na central de recuperação
 * 4. Verificou o email
 * 5. Informou o código recebido no email
 * 6. Cadastrou a nova senha de acesso ao portal
 */

switch ($status) {
    case 0:
        $passo = 1;
        break;
    case 1:
    case 2:
        $passo = 2;
        break;
    case 3:
    case 4:
        $passo = 3;
        break;
    case 5:
    case 6:
        $passo = 4;
        break;
    default:
        $passo = 1;
}

?>
<h3>Status da redefinição</h3>
<ol>
    <li <?php if ($status < 0) { echo 'class="disabled"'; } elseif ($status > 0) { echo
'class="completed"'; } ?>>Informar o ramal</li>
    <li <?php if ($status < 1) { echo 'class="disabled"'; } elseif ($status > 1) { echo
'class="completed"'; } ?>>Atender a chamada</li>

```

```

    <li <?php if ($status < 2) { echo 'class="disabled"'; } elseif ($status > 2) { echo
'class="completed"'; } ?>>Informar o código na central de recuperação</li>
    <li <?php if ($status < 3) { echo 'class="disabled"'; } elseif ($status > 3) { echo
'class="completed"'; } ?>>Verificar o email</li>
    <li <?php if ($status < 4) { echo 'class="disabled"'; } elseif ($status > 4) { echo
'class="completed"'; } ?>>Informar o código recebido no email</li>
    <li <?php if ($status < 5) { echo 'class="disabled"'; } elseif ($status > 5) { echo
'class="completed"'; } ?>>Cadastrar a nova senha de acesso ao portal</li>
</ol>
<script type="text/javascript">
var novo_passo = <?php echo $passo; ?>;
if (passo == 2 && novo_passo == 3) {
    $('#redefinicao_senha').load('passo_3.php');
}
passo = novo_passo;
</script>

```

usuario/public/sms/index.php

```

<?php
require_once(dirname(__FILE__) . '/../../application/controllers/ControllerSMS.php');

$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';

switch ($action) {

    case 'send':
        $controller = ControllerSMS::getInstance();
        $ramal = isset($_POST['ramal']) ? $_POST['ramal'] : '';
        $mensagem = isset($_POST['mensagem']) ? $_POST['mensagem'] : '';
        exit($controller->send($ramal, $mensagem));
        break;

    case 'get':
        $_GET['sTable'] = 'messages';
        $controller = ControllerSMS::getInstance();
        exit($controller->getTableData($_GET));
        break;

    case 'del':
        $ids = isset($_REQUEST['ids']) ? $_REQUEST['ids'] : '';
        $controller = ControllerSMS::getInstance();
        exit($controller->del($ids));
        break;

    default:
        include(dirname(__FILE__) . '/sms.php');
}
?>

```

usuario/public/sms/sms.php

```

<?php
$ramal = isset($_REQUEST['ramal']) ? $_REQUEST['ramal'] : '';
?>
<form id="enviar_sms" action="<?php echo URI_SMS; ?>?action=send" method="post">
    <fieldset>
        <legend>Enviar mensagem (SMS)</legend>

```

```

<p><label for="ramal">Ramal</label><br />
<input type="text" id="ramal" name="ramal" maxlength="4" class="title" value="<?php echo $ramal;
?>" style="width: 70px" />
<br /><em>(Ramal VoIP da PBX-IP)</em></p>
<p><label for="mensagem">Mensagem</label><br />
<textarea id="mensagem" name="mensagem" style="width: 325px; height: 70px"></textarea></p>
<em><b>Observação:</b><br />Mensagens poderão visualizadas apenas nos telefones IPs ou no
portal.</em>
<input type="submit" value="Enviar" style="position: absolute; bottom: 0; left: 0; z-index: 9999"
/>
</fieldset>
</form>
<script type="text/javascript">
$(function() {

/* Form submission (AJAX + JSON)
*****/

//Define the options (functions) to handle the submit and response
var options = {
  beforeSubmit: function() {
    $('#error_box').css('visibility', 'hidden'); //Hide error messages (if exists)
    $('#cboxTitle').html('<span class="loading"></span>'); //AJAX loading gif
  },
  success: function(response, status) {
    $('#error_box').remove(); //Remove error messages (if exists)
    //If no errors occurred, print the success message
    if (response && response.error == null) {
      if (response.sent == 0) {
        $('#cboxTitle').html('<span class="success">Mensagem enviada porém não<br />apresentada no
telefone</span>');
      }
      else {
        $('#cboxTitle').html('<span class="success">Mensagem enviada com sucesso!</span>');
      }
    }
    else {
      if (!response) {
        response = {};
        response.error = 'Ocorreu um erro inesperado';
      }
      //Print the error message
      $('#cboxTitle').html('<span class="error">'+response.error+'</span>');
    }
  },
  error: function(XMLHttpRequest, textStatus, errorThrown) {
    //Print the error message if the request fails (possible reasons: "timeout", "error",
"notmodified" and "parsererror")
    $('#cboxTitle').html('<span class="error">'+textStatus+'<br
/>'+XMLHttpRequest.responseText+'</span>');
  },
  dataType: 'json'
};

$("#enviar_sms").ajaxForm(options);
});
</script>

```

```

usuario/public/cancel_caller.php
<?php
require_once(dirname(__FILE__) . '/../application/include/functions.php');

$from = isset($_REQUEST['from']) ? $_REQUEST['from'] : '';
$to = isset($_REQUEST['to']) ? $_REQUEST['to'] : '';

$file = realpath(dirname(__FILE__) . '/../application/scripts/click2call_cancel.php');
$args = "$from $to";

$cmd = "/usr/bin/php -f $file $args";

$ps = run_in_background($cmd);
?>

```

```

usuario/public/dial_dialer.php
<?php
require_once(dirname(__FILE__) . '/../application/include/Auth.php');
require_once(dirname(__FILE__) . '/../application/include/functions.php');

if (!Auth::check()) {
    exit('Acesso negado');
}

$from = isset($_SESSION['ramal']) ? $_SESSION['ramal'] : '';
$to = isset($_REQUEST['to']) ? $_REQUEST['to'] : '';
$password = isset($_SESSION['senha']) ? $_SESSION['senha'] : '';

$file = realpath(dirname(__FILE__) . '/../application/scripts/click2call_dial.php');
$args = "$from $to $password";

$cmd = "/usr/bin/php -f $file $args";

$ps = run_in_background($cmd);
while (is_process_running($ps)) {
    sleep(2);
}
?>

```

```

usuario/public/index.php
<?php
require_once(dirname(__FILE__) . '/../application/include/Auth.php');
if (Auth::check()) {
    header('Location: painel/');
}
else {
    header('Location: login/');
}
?>

```

```

usuario/public/login_form.php
<div id="login_wrapper">
    <br />
    <form d="login_usuario" name="login_usuario" autocomplete="off" action="<?php echo URI_RAIZ .
'login.php?acao=login'; ?>">

```

```

<p><label>Ramal</label>
<input type="text" id="login_ramal" name="login_ramal" class="title"></p>
<p><label>Senha</label>
<input type="password" id="login_senha" name="login_senha" class="title"></p>
<div id="excedeu_tentativas" class="formfield"></div>
<p><input type="submit" id="login_submit" name="login_submit" value="Login"></p>
<div id="status_login"></div>
</form>
</div>

```

```

usuario/public/usuario_contatos.php

```

```

<?php
require_once(dirname(__FILE__) . '/../application/include/Auth.php');
require_once(dirname(__FILE__) . '/../application/include/LDAP.php');
require_once(dirname(__FILE__) . '/../application/controllers/ControllerContacts.php');
if (!Auth::check()) {
    exit('Acesso negado');
}
$DB = DB::getInstance();
$LDAP = LDAP::getInstance();

$search = isset($_REQUEST['search']) ? $_REQUEST['search'] : '';
?>
<div id="contatos">
    <ul>
        <?php

        /* Lista de contatos
        *****/

        $controller = ControllerContacts::getInstance();
        $contatos = $controller->get(null, array('id', 'nome', 'telefone'), null, null, $_SESSION['ramal'],
$search);

        /* Ramais cadastrados
        *****/

        $filter = '(cn=*)';
        $fields = array('cn');
        $dn = 'ou=sip';
        $response = $LDAP->search($dn, $filter, $fields);

        for ($i=0; $i<$response['count']; $i++) {
            $ramais[$response[$i]['cn'][0]] = $response[$i]['cn'][0];
        }

        /* Ramais ativos
        *****/

        $query = 'SELECT username
                FROM voip.location';

        $response = $DB->query($query);
        $ramais_ativos = array();
        while ($row = mysql_fetch_array($response)) {
            $ramais_ativos[$row['username']] = $row['username'];

```

```

}

/* Chamadas ativas
*****/

$query = 'SELECT from_uri, to_uri
        FROM voip.dialog';

$response = $DB->query($query);
$chamadas_ativas = array();
while ($row = mysql_fetch_array($response)) {
    if (preg_match('/sip:([\d]+)@/', $row['from_uri'], $parts)) {
        $chamadas_ativas[$parts[1]] = $parts[1];
    }
    if (preg_match('/sip:([\d]+)@/', $row['to_uri'], $parts)) {
        $chamadas_ativas[$parts[1]] = $parts[1];
    }
}

/* Organiza a lista de contatos
*****/

$lista_contatos = array();
$lista_contatos[1] = array(); //Disponível
$lista_contatos[2] = array(); //Em uso
$lista_contatos[0] = array(); //Inativo
$lista_contatos[4] = array(); //Celular
$lista_contatos[3] = array(); //Convencional
$sem_contatos = true;
if ($contatos) {
    $sem_contatos = false;
    foreach ($contatos as $contato) {
        $prefixo = substr($contato['telefone'], 5, 4);
        $ramal = substr($contato['telefone'], -4);
        $id = $contato['id'];
        if (isset($ramais[$ramal])) {
            $status = 0; //Ramal cadastrado porém inativo
            if (isset($ramais_ativos[$ramal])) {
                $status = 1; //Ramal ativo (registrado)
                if (isset($chamadas_ativas[$ramal])) {
                    $status = 2; //Ramal em uso
                }
            }
        }
        else {
            $status = 3;
            if (preg_match('/[7-9]/', substr($prefixo, 0, 1))) {
                $status = 4;
            }
        }
        $lista_contatos[$status][$id] = $contato['nome'];
    }
}

/* Imprime a lista de contatos
*****/

$count = 0;
foreach ($lista_contatos as $status=>$contatos) {
    asort($contatos);
}

```



```

switch ($status) {
    case 0: $class = 'phone_disabled'; break;
    case 1: $class = 'phone_available'; break;
    case 2: $class = 'phone_busy'; break;
    case 3: $class = 'phone_pbx'; break;
    case 4: $class = 'phone_mobile'; break;
    default: $class = '';
}

foreach ($contatos as $id=>$contato) {
    if ($count == $_SESSION['numero_max_contatos']) {
        break;
    }
    echo '<li class="'. $class .'" id="'. $id .'" name="'. $status .'"><a
href="#">'. $contato .'\</a></li>';
    $count++;
}
}
?>
</ul>
<?php
if ($sem_contatos) {
echo '<div id="sem_contatos">';
echo '<h4>Você não adicionou nenhum contato!\</h4><br />';
echo '<em><a href="'. URI_CONTATOS .'">Clique aqui</a> para gerenciar sua lista.</em>';
echo '</div>';
}
?>
</div>
<script type="text/javascript">
$(function() {
    $('#contatos ul li').qtip({
        content: 'Carregando...',
        position: {
            corner: {
                tooltip: 'leftBottom',
                target: 'rightMiddle'
            }
        },
        style: {
            border: {
                width: 4,
                radius: 7
            },
            padding: 5,
            width: 335,
            textAlign: 'center',
            tip: true
        },
        api: {
            // Retrieve the content when tooltip is first rendered
            onRender: function() {
                var self = this;
                $.get('<?php echo URI_CONTATOS; ?>info.php',
                    {
                        id: self.elements.target.attr('id'),
                        status: self.elements.target.attr('name')
                    }, function(data) {
                        // Update the tooltip with the results

```

```

                self.show();
                self.updateContent(data);
            }
        });
    },
    show: { solo: true },
    hide: { when: 'mouseout', fixed: true, delay: 500 }
});
});
</script>

```

```
web_reset.pl
```

```

#!/usr/bin/perl
# VoIP@UFSC --- Servico de VoiceMAIL
#
# Guilherme Rhoden/Jonatan H. M.
# 2010/11/12
# v1.0

use strict;
use DBI();
my $host = "adm.voip.ufsc.br";
my $database = "voip";
my $user = "voip-media";
my $pw = "voip.adm,media";
my $x=0;
my $query;
my $ramal;
my $codigo;

$ramal = $ARGV[0];
$codigo = $ARGV[1];

# Connect to the database.
my $dbh = DBI->connect("DBI:mysql:database=$database;host=$host",
                      "$user", "$pw",
                      {'RaiseError' => 1});

# Status
# 0. Inicio
# 1. Informou o ramal
# 2. Atendeu a chamada
# 3. Informou o código na central de recuperação
# 4. Verificou o email
# 5. Informou o código recebido no email
# 6. Cadastrou a nova senha de acesso ao portal

$query = "select email, codigo_email from usuarios_recuperacao_senha where codigo_ura = '$codigo' and ramal
= '$ramal' and data_geracao < date_add(now(), interval 30 minute)";

my $sth = $dbh->prepare( $query );

$sth->execute();

while (my $ref = $sth->fetchrow_hashref() {
#   print "[$x] - Ramal: $ref->{'ramal'}, [$ref->{'total_novas'}], [$ref->{'total_antigas'}]\n";

```

```

envia_email( $ramal, $ref->{'email'} , $ref->{'codigo_email'} );
$x = 1;
}

if ( $x ) {

    $query = "update usuarios_recuperacao_senha set status = 3 where ramal = '$ramal'";
    my $sth2 = $dbh->prepare( $query );
    $sth2->execute();
    print "1";
    exit 0;
} else {
    print "Codigo nao confere, esta expirado ou nao foi gerado\n";
    exit 1;
}

$sth->finish();
# Disconnect from the database.
$dbh->disconnect();

sub envia_email {
    my $ramal = shift;
    my $to = shift;
    my $codigoMail= shift;

    my $from= 'voip@npd.ufsc.br';
    my $subject="[VOIP-UFSC] [$ramal] Redefinição de senha - Portal do Usuário";

    open(MAIL, "|/usr/sbin/sendmail -t");

    ## Mail Header
    print MAIL "To: $to\n";
    print MAIL "From: $from\n";
    print MAIL "Subject: $subject\n";
    print MAIL "Content-Type: text/html; charset=UTF-8\n\n";
    ## Mail Body

    #HTML:
    print MAIL "Prezado(a),<br /><br />" .
        "Foi solicitada a redefinição de sua senha para acesso ao Portal do Usuário do VoIP
Coporativo da UFSC.<br />" .
        "Caso não tenha solicitado a redefinição ou não for o responsável pelo ramal $ramal, por
favor entre em contato com a equipe VoIP\@UFSC, respondendo este e-mail.<br /><br />" .
        "Para completar o processo, informe o código abaixo na interface do Portal do Usuário.<br
/>" .
        "Código de verificação: <strong>$codigoMail</strong><br /><br />" .
        "Observação:<br />" .
        "Este código irá expirar após 30 minutos a partir do horário da solicitação.<br /><br />" .
        "Atenciosamente,<br />" .
        "Equipe VoIP\@UFSC<br />" .
        "Ramal 6343<br />" .
        "<img
src=\"http://adm.voip.ufsc.br/usuario/public/redefinicao/?action=get_image&ramal=$ramal\" /><br />";

    close(MAIL);
}

```

```

exten-srv_add.conf
; 49 servico de reset senha portal usuario

exten => _49XXXX,1,Answer()
exten => _49XXXX,n,Set(ORIGEM1=${CALLERID(num)})
exten => _49XXXX,n,Set(ORIGEM2=${EXTEN:2})
exten => _49XXXX,n,Set(GLOBAL(TENTATIVA)=0)
exten => _49XXXX,n,GotoIf(${ ${ORIGEM1} = ${ORIGEM2} ]?ura-reset-pw-web,teste,1:ura-reset-pw-web,2,1 )
exten => _49XXXX,n,Hangup

[ura-reset-pw-web]
exten => teste,1,Set(TIMEOUT(digit)=2)
exten => teste,n,Set(TIMEOUT(response)=45)
exten => teste,n,Background(ivr/portal_usuario/recuperacao_inicio)
exten => teste,n,WaitExten(20)

exten => 1,1,Goto(ura-reset-pw-web-op1,s,1)
exten => 2,1,Goto(ura-reset-pw-web-op2,s,1)

[ura-reset-pw-web-op1]
exten => s,1,NoOp(Opcao 1 - pedir senha)
exten => s,n,Set(TIMEOUT(digit)=2)
exten => s,n,Set(TIMEOUT(response)=45)
exten => s,n,Background(ivr/portal_usuario/recuperacao_informar)
exten => s,n,WaitExten(10)

exten => _XXXXX,1,Set(RETORNO=${SHELL(/usr/local/voip-scripts/web_reset_pw.pl ${ORIGEM1} ${EXTEN})})
exten => _XXXXX,n,NoOp( Retorno [ ${RETORNO} ] - [ ${EXTEN} ])
exten => _XXXXX,n,GotoIf(${ [ ${RETORNO} ] = "1" ]?ura-reset-pw-web-sucesso,s,1:ura-reset-pw-web-nao-sucesso,s,1)
exten => _XXXXX,n,Hangup()

[ura-reset-pw-web-op2]
exten => s,1,NoOp(Opcao 2 - Sair)
exten => s,n,Playback(ivr/portal_usuario/recuperacao_ate_logo)
exten => s,n,Hangup()

[ura-reset-pw-web-sucesso]
exten => s,1,NoOp(Sucesso codigo confere)
exten => s,n,Playback(ivr/portal_usuario/recuperacao_fim)
exten => s,n,Hangup()

[ura-reset-pw-web-nao-sucesso]
exten => s,1,NoOp(Erro -Codigo Invalido - ${TENTATIVA})
exten => s,2,Set(GLOBAL(TENTATIVA)=${TENTATIVA}+1)
exten => s,3,Playback(ivr/portal_usuario/recuperacao_codigo_invalido)
exten => s,4,GotoIf(${ [ ${TENTATIVA} < 3 ]?ura-reset-pw-web-op1,s,1:ura-reset-pw-web-nao-sucesso,s,5 )
exten => s,5,Playback(ivr/portal_usuario/recuperacao_ate_logo)
exten => s,n,Hangup()

```

Convergência de Mídias em um Ambiente VoIP

Jonatan Hartmann Matschulat

Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900, Florianópolis – SC – Brasil

jonatan@inf.ufsc.br

***Abstract.** IP Telephony is fastly growing in business and residences. The major factor that contributes to this expansion is the advance of the Internet, which provides the channel for VoIP data transportation. The implantation of a IP Telephony system should aim to the its benefits, such as the development of advanced applications and services, based on features provided by the Internet. That was the most important motivation for this project where the main tools and protocols involved in a IP Telephony were studied. The knowledge acquired by the analysis of the VoIP system at UFSC could be used to design and develop an application focused on data convergence in a IP Telephony environment.*

***Resumo.** A telefonia IP está sendo usada cada vez mais no cotidiano das empresas e residências. Um dos principais fatores que contribui para isto é a expansão da rede mundial de computadores, a Internet, que proporciona o canal para transporte dos dados VoIP. É interessante que na implantação de um ambiente de telefonia IP, aproveitemos as vantagens que ele tem a oferecer, como por exemplo, a implementação de novas aplicações e serviços baseados nos recursos que a Internet proporciona. Esta foi a principal motivação para o desenvolvimento deste trabalho. Foram estudadas as principais ferramentas e protocolos envolvidos em uma rede de telefonia IP, seguindo pelo levantamento de necessidades e possíveis soluções para o ambiente VoIP da UFSC e por fim, o desenvolvimento de uma aplicação voltada à convergência de dados em um ambiente de telefonia IP.*

1. Introdução

O aproveitamento da rede de computadores (WAN – *Wide Area Network*) para o transporte dos pacotes de voz da telefonia VoIP favorece sua implantação uma vez que praticamente não há necessidade de utilização de canais de transporte exclusivos para este fim. Além disto, a redução de custos é facilmente percebida em chamadas de longa distância, principalmente entre instituições interconectadas via IP, onde o custo é de uma chamada local quando a instituição de destino completa ligações para a rede pública de telefonia, ou nulo, caso a chamada seja destinada à um ramal dentro da instituição. Todos estes fatores contribuiram e contribuem para a substituição da telefonia convencional pela telefonia IP. Podemos observar esta mudança na própria Universidade Federal de Santa Catarina, a qual adquiriu entre os anos de 2009 e 2010, cerca de 650 telefones IP e já se planeja adquirir mais 1000 aparelhos no próximo ano, conforme a demanda.

2. VoIP

VoIP - *Voice over IP* (Voz sobre IP) é o termo dado a tecnologia de transmissão de dados de voz através de uma rede de comunicação baseada em IP.

De acordo com MEGGELEN, MADSEN e SMITH (2007), VoIP não é apenas um simples método de obter chamadas de longa distância gratuitas. Sua principal vantagem é a possibilidade da transmissão de voz não ser nada mais do que outra aplicação na rede de dados. MEGGELEN, MADSEN e SMITH (2007) tratam isto como uma grande revolução, inevitável e que traz consigo importantes ferramentas *open-source*, como o PBX Asterisk.

2.1. Telefonia IP

Trata-se simplesmente do VoIP agregado de recursos, onde há a presença de *hardphones* ou *softphones*. Assim, os telefones terão IP, recursos aprimorados e poderão apresentar interatividade com aplicações dos servidores da rede. Os demais produtos que farão parte da rede de telefonia IP também se comunicarão via IP, por exemplo: um FAX, uma vez conectado à um ATA (*Analog Telephone Adapter*) (AMARAL, Italo. Cisco Certified, 2008).

2.2. Protocolos

Os protocolos descrevem as mensagens e regras para a comunicação dos dados. No VoIP, os protocolos são responsáveis pela sinalização e transporte dos dados de voz entre as partes comunicantes.

2.2.1 SIP

JOHNSTON (2009) define o SIP – *Session Initiation Protocol* como um protocolo de sinalização, presença e mensagem instantânea, desenvolvido para estabelecer, modificar e encerrar sessões multimídia; requisitar e entregar presença; e enviar e receber mensagens instantâneas.

ROSENBERG (2002) explica que o SIP é baseado em um modelo de transação como o HTTP, no qual há uma requisição e uma resposta. Cada transação consiste em uma requisição feita por um cliente, que é recebida e processada pelo servidor, e que por fim, retorna a resposta ao cliente.

As normas RFC 3261, 3265, 3515 e 3428 definem os seguintes métodos de requisição do protocolo SIP:

- REGISTER: Utilizado para o ramal se registrar na central, indicando seu endereço IP atual;
- INVITE: Utilizado para estabelecer uma sessão entre ramais;
- ACK: Confirma o recebimento de uma mensagem;
- CANCEL: Cancela uma requisição pendente, como por exemplo um convite para estabelecer uma chamada;
- BYE: Encerra uma sessão estabelecida (chamada em andamento);
- OPTIONS: Utilizado para obter informações de um ramal, tais como as capacidades do telefone IP;

- SUBSCRIBE: Inscreve-se em um evento ou notificação;
- REFER: Utilizado para transferência de chamadas;
- MESSAGE: Envio de mensagem instantânea.

2.2.2 H.323

O H.323 é o antecessor do SIP, que vem aos poucos sendo substituído. O H.323 é um dos protocolos mais maduros, robustos e utilizados nos equipamentos que utilizam VoIP por ser um dos mais antigos e definido especificamente com a finalidade de comunicações multimídia.

2.2.3 IAX

Segundo MONTORO e CASILARI (2009), o IAX - *Inter Asterisk eXchange*, é um novo protocolo desenvolvido para o servidor Asterisk, que reúne as funcionalidades de sinalização de chamadas (H.323/SIP) e o transporte de dados multimídia (RTP).

2.2.4 RTP

De acordo com GROSS, 2009 *apud* SCHULZRINNE, H. et al, O RTP provê funções de transporte de rede fim-a-fim ajustadas para aplicações transmitindo dados em tempo-real, como áudio, vídeo ou dados simulados, sobre serviços de rede multicast ou unicast. O RTP não contempla reserva de recursos e não garante qualidade de serviço para serviços em tempo-real.

3. Telefonia IP - UFSC

O sistema de telefonia VoIP na UFSC iniciou através do projeto *fone@RNP*, que é o nome do serviço que permite o encaminhamento de chamadas, isto é, a interconexão das instituições clientes da rede VoIP (Voz sobre IP) da RNP (REDE NACIONAL DE ENSINO E PESQUISA, 2009).

As vantagens observadas com a implantação e utilização do VoIP@UFSC (nome dado ao *fone@RNP* na UFSC) somadas a necessidade de evoluir a rede de telefonia da universidade, levaram a criação do VoIP Corporativo, um ambiente de telefonia IP voltado aos reitores, diretores, coordenadores, professores, secretários e demais servidores da UFSC, especialmente aos que já possuíam um ramal da telefonia convencional.

O sistema de telefonia IP corporativa, chamado de PBX-IP, foi desenvolvido e mantido pela Universidade Federal de Santa Catarina em parceria com a RNP e PoP-SC. É uma solução completa para VoIP adotada também pelo Ministério da Saúde.

4. Convergência de Mídias em um Ambiente VoIP

Na PBX-IP, uma das principais necessidades identificadas foi oferecer serviços que possibilitem a integração e convergência de mídias. Atualmente os usuários dispõem apenas de recursos básicos da telefonia, como o recebimento, efetuação e transferência de chamadas. Existe também o serviço de correio de voz, porém com acesso único através do telefone, onde a navegação é difícil.

Com base nisto, foram levantadas sugestões de aplicações avançadas, que aumentem a flexibilidade, interatividade e integração nos serviços de comunicação.

Tendo em vista os conceitos de convergência em VoIP e Comunicações Unificadas, que segundo MCCHARLES (2010), é uma arquitetura de tecnologia de comunicação que automatiza e unifica várias formas e dispositivos de comunicação humana em um contexto, a solução definida neste projeto foi o desenvolvimento de uma plataforma web que agregaria diversos serviços para o usuário final. Desta forma os recursos oferecidos pela telefonia poderiam ser acessados não apenas a partir de telefone IP ou *softphone*, mas também através de um portal na internet.

5. Portal do Usuário

A plataforma web, denominada Portal do Usuário, foi implementada principalmente na linguagem PHP. A interface web, acessada pelo cliente, foi desenvolvida utilizando HTML para a estrutura e apresentação das páginas, Javascript para requisições AJAX, manipulação do DOM, efeitos visuais e CSS para o estilo das páginas. A biblioteca jQuery foi adotada para a utilização do Javascript e também como base para outros *plugins*.

A arquitetura de funcionamento do Portal do Usuário (figura 1) consiste basicamente em quatro elementos principais:

- Usuário, utilizando um navegador web, faz requisições HTTP ao servidor;
- Servidor Web, que atende as requisições do usuário, realiza as consultas nos bancos de dados (LDAP e MySQL) quando necessário, processa e retorna os dados ao navegador web;
- Diretório LDAP, onde estão armazenados os dados dos ramais VoIP e convencionais, unidades e departamentos da instituição, dentre outras estruturas que fazem parte da PBX-IP;
- Banco de dados MySQL, responsável pelo armazenamento dos dados dos usuários do portal, registros de contabilização de chamadas efetuadas e perdidas, etc.

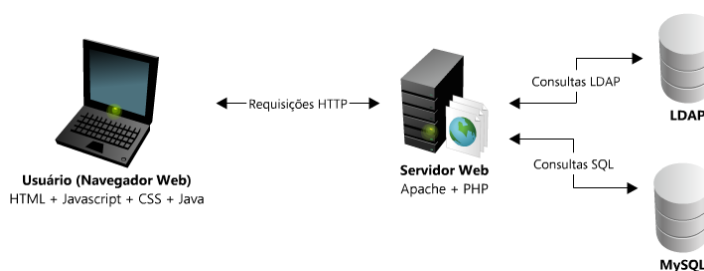


Figura 1 - Arquitetura do portal do usuário

5.1. Lista de Contatos

A agenda (figura 2), disponível no Portal do Usuário, oferece uma interface intuitiva para o cadastramento, edição, exclusão e localização de contatos, na qual o usuário pode buscar dentre os ramais já cadastrados na PBX-IP e adicioná-los em sua lista pessoal. Além disto, a lista de contatos é integrada ao *webphone*, onde o usuário tem a possibilidade de efetuar uma chamada para um contato específico a partir de seu registro na lista.

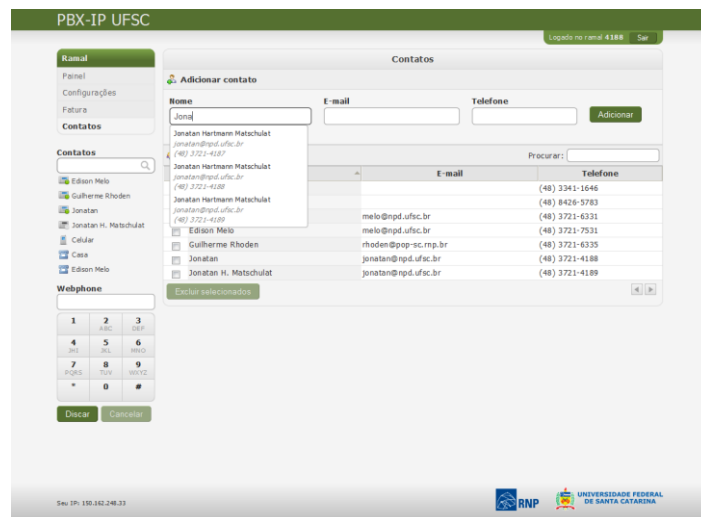


Figura 2 - Portal do usuário: Lista de contatos

5.2. Webphone

Muitas vezes o usuário se encontra longe do seu telefone IP, o que acaba impossibilitando-o de realizar e receber chamadas. Em outros casos, mesmo estando próximo do aparelho, a localização de um contato torna-se difícil devido às limitações de interface. Pensando nisso, surgiu a oportunidade de inclusão de um telefone virtual que possa ser executado no próprio navegador do usuário, sem que haja necessidade de instalação de um *softphone*.

Após uma pesquisa de soluções existentes no mercado e do estudo das mensagens de sinalização utilizando o protocolo SIP, foram levantadas as três seguintes alternativas:

- 1) Java Applet, que utiliza uma solução personalizada da empresa Mizutech e suporta a efetuação e recebimento de chamadas;
- 2) Click2Call, onde o servidor web gera a chamada utilizando mensagens SIP através da biblioteca php-sip;
- 3) Integração com *Softphone*, na qual aciona o *softphone* instalado no computador do usuário.

5.3. Mensagens

O serviço de mensagens tem por objetivo oferecer uma alternativa para envio de pequenas mensagens, funcionando como o SMS – *Short Message Service*. O serviço pode ser utilizado para divulgação de avisos, alertas ou recados para grupos de ramais ou um ramal específico. As mensagens enviadas devem ser apresentadas na tela dos aparelhos telefônicos IP.

Para o desenvolvimento desta aplicação, foi necessário estudar os telefones IPs, mais especificamente os modelos do fabricante Polycom, disponíveis aos usuários dos ramais na universidade. Tais telefones possuem o recurso HTTP Push, o qual permite o envio de uma mensagem ou URL a ser apresentada no aparelho.

O formato de uma requisição via HTTP Push segue o exemplo da figura 3:

```

<PolycomIPPhone>
  <Data priority="critical">
    <html>
      <title>Mensagem de: 4188</title>
      <body>Teste</body>
    </html>
  </Data>
</PolycomIPPhone>

```

Figura 3 - Exemplo de mensagem HTTP Push

5.4. Correio de voz

A fim de melhorar o serviço da secretária eletrônica (correio de voz), foi desenvolvida uma interface web (figura 4) na qual o usuário pode ouvir e apagar os recados de forma rápida e simples. O *plugin* para jQuery utilizado para reprodução das mensagens foi o jPlayer, o qual aproveita os novos recursos de áudio providos pelo e para os navegadores que ainda não o suportam, uma alternativa em Flash. As mensagens do correio de voz são gravadas no banco de dados MySQL em campos do tipo BLOB - *Binary Large Object*, no formato de áudio wav. Porém, como o *plugin* jPlayer não suporta arquivos de áudio em tal formato, foi necessário desenvolver um *script* para conversão sob-demanda das mensagens para que pudessem ser reproduzidas. Este script faz uso do LAME para codificar o áudio mp3 e o ffmpeg para fazer a conversão do wav para mp3.

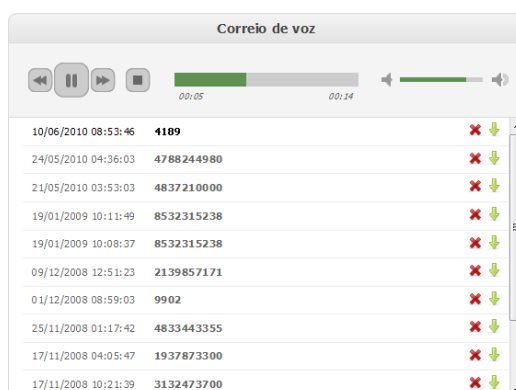


Figura 5 - Portal do usuário: Correio de voz

Além de reproduzir os recados do correio de voz, o sistema é integrado à central VoIP, que notifica o telefone IP do usuário quando um recado é lido. Para isto, após a reprodução do áudio ou exclusão de uma mensagem, uma consulta é feita no banco de dados MySQL para obter a quantidade de recados lidos e não lidos na caixa de mensagens do usuário. Estes dados são enviados ao ramal através de uma mensagem SIP NOTIFY que é tratada e apresentada ao usuário pelo telefone IP.

5.5. Fatura

O módulo de fatura permite ao usuário gerar relatórios da utilização de seu ramal. A interface permite a seleção do período no qual as chamadas foram efetuadas e recebidas para que com base nestes valores, o sistema faça a geração da fatura no formato PDF.

Os registros de eventos SIP são armazenados no banco de dados através do OpenSIPs (OpenSER). Para contabilizar as chamadas completadas e perdidas foi necessário criar um *trigger* (gatilho) no MySQL, que consiste em consultar os parâmetros das mensagens de INVITE e BYE de uma determinada chamada, armazenando o resultado em uma nova tabela. A figura 6 representa a sintaxe utilizada para a criação do *trigger* responsável pela contabilização.

```

1 DELIMITER $$
2
3 USE `voip`$$
4
5 DROP TRIGGER /*!50032 IF EXISTS */ `voipacc`$$
6
7 CREATE
8
9     /*!50017 DEFINER = 'root'@'localhost' */
10    TRIGGER `voipacc` AFTER INSERT ON `acc`
11    FOR EACH ROW BEGIN
12    IF NEW.method = 'BYE' THEN
13        INSERT INTO completed_calls (
14            SELECT DISTINCT
15                a.time AS _datetime,
16                a.from_number AS _from,
17                a.to_number AS _to,
18                TIMEDIFF(b.time, a.time) AS _duration
19            FROM acc a,
20                 acc b
21             WHERE a.method = 'INVITE'
22                   AND b.method = 'BYE'
23                   AND a.callid = b.callid
24                   AND a.callid = NEW.callid);
25    END IF;
26 END;
27 $$
28 DELIMITER ;

```

Figura 6 - Trigger de contabilização da PBX-IP

5.6. Configurações

A tela de configuração desenvolvida (figura 7), além das opções de ajustes de serviços e dados pessoais, oferece a possibilidade de configuração de opções do próprio portal do usuário, como por exemplo o tipo do *webphone* desejado: Java, Click2Call ou Softphone.

Figura 7 - Portal do usuário: Configurações

5.7. Definição de senha

O sistema de definição de senha foi projetado para permitir a ativação das contas dos ramais que ainda não possuem uma senha para acesso ao Portal do Usuário, ou para aqueles que esqueceram sua senha. Para completar o processo de redefinição, o usuário deverá informar duas senhas, uma na central URA, programada no Asterisk e outra que será enviada ao seu *email*.

6. Conclusão

Neste projeto foi possível estudar os principais componentes envolvidos em um ambiente VoIP, desde os protocolos utilizados até as aplicações de alto nível, oferecidas aos usuário final.

Com base no estudo feito, diversas necessidades foram identificadas, as quais foram mapeadas em possíveis soluções para prover maior convergência e integração entre mídias e aplicações.

O portal do usuário foi desenvolvido com o intuito de agregar diferentes serviços, tal como o telefone virtual (*webphone*), correio de voz, fatura, entre outros, em um único local, acessível através de um navegador web. Tais aplicações proporcionam maior facilidade ao usuário, o que conseqüentemente leva ao aumento de produtividade e atinge os principais objetivos relacionados ao conceito de Comunicação Unificadas, que está intimamente ligado à convergência de mídias.

5. Trabalhos futuros

Tendo uma plataforma para desenvolvimento estabelecida juntamente com algumas aplicações, a expansão torna-se mais fácil. Com base nas possíveis soluções e demandas levantadas anteriormente, diversas possibilidades poderão ser exploradas futuramente, a fim de oferecer novas facilidades ao usuário da telefonia IP da UFSC e provendo maior convergência de mídias.

Referências

- AMARAL, Italo. *VoIP vs Telefonia IP*. Disponível em <<http://blog.ccna.com.br/2008/05/14/voip-vs-telefonia-ip/>> Acesso em 9 de julho de 2010.
- GROSS, Fábio Danieleski. *Implementando uma Solução VoIP Baseada em Asterisk com Alta-disponibilidade e Balanceamento de Carga*. Florianópolis, SC: Universidade Federal de Santa Catarina, 2009.
- JOHNSTON, Alan B. *SIP, Understanding the Session Initiation Protocol, Third edition*. Norwood, MA: Artech House 2009.
- MCCHARLES, Rick. *Definition of Unified Communications, Its Features and Benefits*. Disponível em <<http://www.ric.ca/blog/2008/04/definition-of-unified-communic.html>> Acesso em 3 de outubro de 2010.
- MEGGELEN, Jim Van; MADSEN, Leif; SMITH, Jared. *Asterisk, The future of telephony, Second Edition*. Sebastopol, CA: O'Reilly Media, Inc, 2007.
- MONTORO Pablo; CASILARI Eduardo. *A Comparative Study of VoIP Standards with Asterisk*. Spain: University of Malaga, 2009
- REDE NACIONAL DE ENSINO E PESQUISA. *Serviço fone@RNP – Descrição geral*. Disponível em <http://www.rnp.br/_arquivo/voip/fone_rnp.pdf> Acesso em 8 de julho de 2010.
- ROSENBERG, et. al. *Standards Track - RFC 3261, SIP: Session Initiation Protocol, 2002*. Disponível em <<http://www.ietf.org/rfc/rfc3261.txt>> Acesso em 22 de novembro de 2010.