



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
TECNÓLOGO EM REDES DE COMPUTADORES

PEDRO SÁVIO DE OLIVEIRA NOBRE

**VOIPBOX – UMA SOLUÇÃO DE GERENCIAMENTO VOIP PARA
UNIVERSIDADE FEDERAL DO CEARÁ CAMPUS QUIXADÁ**

**QUIXADÁ
2016**

PEDRO SÁVIO DE OLIVEIRA NOBRE

**VOIPBOX – UMA SOLUÇÃO DE GERENCIAMENTO VOIP PARA
UNIVERSIDADE FEDERAL DO CEARÁ CAMPUS QUIXADÁ**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Orientador Prof. Dr. Márcio Espíndola Maia
Coorientador Prof. Me. Michel Sales Bonfim

**QUIXADÁ
2016**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- N673v Nobre, Pedro Sávio de Oliveira.
VoipBox - Uma solução de gerenciamento VoIP para a Universidade Federal do Ceará campus Quixadá / Pedro Sávio de Oliveira Nobre. – 2016.
53 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2016.
Orientação: Prof. Dr. Márcio Espíndola Maia.
Coorientação: Prof. Me. Michel Sales Bonfim.
1. Telefonia pela Internet. 2. Serviços da Web. 3. Software de aplicação. I. Título.

CDD 004.6

PEDRO SÁVIO DE OLIVEIRA NOBRE

**VOIPBOX – UMA SOLUÇÃO DE GERENCIAMENTO VOIP PARA
UNIVERSIDADE FEDERAL DO CEARÁ CAMPUS QUIXADÁ**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Aprovado em: _____ / julho / 2016.

BANCA EXAMINADORA

Prof. Dr. Márcio Espíndola Maia (Orientador)
Universidade Federal do Ceará-UFC

Prof. Me. Michel Sales Bonfim
Universidade Federal do Ceará-UFC

Prof. Dr. Arthur de Castro Callado
Universidade Federal do Ceará-UFC

Aos meus familiares, a minha companheira,
aos amigos, aos colegas de trabalho, e a todos
aqueles que acreditaram em mim.

AGRADECIMENTOS

Gostaria de iniciar meus agradecimentos citando nominalmente meu orientador e coorientador, Márcio Maia e Michel Sales, respectivamente, por todo o esforço em me guiar neste trabalho de conclusão de curso, dedicando horas de conhecimento e suporte nos momentos que mais precisei. Em seguida, agradeço a Luke Russel por seu apoio moral e revisão do *abstract* deste trabalho. Ricardo Costa e Matheus Medeiros, por todo o suporte técnico necessário para conclusão e sucesso deste trabalho. Agradeço especialmente a Laercio Germano, por dedicar dias de revisão de código, juntamente comigo, suporte no design da aplicação, dicas de aprimoramento e suporte emocional essencial para o sucesso deste trabalho, é com muito prazer que dedico um pouco mais de meus agradecimentos ao mesmo. Dedico totalmente este trabalho aos meus familiares, em especial, minha companheira Julie Martins, por estar comigo nos momentos mais difíceis, contribuindo com apoio e garra nesta longa caminhada. Vânia Valesca, amada mãe, e Maria de Lourdes, amada avó, por me ajudarem financeiramente, provendo todos os recursos necessários para meu sucesso, chega a ser indescritível em palavras o que ambas me proporcionaram. Por fim, agradeço novamente a todos os que foram citados e a todos que não foram citados nominalmente, que proporcionaram suporte para o sucesso deste trabalho e finalizo dedicando-o a todos vocês. Foi uma honra conviver com todos estes gigantes que contribuíram para o final desta etapa de minha vida e que certamente seguirão comigo em todas as outras etapas à frente.

"E aqueles que foram vistos dançando foram julgados insanos
por aqueles que não podiam escutar a música."
(Friedrich Nietzsche)

RESUMO

A fim de facilitar a comunicação através de uma infraestrutura de baixo custo, a tecnologia VoIP (*Voice over Internet Protocol*) substituiu a analógica RPTC (Rede Pública de Telefonia Comutada) através da transmissão de dados de voz digitalmente usando IP (*Internet Protocol*). A Universidade Federal do Ceará – Campus Quixadá utiliza esta tecnologia para comunicação com destinos externos, mas eles estão preocupados com a dificuldade no controle desta tecnologia, tais como, gerenciamento de chamadas e ramais. Este trabalho mostra o desenvolvimento de um sistema de monitoramento e controle para o serviço VoIP, fornecendo funcionalidades específicas e personalizadas para os membros da Coordenação de Tecnologia da Informação (CTI). Este trabalho realizou uma avaliação do sistema, com o objetivo de determinar se esta ferramenta atende às necessidades dos usuários CTI satisfatoriamente. No final da avaliação, a solução apresentou resultados positivos, e permitirá a implantação operacional deste sistema de monitoramento e controle do serviço de voz sobre IP na Universidade Federal do Ceará - Campus Quixadá.

Palavras chave: Telefonia pela Internet (VoIP), AMI, Asterisk, MEAN.

ABSTRACT

In order to facilitate communication through a low-cost infrastructure, VoIP (Voice over Internet Protocol) technology replaces analog PSTN (Public Switched Telephone Network) by transmitting voice data digitally using IP (Internet Protocol). The Federal University of Ceará - Quixadá Campus has enabled this technology for communication with external destinations, but they are concerned with monitoring of the VoIP system utilization, such as difficulty to manage calls and extensions. The work presented in this report shows the development of a monitoring and control system for the VoIP service, by providing specific and customized features to the members of the Coordination of Information Technology (CTI) team. This work carried out a study and evaluation of the system, with the goal of determining if such a tool meets the needs of CTI users satisfactorily. At the end of the evaluation, the solution presented positive results, and will enable operational deployment of this system for monitoring and control of voice over IP service on Federal University of Ceará - Quixadá Campus.

Keywords: Internet Telephony (VoIP), AMI, Asterisk, MEAN.

LISTA DE ILUSTRAÇÕES

Figura 1 - Chamada entre um usuário doméstico e usuários corporativos.....	19
Figura 2 - Processo de transmissão de voz em um sistema VoIP.....	20
Figura 3 - Fluxo de Mensagens SIP.....	21
Figura 4 - Caso de Uso número Um do Asterisk.....	22
Figura 5 - Caso de Uso número Dois do Asterisk.....	23
Figura 6 – Integração e Comunicação da <i>stack</i> MEAN	25
Figura 7 - Funcionamento de um Servidor Node.js.	26
Figura 8 - Análise de desempenho Node.js, requisições por segundo.	27
Figura 9 - Análise de desempenho Node.js, tempo em milissegundos de cada requisição.	27
Figura 10 - Nível de interesse AngularJS ao longo dos últimos anos.	29
Figura 11 – Arquitetura e comunicação da solução VoipBox.....	31
Figura 12 – Configuração do MongoDB e LDAP para comunicação.....	32
Figura 13 – Tela de Login VoipBox.....	33
Figura 14 – Tela de Usuários VoipBox	34
Figura 15 – Tela de criação de novo usuário.....	35
Figura 16 – Tela de Chamadas VoipBox.....	36
Figura 17 – Configuração do Asterisk para comunicação.....	36
Figura 18 – Tela de Ramais VoipBox.	37
Figura 19 - Tela de <i>Dashboard</i> VoipBox.....	39
Figura 20 – Diagrama de caso de uso VoipBox.....	40
Figura 21 – Porcentagem geral da avaliação	44
Figura 22 – Quantidade de respostas por questão	44

LISTA DE TABELAS

Tabela 1 - Comparação das Soluções de Monitoramento VoIP.....	17
Tabela 2 - Análise de desempenho do Node.js: Cálculo do décimo número de Fibonacci.	28
Tabela 3 – Contabilização de respostas por questões.....	43

SUMÁRIO

1 INTRODUÇÃO.....	15
2 TRABALHOS RELACIONADOS	16
3 OBJETIVOS GERAIS E ESPECÍFICOS	18
3.1 Objetivos Gerais	18
3.2 Objetivos Específicos	18
4 FUNDAMENTAÇÃO TEÓRICA	18
4.1 Tecnologia VoIP e seus Conceitos	19
4.1.1 Rede Pública de Telefonia Comutada (RPTC).....	19
4.1.2 Transmissão de voz sobre IP	20
4.1.3 Protocolo SIP, sinalização e controle utilizado na tecnologia VoIP	21
4.2 A Ferramenta Asterisk.....	22
4.2.1 AMI - Asterisk Manager Interface.	23
4.3 Stack MEAN (MongoDB, Express, AngularJS e Node.js).....	24
4.3.1 Node.js e Express	25
4.3.2 MongoDB	28
4.3.3 AngularJS	29
5 PROPOSTA.....	30
5.1 Arquitetura VoipBox	30
5.2 Protótipo e Funcionalidades da Solução	31
5.2.1 Login.....	31
5.2.2 Usuários	33
5.2.3 Chamadas	35
5.2.4 Ramais	37
5.2.5 Dashboard	38
5.2.6 Visão geral das Funcionalidades.	39
6 AVALIAÇÃO DA SOLUÇÃO.....	40
6.1 Instalação do Ambiente de Avaliação	40
6.2 Método de avaliação	42
6.3 Resultados da Avaliação	42
6.4 Considerações sobre os resultados.....	45
6.5 Trabalhos Futuros	45
7 CONCLUSÃO.....	46
REFERÊNCIAS	47
APÊNDICES	49
APÊNDICE A – Questionário de Avaliação.....	49
APÊNDICE B – Roteiro da avaliação.....	52
APÊNDICE C – Resultados do Google Forms	53

1 INTRODUÇÃO

A Internet proporciona diversas formas de comunicação e provê meios para desenvolvedores criarem aplicações que facilitam a utilização de ferramentas complexas. O serviço de voz passou a utilizar a Internet como meio para transmissão de pacotes de voz (TRONCO, 2010) sobre o protocolo IP. Esta abordagem, de acordo com Costa (2007), seguida do alto crescimento de usuários VoIP, é empregada pela promessa de economia no custo das ligações, pois na Internet pagamos por quantidade de dados transmitidos e não por tempo de uso como na telefonia convencional.

Na Universidade Federal do Ceará - Campus Quixadá, após a implementação do serviço VoIP por Aguiar (2013), algumas dificuldades foram encontradas e reportadas pela administração técnica do campus, tais como: dificuldade em controlar, monitorar e gerenciar o serviço de voz por meio de arquivos de configuração e linhas de comando; carência de informações sobre o serviço de voz em nível administrativo (e.g. status de ramais, registro de chamadas, linha de crédito para usuários); necessidade de alerta de problemas em tempo hábil.

Existem soluções de monitoramento VoIP como a NextBilling¹ e BillingVoIP² que poderiam ser utilizadas, porém ambas são pagas e esse custo é inviável para o campus. Uma outra solução chamada QVoIP, desenvolvida no trabalho de Nunes et al. (2008), foi importante para análise de outras ferramentas que trabalham com VoIP, porém esta ferramenta é voltada apenas para análise de pacotes de voz sobre IP, abordando métricas como: latência e perda de pacotes. Este trabalho propõe uma ferramenta de nível administrativo, viabilizando gerência de ramais e de chamadas.

A ferramenta Elastix³ também poderia ser uma solução, pois, além de ser *Open Source*, ela trabalha com o monitoramento em nível administrativo. Elastix é uma ferramenta que incorpora a *stack* LAMP (Linux, Apache, MySQL e PHP) como conjunto de tecnologias utilizadas para desenvolvimento. Entretanto, estas diferenças, como a tecnologia e linguagem utilizadas na interface WEB do Elastix, proporcionam a este projeto a pretensão de desenvolver uma nova aplicação utilizando a *stack* MEAN (MongoDB, Express, AngularJS e Node.js), pois de acordo com o trabalho de Poulter, Johnsto e Cox (2015), apesar da *stack* LAMP possuir um alto nível de maturidade para desenvolvedores, a *stack* MEAN oferece

¹ <http://www.nextbilling.com.br/>

² <http://www.virgos.com.br/portal3/billing-voip/>

³ <http://www.elastix.org/>

maior produtividade e desempenho sobre a antiga *stack*. Outra funcionalidade, como autenticação com LDAP, foi ressaltada neste projeto como um diferencial, pois apesar da ferramenta Elastix viabilizar este recurso, a mesma não o viabiliza de forma nativa. Outra ferramenta que pode ser utilizada para monitoramento VoIP é a FreePBX⁴, porém, também, é desenvolvida com abordagem LAMP e, também, a integração com LDAP é apenas possível por meio de recursos externos e não nativos.

Como base complementar, um artigo da Infoworld (2015) afirma que a *stack* MEAN pode ser a nova abordagem de desenvolvimento de aplicações WEB propensa para o futuro. O artigo comenta sobre o desempenho que a abordagem MEAN possui em relação a abordagem LAMP, onde a *stack* MEAN possui a capacidade de processar mais pedidos por segundo e maior taxa de transferência com menor número de solicitações com falhas.

Este projeto tecnológico pretende desenvolver uma aplicação responsável por gerenciar, controlar e monitorar o serviço VoIP da Universidade Federal do Ceará campus Quixadá, com o objetivo facilitar a administração do serviço de voz para os membros da CTI, prover monitoramento em nível administrativo e utilizar uma nova abordagem de desenvolvimento proporcionada pela *stack* MEAN. A solução deverá possuir compatibilidade de integração com o serviço LDAP já consolidado no campus, utilizado como meio de centralizar a base de dados gerais do campus.

Para a validação do projeto, um questionário será utilizado para medir o nível de satisfação dos usuários da CTI (Coordenadoria de Tecnologia da Informação), bolsistas da CTI e alguns professores da Universidade Federal do Ceará campus Quixadá.

É esperado que este projeto possa auxiliar nas decisões de aprimoramento da infraestrutura de redes do campus no quesito VoIP, ser utilizado pela coordenadoria de tecnologia da informação (CTI) com o objetivo de identificar em tempo hábil problemas no serviço de voz e proporcionar uma melhor gestão dos ramais utilizados no campus.

As próximas seções contemplarão os trabalhos utilizados com maior referência bibliográfica para este projeto, os objetivos gerais e específicos do projeto, referencial teórico, detalhamento da nova proposta, avaliação da nova proposta e conclusão, todos, detalhados respectivamente nas seções 2, 3, 4, 5, 6 e 7.

2 TRABALHOS RELACIONADOS

⁴ <https://www.freepbx.org/>

Alguns trabalhos da literatura já desenvolvidos facilitaram a elaboração deste projeto, como o trabalho de Aguiar (2013), que promoveu infraestrutura de telefonia sobre IP na Universidade Federal do Ceará - Campus Quixadá. O trabalho foi capaz de fornecer dados sobre a arquitetura do campus em relação ao serviço de voz e configurações básicas do serviço encontrados em arquivos como o *sip.conf* e *extensions.conf* providos pelo *software* Asterisk, responsável este, por emular recursos de um PABX convencional, utilizado em tecnologia VoIP. Já este projeto, propõe uma ferramenta capaz de controlar e monitorar esta infraestrutura pré-estabelecida.

O trabalho de Apolinário e Bezerra (2013), tiveram uma grande contribuição ao apresentar a solução Elastix, da qual este projeto propõe uma solução parecida, porém de abordagem diferente, utilizando tecnologias diferentes para o desenvolvimento de uma nova solução. O trabalho de Nunes et al. (2008), também teve sua contribuição ao apresentar a Solução QVoIP desenvolvida para monitoramento do serviço VoIP, porém direcionada apenas para um monitoramento a nível de rede, este trabalho propõe um monitoramento mais administrativo do serviço de voz, envolvendo registro de chamadas, gerenciamento de ramais e etc. Nunes et al. (2008) também contribuiu com referencial teórico para este projeto.

Poulter, Johnsto e Cox (2015), teve contribuição ao confrontar em seu trabalho as *stacks* MEAN e LAMP, possibilitando a decisão de utilizar a *stack* MEAN como abordagem de desenvolvimento deste projeto.

Soluções já desenvolvidas, apesar de todas não estarem contempladas em trabalhos científicos, foram de grande ajuda, oferecendo uma base para o desenvolvimento da solução proposta neste projeto. A seguir é possível entrar em alguns detalhes que diferenciam este projeto das soluções já existentes no mercado e visualizar as principais funcionalidades fornecidas e não fornecidas por cada uma.

Tabela 1 - Comparação das Soluções de Monitoramento VoIP.

<i>Soluções</i>	<i>Stack</i>	<i>Livre</i>	<i>Registro Chamadas</i>	<i>Gerenciar Ramais</i>	<i>Tarifação Nativa</i>	<i>Autenticação</i>	<i>LDAP Nativo</i>	<i>Multi-Usuário</i>
Elastix	LAMP	Livre	Sim	Sim	Não	Sim	Não	Não
NextBilling	Desconhecido	Pago	Sim	Sim	Sim	Sim	Não	Sim
FreePBX	LAMP	Livre	Sim	Sim	Não	Sim	Não	Não
Billing VoIP	Desconhecido	Pago	Sim	Sim	Sim	Sim	Não	Sim

VoipBox	MEAN	Livre	Sim	Sim	Não	Sim	Sim	Sim
----------------	------	-------	-----	-----	-----	-----	-----	-----

Fonte: Elaborada pelo autor.

3 OBJETIVOS GERAIS E ESPECÍFICOS

Esta seção apresenta os objetivos gerais e específicos do projeto.

3.1 Objetivos Gerais

Propor uma solução capaz de prover monitoramento e controle em nível administrativo do serviço de voz (VoIP) da Universidade Federal do Ceará campus Quixadá, que utiliza a ferramenta Asterisk.

3.2 Objetivos Específicos

Podem ser listados abaixo os objetivos específicos propostos neste projeto, que têm como meta alcançar o objetivo geral descrito na seção 3.1.

- Identificar soluções já desenvolvidas, seja em trabalhos científicos ou soluções que atuam no mercado;
- Enumerar as funcionalidades voltadas às necessidades de monitoramento e controle a nível administrativo do serviço de voz (VoIP), dos membros da CTI na Universidade Federal do Ceará campus Quixadá;
- Desenvolver uma solução monitoramento e controle a nível administrativo do serviço de voz (VoIP), utilizando apenas ferramentas de código livre;
- Validar a satisfação dos usuários da CTI e professores da Universidade Federal do Ceará campus Quixadá.

A próxima seção apresenta o referencial teórico utilizado para o desenvolvimento deste projeto.

4 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta todos os conceitos necessários para o entendimento deste projeto.

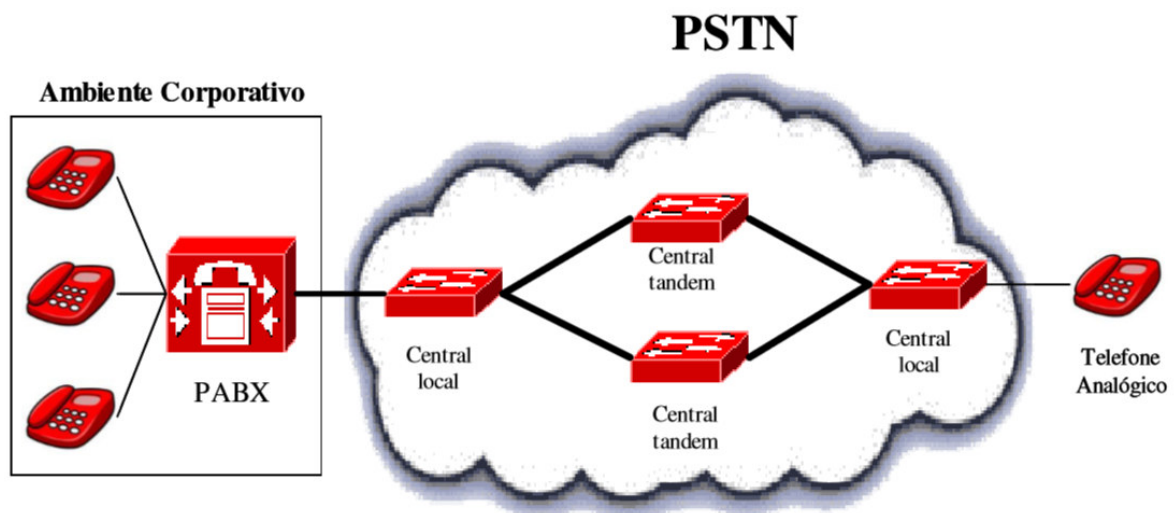
4.1 Tecnologia VoIP e seus Conceitos

Segundo Gokhale e Lu (2005), VoIP é o transporte de voz sobre o protocolo IP, diferentemente da rede pública de telefonia comutada (RPTC ou PSTN em inglês) que é o sistema de telefonia internacional convencional que utiliza fios de cobre para o transporte de dados de voz analógicos. A próxima subseção explica o funcionamento da telefonia convencional com o intuito de melhorar o entendimento da tecnologia VoIP.

4.1.1 Rede Pública de Telefonia Comutada (RPTC)

Para entender melhor a tecnologia VoIP é importante conhecer o funcionamento da RPTC. Este sistema, baseado na técnica de comutação de circuitos, utiliza um caminho fim-à-fim entre dois pontos estabelecendo uma linha dedicada entre as partes, o que evita problemas de congestionamento, problemas estes, comuns quando levados ao contexto que aborda a comutação de pacotes, como por exemplo a Internet (CALLADO et al., 2007). O processo de comunicação entre pares passa por níveis hierárquicos desde o estabelecimento do circuito até a desconexão do mesmo. É possível observar na **Figura 1** como estão distribuídos estes níveis ao longo de centrais telefônicas.

Figura 1 - Chamada entre um usuário doméstico e usuários corporativos.



Fonte: (CALLADO et al., 2007).

Na **Figura 1**, pode-se observar que cada telefone está ligado à uma central local, apesar dos telefones situados no ambiente corporativo estarem ligados a um PABX (*Private Automatic Branch Exchange*), e o mesmo estar ligado a uma central local, pode-se

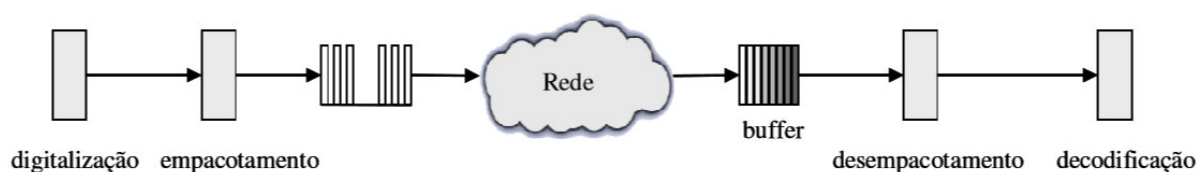
compreender que os três telefones estão ligados à central local mais próxima deles, observada na figura em questão, a diferença é que o telefone ao lado oposto dos demais está ligado diretamente à uma central local e os outros três telefones estão ligados em um PABX, geralmente utilizado em ambientes corporativos com o objetivo de fornecer serviços adicionais, como por exemplo, conferências, transferência de chamadas e ainda proporcionar comunicação interna somente entre usuários desse ambiente privado (CALLADO et al., 2007).

O processo de comunicação de acordo com Callado et al. (2007) é transmitido pelo telefone analógico direto para a central local, onde o sinal analógico é digitalizado e transferido ao núcleo da RPTC, conhecido como central tandem. O sinal, agora digitalizado, segue até a central local do telefone receptor, onde novamente o sinal digital é convertido para analógico e entregue ao destino desejado. A comunicação se repete neste mesmo procedimento em ambas as direções.

4.1.2 Transmissão de voz sobre IP

Como foi possível observar na subseção anterior, o funcionamento da telefonia convencional é análogo à telefonia sobre IP. Pode-se observar que centrais telefônicas podem ser substituídas por roteadores, onde centrais locais são representados por roteadores de borda e centrais tandem representadas por roteadores do núcleo da rede.

Figura 2 - Processo de transmissão de voz em um sistema VoIP.



Fonte: (CALLADO et al., 2007).

O processo de transmissão de voz sobre IP pode ser compreendido pela **Figura 2**. Nesta, é possível observar que o sinal analógico passa por um processo de digitalização que de acordo com Callado et al. (2007), possui três etapas: amostragem do sinal analógico, quantização e codificação, onde na amostragem é retirada uma parte do sinal analógico onde não interfira na perda da informação quando os mesmos são reconstituídos.

Na quantização, os valores da amostragem são convertidos em números finitos e seguidamente, na etapa de codificação, cada valor quantizado recebe um valor binário

(VALE; BRAJTERMAN, 2007). Os dados codificados por algoritmos, chamados *codecs*, são empacotados e utilizam o protocolo IP para encaminhamento dos dados na Internet. Quando os pacotes chegam em seu local de destino, ele é desempacotado e os dados codificados passam por um processo de decodificação com o objetivo de transformar a informação digitalizada, novamente, em sinal analógico.

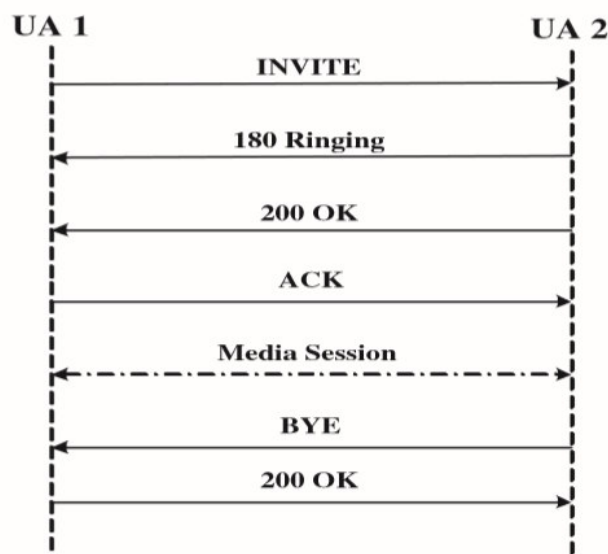
4.1.3 Protocolo SIP, sinalização e controle utilizado na tecnologia VoIP

O SIP é um protocolo da camada de aplicação, desenvolvido em texto como os protocolos HTTP e SMTP, ele tem o objetivo de controlar o estabelecimento de ligações telefônicas e diversas outras aplicações multimídias (CALLADO et al., 2007).

De acordo com Gokhale e Lu (2005), o protocolo SIP é um protocolo P2P (peer-to-peer), onde a comunicação entre os pontos, representados na **Figura 3** por UA1 (*User Agent*) e UA2, é iniciada por uma mensagem chamada INVITE, responsável por inicializar uma chamada. Em seguida o UA2 envia uma resposta, “180 Ringing”, informando o status do mesmo o qual seria tocando, seguidamente de uma outra resposta, “200 OK”, sinalizando o novo status em relação a mensagem de abertura de chamada, INVITE.

O UA1, envia uma mensagem ACK, responsável por confirmar a entrega da última resposta, e logo em seguida uma conexão ponto a ponto entre ambos é estabelecida. O processo de encerramento da chamada, é solicitada pelo UA2 e confirmada com uma mensagem, “200 OK”, pelo UA1.

Figura 3 - Fluxo de Mensagens SIP.



Fonte: (GOKHALE; LU, 2005).

O protocolo SIP é uma solução mais flexível, simples e de fácil implementação em comparação com outros protocolos de sinalização como o H.323, onde muitos que o SIP será dominante no quesito de sinalização de VoIP no futuro (GOKHALE; LU, 2005).

A próxima seção, apresentará a ferramenta Asterisk⁵, responsável por simular os recursos de um PABX convencional utilizando a tecnologia VoIP.

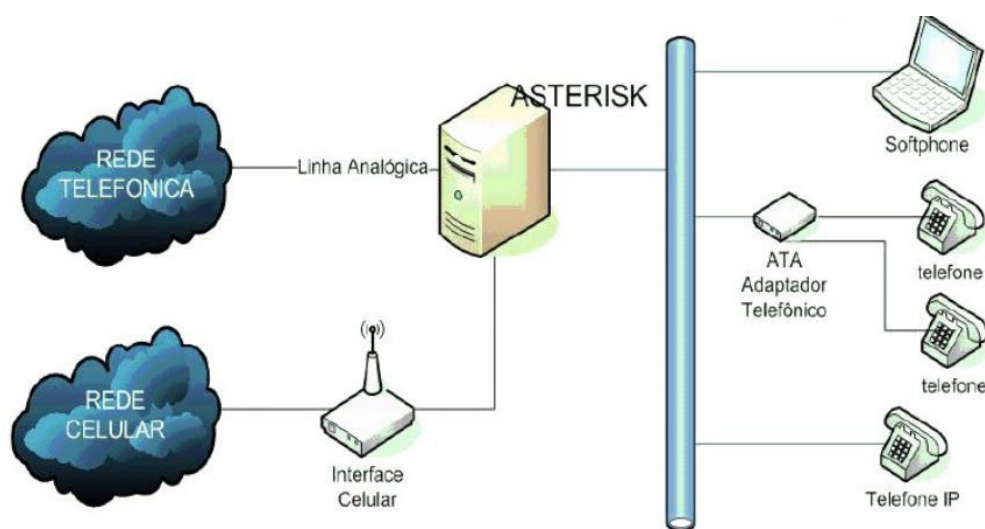
4.2 A Ferramenta Asterisk.

A ferramenta Asterisk, desenvolvida por Mark Spencer e licenciada sob a GPL (*GNU Public License*), licença comum para softwares livres, é definida por Apolinário e Bezerra (2013), como uma ferramenta de software livre, capaz de simular os recursos de um PABX convencional utilizando a tecnologia VoIP. Os mesmos, também, salientam que a ferramenta Asterisk pode substituir um PABX de grande porte, por um computador/servidor capaz de gerenciar todos os ramais telefônicos e todos os recursos de chamadas.

É possível escolher diferentes protocolos de sinalização e *codecs* em sua configuração. De acordo com Aguiar (2013), por questões de desempenho, é indicado a utilização de um mesmo codec entre as extremidades de uma chamada.

O protocolo de sinalização utilizado na infraestrutura de VoIP da Universidade Federal do Ceará campus Quixadá, é o protocolo SIP, abordado na subseção 4.1.3. As **Figuras 4 e 5**, apresentam dois possíveis casos de uso da ferramenta Asterisk.

Figura 4 - Caso de Uso número Um do Asterisk.

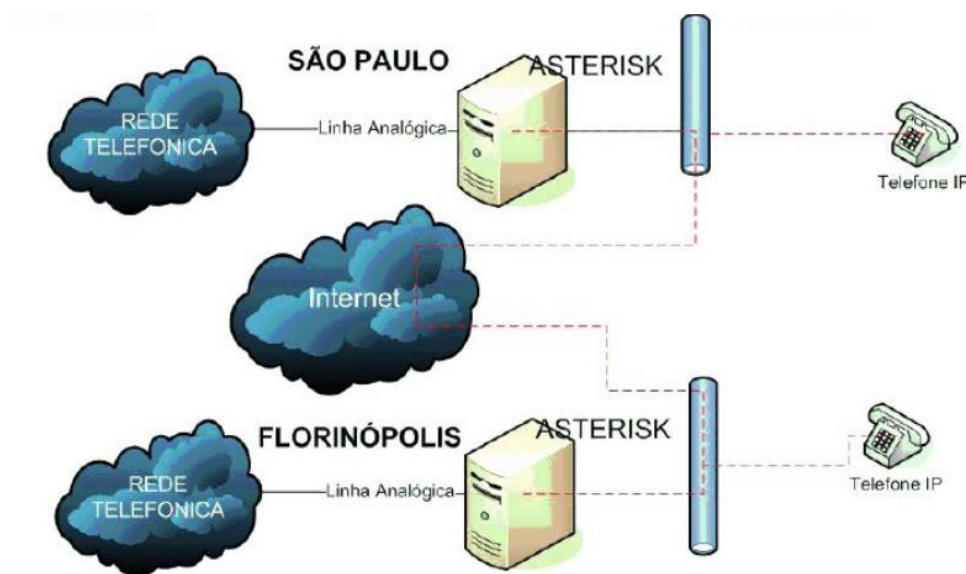


⁵ <http://www.asterisk.org/>

Fonte: (MARTINS, 2006);

Na **Figura 4**, pode-se observar um barramento local capaz de viabilizar a comunicação entre diversos tipos de dispositivos, como: um *notebook* utilizando um programa *Softphone*, *software* responsável por simular recursos de um telefone IP; telefones comuns interligados em um dispositivo ATA (Adaptador de Telefone Analógico), responsável por estender recursos da telefonia digital, tal como voz sobre IP, para telefones analógicos comuns; e um Telefone IP. Todos as chamadas são gerenciadas por um servidor Asterisk que possibilita, também, a comunicação com dispositivos de outras redes por dois caminhos, a telefonia pública convencional, através da linha analógica e a telefonia móvel, através de uma interface de um celular.

Figura 5 - Caso de Uso número Dois do Asterisk.



Fonte: (MARTINS, 2006).

Na **Figura 5**, pode-se observar, um pouco além, a comunicação entre dispositivos (Telefone IP) de diferentes redes em diferentes localizações geográficas através da internet. O gerenciamento da chamada é feito, assim como na **Figura 4**, por servidores Asterisk. Pode-se salientar, também, a possibilidade de comunicação por meio da linha analógica vista na **Figura 4**.

A próxima subseção apresentará a AMI (*Asterisk Manager Interface*), responsável por provê suporte a comandos e leitura de eventos.

4.2.1 AMI - Asterisk Manager Interface.

A partir da versão 1.6 da ferramenta Asterisk, o acesso remoto ao serviço por meio de uma interface chamada AMI, com o objetivo de ler eventos e acionar comandos, foi consolidado nativamente na solução (KAPICAK et al., 2011).

A documentação desta API pode ser encontrada no *wiki*⁶ do Asterisk, onde é possível visualizar 71 tipos de eventos e 113 tipos de comandos, desde sua última atualização feita em julho de 2012, capazes de gerenciar o serviço de voz sobre IP provido pela ferramenta Asterisk.

Bibliotecas em diferentes tipos de linguagem utilizam essa API para gerenciar em tempo real o serviço de voz, um exemplo de solução seria a PAMI⁷ (PHP *Asterisk Manager Interface*), porém esta solução não é viável para este projeto que propõem a utilização de uma nova abordagem envolvendo a *stack* MEAN. A solução utilizada para este projeto foi a biblioteca Asterisk-Manager⁸, desenvolvida para Node.js.

A próxima seção apresentará os conceitos da *stack* utilizada neste trabalho, com objetivo de descrever as tecnologias utilizadas por ela e justificar a escolha da mesma.

4.3 Stack MEAN (MongoDB, Express, AngularJS e Node.js)

De acordo com Poulter, Johnsto e Cox (2015) a *stack* MEAN, como é conhecida, ou simplesmente MEAN, é um conjunto de quatro ferramentas baseadas na linguagem JavaScript, onde juntas proveem componentes de interatividade em ambos os lados, *client-side* e *server-side*, voltadas para aplicações WEB.

O MongoDB provê um bando de dados de objetos, Express provê um framework de roteamento WEB compatível com Node.js, AngularJS provê componentes voltados para o *front-end* da aplicação e Node.js é o compilador JavaScript com componentes voltados para o *back-end* da aplicação.

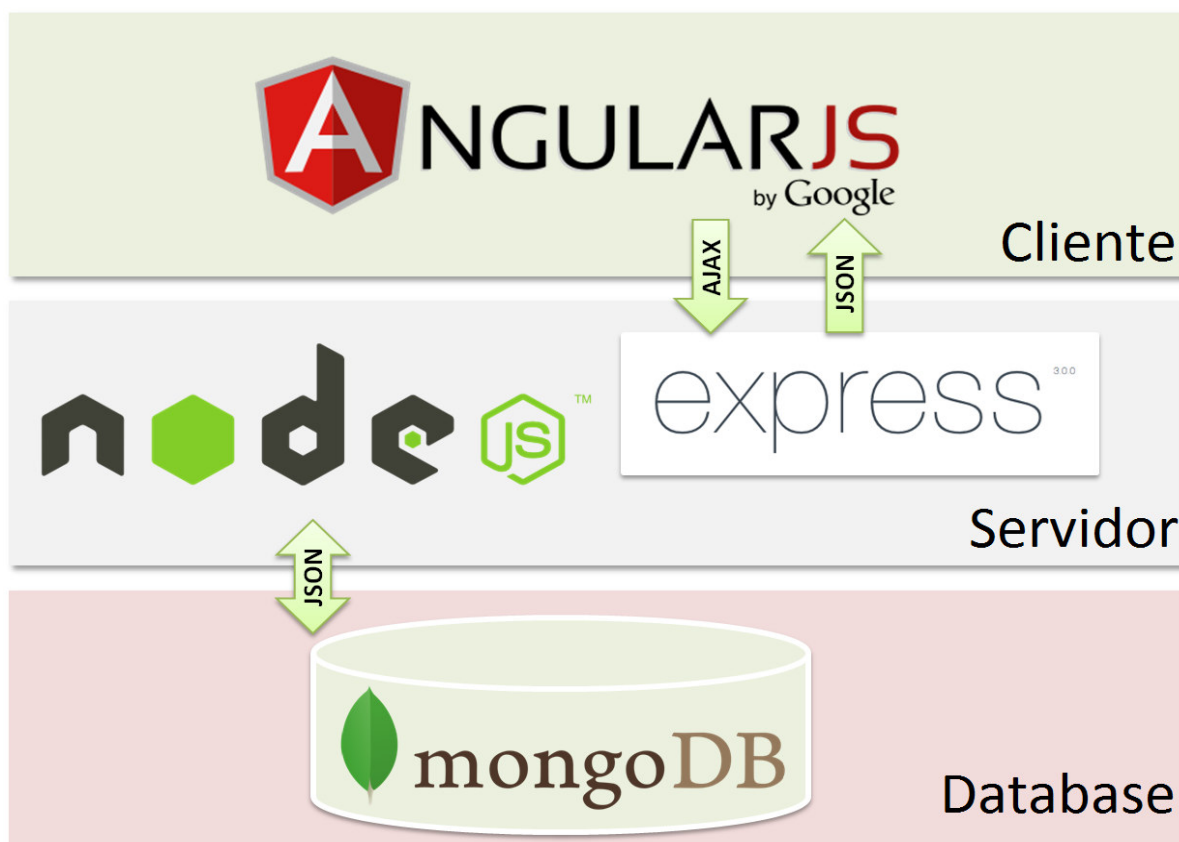
A **Figura 6** apresenta a integração e comunicação de todas as ferramentas da *stack*. Na figura abaixo, é possível observar a comunicação entre o módulo cliente e servidor através de requisições AJAX e mensagens no formato do padrão JSON⁹ (*JavaScript Object Notation*). O acesso ao banco de dados é feito pelo módulo servidor, e comunicam-se através de objetos e mensagens JSON.

⁶ <https://wiki.asterisk.org>

⁷ <http://marcelog.github.io/PAMI/>

⁸ <https://www.npmjs.com/package/asterisk-manager>

⁹ <http://www.json.org/>

Figura 6 – Integração e Comunicação da *stack* MEAN

Fonte: (SILVA, 2014), editada pelo autor.

As próximas subseções detalham cada uma destas ferramentas, apresentando algumas de suas principais vantagens.

4.3.1 Node.js e Express

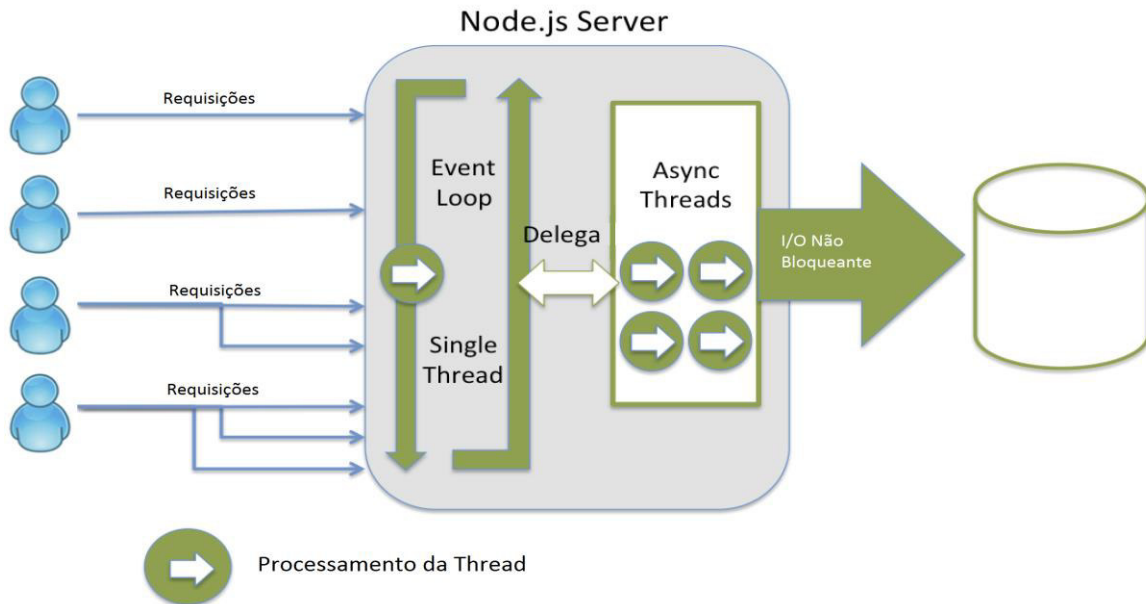
Implementada em C++ e em torno do compilador V8 da Google, a ferramenta Node.js, ou apenas Node, como é conhecida, é a mais importante ferramenta da *stack*, capaz de provê alta performance e baseada em eventos assíncronos (POULTER; JOHNSTO; COX, 2015).

O trabalho de Poulter, Johnsto e Cox (2015) argumenta que a ferramenta em questão, juntamente com o framework Express, aumentam o nível de produtividade, diminuindo o esforço de desenvolvimento necessário, entregando uma solução mais eficiente e de implementação altamente escalável, ou seja, flexível à novas mudanças.

A **Figura 7** apresenta o funcionamento de um servidor Node.js, e demonstra a abordagem utilizada pela arquitetura da ferramenta em questão, diferenciando das demais

soluções existentes e destacando componentes e técnicas, tais como, *Eventloop* e *threads* assíncronas, utilizadas para prover melhor desempenho.

Figura 7 - Funcionamento de um Servidor Node.js.



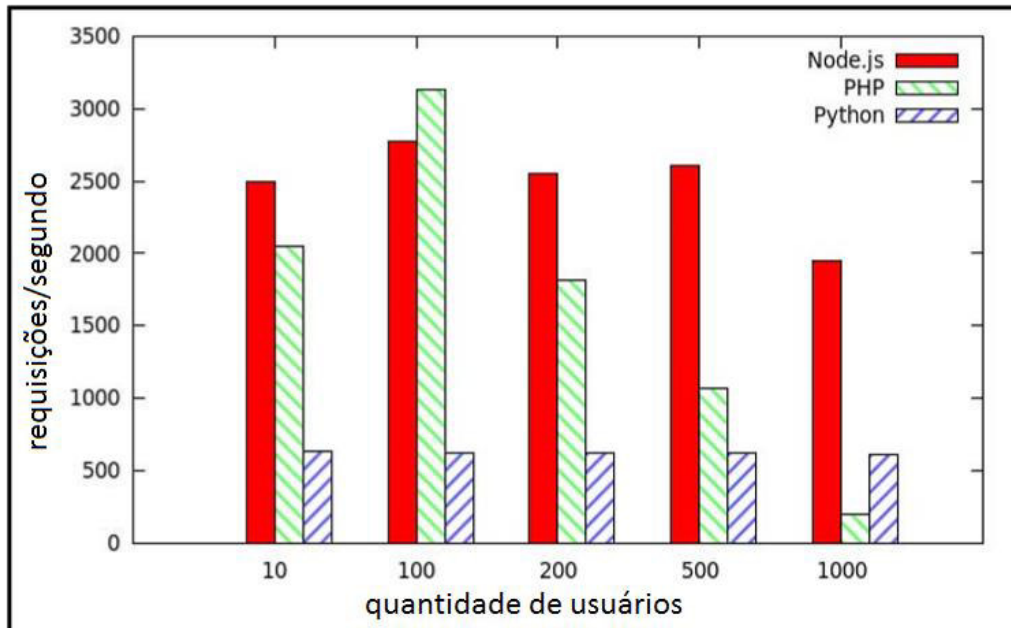
Fonte: (STRONGLOOP, 2014), editada pelo autor.

Na figura acima, pode-se observar uma quantidade de requisições feitas ao servidor Node.js por quatro origens distintas, todas as requisições são processadas, diferentemente de um servidor *multi-thread*, em uma única *thread*, chamada *Eventloop*, à qual delega para *threads* assíncronas, a responsabilidade da execução do código de maneira não bloqueante, ou seja, fornecendo as entradas devidas, e quando a saída estiver pronta, notificar e entregar ao destinatário devido.

O trabalho de Lei, Ma e Tan (2014), demonstra o desempenho obtido em função da abordagem descrita no parágrafo anterior, de um web *server* Node em comparação com um web *server* em Python e outro em PHP.

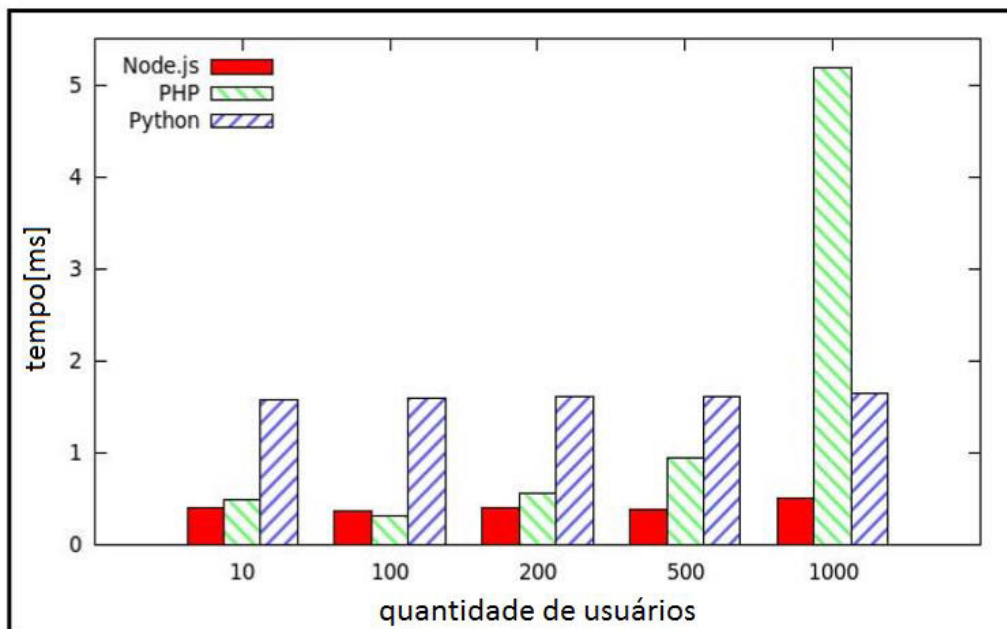
A **Figura 8 e 9**, mostram a performance do cálculo do décimo valor de Fibonacci com o objetivo de descobrir quantas requisições por segundo cada um dos web *services* puderam atender e em quanto tempo cada requisição foi atendida.

Figura 8 - Análise de desempenho Node.js, requisições por segundo.



Fonte: (LEI; MA; TAN, 2014), editada pelo autor.

Figura 9 - Análise de desempenho Node.js, tempo em milissegundos de cada requisição.



Fonte: (LEI; MA; TAN, 2014), editada pelo autor.

Na **Figura 8**, pode-se observar que web *server* em Node.js conseguiu atender mais requisições por segundo em quase todas os cenários. E a **Figura 9**, mostra que a ferramenta Node.js obteve maior desempenho ao atender as requisições em menos tempo que

as demais soluções. Os resultados das médias produzidos por Lei, Ma e Tan (2014), podem ser observados na **Tabela 2** para melhor entendimento.

Tabela 2 - Análise de desempenho do Node.js: Cálculo do décimo número de Fibonacci.

WEB Server	Média dos Cenários: Análise em requisições/segundo	Média dos Cenários: Análise do tempo em Milissegundos
Node.js	2491.77	0.401
Python-Web	633.68	1.578
PHP	2051.22	0.488

Fonte: (LEI; MA; TAN, 2004), editada pelo autor.

Claramente a ferramenta Node.js¹⁰ demonstrou maior performance que as demais nas diferentes análises.

A ferramenta Express¹¹ é um framework de aplicações Web voltados para Node.js. Ela fornece uma básica interface para a ferramenta Node.js, capaz de provê ao desenvolvedor mecanismos para manipulação de roteamento e operações HTTP. O objetivo da ferramenta é facilitar a implementação de aplicações web, em relação a quantidade de código e soluções mais elegantes no desenvolvimento de serviços em relação a implementação diretamente no Node. A próxima seção apresentará o MongoDB, ferramenta utilizada como base de dados da aplicação.

4.3.2 MongoDB

Tradicionalmente, aplicações web têm usado como modelo de banco de dados um sistema relacional, geralmente baseado em SQL, utilizando ferramentas como MySQL e PostgreSQL, porém há uma tendência crescente na utilização de bancos de dados do tipo NoSQL, referenciado como “*Non SQL*” ou “*Non Relational*”, ou seja, “Não SQL ou Não Relacional”, essa tendência deriva da necessidade de um tipo de base de dados mais flexível e de esquema dinâmico que seja facilmente alterado (POULTER; JOHNSTO; COX, 2015).

¹⁰ <https://nodejs.org/en/>

¹¹ <http://expressjs.com/>

De acordo com Poulter, Johnsto e Cox (2015), o MongoDB¹² é um banco de dados NoSQL, orientado a documentos e baseado no formato JSON, o que fornece alta compatibilidade para ambientes JavaScript.

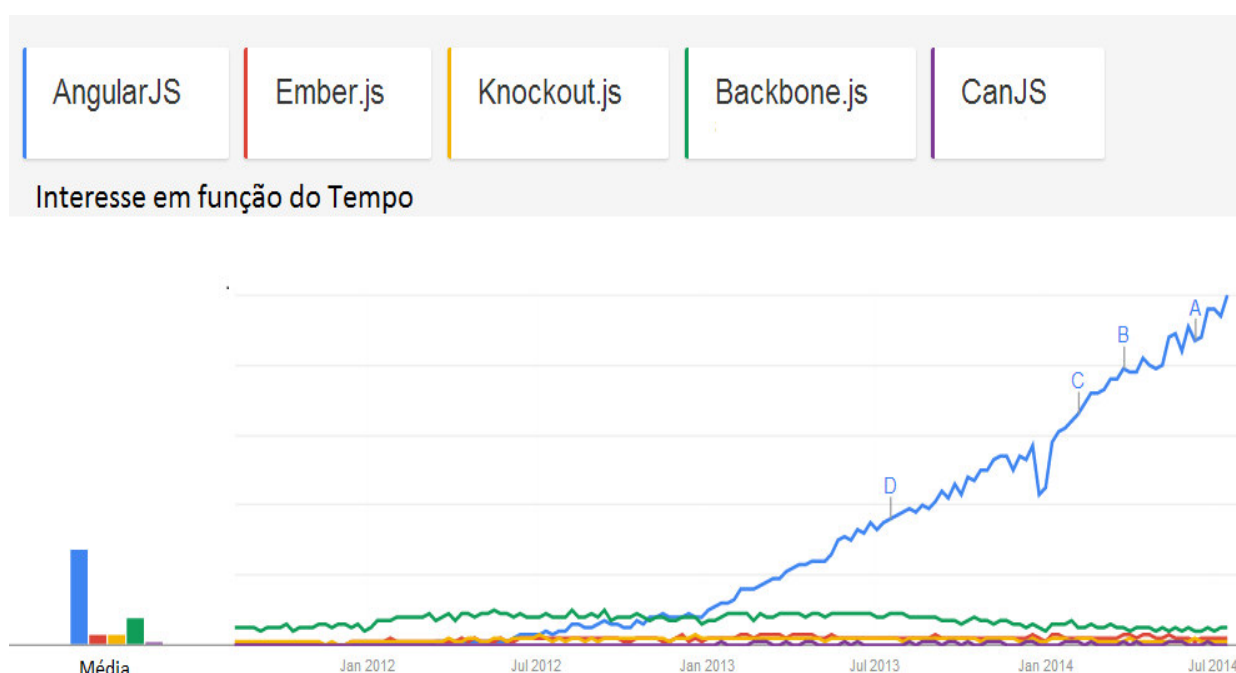
Poulter, Johnsto e Cox (2015) afirmam que uma das vantagens da utilização de um banco de dados orientado a documentos, é que o mesmo, oferece um esquema dinâmico, ou seja, se necessário a alteração de uma determinada estrutura no banco de dados, como a adição de novos parâmetros, não haverá manipulações em grande escala em todo o banco de dados.

4.3.3 AngularJS

AngularJS¹³ é a última parte da *stack*, definido como um framework de código livre que provê recursos para *client-side* baseado no modelo arquitetural MVC (*Model-View-Controller*) (POULTER; JOHNSTO; COX, 2015).

Um gráfico produzido pela ferramenta do Google chamada *Trends*, demonstra que o nível de interesse pela ferramenta teve um grande salto de 2012 a 2014 em comparação à outras soluções já existentes no mercado, como Ember.js, Backbone.js, CanJS e Knockout.js. É possível observar na **Figura 10** o gráfico em questão.

Figura 10 - Nível de interesse AngularJS ao longo dos últimos anos.



¹² <https://www.mongodb.org/>

¹³ <https://angularjs.org/>

Fonte: (DTIDIGITAL, 2016), editada pelo autor.

A próxima seção apresentará o desenvolvimento deste projeto, tal como, ferramentas utilizadas, implementação e validação.

5 PROPOSTA

Este trabalho propõe a ferramenta de gerenciamento chamada VoipBox, baseada nos conceitos abordados na seção **4 Fundamentação Teórica** voltados para solucionar problemáticas descritas na seção **1 Introdução**, seguindo como meta, a obtenção dos objetivos abordados na seção **3 Objetivos Gerais e Específicos**.

Nas seções **5.1** e **5.2** serão apresentadas as principais informações sobre o projeto, tais como, arquitetura, requisitos funcionais e diagrama de caso de uso.

5.1 Arquitetura VoipBox

A aplicação principal é constituída pela *stack* MEAN e trabalha em sua própria arquitetura (**Figura 6**). Entretanto, a solução utiliza bibliotecas que disponibilizam interfaces de comunicação com servidores externos que se encontram na mesma rede local da solução, possibilitando a troca de informação entre servidores terceiros e a aplicação.

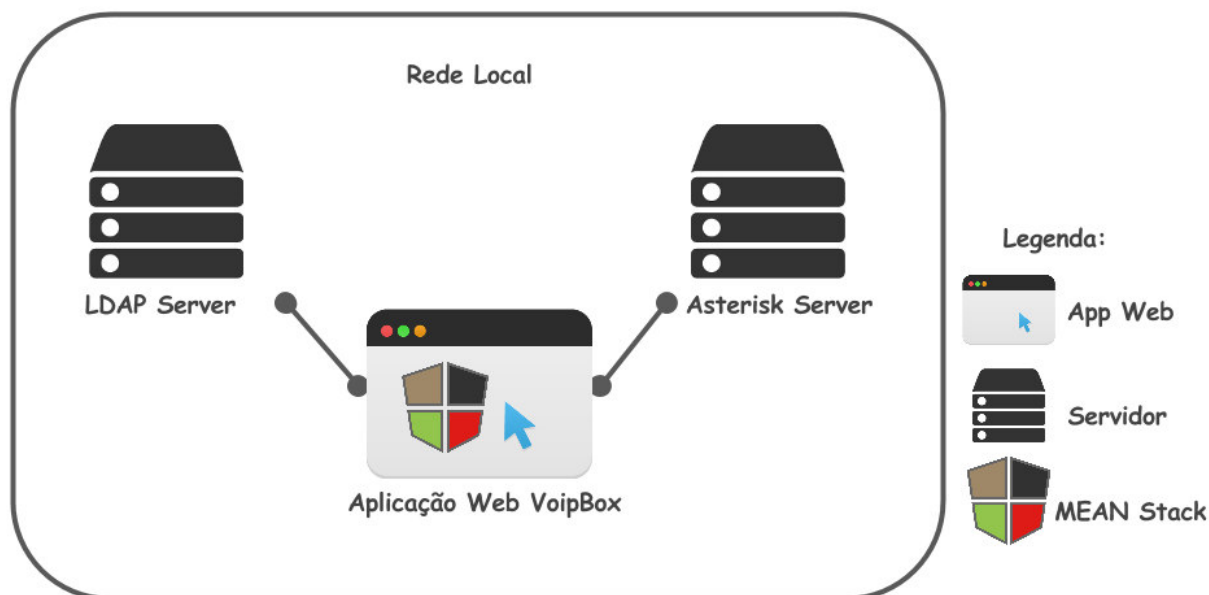
O principal serviço utilizado pela solução é o servidor Asterisk, responsável por provê o serviço de voz sobre IP. Por meio deste, a solução é capaz de capturar em tempo real informações sobre as chamadas, ramais e também intervir diretamente nos arquivos de configuração do servidor, possibilitando assim, a inserção e remoção de novos ramais.

Um outro serviço utilizado pela a solução é o servidor LDAP (*Lightweight Directory Access Protocol*), responsável por provê autenticação de usuários já cadastrados na rede da instituição. Por meio desse serviço a solução é capaz, também, de prover filtro de autenticação por um determinado grupo cadastrado na base LDAP, sendo capaz de estabelecer que permissões os usuários decorrentes deste grupo terão na aplicação, permissões estas de administrador ou apenas de leitura.

Na **Figura 11** é possível observar a arquitetura e comunicação da solução VoipBox entre os dois principais servidores utilizados: Asterisk e LDAP. O símbolo representativo da *stack* MEAN dentro da solução, representa a forma em que a solução

trabalha, tal como, sua comunicação entre os módulos de cliente, servidor e base de dados descritos na seção 4.3, ilustrada na **Figura 6**.

Figura 11 – Arquitetura e comunicação da solução VoipBox.



Fonte: Elaborada pelo autor.

A próxima seção define as funcionalidades da solução, dividindo-as em cinco categorias para melhor entendimento.

5.2 Protótipo e Funcionalidades da Solução

A solução VoipBox conta com funcionalidades levantadas em conjunto com os membros da CTI da Universidade Federal do Ceará - Campus Quixadá. Para melhor compreensão e detalhamento de todas as funcionalidades, pode-se dividi-las em cinco categorias: *Login*, *Dashboard*, *Usuários*, *Ramais* e *Chamadas*. As próximas subseções entram em mais detalhes sobre cada uma destas categorias.

5.2.1 Login

Esta categoria possui em específico duas grandes funcionalidades, autenticação local e autenticação com LDAP. É possível, após a inicialização da aplicação, a realização imediata do *login* de ambas as possibilidades, entretanto é necessário configurar, antes da inicialização da aplicação, o arquivo *default.js* que se encontra no diretório *config/env/* na pasta principal do projeto.

Neste arquivo, é necessário inserir as informações necessárias para comunicação com o servidor LDAP e o banco de dados da aplicação, neste caso, MongoDB. A **Figura 12** demonstra um exemplo de como esse arquivo poderia estar configurado.

Figura 12 – Configuração do MongoDB e LDAP para comunicação

```

mongo:{
  db: "voipbox_db",
  host: "mongodb://127.0.0.1/",
  port: null,
  username: "voipbox",
  password: "segrendo456"
},

ldap:{

  url: 'ldap://127.0.0.1:389',
  dc: 'dc=exemplo,dc=com,dc=br',
  active : true,

  baseLdap: {
    bindSuffix: 'cn=user,dc=exemplo,dc=com,dc=br',
    password: 'segredo789'
  },

  userLdap: {
    bindSuffix: 'ou=funcionarios,dc=exemplo,dc=com,dc=br'
  },

  authList: {
    active : false,
    memberFilter: 'cn=funcionarios_admin,ou=funcionarios,dc=exemplo,dc=com,dc=br',
    powerAuthority: 'administrator'
  }
}
}

```

Fonte: Elaborada pelo autor.

Os atributos presentes na figura acima já estarão no arquivo de configuração, será necessário apenas a atribuição dos valores pelo o usuário administrador da rede da instituição.

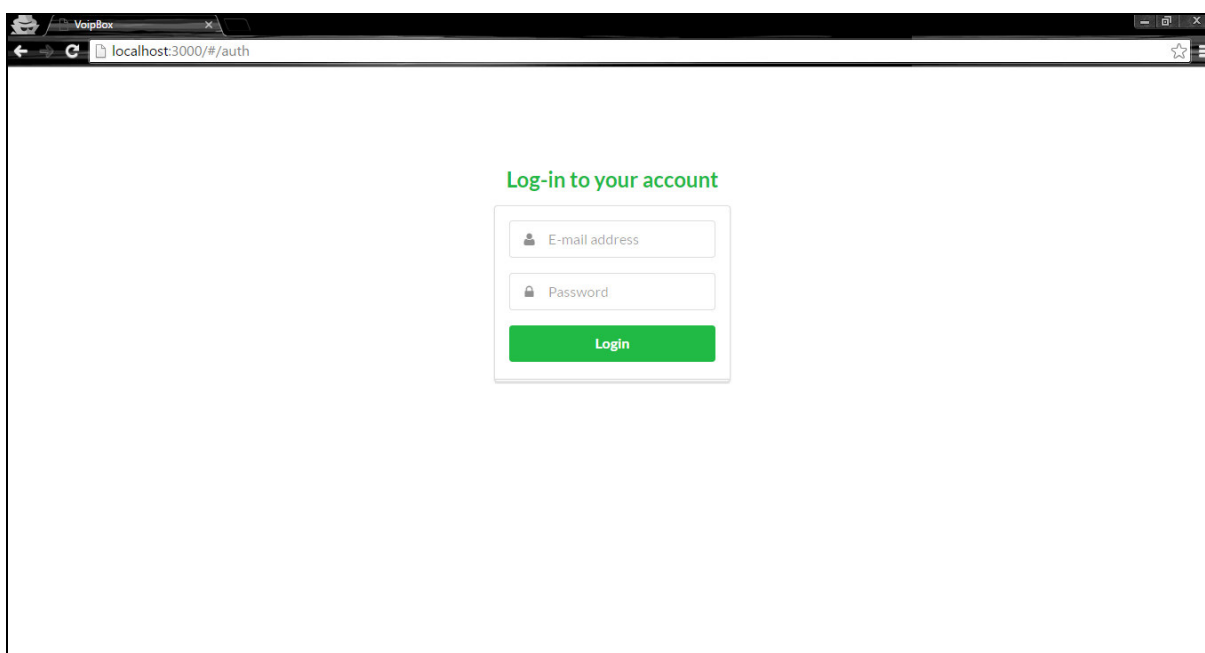
Em relação as informações para comunicação com o MongoDB, os atributos necessários para a realização da interação entre a base de dados e a aplicação é o nome do banco, a *url* que carrega o endereço do serviço na rede local, a porta do serviço, nome do usuário e senha. Caso não possua senha ou usuário o valor a ser inserido deverá ser “*null*”.

Em relação as informações necessárias para a comunicação com o LDAP, é possível observar que os atributos necessários para interação são, a *url* juntamente com a porta responsável pela localização do serviço na rede, o domínio raiz da base LDAP e um atributo responsável por indicar a aplicação se a funcionalidade de *login* com LDAP está

ativada ou não. Caso ativada, será necessário atribuir os valores responsáveis por acessar a base geral, informando o sufixo da base (domínio juntamente com o usuário) e a senha, e o valor necessário para encontrar o ramo da base LDAP que contém as informações de todos os usuários da instituição.

Em seguida, é possível analisar os atributos que compõe o filtro de autenticação por um determinada grupo da base LDAP. Assim como a aplicação necessita de um atributo para compreender se deve utilizar *login* com LDAP ou não, um atributo de mesmo propósito deve ser atribuído para indicar se a aplicação deve levar em conta o filtro de autenticação ou não. Caso ativado, é necessário atribuir o caminho inteiro do grupo seguido de seu usuário e atribuir que tipo de permissão terão os usuários desse grupo, permissões de administrador ou apenas de leitura. A **Figura 13** demonstra a tela de *login* da solução VoipBox.

Figura 13 – Tela de Login VoipBox



Fonte: Elabora pelo autor.

A próxima subseção contempla detalhes da categoria Usuários.

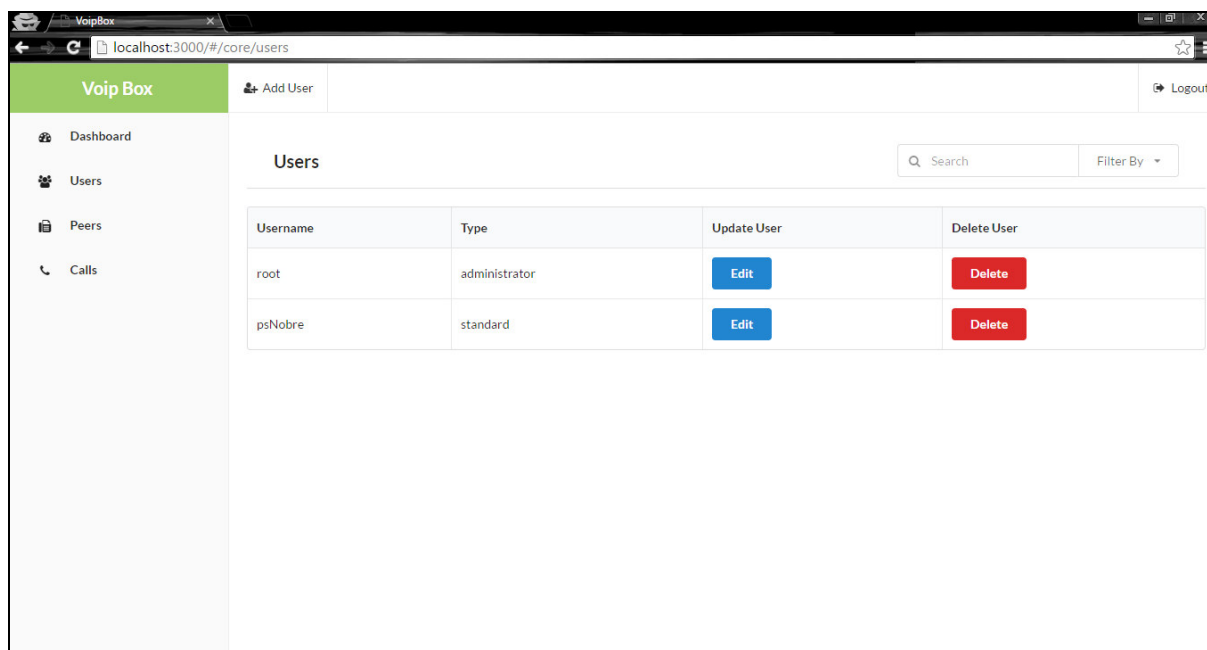
5.2.2 Usuários

Esta categoria é responsável pelo gerenciamento de usuários da aplicação e contém a lista de todos os usuários cadastrados. É importante salientar que todos os usuários

desta categoria são usuários locais e encontram-se persistidos no MongoDB, diferente dos usuários cadastrados na base LDAP.

As funcionalidades presentes nesta categoria são: atualização, remoção e criação de novos usuários. Na **Figura 14** é possível observar a tela de usuários.

Figura 14 – Tela de Usuários VoipBox



Fonte: Elaborada pelo autor.

O MongoDB, como já citado na subseção 4.3.2, é um banco de dados *NoSQL* orientado a documentos e de esquema dinâmico, ou seja, não existe uma maneira de representar a relação entre as tabelas como no modelo relacional, pois não existem tabelas no MongoDB, existem documentos distribuídos em coleções. Porém, é possível citar quais atributos são utilizados para representar cada documento encontrado na coleção usuários.

Cada usuário possui três atributos: nome do usuário, senha e tipo. Quanto ao atributo tipo, é possível classificá-lo como *administrator* (Administrador), capaz de exercer ações de escrita e leitura, e o usuário *standard* (Padrão), podendo apenas exercer funções de leitura.

Na **Figura 15** é possível observar a tela de adição de um novo usuário, onde encontra-se um formulário com campos a serem preenchidos pelos atributos citados no parágrafo anterior, e um campo específico para confirmação da senha, responsável por validar a entrada em questão.

Figura 15 – Tela de criação de novo usuário.

Fonte: Elaborada pelo autor.

A próxima subseção contempla detalhes da categoria Chamadas.

5.2.3 Chamadas

Esta categoria registra em tempo real as chamadas realizadas por todos os ramais cadastrados no servidor Asterisk. As principais funcionalidades desta categoria são: a persistência dos registros de chamada, a remoção dos mesmos e a consulta do estado da chamada em tempo real.

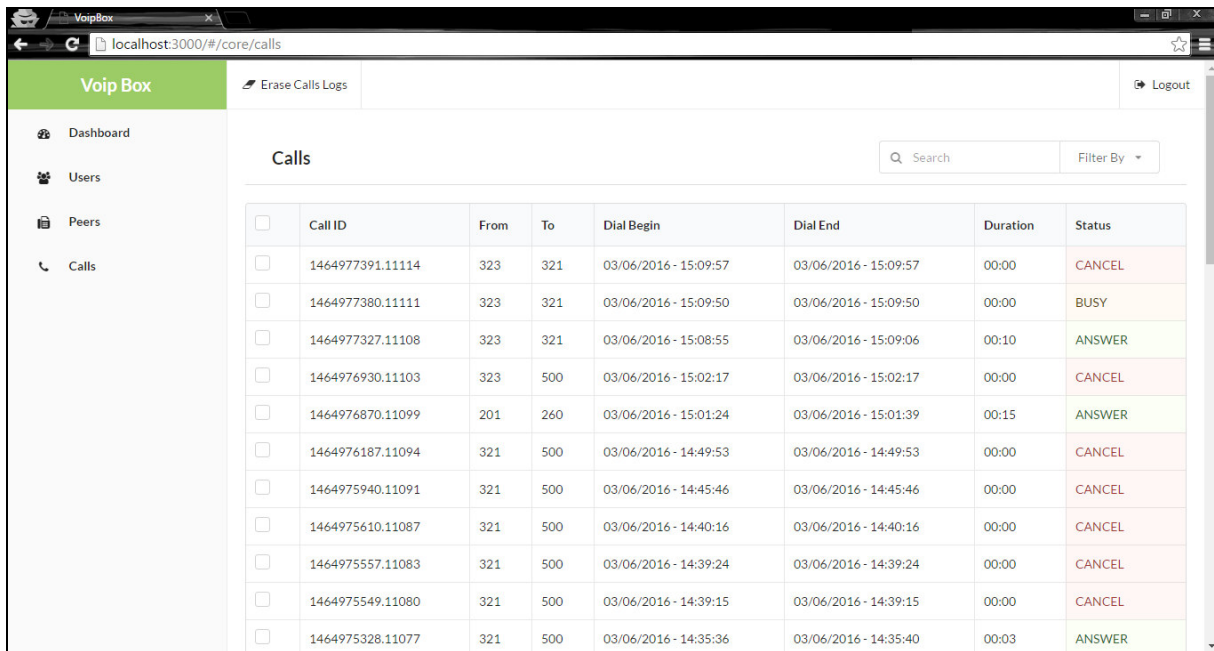
Assim como explicado na subseção anterior, a utilização do MongoDB como base de dados permite apenas o detalhamento dos atributos de cada chamada, representada por um documento, persistida na coleção chamadas.

Cada chamada possui seis atributos: o id da chamada, o ramal de origem, o ramal de destino, o *status* da chamada, o tempo de início da chamada e o tempo de término da chamada. Quanto ao *status* da chamada, é possível classifica-lo em quatro tipos:

- *answer* (Atendida) – Especifica se a chamada foi bem sucedida.
- *noanswer* (Não atendida) – Se a chamada tocou enquanto pôde e não obteve resposta.
- *cancel* (Cancelada) – Se a chamada foi cancelada pelo próprio ramal de origem.
- *busy* (Ocupado) – Caso a chamada seja cancelada pelo ramal de destino ou o mesmo já esteja em alguma outra ligação.

A **Figura 16** apresenta uma ilustração da tela de Chamadas da solução VoipBox.

Figura 16 – Tela de Chamadas VoipBox.



<input type="checkbox"/>	Call ID	From	To	Dial Begin	Dial End	Duration	Status
<input type="checkbox"/>	1464977391.11114	323	321	03/06/2016 - 15:09:57	03/06/2016 - 15:09:57	00:00	CANCEL
<input type="checkbox"/>	1464977380.11111	323	321	03/06/2016 - 15:09:50	03/06/2016 - 15:09:50	00:00	BUSY
<input type="checkbox"/>	1464977327.11108	323	321	03/06/2016 - 15:08:55	03/06/2016 - 15:09:06	00:10	ANSWER
<input type="checkbox"/>	1464976930.11103	323	500	03/06/2016 - 15:02:17	03/06/2016 - 15:02:17	00:00	CANCEL
<input type="checkbox"/>	1464976870.11099	201	260	03/06/2016 - 15:01:24	03/06/2016 - 15:01:39	00:15	ANSWER
<input type="checkbox"/>	1464976187.11094	321	500	03/06/2016 - 14:49:53	03/06/2016 - 14:49:53	00:00	CANCEL
<input type="checkbox"/>	1464975940.11091	321	500	03/06/2016 - 14:45:46	03/06/2016 - 14:45:46	00:00	CANCEL
<input type="checkbox"/>	1464975610.11087	321	500	03/06/2016 - 14:40:16	03/06/2016 - 14:40:16	00:00	CANCEL
<input type="checkbox"/>	1464975557.11083	321	500	03/06/2016 - 14:39:24	03/06/2016 - 14:39:24	00:00	CANCEL
<input type="checkbox"/>	1464975549.11080	321	500	03/06/2016 - 14:39:15	03/06/2016 - 14:39:15	00:00	CANCEL
<input type="checkbox"/>	1464975328.11077	321	500	03/06/2016 - 14:35:36	03/06/2016 - 14:35:40	00:03	ANSWER

Fonte: Elaborada pelo autor.

É importante ressaltar que esta categoria e a que será descrita na próxima subseção, recebem informações diretas do servidor Asterisk. Para que a solução se comunique com o servidor, é necessário acrescentar ao arquivo de configuração da solução, informações sobre o mesmo. A **Figura 17** demonstra um exemplo de como o arquivo *default.js* poderia estar configurado. A figura apresenta dados fictícios.

Figura 17 – Configuração do Asterisk para comunicação.

```

asterisk:{
  host: "127.0.0.1",
  port: 5038,
  username: "admin",
  password: "segredo123",
  events: true
},

```

Fonte: Elaborada pelo autor.

Para a comunicação com o servidor Asterisk, os atributos necessários para a realização desta interação são: o IP do servidor, porta do serviço, nome do usuário e senha relacionados à AMI, descrita na subseção **4.2.1**, e um atributo de eventos que deve ser atribuído como *true* (Verdadeiro), para que seja possível capturar eventos relacionados ao servidor Asterisk em tempo real.

A próxima subseção detalha a categoria Ramais.

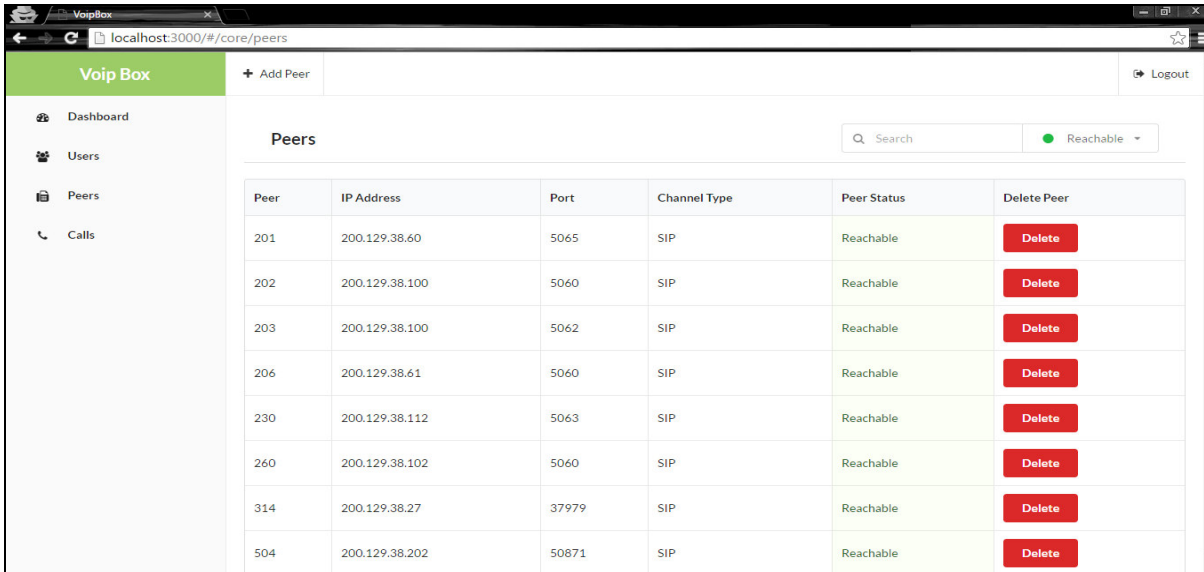
5.2.4 Ramais

Esta categoria é responsável pelo gerenciamento dos ramais, suas principais funcionalidades são: lista dos ramais com seus devidos *status*, adição de um novo ramal e remoção de ramais. Esta categoria, diferentemente das anteriores, não persiste seus dados no banco de dados local da aplicação, a biblioteca que provê comunicação com o serviço de voz, fornece recursos de edição direta nos arquivos de configuração do servidor Asterisk. Este método gera pontos positivos e negativos.

Um dos pontos positivos que pode se destacar, é o controle total do serviço de voz pela aplicação, estendendo a possibilidade de criação de novas funcionalidades para a aplicação em grandes quantidades. Entretanto, isto pode ser visto como um ponto negativo, a partir do momento que a aplicação tem total controle sobre o serviço.

Outro ponto negativo, é a falta de um método consistente para a leitura dos arquivos de configuração, a biblioteca ainda não possui um meio de ler o arquivo e retornar objetos manipuláveis a nível de aplicação. É uma espécie de edição linha por linha, o que impossibilitou a funcionalidade de edição dos ramais, ainda que por momento, mas proposta para trabalhos futuros. A **Figura 18** demonstra a tela de Ramais da solução VoipBox.

Figura 18 – Tela de Ramais VoipBox.



Peer	IP Address	Port	Channel Type	Peer Status	Delete Peer
201	200.129.38.60	5065	SIP	Reachable	Delete
202	200.129.38.100	5060	SIP	Reachable	Delete
203	200.129.38.100	5062	SIP	Reachable	Delete
206	200.129.38.61	5060	SIP	Reachable	Delete
230	200.129.38.112	5063	SIP	Reachable	Delete
260	200.129.38.102	5060	SIP	Reachable	Delete
314	200.129.38.27	37979	SIP	Reachable	Delete
504	200.129.38.202	50871	SIP	Reachable	Delete

Fonte: Elaborada pelo autor.

As informações encontradas nesta categoria sobre cada ramal são o seu número, IP do ramal na rede, porta, o tipo de canal, neste caso apenas SIP, pois é o protocolo de sinalização utilizado no serviço de voz da Universidade Federal do Ceará - Campus Quixadá, e os *status* dos ramais. Os *status* dos ramais podem ser classificados em cinco tipos e divididos em dois grupos, *status* inicial e *status* em tempo de execução.

O grupo do *status* inicial é definido pelo estado em que os ramais se encontram assim que a aplicação é inicializada. Eles podem ser classificados em três tipos:

- *Reachable* (Acessível) – Define os ramais que se encontram dentro da rede e possuem conectividade.
- *Unreachable* (Inacessível) – Define os ramais dentro da rede que perderam conectividade.
- *Unknown* (Desconhecido) – Define os ramais que estão fora da rede, ou ramais adicionados no servidor Asterisk, mas nunca atribuídos a um dispositivo ou *Softphone*.

O grupo do *status* em tempo de execução é definido pelo o estado dos ramais que se autenticaram com o servidor Asterisk no decorrer da execução da aplicação, eles são classificados em dois tipos:

- *Registered* (Registrado) – Define o ramal que se registrou com o serviço de voz.
- *Unregistered* (Desregistrado) – Define o ramal que se desregistrou do serviço de voz.

Isso leva a uma conclusão redundante por parte de alguns *status*, pois um ramal com o *status Registered* possui o mesmo valor que um ramal *Reachable*, o mesmo se aplica aos *status Unregistered e Unreachable*, a única diferença é o tempo que estado foi atribuindo em função da aplicação.

É importante ressaltar que uma inserção ou remoção de um ramal altera diretamente o arquivo de configuração do servidor Asterisk, o que leva à uma releitura deste arquivo por parte da aplicação, ocasionado a retorno dos ramais de *status Registered* para *Reachable* e *Unregistered* para *Unreachable* novamente.

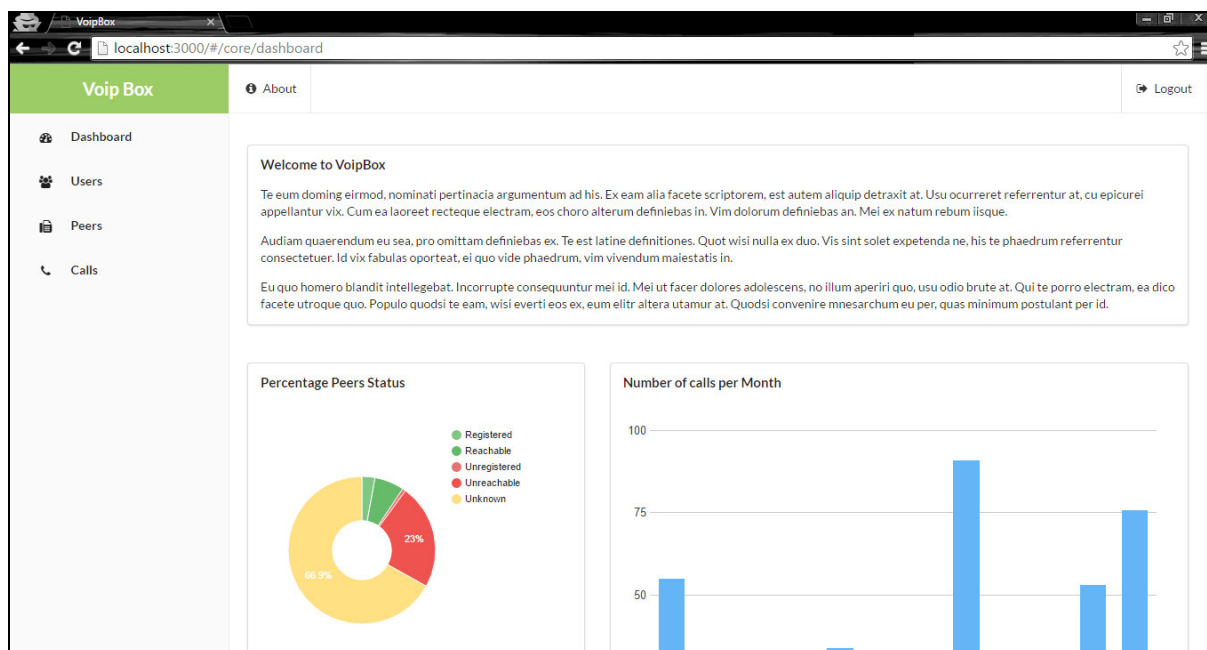
A próxima subseção detalha a última categoria a ser analisada, *Dashboard*.

5.2.5 Dashboard

Esta categoria é responsável por coletar informações de todas as outras categorias já descritas e transformá-las em informações de acesso mais rápido e de fácil entendimento para o usuário. Podem ser citadas como exemplo de informações voltadas para esta categoria, uma lista com as cinco últimas chamadas realizadas, os *status* dos ramais representados em

gráficos de maneira a prover uma visão geral dos ramais na rede e etc. A **Figura 19** demonstra a tela de *Dashboard* da solução VoipBox.

Figura 19 - Tela de *Dashboard* VoipBox.



Fonte: Elaborada pelo autor.

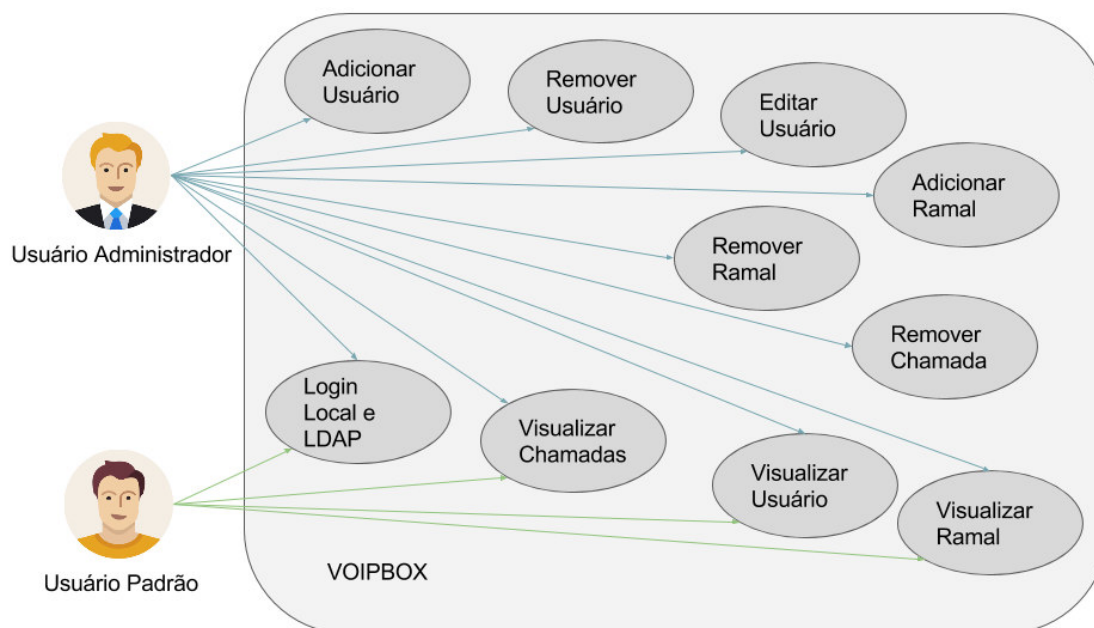
A próxima subseção demonstra uma visão geral de todas funcionalidades apresentadas nas categorias citadas acima.

5.2.6 Visão geral das Funcionalidades.

Depois de detalhada cada funcionalidade. Para obter-se uma visão geral da interação entre cada tipo de usuário e os recursos que a solução VoipBox provê, a **Figura 20** apresenta o diagrama de caso de uso da aplicação.

Na figura que se encontra abaixo, é possível observar os dois usuários já citados na subseção **5.2.2**, a mesma contemple as funcionalidades que cada um pode interagir diretamente. É possível salientar que o usuário padrão se restringe a operações de leitura, em contrapartida, o usuário administrador disponibiliza de interação integral das funcionalidades dispostas na aplicação.

Figura 20 – Diagrama de caso de uso VoipBox.



Fonte: Elaborada pelo autor.

6 AVALIAÇÃO DA SOLUÇÃO

As seções 6.1 e 6.2, descrevem o método utilizado para a avaliação da solução, o ambiente da avaliação, como instalar o ambiente e os resultados da análise qualitativa realizada.

6.1 Instalação do Ambiente de Avaliação

Para avaliação da ferramenta, a Coordenadoria de Tecnologia da Informação (CTI), disponibilizou uma máquina virtual em sua nuvem interna. A máquina possui as seguintes configurações:

- Sistema operacional – Ubuntu Server 14.04 / 64 bits
- Memória – 2GB RAM
- Disco – 40GB

Os próximos comandos que serão descritos serão necessários para a instalação do ambiente. Primeiramente iniciamos com comando “*sudo apt-get install git mongodb nodejs nodejs-legacy npm*”, responsável por instalar as principais ferramentas do ambiente da solução.

Após a instalação das ferramentas principais, utiliza-se o pacote *npm* para instalar duas outras ferramentas importantes para a solução, “*sudo npm install -g bower pm2*”. A ferramenta *bower* é um gerenciador de pacotes utilizado para instalação de pacotes especificamente responsáveis por manuseamento de componentes HTML, CSS e Javascript como AngularJS, jQuery e até mesmo *frameworks* como Bootstrap, Foundation e Semantic-Ui. Já a ferramenta *pm2* é utilizada para gerenciar processos em produção, pode ser utilizada com várias aplicações ao mesmo tempo, e a mesma já possibilita balanceamento de carga entre as aplicações que estão sendo executadas em um mesmo servidor.

Agora que todas as ferramentas estão disponíveis, pode-se iniciar o processo de configuração da aplicação. Inicialmente é necessário fazer o clone da aplicação com a ferramenta *git*, passando o endereço do repositório da solução, “*git clone https://github.com/psNobre/voipbox.git*”. Depois do clone da aplicação, entramos em seu diretório principal, “*cd voipbox*”, e instalamos todas as dependências da solução com o comando “*sudo npm install*”.

Depois de baixadas as dependências da aplicação a nível de servidor, deve-se entrar na pasta *public* como o comando “*cd public*”, e instalar todas as dependências do cliente com o comando “*bower install*”.

Após a instalação de todas as dependências tanto à nível de servidor como cliente, dois últimos passos são necessários para a execução da solução. Deve-se criar o banco de dados e importar o usuário “root” para ter acesso a aplicação. Para isso é necessário voltar ao diretório principal da aplicação com o comando “*cd ..*”, e executar o comando “*mongoimport --db voipbox_db --collection users --file config/db/users.json*”.

Por fim, basta executar o comando “*pm2 start app.js*”, e a aplicação já poderá ser acessada pelo *browser* passando o IP da máquina e a porta da aplicação, a porta inicialmente configurada é a 3000.

É importante salientar que para a aplicação ter comunicação com o servidor Asterisk e LDAP, ou até mesmo mudar a porta que a aplicação deve ter como endereço, é necessário acessar o arquivo *default.js* que se encontra no diretório *config/env/* dentro do diretório principal da aplicação e colocar as informações necessárias para comunicação ilustradas nas **Figuras 12 e 17**.

A próxima seção entrará em detalhes sobre o método utilizado para avaliar a solução e as pessoas que participaram do processo de avaliação da ferramenta.

6.2 Método de avaliação

Para a avaliação da solução VoipBox, foi elaborado um questionário de dezesseis questões, onde as mesmas, levantam assuntos relacionados à usabilidade, funcionalidade, interoperabilidade e confiabilidade. É possível ter acesso ao questionário no **Apêndice A**.

Cada questão possui cinco possíveis alternativas que podem classificá-la como:

- Satisfatória – A ferramenta atende completamente aos critérios questionados pela questão.
- Parcialmente Satisfatório - A ferramenta atende aos critérios questionados, porém alguma funcionalidade ou aspecto da aplicação que seja de prioridade baixo, está em falta.
- Indiferente – Faltam aspectos ou funcionalidades relevantes, mas ainda é possível atingir os objetivos mais importantes da questão.
- Parcialmente Insatisfatório – A ferramenta, em relação ao questionamento, não atende aos requisitos mínimos, porém algo de prioridade média ou baixa foi implementado com sucesso.
- Insatisfatório – A ferramenta não atende em nada questionado e não possui, nem mesmo, aspectos ou funcionalidades de prioridade média ou baixa.

O questionário foi submetido a avaliação de dez pessoas, três bolsistas da CTI, três membros da CTI e quatro professores. É importante ressaltar, que todos os dez avaliadores escolhidos possuem conhecimento das tecnologias e serviços abordados na ferramenta em avaliação.

A avaliação é feita individualmente com cada um dos dez avaliadores, onde os mesmos devem seguir um roteiro de testes com o objetivo de nortear a utilização da aplicação durante o processo de avaliação. Salientamos que o roteiro teve como objetivo nortear, nada na avaliação interfere o avaliador de tentar ações alternativas que não estejam no roteiro. É possível observar com maiores detalhes este roteiro de teste no **Apêndice B**.

A próxima seção apresentará os resultados, entrará em detalhes sobre considerações a respeito dos resultados e mostrará sugestões feitas pelos avaliadores que serão propostas para trabalhos futuros.

6.3 Resultados da Avaliação

Esta seção aborda os resultados coletados sobre a solução VoipBox. A tabela a seguir mostrará a quantidade total de respostas submetidas por cada questão. É importante

informar que, como dez pessoas fizeram a avaliação, foi levado a crer que cada questão teria dez respostas, totalizando 160, entretanto alguns avaliadores não responderam o questionário todo, e as questões 1, 4 e 7, como pode ser observado na tabela, possuem um total de 9 respostas submetidas, diferente das demais. Este ocorrido possibilitou a análise de um total de 157 respostas submetidas e não 160. O motivo de algumas questões não terem sido respondidas não foi informado. Fez-se então necessário, a informação do índice “Total de Respostas” na tabela.

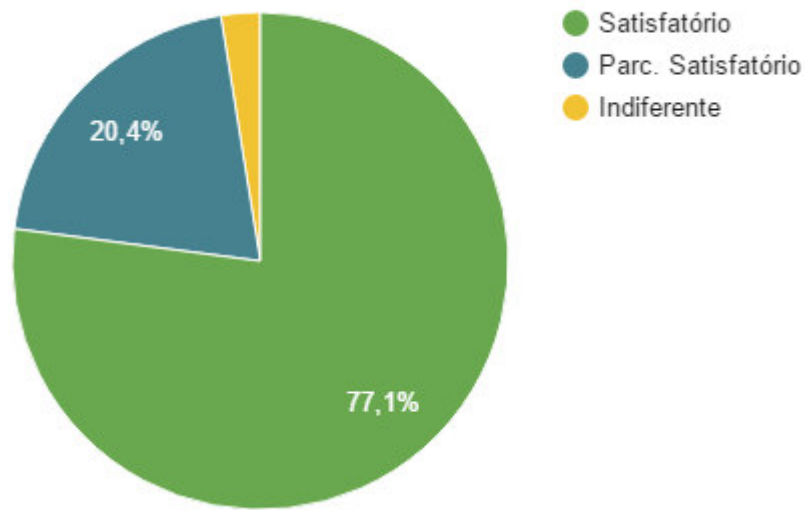
Tabela 3 – Contabilização de respostas por questões.

Questões	Total de Respostas	Satisfatórias	Parcialmente Satisfatórias	Indiferentes	Parcialmente Insatisfatórias	Insatisfatórias
1	9	8	1	0	0	0
2	10	9	1	0	0	0
3	10	9	1	0	0	0
4	9	5	4	0	0	0
5	10	8	2	0	0	0
6	10	8	2	0	0	0
7	9	5	3	1	0	0
8	10	9	1	0	0	0
9	10	5	4	1	0	0
10	10	7	2	1	0	0
11	10	8	1	1	0	0
12	10	8	2	0	0	0
13	10	7	3	0	0	0
14	10	9	1	0	0	0
15	10	6	4	0	0	0
16	10	10	0	0	0	0

Fonte: Elaborada pelo autor.

Analisando a ferramenta de uma forma geral. A **Figura 21** demonstra a porcentagem das respostas enviadas. É possível observar que das 157 respostas enviadas, 121 (71,1%) foram “satisfatórias”, 32 (20,4%) foram “parcialmente satisfatórias”, 4 (2,5%) foram “indiferentes”. As alternativas “parcialmente insatisfatórias” e “insatisfatórias” obtiveram 0% das respostas enviadas. É importante ressaltar que o gráfico não contempla as alternativas que não obtiveram respostas.

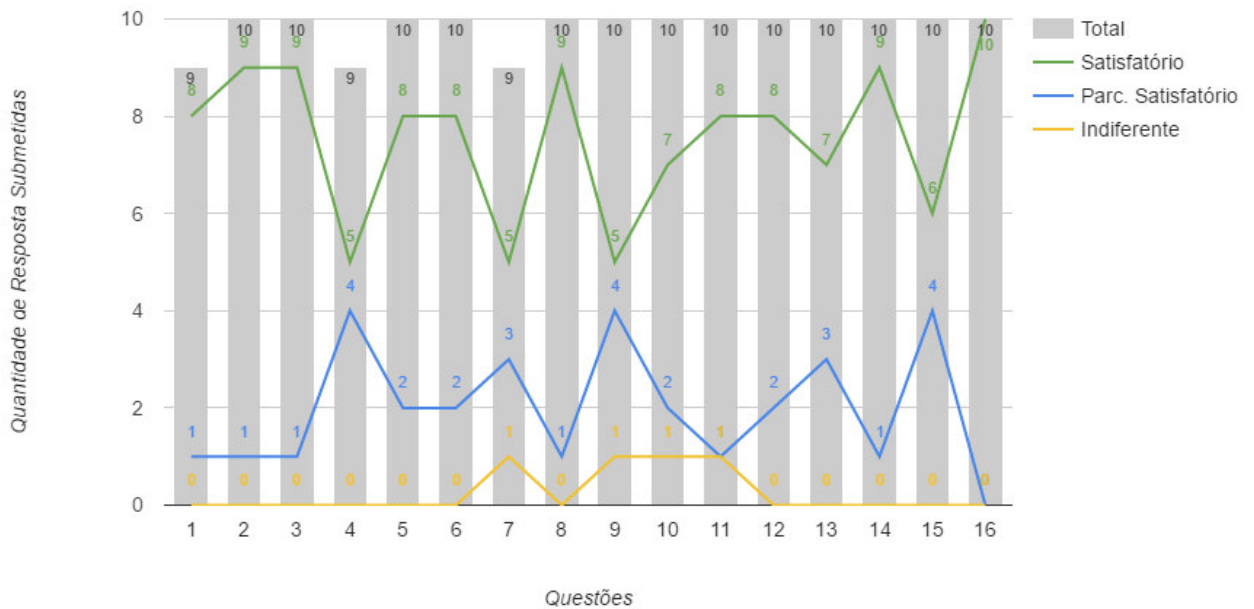
Figura 21 – Porcentagem geral da avaliação



Fonte: Elaborada pelo autor.

Para melhor detalhamento dos dados vistos na **Tabela 3**, a **Figura 22** mostra a quantidade de respostas que cada questão obteve. A figura também apresenta esses dados em relação a quantidade de respostas submetidas.

Figura 22 – Quantidade de respostas por questão



Fonte: Elaborada pelo autor.

É possível analisar na **Figura 22** que na questão 1, por exemplo, de um total de 9 questões submetidas, 8 foram “satisfatórias”, 1 foi “parcialmente satisfatória” e nenhuma

“indiferente”. Caso seja necessário a análise da porcentagem das respostas de cada questão, o **Apêndice C** apresenta os dados retirados diretamente da ferramenta *Google Forms*.

6.4 Considerações sobre os resultados

Após o levantamento dos dados pode-se observar que a grande maioria dos questionamentos sobre a ferramenta, obtiveram resultados satisfatórios e parcialmente satisfatórios. O que vale ressaltar que, como visto na **Figura 21** e de acordo com a descrição das cinco possíveis alternativas na seção **6.2**, 77,1% das questões respondidas pelos usuários atendem completamente os critérios questionados e 20,4% das respostas questionadas necessitam de algumas outras funcionalidades e aspectos, porém de baixa prioridade.

Apenas 2,5% das respostas questionadas foram julgadas pelos avaliadores como questões que faltam funcionalidades ou aspectos relevantes, mas ainda assim, é possível atingir os objetivos mais importantes da questão.

Estas observações levaram este projeto a ter como prioridade em seus trabalhos futuros, intervir nas questões com menos de 80% de satisfação e questões que receberam atribuições “indiferente”, ou seja, as questões 4, 7, 9, 10, 11, 13, 15. Para melhor detalhamento das questões e seus resultados, verificar o **Apêndice C**.

6.5 Trabalhos Futuros

Esta seção utiliza as considerações feitas na seção anterior e as sugestões repassadas pelos avaliadores para elaborar uma lista que contem as próximas atividades que darão continuidade a este projeto. A lista a seguir descreve atividades que coincidem exatamente com as questões em déficit na seção anterior.

1. Implementar a opção de confirmar ou cancelar uma ação de remoção.
2. Implementar no módulo usuário a funcionalidade para editar a senha.
3. Permitir outros tipos de filtro, por exemplo, filtrar as chamadas por data.
4. Mudar a data das chamadas para o padrão americano.
5. Ajustar o posicionamento dos botões.
6. Incluir um botão de ajuda para auxiliar na navegação.
7. Ajustar alguns tratamentos de falhas.

7 CONCLUSÃO

Este trabalho desenvolveu uma solução capaz de monitorar e controlar o serviço de voz sobre IP da Universidade Federal do Ceará - Campus Quixadá, provendo funcionalidades customizadas que facilitou a gestão do serviço pelos membros da coordenadoria de tecnologia da informação (CTI).

O trabalho inicia contextualizando a tecnologia VoIP, definindo os principais conceitos que a caracterizam, e segue demonstrando ferramentas capazes de prover desempenho e outros benefícios, diferenciando este trabalho de outras soluções já existentes no mercado. Ao fim da implementação da solução, este trabalho foi submetido a uma avaliação seguida de um questionário com o objetivo de apurar o nível de satisfação da solução.

O teste de avaliação foi realizado pelos membros da CTI, bolsistas da CTI e alguns professores do campus. Todos aqueles que submeteram o teste de avaliação, possuem conhecimento da tecnologia VoIP. Por fim, o trabalho culminou com resultados positivos e pôde ser concluído como satisfatório para os usuários da ferramenta.

REFERÊNCIAS

- AGUIAR, Otacílio dos Santos de. **Telefonia No Campus Da UFC Em Quixadá Com Voz Sobre IP**. 2013. 46 f. TCC (Graduação) - Curso de Redes de Computadores, Universidade Federal do Ceará, Quixadá, 2013.
- APOLINÁRIO, D. V. S.; BEZERRA, J. C.. **Comunicação VoIP utilizando o Elastix**. 2013. Curso de Ciências da Computação, Instituto de Estudos Superiores da Amazônia, Amazônia, 2013.
- BARRETO, Carlos Egberto Machado. **Utilização Da Tecnologia Voz Sobre IP Na Empresa Libra Ligas Do Brasil S/A**. 2013. 63 f. TCC (Graduação) - Curso de Redes de Computadores, Universidade Federal do Ceará, Quixadá, 2013.
- CALLADO, A. C.; ALMEIDA, G. F.; SILVA, A. M.; BARBOSA, R. S. B. G.; KELNER, J.; SADOK, D. F. H. **Construção de Redes de Voz sobre IP**. In.: Minicursos: 25º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Universidade Estadual do Pará (UEPA), 2007.
- COSTA, Daniel Gouveia. **Comunicações Multimídia na Internet**. Rio de Janeiro: Ciência Moderna Ltda., 2007.
- DTIDIGITAL. **ANGULARJS – UMA INTRODUÇÃO**. 2016. Disponível em: <<http://blog.dtidigital.com.br/angular-js-uma-introducao/>>. Acesso em: 28 abr. 2016.
- GOKHALE, Swapna S.; LU, Jijun. **Signaling Performance of SIP Based VoIP: A Measurement-Based Approach**. 2005. Curso de Computer Science, Department Of Computer Science And Engineering, University Of Connecticut, Storrs, 2005.
- INFOWORD. **MEAN vs. LAMP for the future of programming**. 2015. Disponível em: <<http://www.infoworld.com/article/2937159/application-development/mean-vs-lamp-your-next-programming-project.html>>. Acesso em: 23 mar. 2016.
- LEI, Kai; MA, Yining; TAN, Zhi. **Performance Comparison and Evaluation of Web Development Technologies in PHP, Python and Node.js**. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ENGINEERING, 17., 2014, Shenzhen. 2014. p. 1 - 8.
- MARTINS, Roger Beraldi. **O Futuro da Telefonia**. Paraná: Roger Beraldi Martins, 2006. 49 slides, color.
- NUNES, Fabio Pessoa; SAKURAY, Fabio; MENDES, Leonardo de Souza. **QVoip - Uma ferramenta para o monitoramento da qualidade de ligações VoIP utilizando o modelo-E**. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 26., 2008, Rio de Janeiro: Sbrt, 2008. p. 1 - 6.
- POULTER, Andrew John; JOHNSTO, Steven J.; COX, Simon J.. **Using the MEAN Stack to Implement a RESTful Service for an Internet of Things Application**. 2015. Curso de Faculty Of Engineering And The Environment, University Of Southampton, Southampton, 2015.

SILVA, João. **Ende-to-end javascript with the mean stack**. 2014. Disponível em: <<http://joaopsilva.github.io/talks/End-to-End-JavaScript-with-the-MEAN-Stack/#/introduction>>. Acesso em: 14 jun. 2016.

STRONGLOOP. **What Makes Node.js Faster Than Java?** 2014. Disponível em: <<https://strongloop.com/strongblog/node-js-is-faster-than-java/>>. Acesso em: 27 abr. 2016.

TRONCO, Tânia Regina. **Redes de Nova Geração**. 1. ed. São Paulo: Érica, 2010.

VALE, Eduardo Rodrigues, BRAJTERMAN, Moacyr. **Sistemas Modernos de Telecomunicações. Telefonia Avançada**. Rio de Janeiro, UNIVERSIDADE FEDERAL FLUMINENSE (UFF) e CENTRO DE ESTUDO DE PESSOAL (CEP), 2006.

APÊNDICES

APÊNDICE A – Questionário de Avaliação

1 – Em relação à interoperabilidade, a solução possui comunicação eficaz com serviço de voz e LDAP?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

2 – As funcionalidades apresentam entradas e saídas bem definidas?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

3 – A solução dispõe de segurança de acesso através de senhas?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

4 – A solução oferece de maneira eficaz monitoramento e controle do serviço de voz?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

5 – Em relação à usabilidade, as funcionalidades foram executadas facilmente?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

6 – A solução não apresentou falhas recorrentes em sua execução?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

7 – A solução reage adequadamente quando ocorrem falhas?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

8 – A solução informa ao usuário a entrada de dados inválidos?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

9 – O módulo de Usuários possui funcionalidades suficientes para suprir as necessidades dos usuários?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

10 – O módulo de Ramais possui funcionalidades suficientes para suprir as necessidades dos usuários?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

11 – O módulo de Chamadas possui funcionalidades suficientes para suprir as necessidades dos usuários?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

12 – As informações na tela são apresentadas de forma clara e agradável?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

13 – As telas são autoexplicativas, ou necessitam de ajuda para seu uso?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente

- Parcialmente Insatisfatório
- Insatisfatório

14 – A partir de uma tela qualquer você saberia navegar sem dificuldade para as demais?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

15 – Os botões e regiões clicáveis visualizados rapidamente por você?

- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

16 – As ações executadas por você chegaram ao resultado esperado?

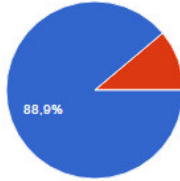
- Satisfatório
- Parcialmente Satisfatório
- Indiferente
- Parcialmente Insatisfatório
- Insatisfatório

APÊNDICE B – Roteiro da avaliação

- 1 – Logar-se com o usuário root, senha root.
- 2 - Vá em ramais (Peers) e adicione um ramal com as seguintes características:
 - Nome do Usuário (Username) = 322
 - Senha (Password) = 1234
 - Extensão (Extesion) = 322
 - Hopedeiro (Host) = dynamic
 - Tipo (Type) = friend
 - Contexto (Context) = e-admin
- 3 - Na lista de ramais (peers) busque o ramal 322.
- 4 – Observe a alteração de status quando, com um softphone, o aplicador da avaliação logar-se com o ramal 322.
- 5 - Delete o ramal 322.
- 6 - Vá em chamadas (Calls) e execute com um softphone uma chamada para o ramal 501.
- 7 - Em seguida, na lista de chamadas, selecione o registro e apague-o.
- 8 - Vá em usuários (Users) e adicione um usuário chamado “teste” de tipo “padrão” (stardard).
- 9 – De volta a lista de usuários. Altere seu nome para “teste1”.
- 10 - Saia da aplicação e faça login com o usuário “teste1”. Tente executar alguma ação.
- 11 – Saia novamente da aplicação e faça login no LDAP com o cpf “05064047304” e senha “12345”.

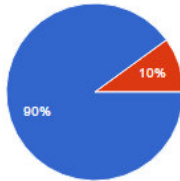
APÊNDICE C – Resultados do Google Forms

1 - Em relação à interoperabilidade, a solução possui comunicação eficaz com serviço de voz e LDAP ?



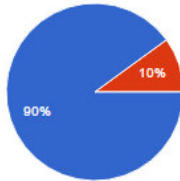
Satisfatório	8	88,9%
Parcialmente Satisfatório	1	11,1%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

2 - As funcionalidades apresentam entradas e saídas bem definidas ?



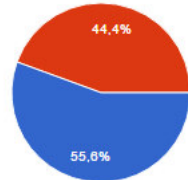
Satisfatório	9	90%
Parcialmente Satisfatório	1	10%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

3 - A solução dispõe de segurança de acesso através de senhas ?



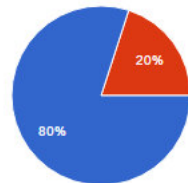
Satisfatório	9	90%
Parcialmente Satisfatório	1	10%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

4 - A solução oferece de maneira eficaz monitoramento e controle do serviço de voz ?



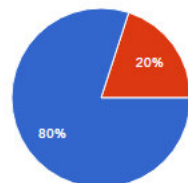
Satisfatório	5	55,6%
Parcialmente Satisfatório	4	44,4%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

5 - Em relação à usabilidade, as funcionalidades foram executadas facilmente ?



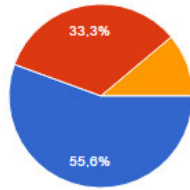
Satisfatório	8	80%
Parcialmente Satisfatório	2	20%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

6 - A solução não apresentou falhas recorrentes em sua execução ?



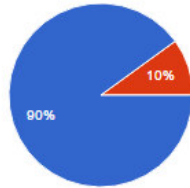
Satisfatório	8	80%
Parcialmente Satisfatório	2	20%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

7 - A solução reage adequadamente quando ocorrem falhas ?



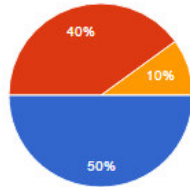
Satisfatório	5	55.6%
Parcialmente Satisfatório	3	33.3%
Indiferente	1	11.1%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

8 - A solução informa ao usuário a entrada de dados inválidos ?



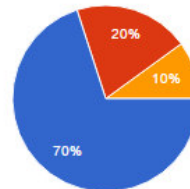
Satisfatório	9	90%
Parcialmente Satisfatório	1	10%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

9 - O módulo de Usuários possui funcionalidades suficientes para suprir as necessidades dos usuários ?



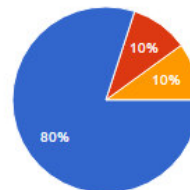
Satisfatório	5	50%
Parcialmente Satisfatório	4	40%
Indiferente	1	10%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

10 - O módulo de Ramais possui funcionalidades suficientes para suprir as necessidades dos usuários ?



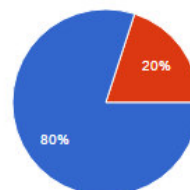
Satisfatório	7	70%
Parcialmente Satisfatório	2	20%
Indiferente	1	10%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

11 - O módulo de Chamadas possui funcionalidades suficientes para suprir as necessidades dos usuários ?



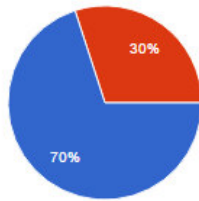
Satisfatório	8	80%
Parcialmente Satisfatório	1	10%
Indiferente	1	10%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

12 - As informações na tela são apresentadas de forma clara e agradável ?



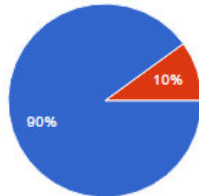
Satisfatório	8	80%
Parcialmente Satisfatório	2	20%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

13 - As telas são autoexplicativas, ou necessitam de ajuda para seu uso ?



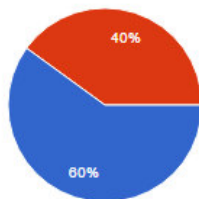
Satisfatório	7	70%
Parcialmente Satisfatório	3	30%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

14 - A partir de uma tela qualquer você saberia navegar sem dificuldade para as demais ?



Satisfatório	9	90%
Parcialmente Satisfatório	1	10%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

15 - Os botões e regiões clicáveis visualizados rapidamente por você ?



Satisfatório	6	60%
Parcialmente Satisfatório	4	40%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%

16 - As ações executadas por você chegaram ao resultado esperado ?



Satisfatório	10	100%
Parcialmente Satisfatório	0	0%
Indiferente	0	0%
Parcialmente Insatisfatório	0	0%
Insatisfatório	0	0%