

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

FLÁVIO SANTA ROSA CORADINI

**IMPLEMENTAÇÃO DE UM AGENTE JADE GRAVADOR DE MÍDIA EM
UMA ARQUITETURA OSGI/JADE UTILIZANDO O NÚCLEO IMS**

VITÓRIA

2017

FLÁVIO SANTA ROSA CORADINI

**IMPLEMENTAÇÃO DE UM AGENTE JADE GRAVADOR DE MÍDIA EM
UMA ARQUITETURA OSGI/JADE UTILIZANDO O NÚCLEO IMS**

Monografia apresentada ao Curso de Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.

Orientadora: Prof. Dra. Rosane Bodart Soares

VITÓRIA

2017

FLÁVIO SANTA ROSA CORADINI

**IMPLEMENTAÇÃO DE UM AGENTE JADE GRAVADOR DE MÍDIA EM
UMA ARQUITETURA OSGI/JADE UTILIZANDO O NÚCLEO IMS**

Monografia submetida ao Curso de Engenharia Elétrica da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Bacharel em Engenharia Elétrica.

Aprovada em 22 de Fevereiro de 2017.

COMISSÃO EXAMINADORA

Prof. Dra. Rosane Bodart Soares
Universidade Federal do Espírito Santo

Prof. Renato Benezath Cabelino Ribeiro
Instituto Federal do Espírito Santo

Prof. Thales de Oliveira Gonçalves
Universidade Federal do Espírito Santo

DEDICATÓRIA

A todos que me apoiaram e me incentivaram nessa longa jornada.

AGRADECIMENTOS

Agradeço primeiramente à minha família, que me deu oportunidade de estudo e incentivo para seguir meu caminho, além de toda a paciência e companheirismo nos momentos difíceis.

À professora Rosane Bodart, pela paciência e compreensão durante todo o período de orientação, além de confiar e aceitar ser orientadora deste trabalho.

Ao Renato Cabelino que me orientou bastante desde a apresentação do projeto por ele desenvolvido até na definição das diretrizes do presente trabalho.

À Juliana de Andrade pela paciência e disponibilidade ao me apresentar todos os componentes por mim usados nesse projeto, sendo esses fundamentais para o desenvolvimento do trabalho.

Ao Thales Gonçalves pela participação no projeto, pelas opiniões sinceras e pela amizade.

Aos companheiros de graduação Leonardo de Assis, Abdo Emanuel, Leonardo Boldrini, Airton, e muitos outros, por terem tornado a graduação uma experiência que transcendeu o enriquecimento acadêmico, e me ensinou lições que levarei por toda vida.

RESUMO

Nos últimos anos, os sistemas de telecomunicação vêm em constante evolução buscando sempre minimizar custos e agregar serviços. Nesse contexto, cada empresa se depara com um cenário altamente competitivo onde ela deve se posicionar de forma estratégica para lidar com seus concorrentes. Assim, visando atrair novos clientes, as corporativas do setor vem introduzindo novos serviços para a próxima geração de rede (*Next Generation Network - NGN*), que por sua vez migram a antiga conjuntura verticalizada dos serviços de telecomunicação, onde os sistemas eram isolados e redundantes, para um novo modelo, sendo esse mais simplificado dada a modularização das atividades oferecendo assim conteúdos e serviços a todos os meios de acesso.

Nesse contexto, diante dos desafios impostos pelo novo modelo de provimento de serviços para a NGN, foi proposto por Ribeiro (2015) uma arquitetura orientada à serviços denominada SANGN (*Service Architecture on Next-Generation Networks*), sendo essa adequada para hospedagem de serviços de telecomunicação utilizando arquitetura OSGi/JADE. Para avaliá-la foi desenvolvido por ele um serviço de Televoto com a finalidade de agregar serviços de voz às redes de próxima geração. Tal trabalho foi aperfeiçoado por Andrade (2016) através da adição de outras funcionalidades, aumentando assim a confiabilidade da arquitetura. Todavia, serviços como a gravação do conteúdo de mídia trafegados na rede, exigidos pela legislação brasileira, não foram contemplados.

Baseado nisso o presente trabalho visa desenvolver um módulo adicional ao serviço do Televoto, um gravador de chamadas SIP. Sua finalidade, bem como seu próprio nome diz, é gravar as chamadas realizadas na plataforma além de fornecer uma interface para consultas referentes à essas ligações.

LISTA DE FIGURAS

Figura 1: Contéudo base para o desenvolvimento do projeto	18
Figura 2: Arquitetura OSGi (Ribeiro, 2015)	21
Figura 3: Padronização FIPA para tecnologias genéricas de agentes (Ribeiro, 2015)	23
Figura 4: Convergência de redes, serviços e dispositivos provida pelo IMS (Ribeiro, 2015)	26
Figura 5: Arquitetura do IMS em camadas (Ribeiro, 2015)	27
Figura 6: Agentes de usuário na arquitetura SIP (Ribeiro, 2015)	31
Figura 7: Arquitetura B2BUA (Ribeiro, 2015)	32
Figura 8: Cabeçalhos presentes no pacote RTP (Ribeiro, 2015)	33
Figura 9: Registro de usuário no IMS (Ribeiro, 2015)	36
Figura 10: Estabelecimento da sessão no IMS (Ribeiro, 2015).....	41
Figura 11: Amostragem de um sinal analógico (CAMARILLO, 2006)	44
Figura 12: Quantização de um intervalo de amplitude (CAMARILLO, 2006)	45
Figura 13: Justificativa do conteúdo base para o desenvolvimento do projeto	48
Figura 14: Ambiente no qual o gravador será inserido.....	51
Figura 15: Máquina de estados dos agentes JADE trabalhados.....	54
Figura 16: Esquema sobre o uso do agente gravador	56
Figura 17: Processo de gerenciamento de portas entre agente e servidor	59
Figura 18: Interface do usuário do cliente web (chamadas)	60
Figura 19: Interface do usuário do cliente web (portas)	60

Figura 20: Elementos presentes no estudo de caso	61
Figura 21: Arquitetura do estudo de caso	64
Figura 22: Interface do JADE Remote Agent Management	66
Figura 23: Etapas do processo de gravação	66
Figura 24: Sinais de áudio enviado e recebido, respectivamente	67
Figura 25: Tráfego SIP na arquitetura IMS.....	68
Figura 26: Corpo da mensagem SIP INVITE.....	69
Figura 27: Arquitetura desenvolvido por Andrade (2016).....	72
Figura 28: Inserindo o gravador SIP na arquitetura desenvolvida por Andrade (2016)	73

LISTA DE TABELAS

Tabela 1: Métodos das mensagens do protocolo SIP	30
Tabela 2: Respostas do protocolo SIP	30
Tabela 3: Detalhamento das mensagens no registro de usuário no IMS	36
Tabela 4: Parâmetros de um desafio DIGEST	37
Tabela 5: Parâmetros de uma resposta à um desafio DISGEST	39
Tabela 6: Detalhamento das mensagens trocadas no estabelecimento da sessão no IMS.....	42
Tabela 7: Funcionalidades disponibilizadas pelo serviço Web.....	58
Tabela 8: URI dos componentes trabalhados	62
Tabela 9: DNS dos componentes do OpenIMS.....	62
Tabela 10: Parâmetros da sessão RTP	65

LISTA DE SIGLAS/ACRÔNIMOS

3GPP - *3rd Generation Partnership Project*

API - *Application Programming Interface*

AS – *Application Server*

B2BUA – *Back to Back User Agent*

CODEC - *Codificador/Decodificador*

CPU – *Central Process Unit*

CSCFs – *Call Session Control Functions*

DNS – *Domain Name System*

FIPA - *Foundation for Intelligent Physical Agents*

HSS – *Home Subscriber Service*

HTTP – *Hypertext Transfer Protocol*

I-CSCF - *Integgorating-CSCF*

IMS - *IP Multimedia Subsystem*

IN - *Intelligent Networks*

IP – *Internet Protocol*

JSON - *JavaScript Object Notation*

JVM – *JAVA Virtual Machine*

LAN – *Local Area Network*

NGN – *Next Generation Network*

NGN – *Next Generation Network*

OSI - *Open Systems Interconnection*

PABX – *Private Automatic Branch Exchange*

P-CSCF – *Proxy-CSCF*

QoS – *Quality of Service*

REST – *Representation State Transfer*

RTCP – *Real Time Control Protocol*

RTP – *Real Time Protocol*

S-CSCF - *Serving-CSCF*

SDP – *Session Description Protocol*

SIP - *Session Initiation Protocol*

SOA – *Service Oriented Architecture*

TCP - *Transmission Control Protocol*

UA – *User Agent*

UAC – *User Agent Client*

UAS – *User Agent Server*

UDP - *User Datagram Protocol*

URI - *Uniform Resource Identifier*

URL - *Uniform Resource Locator*

VM – *Virtual Machine*

VOIP – *Voice over IP*

W3C – *World Wide Web Consortium*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.2	METODOLOGIA	16
1.3	ESTRUTURA DA MONOGRAFIA	17
2	EMBASAMENTO TEÓRICO	18
2.1	ARQUITETURA ORIENTADA À SERVIÇOS	19
2.1.1	<i>Framework OSGi</i>	20
2.2	ARQUITETURA ORIENTADA À AGENTES	22
2.2.1	<i>Java Agent Development Framework (JADE)</i>	22
2.3	REDES DE PRÓXIMA GERAÇÃO (NGN).....	24
2.3.1	Subsistema Multimídia IP (IMS).....	25
2.4.1	Codificação (<i>Encoding</i>).....	43
2.5	SERVIÇO WEB - RESTFUL.....	46
2.6	LEGISLAÇÃO	47
2.7	CONSIDERAÇÕES DO CAPÍTULO	47
3	GRAVADOR SIP	49
3.1	AMBIENTE TRABALHADO	50
3.2	REQUISITOS E FUNCIONALIDADES	51
3.3	MÁQUINA DE ESTADO.....	52
3.4	COMPONENTES DESENVOLVIDOS	55
3.4.1	Record Agent.....	55
3.4.2	RTPExport	57
3.4.3	Serviço Web	57
3.4.4	Cliente Web	59
3.5	ESTUDO DE CASO.....	61
3.5.1	Componentes trabalhados.....	61
3.5.2	Arquitetura elaborada para o estudo de caso	63
3.5.3	Parametrização da sessão RTP	64

3.5.4	Inicialização dos agentes.....	65
3.5.5	Processo de gravação	66
3.5.6	Monitoramento da sessão SIP	68
3.5.7	Desafios e soluções encontradas	69
4	TRABALHOS FUTUROS	72
5	CONCLUSÃO.....	74
	BIBLIOGRAFIA	75

1 INTRODUÇÃO

Nos últimos anos, os sistemas de telecomunicações vêm em constante evolução buscando sempre minimizar custos e agregar serviços. Nesse contexto cada empresa se depara com um cenário altamente competitivo onde ela deve se posicionar de forma estratégica para lidar com seus concorrentes. Assim, visando atrair novos clientes, as corporativas do setor vêm introduzindo novos serviços para a próxima geração de rede (*Next Generation Network - NGN*), que por sua vez migram a antiga conjuntura verticalizada dos serviços de telecomunicação, onde os sistemas eram isolados e redundantes, para um novo modelo, sendo esse mais simplificado dada a modularização das atividades oferecendo assim conteúdos e serviços à todos os meios de acesso.

Essa evolução culmina em uma convergência dos sistemas de telefonia fixa e móvel. Atualmente, o primeiro é ofertado por uma provedora através de serviços de voz junto à um plano de dados para acesso à internet, enquanto o segundo possui as mesmas características dada as similaridades dos terminais de acesso que são constantemente ligados a internet e são usados para troca de voz e mídia.

Nesse âmbito, surgiu um novo conceito denominado redes e serviços convergentes. A convergência é caracterizada pela capacidade da rede de suportar tráfegos oriundos de diferentes fontes de forma transparente, ou seja, unindo definitivamente os sistemas de voz e dados.

Baseado nos novos requisitos na qual está submetida a infraestrutura de telecomunicação, a mesma deve se adequar à essa nova realidade. Uma solução para esse cenário está no uso da arquitetura IMS (*IP Multimedia Subsystem*). Basicamente o IMS é o conjunto de especificações e requisitos que definem uma arquitetura unificada para serviços baseados em IP, permitindo assim a convergência das tecnologias de dados, voz e rede sobre a mesma infraestrutura.

Tal arquitetura traz diversas vantagens aos grupos que a utilizam sejam eles os provedores dos serviços, os operadores de rede ou os próprios clientes. O primeiro grupo garante uma maior portabilidade, mobilidade e flexibilidade para implantar

novos serviços bem como a possibilidade de aumentar o valor agregado dos serviços prestados. O segundo grupo por sua vez passa a ser capaz de aumentar sua autonomia na implementação, manutenção e escalabilidade dos serviços. Por fim, o terceiro grupo, passa a ter mais simplicidade na personalização dos serviços contratados junto as operadoras, aliado à uma maior qualidade e disponibilidade dos mesmos.

Baseado no atual modelo de provimento de serviços para as redes de próxima geração e nos desafios dessa nova abordagem, Ribeiro (2015) propôs a arquitetura SANGN (*Service Architecture on Next-Generation Networks*), sendo essa desenvolvida baseada no *framework* orientado à serviços OSGi em conjuntura com a arquitetura JADE. A união dessas duas tecnologias viabiliza a aplicação dos conceitos estabelecidos pelo IMS tais como a escalabilidade e a convergência pois viabiliza o provimento de diferentes serviços de forma modularizada.

A arquitetura SANGN pôde ser avaliada por meio do desenvolvimento de um serviço de Televoto. Essa escolha teve como finalidade mensurar se os requisitos propostos pela arquitetura IMS podem ser atendidos através da utilização do *framework* OSGi junto ao ambiente de agentes JADE. Apesar do rico trabalho desenvolvido e aspectos como escalabilidade horizontal serem atendidos em conforme as exigências da NGN, o serviço do Televoto apresentou algumas limitações, dentre elas: perda de parte das chamadas, ausência de uma unidade centralizadora das mesmas e um serviço para prover agentes atendedores sob demanda.

Baseado na arquitetura SANGN desenvolvida junto ao serviço do Televoto, Andrade (2016) modificou-a corrigindo as antigas limitações supracitadas viabilizando o provimento de um serviço mais robusto e confiável principalmente em relação ao atendimento das chamadas. A implementação do sistema de Televoto na arquitetura SANGN foi além dos requisitos propostos e se mostrou capaz de ser ainda mais escalonável dada sua estrutura horizontal e modular.

Nesse contexto de provimento de serviços para as redes de próxima geração, um outro aspecto a ser abordado além das questões de tecnológicas é a legislação brasileira. Essa, por sua vez, determina uma série de exigências para as empresas prestadoras de serviços nessa área. De acordo com a Lei do Marco Civil da Internet,

todas as empresas que atuam nesse âmbito devem ter registrado as trocas de informações entre seus usuários por um período mínimo de seis meses e sejam capazes de fornecê-las mediante ordem judicial (BRASIL, 2014).

Baseado nisso o presente trabalho visa desenvolver um módulo adicional ao serviço do Televoto, um gravador de chamadas SIP. Sua finalidade, bem como seu próprio nome diz, é gravar as chamadas realizadas na plataforma além de fornecer uma interface para consultas referentes à essas ligações.

1.1 OBJETIVOS

O presente trabalho tem como objetivo viabilizar a gravação das chamadas SIP executadas dentro da infraestrutura do serviço do Televoto. Nesse intuito, o gravador será desenvolvido como um módulo adicional ao serviço do Televoto e sua finalidade será gravar as chamadas realizadas na plataforma SANGN além de registrar as informações relevantes de cada uma delas em uma base de dados. Além disso, em suporte a tal componente, serão desenvolvidos outros dois fundamentais para o pleno funcionamento e compreensão das ações de gravação, são eles: um serviço Web (RESTful), responsável por expor as funcionalidades de gerenciamento de portas e consulta de chamadas, e um cliente Web, cuja finalidade se resume a exibir, de forma simplificada, todas as chamadas executadas na plataforma e disponibilizá-las para consulta. O gravador utilizará a *framework* orientada à agentes JADE visando facilitar sua posterior integração ao serviço do Televoto enquanto os demais componentes serão baseados nas tecnologias *Node.js* e *Express.js*, sendo essas extremamente eficientes para o desenvolvimento de aplicações orientadas à Web.

1.2 METODOLOGIA

Inicialmente foi realizada uma profunda revisão da literatura abordando os assuntos relevantes para o trabalho. Assim foi possível obter uma rica fundamentação teórica baseada em conceitos como a arquitetura IMS, OSGi, *framework* orientada à agentes JADE além dos protocolos SIP e RTP. Tais protocolos necessitaram uma maior atenção dada a necessidade de o gravador respeitar veemente as regras impostas por cada um deles. Além disso, foram necessários estudos de tecnologias como *Node.js* e *Express.js* visando desenvolver o cliente Web a ser usado na consulta das gravações realizadas e um serviço Web para expor a funcionalidades de consulta e gerenciamento de portas.

Em seguida o foco do trabalho se voltou para o entendimento da arquitetura SANGN. Como gravador SIP desenvolvido irá ser integrado a ela, um estudo detalhado sobre seu funcionamento se mostrou essencial. Essa abordagem viabilizou decisões de projeto fundamentais desde o contexto a ser inserido o gravador bem como as tecnologias envolvidas para seu desenvolvimento.

Assim, com a compreensão do embasamento teórico necessário, o gravador pôde ser desenvolvido sendo o principal objetivo nesse processo garantir que todos os requisitos fossem atendidos que vão desde a gravação integral da chamada como a posterior consulta de todo conteúdo envolvido via cliente Web.

Todo o trabalho pôde ser avaliado via realização de inúmeras chamadas entre diferentes clientes garantido assim a consistência da plataforma. Por fim, foram feitas análises referentes às futuras aplicações que poderão ser beneficiadas com o gravador SIP desenvolvido.

1.3 ESTRUTURA DA MONOGRAFIA

O presente projeto foi dividido em 5 capítulos visando organizar e simplificar o assunto a ser apresentado. O capítulo 2 irá apresentar o referencial teórico utilizado como base para compreensão do contexto e das possibilidades disponíveis para o desenvolvimento do gravador SIP. O capítulo 3 detalha como os componentes do gravador foram trabalhados baseados no ambiente onde o módulo gravador será inserido e nas características das mensagens trocadas entre diferentes terminais conectados à uma rede IMS. Além disso, será apresentado um estudo de caso mostrando na prática o funcionamento dos componentes desenvolvidos. O capítulo 4 detalha os trabalhos futuros que poderão vir a se beneficiar das funcionalidades do gravador SIP e como esse acoplamento deve ser realizado. Por fim, o capítulo 5 irá sumarizar tudo que foi apresentado no projeto e apresentar as principais conclusões referentes ao trabalho.

2 EMBASAMENTO TEÓRICO

A desenvolvimento do gravador SIP pode ser tratado como uma atividade teoricamente simples, mas, por estar inserido em uma arquitetura IMS, o contexto trabalhado se mostra extremamente complexo. Dominar esse cenário se baseia na compreensão de inúmeros conceitos como as redes de próxima geração (NGN), subsistema multimídia IMS, protocolos SIP e RTP, *framework* OSGi, arquitetura orientada à agentes JADE, dentre outros. Além disso, diversos temas oriundos do âmbito de Web devem ser abordados para o pleno entendimento do serviço e cliente Web a serem desenvolvidos. A Figura 1 exemplifica a hierarquia e o conteúdo a ser abordado. Visando melhorar a compreensão do trabalho, este capítulo traz detalhadamente cada tópico supracitado.

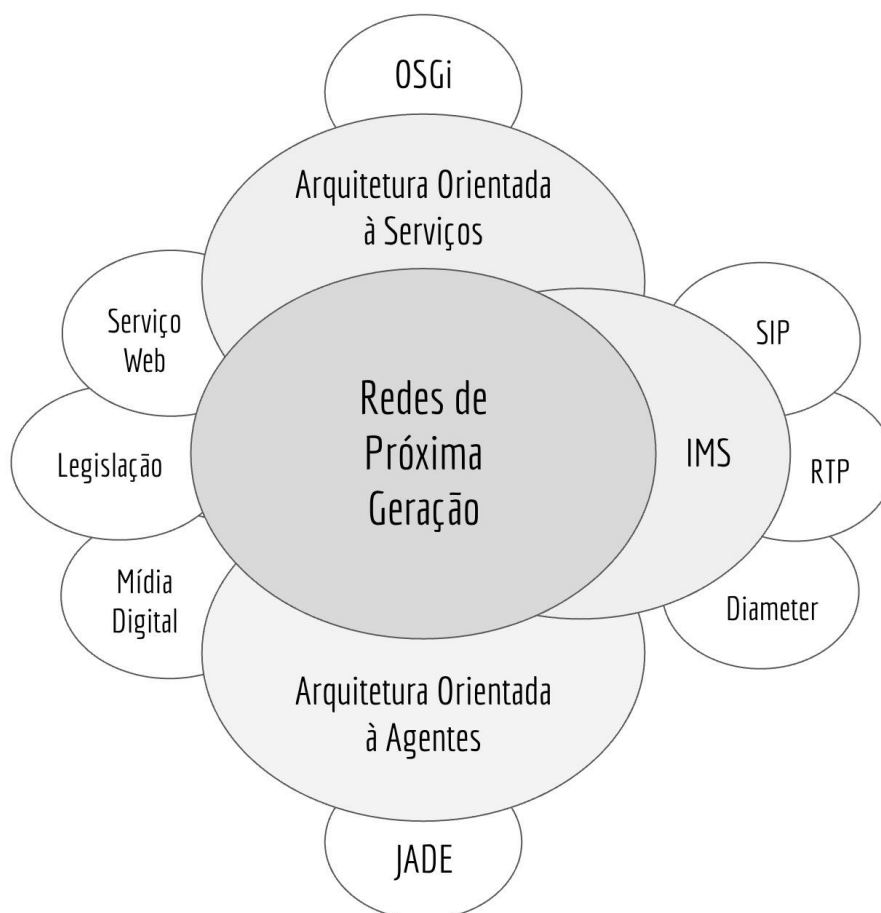


Figura 1: Conteúdo base para o desenvolvimento do projeto

2.1 ARQUITETURA ORIENTADA À SERVIÇOS

Atualmente, independente do assunto, a crescente complexidade para a resolução de problemas é algo recorrente dada a vasta abrangência em cada abordagem, logo, em telecomunicação não é diferente. Nesse nicho, como as possibilidades são muito diversas e novos desafios surgem a cada instante, a resolução de cada um desses pode ser alcançada por meio do uso de serviços. Dado o alto número de finalidades, eles devem trabalhar de forma independente e integrada visando suportar os requisitos da infraestrutura. A arquitetura orientada à serviços (SOA) é um padrão de projeto que promove a modularização e escalabilidade entre serviços distintos caracterizando assim um baixo acoplamento entre as partes além desses serem intercambiáveis (JUNIOR, 2007).

Uma das características de suma importância para a arquitetura SOA é a independência e escalabilidade dos serviços. Essa, por sua vez, simplifica a reutilização desses poupando tempo aos projetistas durante o desenvolvimento de novas aplicações. Outro importante aspecto consiste na abstração das funcionalidades de cada serviço por meio de interfaces, bastando ao projetista um conhecimento básico sobre o mesmo para utilizá-lo na prática. Segundo Graham (2008), SOA consiste no incremento do nível de abstração de modo que os requisitos e processos de negócios possam ser discutidos em uma linguagem compreendida por pessoas de diferentes áreas sejam elas profissionais de TI ou do universo dos negócios.

No mundo dos negócios a flexibilidade provida por uma arquitetura orientada à serviços se mostra fundamental. A capacidade de determinar diferentes soluções para as necessidades de um negócio justifica a utilização da SOA. Segundo Bieberstein (2006), a flexibilidade em lidar com novos cenários e a capacidade de recombinar e integrar serviços é o que fornece um estreitamento entre os negócios e a área de TI.

Visando trabalhar junto à uma arquitetura orientada à serviços, a SANGN, existe uma tecnologia cujo estudo é justificado pois essa é capaz de implementar as características supracitadas, a OSGi (*Open Source Gateway Initiative*).

2.1.1 Framework OSGi

OSGi (*Open Source Gateway Initiative*) é um conjunto de especificações que compõem uma plataforma dinâmica modularizada para plataforma Java. Tais especificações visam reduzir a complexidade dos desenvolvedores dada sua arquitetura orientada à serviços (SOA). Segundo a responsável por controlar essas padronizações, a OSGi Alliance (2009), o OSGi deve ser considerado um *middleware* universal.

Uma das principais características do OSGi é a flexibilidade, pois, dada a capacidade da adição ou remoção de módulos JAVA em tempo de execução, o desenvolvedor simplifica a coesão entre as partes do sistema. Além disso, todos os módulos desenvolvidos podem colaborar entre si dentro de uma única máquina virtual java (JVM).

O uso do OSGi para o desenvolvimento de plataformas como a SANGN é justificado pelo fato de ser uma arquitetura padronizada e aberta. Isso proporciona aos desenvolvedores, e provedores de serviços a capacidade de gerenciar e desenvolver novas ferramentas de uma forma mais prática e simples (Turner, 2004).

O processo de construção de aplicações com essa tecnologia envolve o desenvolvimento e composição de módulos (*bundles*) de serviços específicos junto à meta-arquivos, responsáveis por descreverem os provimentos e dependências que devem ser instanciados para plena execução dos componentes. Além disso, para que cada serviço esteja disponível, cada um possui um mecanismo de interface JAVA no intuito de se registrarem localmente. Esse modelo de controle centralizado dos módulos permite que os mesmos determinem suas próprias dependências de serviços.

Basicamente, a plataforma OSGi é composta por 5 camadas distintas, são elas: *bundles*, serviços, ciclo de vida, módulos e segurança. A disposição dessas pode ser melhor compreendida vide Figura 2 e, conforme OSGi Alliance (2016), a atribuição de cada uma pode ser compreendida como segue:

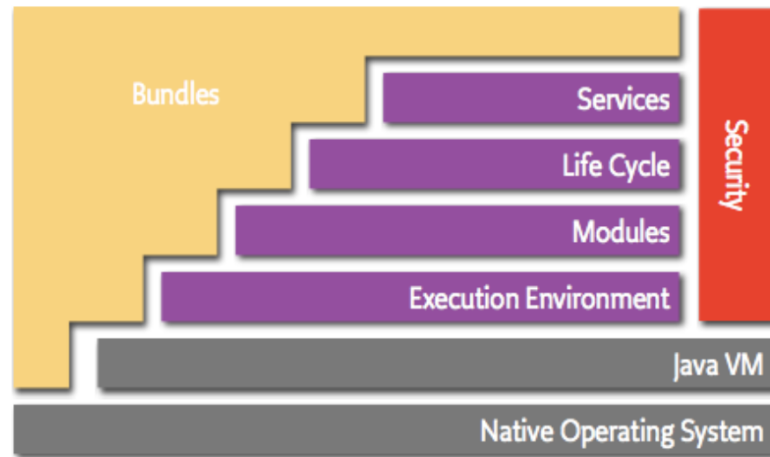


Figura 2: Arquitetura OSGi (Ribeiro, 2015)

- **Camada de *bundles*:** Define um esquema para implementação e manutenção dos *bundles*. Nessa camada, cada *bundle* é capaz de executar diversas ações, como por exemplo registrar e pesquisar serviços existentes visando se adaptar as necessidades dos dispositivos que fazem seu uso além de receber notificações sobre o estado de cada um deles.
- **Camada de serviços:** Sua principal atribuição é simplificar o desenvolvimento e distribuição dos *bundles*. Essa característica é obtida através da separação das especificações de cada serviço e sua respectiva implementação. Assim cada desenvolvedor pode, por meio das especificações de cada serviço, conectá-los de forma prática e transparente.
- **Camada de segurança:** Responsável por garantir a segurança da plataforma, minimizando as vulnerabilidades da mesma.
- **Camada de módulos:** Responsável por estabelecer as regras nas quais os pacotes JAVA e os *bundles* estarão submetidos. Assim, as relações de dependência e até mesmo ocultação entre esses pacotes e os *bundles* podem ser definidas. Além disso, essa camada é capaz de operar de forma independente das de serviço e ciclo de vida.

- **Camada de ciclo de vida:** Todo *bundle* inserido na plataforma OSGi possui um ciclo de vida que pode ser definido através dos seguintes estados: ativo, parado, inicializado, resolvido e instalado. Essa camada é responsável por atribuir um ciclo de vida a cada *bundle*. Além disso, essa fornece uma API capaz de gerir as operações de cada serviço na plataforma baseado nos estados.

2.2 ARQUITETURA ORIENTADA À AGENTES

Nos últimos tempos, muitas empresas têm valorizado o uso de sistemas de software orientado à agentes por meio da utilização de ferramentas e metodologias no intuito de auxiliar o desenvolvimento de aplicações mais modulares e coesas. Dentre essas ferramentas está o JADE (*Java Agent Development Framework*), um *framework*, desenvolvido em Java, para desenvolvimento de aplicações orientadas à agentes.

Para a execução do presente trabalho compreender o funcionamento e o uso de agentes JADE é fundamental, pois, além desse embasamento ser necessário para o entendimento da arquitetura SANGN, o gravador SIP desenvolvido será codificado como um agente JADE.

2.2.1 *Java Agent Development Framework (JADE)*

O *framework* JADE foi inicialmente desenvolvida pela Telecom Itália em parceria com a Universidade de Parma seguindo o padrão FIPA (*Foundation For Intelligent Physical Agents*). Atualmente o projeto é *open source* com licença LGPL (*Lesser General Public License*) (LGPL, 2007).

A FIPA é uma associação internacional de várias companhias que trabalham para especificar tecnologias genéricas de agentes. O principal objeto da associação é promover, utilizando um conjunto de tecnologias distintas, sistemas complexos com grande interoperabilidade. Visando a operação e gerenciamento de agentes, a FIPA

define um modelo que atende esses requisitos (TEIXEIRA, s.d.), sendo esse descrito conforme Figura 3.

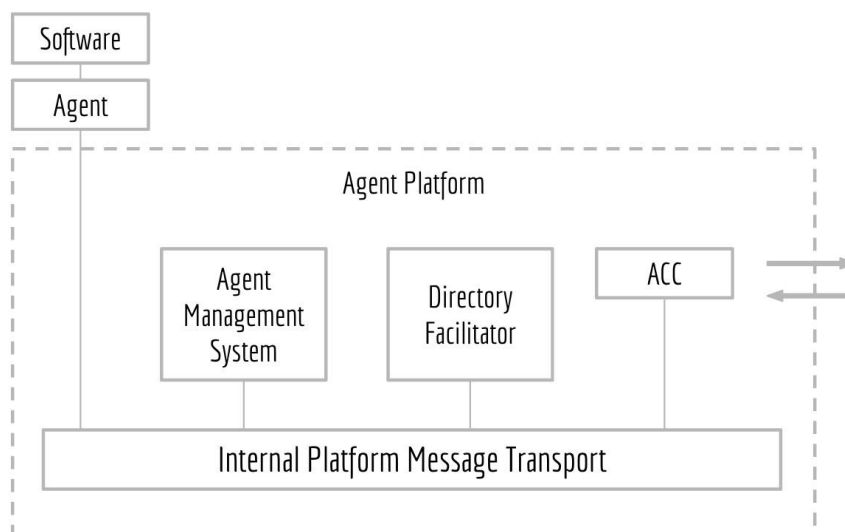


Figura 3: Padronização FIPA para tecnologias genéricas de agentes (Ribeiro, 2015)

Basicamente o modelo é composto por três principais elementos, são eles: AMS (*Agent Management System*), DF (*Directory Facilitator*) e ACC (*Agent Communication Channel*). O primeiro é responsável por gerenciar o ciclo de vida dos agentes e gerenciar a plataforma. O segundo tem como finalidade fornecer um diretório que define os agentes disponíveis na plataforma. Por fim, o ACC gerencia toda a comunicação dos agentes internamente e externamente à plataforma.

Segundo Bellifemine (2007), baseado no modelo apresentado, responsável por definir o paradigma de agentes, o JADE pode ser classificado como um *middleware* para desenvolvimento e execução de aplicações P2P (*Peer-to-Peer*).

Dentre as principais características dessa arquitetura, existe a capacidade de gerenciamento de ciclo de vida de cada agente, bem como a abstração de implementação por meio do uso de classes com a finalidade de criar novos agentes e comportamentos, envio de mensagem entre os elementos instanciados, dentre outras.

O gravador SIP a ser desenvolvido irá seguir todas as regras da arquitetura JADE, pois o mesmo será codificado como um agente, facilitando assim sua posterior integração à arquitetura SANGN.

2.3 REDES DE PRÓXIMA GERAÇÃO (NGN)

Nos últimos anos as redes de telefonia fixas e móveis passaram por grandes mudanças. Em meados dos anos 80, o primeiro robusto sistema de telecomunicação foi introduzido. Esse, denominado primeira geração (1G), se limitava a fornecer serviços básicos cuja ênfase estava naqueles associados à voz. Na década seguinte, o sistema evoluiu do 1G para o 2G. As principais melhorias com tal transição foram a introdução de serviços de dados além de uma melhor experiência para o usuário. A terceira geração (3G) junto a sua evolução, o LTE (*Long Term Evolution*), permitiram um incremento significativo na taxa de dados bem como a adição de vários serviços multimídia. Atualmente estamos na quarta geração, o 4G, sendo essa, além de mais veloz, totalmente baseado em IP (*Internet Protocol*). Essa característica em particular trouxe uma ampla flexibilidade para os serviços prestados, independentemente do tipo. Nesse contexto de constante evolução é esperado que, com a quinta geração (5G), as taxas de transmissão sejam ampliadas ainda mais junto ao desenvolvimento de uma infraestrutura capaz de lidar de forma transparente com a internet das coisas (*Internet of Things - IoT*). Além disso, essa evolução culmina em uma convergência dos sistemas de telefonia fixa e móvel, pois devido as similaridade dos terminais de acesso por estarem constantemente ligados (*always-on*) e conectados à internet (*always-connected*) e serem utilizados para troca de voz e mídia, tal unificação visa simplificar e otimizar o sistema como um todo (POIKSELKA, 2006).

Atualmente, o crescente número de usuários junto à suas novas exigências por serviços inovadores, as desregulamentações dos mercados e a falta de padronização das atividades do setor, impulsionaram a criação de um novo conceito, as Redes de Próxima Geração (*Next Generation Networks - NGN*).

A NGN é muito mais que um simples conceito. Ela abrange diversas questões técnicas e econômicas. Tecnicamente, a flexibilidade e autonomia trazida por esse conceito aos profissionais do setor de telecomunicação permite a introdução de novos serviços de forma simples e prática. Economicamente, além de reduzir o tempo de trabalho gasto em ações de manutenção ou atualização do sistema instaurado, a NGN torna possível personalizar serviços de voz e dados de acordo com as exigências dos usuários, fato que muitas vezes maximiza a rentabilidade sobre cada indivíduo, aumentando assim os lucros da plataforma.

Além das questões técnicas e econômicas, para que a NGN possa suportar funcionalidades como a convergência de serviços associados voz e dados é necessário um núcleo muito importante, denominado IMS (*IP Multimedia Subsystem*). Esse, por sua vez, fornece às operadoras de telecomunicação a capacidade de desenvolver uma infraestrutura única baseada em IP, independentemente do tipo de acesso, onde a implementação de serviços pode ocorrer de forma prática e ser orientado tanto a voz quanto a dados. O IMS pode ser definido como uma arquitetura orientada à serviços, sendo essa padronizada pelo 3GPP (*3rd Generation Partnership Project*) (AHSON, 2008).

2.3.1 Subsistema Multimídia IP (IMS)

A arquitetura IMS veio para suprir diversas limitação das redes atuais. A telefonia móvel, por exemplo, apesar de seguir os conceitos de *always-on* e *always-connected*, apresenta uma série de limitantes quanto a qualidade do serviço (QoS - *Quality of Service*), faturamento e flexibilidade.

A crescente demanda de dados por parte dos usuários para desfrutarem de forma estável sessões mídia em tempo real justifica o provisionamento de uma maior qualidade nas conexões estabelecidas. O QoS (*Quality of Service*) é o parâmetro que define a qualidade da sessão estabelecida entre dois terminais, sendo fundamental para controle das chamadas realizadas na arquitetura. O IMS cuida do estabelecimento da sessão e sincronização da mesma com provisão do QoS de modo que os usuários tenham uma experiência previsível (CAMARILLO, 2006).

Um grande desafio enfrentado pelas operadoras de telefonia consiste no faturamento das sessões estabelecidas. Apesar dos serviços serem executados conforme solicitado pelo cliente, as empresas enfrentam problemas para faturar com precisão esses serviços e monitorar os receptivos QoS (RUSSEL, 2008).

Em relação à flexibilidade, o IMS garante o fornecimento de serviços integrados e maleáveis como mensagens instantâneas, voz, vídeo, transmissão de TV, podendo esses serem incluídos à arquitetura facilmente. Essa característica viabiliza uma customização baseada nas preferências de cada usuário enriquecendo a experiência de cada indivíduo na plataforma.

O IMS é fundamental para o fornecimento da esperada convergência de redes, serviços e equipamentos proposta pela NGN. Uma rede como a IMS, capaz de interagir com diversos serviços, viabiliza, através do multiacesso, a reutilização de toda infraestrutura para que as operadoras introduzam serviços de voz, dados, mobilidade e vídeo (*quadruple-play*) simultaneamente. A finalidade do IMS é unir tecnologias de telecomunicação distintas em um ambiente IP. A Figura 4 apresenta a convergência de serviços, redes e dispositivos propostas pelo IMS, por meio do uso de um núcleo em comum (POIKSELKA, 2006).

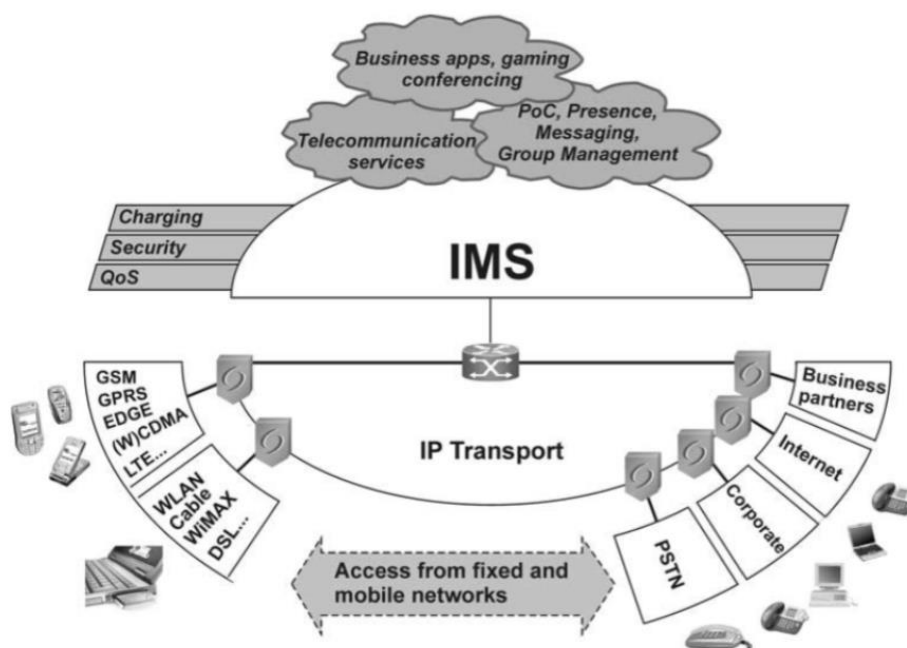


Figura 4: Convergência de redes, serviços e dispositivos provida pelo IMS (Ribeiro, 2015)

2.3.1.1 Planos do IMS

O IMS possui sua arquitetura dividida em três camadas conforme Figura 5, são elas: plano de mídia e transporte, plano de controle e sinalização e plano de serviços e aplicações.

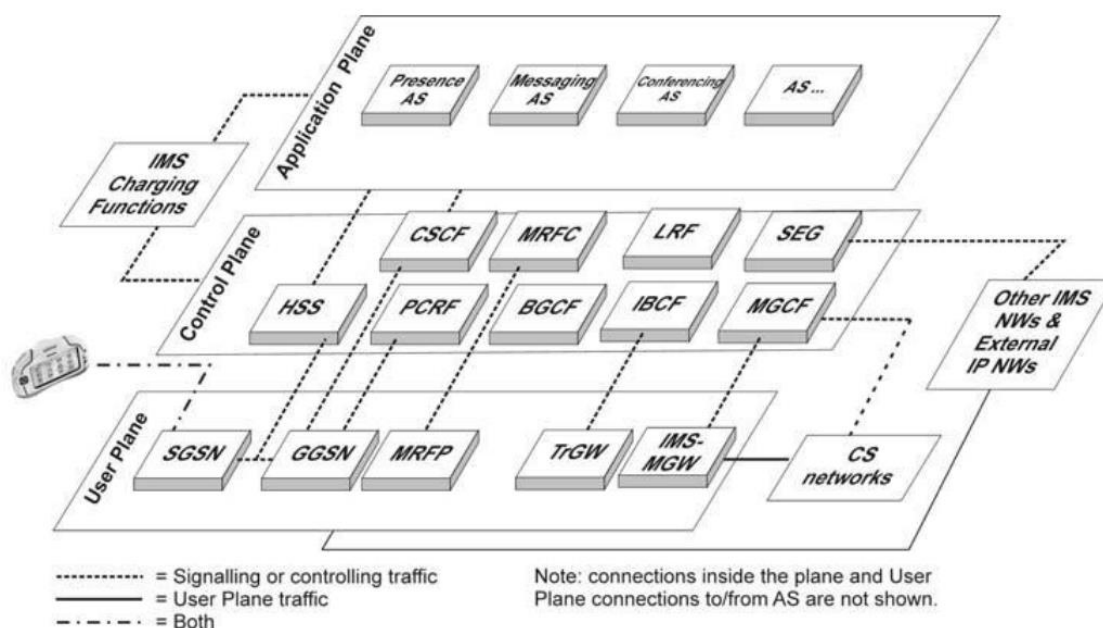


Figura 5: Arquitetura do IMS em camadas (Ribeiro, 2015)

- Plano de serviço e aplicações:** Nesse plano podem coexistir vários servidores de aplicação (AS - *Application Server*). Existem muitas finalidades para o AS tais como: provimento de serviços, conteúdos, servidor de presença e até mesmo videoconferência. Além disso, as operadoras podem fornecer diferentes serviços IMS bem como implementar seus próprios servidores de aplicação. Segundo Ahson e Ilyas (2009), cada AS suporta os protocolos SIP e DIAMETER e suas interfaces se comunicam diretamente com os C-SCFs da rede.
- Plano de controle e sinalização:** Responsável por acomodar componentes como o P-CSCF (*Proxy Call State Control Function*), I-CSCF (*Interrogation Call State Control Function*) e S-CSCF (*Serving Call State Control Function*). Esses têm como finalidade oferecer suporte à serviços de multimídia

baseados no protocolo SIP. Todo o processamento da sinalização SIP para estabelecimento das sessões entre os terminais da arquitetura ocorre nesse plano (CHANG *et al.*, 2010). Além disso, outro componente fundamental inserido nessa camada é o HSS (*Home Subscriber Server*). Esse é responsável por hospedar a base de dados da infraestrutura e centralizar os serviços bem como implementar os perfis de usuário.

- **Plano de mídia e transporte:** Representa as diferentes tecnologias de acesso à arquitetura. Nessa camada, os clientes oriundos de diversas infraestruturas obtêm conectividade à rede IP do IMS.

2.3.1.2 Protocolos do IMS

A organização da arquitetura é definida em camadas, o que permite uma melhor organização das responsabilidades e simplifica a integração de novos módulos à plataforma. Visando oferecer tal capacidade é fundamental que o IMS faça uso de protocolos robustos, como por exemplo o SIP (*Session Initiation Protocol*), RTP (*Real Time Transport Protocol*) e o *Diameter*.

Na rede IMS, existem duas categorias de tráfego, um relacionado ao controle e sinalização, denominado tráfego de sinalização, e outro associado ao transporte de conteúdo multimídia, conhecido como tráfego de dados real. O primeiro é responsável pelo SIP enquanto o segundo pelo RTP. Essa separação entre os tráfegos tem como finalidade simplificar o gerenciamento das sessões além de tornar o sistema mais adaptável às mudanças funcionais (OGUEJIOFOR, 2007).

Por fim, o protocolo *Diameter* é capaz de fornecer a proteção necessária para acesso à rede e, conseqüentemente, seu administrador pode faturar sobre os clientes pelos serviços utilizados.

Todos os protocolos mencionados possuem suas interfaces baseadas no protocolo IP visando garantir uma maior simplicidade da inserção dos mesmos no modelo da internet. Nas seções seguintes serão apresentados em detalhes os protocolos supracitados.

2.3.1.2.1 SESSION INITIATION PROTOCOL (SIP)

O protocolo SIP tem como finalidade o controle e sinalização das sessões estabelecidas. Basicamente, uma sessão pode ser definida como um fluxo de dados capazes de transportar diferentes tipos de mídia entre duas partes, como por exemplo uma chamada telefônica (*Voice Over IP - VOIP*), videoconferência, dentre outros (HERSENT, 2011).

Criado em 1996, sendo especificado no IETF pela RFC 3261 (IETF, 2002), o SIP tem sua atuação no nível de aplicação, vide modelo OSI, e é responsável por configurar, modificar e finalizar sessões multimídias em tempo real entre emissores e receptores inseridos em uma rede IP. Segundo Kurose (2006), dentre todos os protocolos de sinalização, o SIP é aquele que se mostra o mais promissor, pois, além de simples, ele irá ser capaz de abraçar todo o tráfego telefônico, que antes era centralizado em centrais PABX, migrando-os para a rede IP.

As principais características do protocolo SIP são:

- Desenvolvido especificamente para redes IP;
- Similar ao HTTP, é baseado em texto;
- Por ser um protocolo que visa atender os requisitos básicos de um protocolo de controle e sinalização de sessões, possui um número de sinalizações reduzidos se comparado a outros protocolos. Tal fato viabiliza uma maior qualidade de serviço (QoS);
- Independência entre os tráfegos de sinalização e dados;
- Compatível com qualquer protocolo de transporte. Geralmente o UDP é utilizado.

As mensagens SIP utilizam sintaxe HTTP/1.1, codificação UTF-8 e conjunto de caracteres ISO 10646. Todas essas particularidades tornam o protocolo compatível com as tecnologias IP já existentes e mais simples para serem adotadas.

As Tabela 1 e

Tabela 2 apresentam os métodos de mensagens e as respostas usadas pelo protocolo SIP.

Tabela 1: Métodos das mensagens do protocolo SIP

MÉTODO	DESCRIÇÃO
INVITE	Inicializa a chamada ou os parâmetros da mesma
ACK	Confirma um pedido, como por exemplo do tipo INVITE
BYE	Finaliza a chamada
CANCEL	Cancela o processo de busca ou discagem
OPTIONS	Utilizado para reconhecimento das capacidades do cliente
REGISTER	Registra a localização atual de um usuário na rede
INFO	Envia informações durante a sessão que não alteram seu estado

Tabela 2: Respostas do protocolo SIP

CÓDIGO	DESCRIÇÃO	EXEMPLO
1XX	Faixa de código para informações	100 - Trying, 180 - Ringing
2XX	Faixa de código para sucesso	200 - OK
3XX	Faixa de código para redirecionamento	300 - Moved Temporarily
4XX	Faixa de código para erros de clientes	404 - Not Found
5XX	Faixa de código para erros de servidores	501 - Not implemented
6XX	Faixa de código para falhas globais	603 - Declined

Todas os métodos de mensagem e códigos de resposta apresentados garantem que a sinalização entre as partes seja efetuada com sucesso. Nesse contexto, existe um outro protocolo denominado SDP (*Session Description Protocol*), atuante no corpo da mensagem SIP, que garante aos membros da ligação a obtenção do IP do

destinatário, a capacidade de transmissão disponível, o tipo de mídia a ser enviado e a porta disponível para estabelecimento da sessão RTP. O SPD viabiliza toda parametrização necessária para o RTP atuar, com as devidas configurações, na transmissão dos dados entre as partes.

2.3.1.2.1.1 Elementos da Arquitetura SIP

A arquitetura SIP se baseia no conceito de cliente-servidor. Segundo RFC 3261 (2002), cada cliente (UAC - *User Agent Client*) pode ser definido como um elemento da rede capaz de enviar solicitações SIP bem como receber respostas desse protocolo, enquanto um servidor (UAS - *User Agent Server*) é o elemento responsável por receber solicitações SIP, a fim de atendê-las, e enviar as respectivas respostas baseado no tipo de requisição aos clientes. A Figura 6 exemplifica a relação entre cada agente de usuário na arquitetura SIP.

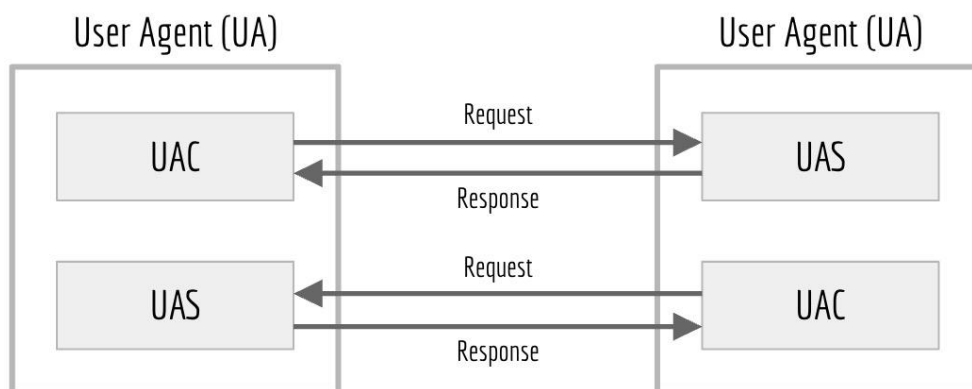


Figura 6: Agentes de usuário na arquitetura SIP (Ribeiro, 2015)

A abordagem apresentada na Figura 6 detalha, de forma simplificada, a comunicação entre dois agentes de usuário, entretanto existem implementações mais robustas que agregam mais recursos à aplicação como o B2BUA (*Back to Back User Agent*). No caso do serviço de Televoto, o B2BUA viabilizou a abstração dos endereços dos atendedores para os clientes. Conforme Figura 7, pode-se observar

que ele atua como um intermediador entre UAS e o UAC podendo assim controlar os planos de sinalização e mídia.

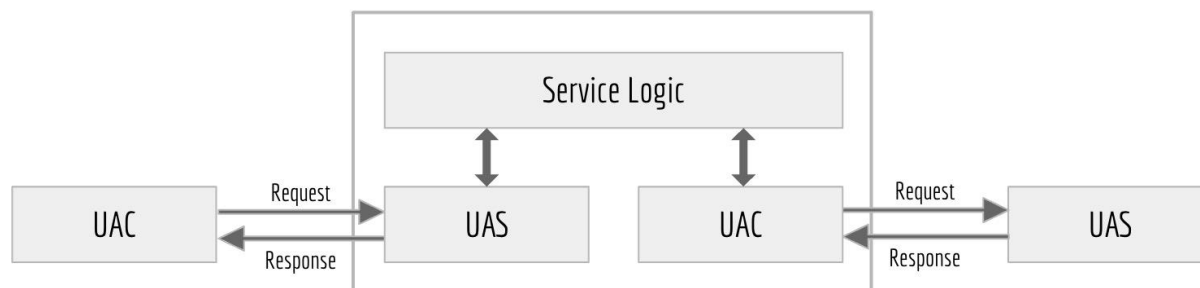


Figura 7: Arquitetura B2BUA (Ribeiro, 2015)

Um aspecto importante sobre os agentes de usuário (UA) é que eles podem se comportar tanto como UAC ou UAS. Esse dinamismo traz uma maior autonomia aos componentes da rede como por exemplo *Softphones*, telefones IP, *Gateways*, dentre outros. Todavia, existem elementos, como os servidores SIP, que possuem atividades muito específicas na arquitetura. Cada atividade caracteriza um tipo de servidor, podendo classificá-los como:

- **Register Server:** Responsável por receber solicitações de registro de um UAC além de armazenar informações como seu IP e sua identificação no banco de dados correspondente para futuras consultas;
- **Proxy Server:** Responsável por encaminhar pedidos de conexão de um UAC à outro servidor *Proxy* ou à um UAS. Além disso, ele pode trabalhar com questões referentes à autenticação, controle de acesso e roteamento;
- **Redirect Server:** Responsável por receber pedidos de conexão de um UAC e redireciona esses ao respectivo UAS.
- **Database Server:** Armazena as principais informações de cada cliente, como por exemplo: usuário, senha, IP, ramal, dentre outros.

2.3.1.2.2 Real Time Transport Protocol (RTP)

O protocolo RTP tem como finalidade permitir o transporte de mídia em tempo real, sendo seu tipo arbitrário, mas geralmente esse se resume à áudio e/ou vídeo. O RTP surgiu no intuito de solucionar problemas presentes nas redes IP associados ao transporte de dados em tempo real como o *jitter*, desordenamento de pacotes na recepção, dentre outros. Dentre as principais funções do RTP estão: identificar o tipo de informação transportada, adicionar indicadores temporais à informação enviada, gerenciar a chegada dos pacotes no destinatário além de permitir tanto *unicast* como *multicast* (HERSENT, 2011).

O RTP geralmente é transportado sob protocolo UDP (*User Datagram Protocol*). Esse, diferentemente do TCP, não é orientado à conexão, logo ele não garante o perfeito envio dos dados ao receptor, justificando assim o uso do RTCP (*RTP Control Protocol*) que é responsável por monitorar a qualidade do serviço (QoS) e levantar informações sobre os participantes da sessão (CAMARILLO, 2006). A estratégia usada pelo RTCP é monitorar o número de pacotes enviados e recebidos pelas partes envolvidas e extrair as devidas conclusões sobre a sessão.

A Figura 8 detalha os cabeçalhos que compõe o pacote usado na transmissão de mídia em uma sessão RTP.

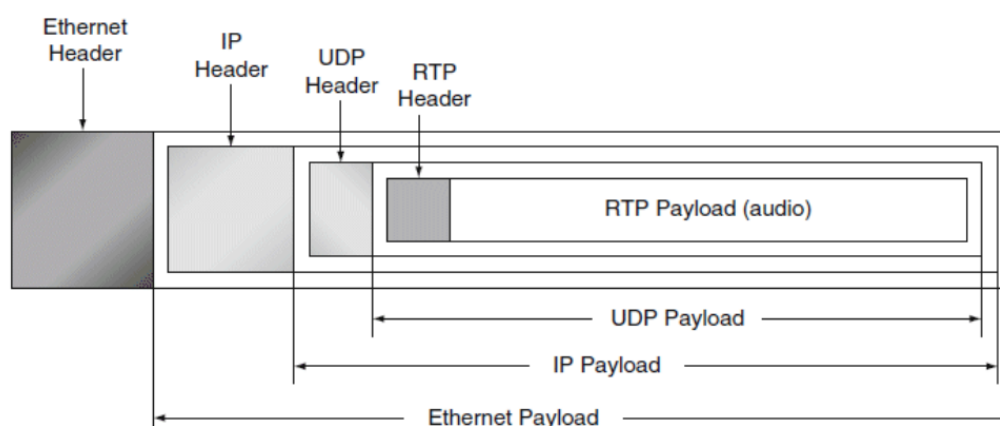


Figura 8: Cabeçalhos presentes no pacote RTP (Ribeiro, 2015)

2.3.1.2.3 Diameter

O protocolo *Diameter* possui uma grande responsabilidade na infraestrutura IMS, pois tem como finalidade lidar com questões associadas à *Authentication*, *Authorization* e *Accounting*. Por esses motivos ele também é conhecido como protocolo AAA. De forma sucinta, os termos apresentados podem ser compreendidos como:

- **Authentication:** Processo de verificação da identidade do usuário que por sua vez é realizado via análise das credenciais apresentados pelo indivíduo. Certificações digitais são comumente atribuídas à *username* e *password*. Responder questões como “Quem é o usuário?”
- **Authorization:** Após a autenticação, a autorização tem como finalidade identificar os privilégios atribuídos a cada indivíduo autenticado. Na área de TI essa característica de cada usuário é conhecida como perfil de acesso e geralmente são armazenados em uma base de dados acessada pela aplicação. Responder questões como “O que o usuário tem permissão?”
- **Accounting:** Processo de coleta das informações referentes às atividades de cada usuário na plataforma. Isso viabilizada questões como a “bilhetagem” por parte do provedor sobre os usuários baseado nos serviços por ele consumidos. Responder questões como “O que o usuário fez?”

Baseado nas funcionalidades apresentados, o protocolo *Diameter* garante a proteção e o controle para o acesso a rede. Oriundo do protocolo AAA denominado RADIUS (*Remote Authentication Dial In User*), o *Diameter* inclui diversas melhorias tais como maior contabilidade na entrega das mensagens e tratamento de erros.

2.3.1.3 Identidade de usuários

Um importante aspecto da arquitetura IMS é a identificação de cada usuário na infraestrutura. Nela, cada identidade é conhecida como identidade pública de usuário (*IMS Public Identity - IMPU*). Essas são fundamentais para comunicação entre os demais usuários e serviços (AL-BEGAIN *et al.*, 2009).

A formatação de cada identidade pública de usuário é definida conforme uma URI (*Uniform Resource Locator*) de telefone (URI TEL) ou URI SIP. O primeiro possui um formato padrão como **SIP: +1-212-555-0293@operator.com**, enquanto o segundo como **username@operator.com**.

Diferentemente da identidade pública, cada assinante IMS também é associado à uma identidade privada (*IMS Private Identity - IMPI*). O formato usado para essa modalidade é o NAI (*Network Access Identifier*) sendo esse descrito como **user@domain** (CAMARILLO, 2006). A identidade privada é usada basicamente para fins de identificação e assinatura e não para encaminhamento de solicitações SIP. Cada identidade privada de usuário pode estar associada a múltiplas identidades públicas, sendo todas essas definições feitas na base de dados do HSS.

2.3.1.4 Registro no IMS

Um dos principais procedimentos executados no IMS é o registro de usuário na rede. Sua finalidade é autenticar e autorizar o acesso de cada participante aos recursos da infraestrutura. O registro é iniciado a partir do momento que o terminal IMS (usuário) obtém conectividade à rede IP e identifica o endereço do P-CSCF. A Figura 9 exemplifica todas as etapas envolvidas no processo enquanto a Tabela 3 detalha cada uma dessas. Após o registro do terminal IMS, o mesmo se torna capaz tanto de iniciar como ser convidado a participar de sessões SIP.

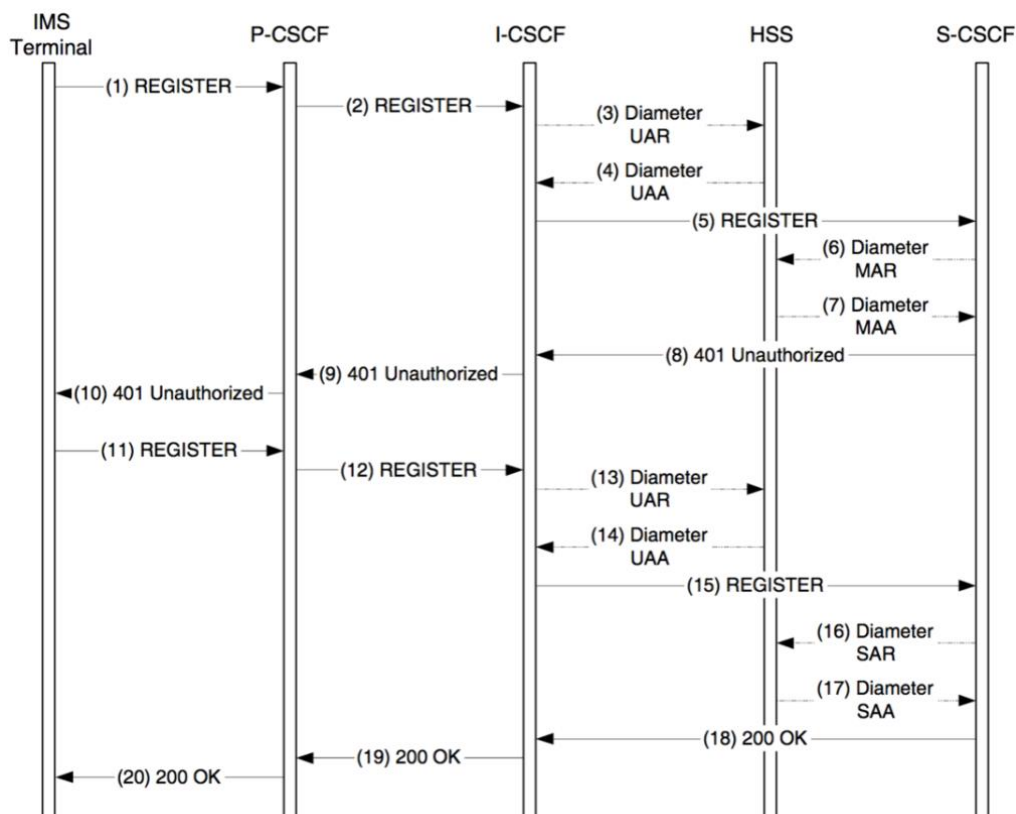


Figura 9: Registro de usuário no IMS (Ribeiro, 2015)

Tabela 3: Detalhamento das mensagens no registro de usuário no IMS

	AÇÃO
1	O terminal IMS envia a identidade do usuário ao P-CSCF via mensagem SIP REGISTER.
2	O P-CSCF processa a solicitação SIP <i>Register</i> e utiliza o nome do domínio local fornecido para encontrar o endereço do I-CSCF.
3	O I-CSCF consulta o HSS por recursos a serem usados na seleção do S-CSCF.
4	
5	O I-CSCF encaminha a solicitação <i>Register</i> ao S-CSCF obtida no passo anterior.
6	O S-CSCF identifica que o usuário não está autorizado e, por causa disso, recupera os dados de autenticação do mesmo no HSS
7	
8	O S-CSCF envia uma mensagem "401 UNAUTHORIZED" ao terminal IMS contendo um desafio DIGEST para a autenticação.
10	
11	O terminal IMS processa o desafio para autenticação e encaminha a resposta ao P-CSCF via SIP REGISTER.

12	O P-CSCF processa a solicitação SIP REGISTER e utiliza o nome do domínio local fornecido para encontrar o endereço do I-CSCF.
13 14	O I-CSCF consulta o HSS por recursos a serem usados na seleção do S-CSCF.
15	O I-CSCF encaminha a solicitação REGISTER ao S-CSCF obtida no passo anterior.
16 17	A autenticação enviada é verificada pelo S-CSCF e, em caso de aprovação, uma cópia do usuário é feita.
18 20	O S-CSCF aprova o registro e dispara uma resposta “200 OK” que por sua vez é encaminhada até o terminal IMS.

A Tabela 3 descreve de forma sucinta como a autenticação ocorre no domínio SIP. O mecanismo utilizado para esse fim é a autenticação DIGEST. Essa, por sua vez, possui sua origem no âmbito HTTP e está descrita na RFC 2671(1999). Seu mecanismo de funcionamento é muito simples, pois se baseia em hashes criptográficas no intuito de evitar a transmissão de credenciais desprotegidas. A função de uma hash é mapear dados de comprimento variável para fixo. O algoritmo utilizado no SIP DIGEST é o MD5 (*Message Digest Algorithm*) sendo cada hash de 128 bits unidirecional descrito pela RFC 1321 (1992), cujo principal uso está nos protocolos P2P (*Peer-to-Peer*) durante verificação de integridades e logins.

A autenticação DIGEST é realizada por meio da verificação, entre as partes envolvidas, sobre o conhecimento de um segredo comum. Quando o servidor quer autenticar um usuário, esse gera um desafio DIGEST, baseado nesse segredo, e o envia ao usuário. Um desafio típico que irá trafegar na plataforma desenvolvida possui a seguinte apresentação conforme Tabela 4:

Tabela 4: Exemplo de parâmetros de um desafio DIGEST

PARÂMETRO	VALOR
realm	ims.vi.ifes.edu.br
Qop	auth,auth-int
nounce	dcd98b7102dd2f0e8b11d0f600bfb0c093
opaque	
algorithm	MD5

Basicamente, o desafio enviado ao usuário corresponde a um conjunto de parâmetros cuja finalidade de cada um pode ser compreendida como:

- **realm**: O parâmetro realm é obrigatório e precisa estar presente em todos os desafios. Seu objetivo é identificar credenciais dentro de uma mensagem SIP. No caso do SIP, ele é normalmente definido igual ao domínio do servidor proxy.
- **qop**: Definido como Quality of Protection, esse parâmetro especifica quais os esquemas de proteção o servidor pode suportar. O valor auth indica apenas autenticação, enquanto o auth-int indica que tal processo deve ocorrer sob alguma proteção de integridade.
- **nonce**: Corresponde a uma string de dados (texto), especificada pelo servidor, gerada toda vez que um servidor lança um desafio DIGEST. O parâmetro nonce é construído geralmente igual ao hash MD5 a partir de alguns dados. Os dados geralmente incluem um carimbo de tempo e uma frase secreta do servidor gerador. Isso garante que cada tag nonce tenha vida útil limitada e única. Dessa forma esse parâmetro se comporta como identificador que garante que as credenciais DIGEST recebidas foram realmente geradas por um desafio DIGEST particular, além de limitar a vida útil da resposta DIGEST, evitando assim possíveis ataques de repetição.
- **opaque**: Corresponde a uma string de dados (texto) opaco. Se existir algum estado que necessite ser mantido entre o envio do desafio e o recebimento da resposta, o servidor pode passá-lo ao cliente usando esse parâmetro e ler sua resposta em outro momento assim que a resposta DIGEST enviada pelo usuário chegar.
- **algorithm**: Algoritmo usado para calcular hashes.

Após receber o desafio DIGEST, o agente de usuário gera uma resposta, também DIGEST, contendo as credenciais e envia a mesma ao servidor. A resposta pode ser descrita conforme Tabela 5:

Tabela 5: Exemplo de parâmetros de uma resposta à um desafio DIGEST

PARÂMETRO	VALOR
username	1200@ims.vi.ifes.edu.br
realm	ims.vi.ifes.edu.br
qop	Auth
nonce	dcd98b7102dd2f0e8b11d0f600bfb0c093
uri	sip:ims.vi.ifes.edu.br
nc	00000001
cnonce	0a4f113b
response	6629fae49393a05397450978507c4ef1
opaque	

Observando a resposta DIGEST é similar ao desafio. Os parâmetros que possuem o mesmo nome, também possuem as mesmas finalidades aos do desafio. O significado dos demais podem ser compreendidos como:

- **uri** - O parâmetro contém a URI que os clientes desejam acessar.
- **nc** - Contagem de nonce, o valor é o número de contagem hexadecimal do número de requisições (incluindo a requisição atual) que o cliente tem enviado com o valor nonce nessa requisição. O objetivo dessa diretiva é permitir ao servidor detectar repetições de requisições, mantendo sua própria cópia dessa contagem.
- **cnonce** - Corresponde a uma string de dados (texto) opaco marcada fornecida pelo cliente e usada tanto pelo cliente quanto pelo servidor para evitar ataques devido a escolha de texto plano, para fornecer autenticação mútua, e fornecer alguma proteção de integridade da mensagem.
- **response** - Corresponde a uma string de dados (texto) computada pelo agente usuário que tenta provar que o usuário conhece uma senha.

Assim que o servidor recebe a resposta DIGEST, ele recalcula o valor do parâmetro **response** para fins de comparação, usando os atributos fornecidos pelo cliente e a senha armazenada no próprio servidor. Caso o resultado seja idêntico a resposta recebida pelo cliente, a autenticação do cliente é finalizada.

2.3.1.5 Estabelecimento da sessão

Além do registro de usuários à rede IMS, outro importante procedimento executado na infraestrutura é o estabelecimento da sessão. Para que um usuário registrado possa usufruir dos recursos disponíveis, é necessária uma sessão junto a outro usuário ou a um servidor de aplicação (AS).

Na Figura 10 está exemplificado o fluxo de mensagens entre dois terminais IMS localizados em redes domésticas distintas, enquanto na Tabela 6 está apresentado o detalhamento de cada mensagem. A ideia é apresentar um cenário completo, sendo o presente trabalho apenas uma simplificação dessa arquitetura. Essa situação é equivalente à dois usuários estabelecendo uma chamada Voip, estando em países distintos.

É possível notar que existe uma separação entre o tráfego de sinalização e mídia, no qual o primeiro atravessa um conjunto de CSCFs, enquanto o segundo se estabelece ponto-a-ponto entre os terminais a rede IMS (CAMARILLO, 2006).

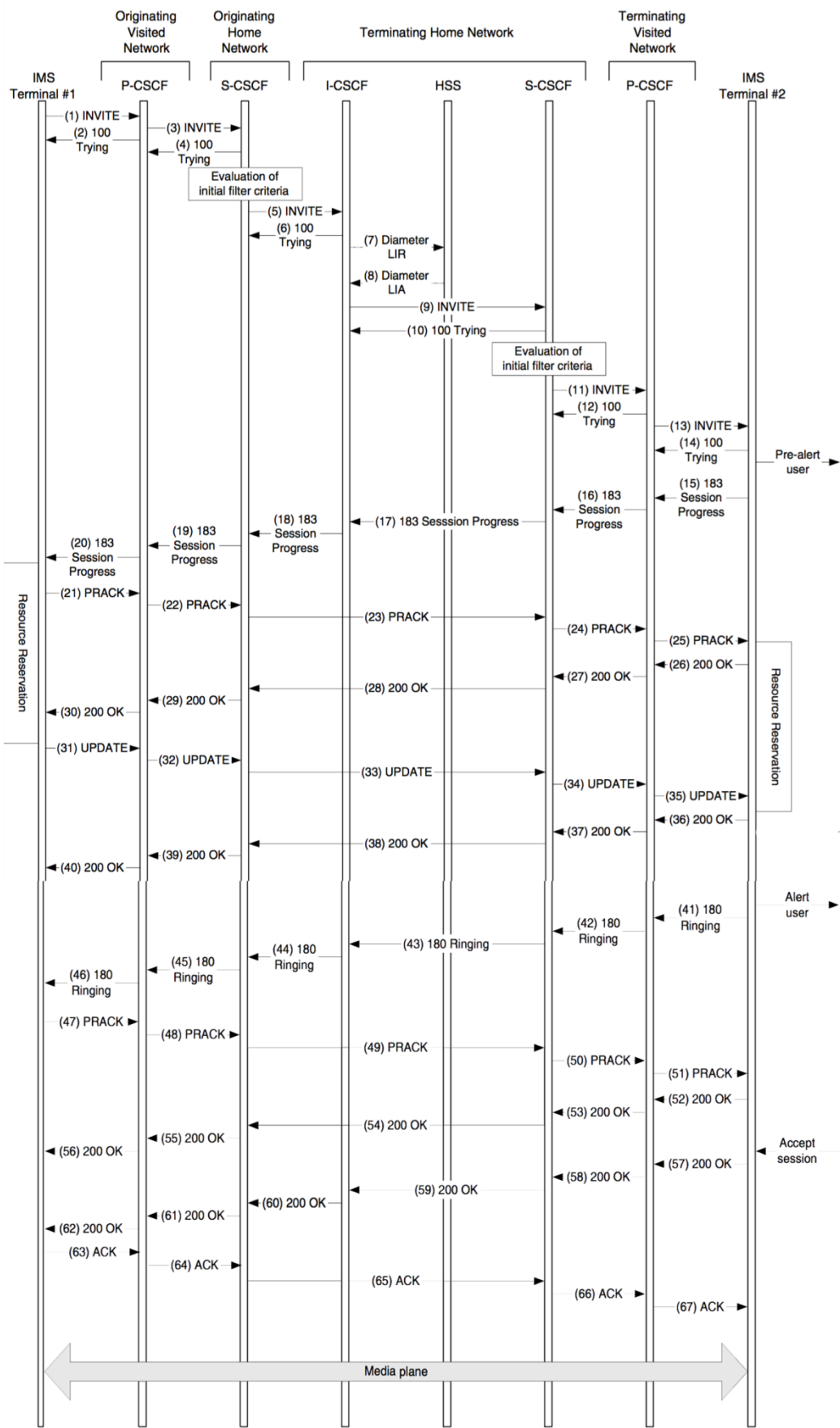


Figura 10: Estabelecimento da sessão no IMS (Ribeiro, 2015)

Tabela 6: Detalhamento das mensagens trocadas no estabelecimento da sessão no IMS

AÇÃO	
1	O terminal IMS #1 envia uma solicitação SIP INVITE ao P-CSCF
3	O P-CSCF verifica a identidade do usuário de origem e encaminha o SIP INVITE ao S-CSCF correspondente.
5	O S-CSCF processa a mensagem. Nesse processo, ele executa o controle do serviço podendo incluir interações com determinado AS além de determinar a porta de entrada do usuário IMS #2 à rede. Ao final a solicitação SIP INVITE é encaminhada ao I-CSCF da rede do usuário #2
7	O I-CSCF recebe o SIP INVITE e contata, via Diameter, o HSS para encontrar o S-CSCF que está servindo o terminal IMS #2.
9	Agora com o S-CSCF correspondente à rede do terminal IMS #2, o I-CSCF encaminha à ele a mensagem SIP INVITE.
11	O I-CSCF encaminha o SIP INVITE ao P-CSCF do terminal IMS #2
13	Após processamento adicional tratando questões como privacidade e controle, o P-CSCF, entrega a mensagem SIP INVITE oriunda do terminal #1 ao #2.
15 20	O terminal IMS #2 envia uma <i>Provisional Response</i> “ <i>Session Progress -183</i> ” ao terminal IMS #1
21 25	O terminal IMS #1 envia uma mensagem PRACK (<i>Provisinal Acknowledge</i>) ao terminal IMS #2, confirmando o recebimento da <i>Provisional Response</i> “ <i>Session Progress - 183</i> ”
26 30	O terminal IMS #2 envia uma mensagem OK ao terminal IMS #2.
31 35	Quando existem condições mandatórias para o estabelecimento da sessão, assim que o terminal IMS #1 obtém o QoS suportado pela rede, ele envia uma requisição UPDATE com a devida parametrização para a comunicação. Dessa maneira ambas as partes se alinham sob as condições na qual a sessão deve ser estabelecida.
36 40	O terminal IMS #2 envia uma mensagem OK ao terminal IMS #2 confirmando as atualizações.
41 46	Com a devida parametrização da sessão por ambas as partes, o terminal #2 envia uma mensagem “180 <i>Ringin</i> g” até o terminal #1
47 51	O terminal IMS #1 envia uma mensagem PRACK (<i>Provisinal Acknowledge</i>) ao terminal IMS #2, confirmando o recebimento da <i>Provisional Response</i> “180 - <i>Ringin</i> g”
52 56	O terminal IMS #2 envia uma mensagem OK ao terminal IMS #1 confirmando o recebimento do PRACK.
57 62	O terminal IMS #2 envia uma mensagem OK ao terminal IMS #1 confirmando a abertura das portas para o estabelecimento da sessão.
63 67	O terminal IMS #1 envia uma mensagem OK ao terminal IMS #2 confirmando a abertura das portas garantindo assim o estabelecimento da sessão entre as partes.
	Sob protocolo RTP, o plano de mídia é estabelecido e o conteúdo de mídia é trocado entre os terminais IMS.

2.4 MÍDIA DIGITAL

Mídia digital consiste em qualquer mídia capaz de ser codificada em um formato compreendido por uma máquina podendo ser criadas, modificadas, distribuídas e armazenadas em dispositivos eletrônicos. Na atual conjuntura dos computadores pessoais e dos dispositivos móveis alinhado à crescente demanda por maiores capacidades de transmissão de dados, as mídias digitais continuam em uma crescente evolução viabilizando cada vez mais o intercâmbio de informações de uma forma rápida e prática. Um dos pilares para essa evolução está nas novas técnicas de codificação desenvolvidas culminando assim em melhorias significativas tanto na transmissão quanto na recepção dos dados.

2.4.1 Codificação (*Encoding*)

Codificação de mídia não é uma tecnologia oriunda do domínio da Internet. A maioria das técnicas existentes foram feitas para os sistemas IP e posteriormente adaptadas para o IMS (CAMARILLO, 2006). Sua principal finalidade é converter um sinal analógico para digital, sendo esse capaz de ser transmitido através da rede e, por meio de um decodificador, o mesmo possa ser recuperado ao seu estado original.

O algoritmo usado para codificar (encode) e decodificar (decode) sinais é conhecido como codec. Baseado em suas finalidades, cada um deles possui uma largura de banda (*bandwidth*) e qualidade de sinal distintas.

O presente trabalho irá focar no estudo do sinal de voz, pois esse é a premissa básica para comunicação via Voip. Para exemplificar os elementos envolvidos na codificação como amostragem, quantização, serão apresentadas técnicas como o PCM (*Pulse Code Modulation*), GSM-FR (*Global System for Mobile - Full Rate*) além de estratégias como a *Linear Prediction*, pois elas possibilitam larguras de banda ainda mais enxutas para transmissão.

2.4.1.1 Amostragem (*Sampling*)

Tomando a codificação PCM como base, essa, assim como qualquer codificador, tem como finalidade converter um sinal de analógico para digital. O PCM pode possuir uma amostragem de 8000 Hz, ou seja, ele tem a capacidade de coletar 8000 amostras por segundo do sinal analógico. A amostragem apresentada na Figura 11 mostra que cada amostra irá conter um intervalo de 0.125 ms do sinal de voz.

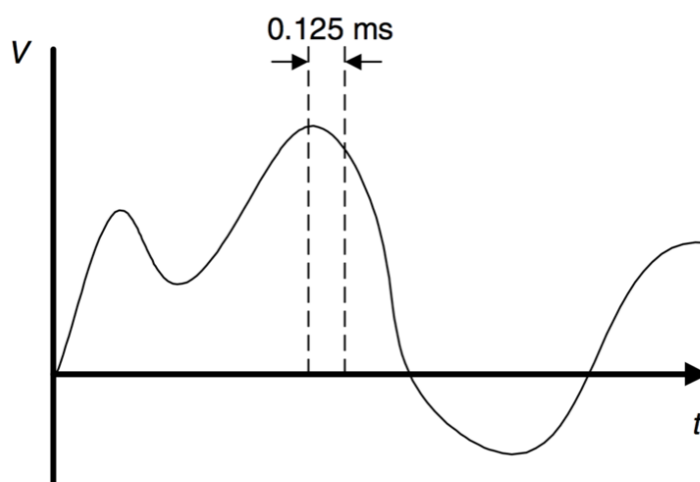


Figura 11: Amostragem de um sinal analógico (CAMARILLO, 2006)

Para cada amostra será necessário codificar a amplitude do áudio em apenas um ponto, entretanto o valor pode variar nesse intervalo, fato que pode resultar em distorções no sinal amostrado. No caso da voz humana, uma amostragem de 8000 Hz é suficiente para garantir a compreensão do sinal no receptor sem maiores problemas.

2.4.1.2 Quantização (*Quantization*)

Codecs digitais como o PCM trabalham com valores discretos, logo eles devem fornecer um valor de amplitude para cada amostra. O PCM por exemplo trabalha com a codificação do sinal usando 8 bits para definir o valor da amplitude, assim é possível determinar uma resolução de 256 diferentes valores. O mapeamento dos

valores de amplitude com os 256 níveis definidos pelos 8 bits é feito dividindo o valor do intervalo de amplitude pelos números de níveis. Esse processo é conhecido como quantização e está descrito na Figura 12 (CAMARILLO, 2006).

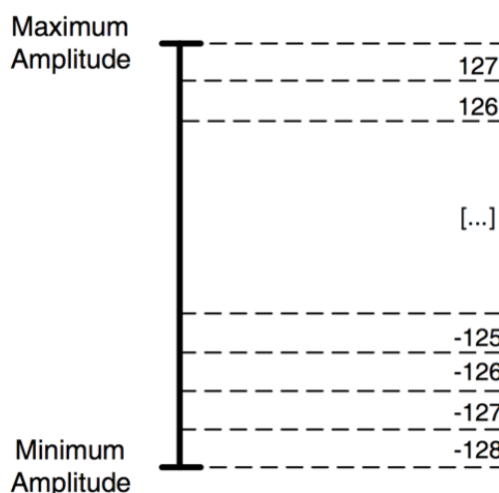


Figura 12: Quantização de um intervalo de amplitude (CAMARILLO, 2006)

2.4.1.3 Largura de Banda (*Bandwidth*)

No contexto das redes de telecomunicação a largura de banda define a capacidade de transmissão dos dados. No caso do PCM, com uma amostragem de 8000Hz com 8 bits por amostra, a largura de banda obtida é de 64kbit/s.

Existem diversas técnicas de codificação, como por exemplo a *Linear Prediction*. Essa é capaz de minimizar a largura de banda necessária para transmitir uma mesma informação, podendo reduzir esses valores a patamares abaixo de 10kbit/s. Tal capacidade existe devido as peculiaridades da voz humana, pois esse sinal, quando amostrado, apresenta intervalos correlacionados de forma que a *Linear Prediction* aproveita essa característica e usa uma amostragem com intervalos de tempo maiores (20ms) para codificação, reduzindo a largura de banda usada e mantendo praticamente a mesma qualidade do sinal transmitido (CAMARILLO, 2006). As técnicas de codificação presentes no projeto possuem sempre o mesmo intuito do que o previamente citado, melhor aproveitar a largura de banda disponível.

2.5 SERVIÇO WEB - RESTFUL

Representation State Transfer (REST) é um estilo arquitetural para sistemas hipermídia distribuídos que enfatiza a generalização das interfaces, a escalabilidade da interação entre os componentes e a instalação independente dos mesmos. O paradigma REST reúne um grupo de critérios a serem incorporados ao projeto de aplicações distribuídas, porém não existe qualquer conexão direta entre este paradigma e algum protocolo em específico (FIELDING, 2000).

Segundo Richardson (s.a.), o termo “Serviços RESTful” é utilizado para designar Serviços Web que seguem os critérios defendidos pelo paradigma REST. Adicionalmente, os autores conectaram REST ao Hypertext Transfer Protocol (HTTP) que, por sua vez, foi definido pela W3C como protocolo padrão na transmissão de mensagens na Web, trazendo assim, um aspecto tecnológico concreto àquele grupo de critérios. Tal compatibilidade faz com que todo sistema projetado de acordo com o paradigma RESTful possa ser utilizado por todos os dispositivos conectados à Web, tornando o sistema inteiramente portátil, sem qualquer dependência adicional em termos que hardware ou software (RICHARDSON, s.a.).

Na prática, um serviço RESTful pode ser usado por qualquer aplicação desde que essa seja capaz de trocar mensagens HTTP sob URLs (*Uniform Resource Locator*) definidas pelo serviço. Cada uma geralmente é encarregada por executar uma finalidade específica no servidor que hospeda o serviço.

Uma das principais características de uma aplicação RESTful é o tipo de dado utilizado para troca de mensagens HTTP. Na maioria dos casos tal troca ocorre através de objetos JSON (*Javascript Object Notation*). Esse formato é utilizado por possuir uma formatação leve de simples geração e interpretação

Baseado nas características citadas, o gravador a ser desenvolvido fará o uso da abordagem RESTful no intuito de enriquecer suas funcionalidades e simplificar o processo de desenvolvimento por meio da modularidade trazida por tal abordagem.

2.6 LEGISLAÇÃO

Nesse contexto de provimento de serviços para as redes de próxima geração, um outro aspecto a ser abordado além das questões de tecnológicas é a legislação brasileira. Essa determina uma série de exigências para as empresas prestadoras de serviços de telecomunicação. De acordo com a Lei do Marco Civil da Internet, todas as empresas que atuam nesse âmbito devem ter registrado as trocas de informações entre seus usuários por um período mínimo de seis meses e sejam capazes de fornecê-las mediante ordem judicial (BRASIL, 2014).

Um recente caso no Brasil foi a prisão do vice-presidente do Facebook na América Latina, Diego Jorge Dzodan, após a rede social descumprir decisão judicial referente ao compartilhamento informação trocadas por traficantes de drogas por meio do aplicativo de troca de mensagens WhatsApp. O Facebook adquiriu o WhatsApp desde o ano de 2014 (O GLOBO, 2016). Situações como essas apenas fortalecem a necessidade de adequar qualquer que seja a aplicação desenvolvida às normas brasileiras vigentes.

Baseado nisso o presente trabalho visa desenvolver um módulo adicional ao serviço do Televoto, um gravador de chamadas SIP. Sua finalidade, bem como seu próprio nome diz, é gravar em todas as chamadas realizadas na plataforma além de fornecer uma interface para consultas referentes à tais ligações. Essas novas funcionalidades irão enquadrar a arquitetura SANGN aos pré-requisitos da Lei do Marco Civil da Internet.

2.7 CONSIDERAÇÕES DO CAPÍTULO

O capítulo apresentado detalhou diversos conceitos usados como base para o desenvolvimento do projeto. O IMS será a arquitetura usada como ponto de partida para o desenvolvimento do gravador SIP, justificando assim um bom entendimento dessa plataforma. Os protocolos SIP, RTP e *Diameter* tem sua importância justificada por serem fundamentais para determinar como as ações do gravador irão proceder. Outros conceitos como o tipo de mídia, codificação, amostragem e

quantização são usados para parametrizar o gravador SIP de forma que a melhor configuração do dispositivo seja definida. Além das questões técnicas, existem questões jurídicas que justificam o desenvolvimento de tal gravador, baseados nos artigos definidos pela Lei do Marco Civil da Internet, tornando assim seu entendimento plausível para o presente projeto. A Figura 13 detalha os principais motivos usados como justificativa para cada embasamento abordado.

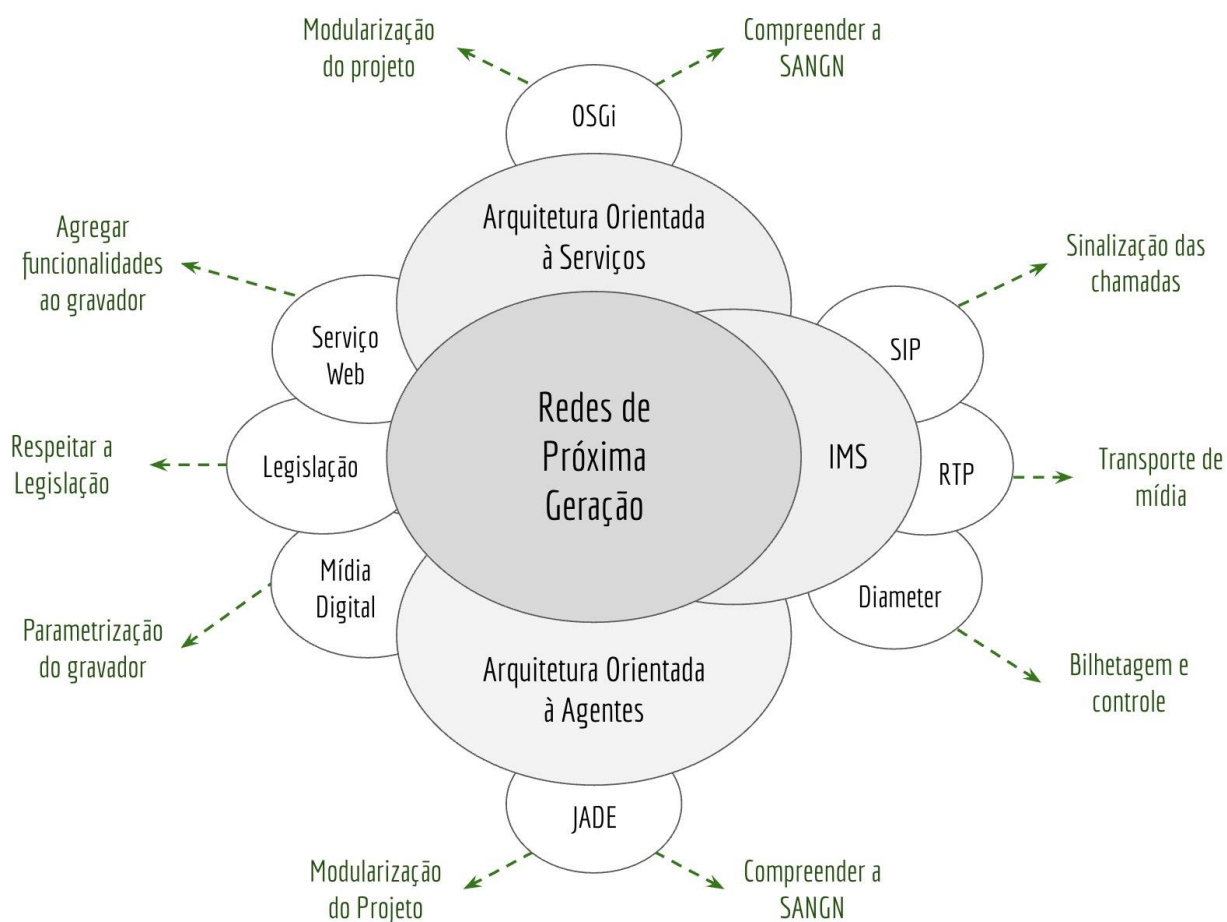


Figura 13: Justificativa do conteúdo base para o desenvolvimento do projeto

3 GRAVADOR SIP

O presente projeto, como dito a priori, tem como finalidade desenvolver um gravador SIP e adicioná-lo ao serviço do Televoto. Essa tarefa envolve um entendimento não apenas dos fundamentos do protocolo SIP ou da arquitetura IMS, mas também de uma profunda compreensão da arquitetura do serviço do Televoto, local onde o gravador será futuramente inserido.

O cenário escolhido para o desenvolvimento do módulo gravador será a comunicação entre dois terminais IMS. Um deles corresponderá a um agente ligador (*Caller*), que representará uma pessoa, enquanto o outro será um agente atendedor (*Callee*), que corresponderá a uma secretária eletrônica.

A metodologia usada para o desenvolvimento do gravador SIP envolveu primeiramente a definição do escopo que será trabalhado seguido pela determinação de uma série de requisitos e funcionalidades do dispositivo, visando tornar o desenvolvimento do trabalho mais objetivo. A partir desse ponto, foi estabelecida uma máquina de estado com os estágios das trocas de mensagens SIP de forma que fosse possível determinar precisamente o instante de início da gravação.

Na etapa de desenvolvimento os componentes criados foram: o agente gravador **RecordAgent**, responsável por disponibilizar os recursos de gravação de forma orientada à serviços por meio de um agente JADE, a classe **RTPExport**, responsável por gravar e registrar as principais informações referentes às chamadas efetuadas e o **Cliente Web**, utilizado para posterior consulta aos registros armazenados e o **serviço RESTful**, usado tanto para o provimento de funcionalidades usadas no gerenciamento das portas das sessões RTP como para a exposição do conteúdo armazenado para o cliente Web.

O processo de validação do trabalho se deu por meio da análise de um estudo de caso que envolve a aplicação do módulo gravador SIP na plataforma SANGN abordando todas as etapas envolvidas, desde a parametrização da sessão RTP,

monitoramento das sessões SIP e RTP, gravação da chamada em disco e posteriormente a consulta das mesmas.

3.1 AMBIENTE TRABALHADO

Uma profunda compreensão sobre o ambiente onde o gravador SIP será inserido foi de extrema valia para o desenvolvimento do trabalho. Basicamente, o ambiente corresponde à uma rede local composta por 4 elementos principais conforme ilustra a Figura 14. Suas atribuições são:

- Um servidor de DNS (*Domain Name System*) responsável por cuidar do encaminhamento das requisições aos devidos dispositivos e portas e abstrair os endereços IP dos elementos da rede por meio de URLs;
- Um servidor IMS responsável por armazenar todos os elementos de tal arquitetura, cuidando desde o processamento das mensagens como o armazenamento de todo conteúdo das ligações;
- Um agente ligador (*Caller*) responsável por iniciar e estabelecer a ligação com o agente recebedor (*Callee*) por meio da troca de mensagens SIP junto ao servidor IMS;
- Um agente recebedor (*Callee*) responsável por processar e estabelecer a ligação com o agente ligador (*Caller*) por meio da troca de mensagens SIP junto ao servidor IMS.

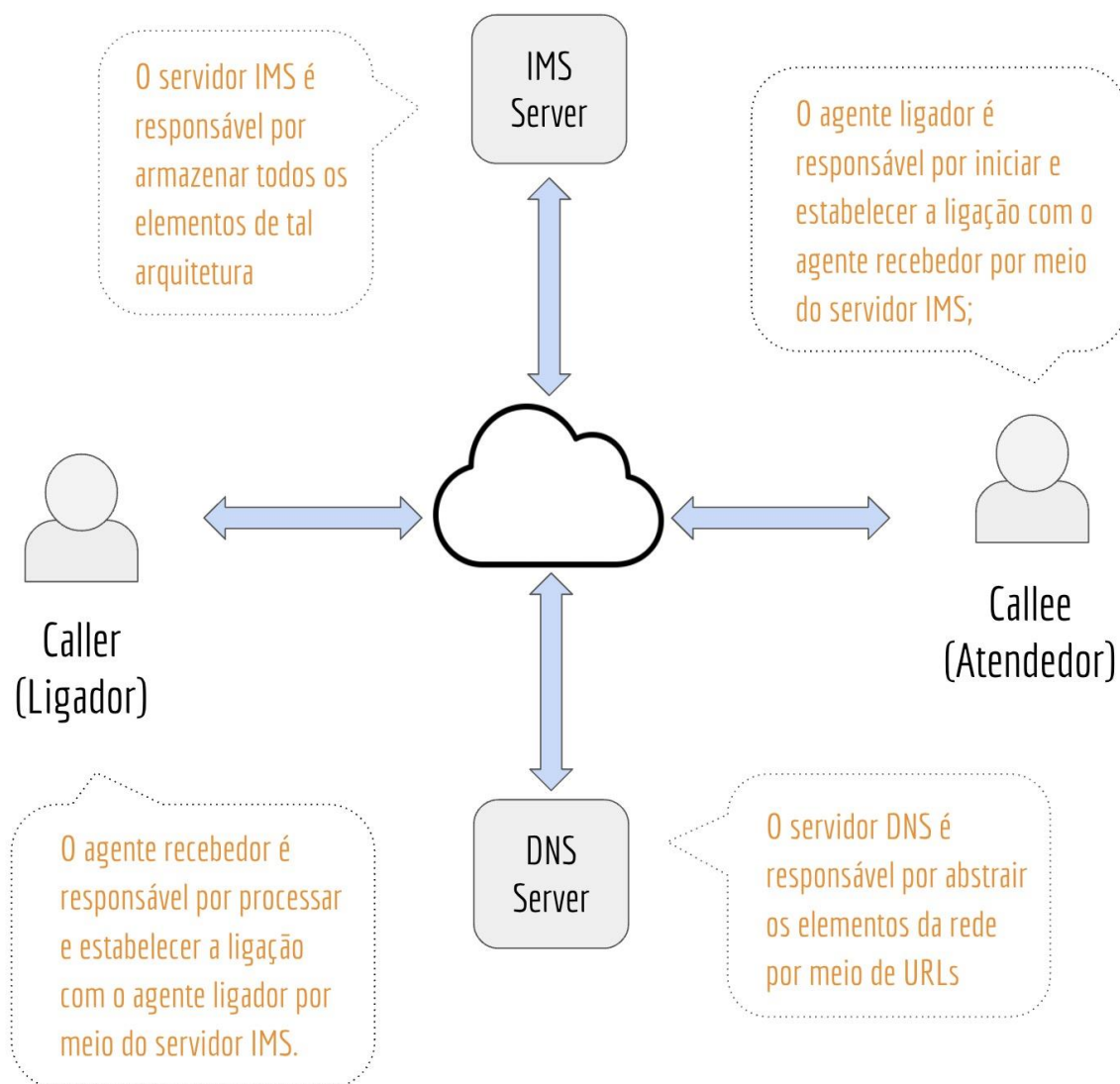


Figura 14: Ambiente no qual o gravador será inserido

3.2 REQUISITOS E FUNCIONALIDADES

O gravador SIP a ser desenvolvido deve respeitar uma série de requisitos e cumprir certas funcionalidades. As determinações a serem apresentadas definirão o escopo a ser trabalhado, tornando assim todo o trabalho mais objetivo, são elas:

- O gravador SIP deve gravar todo conteúdo de áudio de cada chamada em um arquivo físico além de disponibilizá-lo para futuras consultas. A

parametrização da gravação a ser adotada deve visar otimizar a relação da qualidade da chamada junto ao consumo em disco;

- O módulo gravador deve ser desenvolvido na forma de agente JADE visando facilitar a sua posterior integração com a arquitetura SANGN;
- Registrar as informações mais importantes da ligação tais como: as partes envolvidas na chamada (URI), horário da ligação, dentre outros. Todos os registros devem ser armazenados em uma base de dados unificada;
- Disponibilizar em um painel todo conteúdo associado às gravações efetuadas para posteriores consultas.
- O gravador deve ser inserido na máquina de estado dos agentes recebedor e ligador, respeitando assim a modularidade do código usado como base para o projeto;
- Todos os parâmetros do gravador SIP devem ser acessíveis via arquivo de configuração no intuito de facilitar a alteração desses, eliminando assim a necessidade de se gerar uma nova versão do aplicativo a cada alteração;
- Cada chamada terá duração aproximada de 20 segundos, caracterizando assim um serviço de Televoto como deve ser, simples e objetivo.

3.3 MÁQUINA DE ESTADO

Visando respeitar as determinações impostas pelos protocolos SIP e RTP, uma máquina de estado foi desenvolvida no intuito de modularizar o desenvolvimento dos componentes a facilitar o entendimento dos estados durante a troca de mensagens. A Figura 15 apresenta uma máquina de estado conceitual para exemplificar como os agentes *Caller* e *Callee* foram desenvolvidos. No total foram definidos sete estados, são eles: REGISTERING, IDLE, RINGING, WAIT_PROV, WAIT_FINAL, WAIT_ACK e ESTABLISHED. Cada um possui fundamental importância na atribuição das ações de cada componente em instantes distintos. A função de cada estado é definida como:

- **REGISTERING:** Estado responsável pelo registro do agente JADE, seja o *Caller* ou o *Callee* na infraestrutura IMS. Caso a resposta obtida seja 401 (*Not Found*) o componente permanece no estado corrente e efetua novas tentativa de registro. Caso para alguma tentativa de registro a resposta seja 403 (*Invalid Credentials*), as tentativas de registro são encerradas.
- **IDLE:** Esse estado define uma situação de espera para o componente, estando esse aberto à diversos tipos de requisição.
 - No caso de *Caller*, dado seu comportamento ativo e responsável pelo início da comunicação, assim que o tal estado é ativado, o próprio envia uma requisição INVITE ao destinatário e em seguida altera seu estado para WAIT_PROV.
 - No caso do *Callee*, o estado IDLE habilita-o a processar a mensagem INVITE e em seguida disparar ao agente *Caller* uma resposta *Provisional* (183) confirmando seu recebimento. Na sequência seu estado é alterado para RINGING e em seguida para WAIT_ACK, sendo essa transição determinada pelo envio, ao agente ligador, da resposta 200 (OK) contendo o protocolo SDP.
- **RINGING:** Define o estado no qual o componente *Callee* envia um *feedback* à infraestrutura que ele está no caminho para estabelecimento de uma chamada.
- **WAIT_PROV:** Como o próprio nome diz, esse estado informa que o *Caller* está aguardando o recebimento de uma resposta *Provisional* (183) do *Callee*. Basicamente sua função é confirmar o recebimento da mensagem INVITE pelo destinatário.
- **WAIT_FINAL:** Estado intermediário entre a confirmação por parte do *Callee* ao *Caller* sobre o início da sessão RTP e o próprio estabelecimento da mesma.

- **WAIT_ACK**: Estado de repouso enquanto a mensagem ACK não é recebida. Assim que o *Callee* a recebe, ele estabelece a sessão RTP junto ao agente ligador *Caller*.
- **ESTABLISHED**: Esse estado confirma o estabelecimento da sessão RTP. Nele, todas as partes envolvidas são capazes de se localizar na rede e iniciar a transmissão dos dados.

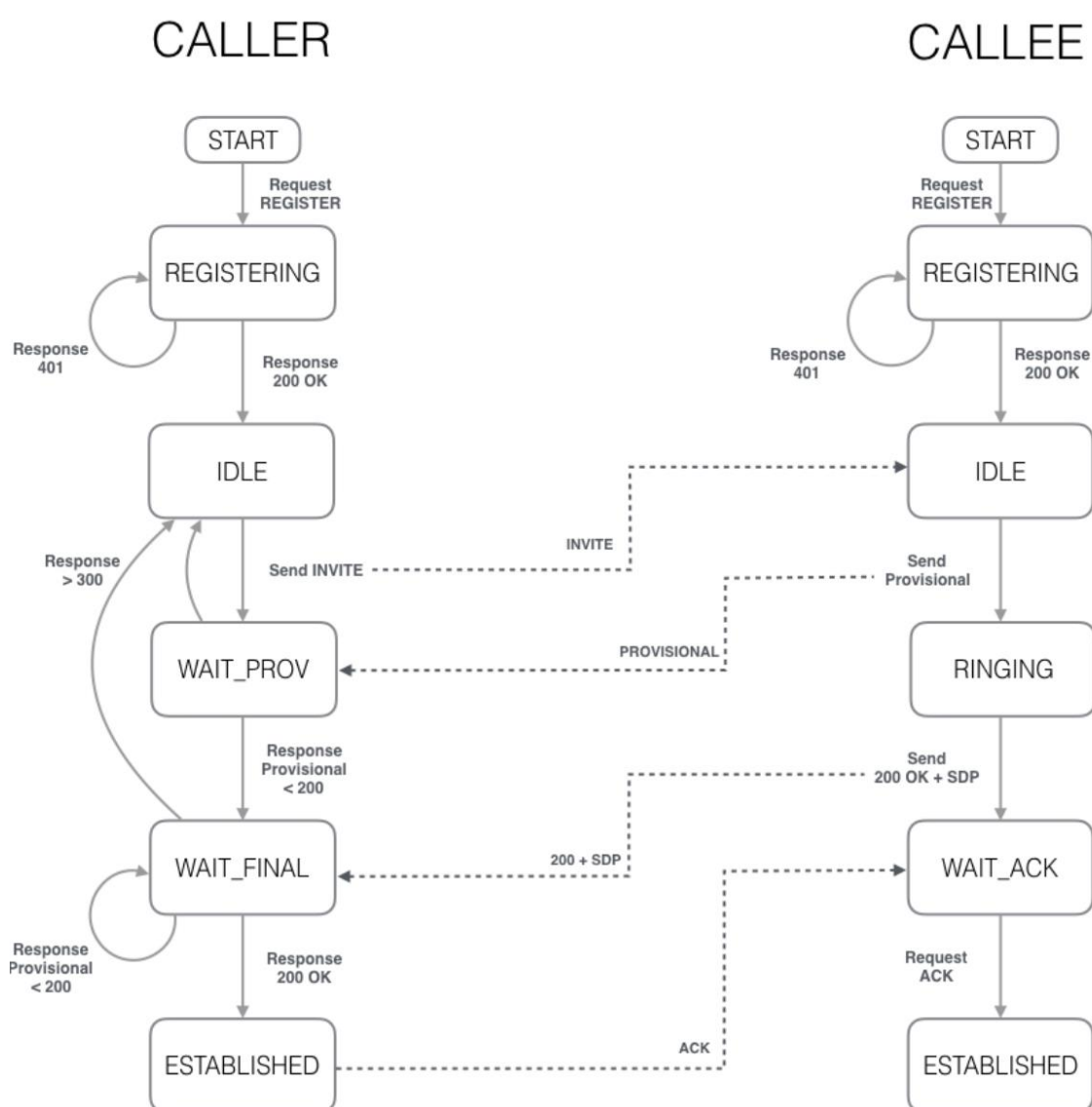


Figura 15: Máquina de estados dos agentes JADE trabalhados

Dentre os estados apresentados, existem outros comportamentos que não foram abordados na Figura 15, pois o foco era exemplificar as etapas necessárias envolvidas desde o início da troca de mensagens entre as partes até o estabelecimento da sessão RTP.

3.4 COMPONENTES DESENVOLVIDOS

A nível de desenvolvimento, os componentes trabalhados foram: o agente gravador denominado **RecordAgent**, responsável por disponibilizar os recursos de gravação de forma orientada à serviços por meio de um agente JADE, a classe **RTPEXport**, responsável por gravar e registrar as principais informações referentes às chamadas efetuadas, o **serviço RESTful** usado para duas funções, a primeira se resume à gerenciar as portas a serem usadas pelo RecordAgent no estabelecimento das sessões RTP e a segunda consiste em disponibilizar o conteúdo das chamadas através de objetos JSON. Por fim, um **cliente Web** que, por meio de uma interface acessível ao usuário, exibe as informações relevantes de cada chamada.

Os dois primeiros componentes foram desenvolvidos usando a linguagem orientada à objeto JAVA junto ao *package* JAVAX, cuja finalidade é encapsular e padronizar diversas extensões JAVA como por exemplo às funcionalidades associadas à troca de mensagens SIP e RTP em uma rede. O serviço RESTful e o cliente Web usaram a linguagem *javascript* cuja implementação se baseou na tecnologia *Node.js* junto com a framework *Express.js*, usada para o desenvolvimento de aplicações Web.

Todo o conteúdo desenvolvido está disponível para consulta em um projeto no Github (web site usado para compartilhamento de código) sob a URL: <https://github.com/FlavioCoradini/sip-client>.

3.4.1 Record Agent

A classe RecordAgent tem como finalidade exportar as funcionalidades necessárias para gravação de uma sessão RTP, respeitando todos os conceitos de

modularidade fornecidos pela abordagem de desenvolvimento JADE. As funcionalidades dessa classe são:

- Expor, de forma clara para a aplicação que fará o seu uso, as funcionalidades para gravação da sessão RTP;
- Por se tratar de um agente JADE, as funcionalidades do RecordAgent não bloqueiam a execução da aplicação que faz o seu uso, pois tudo ocorre em *Threads* independentes da principal;
- Executar as funcionalidades da classe RTPExport.

Um importante aspecto a ser ressaltado sobre o Record Agent se refere à como a plataforma fará o uso de suas funcionalidades. A Figura 16 caracteriza esse processo:



Figura 16: Esquema sobre o uso do agente gravador

Note que o agente gravador é sempre iniciado e finalizado por um agente atendedor, caracterizando assim uma relação de "PAI" e "FILHO" entre tais agentes JADE. Essa relação é fundamental para o entendimento da posterior integração dos elementos desenvolvidos.

3.4.2 RTPExport

A classe RTPExport tem como finalidade gravar o conteúdo de mídia que trafegou na sessão RTP estabelecida em disco e registrar na base de dados do serviço do Televoto as informações referentes a cada chamada. Além disso, possui a validação necessária para expor eventuais problemas que podem ocorrer durante o processo de gravação, como por exemplo: corrupção da mídia, instabilidades ou não existência da sessão RTP, dentre outros.

Seus parâmetros de entrada se resumem apenas ao caminho físico no qual o conteúdo de mídia será gravado, a duração da sessão em segundos e a URL da sessão RTP. A classe em questão também consome parte das funcionalidades do serviço RESTful de gerenciamento de portas usadas nas sessões RTP.

3.4.3 Serviço Web

O serviço Web foi desenvolvido com o foco em duas ações: a primeira consiste em disponibilizar, em formato JSON, as chamadas realizadas e armazenadas na base de dados do Televoto, enquanto a segunda se resume a gerenciar o uso das portas usadas pelos agentes *Caller* e *Callee* para o estabelecimento das sessões RTP. Tal divisão proporcionou o desenvolvimento de dois subcomponentes, são eles respectivamente: **Repositório de Chamadas** e o **Gerenciador de Portas**.

As tecnologias usadas para o provimento de tais elementos foram o *Node.js* junto ao *framework Express.js*. O primeiro pode ser compreendido como uma plataforma, desenvolvida na linguagem *javascript*, no intuito de construir aplicações de rede rápidas e escaláveis. O segundo consiste em um pacote que, quando adicionado ao *Node.js*, agrega à plataforma funcionalidades do âmbito da Web, como por exemplo o desenvolvimento aplicações orientadas à serviço, no caso do trabalho foi o REST.

Baseado nessas duas tecnologias, o serviço Web trabalhado possui as seguintes funcionalidades disponibilizadas conforme Tabela 7. Elas podem ser consultadas e/ou executadas via requisições HTTP sob as URLs descritas.

Tabela 7: Funcionalidades disponibilizadas pelo serviço Web

URL	AÇÃO
http://localhost:3000/records	Consulta as chamadas realizadas
http://localhost:3000/ports	Consulta as portas
http://localhost:3000/ports/getAvailable	Obtém uma porta e a torna indisponível
http://localhost:3000/ports/availablePort/port_number	Torna a porta informada disponível

3.4.3.1 Repositório de Chamadas

O serviço Web possui como uma de suas finalidades disponibilizar, em formato JSON, as chamadas realizadas e armazenadas na base de dados do Televoto. Dada essa característica um dos subcomponentes do serviço foi denominado Repositório de Chamadas.

3.4.3.2 Gerenciador de Portas

Inicialmente o serviço Web teria apenas como característica expor as informações referentes às chamadas executadas na plataforma do Televoto, enquanto o gerenciamento das portas seria executado por meio de um servidor proxy denominado **RTPProxy**. Esse, por sua vez, seria responsável por determinar uma faixa de valores (*range*) de portas disponíveis para o estabelecimento das sessões RTP, entretanto, dada a complexidade em termos de configuração, uma alternativa teve de ser encontrada para facilitar tal gestão. Baseado nisso foi desenvolvido um subcomponente denominado **Gerenciador de Portas**, cuja finalidade é gerenciar uma determinada faixa de valores portas a ser usada pelo servidor para o estabelecimento de sessões RTP.

O processo de gerenciamento de portas pode ser compreendido conforme Figura 17.

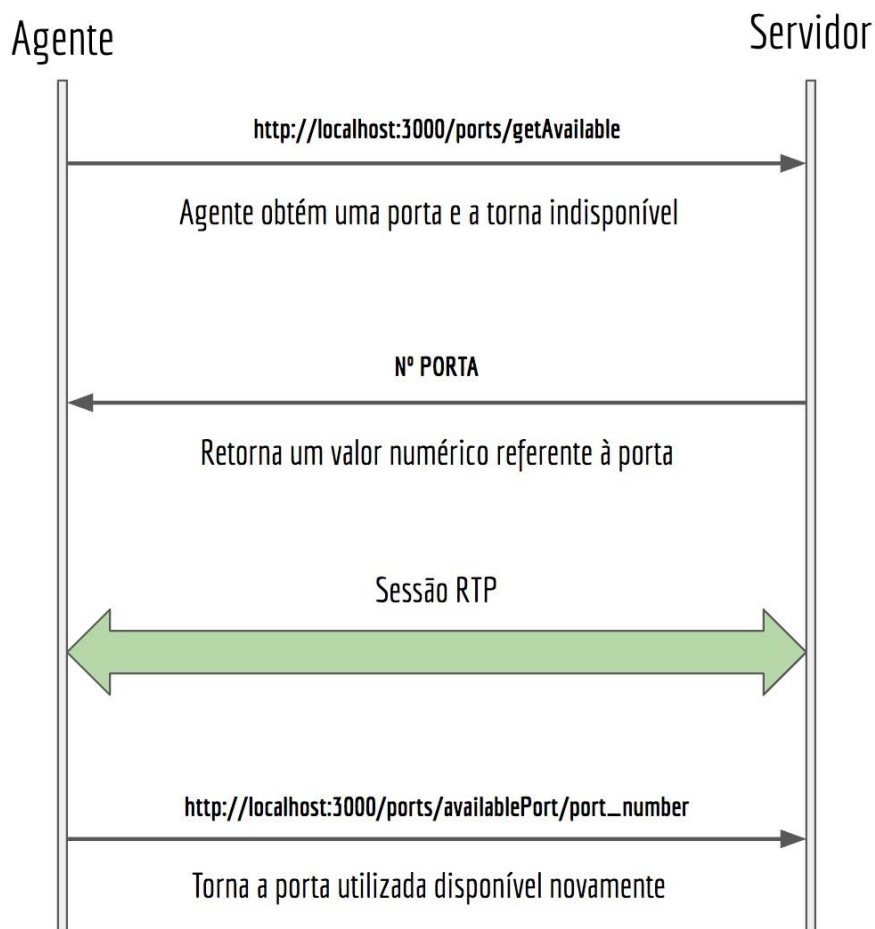


Figura 17: Processo de gerenciamento de portas entre agente e servidor

3.4.4 Cliente Web

A aplicação web desenvolvida tem como finalidade disponibilizar todas as informações relevantes sobre as ligações efetuadas na plataforma SANGN, como por exemplo: horário da ligação, duração, conteúdo e URI dos agentes ligados e atendedores. Além disso, o aplicativo permite o acesso aos arquivos físicos armazenados em disco e disponibilizá-los para download. Todas essas características tornam o trabalho de consulta e controle sobre o serviço de Televoto muito mais simplificado.

Da mesma maneira que o serviço Web desenvolvido, a tecnologia usada para o desenvolvimento foi o *Node.js* junto ao *Express.js*. O banco de dados a ser

consultado pelo aplicativo é o mesmo compartilhado com o serviço do Televoto. Esse corresponde à uma instância do MySQL localizada no servidor OpenIMS.

A interface da aplicação é bem simples e intuitiva, conforme Figura 18 e Figura 19. A primeira se resume apenas a exibir as seguintes informações: identificação da chamada (Call-ID), as URIs dos agentes *Caller* e *Callee*, o horário da chamada e o link para download do conteúdo da ligação enquanto a segunda exibe a faixa de portas utilizadas pelos clientes e informa se essas estão disponíveis para serem utilizadas.

Records

Show 10 entries Search:

Call-ID	Caller	Callee	Time	
05f7b9c56fb49cfbced3264a2355e29e@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/10/15 22:53:35	
14376d8090b015f954386b6099ff4296@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/11/01 22:21:51	
292e1818218a1765d9eb88e7f012e229@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/11/01 21:12:46	
2f6d5011c8e1d65ccb0d1e62f143ecb4@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/11/01 22:17:24	
acd78e0d49c96c294c750f82b409d0bb@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/11/01 22:15:10	
e37b0dcb53fe858e94cd7651a20bb168@172.16.28.1	<sip:1202@ims.vi.ifes.edu.br>	<sip:1200@ims.vi.ifes.edu.br>	2016/11/01 21:16:38	

Showing 1 to 6 of 6 entries Previous 1 Next

Figura 18: Interface do usuário do cliente web (chamadas)

Ports

Show 10 entries Search:

PORT NUMBER	AVAILABLE
30000	false
30001	true
30002	true
30003	true
30004	true
30005	true
30006	true
30007	true
30008	true
30009	true

Showing 1 to 10 of 1,001 entries Previous 1 2 3 4 5 ... 101 Next

Figura 19: Interface do usuário do cliente web (portas)

3.5 ESTUDO DE CASO

Visando validar o gravador proposto, foi estabelecido um cenário de comunicação entre dois terminais IMS inseridos na arquitetura do serviço do Televoto. Um deles corresponderá a um agente ligador (*Caller*), que representará uma pessoa, enquanto o outro será um agente atendedor (*Callee*), que corresponderá a uma secretária eletrônica. O conteúdo da mídia a ser trocado corresponderá a um arquivo de áudio no formato wav armazenado em disco com duração de 20 segundos.

Serão detalhados os componentes trabalhados, a arquitetura do estudo de caso, a parametrização da sessão RTP, o processo de gravação, o monitoramento da sessão SIP e os problemas enfrentados.

3.5.1 Componentes trabalhados

Todos os componentes trabalhados estarão na mesma rede local cuja topologia envolverá 4 elementos distintos conforme Figura 20, são eles:



Figura 20: Elementos presentes no estudo de caso

Agents: Trata-se de um Macbook Pro usado para o desenvolvimento dos componentes JADE. O ambiente de desenvolvimento utilizado foi o Eclipse IDE. Nele serão executados os agentes *Caller* e *Callee* sendo que cada um será registrado no OpenIMS conforme Tabela 8 de URIs. Note que os agentes não estão identificados por seus nomes e sim pelos números 1202 e 1200 respectivamente.

Tabela 8: URI dos componentes trabalhados

COMPONENTE	SIP URI
Caller	sip:1202@ims.vi.ifes.edu.br
Callee	sip:1200@ims.vi.ifes.edu.br

OpenIMS: Trata-se de um servidor Linux (Ubuntu - 64 bits) portador da arquitetura IMS (OpenIMS - Kamailio). Sua inicialização consiste na ativação dos seguintes elementos do IMS: I-CSCF, P-CSCF, S-CSCF e HSS. Cada um possui uma porta específica na instância, entretanto, visando simplificar a localização desses elementos na rede, URLs foram definidas no servidor de DNS para identificar cada um. As URLs são mostradas na Tabela 9:

Tabela 9: DNS dos componentes do OpenIMS

COMPONENTE	IP : PORTA	DNS
Localhost	172.16.28.1	ims.vi.ifes.edu.br
I-CSCF	172.16.28.1:5060	icscf.ims.vi.ifes.edu.br
P-CSCF	172.16.28.1:4060	pcscf.ims.vi.ifes.edu.br
S-CSCF	172.16.28.1:6060	scscf.ims.vi.ifes.edu.br
HSS	172.16.28.1	hss.ims.vi.ifes.edu.br

Infraestrutura: Trata-se de um Servidor Linux (Ubuntu - 64 bits) portador da tabela de DNS compartilhada pelos elementos da rede. Essa instância é de suma importância para funcionamento da arquitetura, pois nela estão definidas as rotas de todos os elementos do IMS. Além disso, dada a definição manual da rede **172.16.28.0** sob máscara **255.255.252.0**, o servidor de DNS simplifica a configuração da arquitetura de redes distintas, dada a abstração dos IPs por meio de URLs.

Sniffer: Trata-se de um servidor Linux (Ubuntu - 64 bits) capaz de acessar a interface de rede na qual estão inseridos os elementos da rede. O programa utilizado para monitorar o tráfego foi o Wireshark. Tal ferramenta viabiliza o monitoramento da entrada e saída de dados, em diferentes protocolos, à qual o

computador está ligado. Essa ação se mostrou de extrema valia para o trabalho, pois permitiu compreender o real tráfego de pacotes SIP na rede.

3.5.2 Arquitetura elaborada para o estudo de caso

A arquitetura montada para o estudo em questão está ilustrada na Figura 21. É importante ressaltar que:

- A relação mais importante na arquitetura se baseia na tripla: núcleo IMS e os agentes ligador (Caller) e atendedor (Callee).
- O núcleo IMS contém a base de dados que será consumida pelas aplicações.
- O servidor de DNS é usado apenas para o mapeamento das URLs definidas no servidor IMS.
- A ferramenta Wireshark se comporta como um espectador na comunicação entre os demais elementos, pois sua atribuição no estudo de caso se limitou ao monitoramento do tráfego SIP na rede.

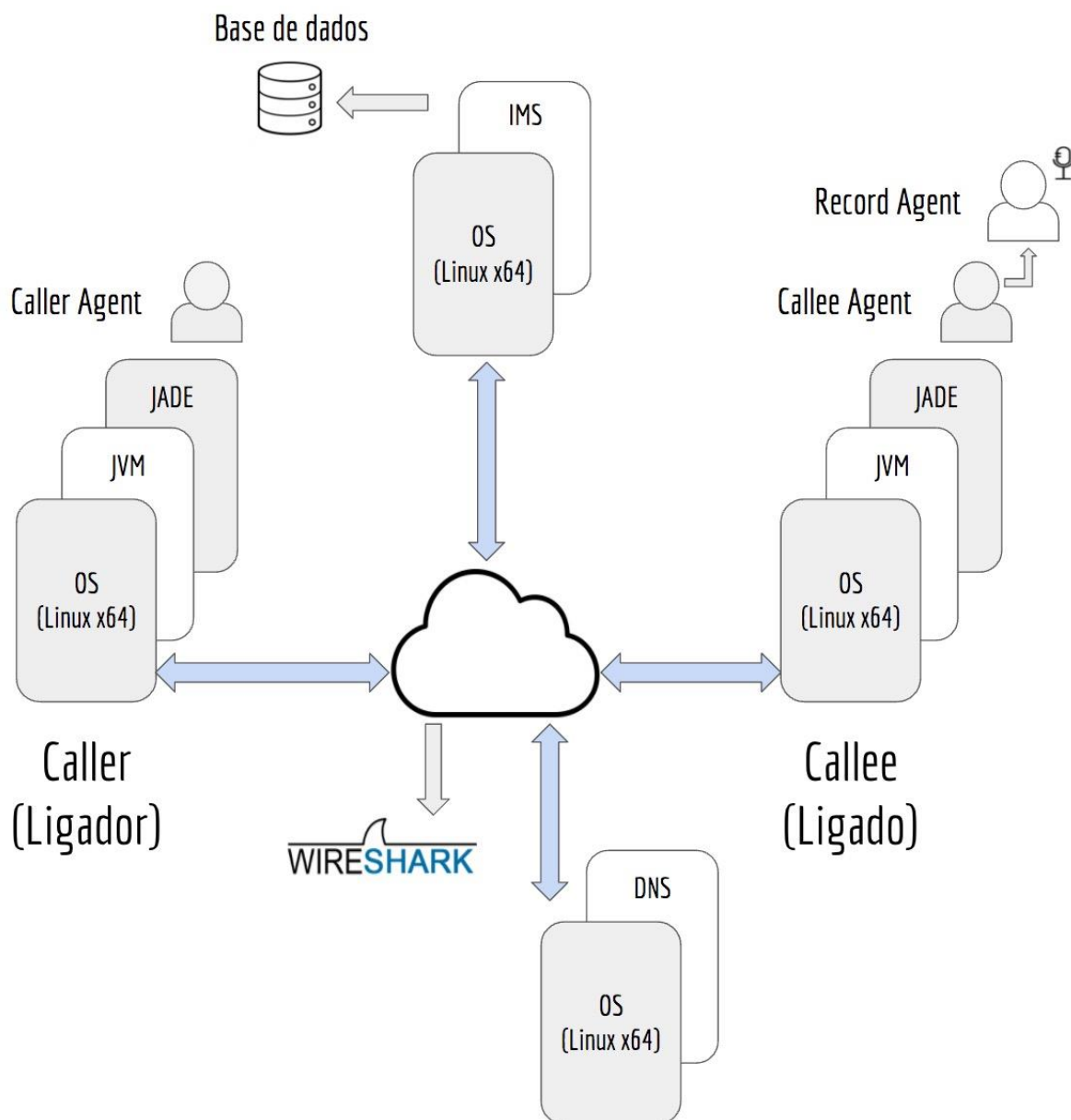


Figura 21: Arquitetura do estudo de caso

3.5.3 Parametrização da sessão RTP

Um importante aspecto para o estabelecimento da sessão RTP é a parametrização da mesma. Esse fato é justificado pela necessidade de um entendimento entre as partes envolvidas, pois aspectos técnicos como portas utilizadas, IP do host, codificação, tipo de mídia são de suma importância para a devida comunicação. Todas essas informações são carregadas via SDP (*Session Description Protocol*),

protocolo encapsulado pelo SIP, responsável pela descrição sobre como a sessão de mídia será estabelecida.

Os dois elementos trabalhados, *Caller* e *Callee*, possuem atribuições distintas no estabelecimento da sessão. O primeiro, por possuir um comportamento ativo, é determinante na parametrização, pois as regras por ele descritas serão consideradas pelo receptor. O segundo elemento recebe todas as instruções sobre como ele vai efetivar e processar a sessão via SIP + SDP. Todas devidas considerações são coletadas e usadas para configuração da sessão RTP. O sincronismo entre as partes é fundamental. Os parâmetros adotados para a sessão seguem conforme Tabela 10:

Tabela 10: Parâmetros da sessão RTP

PARÂMETRO	VALOR
IP	(Dinâmico)
Porta	30000
Codificação	GSM_RTP
Sample Rate	8000 samples
Sample Size	8 bits
Channels	1

3.5.4 Inicialização dos agentes

O processo de inicialização dos agentes é uma tarefa simples, se limitando apenas pela execução de um comando que ativa simultaneamente os agentes *Caller* e *Callee*. O acompanhamento dos agentes ativos deve ser efetuado via *JADE Remote Agent Management GUI*, sendo essa interface fundamental para acompanhar não apenas os agentes citados, mas também todo funcionamento do agente gravador **RecorderAgent**. A Figura 22 apresenta a interface da ferramenta exibindo todos os agentes ativos durante o processo de gravação.

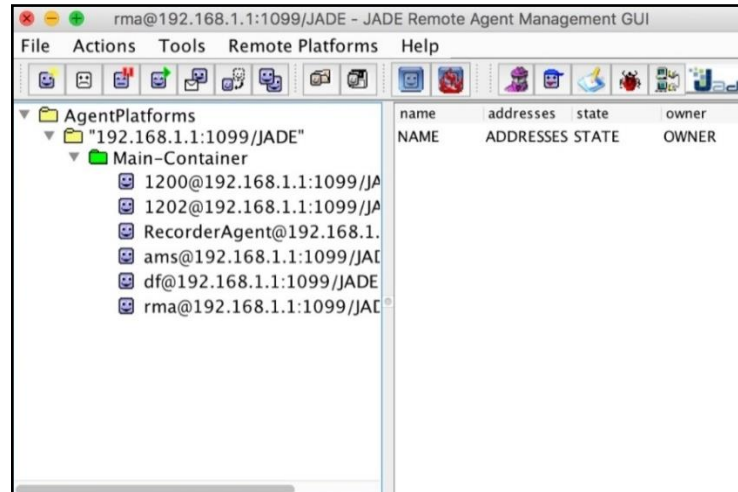


Figura 22: Interface do JADE Remote Agent Management

3.5.5 Processo de gravação

O processo de gravação envolve uma série de etapas como por exemplo a inicialização dos agentes, o estabelecimento das sessões SIP e RTP e o armazenamento em disco do conteúdo trocado entre as partes. A Figura 23 detalha como tais etapas ocorrem na prática.

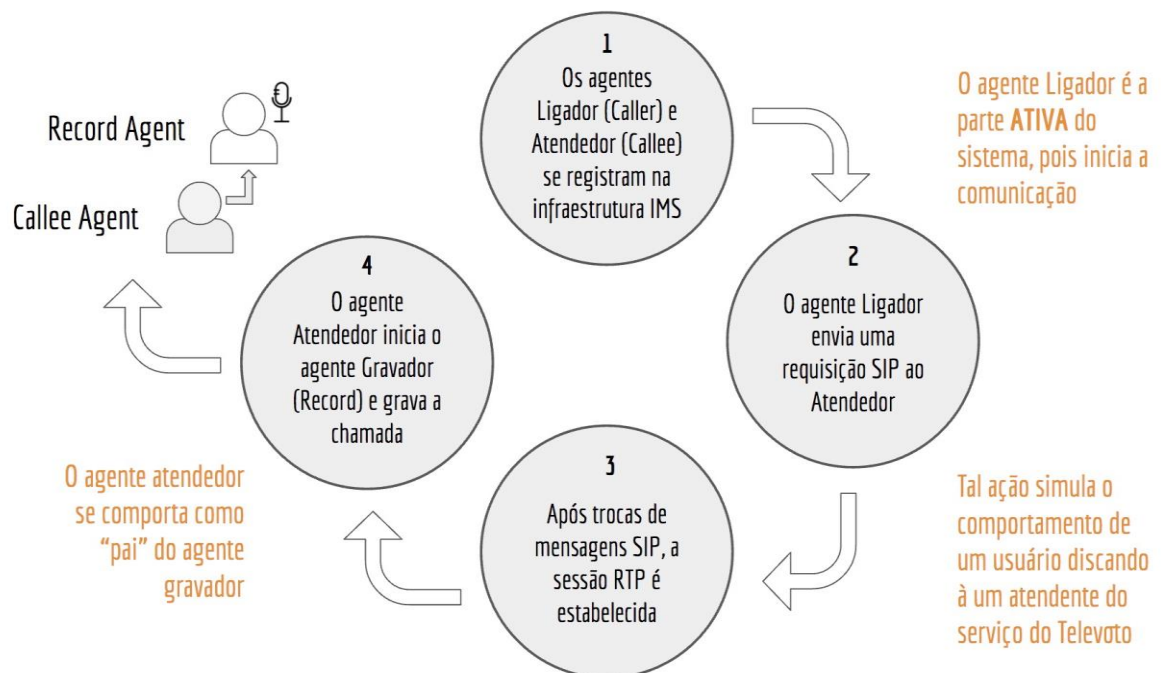


Figura 23: Etapas do processo de gravação

Na etapa 1 os agentes são inicializados, e a partir daí os mesmos se registram na infraestrutura IMS para poderem efetuar chamadas.

A etapa 2 consiste no início das ações de comunicação entre as partes. Nela o agente ligador se comporta como um elemento ativo no ambiente de trabalho, pois ele, assim que finaliza o seu registro no IMS, envia imediatamente uma requisição INVITE ao agente atendedor.

A etapa 3 detalha o momento no qual a sessão RTP é estabelecida. Esse processo é iniciado assim que o agente atendedor recebe a resposta ACK. Nesse instante, baseado na troca de mensagens contendo o protocolo responsável por descrever os parâmetros da sessão, o SDP, ambos participantes da comunicação são informados efetivamente sobre qual porta e como a comunicação deve ser estabelecida.

Por fim, a etapa 4 representa como a gravação é efetuada. Primeiramente o agente atendedor inicializa o agente gravador (Record Agent) que armazena o conteúdo da chamada em disco. Assim, após o tempo de 20 segundos de gravação, tal agente é desativado e a sessão RTP é encerrada.

Ao final das etapas apresentadas, o resultado da troca do conteúdo de mídia entre os agentes pode ser observado conforme Figura 24.

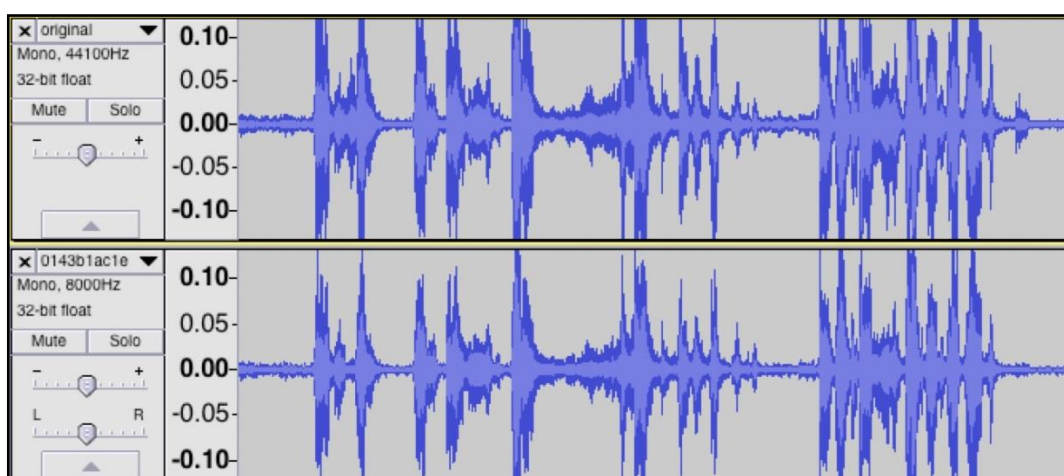


Figura 24: Sinais de áudio enviado e recebido, respectivamente

3.5.6 Monitoramento da sessão SIP

O monitoramento da sessão SIP foi realizado utilizando um programa denominado Wireshark. A execução do mesmo ocorre dentro da instância Linux denominada Sniffer, que por sua vez está inserida na rede do IMS sob IP 172.16.28.5. A ferramenta de monitoramento usada para analisar o tráfego de rede se mostrou fundamental para o entendimento de como, na prática, acontecem as trocas de mensagens entre os agentes *Caller* e *Callee* junto ao proxy da arquitetura IMS. A Figura 25 apresenta todo o tráfego SIP realizado na comunicação entre as partes.

No.	Time	Source	Destination	Protocol	Info
198	1464.017087	172.16.28.1	172.16.28.3	SIP	Request: REGISTER sip:ims.vi.ifes.edu.br
199	1464.017090	172.16.28.1	172.16.28.3	SIP	Request: REGISTER sip:ims.vi.ifes.edu.br
232	1464.115924	172.16.28.3	172.16.28.1	SIP	Status: 401 Unauthorized - Challenging the UE (0 bindings)
233	1464.117563	172.16.28.3	172.16.28.1	SIP	Status: 401 Unauthorized - Challenging the UE (0 bindings)
234	1464.150831	172.16.28.1	172.16.28.3	SIP	Request: REGISTER sip:ims.vi.ifes.edu.br
235	1464.150842	172.16.28.1	172.16.28.3	SIP	Request: REGISTER sip:ims.vi.ifes.edu.br
327	1464.278515	172.16.28.3	172.16.28.1	SIP	Status: 200 OK - SAR succesful and registrar saved (1 bindings)
363	1464.326215	172.16.28.3	172.16.28.1	SIP	Status: 200 OK - SAR succesful and registrar saved (1 bindings)
364	1464.353117	172.16.28.1	172.16.28.3	SIP/SDP	Request: INVITE sip:1200@ims.vi.ifes.edu.br, with session description
375	1464.354766	172.16.28.3	172.16.28.1	SIP	Status: 100 trying -- your call is important to us
400	1464.358200	172.16.28.3	172.16.28.1	SIP/SDP	Request: INVITE sip:1200@172.16.28.1:5062, with session description
401	1464.378874	172.16.28.1	172.16.28.3	SIP	Status: 180 Ringing
417	1464.379954	172.16.28.1	172.16.28.3	SIP/SDP	Status: 200 OK, with session description
445	1464.381767	172.16.28.3	172.16.28.1	SIP	Status: 180 Ringing
484	1464.384570	172.16.28.3	172.16.28.1	SIP/SDP	Status: 200 OK, with session description
485	1464.403199	172.16.28.1	172.16.28.3	SIP	Request: ACK sip:1200@172.16.28.1:5062
508	1464.405864	172.16.28.3	172.16.28.1	SIP	Request: ACK sip:1200@172.16.28.1:5062

Figura 25: Tráfego SIP na arquitetura IMS

Baseado na troca de mensagens SIP é possível fazer as seguintes observações:

- O processo de estabelecimento do registro e autenticação dos elementos *Caller* (1202@172.16.28.1:5064) e *Callee* (1200@172.16.28.1:5062) ocorre simultaneamente a partir das mensagens 198 até a 363.
- Todo tráfego entre os elementos registrados ocorre por meio do proxy do IMS sob o endereço 172.16.28.3 : 4060.
- A comunicação é encerrada a partir do recebimento da mensagem SIP BYE.

Além do monitoramento do tráfego SIP o Wireshark permite a observação de todos os protocolos presentes em cada pacote. Na Figura 26 foi detalhada a primeira

mensagem SIP INVITE trocada na comunicação, onde o agente *Caller* (1202@172.16.28.1:5064) convida o agente *Callee* (1200@172.16.28.1:5062) para iniciar a comunicação. Note que no corpo da mensagem enviada existe o protocolo SDP contendo os dados a serem usados pelo receptor para a devida parametrização RTP.

```

▷ Ethernet II, Src: Vmware_c0:00:02 (00:50:56:c0:00:02), Dst: Vmware_e8:6e:7f (00:0c:29:e8:6e:7f)
▷ Internet Protocol, Src: 172.16.28.1 (172.16.28.1), Dst: 172.16.28.3 (172.16.28.3)
▷ User Datagram Protocol, Src Port: ca-1 (5064), Dst Port: dsmeter_iatc (4060)
▽ Session Initiation Protocol
  ▷ Request-Line: INVITE sip:1200@ims.vi.ifes.edu.br SIP/2.0
  ▽ Message Header
    Call-ID: 5fc2e9b8ad26206e466457264dba0636@172.16.28.1
    ▷ CSeq: 865 INVITE
    ▷ From: <sip:1202@ims.vi.ifes.edu.br>;tag=456249
    ▷ To: <sip:1200@ims.vi.ifes.edu.br>
    ▷ Via: SIP/2.0/UDP 172.16.28.1:5064;rport;branch=z9hG4bK-343333-488c5cb959a55a0029d52fcb7e5be885
    Max-Forwards: 70
    ▷ Contact: <sip:ims.vi.ifes.edu.br@172.16.28.1:5064;transport=udp>
    Allow: INVITE, ACK, CANCEL, BYE, MESSAGE, OPTIONS, NOTIFY, PRACK, UPDATE, REFER
    Privacy: none
    P-Preferred-Identity: <sip:1202@ims.vi.ifes.edu.br>
    Route: <sip:proxy@172.16.28.3:4060;lr>,<sip:orig@scscf.ims.vi.ifes.edu.br:6060;lr>
    Content-Type: application/sdp
    Content-Length: 110
  ▽ Message Body
    ▽ Session Description Protocol
      Session Description Protocol Version (v): 0
      ▷ Owner/Creator, Session Id (o): - 3687555471 3687555471 IN IP4 172.16.28.1
      Session Name (s): -
      ▷ Connection Information (c): IN IP4 172.16.28.1
      ▷ Time Description, active time (t): 0 0
      ▷ Media Description, name and address (m): audio 30000 RTP/AVP 0

```

Figura 26: Corpo da mensagem SIP INVITE

3.5.7 Desafios e soluções encontradas

Durante o desenvolvimento do projeto uma série de desafios foram superados até o devido funcionamento da aplicação. A criação do gravador ocorreu sobre uma plataforma IMS já configurada, entretanto os agentes *Caller* e *Callee* tiveram de ser trabalhados de modo que seus registros na infraestrutura fossem efetuados com sucesso. O processo de autenticação de um terminal IMS é baseado no protocolo DIGEST. Basicamente ele é baseado no envio de credenciais em hashes criptográficas para evitar a transferência de senhas de autenticação via texto puro.

Todavia, todas essas ações são habilitadas apenas para usuários IMS previamente registrados na plataforma. Como o intuito do presente projeto se resumiu a desenvolver um gravador SIP, foram utilizados dois registros já existentes na plataforma, os usuários 1202 e 1200 que foram associados aos *Caller* e *Callee* respectivamente.

Além disso, outra dificuldade enfrentada foi a configuração da arquitetura da rede. Como a plataforma SANGN já possuía as definições de IP estáticos e um servidor de DNS, foi necessário compreender todos os encaminhamentos de pacotes dentro da rede. Esse processo, por estar em ambiente Linux, foi um tanto custoso para ser completamente assimilado.

Dentre os limitantes apresentados, aquele que mais se mostrou complexo para ser contornado, foi a tentativa de configuração de um servidor proxy RTP (RTPProxy). Esse processo necessitou de uma série de ajustes no servidor que levaram em risco o funcionamento de toda a aplicação IMS. Com isso foi desenvolvido um serviço REST para gerenciamento de portas com a mesma finalidade do proxy RTP, fato que contornou a dificuldade imposta por sua configuração.

3.5.7.1 Considerações do capítulo

O capítulo apresentado detalhou todo o processo de desenvolvimento do gravador SIP. As etapas que envolveram o detalhamento do ambiente onde o gravador será trabalhado, a definição dos requisitos e funcionalidades do mesmo, a compreensão da máquina de estado na qual os agentes descritos, foram fundamentais e viabilizaram as decisões referentes ao desenvolvimento dos componentes que integram ou auxiliam o gravador SIP como por exemplo o RecordAgent, RTPExport, o serviço RESTful e o cliente Web.

O processo de validação também foi amplamente abordado exibindo as estratégias adotadas para testar a aplicação desenvolvida. Etapas como a parametrização da sessão RTP justificaram, por exemplo, a importância do protocolo SDP para sincronização das sessões entre os terminais IMS. Além disso, o detalhamento do

tráfego da rede via Wireshark se mostrou muito esclarecedor sobre como realmente a comunicação ocorre entre os agentes.

Por fim, os principais problemas enfrentados foram abordados no intuito de mostrar a complexidade das questões envolvidas bem como justificar as ações adotadas para contorná-los.

4 TRABALHOS FUTUROS

Usando o gravador SIP como base, uma proposta de continuidade do trabalho apresentado consiste em trazer os componentes trabalhados para o real cenário da plataforma SANGN junto ao serviço do Televoto. Atualmente o projeto envolve uma execução isolada de dois agentes JADE que trocam mensagens SIP no intuito de validar o processo de gravação. Como essa etapa já foi superada, inserir o gravador no projeto desenvolvido por Ribeiro (2015) e Andrade (2016), irá agregar ainda mais valor ao trabalho desenvolvido por meio da adição das funcionalidades de gravação e gerenciamento das chamadas realizadas na plataforma.

A Figura 27 apresenta a arquitetura desenvolvida por Andrade (2016) cujos principais elementos são a Fila de Atendimento e o AS Representante. Tal estrutura possui a finalidade de alocar dinamicamente agentes atendedores para processamento das chamadas.

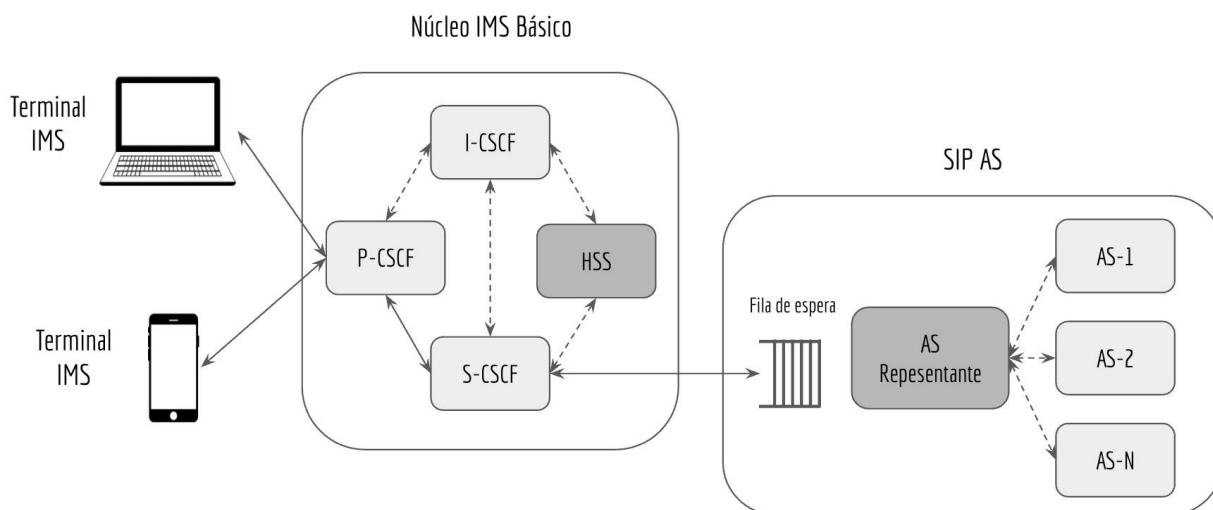


Figura 27: Arquitetura desenvolvida por Andrade (2016)

Nesse contexto, o gravador SIP poderá ser inserido conforme Figura 28. Tal distribuição de elementos é baseada na característica do gravador ser um agente JADE instanciado por um agente atendedor.

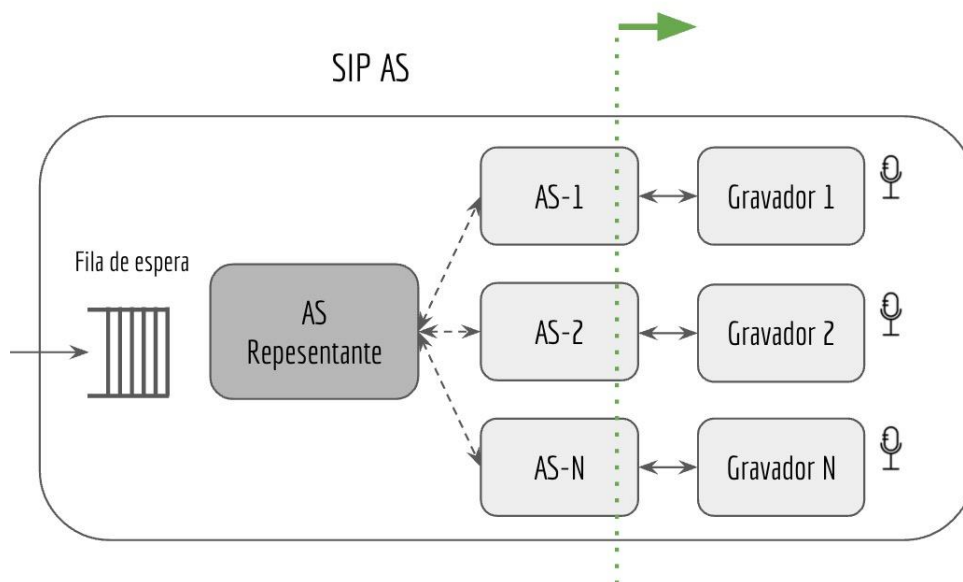


Figura 28: Inserindo o gravador SIP na arquitetura desenvolvida por Andrade (2016)

Uma outra adição ao presente projeto se resume à migração do gerenciamento das portas das sessões RTP para um nível de rede inferior, ou seja, trazendo a funcionalidade desse controle, que atualmente reside no serviço RESTful, para um servidor proxy (RTPProxy) responsável pela distribuição de portas entre os terminais IMS. Assim, tanto o estabelecimento das chamadas, como o estabelecimento das sessões RTP serão centralizados no OpenIMS, trazendo uma maior confiabilidade e eficiência à plataforma. Essa tentativa foi algo presente no desenvolvimento do projeto, mas, dada a dificuldade de configuração em ambiente Linux, o servidor proxy foi substituído pelo serviço RESTful.

Uma questão a ser abordada baseado na Figura 28 são os possíveis problemas que poderão surgir quanto ao espaço em disco para armazenamento, pois, conforme determinado pela legislação brasileira, o prazo de 6 meses de gravação de conteúdo deve ser respeitado, o que certamente irá representar valores expressivos em termos de dados.

5 CONCLUSÃO

O presente trabalho desenvolveu um gravador SIP no intuito de agregar ao serviço do Televoto, desenvolvido por Ribeiro (2015) e aprimorado por Andrade (2016), as funcionalidades de gravação de chamadas que, de acordo com a Lei do Marco Civil da Internet (BRASIL, 2004), são uma exigência para ferramentas que atuam no âmbito da internet.

Todos os objetivos apresentados no projeto tiveram como finalidade executar a gravação de chamadas SIP. Nesse processo, toda codificação trabalhada foi orientada à agentes, fato que irá viabilizar seu futuro acoplamento ao serviço do Televoto. Dentre as premissas definidas para gravação, o projeto executou com êxito todas as ações estabelecidas desde a gravação e registro das principais informações associadas à cada chamada efetuada na plataforma IMS e em seguida todo o conteúdo foi disponibilizado para consulta através do cliente Web desenvolvido. Ambos elementos trabalhados, tanto o gravador quanto o cliente Web, fizeram uso de outra ferramenta criada, o serviço RESTful, sendo esse responsável não apenas por fornecer as URLs de acesso ao conteúdo das ligações, mas também pelo gerenciamento das portas a serem utilizadas pelos agentes durante a transmissão e gravação das chamadas sob sessões RTP.

Durante o desenvolvimento do projeto, as etapas que envolveram o detalhamento do ambiente onde o gravador será trabalhado, a definição dos requisitos e funcionalidades do mesmo, a compreensão da máquina de estado na qual os agentes estiveram submetidos, foram fundamentais e viabilizaram as decisões referentes ao desenvolvimento dos componentes RecordAgent, RTPExport, o serviço RESTful e o cliente Web.

No contexto da crescente evolução dos sistemas de telecomunicação, o trabalho desenvolvido provou que a medida que novas demandas surjam nesse setor, soluções baseadas em serviços podem ser facilmente adotadas, aumentando assim os recursos e a confiabilidade de todo sistema.

BIBLIOGRAFIA

AHSON, S. A.; ILYAS, M. **IP Multimedia Subsystem (IMS) Handbook**. CRC Press, 2009.

AHSON, S. A.; ILYAS, M. **IP Multimedia Subsystem (IMS) Handbook**. Nova York: CRC Press, 2008. ISBN 978-1-4200-6459-9.

AL-BEGAIN, K. et al. **IMS: A Development and Deployment Perspective**. 1a. ed. [s.i.]. Wiley, 2009.

ANDRADE, J. C. S.; **Implementação e avaliação de serviços multimídia escaláveis em uma arquitetura OSGi/JADE utilizando o núcleo IMS**. Vitória: Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (UFES), 2016.

BELLIFEMINE, F.; GIOVANNI, C.; GREENWOOD, D. **Developing Multi Agent Systems with JADE**. John Wiley& Sons Ltd. Inglaterra. 2007.

BIEBERSTEIN, N.; et al. **Service-Oriented Architecture Compass: Business Value**, Planning and Enterprise Roadmap. IBM Press, 2006.

BRASIL. Lei nº 12.965, de 23 de Abril de 2014. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm

CAMARILLO, G.; GARCÍA-MARTÍN, A. M. **The 3G IP Multimedia Subsystem (IMS)**. 2a. ed. Torquary: Wiley, 2006.

CHANG, K; CHEN, C; CHEN, J.; CHAO, H. **Challenges to Next Generation Services in IP Multimedia Subsystem**. Journal of Information Processing Systems, June 2010, Volume 6, 10.3745/JIPS.2010.6.2.129.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000.

GRAHAM, I. **Requirements Modelling and Specification for Service Oriented Architecture**. Wiley, 2008.

HERSENT, O. **IP Telephony: Deploying VoIP Protocols and IMS Infrastructure**. 2a. ed. [s.i.]: Wiley, 2011.

IETF. **SIP: Session Initiation Protocol**. RFC 3261. 2002. Disponível em: <https://www.ietf.org/rfc/rfc3261.txt>

IETF. **Extension Mechanisms for DNS (EDNS0)**. RFC 2671. 1992. Disponível em: <https://www.ietf.org/rfc/rfc2671.txt>

IETF. **The MD5 Message-Digest Algorithm**. RFC 1321. Disponível em: <https://www.ietf.org/rfc/rfc1321.txt>

JUNIOR, S.; LUIZ, A. **Integração de Sistemas com JAVA**. Rio de Janeiro: Brasport, 2007.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: uma abordagem top-down**. 3a ed. Pearson Addison Wesley. São Paulo, 2006.

LGPL. **GNU General Public License**. Version 3. 2007. Disponível em: <https://www.gnu.org/licenses/gpl-3.0.en.html>

O GLOBO. **Facebook diz estar “desapontado” com prisão de vice-presidente em SP**. Disponível em: <http://g1.globo.com/sao-paulo/noticia/2016/03/facebook-diz-estar-desapontado-com-prisao-de-vice-presidente-em-sp.html>

OGUEJIOFOR, E. et al. **Developing SIP and IP Multimedia Subsystem (IMS) Applications**. 1a. ed. [s.i.]: IBM Redbooks, 2007.

OSGi Alliance. **OSGi Service Platform: Core Specification**, Release 4, Version 4.2. Technical report, 2009. Disponível em: <http://osgi.org/download/r4v42/r4.core.pdf>.

OSGi Alliance. **OSGi Service Platform: Core Specification**, Release 6. Specification. 2016, Disponível em: <https://osgi.org/javadoc/r6/core/index.html>

POIKSELKA, M.; MAYER, G. **The IMS: IP Multimedia Concepts and Services**. 3a. ed. Chichester: Willey, 2006. ISBN 978-0-470-72196-4.

RIBEIRO, R. B. C. **Escalabilidade de serviços multimídia de telecomunicações por meio do uso da arquitetura orientada a serviços OSGi integrada ao ambiente de Agentes JADE**. Vitória: Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (UFES), 2015.

RIBEIRO, R. B. C.; MARTINELLO, M.; SANTOS, C. A. S. **An Architectural Framework for Delivering SIP-AS Multimedia Services Based on JADE/OSGi Technology**. International Journal of Next-Generation Networks (IJNGN), 6, n. 3, Setembro 2014.

RICHARDSON, L.; RUBY, S. **RESTful Web Services**. [s.i: s.n.]. p. 433

RIBEIRO, R. B. C.; SOARES, R. B. **The application of JADE and OSGi technologies in the telecommunications services architecture**. International Conference on Computer as a Tool (EUROCON), Lisboa, Abril 2011.

RUSSEL, T. **The IP Multimedia Subsystem - Session Control and Other Network Operations**. Nova York: McGraw Hill, 2008. ISBN 0-07-159464-7.

TEIXEIRA, F. V. **JADE - Java Agent Development framework**, [s.d.].

TURNER, K. J. M.; Marples, E. H.; DAVID, J. **Service Provision: Technologies for Next Generation Communications**. John Wiley & Sons Ltd, 2004.