

Uni-FACEF CENTRO UNIVERSITÁRIO MUNICIPAL DE FRANCA

LUIZ ROGÉRIO DOS SANTOS

Implementação de PABX sobre voz IP
com gravações de ligações e registros

FRANCA

2017

LUIZ ROGÉRIO DOS SANTOS

Implementação de PABX sobre voz IP
com gravações de ligações e registros

Trabalho de Conclusão de Curso apresentado no
Departamento de Sistemas de Informação como
requisito básico para conclusão do curso de
Bacharelado em Sistemas de Informação

Orientador: Silvio Carvalho Neto

FRANCA

2017

LUIZ ROGÉRIO DOS SANTOS

Implementação de PABX sobre voz IP
com gravações de ligações e registros

Trabalho de Conclusão de Curso apresentado no
Departamento de Sistemas de Informação como
requisito básico para conclusão do curso de
Bacharelado em Sistemas de Informação

Franca-SP, de de 2017.

BANCA EXAMINADORA:

Orientador: _____

Prof. Dr. Silvio Carvalho Neto

Centro Universitário Municipal de Franca – Uni-FACEF

Examinador 1: _____

Prof. Daniel Facciolo Pires

Centro Universitário Municipal de Franca – Uni-FACEF

Examinador 2: _____

Prof. Letícia Faleiros

Centro Universitário Municipal de Franca – Uni-FACEF

AGRADECIMENTOS

Primeiramente agradeço a Deus, a minha família que sempre me apoia e a memória de minha mãe, que sempre esteve ao meu lado. Meus agradecimentos também ao meu orientador Silvio Carvalho Neto, pelo apoio, paciência e compreensão.

RESUMO

Este trabalho apresenta um estudo sobre as tecnologias de voz sobre IP (Voip), juntamente com o sistema Asterisk, um software de código aberto, que roda em sistema operacional Linux, que surgiu graças ao trabalho de Mark Spencer e Jim Dixon, que tem a capacidade de poder oferecer mais recursos que os oferecidos por um PABX tradicional. Justamente por causa do seu custo, o Asterisk tornou o uso do PABXIP muito mais acessível para as empresas. O estudo mostra a maior parte de arquitetura do software, oferecendo uma ampla visão da maioria de seus recursos disponíveis. O trabalho também apresenta um modelo de negócios baseado na solução que foi desenvolvida, tendo como base um produto que pode acrescentar soluções para as empresas usando código aberto e diminuindo custos com licença e ligações. Uma implementação da solução foi realizada e descrita no decorrer da monografia, sendo possível entender como se faz as configurações de um servidor PABXIP baseado em Asterisk, em um servidor Linux, pronto para realizar ligações entre ramais e também armazenando todos os registros em um banco de dados. Foi criada uma interface web utilizando o Framework Laravel e linguagem de programação PHP, para que as informações das ligações do servidor fossem consultadas. O estudo mostra as vantagens e desvantagens em adquirir um servidor Asterisk e suas diferenças com um PABX tradicional, mostra também como as empresas podem se beneficiar dessa tecnologia e ao mesmo tempo economizar dinheiro.

ASBTRACT

This work presents a study on the technologies of voice on IP (Voip), together with the system Asterisk, a software of open code, which wheel in operating system Linux, which appeared thanks to the work of Mark Spencer and Jim Dixon, who has the capacity to be able to offer more resources than the offered ones for a traditional PABX. Just because of his cost, the Asterisk made the use of the much more accessible PABXIP for the enterprises. The study shows most software architecture, providing a broad view of most of its available features. The work also presents a business model based on the solution that was developed, based on a product that can add solutions for companies using open code and reducing license costs and connections. An implementation of the solution was carried out and described in the monograph, being possible to understand how to do a PABXIP server settings based on Asterisk, a Linux server, ready to make calls between extensions and also storing all records in a database. A web interface was created using the Laravel Framework and the PHP programming language, so that the information of the connections from the server to be queried. The study shows the advantages and disadvantages in acquiring an Asterisk server and their differences with a traditional PBX, it also shows how companies can benefit from this technology and at the same time save money.

SUMÁRIO

1. INTRODUÇÃO	9
2. TECNOLOGIAS PARA IMPLANTAÇÃO DE SISTEMAS DE PABX VIA IP.....	12
2.1. PABX TRADICIONAL.....	12
2.2. CONCEITO DE IP	13
2.3. CONCEITO DE VOIP	14
2.4. PABX IP	15
2.5. ASTERISK.....	16
2.5.1. História da Telefonia Zapata e como ela se relaciona com PBX Asterisk	16
2.6. OBSTÁCULOS COM A IMPLANTAÇÃO DE PABX VIA IP	18
2.7. BENEFÍCIOS COM A IMPLANTAÇÃO DE PABX VIA IP.....	18
3. ARQUITETURA DO ASTERISK	20
3.1. CODECS	20
3.2. PROTOCOLOS	20
3.3. MÓDULOS DO ASTERISK	21
3.3.1. Aplicações.....	21
3.3.2. Módulos Ponte	22
3.3.3. Módulos de gravação de detalhes de chamadas	23
3.3.4. Módulos de Registros de Evento de Canal	23
3.3.5. Controladores de canal	24
3.3.6. Tradutores de codec	25
3.3.7. Funções do Dialplan.....	26
3.3.8. Módulos PBX	27
3.3.9. Módulos de recursos	28
4. MODELO DE NEGÓCIO	30
4.1. MODELO DE NEGÓCIO DE UMA EMPRESA DE TELEFONIA ASTERISK ..	33
4.1.1. Segmento de Clientes	34
4.1.2. Proposta de valor	34

4.1.3.	Canais	34
4.1.4.	Relacionamento com clientes	35
4.1.5.	Fontes de receita	36
4.1.6.	Recursos principais	36
4.1.7.	Atividades chave	36
4.1.8.	Parcerias principais	37
4.1.9.	Estrutura de custos	37
5.	IMPLEMENTAÇÃO DE ASTERISK PARA PABXIP	39
5.1.	INSTALAÇÃO DO SERVIDOR.....	39
5.2.	INSTALAÇÃO DO ASTERISK.....	40
5.3.	INSTALAÇÃO DO MYSQL.....	46
5.4.	CONFIGURANDO O ASTERISK PARA CONEXÃO COM MYSQL	49
5.5.	CONFIGURAÇÃO DE RAMAIS.....	54
5.6.	INSTALAÇÃO E CONFIGURAÇÃO DA INTERFACE GRÁFICA	60
6.	CONSIDERAÇÕES FINAIS	67
	REFERÊNCIAS.....	69
	APÊNDICES	71

1. INTRODUÇÃO

Com a evolução da internet, surgiram também novos meios de comunicação e a tecnologia VOIP é uma delas. A voz sobre IP causou uma grande evolução nos sistemas de telecomunicações, pois com ela, foi possível a realização de ligações internacionais sem custo algum, somente usando a conexão de banda larga. Estas ligações também podem ser efetuadas de um computador com um software gratuito configurado a um PABXIP e um fone com microfone conectados.

Porém, quando essa tecnologia surgiu, somente as grandes corporações eram capazes de desfrutar de sua utilização, pois o custo com equipamentos e licenças de softwares era muito grande, dificultando muito essa migração de recursos. Ademais, o custo com manutenções em equipamentos analógicos era muito grande, juntamente com a pouca capacidade de expansão e o fato de que poucos recursos poderiam ser adicionados aos sistemas. Um ponto negativo também na telefonia analógica é que a sua infraestrutura tem que ser separada da rede de dados, com a tecnologia VOIP isso não é mais necessário, uma vez que a voz corre sobre a mesma infraestrutura de dados.

Essa pesquisa tem como objetivo apresentar uma solução baseada em software livre ou seja, que utiliza uma licença GPL (*General Public Licence*), que é capaz de substituir um sistema PABX convencional, com baixo custo de investimento e ao mesmo tempo obter maiores recursos que podem agregar grandes valores aos negócios. Esse trabalho teve o intuito também de mostrar como é instalado e configurado um PABXIP, para que sejam consultados os registros de ligações, bem como o áudio gravado de cada chamada com suas respectivas durações. Durante a pesquisa foi apresentada a maior parte da estrutura do Asterisk, o qual é um software PABXIP, mostrando assim a sua grande capacidade de personalização e adaptação para cada tipo de negócio.

O Asterisk é um software de código aberto que tem a capacidade de atender desde uma pequena até uma grande empresa. Com esse intuito, surgiram os estudos e pesquisas sobre o mesmo, por ser uma grande solução e que, ainda por muitos, é desconhecida, além de tudo gratuita. Essa pesquisa poderá contribuir com a ampliação do conhecimento para todos que tenham interesse em diminuir os custos com telefonia e obter mais recursos nos quais podem agregar um grande valor as

corporações, ou mesmo, apenas com o intuito de obter maiores informações sobre a estrutura do sistema PABXIP.

O método de pesquisa é exploratório, com pesquisa bibliográfica e de campo a partir da implantação de um sistema de PABXIP com o uso do Asterisk. A pesquisa bibliográfica inicial se deu por análise em livros de redes, telefonia, voip, Asterisk, trabalhos acadêmicos e sites relacionados ao tema. Posteriormente, foi realizada uma implementação de um exemplo de PABXIP, que possui uma interface gráfica, sendo capaz de consultar todos os registros necessários de uma ligação realizada ou recebida, que poderia ser utilizada em um ambiente de produção real.

Desta forma, o trabalho foi dividido em algumas partes. No capítulo dois foram apresentados os conceitos de PABX tradicional, logo na sequência os conceitos de IP e como é o seu funcionamento, o que é VOIP, como surgiu e como ele se comporta na rede de dados, os conceitos de PABXIP e suas funcionalidades, o que é o Asterisk em uma breve apresentação, a história de como surgiu o Asterisk, por quem foi criado e evoluído. Finalizando o capítulo dois, são apresentados itens mostrando quais as dificuldades para implementar e quais as vantagens em obter sistemas PABXIP nas organizações.

O capítulo três apresenta a arquitetura do Asterisk. Neste capítulo é possível ter uma visão da maioria dos seus arquivos de configuração e obter uma noção do que o sistema é capaz de fazer.

O capítulo quatro tem como objetivo apresentar um modelo de negócio, a fim de tornar o sistema apresentado em um produto que pode ser rentável e iniciar uma startup com base nessa proposta.

O quinto capítulo mostra como foi realizada a pesquisa e a implementação do PABXIP Asterisk, tendo uma base de tudo o que foi realizado nessas etapas. Foi realizada uma instalação de um servidor PABXIP, no qual foi configurado para ser capaz de realizar ligações entre ramais, gravar os áudios das mesmas e também gravar todos os registros em um banco de dados, sendo possível assim criar uma interface web para que essas informações possam ser acessadas.

No sexto capítulo são apresentadas as considerações finais, mostrando uma conclusão sobre possíveis melhorias nas configurações do software e também os benefícios em que o sistema pode trazer para as empresas.

Por fim, no apêndice, é apresentada a documentação de engenharia de software, com os diagramas, casos de uso, regras de negócio, requisitos, dentre outros documentos.

2. TECNOLOGIAS PARA IMPLANTAÇÃO DE SISTEMAS DE PABX VIA IP

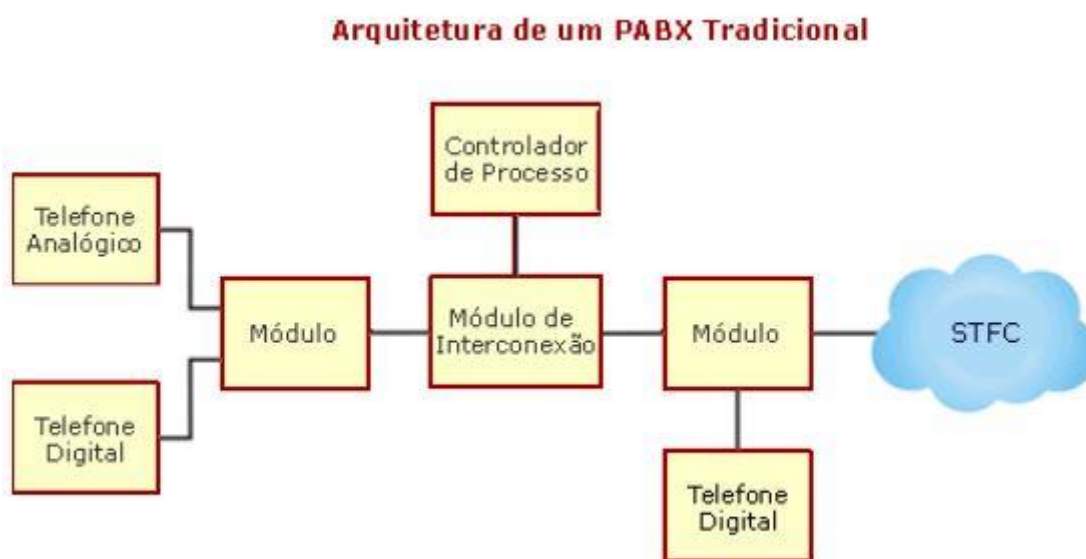
2.1. PABX TRADICIONAL

Segundo Sato (2004), o telefone é um dos equipamentos mais antigos de um escritório. A tecnologia que está por trás dele vem sofrendo grandes transformações nos últimos anos. Os sistemas de PABX tradicionais estão desatualizados perante o mercado global. As empresas estão tomando atitudes de mudanças para melhorar isso, elas estão mudando o tráfego de voz para a rede de dados afim de diminuir custos e obterem maiores funcionalidades para aumentar a performance dos seus funcionários.

Por volta dos anos 80, foram desenvolvidos os PABXs tradicionais (analógicos), os microprocessadores eram muito caros e muito limitados. A rede de dados ainda era desconhecida (SATO, 2004). O PABX tradicional geralmente utiliza tecnologias proprietárias de cada fabricante obrigando a empresa que o adquiriu, sempre quando tiver que realizar manutenção ou mesmo adicionar uma função, depender de técnicos especializados da marca, fazendo com que isso eleve o custo das operações (SATO, 2004). Segundo Sato (2004), o PABX tradicional funciona da seguinte forma:

Ele possui um controlador de processos, executando um software que no qual é responsável por todas as funções do sistema. Existem os dispositivos de ponta (*endpoints*) que são ligados ao sistema e usam suas funcionalidades. Existem os telefones digitais e analógicos que para funcionar são ligados aos cartões de interfaces de módulos do PABX. Os Módulos são placas que são responsáveis por interligar os dispositivos de ponta, gateways ou outros dispositivos que o PABX suporta. Existem diferentes tipos de dispositivos que podem ser ligados ao PABX afim de fazerem a comunicação com a STFC (Sistema de Telefonia Fixa Comutada). Outros módulos também podem ser usados como os de interconexão que fazem a conversão de portas diferentes para que outros dispositivos possam funcionar (SATO, 2004). Na figura 1 explica como funciona o PABX tradicional.

Figura 1: Funcionamento de um PABX Tradicional



Fonte: Sato (2004).

2.2. CONCEITO DE IP

O Protocolo TCP/IP (Protocolo de Controle de Transmissão / Protocolo de Internet) surgiu pelo desenvolvimento da DARPA (*Defense Advanced Research Projects Agency*) dos Estados Unidos, por volta da década de 70. Surgiu então a ARPANET que tinha como fundamento de pesquisas que futuramente dividiu-se com a chamada MILNET que era voltada para fins militares (BRANDINO, 1998).

De acordo com os autores Parziale et al (2006), o protocolo TCP/IP (Protocolo de Controle de Transmissão / Protocolo de Internet) é usado mundialmente para conexão de hosts na rede de internet. O principal objetivo do protocolo TCP/IP foi fazer a interconexão entre redes distintas e de grande distância geográfica, no início conhecida como *internetwork* ou internet.

Parziale et al (2006), ainda mencionam que para os dispositivos conectados à internet possam funcionar, é necessário que cada um tenha o seu endereço na rede, ou seja, o seu endereço IP, para que os pacotes que trafegam

saibam para onde ir. O endereço IP consiste em duas partes, o número da rede e o número do host.

O endereçamento IP somente é responsável por rotear os pacotes na rede a fim de enviar o mesmo para o endereço correto, mas não é capaz de solicitar uma conexão com o dispositivo ou mesmo verificar se ele está disponível na rede. Ai entra o Protocolo TCP que trabalha em uma camada superior no qual faz este controle dos pacotes para aqueles que precisam de garantia de entrega, ou UDP (*User Datagram Protocol*) que trabalha para pacotes sem garantia de entrega. (BRANDINO, 1998).

Enfim TCP/IP é o protocolo de rede que é usado para trafegar os dados na rede e na internet afim de que possam chegar ao seu destino com o mínimo de falhas possíveis e garantir a comunicação entre os dispositivos no mundo todo (PARZIALE et al, 2006).

2.3. CONCEITO DE VOIP

De acordo com Collins (2004), VOIP significa tráfego de voz sobre o Protocolo de Internet. A qualidade da voz depende da banda que é disponibilizada para que ocorra o tráfego. No entanto com a evolução das tecnologias nos últimos anos e o aumento da velocidade da internet isso vem melhorando a cada dia.

A transmissão da voz sobre IP (VOIP) é realizada da seguinte forma, o protocolo RTP carrega a voz codificada pelo dispositivo, transformando em pacotes divididos em amostras significativas, anexando um cabeçalho RTP (*Real-time Transport Protocol*), esses pacotes são enviados para o protocolo UDP (*User Datagram Protocol*) onde mais um cabeçalho é anexado no pacote marcando que é UDP, enfim sendo enviado para o IP que é realizada uma marcação dos endereços IP de origem e destino, onde o mesmo é roteado para seu destino (COLLINS, 2004; ANDRADE et al. 2010).

2.4. PABX IP

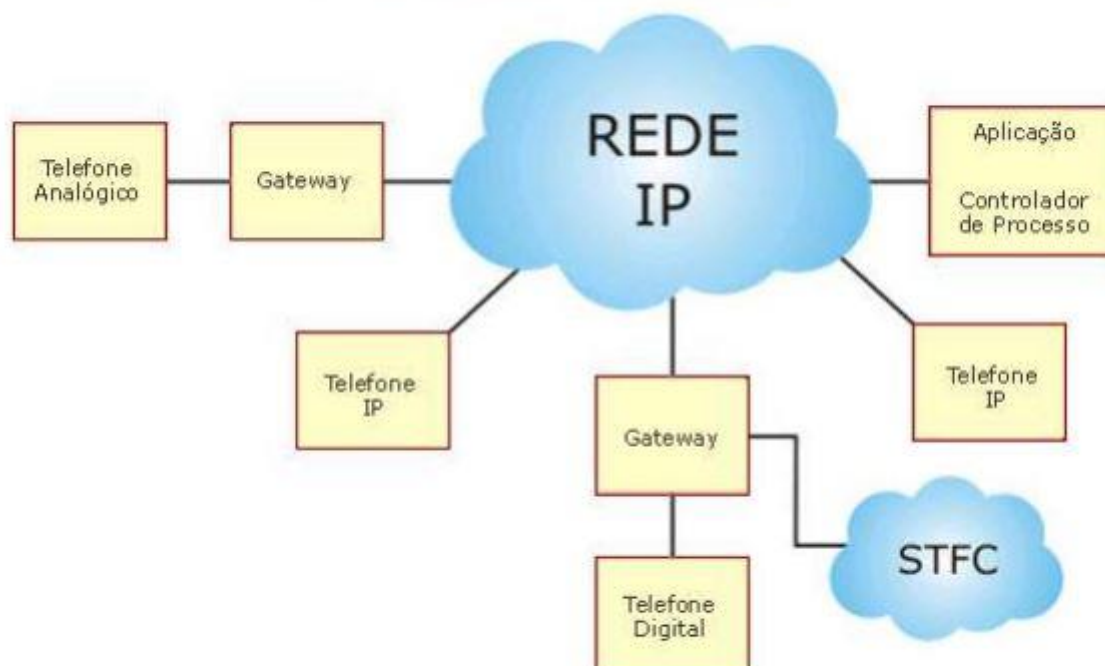
De acordo com Sato (2004), os PABX's convencionais estão presentes nas empresas desde a década de 80 e muito pouco foi evoluído. Depois de muitos anos com a evolução tecnológica das empresas e dos computadores começaram a introduzir no mercado a mudança da arquitetura do PABX. O VOIP surgiu e facilitou essa mudança, sendo capaz de usar a mesma rede de dados com a de voz, mantendo a qualidade das ligações e a confiança. Com a diminuição de custos dos links de internet, dos DSPs (processadores de sinais digitais) que são usados por telefones IP's, a possibilidade para a implantação do VOIP tornou-se viável.

De acordo com Sato (2004), O PABXIP permite que os componentes sejam distribuídos ao longo da rede IP para que sejam transmitidas as ligações. O componente que controla todo o processo é baseado em um software que funciona em cima de um sistema operacional sendo eles Windows, Unix ou Linux, tendo a função de um servidor. Essa utilização do software PABXIP sobre hardwares de servidores possui uma grande vantagem que é a redução de custos para as empresas (SATO, 2004).

Os dispositivos (*endpoints*) são conectados à rede IP onde é necessário que cada um tenha um endereço IP para que possa se comunicar com o servidor e os outros dispositivos da rede (Ramais) (Sato, 2004). Para que as ligações possam ser transmitidas para a rede STFC (Serviço Telefônico Fixo Comutado) são usados os dispositivos chamados de interfaces *Gateway*, que por sua vez fazem a conversão do sinal analógico ou do sinal digital, esses gateways geralmente são usados para converter o sinal de telefones analógicos para a rede digital evitando assim que aumente o custo de migração para aparelhos digitais (Sato, 2004). Um exemplo da arquitetura do PABXIP é mostrado por Sato (2004) na figura 2.

Figura 2: Arquitetura de um PABX Tradicional

Arquitetura de um PABX Tradicional



Fonte: Sato(2004).

2.5. ASTERISK

Segundo Gonçalves (2005), o Asterisk é um software PABX que utiliza o conceito de software livre (GPL – *General Public Licence*). Foi desenvolvido pela empresa Digium. Com o Asterisk é possível conectar em tempo real com redes PSTN (*Public Service Telephony Network*) e redes Voip. A Digium é um dos principais patrocinadores do Asterisk, sendo uma empresa importante no mercado de PABX de código aberto. Uma tradução do site oficial da *Zapata Telephony* explica como surgiu o sistema asterisk juntamente com as placas FXO (*Foreign eXchange Office*) que revolucionaram o mercado de telefonia no mundo. Logo abaixo há um pequeno resumo do surgimento do asterisk contado por Dixon (2009).

2.5.1. História da Telefonia Zapata e como ela se relaciona com PBX Asterisk

Segundo Dixon (2009), há mais de 25 anos atrás a empresa AT&T passou a oferecer uma API aos usuários para que possam customizar as funcionalidades de seu sistema, que era um correio de voz automático, bastante limitado. Esse sistema se chamava Audix. O Audix era muito caro, custava milhares de dólares. (Dixon, 2009). Na época, a dificuldade maior era o poder de processamento das placas FXO (*Foreign eXchange Station*), pois quanto maior o poder de processamento, mais caro ficava a placa. Mas com o aumento do poder de processamento dos computadores, Jim Dixon adquiriu uma placa Mitel89000C *ISDN Express Development Card* e então escreveu um *driver* para o FreeBSD (Dixon, 2009).

Conforme Dixon (2009), descobriu-se que a placa ocupa muito pouco processamento de um Pentium III 600Mhz. Isso provou que se não fosse a limitação do I/O (*Input/Output*), onde a placa não gerenciava de forma eficiente, poderia atender de 50 a 75 canais (DIXON, 2009). Com isso, Dixon (2009), desenvolveu um cartão ISA (*Industry Standard Architecture*) que usasse o I/O de forma eficiente (ALECRIM, 2008). Assim conseguindo dois T1s com 48 canais de dados transferidos sobre o barramento, e o computador gerenciou isso sem apresentar problemas. Dixon conseguiu vender 50 placas e ainda colocou o projeto na web para que outras pessoas pudessem usar (DIXON,2009).

Conforme relata Dixon (2009), o mesmo colocou o nome do projeto de ZAPATA em homenagem ao mexicano Emiliano Zapata e chamou a placa de “tormenta” pelo fato de que ela causaria um impacto no mercado de telefonia. Com o projeto na web surgiram as pessoas que perguntavam se ele tinha o *driver* para Linux, mas como Jim Dixon não tinha experiência com Linux, comprou uma distribuição Red Hat 6.0 e tentou traduzir um *driver*, só que sem sucesso (Dixon, 2009). Dixon (2009) resolveu jogar na web e em 48 horas recebeu um e-mail de uma pessoa chamada Mark Spenser, que ofereceu ajuda no projeto e que tinha algo que seria perfeito para o projeto, que seria o “Asterisk”.

O Asterisk era um sistema que só funcionava na simulação e não tinha interfaces que o faria comunicar realmente com as linhas de telefonia e ramais reais. Com o cartão Zapata, isso permitiu que se tornasse um sistema PABX completo e real, no qual teria um preço justo de telefonia na época (Dixon, 2009). De acordo com Dixon (2009), ele e Mark Spenser formaram um time completo, pois Mark

entendia muito de redes e Linux, coletando informações de telefonia passadas por Dixon. Assim tornaram o Asterisk em um sistema PABX completo e real, revolucionando o mercado de telefonia (DIXON, 2009). O cartão ISA que Jim Dixon desenhou o “Tormenta”, ainda hoje é vendido por empresas como a Digium que tem o nome de T400P e E400P, a empresa Varion também vende o cartão com o nome de V400P (DIXON, 2009). Os arquivos do projeto estão disponíveis no site da digium www.digium.com, segundo Dixon (2009).

2.6. OBSTÁCULOS COM A IMPLANTAÇÃO DE PABX VIA IP

Segundo Gonçalves (2005), o Asterisk usa o CPU do servidor para realizar o processamento dos canais de voz, ao invés de usar o DSP (*processador de sinais de digitais*) dedicado a cada canal. Isso permitiu a redução do custo das placas E1/T1. Assim, o sistema ficou muito dependente do desempenho do CPU. A recomendação seria usar uma máquina dedicada para ser o servidor do Asterisk e rodar sempre em uma VLAN para evitar ataques de *broadcasts* causados por loops ou vírus que podem comprometer o tráfego de voz na rede, pois o processamento das placas de rede fica comprometido quando isso acontece.

2.7. BENEFÍCIOS COM A IMPLANTAÇÃO DE PABX VIA IP

Existem várias vantagens em trocar o sistema PABX tradicional por um sistema PABX Asterisk. A empresa passa a ter controle sobre o sistema e não depender totalmente do analista de telecomunicações, podendo ela mesma adicionar ramais, alterar rotas de ligações, puxar gravações, etc.

Alguns benefícios citados por Gonçalves (2005) são:

- O empregado pode ficar conectado com um ramal em sua casa no PABX da empresa sobre conexão banda larga.

- Conectar vários escritórios em diferentes cidades conectados por VPN (*Virtual Private Network*), assim não tendo nenhum custo com ligações de escritório para empresa ou vice-versa.
- Redução de custos tanto com equipamentos quanto com conta telefônica.
- Ter controle do próprio sistema de telefonia, é de fácil configuração os ramais e outras configurações de rotas, pela interface gráfica.
- Implementação de filas de atendimento de chamadas, onde os agentes ficam ativos para atenderem as ligações que o sistema recebe.
 - Músicas de espera para os clientes da fila.
 - URA personalizada.
 - Registro das chamadas em detalhes.

Destaca-se que muitos outros recursos podem ser implementados nesse sistema.

3. ARQUITETURA DO ASTERISK

3.1. CODECS

Segundo Gonçalves (2005), o Asterisk suporta vários tipos de codecs. A função do codec é converter o sinal de voz de forma que os pacotes de dados fiquem menores e mais leves para trafegar na rede garantindo assim o melhor desempenho e mantendo a qualidade das ligações. Gonçalves (2005) cita que alguns CODECS como o g.729 codificam até 8 Kilobits por segundo, comprimindo de 8 para 1. Na tabela 1 contém alguns exemplos de codecs que o Asterisk suporta, com base em Gonçalves (2005):

Tabela 1. CODECS do Asterisk

Nome do Codec	Descrição
G.711 ulaw	Utilizado nos EUA – 64 Kbps.
G.711 alaw	Usado na Europa e no Brasil – 64 Kbps.
G.723.1	Precisa de licenciamento 5.3-6 Kbps.
G.726	32Kbps no Asterisk 1.0.3, 16/24/32/40 Kbps no CVS HEAD.
G.729	Precisa de licença, a menos que esteja usando o modo passthru. Versão gratuita disponível para uso em países sem patentes ou para uso educacional. (8Kbps)
GSM	12-13 Kbps
iLBC	15 Kbps
LPC10	2.5 Kbps
Speex	2.15-44.2 Kbps

Fonte: Gonçalves (2005).

3.2. PROTOCOLOS

Gonçalves (2005) cita que os protocolos servem para sinalizar os pacotes enviados afim de que isso garanta que cheguem em seu destino corretamente. Segundo Castelucci (2011), protocolos são formados por elementos

que identificam os pacotes na rede e possuem regras que são interpretadas pelas interfaces de comunicação com o intuito de que diferentes interfaces possam comunicar entre si. O Asterisk trabalha com alguns protocolos de comunicação. De acordo com Gonçalves (2005), um dos mais usados é o SIP (*Session Initiated Protocol*). Ainda segundo o autor, o Asterisk suporta os seguintes protocolos:

- SIP
- H323
- IAXv1 e v2
- MGCP
- SCCP (Cisco Skinny).

3.3. MÓDULOS DO ASTERISK

Segundo Bryant, Madsen e Meggelen (2013), o Asterisk é constituído em módulos que são carregados junto com o sistema, esses módulos permitem por exemplo que o Asterisk conecte com o banco de dados, ou mesmo suba um canal de comunicação para uma interface externa. Os módulos do Asterisk carregam com base no arquivo `/etc/asterisk/modules.conf`. A seguir Bryant, Madsen e Meggelen (2013) citam alguns dos principais módulos para um melhor entendimento da arquitetura do Asterisk.

3.3.1. Aplicações

De acordo com Bryant, Madsen e Meggelen (2013), as aplicações de Dialplan são carregadas com base no arquivo `extensions.conf` que são responsáveis por definir várias ações do sistema como por exemplo realizar conexões de saída para recursos externos, configurar ações de discagem e outras funções. Está disponível na tabela 2 algumas de suas principais aplicações:

Tabela 2. Aplicações do Asterisk

Nome do arquivo	Especificações
<i>app_cdr</i>	É usada para gravar os registros das ligações.
<i>app_chanspy</i>	Permite que um canal possa ouvir o áudio de outro canal.
<i>app_dial</i>	É utilizado para conectar canais juntos ou seja, fazer chamadas telefônicas, é essencial para a aplicação.
<i>app_record</i>	Grava registros de áudio recebidos em arquivo.
<i>app_record</i>	Fornecer alguns utilitários relacionados ao reconhecimento de voz.
<i>app_system</i>	<i>Executa comandos em um shell do Linux, é considerado útil.</i>

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.2. Módulos Ponte

Bryant, Madsen e Meggelen (2013), explicam que os módulos ponte (*bridging modules*) fornecem alguns diferentes recursos que são usados em diferentes situações dependendo do que a ponte necessita. Esses módulos são configurados no arquivo *app_confbridge*. Alguns dos arquivos de configuração de módulos ponte estão listados na tabela 3:

Tabela 3. Módulos de ponte do Asterisk

Nome do arquivo	Especificações
<i>bridge_multiplexed</i>	Este é configurado para realizar a multiplexação em salas de conferências com suporte a vários participantes.
<i>bridge_simple</i>	<i>Executa uma ponte simples de canal para canal.</i>
<i>bridge_softmix</i>	É usado para executar multiplexação em pontes simples em conferências suportando vários participantes.

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.3. Módulos de gravação de detalhes de chamadas

De acordo com Bryant, Madsen e Meggelen (2013), o CDR (*Call Detail Recording*), foi projetado para facilitar os métodos de armazenamento de registros de detalhes de gravações. Dando suporte para armazenar os CDRs em arquivos que é por padrão, em um banco de dados, em *Remote authentication dial in user servisse* (RADIUS), ou syslog o arquivo onde são armazenados os logs do Linux. Segue alguns detalhes dos arquivos de configuração do CDR na tabela 4:

Tabela 4. Módulos de gravação de detalhes de chamada do Asterisk

<i>Nome do Arquivo</i>	<i>Especificações</i>
<i>cdr_adaptive_odbc</i>	este arquivo permite a escrita de CDRs através de ODBC, permitindo adicionar campos personalizados.
<i>cdr_csv</i>	Este grava CDRs no disco com separação por vírgulas.
<i>cdr_custom</i>	Grava CDRs em um arquivo CSV, que permite a adição de campos personalizados.
<i>cdr_odbc</i>	Executa a gravação de CDRs através de framework ODBC.
<i>cdr_pgsq</i>	Direciona os registros para um banco de dados PostgreSQL.
<i>cdr_radius</i>	Grava CDRs para o servidor RADIUS, um detalhe é que ele não suporta campos personalizados.
<i>cdr_sqlite3_custom</i>	Direciona os registros para um banco SQLite3 permitindo campos personalizados.
<i>cdr_syslog</i>	Faz a gravação dos CDRs no syslog.
<i>cdr_tds</i>	Realiza a gravação dos registros para um banco Microsoft SQL ou Sybase. (Este requer uma versão antiga do libtds).

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.4. Módulos de Registros de Evento de Canal

Segundo Bryant, Madsen e Meggelen (2013), o CEL (Controle de Eventos de Chamada), pode oferecer um controle amplo das chamadas. Mas ele

requer um planejamento mais pontual do dialplan (plano de discagens). Os autores mencionam alguns módulos do CEL e suas funcionalidades na tabela 5:

Tabela 5. Módulos de Registros de Evento de Canal

Nome do arquivo	Funcionalidade
<i>cel_custom</i>	armazena os registros em disco / arquivo
<i>cel_manager</i>	Controle de Eventos de Chamada para AML do Asterisk.
<i>cel_odbc</i>	<i>CEL para ODBC</i>
<i>cel_pqsql</i>	<i>CEL armazenados em banco de dados PostgreSQL.</i>
<i>cel_radius</i>	<i>Armazena os registros em serviço RADIUS.</i>
<i>cel_sqlite3_custom</i>	<i>direcionado para banco de dados SQLite3.</i>
<i>cel_tds</i>	<i>Envia os registros para Microsoft SQL ou Sybase.</i>

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.5. Controladores de canal

Bryant, Madsen e Meggelen (2013), mencionam que sem os drivers de canais, o Asterisk não conseguiria realizar chamadas, pois cada driver é específico para o tipo de canal como: (SIP, ISDN e outros), ou tipo de protocolo. O módulo de canal atua no Asterisk como um gateway em seu núcleo. Segue alguns dos controladores e suas funções na tabela 6:

Tabela 6. Controladores de Canal do Asterisk

Nome do arquivo	Funcionalidade
Chan_agent	Fornece um canal de agente para fila de chamadas
Chan_alsa	Realiza a conexão com o Advanced Linux Sound Architecture
Chan_dahdi	fornece a conexão com as placas PSTN que usam o canal DAHDI.
Chan_gtalk	Fornece a conexão para o Google Talk, porém não é mais utilizada a partir da versão 11 do Asterisk.
Chan_iax2	conecta os terminais IAX2.
Chan_local	Trata uma parte do dialplan como um canal
Chan_misdn	Fornece conexão a cartões ISDN suportados pela MISDN
Chan_multicast_rtp	Fornece conexão para multicast RTP (Realtime Transport Protocol).

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.6. Tradutores de codec

Bryant, Madsen e Meggelen (2013), explicam que os tradutores de Codec tem o papel de realizar a tradução dos diferentes codecs que são usados nas aplicações. Eles mencionam o seguinte exemplo: “Então, se uma chamada entra em um circuito PRI (usando G.711) e precisa ser passado para um canal SIP comprimido (por exemplo, usando G.729, um dos muitos codecs que o SIP pode lidar), o conversor de codec relevante executaria a conversão” (BRYANT, MADSEN e MEGGELEN, 2013). Os autores alertam que um codec como o G.729 que usa um algoritmo complexo de codificação, pode causar um impacto significativo na CPU e é

recomendado consultar os fabricantes como da Sangoma e Digium para não ter maiores problemas com processamento. Os autores mencionam alguns arquivos de codec que contem no módulo Tradutor de Codec na tabela 7:

Tabela 7. Tradutores de Codec

Nome do arquivo	Descrição
Codec_alaw	codec PCM A-law usado em todo o mundo (exceto Canadá / EUA) no PSTN Essential
Codec_g729	É um codec que não vem nativo no Asterisk, pois não é de domínio público, mas pode ser adquirido pela Digium.
Codec_dahdi	usa o cartão de transcodificação de hardware Digium proprietário.
Codec_g722	É um codec de áudio de banda larga.
Codec_gsm	Um sistema global para codec de comunicações Móveis (GSM).
Codec_ulaw o Mu-law	O codec PCM é usado no Canadá/EUA em PSTN Essential.

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.7. Funções do Dialplan

As funções do Dialplan são usadas para complementar as aplicações do Dialplan, por exemplo, as manipulações de tempo e data ou até mesmo a conectividade com ODBC. (BRYANT, MADSEN e MEGGELEN 2013). Na tabela 8 os autores mostram alguns dos arquivos de configurações das funções do Dialplan e suas funcionalidades.

Tabela 8. Funções do Dialplan

Nome do arquivo	Funcionalidade
func_audiohookinherit	Permite que as chamadas sejam gravadas após a transferência.
Func_blacklist	Grava / lê uma lista negra em astdb, é muito útil.
Func_callerid	Obtém / define CallerID
Func_cdr	Define uma variável CDR
Func_channel	Obtém / define a informação do canal
Func_connectedline	Muda informações da linha conectada em aparelhos suportados
Func_curl	Usa cURL para obter dados de uma URL
Func_devstate	Obtém o estado do dispositivo
Func_dialgroup	Cria um grupo para discagem simultânea
Func_dialplan	Valida que o destino designado existe no dialplan
Func_global	Obtém / define a contagem de canais para membros de um grupo
Func_module	Verifica se o modulo fornecido está carregado na memória
Func_odbc	Permite a integração do dialplan com recursos do ODBC
Func_redirecting	Fornece acesso a informações sobre onde a chamada foi direcionada.
Func_speex	Reduz o ruído e executa o ganho / perda de dB em um fluxo de áudio.
Func_sysinfo	Obtém informações do Sistema como RAM, swap, carga média e etc.
Func_volume	Define o volume em um canal

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.8. Módulos PBX

Segundo Bryant, Madsen e Meggelen (2013), os módulos PBX são arquivos que oferecem o controle e configurações mais aprimoradas para o Asterisk. Por exemplo o pbx_config que é um módulo que carrega o dialplan do Asterisk. Na tabela 9 segue alguns dos principais módulos do Asterisk:

Tabela 9. Modulos do PBX

Nome do Arquivo	Funcionalidade
Pbx_ael	Oferece uma linguagem de script dialplan que se parece com uma linguagem de programação moderna.
Pbx_config	Esta é a linguagem de discagem tradicional e mais popular do Asterisk, sem ela o módulo do Asterisk não consegue ler o extensions.conf.
Pbx_undi	Executa pesquisas de dados em sistemas Asterisk remotos.
Pbx_loopback	Realiza algo semelhante a um include dialplan, mas de forma depreciative.
Pbx_lua	Permite a criação de um dialplan usando a linguagem de script Lua.
Pbx_realtime	Fornecer funcionalidades relacionadas à arquitetura Asterisk Realtime.
Pbx_spool	Fornecer suporte de spool de saída relacionado a arquivos de chamada do Asterisk.

Fonte: Bryant, Madsen e Meggelen (2013).

3.3.9. Módulos de recursos

De acordo com Bryant, Madsen e Meggelen (2013), os módulos de recursos, são módulos que servem para integrar o Asterisk com recursos externos, como os arquivos de configuração Backends. O Asterisk é configurado pelos seus arquivos de texto em /etc/asterisk/ por padrão. Existem alguns módulos alternativos de configuração de banco de dados, alguns exemplos seguem na tabela 10:

Tabela 10. Modulos de Recursos de banco de dados.

Nome do Arquivo	Funcionalidade
Res_config_curl	Puxa as informações das configuração usando CURL
Res_config_odbc	Puxa as informações de configuração usando ODBC
Res_config_pgsq	Puxa as informações de configuração usando o PostgreSQL
Res_config_sqlite	Puxa as informações de configuração usando SQLite

Fonte: Bryant, Madsen e Meggelen (2013).

Bryant, Madsen e Meggelen (2013), mencionam também os modulos de temporização, pois alguns recursos do Asterisk necessitam de fonte de tempo, por exemplo conferências usando o aplicativo ConfBridge. Para isso existe alguns arquivos de configuração para o uso dessas aplicações, por exemplo o arquivo `res_timing_dahdi`, que fornece o tempo usando a interface do Kernel DAHDI, o arquivo `res_timing_timerfd` que retorna o tempo usando a API `timerfd` fornecida por versões mais recentes do Kernel do Linux. O Asterisk também fornece o modulo de calendário onde é retornado a data, sendo possível conectar em determinados servidores de calendário. (Bryant, Madsen e Meggelen, 2013).

Segundo Bryant, Madsen e Meggelen (2013), os módulos complementares são componentes onde a comunidade colaboradora do Asterisk, realizam o desenvolvimento afim de adicionar outras funções ao sistema. Esses softwares possuem diferentes distribuições sendo elas com direitos de propriedades do código ou código aberto. Esses softwares possuem diferentes funcionalidades, por exemplo uma aplicação para consultas MySQL, entre outras.

4. MODELO DE NEGÓCIO

Segundo Osterwalder e Pigneur (2010), o modelo de negócio é uma forma de descrever um negócio em uma linguagem que permite ser facilmente manipulada e de fácil entendimento, facilitando a criação de novas estratégias e obtendo ampla visão do que está sendo criado ou proposto.

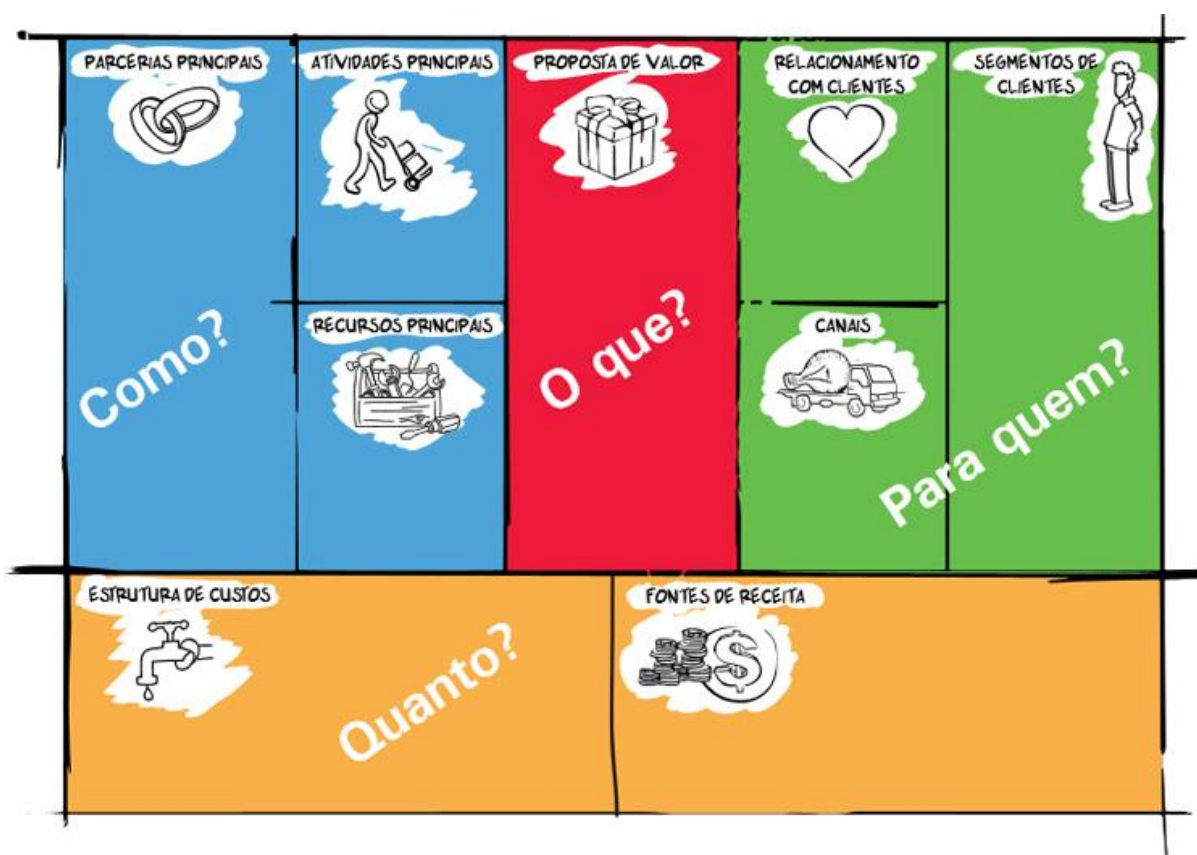
De acordo com Osterwalder e Pigneur (2010), o modelo de negócio permite que seja proposta uma ideia de uma empresa montada e como ela iria funcionar, que geralmente é colocada em um quadro, permitindo que o empreendedor possa ter uma boa visão de todas as partes relevantes para que a ideia seja colocada em prática, assim sendo possível verificar os valores que a empresa irá agregar, tanto a sociedade quanto ao empreendedor, também os nichos de mercados, quais são exatamente seus clientes, qual será a relação entre empresa com fornecedores e muito outros elementos importantes na hora da criação de um negócio.

Osterwalder e Pigneur (2010) mostram que para montar um modelo de negócio devemos usar nove elementos, são eles: Segmentos de Clientes, Proposta de Valor, Canais, Relacionamento com Clientes, Fontes de Receita, Recursos Principais, Atividades-Chave, Parcerias Principais e Estrutura de Custo. A figura 3 ilustra um quadro no formato para criação de um modelo de negócio.

De acordo com os autores Osterwalder e Pigneur (2010), é importante seguir os tópicos e ir analisando como seria a empresa, o primeiro tópico do quadro a ser preenchido é o Segmento de Clientes. Na questão do segmento de clientes o empreendedor deverá analisar quais serão os clientes nos quais a empresa irá atender, devendo descrever qual é o nicho, o foco de clientes da empresa. É um dos principais tópicos pois sem clientes nenhuma empresa sobrevive (OSTERWALDER e PIGNEUR, 2010).

O Segundo tópico é a Proposta de valor, segundo Osterwalder e Pigneur (2010), neste campo é necessário descrever o pacote de produtos e serviços que criam valor para os clientes mencionados no primeiro tópico. A proposta de valor é um pacote que atende as necessidades de um cliente específico ou resolve algum problema gerando valor à empresa atendida, a proposta pode ser um produto inovador ou até mesmo um já existente mas com atributos adicionais.

Figura 3. Quadro de modelo de Negócios



Fonte: Sebrae (2017)

De acordo com Osterwalder e Pigneur (2010), o terceiro tópico é chamado de Canais. Em Canais, é descrito como a empresa irá se comunicar e alcançar seus segmentos de clientes para que seja entregue a proposta de valor. Os canais têm a função de ampliar o conhecimento dos clientes sobre os produtos e serviços oferecidos pela empresa, ajuda a avaliar a proposta de valor, permite que o cliente adquira produtos e serviços, leva a proposta diretamente ao cliente e fornece suporte após a compra do serviço (OSTERWALDER e PIGNEUR, 2010).

O quarto tópico é o Relacionamento de Clientes. Neste componente o empreendedor deverá descrever como a empresa se relacionará com o seu cliente, se será uma relação pessoal, automatizada, se é uma relação para ampliação de vendas, se a empresa quer manter a fidelização do cliente ou se é para conquistar novos clients. Tudo isso vai depender do objetivo do plano de negócio que foi traçado.

O tipo de relação que a empresa tem com o cliente influencia muito nos negócios e na experiência do cliente com a empresa (OSTERWALDER e PIGNEUR, 2010).

A quinta parte, conforme mencionado pelos autores Osterwalder e Pigneur (2010), é a Fonte de Receita. A Fonte de Receita representa a receita da empresa, ou seja, como a empresa irá ganhar dinheiro, de onde virá o dinheiro. É necessário verificar também se o valor a ser cobrado é o que os clientes estão dispostos a pagar, e também verificar se com esse valor é possível cobrir as despesas da empresa. Existem tipos diferentes de receitas, que deverão ser analisadas como o produto será adquirido, se de uma renda única ou se o cliente irá pagar uma mensalidade, que no caso será uma renda de suporte ou manutenção.

Recursos Principais consistem no sexto tópico. Nesse passo é necessário descrever todos os principais recursos que a empresa deverá ter, para que ela possa se manter e atender as necessidades dos seus clientes. Esses recursos são tantos físicos quanto humanos. Por exemplo, a empresa precisa de um espaço físico para alocar suas máquinas, seus computadores, irá também precisar de pessoas para manusear as máquinas e os computadores, para transportar os produtos produzidos, também terá que ter um caminhão. Para comprar a matéria-prima também é necessário ter um capital reservado a fim de manter os recursos. Todos os principais recursos deverão ser descritos neste tópico (OSTERWALDER e PIGNEUR, 2010).

Atividades-Chave é a sétima parte a ser preenchida no quadro. Segundo Osterwalder e Pigneur (2010), as atividades-chave são as principais atividades que um modelo de negócio deverá executar para poder atender as necessidades de seus clientes e que geram valor a eles. Neste tópico, o empreendedor deverá descrever todas as atividades que o seu negócio irá realizar para concluir a proposta de valor. Por exemplo, em uma empresa de software, a sua atividade-chave é o desenvolvimento de software.

A penúltima e oitava fase é Parcerias Principais, conforme mencionado por Osterwalder e Pigneur (2010), em parcerias principais é necessário descrever possíveis parcerias que a empresa poderá ter. Essas parcerias são importantes pois podem otimizar a entrega do produto, a qualidade, a satisfação com o cliente e também até a economia na matéria prima ou mão de obra. Muitas empresas fazem essas parcerias estratégicas pois é bom para os dois lados, por exemplo, a empresa

faz uma parceria com um fornecedor, compra a matéria-prima por um valor menor, porém tem a fidelidade de comprar somente com ele, garantindo a entrega no prazo e a qualidade no produto. Isso pode ajudar muito nas vendas e na lucratividade dos dois lados.

Nono tópico a ser preenchido no quadro de modelo de negócios é a Estrutura de Custos. De acordo com Osterwalder e Pigneur (2010), na Estrutura de Custos o empreendedor deverá esboçar todos os principais custos que são necessários para o funcionamento do negócio. Para que a empresa possa gerar receita ela depende de atividades que contém um custo, e todos esses custos terão que ser descritos para que tenham uma visão da viabilidade do negócio. Todas as atividades principais que foram descritas deverão ser analisadas em termos de seus custos. Assim será possível visar a lucratividade da empresa. Algumas perguntas deverão ser respondidas para isso, por exemplo: Quais são os custos mais importantes do modelo de negócios? Quais os recursos principais mais caros? Quais atividades chave mais caras? Assim finalizando todo o quadro de modelo de negócios, é possível fazer uma análise geral do modelo, concluindo todo o ciclo de uma empresa e dando visão ao empreendedor se é viável ou não realizar o investimento.

O que é uma Startup? Segundo Normand (2017), startup é uma empresa que começa sem dinheiro, com uma ideia e visão de longo prazo, o seu ativo principal são seus fundadores e colaboradores. As startups são empresas que buscam o crescimento exponencial, na qual desejam atingir milhões ou bilhões de consumidores no mundo em um curto período. Para isso é usada a tecnologia em seus processos mais tradicionais sendo mais eficientes e de menor custo, diz Normand (2017).

A ideia é que o projeto mostrado anteriormente usando Asterisk com Laravel, possa se tornar uma Startup, analisando o produto, montando um modelo de negócio voltado para uma empresa neste segmento pode ser interessante, pensando nisso, foi criado um modelo de negócio em cima deste produto, a fim de despertar o interesse dos empreendedores.

4.1. MODELO DE NEGÓCIO DE UMA EMPRESA DE TELEFONIA ASTERISK

4.1.1. Segmento de Clientes

O produto desenvolvido, dá suporte para pequenas e médias empresas, claro que uma versão mais completa pode dar suporte também para empresas de grande porte sem maiores problemas, mas o produto mostrado é focado para pequenas e médias empresas. O PABXIP pode ser instalado em qualquer empresa que precise de uma URA de atendimento, que necessite tirar relatórios dos números de ligações mensais, diárias ou alguma ligação específica. Portanto, toda empresa que possui um PABX analógico, que tem grande custo de manutenção, ou não, que queira modernizar, atualizar ou diminuir custos com ligações e ter uma visão mais completa do seu sistema de telefonia, pode adquirir o produto.

4.1.2. Proposta de valor

Para esse produto a proposta de valor é principalmente a economia na manutenção dos equipamentos de telefonia. Outros fatores também agregam valores aos clientes que vão usar o sistema, são eles:

- Gerenciamento do sistema de telefonia
- Redução de custos com manutenção e licenças de equipamentos e softwares
- Manutenção reduzida
- Redução no valor da conta de telefone.
- Possibilidade de realizar métricas das ligações efetuadas e recebidas
- Gravações das ligações
- Relatórios de ligações
- Obter os próprios equipamentos de telefonia

4.1.3. Canais

Os principais canais de comunicações onde serão divulgados o produto e os serviços para essa empresa, serão os mencionados abaixo:

Internet – Pela internet será realizada divulgações através do site da empresa, campanhas no Facebook, Instagram, LinkedIn e outros meios de comunicação online.

Indicações de clientes – Seria a famosa propaganda boca a boca, que no caso algum cliente case de sucesso que já usa o produto pode fazer a indicação do mesmo para um outro possível cliente, aumentando ainda mais as chances de efetivação da venda.

Contato diretamente com a empresa – Um consultor de vendas, pode entrar em contato com as empresas que tem potencial para uma possível negociação, afim de apresentar o produto e suas soluções, destacando suas vantagens.

Outros canais após o cliente adquirir o produto serão utilizados, por exemplo: Um cliente necessita de um atendimento do helpdesk para uma manutenção no sistema, vai acessar a interface de abertura de chamados, onde irá descrever o problema e solicitando a ajuda, a equipe irá receber essa solicitação e escalonará o chamado para o profissional responsável por atender a ordem de serviço, assim facilitando o atendimento as empresas.

4.1.4. Relacionamento com clientes

O objetivo deste plano de negócio é gerar um modelo de uma startup na qual irá prestar serviços e suporte no sistema de telefonia, que oferece para seus clientes. O tipo de relacionamento que a empresa terá com seus clientes será de duas formas, uma é a relação pessoal e outra é relação automatizada. A relação pessoal, irá acontecer quando os gestores ou técnicos precisarem falar diretamente com o cliente, seja por uma negociação, ou até mesmo por uma manutenção no local da empresa do cliente.

A relação automatizada, será através do sistema de chamados no qual o cliente irá solicitar algum atendimento ou esclarecimento de dúvidas, usando o sistema, e todas as respostas serão enviadas pelo mesmo, pode ser que algum atendimento específico seja necessário realizar uma ligação diretamente para o cliente e através do telefone o contato é estabelecido. O contato automatizado tem o intuito de agilizar e deixar todas as solicitações registradas em um sistema sendo possível o monitoramento de cada cliente e até mesmo tirar relatórios nos quais

possam servir para atendimentos não de manutenção e sim de prevenção de problemas futuros, ajudando na estabilidade do sistema de telefonia e até mesmo mantendo a qualidade nos produtos, fidelizando ainda mais o cliente.

4.1.5. Fontes de receita

As receitas da empresa serão geradas a partir de duas frentes, uma é a implementação inicial do sistema, ou seja, a instalação do sistema na empresa, na qual irá gerar uma receita. Outra receita será o valor do contrato mensal de uso do sistema e suporte do mesmo, caso o cliente precise de algum suporte de manutenção ou configurações de acordo com o seu perfil durante os meses de uso. Essa segunda opção é a mais importante, a qual irá garantir receita de todos os clientes todos os meses. O valor da implementação e o valor da mensalidade, vai depender da quantidade de ramais que a empresa tem e de seu fluxo de ligações, o valor será calculado tomando como base esses dois fatores.

4.1.6. Recursos principais

A startup terá como recursos principais os seguintes itens:

- A empresa deverá ter um escritório onde tenha uma sala para o atendimento do helpdesk.
- A empresa deverá ter um carro caso precise deslocar até o cliente ou outros serviços relacionados, como buscar um servidor ou entregar.
- Um servidor para armazenar as informações e alocar o sistema de gerenciamento de chamados e administrativo da empresa.
- Um funcionário analista em telefonia.
- Um funcionário analista em desenvolvimento web e com conhecimentos de telefonia e Asterisk.
- Equipamentos para o funcionamento da empresa: telefones, mesas e cadeiras para escritório, computadores, notebooks.

4.1.7. Atividades chave

A principal atividade é oferecer um sistema de telefonia onde o cliente possa substituir o PABX analógico por um digital que funciona com tecnologia IP. Onde ele possa monitorar o fluxo de ligações, gravar chamadas, visualizar relatórios das ligações realizadas, para que o cliente possa usar isso a seu favor, usando de técnicas estatísticas.

Outras atividades são: prestar consultoria as empresas com o intuito de minimizar os custos com as contas telefônicas e dar o suporte necessário para que o sistema se mantenha estável.

4.1.8. Parcerias principais

As principais parcerias que a empresa inicialmente deverá ter para que possa economizar com tempo e entrega de produtos será com os atores descritos abaixo:

- Fornecedores de Servidores
- Fornecedores de equipamentos telefônicos
- Fornecedores de equipamentos de rede (switch, roteadores, etc.)
- Operadoras de telefonia Voip
- Operadoras de telefonia

4.1.9. Estrutura de custos

Para que a empresa se mantenha em pleno funcionamento é necessário que a mesma pague todas as suas despesas. Um breve levantamento mostra alguns dos principais possíveis custos que a startup terá durante o mês, e estão descritos abaixo:

- Aluguel do escritório
- Funcionários – Salário, férias, impostos
- Internet - pacote de internet mensal
- Telefone – custos de telefonia
- Energia elétrica

- Investimento em equipamentos (Telefones, computadores, móveis, impressoras, servidor, equipamentos de rede.)
- Carro (combustível, Impostos)
- Escritório de contabilidade – mensalidade do escritório de contabilidade, pois no início a empresa não terá um departamento financeiro.
- Pró-labore dos sócios

5. IMPLEMENTAÇÃO DE ASTERISK PARA PABXIP

O objetivo deste trabalho é mostrar que a tecnologia Open Source pode ser usada como uma solução para as empresas e ajudá-las a economizar dinheiro com manutenção de licenças de equipamentos de telefonia. Como exemplo foi implantada uma versão do sistema em uma Instituição de Ensino Superior. O coordenador de tecnologia foi procurado para que fosse mostrada a proposta de tema, assim analisada por ele. Foi aprovada para a pesquisa e então recebeu-se a proposta de instalar um servidor na própria instituição de ensino. Depois de uma análise da infraestrutura e também algumas conversas e reuniões com o setor de TI da instituição, foi aprovada implementação deste servidor. Sendo assim, partiu-se para o processo de implementação de uma máquina virtual onde foi configurado um servidor PABXIP Open Source, que permite ligações de ramal para ramal dentro da instituição de ensino superior. Essas ligações podem ser monitoradas a partir de uma interface web que mostra os detalhes das ligações, como hora, ramal de destino, ramal de origem, tempo de ligação e até a gravação do áudio falado durante a ligação.

5.1. INSTALAÇÃO DO SERVIDOR

Neste capítulo será descrito passo a passo como foi realizada uma implementação de um servidor PABXIP, usando o sistema operacional Linux Ubuntu Server com Asterisk. Foi implementada também uma interface gráfica usando PHP e o Framework Laravel para a administração das ligações entre os ramais, permitindo assim ouvir o áudio da ligação, visualizar a data, a hora, ver qual é o ramal de origem e destino e também baixar o arquivo de áudio para escuta externa.

Na primeira parte da implementação foi realizada a instalação do Virtual Box Versão 5.1.6 com 64 bits. O Virtual Box é um software de virtualização x86 e AMD64 / Intel64 para uso de empresas ou também doméstico. É uma solução de código aberto nos termos da versão GNU General Public Licence (GPL) 2 (VIRTUAL BOX, 2017) . Foi criada uma máquina virtual com o sistema operacional Ubuntu Server versão 14.04.5 LTS de 64 bits. O Processador é um i3-2310M 2.10GHz, a máquina

virtual possui 1 GB de memória RAM, 18GB de armazenamento do disco, uma placa de rede em modo Bridge de 100MBs.

5.2. INSTALAÇÃO DO ASTERISK

Para iniciar a instalação do Asterisk, é necessário fazer a atualização dos pacotes do Ubuntu usando o comando 'apt-get update' para que o Linux possa baixar os pacotes. Após isso, o 'apt-get upgrade' para que o sistema possa instalar os pacotes de atualizações que foram baixados. Conforme ilustra as figuras 4 e 5. A figura 4 ilustra a execução do comando apt-get update. A figura 5 ilustra a execução do comando apt-get upgrade.

Figura 4. Execução do comando apt-get update

```
root@ubuntu:/home/tbs# apt-get update
Obter:1 http://security.ubuntu.com trusty-security InRelease [65,9 kB]
Obter:2 http://security.ubuntu.com trusty-security/main Sources [133 kB]
Obter:3 http://security.ubuntu.com trusty-security/restricted Sources [4.955 B]
Obter:4 http://security.ubuntu.com trusty-security/universe Sources [59,3 kB]
Obter:5 http://security.ubuntu.com trusty-security/multiverse Sources [3.189 B]
Obter:6 http://security.ubuntu.com trusty-security/main amd64 Packages [628 kB]
Ign http://br.archive.ubuntu.com trusty InRelease
Obter:7 http://br.archive.ubuntu.com trusty-updates InRelease [65,9 kB]
Atingido http://br.archive.ubuntu.com trusty-backports InRelease
Obter:8 http://br.archive.ubuntu.com trusty Release.gpg [933 B]
Obter:9 http://br.archive.ubuntu.com trusty-updates/main Sources [400 kB]
Obter:10 http://security.ubuntu.com trusty-security/restricted amd64 Packages [1
```

Fonte: Elaborado pelo autor

Figura 5. Execução do comando `apt-get upgrade`

```

root@ubuntu:/home/tbs# apt-get upgrade
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
Calculando atualização... Pronto
Os pacotes a seguir serão mantidos em suas versões atuais:
  linux-generic-lts-xenial linux-headers-generic-lts-xenial
  linux-image-generic-lts-xenial
Os pacotes a seguir serão atualizados:
  apport bind9-host dnsutils isc-dhcp-client isc-dhcp-common klibc-utils
  libbind9-90 libc-bin libc-dev-bin libc6 libc6-dev libdns100 libgcrypt11
  libgcrypt11-dev libgnutls-dev libgnutls-openssl27 libgnutls26 libgnutlsxx27
  libisc95 libisc95-dev libisc95-g90 libklibc libldap-2.4-2 libldap2-dev
  liblwres90 libnl-3-200 libnl-genl-3-200 libnss3 libnss3-nssdb libpam-smbpass
  libpq-dev libpq5 libsmbclient libtasn1-6 libtasn1-6-dev libtiff5
  libtiff5-dev libtiffxx5 libwbclient0 linux-libc-dev logrotate
  multiarch-support ntpdate python-samba python3-apport python3-problem-report
  python3-software-properties samba samba-common samba-common-bin samba-doc
  samba-dsdb-modules samba-libs samba-vfs-modules smbclient
  software-properties-common sudo winbind
58 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 3 não atua-
lizados.
É preciso baixar 25,1 MB de arquivos.
Depois desta operação, 21,5 kB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] s
Obter:1 http://br.archive.ubuntu.com/ubuntu/ trusty-updates/main libc6-dev amd64 2.1
9-0ubuntu6.13 [1.914 kB]

```

Fonte: Elaborado pelo autor

Outros pacotes adicionais também são necessários para que o Asterisk funcione corretamente, para isso fizemos a instalação dessas dependências (quais?) com o seguinte comando:

```
apt-get install build-essential wget libssl-dev libncurses5-dev libnewt-dev
libxml2-dev linux-headers-$(uname-r) libsqlite3-dev uuid-dev git-core
subversion libjansson-dev sqlite autoconf automake libtool libncurses5-
dev -y
```

Após a instalação das dependências, é necessário baixar os arquivos nos quais vamos usar para instalar o Asterisk. Primeiramente acessa-se a pasta `src` onde salva-se os arquivos, com o comando `cd /usr/src/`. Estando dentro da pasta mencionada, é necessário digitar o comando a seguir para baixar o arquivo do driver DAHDI (Digium / Asterisk Hardware Device Interface) que segundo o site oficial www.asterisk.org é uma tecnologia de interface de dispositivo de código aberto que controla as placas Digium e outras placas de telefonia. Sem esse driver não é possível

instalar e usar o Asterisk. A seguir está o comando para baixar o driver:

wget downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-2.10.2+2.10.2.tar.gz

Em seguida, é necessário descompactar o arquivo com o comando `tar xzvf 'nome do arquivo'`. Feito isso é criada uma pasta onde devemos acessá-la para fazer a instalação. Antes de realizar a compilação do arquivo será necessário instalar mais uma dependência, o comando é `apt-get install linux-headers-`uname -r``. Depois de instaladas essas dependências, compila-se o driver DAHDI com o comando `make && make install && make config`. O final da instalação do driver DAHDI com o comando anterior é ilustrado na figura 6.

Figura 6. Instalação do driver DAHDI

```
#####
###
### DAHDI tools installed successfully.
### If you have not done so before, install init scripts with:
###
###   make config
###
#####
make[1]: Saindo do diretório `/usr/src/dahdi-linux-complete-2.10.2+2.10.2/tools'
make -C tools config
make[1]: Entrando no diretório `/usr/src/dahdi-linux-complete-2.10.2+2.10.2/tools'
install -D dahdi.init /etc/init.d/dahdi
/usr/bin/install -c -D -m 644 init.conf.sample /etc/dahdi/init.conf
/usr/bin/install -c -D -m 644 modules.sample /etc/dahdi/modules
/usr/bin/install -c -D -m 644 xpp/genconf_parameters /etc/dahdi/genconf_parameters
/usr/bin/install -c -D -m 644 modprobe.conf.sample /etc/modprobe.d/dahdi.conf
/usr/bin/install -c -D -m 644 blacklist.sample /etc/modprobe.d/dahdi.blacklist.conf
/usr/bin/install -c -d /etc/udev/rules.d
/usr/bin/install -c -D -m 644 dahdi.rules /etc/udev/rules.d/
/usr/sbin/update-rc.d dahdi defaults 15 30
Adding system startup for /etc/init.d/dahdi ...
/etc/rc0.d/K30dahdi -> ../init.d/dahdi
/etc/rc1.d/K30dahdi -> ../init.d/dahdi
/etc/rc6.d/K30dahdi -> ../init.d/dahdi
/etc/rc2.d/S15dahdi -> ../init.d/dahdi
/etc/rc3.d/S15dahdi -> ../init.d/dahdi
/etc/rc4.d/S15dahdi -> ../init.d/dahdi
/etc/rc5.d/S15dahdi -> ../init.d/dahdi
DAHDI has been configured.

List of detected DAHDI devices:

No hardware found
make[1]: Saindo do diretório `/usr/src/dahdi-linux-complete-2.10.2+2.10.2/tools'
```

Fonte: Elaborado pelo autor

Após instalado o driver DAHDI, instala-se o Libpri. O Libpri é uma biblioteca de código aberto que encapsula os protocolos usados para se comunicar através de interfaces de taxa primária ISDN (T1, E1, J1). Libpri é uma dependência para Asterisk e DAHDI se a sinalização PRI for utilizada. (Asterisk , 2017). Primeiro deve-se executar o comando para fazer o download do Libpri, de acordo com o comando a seguir:

wget <http://downloads.asterisk.org/pub/telephony/libpri/libpri-current.tar.gz>

Após terminar o download do arquivo, descompacta-se o mesmo com comando `tar zxvf 'nome do arquivo'`. Em seguida, acessa-se a pasta que será criada com o arquivo descompactado usando o comando `cd /'nome da pasta'` e executa-se o comando `make && make install` conforme ilustra-se uma parte da saída do comando na figura 7.

Figura 7. Instalação do libpri

```
root@asterisk:/usr/src/libpri-1.6.0# make && make install
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT
copy_string.o -MF .copy_string.o.d -MP -c -o copy_string.o copy_string.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT
pri.o -MF .pri.o.d -MP -c -o pri.o pri.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT
q921.o -MF .q921.o.d -MP -c -o q921.o q921.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT
prished.o -MF .prished.o.d -MP -c -o prished.o prished.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT
q931.o -MF .q931.o.d -MP -c -o q931.o q931.c
```

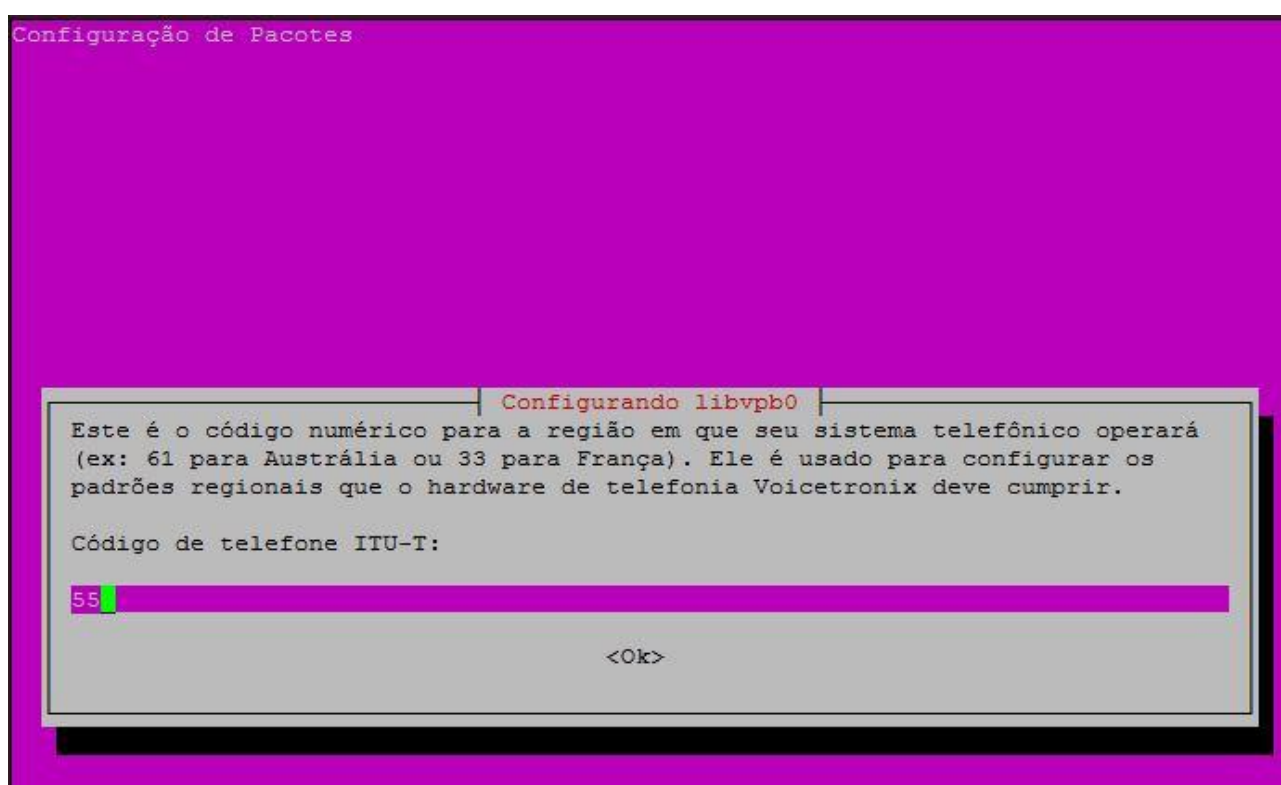
Fonte: Elaborado pelo autor

Agora é feito o download do arquivo de instalação do Asterisk, usa-se o seguinte comando:

wget <http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-13-current.tar.gz>

Depois de realizado o download do Asterisk, descompacta-se o arquivo com o comando `tar zxvf 'nome do arquivo'`, conforme feito nos arquivos anteriores. Depois que terminada a execução da tarefa, acessa-se a pasta gerada para realizar a devida instalação do Asterisk digitando o comando `cd /'nome da pasta'`. Estando dentro da pasta do Asterisk, será necessário digitar o comando `./contrib/scripts/install_prereq install && ./bootstrap.sh` antes da instalação para configurar os arquivos do Asterisk e outros arquivos extras que complementam a instalação. Durante a execução deste comando, a instalação irá pedir que selecione o código do país a ser usado, e como estamos no Brasil, vamos usar o código 55 conforme ilustra a figura 8.

Figura 8. Definindo código do país Brasil



Fonte: Elaborado pelo autor

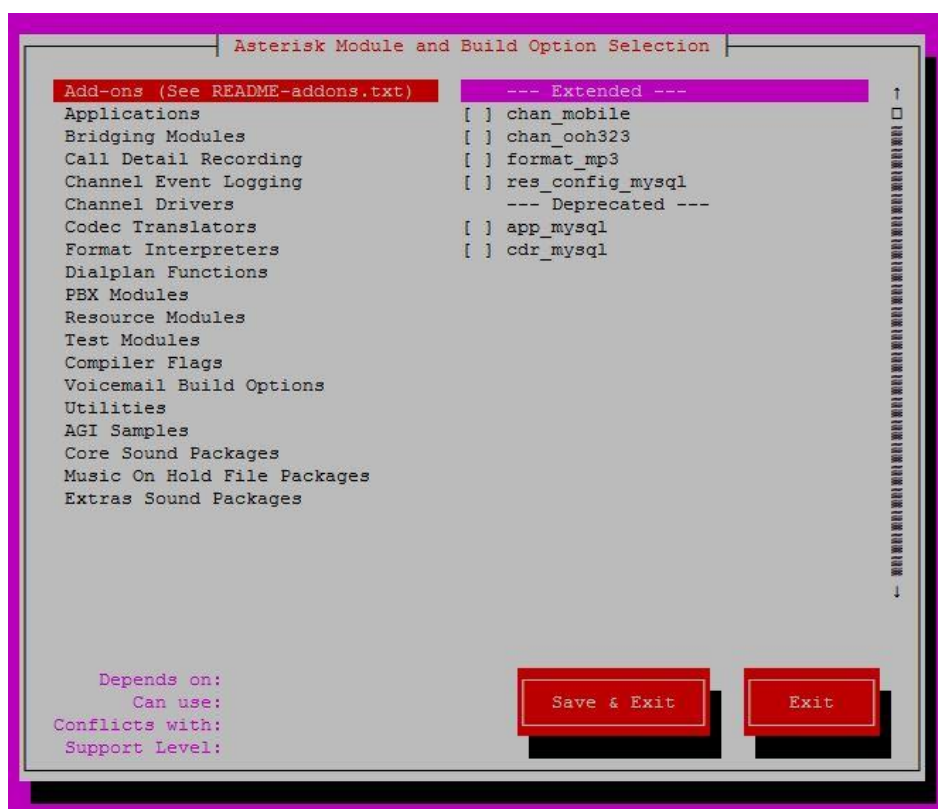
A instalação irá terminar com a mensagem 'install completed successfully', podendo assim prosseguir com o próximo comando. Em seguida, mais

um comando deverá ser executado assim finalizando a instalação do Asterisk. Segue o último comando:

./configure && make menuselect && make && make install && make config && make samples

Durante a instalação aparecerá uma tela para adicionar Add-ons, caso não queira adicionar, seleciona-se o botão Save & Exit e continua a instalação. A instalação sendo feita não será necessária adicionar nenhuma outra opção então, apenas continuar a instalação normalmente. A figura 9 ilustra como é a tela dos Add-ons.

Figura 9. Opções de Add-Ons do Asterisk



Fonte: Elaborado pelo autor

Após selecionar a opção Save & Exit, a instalação irá continuar normalmente, pode demorar um pouco dependendo do poder de processamento do servidor. Quando terminar a execução do comando de instalação do Asterisk, é

necessário iniciar o serviço do driver Dahdi. Uas-se o comando ‘/etc/init.d/dahdi start’ para que o serviço do Asterisk possa subir sem erros. Após digitado o comando, ele retornará na tela os módulos Dahdi iniciados, confirmando assim o sucesso de sua execução. A figura 10 ilustra exatamente o retorno da inicialização do driver Dahdi.

Figura 10. Iniciando o serviço do driver Dahdi.

```
root@asterisk:/usr/src/asterisk-13.17.0# /etc/init.d/dahdi start
Loading DAHDI hardware modules:
  wct4xxp: done   wcte43x: done   wcte12xp: done   wcte13xp: done   wct1xxp: done   w
ctell1xp: done   wctdm24xxp: done   wcaxx: done   wcfxo: done   wctdm: done   wcb4xxp:
done   wctc4xxp: done   xpp_usb: done
Running dahdi_cfg: done.
```

Fonte: Elaborado pelo autor

Com a inicialização do driver Dahdi, agora é possível iniciar o Asterisk. Com o comando ‘/etc/init.d/asterisk start’ o Linux retornará uma mensagem de OK, conforme ilustra a figura 11.

Figura 11. Iniciando o serviço do Asterisk.

```
root@asterisk:/usr/src/asterisk-13.17.0# /etc/init.d/asterisk start
* Starting Asterisk PBX: asterisk [ OK ]
root@asterisk:/usr/src/asterisk-13.17.0#
```

Fonte: Elaborado pelo autor

Agora o Asterisk está instalado e com o serviço no ar de acordo com o que precisamos para realizar o restante das configurações.

5.3. INSTALAÇÃO DO MYSQL

Segundo Neves e Ruas (2005), um SGBD (Sistema de Gestão de Base de Dados) é um software que faz o gerenciamento de todo acesso sobre uma base de dados, com o intuito de garantir a sua integridade, a segurança e restauração. O Mysql é um SGBD que será usado para armazenar as informações das ligações realizadas no PABXIP. Nesta seção será mostrado como foi feita a instalação do

Mysql, como a criação das tabelas e as devidas configurações para que o Asterik possa enviar os dados de CDR para o banco de dados.

De início o comando ***'sudo apt-get install mysql-server libmyodbc – y'*** , será executado sendo a instalação de um arquivo de dependência e o próprio Mysql Server. Durante a instalação do Mysql Server será exibida uma tela pedindo uma senha para o utilizador 'root' do Mysql. Deverá ser inserida esta senha duas vezes e selecionado o botão 'OK', para continuar. Assim que terminada a instalação, cria-se um arquivo sql para que seja executada um comando afim de criar a tabela em que o Asterisk irá inserir as informações. Primeiramente, cria-se o arquivo na pasta ***'tmp'*** do Linux, pois quando o sistema operacional reiniciar o arquivo será apagado e não deixará lixo no sistema. Digita-se o comando ***'cd /tmp'*** , assim estará dentro da pasta mencionada, então abra um programa editor de texto de sua preferência, nesse caso usa-se o Nano, conforme mostra-se com detalhes abaixo:

nano cdr.sql

Após abrir o editor de texto, insere-se as seguintes informações de criação de tabela com os atributos do Quadro 1 a seguir:

Quadro 1. SQL para criação da tabela CDR

```

CREATE TABLE cdr (
  calldate datetime NOT NULL default '0000-00-00 00:00:00',
  clid varchar(80) NOT NULL default "",
  src varchar(80) NOT NULL default "",
  dst varchar(80) NOT NULL default "",
  dcontext varchar(80) NOT NULL default "",
  channel varchar(80) NOT NULL default "",
  dstchannel varchar(80) NOT NULL default "",
  lastapp varchar(80) NOT NULL default "",
  lastdata varchar(80) NOT NULL default "",
  duration int(11) NOT NULL default '0',
  billsec int(11) NOT NULL default '0',
  disposition varchar(45) NOT NULL default "",
  amaflags int(11) NOT NULL default '0',
  accountcode varchar(20) NOT NULL default "",
  uniqueid varchar(32) NOT NULL default "",
  userfield varchar(255) NOT NULL default "",
  did varchar(50) NOT NULL default "",
  recordingfile varchar(255) NOT NULL default "",
  KEY `calldate` (`calldate`),
  KEY `dst` (`dst`),
  KEY `accountcode` (`accountcode`),
  KEY `uniqueid` (`uniqueid`)
);

```

Fonte: Elaborado pelo autor

Assim que estiverem inseridas as informações no arquivo digite CTRL + O para salvar a inserção e CTRL + X para sair da edição de texto ou os comandos referente ao editor que estiver usando. Após esse momento vamos acessar o Mysql para criar o banco de dados. Para acessar o SGBD e criar o banco de dados onde irá a tabela que ficarão as informações seguem a sequência de comandos e suas respectivas funções:

mysql -u root -p

Após o comando acima digitar a senha do usuário root. Estando dentro do prompt do Mysql vamos criar o banco de dados chamado 'asterisk', usando os seguintes comandos:

CREATE DATABASE asterisk;

Pronto, o banco de dados foi criado com sucesso o Mysql retornará uma mensagem como esta: ***'Query OK, 1 row affected (0.00 sec).'***

Com o banco de dados criado vamos acessá-lo para a execução da query que vai criar a tabela que por ventura está armazenada na pasta ***'/tmp'***. Digite o comando ***'USE asterisk;'*** para acessar o banco de dados 'asterisk', o Mysql retornará ***'Database changed'***, agora que estamos no banco 'asterisk' vamos chamar o arquivo sql criado anteriormente para criarmos a tabela com o comando ***'SOURCE /tmp/cdr.sql;'*** o Mysql vai retornar a mensagem ***'Query OK, 0 rows affected (0.02 sec).'***, confirmando assim a efetividade da execução.

5.4. CONFIGURANDO O ASTERISK PARA CONEXÃO COM MYSQL

Para que o Asterisk possa se conectar ao banco vamos usar o recurso ODBC (*acrônimo para Open Database Connectivity*), segundo Microsoft (2016), é uma API para configuração de aplicativos de modo que possam acessar bancos de dados específicos, ou seja, o ODBC permite que um único aplicativo possa acessar tipos diferentes de banco de dados. Algumas configurações será necessário fazer nos arquivos de drivers do ODBC. Abaixo está descrito os caminhos dos arquivos e o conteúdo de cada um:

O arquivo ***odbcinst.ini*** que está no caminho ***etc/odbcinst.ini*** n deverá ser modificado com as seguintes informações do Quadro 2:

Quadro 2. Alterações no arquivo odbcinst.ini

```
[Default]
Driver = /usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so
[MySQL]
Description = MySQL driver
Driver = /usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so
Setup = /path/to/libodbcmyS.so
```

Fonte: Elaborado pelo autor

Na linha que contem Driver=, será necessário inserir o caminho do arquivo *libmyodbc.so*, para isso é necessário digitar o comando `locate libmyodbc.so` para que seja mostrado seu caminho, com isso é só copiar o caminho e substituir as linhas mencionadas, neste caso o arquivo está localizado no seguinte caminho, **'/usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so'**. As informações acima indicam que o banco de dados que será usado é um banco contido no SGBD Mysql. O próximo arquivo que será modificado é o **odbc.ini** que está contido no caminho **/etc/odbc.ini**, o conteúdo que deverá conter nesse arquivo é de acordo com o Quadro 3:

Quadro 3. Alterações no arquivo odbc.ini

```
[MySQL-asterisk]
Driver = MySQL
Description = MySQL Connector for Asterisk
Server = localhost
Port = 3306
Database = asterisk
username = nomeusuario
password = senha
Option = 3
Socket = /var/run/mysqld/mysqld.sock
```

Fonte: Elaborado pelo autor

Neste arquivo será indicado o driver configurado no arquivo anterior e também o nome do banco de dados onde será feita a conexão, a porta, o caminho onde o servidor se encontra, nome do usuário de conexão com o banco e a senha na qual o usuário se conecta, nesse caso deverá ser substituídas pelo usuário e senha que estiver usando no momento da instalação do Mysql. Mais um arquivo que será

modificado é o arquivo **res_odbc.conf** que está no caminho **/etc/asterisk/res_odbc.conf**, o conteúdo dele está descrito abaixo:

Quadro 4. Alterações no arquivo res_odbc.conf.

```
[asterisk]  
enabled=yes  
dsn=MySQL-asterisk  
username=usuario  
password=senha  
pooling=no  
limit=1  
pre-connect=yes  
share_connections=yes  
sanitysql=select 1  
isolation=repeatable_read
```

Fonte: Elaborado pelo autor

Esse arquivo é o arquivo do Asterisk que busca a conexão no ODBC e também deve ser configurado com os dados do banco, o *dsn* busca o alias criado no arquivo *odbc.ini* onde colocamos o nome de [MySQL-asterisk] assim ele consegue localizar as configurações. Outro arquivo de configuração do Asterisk também deverá ser editado é o *cdr_odbc.conf* que se localiza no caminho **/etc/asterisk/cdr_odbc.conf** seu conteúdo está logo abaixo:

Quadro 5. Alterações no arquivo cdr_odbc.conf

```
[global]  
dsn=asterisk  
loguniqueid=yes  
table=cdr  
dispositionstring=yes  
usegmttime=no  
hrttime=yes
```

Fonte: Elaborado pelo autor

Nele é especificado também o nome do *dsn* e também é adicionado o nome da tabela do banco de dados que nesse caso é a tabela *cdr*. O arquivo

cdr_manager.conf que está no caminho */etc/asterisk/cdr_manager.conf* deverá conter as seguintes informações:

*Quadro 6. Alterações no arquivo *cdr_manager.conf**

<pre>[general] enabled = yes</pre>

Fonte: Elaborado pelo autor

Essa configuração deixa habilitado o módulo *cdr_manager.conf*. Um outro arquivo é o *cdr_adptivve_odbc.conf* que está no caminho *etc/asterisk/cdr_adaptive_odbc.conf* onde vamos inserir os seguintes dados:

*Quadro 7. Alterações no arquivo *cdr_adaptive_odbc.conf**

<pre>[asteriskcdr] connection=asterisk table=cdr alias start=calldate</pre>

Fonte: Elaborado pelo autor

E por último vamos acessar o arquivo */etc/asterisk/modules.conf* e retirar o sinal de ponto e vírgual, para que ative as opções do ODBC. Após fazer esse procedimento é necessário reiniciar o serviço do Asterisk com o comando ***sudo service asterisk restart***, conforme mostra a figura 10. Para verificar se o Mysql está configurado corretamente vamos digitar o comando ***odbcinst -q -d***, o resultado que o sistema deverá retornar é esse: ***[MySQL]***, indicando que está tudo certo com o Mysql, outro comando também que devemos executar para verificar se o Asterisk está conectado corretamente com o CDR no ODBC é dentro da CLI do Asterisk, segue abaixo os passos e os retornos dos comandos:

Primeiro devemos acessar a CLI do Asterisk:

asterisk -vvvr

Proximo passo é o comando:

odbc show all

O retorno é o mostrado na figura 11:

Figura 11. Verificando se o ODBC está conectado corretamente no Mysql.

```
asterisk*CLI> odbc show all

ODBC DSN Settings
-----

Name:    asterisk
DSN:     MySQL-asterisk
Last connection attempt: 1969-12-31 21:00:00
Number of active connections: 1 (out of 1)
```

Fonte: Elaborado pelo autor

O último passo é o comando:

cdr show status

No qual o retorno deverá ser o mostrado na figura 12, observando que contém as opções ODBC e Adaptive ODBC carregas, indicando a correta configuração:

Figura 12. Verificando se o cdr está configurado corretamente.

```
asterisk*CLI> cdr show status

Call Detail Record (CDR) settings
-----

Logging:           Enabled
Mode:              Simple
Log unanswered calls: No
Log congestion:    No

* Registered Backends
-----

  cdr-custom
  ODBC
  cdr_manager (suspended)
  radius
  csv
  Adaptive ODBC
  res_config_sqlite
```

Fonte: Elaborado pelo autor

Enfim as configurações de conexão com o banco de dados do Asterisk estão prontas para uso. Neste momento tudo o que for realizado de ligações e chamadas serão registradas na tabela cdr do banco de dados.

5.5. CONFIGURAÇÃO DE RAMAIS

Nesta seção será feita a configuração dos ramais para que dispositivos diferentes na mesma rede possam realizar ligações entre si, ou seja, ramal com ramal. Para isso primeiro será configurado o *dialplan* do Asterisk, essas configurações serão realizadas no arquivo *extensions.conf* dentro da pasta do Asterisk. Segundo Gonçalves (2005), o plano de discagem (*dialplan*) é a peça principal do Asterisk, é onde as configurações que são realizadas no arquivo *extensions.conf*, são responsáveis por todas as chamadas de entrada e saída que são estabelecidas no PABXIP. Para isso será criado um backup do arquivo *extensions.conf*, usando o comando `mv /etc/asterisk/extensions.conf /etc/asterisk/extensions.conf.bk`, para podermos gerar um novo arquivo em branco. Nesse novo arquivo será inserida as configurações de *dialplan* para que o PABXIP saiba para onde direcionar as ligações. Segue no Quadro 8 as configurações que foram usadas nesse servidor.

Quadro 8. Configurações do arquivo *extensions.conf*

```
[globals] ;define variáveis globais
TELEFONISTA=SIP/1000

[general] ;Opções gerais do Dialplan
writeprotect=no ;Mono de leitura
static=yes ;Modo estático
[ramais]
exten => _1XXX,1,Set(MONITOR_FILENAME=${UNIQUEID})
exten => _1XXX,n,Mixmonitor(${MONITOR_FILENAME}.wav)
exten => _1XXX,n,Dial(SIP/${EXTEN},10,tTkK)
exten => _1XXX,n,HangUp()
```

Fonte: Elaborado pelo autor

Na macro [ramais] a primeira linha significa que qualquer número digitado que começar com 1 e for de quatro dígitos o dialplan vai adicionar na variável *MONITOR_FILENAME* o *UIQUEID* da ligação, para cada ligação o CDR do Asterisk gera um número ID para identificar aquela chamada como única, sendo assim

possível localizar essa chamada caso preciso. Na segunda linha o Dialplan chama o aplicativo Mixmonitor que vem por padrão no Asterisk e gera uma gravação do que for conversado nesta chamada, lembrando que se caso não houver comunicação entre os ramais o Mixmonitor não irá gerar nenhum arquivo, o detalhe é que a variável `MONITOR_FILENAME`, será inserida no nome do arquivo que no caso contém o `UNIQUEID` da ligação, sendo assim o arquivo terá como nome o `UNIQUEID.wav`, que será um número aleatório, mais adiante vamos entender melhor o por que foi configurado para gravar o áudio com o ID da ligação.

O próximo passo é configurar o `sip.conf`, é o arquivo onde são registrados os números de ramais do Asterisk. Segundo Gonçalves (2005), SIP é um protocolo de sinalização que é usado para estabelecer as conexões de chamada, ele determina alguns fatores necessários para que aja comunicação, como, ponto de destino, sinalização de telefonia, identificadores de chamada e outros. No arquivo `sip.conf` que se encontra no caminho `/etc/asterisk/sip.conf` foram inseridas as configurações de três ramais sendo ele o 1000, 1001 e 1002. Conforme mostra as linhas do Quadro 9 a seguir:

Quadro 9. Configuração inserida no arquivo sip.conf

```
[general] ;variáveis globais
udpbindaddr=0.0.0.0:5060 ;indica qual a faixa de ip e porta que serão
aceitas
context=ramais ;nome do contexto
disallow=all ;desabilita todos os ramais
allow=ulaw,alaw,gsm ;habilita somente os codecs mencionados

[1000] ;indica o número do ramal
defaultuser=1000 ;nome do usuário para o ramal
secret=12345 ;senha
type=friend ;significa que pode receber e fazer ligações
host=dynamic
nat=force_rport,comedia ;pode ser usado no celular, pois fica atrás do
firewall e força a saída da ligação.
dtmfmode=rfc2833
```

```
callgroup=1 ;grupo de ligações caso queira puxar ligações  
pickupgroup=1  
language=pt_BR
```

```
[1001]  
defaultuser=1001  
secret=12345  
type=friend  
host=dynamic  
nat=force_rport,comedia  
dtmfmode=rfc2833  
callgroup=1  
pickupgroup=1  
language=pt_BR
```

```
[1002]  
defaultuser=1002  
secret=12345  
type=friend  
host=dynamic  
nat=force_rport,comedia  
dtmfmode=rfc2833  
callgroup=1  
pickupgroup=1  
language=pt_BR
```

Fonte: Elaborado pelo autor

Depois de inseridas as configurações no arquivo sip.conf, salve o arquivo e saia do editor, com isso é necessário acessar o módulo CLI do Asterisk para realizar o reload do serviço SIP, para que sejam carregadas as configurações inseridas, acessando a CLI com o comando **asterisk -vvvr**, depois digitar **sip reload**,

após o retorno do sistema, usar o comando `sip show peers` para que verificar se os ramais foram configurados corretamente, conforme ilustra a figura 13.

Figura 13. Recarregando o arquivo sip.conf para ativar os ramais inseridos.

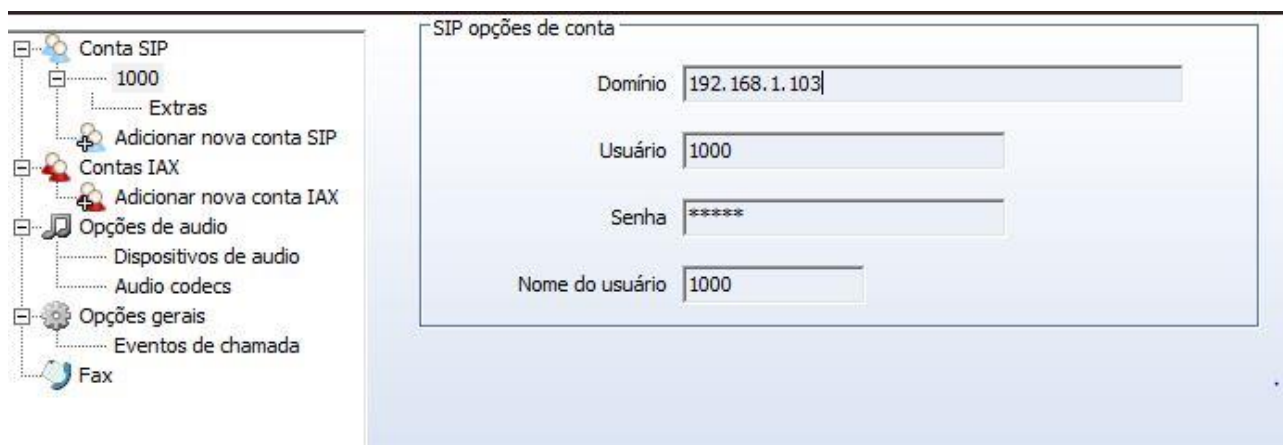
```
Connected to Asterisk 13.17.0 currently running on asterisk (pid = 1187)
asterisk*CLI> sip reload
Reloading SIP
  == Parsing '/etc/asterisk/sip.conf': Found
  == Parsing '/etc/asterisk/users.conf': Found
  == Using SIP CoS mark 4
  == Parsing '/etc/asterisk/sip_notify.conf': Found
asterisk*CLI> sip show peers
Name/username      Host                      Dyn Forcerport
Comedia    ACL Port    Status    Description
1000/1000
Yes          0    Unmonitored
1001/1001
Yes          0    Unmonitored
1002/1002
Yes          0    Unmonitored
3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 0 online, 3 offline]
asterisk*CLI>
```

Fonte: Elaborado pelo autor

Podemos observar que são mostrados os ramais 1000, 1001 e 1002 no retorno do comando, sendo assim, os ramais foram configurados com sucesso, somente ainda não estão configurados nos respectivos terminais.

Agora falta somente a configuração dos terminais, no entanto, usei o aplicativo Zoiper, que é um softphone VOIP que permite enviar mensagens, fazer chamadas de voz e vídeos e pode ser configurado em qualquer provedor Voip ou PBX (ZOIPER, 2017). Para realizar o registro do ramal no Zoiper é necessário abrir as suas configurações e acessar as opções SIP, preencher as informações da conta SIP, na opção Domínio inserir o IP do servidor PABXIP, em Usuário inserir o defaultuser configurado no arquivo SIP que no caso é o 1000, em senha, inserir a senha configurada no campo secret do arquivo SIP e por fim o nome do usuário que aparecerá no display do telefone, para concluir a configuração clicar no botão OK. A figura 14 ilustra com mais detalhes essa configuração.

Figura 14. Configurando um ramal no Zoiper, instalado em um computador.



Fonte: Elaborado pelo autor

Para verificar se realmente o ramal foi registrado no servidor você pode digitar o comando `sip show peers` no CLI do Asterisk ele mostrará no campo Host o IP do dispositivo que foi registrado e no campo Name/username o número do ramal respectivo, o seguinte retorno está ilustrado na figura 15 confirmando o sucesso da configuração.

Figura 15. Verificando se o ramal foi reconhecido pelo Asterisk

```
asterisk*CLI> sip show peers
Name/username      Host                Dyn Forcerport Comedia   ACL Port   Status
Description
1000/1000          192.168.1.102      D Yes           Yes        5060       Unmonit
ored
1001/1001          (Unspecified)      D Yes           Yes         0          Unmonit
ored
1002/1002          (Unspecified)      D Yes           Yes         0          Unmonit
ored
3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 1 online, 2 offline]
```

Fonte: Elaborado pelo autor

Para que possa estabelecer uma ligação é necessário realizar o mesmo procedimento acima em outro ramal para que tenha no mínimo dois dispositivos configurados na rede. Após realizado o procedimento já é possível realizar uma ligação de ramal para ramal, na CLI do Asterisk é possível verificar o que acontece durante uma ligação, a figura 16 ilustra exatamente isso.

Figura 16. Log de uma ligação realizada no Asterisk.

```

== Using SIP RTP CoS mark 5
-- Executing [1000@ramais:1] Set("SIP/1001-00000000", "MONITOR_FILENAME=1500577864.0") in new stack
-- Executing [1000@ramais:2] MixMonitor("SIP/1001-00000000", "1500577864.0.wav") in new stack
-- Executing [1000@ramais:3] Dial("SIP/1001-00000000", "SIP/1000,10,tTkK") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/1000
== Begin MixMonitor Recording SIP/1001-00000000
-- SIP/1000-00000001 is ringing
-- SIP/1000-00000001 answered SIP/1001-00000000
-- Channel SIP/1000-00000001 joined 'simple_bridge' basic-bridge <308c2e48-ca8f-48f2-866c-e31e81a93e8a>
-- Channel SIP/1001-00000000 joined 'simple_bridge' basic-bridge <308c2e48-ca8f-48f2-866c-e31e81a93e8a>
-- Channel SIP/1000-00000001 left 'simple_bridge' basic-bridge <308c2e48-ca8f-48f2-866c-e31e81a93e8a>
-- Channel SIP/1001-00000000 left 'simple_bridge' basic-bridge <308c2e48-ca8f-48f2-866c-e31e81a93e8a>
== Spawn extension (ramais, 1000, 3) exited non-zero on 'SIP/1001-00000000'
== MixMonitor close filestream (mixed)
== End MixMonitor Recording SIP/1001-00000000
asterisk*CLI>

```

Fonte: Elaborado pelo autor

Nas primeiras linhas o Asterisk vai executando o que está configurado no `extensions.conf`, como podemos ver, é inserido o valor do `UNIQUEID` na variável `MONITOR_FILENAME`, depois é ativado o `MixMonitor` e gravado o arquivo `.wav`, por último é executada a última linha do `Dialplan` mostrando que o canal SIP 1001 se conectou com o SIP 1000. As linhas posteriores mostram que o `MixMonitor` está realizando a gravação do áudio da ligação e no final mostra que foi desconectada a ligação e gerado um arquivo encerrando o aplicativo `MixMonitor`. O arquivo de áudio é salvo na pasta `/var/spool/asterisk/monitor/`, caso queira consultar o arquivo, como podemos ver na figura 17 o arquivo foi salvo com uma sequência de números que significam o `UNIQUEID` gerado na ligação que foi realizada, o arquivo ficou com o seguinte nome **1500577864.0.wav**.

Figura 17. Arquivo de áudio gerado pelo Mixmonitor do Asterisk.

```

root@asterisk:/var/spool/asterisk/monitor# ls
1500577864.0.wav

```

Fonte: Elaborado pelo autor

Neste ponto da instalação já temos o Asterisk devidamente instalado, os ramais configurados e prontos para fazer ligações, as ligações são registradas e salvas no banco de dados, o arquivo de áudio gerado na ligação é devidamente salvo pelo aplicativo MixMonitor, conforme precisamos. No entanto as configurações do PABXIP estão prontas, agora é só usar o mesmo, mas nessa implementação vamos criar uma interface gráfica para que facilite a consulta das informações de CDR.

5.6. INSTALAÇÃO E CONFIGURAÇÃO DA INTERFACE GRÁFICA

A interface gráfica criada para monitorar as ligações realizadas no Asterisk foi criada a partir do Framework Laravel. Segundo DevMedia (2017), o Laravel é um Framework PHP muito usado para o desenvolvimento de aplicações, pois ele usa a arquitetura MVC (Model, Viewer, Controller), com características de boa performance e segurança. Toda a documentação de engenharia de software referente a interface gráfica que foi desenvolvida, está no Apêndice, como diagramas de caso de uso, levantamento de requisitos, regras de negócio, matriz de rastreabilidade, diagrama BPMN, Atividade, Estado, Sequencia e prototipação de telas. Antes de iniciar todas as configurações do Laravel, vamos instalar o Apache para que possa subir a aplicação que será gerada. O Apache é um servidor Web, ele é responsável pela publicação de documentos, imagens ou qualquer outro objeto que venha a ser acessado por um cliente através de um navegador (DevMedia, 2017).

A instalação do Apache se dá a partir do comando ***apt-get install apache2***, após a instalação do apache podemos confirmar a sua efetividade digitando o IP do servidor no navegador, se aparecer uma página do Apache2 é porque ocorreu tudo bem com a instalação e o servidor está pronto para receber as publicações. Para que o Laravel possa funcionar é necessário a instalação de algumas dependências, a seguir segue um passo a passo de comandos necessários antes de sua instalação.

Adicionar o repositório do PHP nos repositórios do Linux:

add-apt-repository ppa:ondrej/php

Instalando o pacote de linguagem UTF-8 para que não ocorra erros com caracteres:

```
apt-get install -y language-pack-em-base  
LC_ALL=em_US.UTF-8 add-apt-repository ppa:ondrej/php
```

Agora que os repositórios foram adicionados, vamos baixar os arquivos com o comando:

```
apt-get update
```

Após baixar os pacotes podemos fazer a instalação do PHP 7.0, com o seguinte comando:

```
apt-get install php7.0
```

Depois de instalado o PHP 7.0, vamos instalar as ligações para mysql do php, para que o mesmo possa se comunicar com o banco de dados. O comando é:

```
apt-get install php7.0-mysql
```

Neste momento o PHP está pronto para receber as próximas configurações. Os próximos comandos são referentes a instalação dos pacotes do Laravel onde será a base para o projeto. A instalação será feita conforme está descrito na documentação que consta no site oficial do Laravel. Segundo Laravel (2017), primeiramente é necessário instalar o Composer:

```
composer global require "laravel/installer"
```

É necessário mover o executável do Composer para que o comando possa ser reconhecido pelo sistema, para isso vamos executar o comando **mv composer.phar /usr/local/bin/composer**.

Após o Composer instalado, já é possível criar um projeto em branco na pasta que for especificada (LARAVEL, 2017). Nesse caso o projeto foi criado na pasta

/var/www/html/ através do comando **composer create-project --prefer-dist laravel/'nome do projeto'**.

Após a execução do último comando já temos um projeto em branco pronto para ser editado, mas antes são necessárias algumas configurações nos arquivos no Laravel. No diretório **public** é necessário que seja configurado um virtual host no Apache2 para que quando seja requisitado o IP no navegador, ele direcionar para a pasta **public** (LARAVEL, 2017). Outra configuração necessária é criar uma cópia do arquivo **.env.example** para **.env**, alterar as linhas de conexão com o banco de dados Mysql com o nome do banco, usuário e senha, após isso salvar o arquivo e fechar (LARAVEL, 2017). Segundo a documentação Laravel (2017), o Laravel possui uma chave de string aleatória que serve para criptografar as sessões de usuários, para que seja gerada a chave é preciso digitar o comando **php artisan key:generate**, caso ainda não estiver adicionada no arquivo **.env** que configuramos anteriormente, se ainda não estiver basta digitar o comando, ele mostrará uma chave, copie ela e adicione na linha **APP_KEY=** do arquivo, na qual é a linha 2. A figura 18 mostra melhor como deve ser feita a configuração do arquivo **.env**.

Figura 18. Configuração do arquivo .env do Laravel.

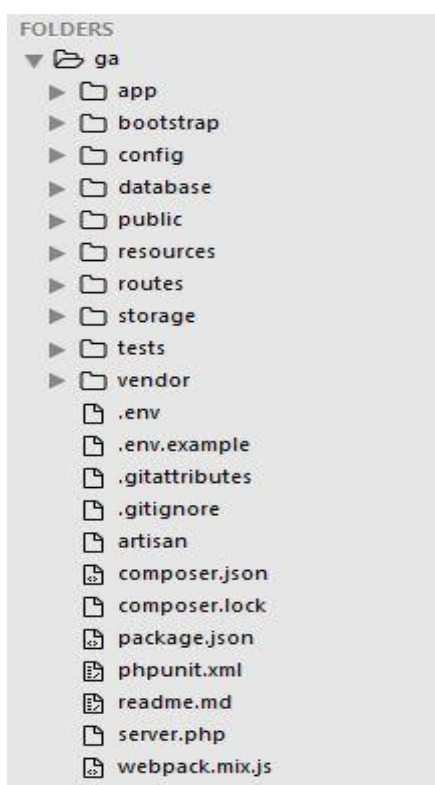
```
APP_ENV=local
APP_KEY=base64:RcrTTpEHTHfQlR4pcpi2tTF2Qhu/B5OzaMGp6+a/YMo=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=asterisk
DB_USERNAME=root
DB_PASSWORD=
```

Fonte: Elaborado pelo autor

Após realizadas todas essas configurações, temos um projeto em Laravel em branco pronto para ser editado. A estrutura de arquivos geradas em um projeto, são as que estão ilustradas na figura 19.

Figura 19. Estrutura de arquivos do Laravel



Fonte: Elaborado pelo autor

Dentro da pasta *app* foi criada uma classe chamada '*Ligacoes*' na qual armazena as informações da tabela *cdr*. Essas informações são buscadas no controller, que fica na pasta *app\Http\Controllers*, o arquivo *HomeController.php* contém uma função na qual faz o select no banco de dados busca as informações e salva na variável *\$registros*, retornando assim os registros na view *home* com uma paginação de 8 registros, no caso será mostrado somente os últimos 8 registros, caso queria ver mais é necessário avançar nos botões em baixo dos registros, conforme é mostrado na figura 20, a linha 25 até a linha 28, o código que foi implementado, explicando o que foi dito anteriormente.

Figura 20. Configuração da Controller do Laravel.

```

25     public function index()
26     {
27         $registros=Ligacoes::select(array('calldate','src','dst','
            billsec','disposition','uniqueid'))->orderBy('calldate',
            'desc')->paginate(8);
28         return view('home')->with("ligacoes",$registros);
29     }
30 }

```

Fonte: Elaborado pelo autor

Na pasta *resources\views* temos o arquivo [home.blade.php](#) nele foram inseridas as variáveis onde buscamos as informações no banco de dados, assim podendo ser mostradas na interface. Foi criado um link simbólico na pasta *public* com o nome de *audio*, no qual dá acesso direto a pasta onde ficam armazenados os áudios gravados das ligações, com os respectivos números *UNIQUEID*, assim conseguimos acessar o ID das ligações pelo banco e conseguimos referenciar o mesmo no código possibilitando acessar os áudios das ligações pela interface, onde o usuário consegue tanto ouvir o áudio ou quanto fazer o download, clicando no botão ao lado.

As linhas 19 até a 43 ilustram o que foi mencionado anteriormente, tendo uma melhor visão do que foi feito na aplicação, segue a figura 21.

Figura 21. Código do arquivo de front-end da interface do sistema


```

19 @foreach($ligacoes as $item)
20 <tr>
21 <td class="text-center"><b>{{date('d/m/Y H:i:s', strtotime($item->calldate))}}</b></td>
22 <td class="text-center"><span class="label label-default">{{ $item->src }}</span></td>
23 <td class="text-center"><span class="label label-default">{{ $item->dst }}</span></td>
24 <td class="text-center">
25 @if($item->disposition == 'NO ANSWER')
26 <span class="label label-danger">NÃO ATENDIDA</span>
27 @else
28 <span class="label label-success">ATENDIDA</span>
29 @endif
30 </td>
31 <td class="text-center">
32 @if($item->billsec > 0)
33 <audio controls>
34 <source src="{{url('/')}}/audio/{{ $item->uniqueid }}.wav" type="audio/ogg">
35 <source src="{{url('/')}}/audio/{{ $item->uniqueid }}.wav" type="audio/mpeg">
36 Your browser does not support the audio element.
37 </audio>
38 @else
39 <span class="label label-danger">SEM ÁUDIO</span>
40 @endif
41 </td>
42 </tr>
43 @endforeach





```

Fonte: Elaborado pelo autor

A aplicação trabalha diretamente com o Asterisk, possibilitando o administrador do PABXIP, analisar as saídas de ligações, quais ramais estão sendo usados, os destinos das ligações, ou até mesmo ouvir as gravações das ligações. Essas informações são muito úteis para empresas nas quais necessitam ouvir as ligações, em caso de empresas relacionadas a vendas, atendimento ao cliente, call centers, etc. A figura 22 mostra exatamente como ficou a interface, que busca as informações no banco de dados mysql e também os áudios gravados das ligações, nela podemos ver as datas e horários das ligações, o número de origem e número de destino, o status da ligação, juntamente com o tempo da ligação e áudio se caso gerado.

Figura 22. Interface do sistema pronta

PABX-IP-UNIFACEF

Data / Hora	Número de Origem	Número de Destino	Status	Tempo / Áudio
26/07/2017 09:16:33	1001	1000	OCUPADO	SEM ÁUDIO
26/07/2017 09:16:18	1001	1000	NÃO ATENDIDA	SEM ÁUDIO
26/07/2017 09:15:57	1000	1001	ATENDIDA	▶ 0:00 / 0:05 
25/07/2017 22:39:43	1001	1000	NÃO ATENDIDA	SEM ÁUDIO
25/07/2017 21:03:43	1001	1000	ATENDIDA	▶ 0:00 / 0:08 
25/07/2017 21:01:48	1001	1000	ATENDIDA	▶ 0:00 / 0:29 
25/07/2017 15:06:10	1000	1001	NÃO ATENDIDA	SEM ÁUDIO
25/07/2017 14:20:54	1001	1000	ATENDIDA	▶ 0:00 / 0:15 

« 1 2 »

Fonte: Elaborado pelo autor

Conforme visto neste capítulo, é possível desenvolver uma solução Open Source, na qual pode resolver vários problemas de pequenas e médias empresas, diminuindo os custos com investimento em telefonia, manutenção de equipamentos, manutenção no sistema, obtendo uma boa usabilidade, assim foi possível criar uma interface para verificar as ligações do CDR, além disso, é possível desenvolver um produto mais completo, criar interfaces para administração de ramais, criação, exclusão, alteração de informações, grupos de capturas e muitas outras opções, também uma interface para configuração de rotas de ligações, no caso se a empresa usar mais de uma operadora de telefonia.

6. CONSIDERAÇÕES FINAIS

Este trabalho fez um estudo sobre as tecnologias Voip, mostrando como a voz navega sobre o protocolo TCPIP e como essa tecnologia foi capaz de causar um grande impacto nos sistemas de telefonia tradicionais. Um estudo também foi feito sobre o software Asterisk, apresentando desde o seu surgimento até como é a sua arquitetura e lógica de funcionamento. Uma implementação de um servidor Asterisk foi realizada a fim de mostrar um exemplo prático de implantação de Voip.

Como relatado ao longo do trabalho, o avanço da tecnologia Voip veio para realizar uma grande mudança na forma como vemos a telefonia. Por exemplo, uma ligação internacional teria um custo muito alto se usássemos uma operadora de telefonia qualquer, com o Voip, é possível realizar ligações internacionais com baixo custo somente usando internet banda larga. Com essas mudanças, vieram grandes investimentos das empresas em licenças de produtos e equipamentos, e o alto custo dessas tecnologias se tornou um impedimento para que empresas, sem condições de fazer investimentos nesta área, expandissem seus negócios.

Foi mostrado que o Asterisk veio para quebrar esse paradigma e que é possível obter um sistema avançado de telefonia com baixo custo de investimento, pelo sistema Asterisk ser um sistema de código aberto e fornecer uma vasta gama de opcionais de configurações de acordo com o perfil de cada empresa. Por exemplo, uma empresa que realiza ligações para várias operadoras de celulares diferentes, iria gastar muito se usasse a saída de telefonia fixa, para esse perfil de empresa é possível instalar um gateway GSM, usando chips de várias operadoras, e o Asterisk faz a portabilidade para a operadora respectiva ao chip instalado, assim a ligação sairia muito mais barata, causando um impacto muito grande na conta telefônica, em termos de redução de valores pagos.

Um outro exemplo seria uma empresa que possui uma linha de 0800, é possível configurar no Asterisk filas de chamadas nas quais são cadastradas agentes, que ficam disponíveis para atender essa fila, e as chamadas em fila ficam em um monitor podendo ser visualizadas em tempo real, típico de perfil de Call Centers. Muitos outros perfis são possíveis de ser configurados no Asterisk, tais configurações vão depender das regras de negócio de cada empresa.

Foi apresentado um exemplo de configuração de um PABXIP que pode realizar ligações entre a rede e também a configuração de uma interface web, em que é possível fazer o monitoramento dessas ligações. Esses são somente recursos básicos do que o Asterisk pode oferecer para uma empresa.

Pode-se considerar como sugestões para trabalhos futuros a implementação de outras interfaces para configurações mais avançadas. Por exemplo, pode ser configurada uma página web para que seja feito o cadastro de novos ramais e o perfil de cada um, como grupos de capturas de chamadas e etc, tornando possível que várias opções do sistema sejam configuradas por interface web. Uma outra configuração é a criação de uma URA inteligente, a qual apresenta uma mensagem em horário comercial e uma outra mensagem após o termino do expediente. Ainda como trabalho futuro, sugere-se efetivamente fazer com que os funcionários de uma empresa façam uso do PABXIP no dia a dia do seu trabalho, e realizar uma avaliação efetiva e minuciosa em relação a opinião dos usuários ao adotar esta alternativa.

Em termos financeiros, o Asterisk é vantajoso para as empresas. Como exemplo uma licença anual para uma empresa que faz até 64 chamadas simultâneas usando o software da linha Enterprise da empresa 3cx custa em torno de R\$ 5.000,00, ou para 512 chamadas simultâneas pode custar até R\$ 25.000,00 (3cx, 2017), lembrando que uma licença particular tem suas limitações, para obter todos os recursos do sistema a empresa tem que adquirir as licenças separadamente ou comprar uma licença completa. Enquanto um servidor Asterisk teria somente o custo de uma máquina servidor, com todos os recursos disponíveis. Caso for contratar um serviço de manutenção do sistema Asterisk a empresa pagará em média R\$ 2400,00 anual dependendo do número de ramais. Desta forma, conclui-se que o Asterisk é um poderoso PABXIP que pode oferecer várias soluções em telefonia para as empresas por um custo muito baixo em comparação com outras tecnologias, e este trabalho mostra um exemplo de como pode este sistema ser utilizado.

REFERÊNCIAS

3CX (2017). PABX IP Comprar. Disponível em <https://www.3cx.com.br/> . Acesso em 10 de agosto de 2017.

ALECRIM (2008). Barramentos: ISA, AGP, PCI, PCI Express, AMR e outros. Disponível em <https://www.infowester.com/barramentos.php> . Acesso em 30 de maio de 2017.

ANDRADE et al. (2010) Comunicação Através da Tecnologia Voip. Disponível em <http://univar.edu.br/revista/downloads/tecnologiavoip.pdf> Acesso em 18 de abril de 2017 as 22:44

BRANDINO (1998). Apostila TCP/IP. Disponível em <http://grsecurity.com.br/apostilas/TCP/training-networking-tcpip.pdf> . Acesso em 31 de maio de 2017.

BRYANT, MADSEN e MEGGELEN (2013), ASTERISK (2017) The Definitive Guide. Disponível em http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html/asterisk-book.html#asterisk-Revolution-SECT-1.1 Acesso em 18 de abril de 2017 as 22:08.

CASTELUCCI, D. (2011). PROTOCOLOS DE COMUNICAÇÃO EM REDES DE COMPUTADORES. Disponível em <https://daniellacastelucci.wordpress.com/2011/04/08/protocolos-de-comunicacao-em-redes-de-computadores/> . Acesso em 15 de maio de 2017 as 19:27.

COLLINS (2004). Carrier Grade Voice Over IP. Disponível em <http://www.voiceip.com.ua/lit/Carrier%20Grade%20Voice%20Over%20IP.pdf> . Acesso em 01 de abril de 2017.

CONÇALVES (2005). Asterisk PBX – Guia de Configuração. Disponível em <https://docs12.minhateca.com.br/511897158,BR,0,0,Livro-Asterisk-Curso-Completo.pdf> . Acesso em 14 de abril de 2017.

DEV MEDIA (2017). Introdução ao Laravel Framework PHP. Disponível em <http://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173> . Acesso em 21 de julho de 2017.

DIXONZAPATA (2009). Historia da Telefonía Zapata e como ela se relaciona com PBX Asterisk. Disponível em <http://www.zapatatelephony.org/>. Acesso em 15 de maio de 2017.

LARAVEL (2017). Installation. Disponível em <https://laravel.com/docs/5.4/installation> . Acesso em 22 de julho de 2017.

NEVES e RUAS (2005). Guia Prático do Mysql. Disponível em <http://www.centroatl.pt/titulos/tecnologias/imagens/excerto-e-book-ca-oguiapraticodomysql.pdf> . Acesso em 18 de julho de 2017.

NORMAND (2017). Vale do Silício – Entenda como funciona a região mais inovadora do planeta. Disponível em <http://www.valedosilicio.com/> . Acesso em 27 de julho de 2017.

OLITEL (2017) Importância do Voip para as empresas. Disponível em <http://www.olitel.com.br/o-que-e-o-voip-e-por-que-ele-e-importante-para-empresas/> .Acesso em 04 de abril de 2017 as 23:16.

OSTERWALDER e PIGNEUR. Business Model Generation. Inovação em Modelos de Negócios. Rio de Janeiro: Alta Books, 2010.

PARZIALE, BRITT, DAVIS, FORRESTER, LIU, MATTHEWS e ROSSELOT (2006). TCP/IP Tutorial and Technical Overview. Disponível em <https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf> . Acesso em 31 de maio de 2017.

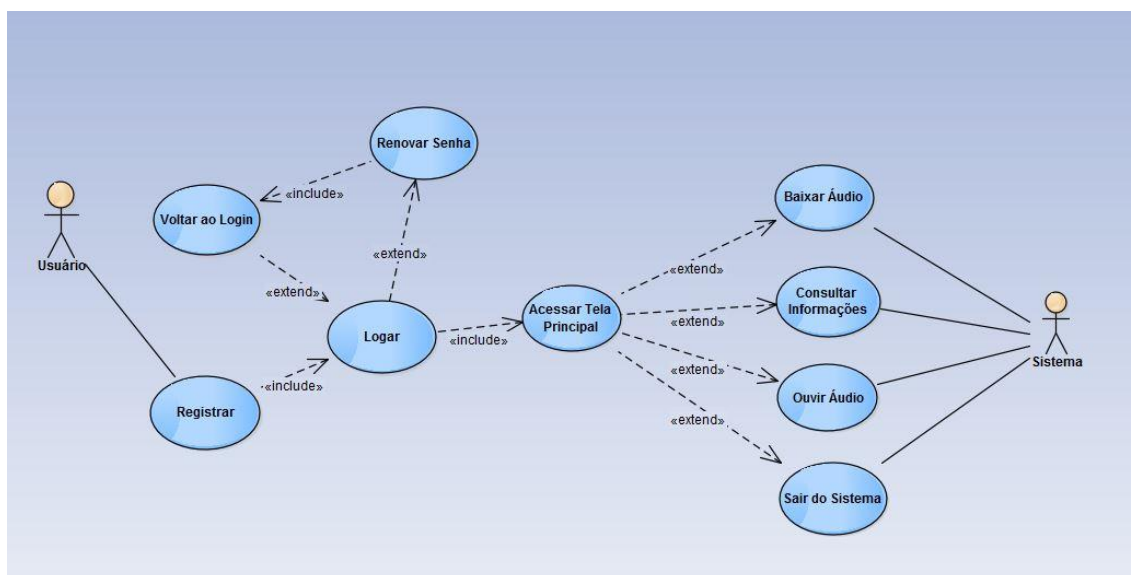
SATO (2004). Teleco Inteligência em Telecomunicações. Disponível em <http://www.teleco.com.br/tutoriais/pdf2010/tutorialpabx.pdf> . Acesso em 30 de maio de 2017.

ZOIPER (2017). Zoiper. Disponível em <https://www.zoiper.com/> . Acesso em 21 de julho de 2017.

APÊNDICES

Documentação de Caso de Uso

Diagrama de Caso de Uso



Índices de Caso de Uso

UC 001 Registrar

Registrar tem como objetivo, realizar o cadastro do usuário para que o mesmo possa ter acesso a tela principal do sistema, de acordo com o RF 002.

UC 002 Logar

Logar tem como objetivo, realizar a entrada de dados do usuário do sistema para que faça a autenticação e em seguida a liberação da tela principal do sistema de acordo com o RF 001.

UC 003 Renovar Senha

Renovar Senha tem como objetivo, fazer o recadastro da senha do usuário caso o mesmo esqueça a senha cadastrada anteriormente, através do envio de e-mail, de acordo com o RF 004.

UC 004 Acessar Tela Principal

Acessar Tela Principal tem o objetivo de consultar as informações geradas pelos usuários do PABXIP, também ouvir o áudio e fazer o download do mesmo caso necessário, de acordo com o RF 003.

UC 005 Baixar Áudio

Baixar áudio tem o objetivo fazer o download do arquivo de áudio gerado durante uma ligação estabelecida, de acordo com o RF 003.

UC 006 Consultar Informações

Consultar Informações tem como objetivo fornecer as informações geradas durante as ligações do PABXIP, de acordo com o RF 003.

UC 007 Ouvir áudio

Ouvir áudio tem o objetivo permitir que o usuário possa ouvir o áudio sem a necessidade de realizar o download do mesmo, de acordo com o RF 003.

Caso de Uso – Registrar	
ID	UC 001
Descrição	Registrar tem como objetivo, realizar o cadastro do usuário para que o mesmo possa ter acesso a tela principal do sistema, de acordo com o RF 002.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário do sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário clica no botão registrar da tela de login do sistema e será direcionado para a tela de registro 2. O mesmo irá realizar o seu cadastro inserindo as informações 3. Após cadastrado devidamente será direcionado para a tela de login para prosseguir.
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 002
Extensão	Não possui

Caso de Uso – Logar	
ID	UC 002
Descrição	Logar tem como objetivo, realizar a entrada de dados do usuário do sistema para que faça a autenticação e em seguida a liberação da tela principal do sistema de acordo com o RF 001.
Ator Primário	Usuário

Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário irá acessar a tela de login do sistema 2. Entrará com os dados de login, e-mail e senha 3. O sistema irá autenticar o usuário e liberar a tela principal
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 001
Extensão	Não possui

Caso de Uso – Renovar Senha	
ID	UC 003
Descrição	Renovar Senha tem como objetivo, fazer o recadastro da senha do usuário caso o mesmo esqueça a senha cadastrada anteriormente, através do envio de e-mail, de acordo com o RF 004.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário ao tentar efetuar o login sem sucesso irá clicar no botão renovar senha. 2. A tela será direcionada para a tela de renovação de senha 3. O usuário irá inserir o e-mail que foi usado no cadastro. 4. No e-mail irá realizar os procedimentos para renovação de senha.
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 004
Extensão	Não possui

Caso de Uso – Acessar Tela principal	
ID	UC 004
Descrição	Acessar Tela Principal tem o objetivo de consultar as informações geradas pelos usuários do PABXIP, também ouvir o áudio e fazer o download do mesmo

	caso necessário, de acordo com o RF 003.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Após o funcionário realizar o seu login do sistema. 2. O mesmo será direcionado para a tela principal 3. Sendo possível visualizar as informações.
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 003
Extensão	Não possui

Caso de Uso – Baixar áudio	
ID	UC 005
Descrição	Baixar áudio tem o objetivo fazer o download do arquivo de áudio gerado durante uma ligação estabelecida, de acordo com o RF 003.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Após o usuário realizar o login no sistema e acessar a tela principal. 2. Ele terá acesso as informações geradas nas ligações inclusive os áudios gravados. 3. O mesmo poderá realizar o download do arquivo gerado.
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 003
Extensão	Não possui

Caso de Uso – Consultar Informações	
ID	UC 006

Descrição	Consultar Informações tem como objetivo fornecer as informações geradas durante as ligações do PABXIP, de acordo com o RF 003.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Após o login do usuário no sistema 2. O mesmo é direcionado para a tela principal do sistema. 3. Na tela principal tem o acesso a todas as informações das ligações podendo analisa-las.
Pós-condição	Não possui
Cenário Alternativo	Não Possui
Inclusão	RF 003
Extensão	Não possui

Caso de Uso – Ouvir Áudio	
ID	UC 007
Descrição	Ouvir áudio tem o objetivo permitir que o usuário possa ouvir o áudio sem a necessidade de realizar o download do mesmo, de acordo com o RF 003.
Ator Primário	Usuário
Pré-condição	Ter perfil usuário no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Após consultar as informações das ligações é possível ouvir o áudio das ligações na própria tela clicando no botão play.
Pós-condição	Pedido realizado
Cenário Alternativo	Não Possui
Inclusão	RF 003
Extensão	Não possui

	RF001	RF002	RF003	RF004
UC001		X		
UC002	<u>X</u>			
UC003				<u>X</u>
UC004			X	
UC005			<u>X</u>	
UC006			<u>X</u>	
UC007			<u>X</u>	

1. Introdução

Documentação do sistema desenvolvido para consultar os relatórios de ligações do Asterisk

1.1 Objetivo do documento

Esse documento tem como objetivo apresentar uma documentação de modelagem para o desenvolvimento de um sistema que faz a consulta dos registros de ligações do Asterisk PABXIP.

O público alvo deste documento são as pessoas que estarão envolvidas no processo de desenvolvimento do software com habilidades em boas práticas para o desenvolvimento apoiado a documentação com diagramas a padrões UML.

1.2 Escopo

O sistema de Consultas de ligações fornece uma visão das funcionalidades do sistema que são:

- ➔ Tela de Login;
- ➔ Cadastro de Usuário;
- ➔ Tela de recuperação de senha;
- ➔ Tela Principal com as informações das ligações;

1.3 Benefícios Esperados

A aplicação irá auxiliar na consulta das informações geradas durante as ligações realizadas pelos usuários do PABXIP. O administrador do PABXIP, poderá acessar as informações com facilidade, usar das informações para tomar decisões, ouvir o áudio das ligações e até fazer o download do áudio para análise.

2. Especificações do Requisitos

RF 001 – Tela de Login	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa		
Descrição: O sistema deverá solicitar login e senha para que o usuário se autentique no sistema e tenha acesso a Tela Principal.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Obrigatoriedade	Permanência
RNF 1.1	O sistema deverá apresentar uma mensagem de alerta caso o usuário e/ou senha estejam incorretos.	Interface	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF 1.2	Se o usuário e senha que o funcionário fornecer estiverem corretos o sistema redirecionará o mesmo para o Menu Principal	Interface	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RF 002 – Cadastro de Usuário	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa		
Descrição: O sistema possibilitará ao usuário cadastrar para acessar o sistema				
Requisitos não funcionais				
Nome	Restrição	Categoria	Obrigatoriedade	Permanência
RNF 2.1	Para cadastrar novos usuários será obrigatório fornecer os seguintes dados: Nome do usuário, E-mail e Senha.	Interface	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF 2.2	O sistema não permitirá o cadastro do cliente caso haja outro usuário com	Segurança	() Desejável (X) Obrigatório	(X) Permanente () Transitório

	mesmo nome e e-mail iguais.			
RNF 2.3	O sistema não permitirá o cadastro de usuário caso não seja inserido as informações mínimas, nome e e-mail.	Segurança	() Desejável (x)Obrigatório	(X) Permanente () Transitório

RF 003 – Tela Principal	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa		
Descrição: O sistema possibilitará ao funcionário visualizar as informações geradas nas ligações, como Data da ligação, número de origem, número de destino, status do atendimento, áudio, tempo de ligação e fazer o download do áudio.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Obrigatoriedade	Permanência
RNF 3.1	O usuário não poderá alterar nenhuma informação.	Interface	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RF 004 – Tela de Renovação de Senha	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa		
Descrição: O sistema possibilitará ao funcionário alterar a sua senha enviando um e-mail para o mesmo, desde que o e-mail já esteja incluso no cadastro.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Obrigatoriedade	Permanência
RNF 3.1	O usuário não poderá alterar a senha se o e-mail inserido não estiver no cadastro.	Interface	() Desejável (X) Obrigatório	(X) Permanente () Transitório

Regras de Negócio

RN 001

O sistema não deverá permitir que o usuário entre no Menu Principal do sistema sem se identificar através de usuário e senha, caso o usuário e senha não estejam corretos o sistema deverá mostrar uma mensagem de alerta.

RN 002

Não poderá ser cadastrado um novo usuário quando já houver um outro usuário cadastrado com o mesmo nome e e-mail para evitar dados duplicados no sistema.

RN 003

O sistema não permitirá que o usuário altere ou exclua algum registro gerado das ligações, inclusive os áudios.

RN 004

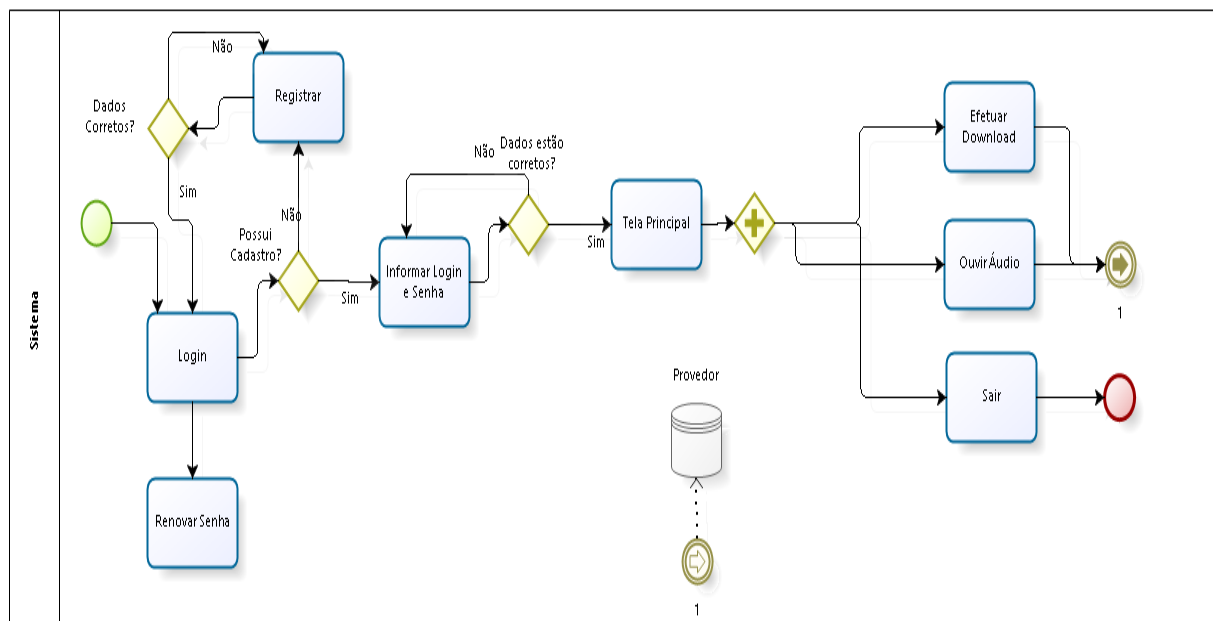
O sistema não permitirá que o usuário sua senha se o e-mail informado não for o mesmo do cadastro.

Matriz de Rastreabilidade

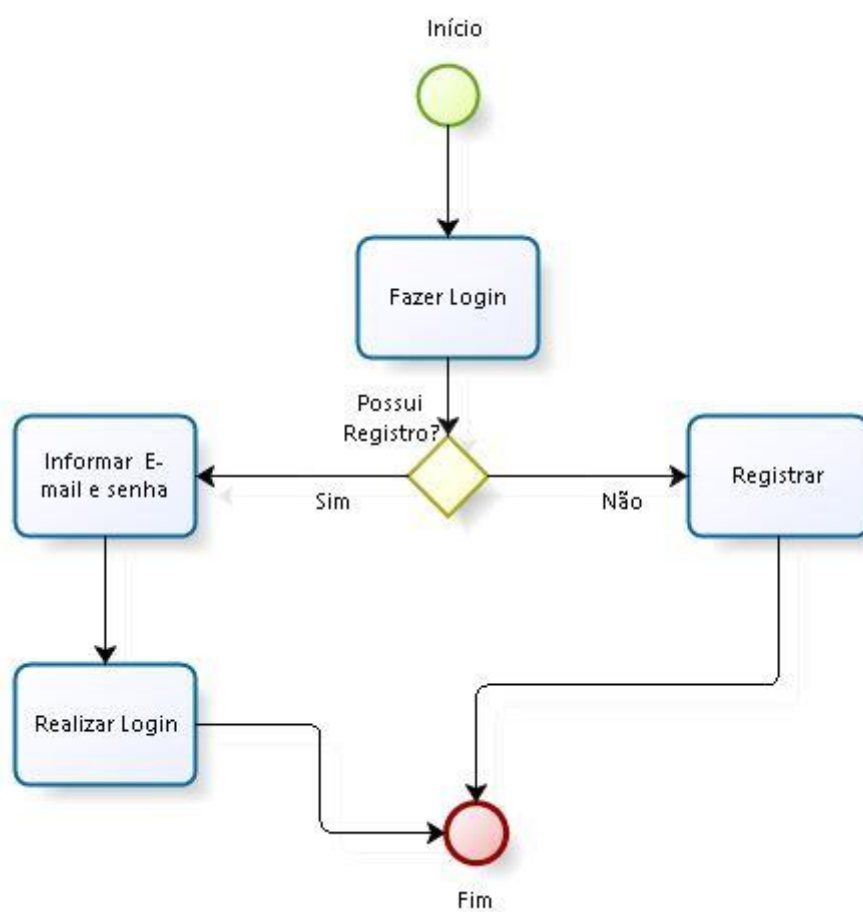
Matriz de Rastreabilidade entre Requisitos e Regras de Negócio

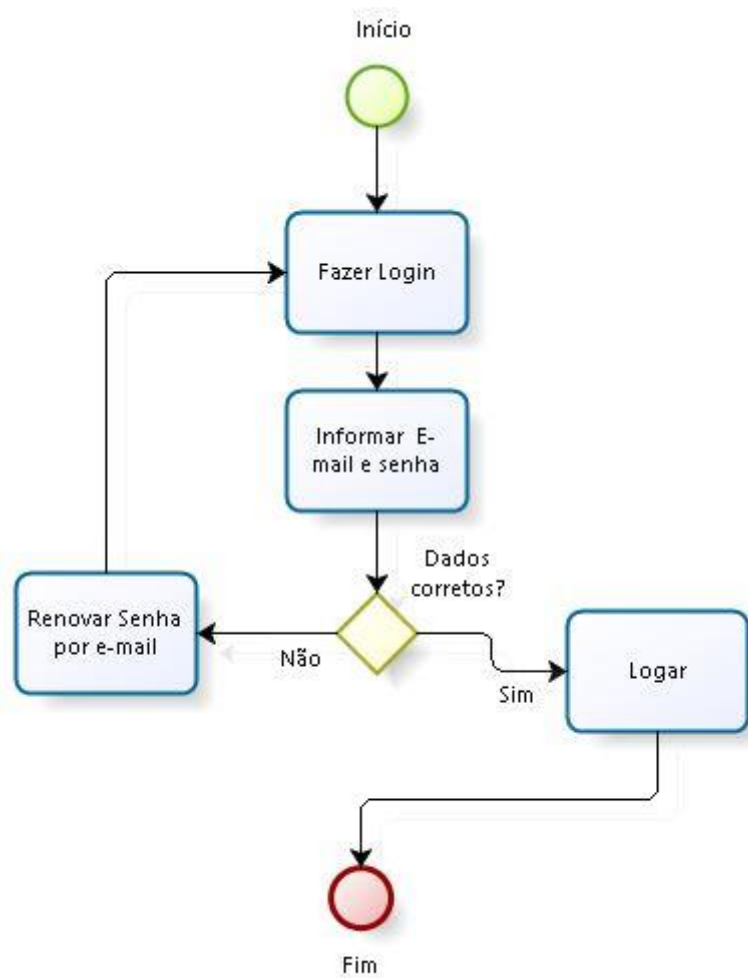
	RF 001	RF 002	RF 003	RF 004	
RN 001	X				
RN 002		X			
RN 003			X		
RN 004				X	

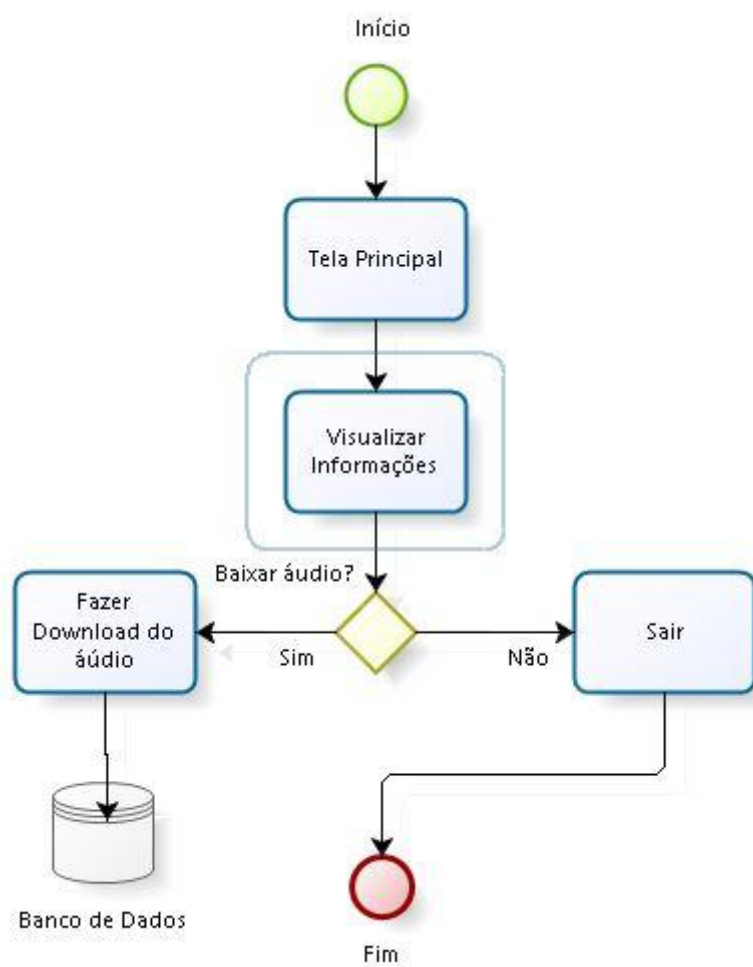
Diagrama BPMN



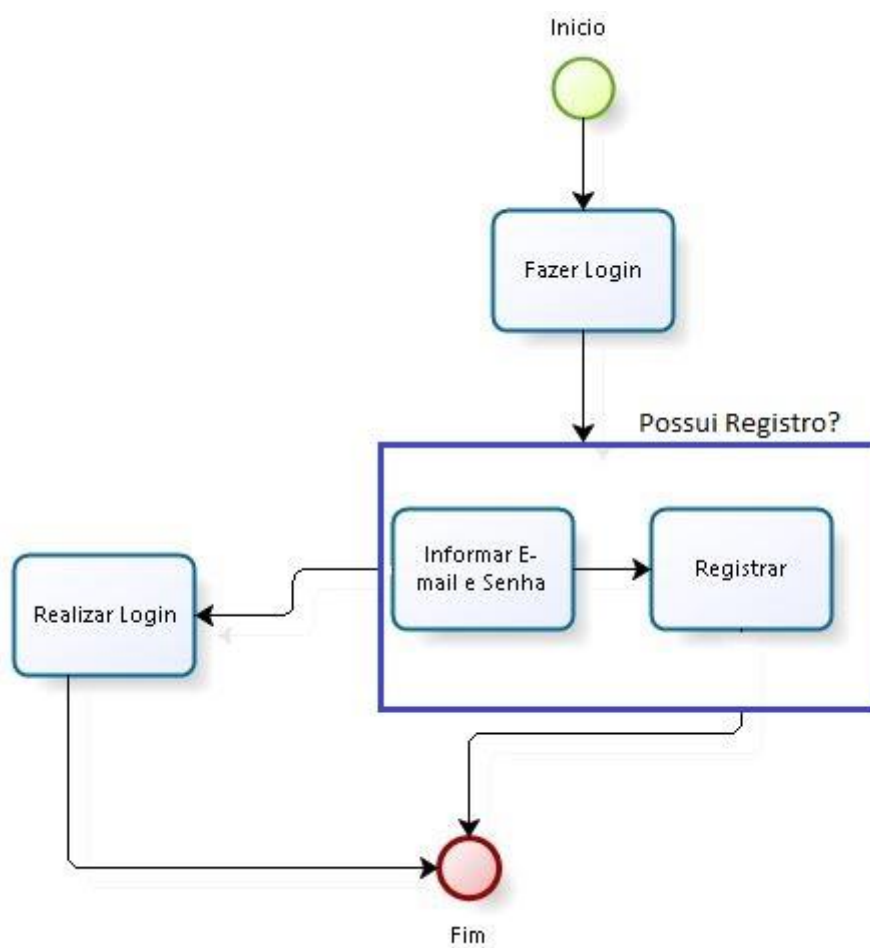
Diagramas de Atividades

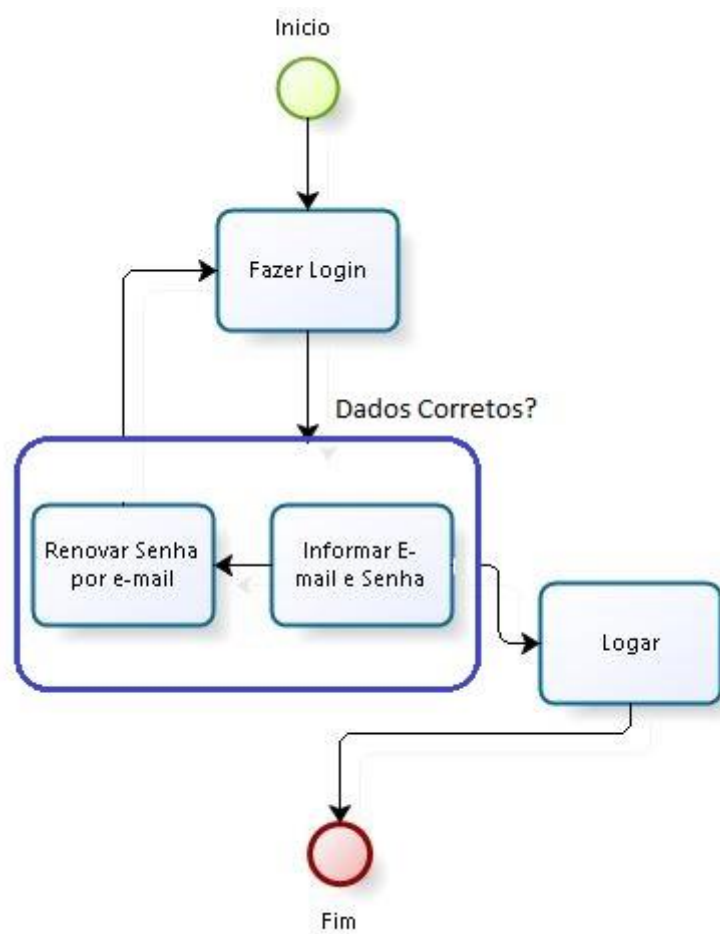


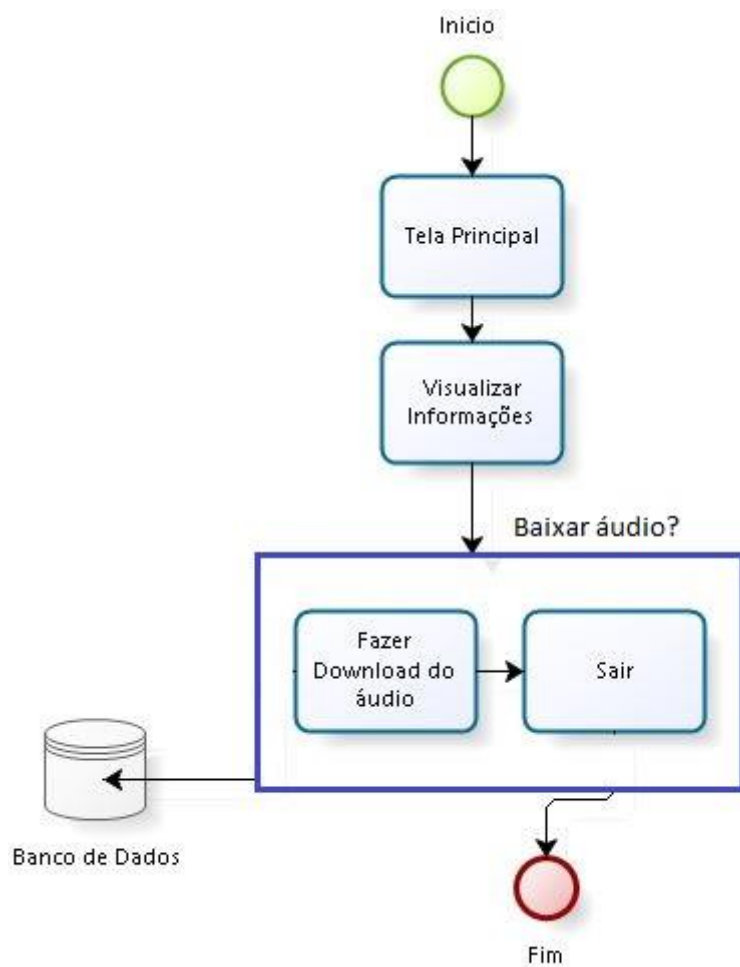




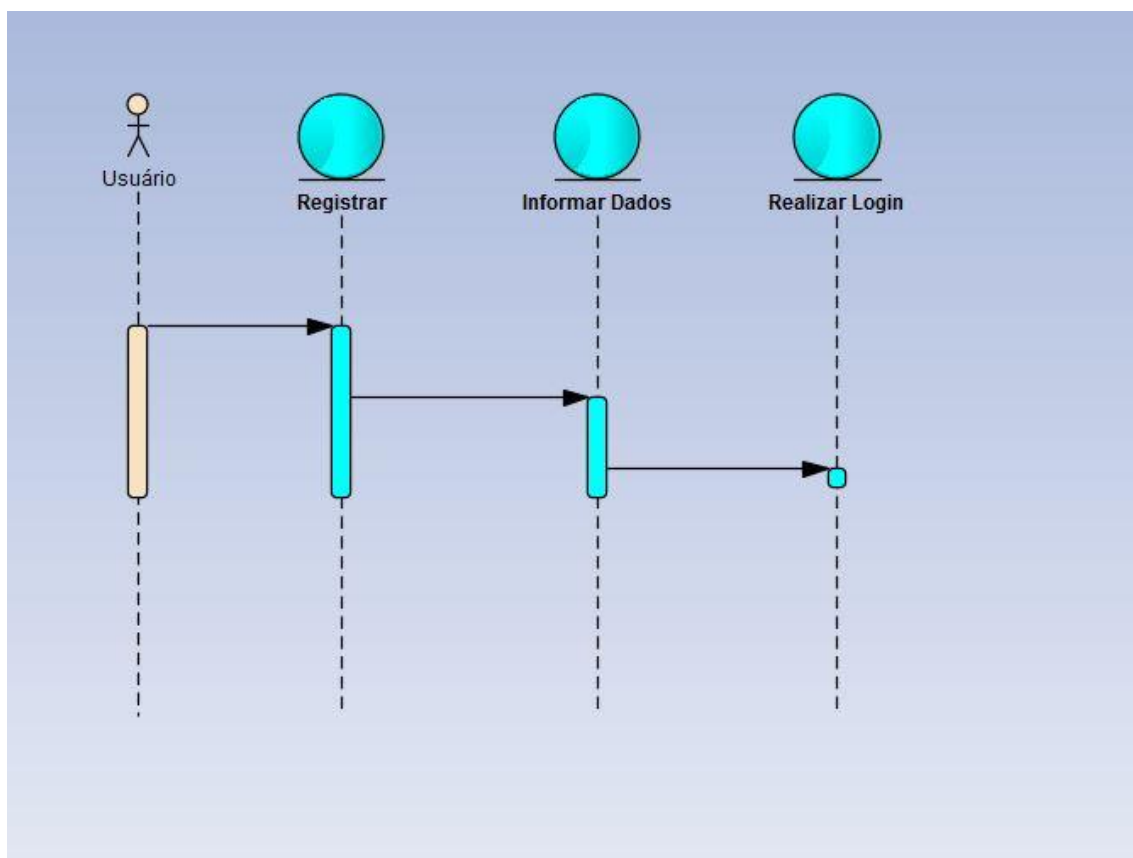
Diagramas de Máquina de Estado

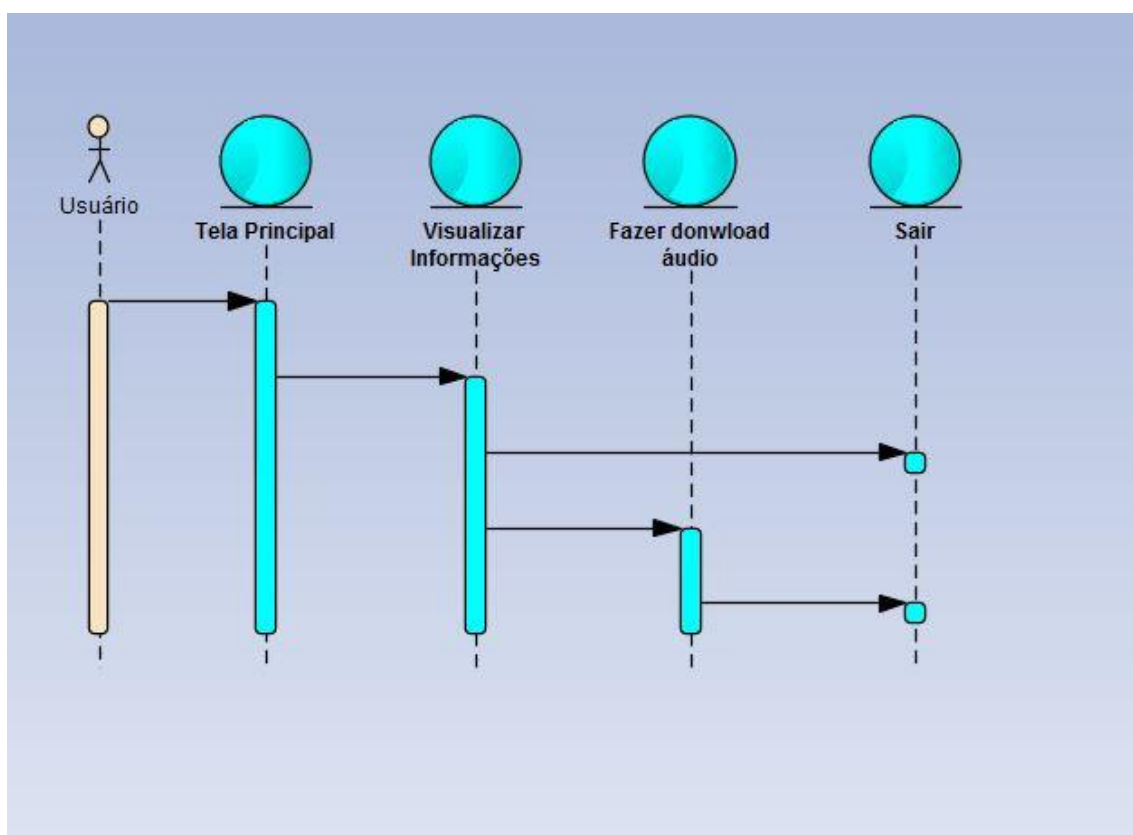
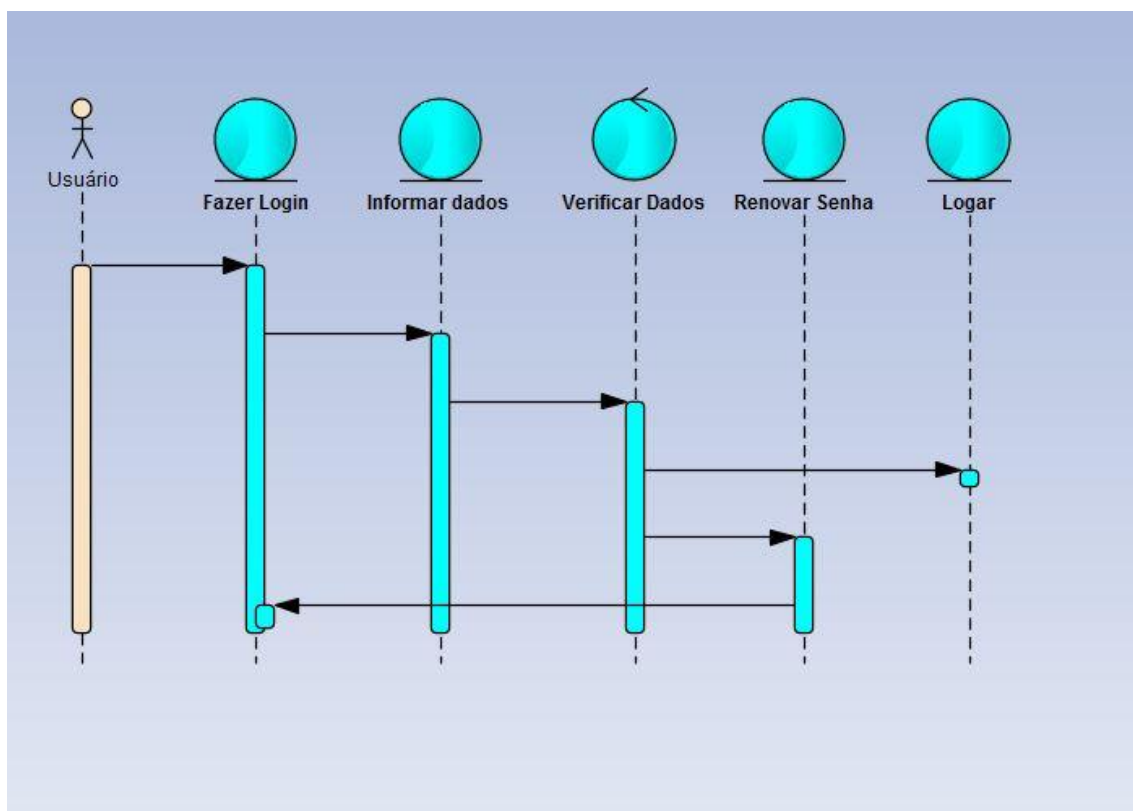






Diagramas de Sequencia





Telas de Prototipação

PABX-IP-UNIFACEF

[Login](#)[Register](#)

Login

E-Mail Address

Password

☐ Remember Me

Login

[Forgot Your Password?](#)

PABX-IP-UNIFACEF

[Login](#)[Register](#)

Register

Name

E-Mail Address

Password




Confirm Password

Register

Reset Password

E-Mail Address

Send Password Reset Link

PABX-IP-UNIFACEF				
Data / Hora	Número de Origem	Número de Destino	Status	Tempo / Áudio
26/07/2017 09:16:33	1001	1000	OCUPADO	SEM ÁUDIO
26/07/2017 09:16:18	1001	1000	NÃO ATENDIDA	SEM ÁUDIO
26/07/2017 09:15:57	1000	1001	ATENDIDA	▶ 0:00 / 0:05 
25/07/2017 22:39:43	1001	1000	NÃO ATENDIDA	SEM ÁUDIO
25/07/2017 21:03:43	1001	1000	ATENDIDA	▶ 0:00 / 0:08 
25/07/2017 21:01:48	1001	1000	ATENDIDA	▶ 0:00 / 0:29 
25/07/2017 15:06:10	1000	1001	NÃO ATENDIDA	SEM ÁUDIO