

**FUNDAÇÃO, CENTRO DE ANÁLISE PESQUISA E INOVAÇÃO TECNOLÓGICA
FACULDADE FUCAPI (INSTITUTO DE ENSINO SUPERIOR FUCAPI)
COORDENAÇÃO DE GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO**

MARCOS ROBERTO GARCIA BAHIENSE JÚNIOR

**CALL CENTER AUTOMATIZADO: integrando um PABX baseado em asterisk
com o GSAN**

**MANAUS
2015**

MARCOS ROBERTO GARCIA BAHIANSE JÚNIOR

**CALL CENTER AUTOMATIZADO: integrando um PABX baseado em asterisk
com o GSAN**

Monografia apresentada ao curso de graduação em Sistemas de Informação da Facupi (Instituto de Ensino Superior Fucapi), como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação Área de Concentração: Desenvolvimento e Análise de software.

Orientador: Prof Carlos Augusto Mar, M.Sc.

**MANAUS
2015**

MARCOS ROBERTO GARCIA BAHIANSE JÚNIOR

**CALL CENTER AUTOMATIZADO: integrando um PABX baseado em asterisk
com o GSN**

Monografia apresentada ao curso de graduação em Sistemas de Informação da Facupi (Instituto de Ensino Superior Fucapi), como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação Área de Concentração: Desenvolvimento e Análise de software.

Aprovada em 14/12/2015, por:

Prof Carlos Augusto Mar, M.Sc.
Faculdade Fucapi
Orientador

Prof. Sergio Roberto C. Vieira, M.Sc.
Faculdade Fucapi
Examinador

Profª Kelen Acquati, M.Sc.
Faculdade Fucapi
Examinadora

MANAUS
2015

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me privilegiar com a vida;

A minha esposa Karoline por ter tido paciência e compreensão durante o período que tive

que me dedicar a este trabalho e não me deixar desistir em meio as dificuldades;

Aos meus familiares por sempre me apoiarem;

Aos meus amigos por proporcionarem momentos de descontração;

Ao meu amigo Wellington que me fez enxergar o potencial deste trabalho;

Ao meu orientador Carlos Mar que acreditou que seria possível e muito ajudou na concretização deste trabalho.

RESUMO

Este trabalho apresenta um dos principais sistemas utilizado para gestão de operações comerciais e controle de execução de serviços do setor de Saneamento Básico Brasileiro, o sistema GSAN, com melhorias no que diz respeito ao Atendimento ao Público, resultado da padronização dos atendimentos de primeiro nível e automatização dos atendimentos dos serviços Obter 2ª via de Conta, Informar Falta de Água e Solicitar Restabelecimento da Ligação de Água, integrados a uma central de atendimento personalizada através da ferramenta Asterisk, que permite a utilização de voz sobre IP, além do uso convencional da telefonia pública como meio de comunicação com o cliente. Realizado para possibilitar a redução dos custos com atendimento ao cliente e pelo fato das empresas de saneamento serem altamente demandada pela população diariamente. Após o estudo aprofundado dos sistemas envolvidos, foi possível identificar uma forma de integrar as tecnologias de paradigmas diferentes, com a utilização de um *Middleware* intermediário responsável pela comunicação via protocolo SOAP e a interface de comunicação AGI, respectivamente para interligar os sistemas GSAN e Asterisk. Foram realizados experimentos sobre o produto gerado, após a aplicação dos diversos cenários de testes foi demonstrado uma automatização dos processos burocráticos no registro de atendimentos que favorece a uma possível redução nos custos de atendimentos.

Palavras-chaves: GSAN. Asterisk. Call Center. Middleware. Saneamento Básico Brasileiro.

ABSTRACT

This work presents one of the main systems used for managing business operations and execution control services of the Brazilian basic sanitation sector, GSAN system, with improvements with regard to the Public Service as a result of standardization of top-level visits and automation of care services second copy account, Inform Water Lack and Request Restoration of connection, integrated to a central personalized service through Asterisk tool, which allows the use of voice over IP in addition to the conventional use of public telephony as a means communication with the client. Carried out to enable the reduction of customer service costs and because the sanitation companies are highly requested by the population daily. After thorough study of the systems involved, it was possible to identify a way to integrate the different paradigms technologies, using an intermediary Middleware responsible for communicating via SOAP protocols and a communication interface AGI respectively to interconnect GSAN and Asterisk systems. Experiments were carried out on the product generated, after the implementation of the various scenarios testing was demonstrated one automation of bureaucratic processes in the registration requests favoring a possible reduction in care costs.

Key-words: GSAN. Asterisk. Cell Center. Middleware. Brazilian basic sanitation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Implantações do Sistema GSAN	16
Figura 2 – Arquitetura Detalhada do Sistema GSAN	19
Figura 3 – Exemplo de configuração do <i>script</i> build	21
Figura 4 – Execução do <i>script</i> de build	21
Figura 5 – Executando o sistema GSAN	22
Figura 6 – Acessando página inicial do Sistema	23
Figura 7 – Acessando página inicial da Interface WEB do Disc-OS	26
Figura 8 – Diagrama de implantação da solução	32
Figura 9 – Interface dos serviços automatizados.	35
Figura 10 – Declaração da servlet do Webservice.	35
Figura 11 – Declaração do <i>EndPoint</i> dos serviços.	36
Figura 12 – Mapeamento dos serviços para consumo via AGI	37
Figura 13 – Fluxo de Identificação de Cliente no Middleware.	37
Figura 14 – Geração do código fonte para consumo do Webservice.	38
Figura 15 – Executando o sistema Integrador.	39
Figura 16 – Cadastro de um Ramal SIP.	40
Figura 17 – Diagrama do Fluxo da Unidade de Resposta Audível	41
Figura 18 – Fluxo de Identificação de Cliente no Asterisk.	42
Figura 19 – Configurar Ramal no Zoiper	43
Figura 20 – Diagrama de sequência utilizando a suíte de teste.	45
Figura 21 – Declaração dos contextos de teste no Asterisk	48
Figura 22 – Obter 2ª via - Cenário de Teste 1	49
Figura 23 – Obter 2ª via - Cenário de Teste 2	50
Figura 24 – Obter 2ª via - Cenário de Teste 3	50
Figura 25 – Informar Falta de Água - Cenário de Teste 1	51
Figura 26 – Informar Falta de Água - Cenário de Teste 2	51
Figura 27 – Informar Falta de Água - Cenário de Teste 3	52
Figura 28 – Restabelecimento da Ligação de Água - Cenário de Teste 1 . .	53
Figura 29 – Restabelecimento da Ligação de Água - Cenário de Teste 2 . .	53
Figura 30 – Restabelecimento da Ligação de Água - Cenário de Teste 3 . .	54
Figura 31 – Obter 2ª Via de Conta - Detalhes execução dos testes	54
Figura 32 – Obter 2ª Via de Conta - E-mail recebido pelo cliente	55
Figura 33 – Informar Falta de Água - Detalhes execução dos testes	55
Figura 34 – Informar Falta de Água - RA gerado para o Cenário 1	56
Figura 35 – Informar Falta de Água - RA gerado para o Cenário 2	56

Figura 36 – Informar Falta de Água - RA gerado para o Cenário 3	57
Figura 37 – Restabelecimento da Ligação de Água - Detalhes execução dos testes	57
Figura 38 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 1	58
Figura 39 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 2	58
Figura 40 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 3	59

LISTA DE QUADROS

Quadro 1 – Principais atributos do imóvel	17
Quadro 2 – Tecnologias utilizadas	33
Quadro 3 – Recursos computacionais utilizados	47

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	GSAN	16
2.1.1	GSAN CONCEITOS	17
2.1.2	GSAN ARQUITETURA	19
2.1.3	GSAN CONFIGURAÇÃO	20
2.1.4	GSAN - DETALHAMENTO TÉCNICO	23
2.2	ASTERISK	24
2.2.1	ASTERISK CONCEITOS	24
2.2.2	ASTERISK INSTALAÇÃO	25
2.2.3	ASTERISK - DETALHAMENTO TÉCNICO	26
2.3	TRABALHOS RELACIONADOS	27
2.4	TECNOLOGIAS UTILIZADAS	28
2.4.1	SIMPLE OBJECT ACCESS PROTOCOL	28
2.4.2	ASTERISK GATEWAY INTERFACE	28
2.4.3	WEBSERVICES	29
2.4.4	UNIDADE DE RESPOSTA AUDÍVEL	29
2.4.5	MIDDLEWARE	29
2.4.6	DISC-OS	29
2.4.7	CODEC	29
2.4.8	JUNIT FRAMEWORK	29
3	MIDDLEWARE: GSAN E ASTERISK	31
3.1	ARQUITETURA DA SOLUÇÃO	31
3.2	ETAPAS DA INTEGRAÇÃO	33
3.2.1	IMPLEMENTAÇÃO DE WEBSERVICES NO GSAN	33
3.2.2	IMPLEMENTAÇÃO DO MIDDLEWARE	36
3.2.3	CUSTOMIZAÇÃO DO ASTERISK	39
3.2.4	DESENVOLVIMENTO DA SUÍTE DE TESTES AUTOMATIZADOS	43
4	DEMONSTRAÇÃO DA SUÍTE DE TESTES AUTOMATIZADOS	47
4.1	PREPARAÇÃO DO AMBIENTE DE TESTE	47
4.2	CENÁRIOS DE TESTE	48
4.2.1	OBTER 2ª VIA DE CONTA	49
4.2.2	INFORMAR FALTA DE ÁGUA	50
4.2.3	SOLICITAR RESTABELECIMENTO DA LIGAÇÃO DE ÁGUA	52

4.3	EXECUÇÃO DOS CENÁRIOS DE TESTE	54
4.3.1	OBTER 2ªVIA DE CONTA	54
4.3.2	INFORMAR FALTA DE ÁGUA	55
4.3.3	SOLICITAR RESTABELECIMENTO DA LIGAÇÃO DE ÁGUA . . .	57
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	60
5.1	CONSIDERAÇÕES FINAIS	60
5.2	TRABALHOS FUTUROS	60
	REFERÊNCIAS	62
	APÊNDICES	63
	APÊNDICE A – PROCESSO DE CONFIGURAÇÃO DA JDK NA IDE DE DESENVOLVIMENTO	64
	APÊNDICE B – CONFIGURAR VARIÁVEL DE AMBIENTE DO SERVIDOR DE APLICAÇÃO JBOSS	67
	APÊNDICE C – CONFIGURAÇÃO DO FLUXO DA URA	68
	APÊNDICE D – REGISTROS DE ATENDIMENTOS GERADOS .	72

1 INTRODUÇÃO

A Secretaria Nacional de Saneamento Ambiental (SNSA) que integra o Ministério das Cidades (MC) visando elevar o nível de desempenho e eficiência das empresas de abastecimento de água e coleta de esgoto teve a iniciativa de promover o desenvolvimento de um software que pudesse atender as necessidades básicas do setor de saneamento de um modo geral.

Por meio do Programa de Modernização do Setor de Saneamento (PMSS, 2015) efetuou a contratação de uma empresa de tecnologia da informação brasileira para executar o projeto concebido. Nesse cenário, surge então o sistema GSAN¹ que se trata de um sistema desenvolvido com tecnologias de software livre, para a gerência de operações comerciais e de controle da execução de serviços internos das companhias de saneamento, o software atualmente encontra-se disponível gratuitamente no portal do software público brasileiro (PORTAL DO SOFTWARE PÚBLICO BRASILEIRO, 2007).

Mesmo com a modernização do setor de saneamento através do sistema GSAN que atualmente está implantado em 10 companhias estaduais das quais 4 estão em processo de migração (PMSS, 2015), ainda há grandes desafios a serem superados e um deles será abordado neste trabalho.

O Atendimento ao público trata-se de uma das frentes que as empresas de saneamento necessitam disponibilizar aos seus clientes. Muitas das vezes o valor envolvido em manter disponível uma infraestrutura que atenda a necessidade da empresa, com equipes de *Call Center* ou mesmo com atendimento presencial, podem gerar custos astronômicos dependendo da quantidade e qualidade da mão de obra contratada, aquisição de licenças para soluções proprietárias entre outros fatores que podem contribuir para variação do valor.

Atualmente o uso de software *Open Source* nas empresas tem se tornado bastante comum (MEIRELLES, 2014), com intuito de apoiar o negócio, como é o caso do software Asterisk que implementa facilidades no uso de tecnologias como PABX² tanto para linhas telefônicas convencionais como também por VoIP³ que utiliza a transmissão de voz sobre um rede IP⁴ com padrão de qualidade de serviço (QoS), permitindo a utilização de URA (Unidade de Resposta Audível) (VIEIRA, 2007) como linha de frente no atendimento ao cliente.

¹ GSAN - Sistema Integrado de Gestão de Serviços de Saneamento.

² PABX - *Private Automatic Branch Exchange*.

³ VoIP - *Voice over Internet Protocol*

⁴ IP - *Internet Protocol*

Problema

Atualmente o GSAN, atende grande parte das companhias de saneamento brasileiras, como é o caso de companhias como por exemplo CAERN⁵, CAER⁶, COMPESA⁷, MANAUS AMBIENTAL entre outras citadas no seguinte referencial teórico 2.1.

Essas empresas fazem uso do sistema para gerenciar as suas informações operacionais e gerenciais, que de certa forma atende as demandas internas. Entretanto, no aspecto do atendimento público, existem lacunas que ainda precisam ser atendidas de forma plena, principalmente por se tratarem de organizações altamente demandadas pela população, sendo responsáveis por atender diversos tipos de clientes que variam desde pequenos vilarejos até grandes metrópoles.

A falta de padronização nos atendimentos, o grande fluxo de transferência entre ramais e a variação nos tempos de atendimentos são muito comuns, pelo fato de todo atendimento de Primeiro Nível⁸ normalmente ser realizado por pessoas ou PA (Posto de Atendimento), ou seja, são recursos caros. Normalmente os atendentes respondem por um determinado setor da empresa, encarregado em solucionar tipos específicos de problemas, possibilitando muita das vezes a realização de transferência para outros ramais até que o cliente consiga concluir uma solicitação, o que pode gerar desconforto e insatisfação com os serviços de atendimento.

Existe uma grande dificuldade das empresas de saneamento, em disponibilizar uma estrutura de *Call Center* que atenda as expectativas dos clientes.

A dificuldade em manter um *feedback* rápido com o cliente, a falta de canais de comunicação flexíveis que permitam uma disponibilidade maior inclusive fora do horário comercial, a ineficiência na triagem dos atendimentos, são questões rotineiras enfrentadas no cotidiano das empresas de saneamento.

OBJETIVO

A finalidade deste trabalho propõe atender ao objetivo geral e aos objetivos específicos descritos a seguir.

OBJETIVO GERAL

Desenvolver uma integração do sistema GSAN com a ferramenta de PABX chamada Asterisk para permitir o atendimento automático destinados aos serviços Obter 2ª via de conta, Informar falta de água e Solicitar restabelecimento da ligação de água.

⁵ CAERN - Companhia de Água e Esgotos do Rio Grande do Norte

⁶ CAER - Companhia de Água e Esgotos de Roraima

⁷ COMPESA - Companhia Pernambucana de Saneamento

⁸ Atendimento de Primeiro Nível Refere-se a recepção inicial de todo atendimento.

OBJETIVO ESPECÍFICO

Para alcance deste objetivo, fazem necessários os seguintes passos:

- Implementar a integração entre os sistemas GSAN e Asterisk.
- Elaborar cenários de teste para os serviços Obter 2ª via de conta, Informar falta de água e Solicitar restabelecimento da ligação de água.
- Desenvolver uma suíte de testes automatizados com foco na integração dos sistemas.
- Contribuir com a comunidade de software livre, disponibilizando as novas funcionalidades desenvolvidas do sistema GSAN.

JUSTIFICATIVA

A integração entre sistema GSAN com uma ferramenta de PABX será um experimento de cunho prático, realizado para atender a uma demanda do setor de saneamento brasileiro, que atualmente sofre com a dificuldade em fornecer uma comunicação que atenda as expectativas dos clientes através de seu sistema de informação principal.

Com base nas informações disponibilizadas no Relatório de Análise Regulatória da Companhia de Águas de Joinville (CAJ) (JOINVILLE, 2014) situada em no estado de Santa Catarina, divulgado em 2014, demonstra a ineficiência enfrentada pelo setor de saneamento no que diz respeito ao Atendimento ao Público, a companhia considerada universalizada por atender mais 99% da população urbana com abastecimento de água, somando um total de aproximadamente 508.097 habitantes no município, atualmente enfrenta um número acentuado de reclamações.

O indicador de Número de Reclamações em 2013 apresentou a média anual de 13,78 reclamações/mil ligações, que reflete na média anual do tempo de espera das ligações que no mesmo ano registrou 75,2 segundos por atendimento. Com uma quantidade notória de reclamações diárias se torna custoso atender todas solicitações individualmente utilizando somente atendentes sem que haja otimizações nos atendimentos, deixando evidente o quão necessário se faz adotar medidas para melhorar os sistemas de *Call Center*.

Conforme informações disponíveis no Sistema Nacional de Informações do Setor de Saneamento (SNIS) (SNIS, 2015), especificamente a Região Norte do país possui um dos piores índices de perda de faturamento do país, consequentemente gera lucros menores e enfrenta dificuldade na ampliação do acesso à população aos serviços de saneamento, dificultando ainda mais investimentos por parte das companhias em tecnologias renovadores para o setor de saneamento.

Visando propor soluções viáveis que possam agregar valor à empresa sem acarretar em custos elevados, utilizando de soluções em software *Open Source* com tecnologias

compatíveis, é possível tornar o próprio sistema principal de uma empresa de saneamento o GSAN, capaz de suprir através dos recursos da ferramenta Asterisk, as necessidades de melhoria no atendimento ao público realizado nas Centrais de Atendimento.

Consequentemente favorece a redução de custos e propicia ao cliente final um melhor e mais efetivo relacionamento com a empresa prestadora de serviço.

MÉTODO DE INVESTIGAÇÃO

A metodologia utilizada para realização do presente trabalho foi dividida da seguinte forma:

- Pesquisa bibliográfica para obter o embasamento teórico sobre funcionamento dos sistemas envolvidos.
- Identificação de uma possível forma de integração entre ambos.
- Desenvolvimento da integração entre os sistemas.
- Desenvolvimento de uma suíte de testes automatizados.
- Experimentação utilizando a suíte de testes automatizados.

ESTRUTURAÇÃO DA MONOGRAFIA

Após este capítulo introdutório, que basicamente visa contextualizar e caracterizar o tema de pesquisa, o trabalho realizado foi dividido em cinco capítulos, conforme descrito abaixo:

Capítulo 2 – Fundamentação Teórica – Este capítulo tem como objetivo apresentar os principais conceitos dos sistemas e *frameworks* utilizados como base no desenvolvimento deste trabalho.

Capítulo 3 – Middleware: GSAN e Asterisk – Trata-se da implementação realizada para integração entre os sistemas, apresentando as principais etapas para elaboração da comunicação entre os sistemas.

Capítulo 4 – Experimentação Utilizando a Suíte de Testes Automatizados – Tem como característica a preparação do ambiente de teste, o planejamento e execução das experimentações utilizando a suíte de testes automatizados para validar a integração entre os sistemas GSAN e Asterisk.

Capítulo 5 – Considerações Finais e Trabalhos Futuros – Finalmente, no quinto capítulo, apresentam-se a conclusão que foi obtida e as recomendações para trabalhos futuros reunindo os comentários finais deste trabalho de pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

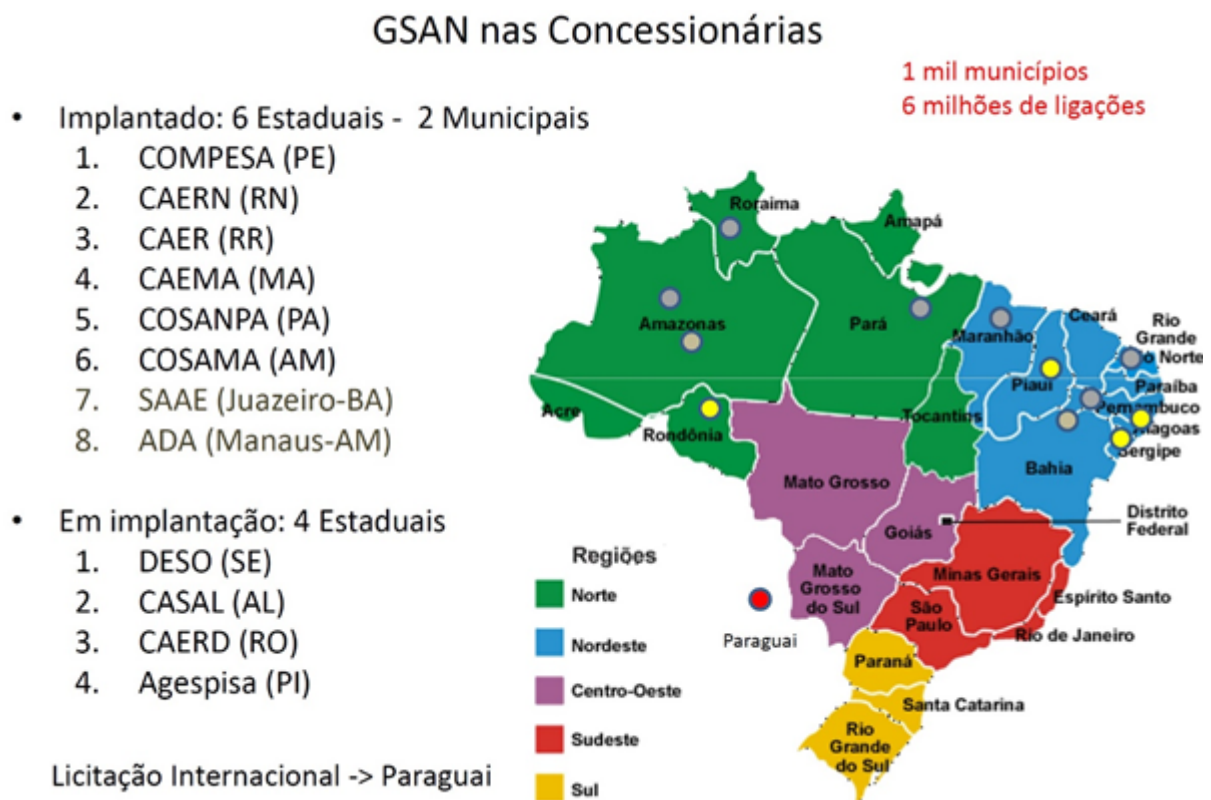
Neste capítulo será apresentado o conceito dos sistemas e frameworks utilizados neste trabalho, com a intenção de familiarizar o leitor com as notações que serão amplamente utilizadas no decorrer do trabalho.

2.1 GSAN

O sistema de código aberto GSAN (Sistema Integrado de Gestão de Serviços de Saneamento) desenvolvido inicialmente pela empresa IPAD (Instituto de Planejamento e Apoio ao Desenvolvimento Tecnológico e Científico) em 2005 por meio do Programa de Modernização do Ministério das Cidades, atualmente mantido e disponibilizado pelo Portal de Software Livre Brasileiro, sendo constantemente melhorado e aperfeiçoado pelos prestadores de serviços e interessados.

Propõem-se em atender as principais demandas de gestão de operações comerciais e controle de execução de serviços nas companhias de saneamento do Brasil, que atualmente utilizam o sistema em grande escala, conforme exposto a seguir pela figura 1:

Figura 1 – Implantações do Sistema GSAN



Fonte: Portal do software Público Brasileiro (2007).

Conforme demonstrado acima o sistema está em funcionamento em aproximadamente 1 mil municípios brasileiros, concentrado principalmente na região Norte e Nordeste do Brasil. Atualmente o sistema está preparado para atender companhias de pequeno e médio porte, provendo soluções flexíveis através de parametrizações em tabelas de banco de dados, adequando-se a realidades distintas e tornando-se referência em software para o setor de saneamento básico brasileiro.

2.1.1 GSAN CONCEITOS

O sistema em sua concepção foi dividido nos seguintes módulos descritos abaixo:

- **Atendimento ao Público:** Responsável principalmente em fornecer acesso rápido as informações dos clientes/imóveis e possibilita o registro dos atendimentos realizados.
- **Cadastro:** Responsável em permitir a inserção, alteração e exclusão das entidades básicas só sistema.
- **Micromedição:** Aborda as regras de consistência e análise de leitura e consumo, provendo meios para verificar o comportamento do consumo de água do imóvel.
- **Faturamento:** Responsável principalmente em realizar o cálculo para precificar o consumo e gerar/emitir as faturas.
- **Arrecadação:** Disponibiliza meios para efetuar a baixa de débitos dentre outras rotinas dessa natureza.
- **Segurança:** Possibilita a gestão sobre as permissões de usuários/funcionários.
- **Cobrança:** Fornece meios de emitir formas de cobranças personalizadas.
- **Contabilização:** Realiza a contabilidade e possibilita integrações para sistemas externos.

Alguns dos conceitos de saneamento serão abordados neste trabalho, portanto faz-se necessário conhecer e entender como o sistema GSAN aborda essas questões.

O imóvel no sistema deve possuir relação com os seguintes itens (Tabela 1):

Quadro 1 – Principais atributos do imóvel

Atributo	Descrição	Obrigatório
Localidade	Um conjunto populacional	Sim
Setor Comercial	Conjunto de quadras, semelhante ao bairro	Sim
Quadra	Denominado normalmente de "quarteirão"	Sim
Lote	Uma subdivisão da Quadra	Sim
Sub-Lote	Uma subdivisão da Lote	Sim

Fonte: Autoria Própria.

Tais relacionamentos são necessários para constituir a matrícula do imóvel, que forma um código único e representa a localização exata do imóvel.

Matrícula: [Localidade].[Setor Comercial].[Quadra].[Lote].[Sub-Lote]

Ex: 001.015.080.0120.001

O imóvel pode conter uma Ligação seja ela de Água, Poço e Esgoto, após o cliente realizar o cadastro do imóvel e solicitar a ligação, normalmente quando o imóvel obtém uma Ligação de Água ou Poço, é cobrado uma taxa mensal referente ao Esgoto variando em muito dos casos de 80% a 100% do valor a ser cobrado pelo consumo de água. Existem as possíveis situações para a Ligação do imóvel:

- **Ligado:** Imóvel está conectado à rede de distribuição de água.
- **Potencial:** Imóvel está localizado fora do alcance da rede de distribuição de água.
- **Factível:** Imóvel está localizado dentro do alcance da rede de distribuição de água, mas que nunca esteve conectado a ela.
- **Cortado:** Imóvel que possui um dispositivo de vedação do fluxo de água no intuito de interromper o abastecimento.
- **Suprimido:** Imóvel que teve o ramal de água retirado para a interrupção definitiva do abastecimento de água.

O relacionamento entre Clientes e Imóveis no sistema pode ocorrer das seguintes formas:

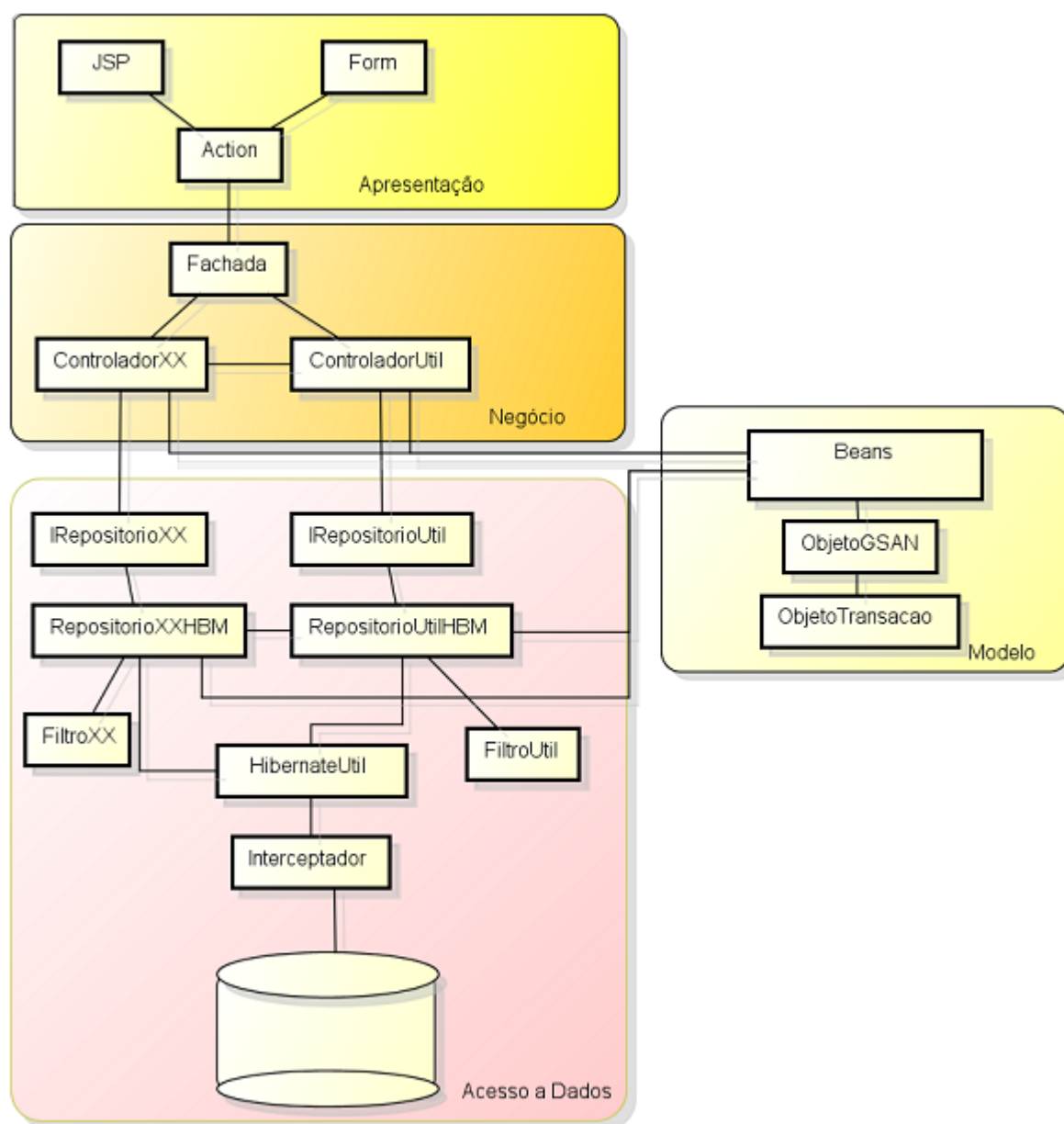
- **USUÁRIO** – Pessoa que reside no imóvel.
- **PROPRIETÁRIO** – Pessoa que possui a propriedade do bem de direito.
- **RESPONSÁVEL** – Pessoa responsável pelo pagamento de débitos do imóvel.

As solicitações realizadas pelos clientes aos atendentes são denominadas Registros de Atendimentos pelo sistema, comumente chamada de RA, para cada tipo de solicitação seja ela Solicitar Ligação, Solicitar Corte, Informar Falta de Água entre outras, o sistema possui tipos de serviços que devem ser realizados para atender à solicitação, a execução destes serviços é representado por uma entidade denominada Ordem de Serviço, comumente chamada de OS, o sistema permite que seja configurada a criação de OS automáticas para determinadas solicitações, tornando menos burocrático a formalização das reclamações recebidas.

2.1.2 GSAN ARQUITETURA

O Sistema GSAN foi desenvolvido fundamentalmente utilizando a plataforma JEE¹, propriedade da *Oracle Corporation*, em sua versão 5, na época a mais recente. Utiliza os principais serviços e tecnologias oferecidos pela plataforma, como por exemplo, EJB², JMS³ e JSP⁴ 2.1. O GSAN possui uma arquitetura que implementa diversos padrões de projeto, visando facilitar a manutenibilidade e manter a organização dos componentes, segue abaixo o diagrama de componentes conforme a figura 2:

Figura 2 – Arquitetura Detalhada do Sistema GSAN



Fonte: Autoria Própria.

¹ JEE - *Java Enterprise Edition*

² EJB - *Enterprise Java Beans*

³ JMS - *Java Message Service*

⁴ JSP - *Java Server Pages*

A camada de apresentação utiliza recurso nativo da plataforma JEE para Web, sendo a JSP para construção dos layouts e páginas a serem exibidas, juntamente com o *framework* Apache Struts versão 1.2 atuando como controlador das requisições, além de Javascript e CSS para tratar o comportamento e aparência das páginas. A fachada é um ponto de comunicação entre as camadas de apresentação e a camada de negócio e implementa o padrão de projeto *Singleton*⁵. A principal função da fachada é centralizar todas as chamadas de métodos da camada de negócio para que outras aplicações ou outras camadas superiores possam utilizar seus serviços. As Classes de Controladores EJB são responsáveis por garantir toda a regra de negócio do sistema, elas são implementadas utilizando a especificação do *Enterprise Java Beans* versão 2.1. As classes de Repositório são classes da aplicação que utilizam o padrão de projeto *Singleton*, a responsabilidade desta classe consiste em assegurar que todos os métodos de persistência ou serviço de consulta com o banco de dados.

O sistema GSAN foi projetado para ser independente da solução de Banco de Dados utilizada, acoplado ao *framework* Hibernate que trata da persistência Objeto/Relacional (ORM), possibilita o isolamento da camada de Persistência.

A aplicação dos Padrões de Projetos renomados, tornar o código mais organizado e entendível, facilitando futuras manutenções, a utilização do padrão MVC⁶ como estrutura arquitetural faz com que exista isolamento entre as camadas de Modelo, Visualização e Controle da aplicação, tornando a organização de pacotes bem estruturada.

2.1.3 GSAN CONFIGURAÇÃO

A configuração do ambiente de desenvolvimento se trata de um passo fundamental para execução deste trabalho prático. Primeiramente será preciso obter a versão do sistema que se encontra disponível no site do Portal do Software Livre, que atualmente disponibiliza o código fonte do sistema GSAN e demais arquivos de configuração do ambiente, no *github*⁷ para a comunidade de desenvolvedores e interessados.

Com o código fonte em mãos é necessário o auxílio de uma IDE⁸ de desenvolvimento para realizar a manutenção e construção dos novos serviços, foi utilizado neste trabalho a IDE Eclipse Juno para realizar esta tarefa, o processo de configuração da IDE pode ser visto descrito nos anexos deste trabalho.

O processo de empacotamento para geração do EAR (*Enterprise Archive*) para disponibilização, utiliza a ferramenta Apache Ant versão 1.6.2, normalmente a versão disponibilizada pela comunidade possui *script* de *build*⁹ para serem executados, no entanto

⁵ *Singleton* refere-se ao Padrão de Projeto que garante a existência de somente uma instância de determinada classe.

⁶ MVC - *Model View Controller*

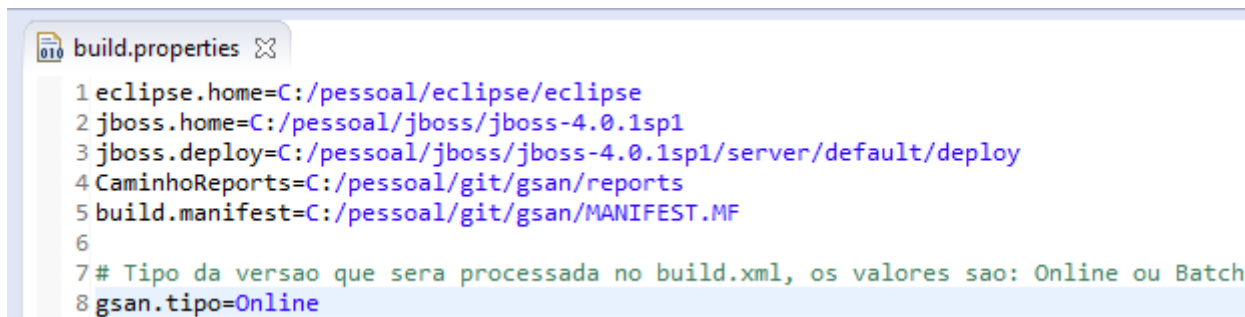
⁷ Github - Comunidade de código fonte.

⁸ IDE - *Integrated Development Environment*

⁹ *Scripts* de *build* refere-se a instruções para realizar o empacotamento da aplicação.

é preciso configurar os locais adequados para geração do pacote, conforme segue o exemplo abaixo na figura 3:

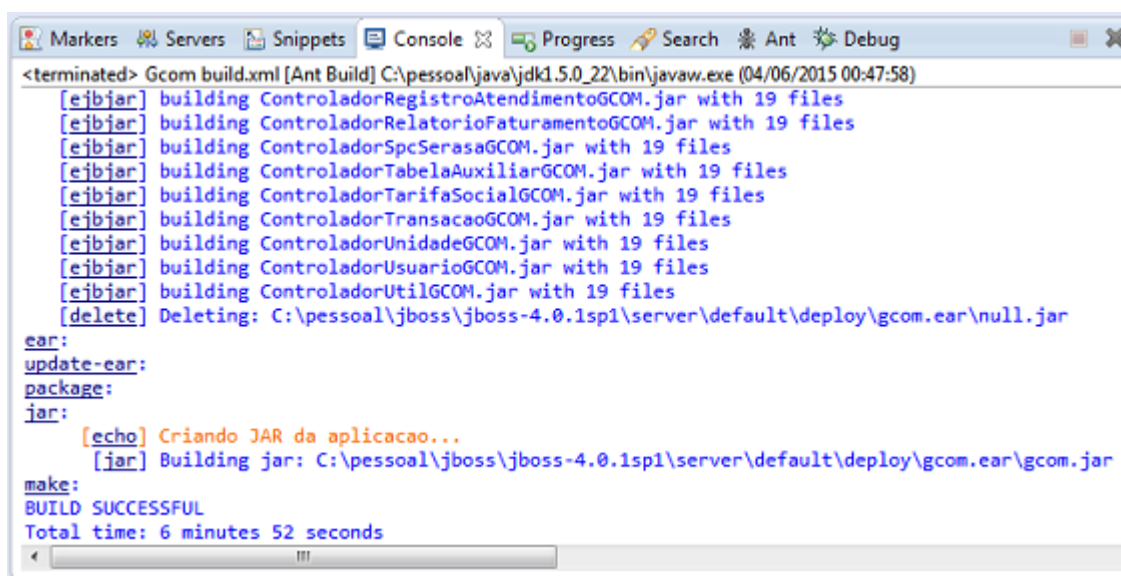
Figura 3 – Exemplo de configuração do *script* build



Fonte: Autoria Própria.

A execução do *script* pode ser realizada dentro da IDE executando o seguinte procedimento, após localizar o arquivo *build.xml* dentro na raiz do projeto, ao clicar com o botão esquerdo e selecionar a opção *Run as > Ant Build*, será acionado a execução da instrução *make* padrão do *script*, para construção do pacote a ser disponibilizado, conforme visto na figura 4:

Figura 4 – Execução do *script* de build



Fonte: Autoria Própria.

Para executar a aplicação faz-se necessário a utilização de um servidor de aplicação que implemente as principais interfaces de serviços da plataforma JEE que serão consumidos pela aplicação, neste trabalho foi utilizado o projeto *Open Source Jboss Community* na versão 4.0.1, compatível com as tecnologias utilizadas no GSAN, a configuração deste Servidor de Aplicação Web pode ser consultado nos anexos deste trabalho. Para executar

o sistema GSAN, utilizando o terminal de comando do sistema operacional (*Command Prompt*) basta digitar run e pressionar a tecla Enter será iniciado o servidor de aplicação e executará o sistema GSAN, visto na figura 5:

Figura 5 – Executando o sistema GSAN

```

Administrator: C:\windows\system32\cmd.exe - run
14:23:30,233 INFO [SettingsFactory] Order SQL updates by primary key: disabled
14:23:30,233 INFO [SettingsFactory] Query translator: org.hibernate.hql.ast.AST
QueryTranslatorFactory
14:23:30,233 INFO [ASTQueryTranslatorFactory] Using ASTQueryTranslatorFactory
14:23:30,233 INFO [SettingsFactory] Query language substitutions: {yes='Y', no=
'N'}
14:23:30,233 INFO [SettingsFactory] Second-level cache: disabled
14:23:30,233 INFO [SettingsFactory] Query cache: disabled
14:23:30,233 INFO [SettingsFactory] Optimize cache for minimal puts: disabled
14:23:30,233 INFO [SettingsFactory] Structured second-level cache entries: disa
bled
14:23:30,233 INFO [SettingsFactory] Echoing all SQL to stdout
14:23:30,233 INFO [SettingsFactory] Statistics: disabled
14:23:30,233 INFO [SettingsFactory] Deleted entity synthetic identifier rollback
k: disabled
14:23:30,233 INFO [SettingsFactory] Default entity-node: POJO
14:23:30,233 INFO [SessionFactoryImpl] building session factory
14:23:34,296 INFO [SessionFactoryObjectFactory] Not binding factory to JNDI, no
JNDI name configured
14:23:34,296 INFO [NamingHelper] JNDI InitialContext properties:{<}
14:23:34,296 INFO [SessionFactoryImpl] Checking 0 named queries
14:23:34,406 INFO [EARDeployer] Started J2EE application: file:/C:/pessoal/jbos
s/jboss-4.0.1sp1/server/default/deploy/gsan.ear/
14:23:34,437 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0.0-808
0
14:23:34,499 INFO [ChannelSocket] JK2: ajp13 listening on /0.0.0.0:8009
14:23:34,499 INFO [JkMain] Jk running ID=0 time=0/15 config=null
14:23:34,515 INFO [Server] JBoss (MX MicroKernel) [4.0.1sp1 (build: CUSTag=JBos
s_4_0_1_SP1 date=200502160314)] Started in 2m:24s:186ms

```

Fonte: Autoria Própria.

A solução adotada para banco de dados neste trabalho será o PostgreSQL na versão 9.3.4 e PgAdmin versão 1.18.1 como SGBD (Sistema de Gerenciamento de Banco de Dados), no próprio site existe o guia de instalação para desenvolvedores tornando esse passo bem intuitivo.

Na versão do sistema GSAN disponibilizada para a comunidade, existe um diretório chamado *migrations* que contém os *scripts* de banco de dados necessários para criação das tabelas principais que o sistema exige para funcionar corretamente, e também disponibiliza todas instruções necessárias para a configuração dos datasources comercial e gerencial que serão utilizados no sistema GSAN. Para acessar o sistema em execução, basta digitar o seguinte endereço no navegador <http://127.0.0.1:8080/gsan>, caso tudo ocorra bem deverá ser apresentada a página conforme visto na figura 6 abaixo:

Figura 6 – Acessando página inicial do Sistema



Fonte: Autoria Própria.

A credencial de acesso criada por padrão é:

Login: admin

Senha: gcom

Após inserir a credencial de acesso acima teremos acesso a todas as funcionalidades dos módulos do sistema GSAN.

2.1.4 GSAN - DETALHAMENTO TÉCNICO

O GSAN por ser um sistema de informação adaptável a empresas de pequeno, médio e grande porte, contemplando soluções dos mais diversos requisitos, entre eles Cadastramento, Micromedicação, Faturamento, Arrecadação, Cobrança, Negativação e Atendimento ao Cliente. Fornecendo de forma razoavelmente flexível as configurações e detalhes operacionais das rotinas, disponível em ambiente Web utilizando recursos de tecnologias software livre.

Para realização de melhorias no sistema GSAN faz-se necessário o ter conhecimento sobre os principais frameworks utilizados, a linguagem de programação utilizada e alguns

conceitos de Saneamento que serão abordados. Dos principais *frameworks* utilizados, destaco o uso dos seguintes:

Hibernate¹⁰: Trata-se de um robusto framework de persistência de objetos relacionais, que fornece facilmente meios para realizar o mapeamento das entidades do sistema e diminui a complexidade de acesso a base de dados.

Apache Struts¹¹: Tem como característica principal a sua utilização na construção de controladores utilizando o padrão *Model View Control* (MVC) que se trata da separação das camadas utilizadas em uma aplicação, fornecendo uma maior organização no código fonte e contribui para futuras manutenções (FOWLER, 2003).

O sistema GSAN faz uso da plataforma Java, lançado na versão Java Develop Kit (JDK) 1.5, utilizando recursos especificados pela *Java Enterprise Edition* (JEE) (PORTAL DO SOFTWARE PÚBLICO BRASILEIRO, 2007), essencialmente o container *Enterprise Java Bean* (EJB), *Java Server Pages* (JSP) e *Servlets* que são executadas dentro de um servidor de aplicação Java EE.

2.2 ASTERISK

A ferramenta de código aberto Asterisk desenvolvida pela Digium, disponibiliza as principais funcionalidades que um *Call Center* necessita, dentre elas a criação de Ramais, Troncos, Rotas, configuração Plano de Discagens, Gravação de Voz, Conferência, Filas, Unidade de Resposta Audível entre diversos outros recursos que podem ser explorados e utilizados, podendo ser utilizado como um PABX IP, assim como se integrado a soluções VoIP ou a rede de telefonia pública. Com tantos recursos disponíveis a ferramenta Asterisk está muito bem preparada para atender as expectativas, principalmente pelo fato de disponibilizar protocolos de comunicação com sistemas externos, por exemplo, o protocolo AGI (*Asterisk Gateway Interface*), permite o consumo de recursos externos ao Asterisk, já o protocolo AMI (*Asterisk Manager Interface*) permite que aplicações externas enviem ordens para serem executadas no Asterisk, dessa forma a solução pode ser muito bem integrada a sistemas legados.

2.2.1 ASTERISK CONCEITOS

A ferramenta tem como base o Plano de Discagem, sendo ele responsável em definir o que deve acontecer, seja no momento em que for recepcionada uma ligação ou quando for digitado algum número. No plano de discagem podemos definir separações lógicas denominadas contextos, responsáveis em definir um comportamento utilizando os recursos

¹⁰ Hibernate - Framework de mapeamento objeto relacional.

¹¹ Apache Struts - Framework Web para desenvolvimento em camadas.

nativos para determinar as instruções (extensões), por exemplo, podemos criar um contexto para definir o que deve acontecer ao receber ligações da rede pública e outro para definir o comportamento para as ligações advindas de ramais internos:

[PSTN]

```
exten => 2000,1,Answer();
```

Contexto chamado “PSTN”, utiliza a extensão 2000, com prioridade 1 e aplicação *Answer*.

[RAMAIS_INTERNOS]

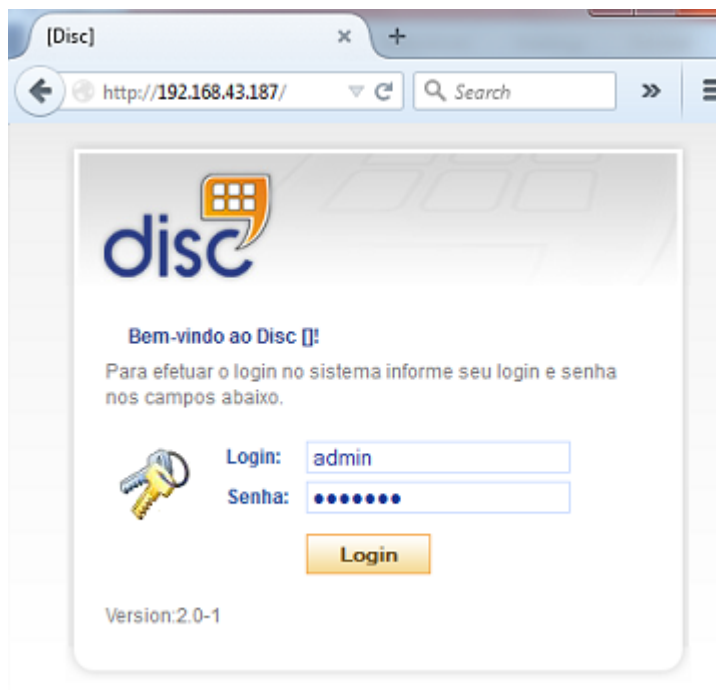
```
exten => 2001,1, Playback(AVISO_GERAL);
```

Contexto chamado “RAMAIS_INTERNOS”, utiliza a extensão 2001, com prioridade 1 e aplicação *Playback* para tocar o áudio *AVISO_GERAL*.

O número 2000 no contexto “PSTN” utilizado na extensão representa o número informado pelo Originador da chamada, a prioridade trata-se de um parâmetro que representa a ordem de execução das aplicações, normalmente descritos de forma sequencial, já as aplicações são utilizadas para realizar uma ação qualquer. Em cada extensão (*exten*), podemos utilizar recursos de aplicativos nativos da ferramenta, sejam eles para atender, desligar, gravar o áudio entre outros recursos, dessa forma podemos definir o comportamento para o contexto, assim como realizar a transferência para outros contextos.

2.2.2 ASTERISK INSTALAÇÃO

A instalação do Asterisk muita das vezes é uma tarefa cansativa e exige bastante atenção, pois a configuração deve ser realizada em arquivos de texto em uma sintaxe estabelecida pela ferramenta, atualmente existem diversas soluções que fornecem uma interface gráfica para tornar esse processo mais intuitivo e prático, neste trabalho foi utilizada uma distribuição chamada Disc-OS na versão 2.0, que disponibiliza uma interface web para realizar a configuração do Asterisk 1.4, para realizar o processo de instalação do Disc-OS, foram seguidos os passos descritos por Jilsimaico Darú (DARÚ, 2008), após a realização dos procedimentos, ao iniciar a distribuição Disc-OS automaticamente é iniciado o serviço do Asterisk, dessa forma para acessar o sistema basta digitar no navegador o endereço IP do terminal que está executando o sistema, visualizando a página conforme a figura 7 abaixo:

Figura 7 – Acessando página inicial da Interface WEB do Disc-OS

Fonte: Autoria Própria.

Para acessar as funcionalidades do sistema basta inserir a seguinte credencial de acesso:

Login: admin

Senha: disc-os

2.2.3 ASTERISK - DETALHAMENTO TÉCNICO

A ferramenta de código aberto Asterisk¹², tem algumas características importantes e fundamentais para o estudo além de ser uma implementação de uma central telefônica que permite que clientes se comuniquem, tem outros recursos interessantes que fazem da ferramenta uma peça chave no processo da integração proposta, recursos como respostas interativas, correios de voz, realização de conferencias, distribuição automática de chamadas, além de ser flexível a adição de novos recursos tanto por meio de scripts na própria linguagem do Asterisk como também por meio de códigos em linguagem C entre outras formas de customização da ferramenta. Desenvolvido pela empresa Digium sob licença GPL¹³, atualmente portátil em versões Linux, Windows e Mac OS, suportando protocolos de Voz sobre IP (VoIP), assim como SIP e H.323 entre outros. O próprio Asterisk contém um protocolo próprio chamado IAX fornecendo um melhor desempenho entre os entroncamentos entre os servidores Asterisk para casos de maior complexidade.

¹² Disponível em <<http://www.asterisk.org>>

¹³ GPL - *General Public Lisencc*

2.3 TRABALHOS RELACIONADOS

Este trabalho de pesquisa e desenvolvimento se assemelha ao trabalho descrito por Guilherme (VIEIRA, 2007), que também utilizou recursos do *software* Asterisk para desenvolver uma Sistema de criação de planos de discagem de forma prática, explanando aspectos da ferramenta e expondo as dificuldades encontradas. Apesar de ambos utilizarem os recursos do Asterisk, há divergência no objetivo onde este se destaca o fato de realizar uma integração com outro software, visando solucionar uma demanda do setor de Saneamento.

No trabalho desenvolvido por Jilcimaico (DARÚ, 2008), aborda com clareza a utilização da distribuição Disc-OS como interface WEB do Asterisk, além de descrever os principais conceitos envolvidos na utilização do software, demonstra os procedimentos necessários para realizar a instalação da ferramenta e configuração dos recursos essenciais para um *Call Center*, assemelhando-se este ao fato de também utilizar a distribuição Disc-OS que propõe uma interface WEB para a configuração do Asterisk.

O trabalho desenvolvimento por Humberto (CAMPOS, 2007), utiliza o *software* Asterisk para realizar uma integração com um sistema externo que calcula os valores de cada ligação realizada com um módulo chamado de “tarifador” e exibe os valores calculados em um hardware próprio. A integração utilizou como referência tabelas em banco de dados para reconhecer eventos ocorridos e disparar ações a serem tomadas, assemelhando-se a este trabalho o fato de utilizar os recursos do Asterisk para disparar ações a sistemas externos, no entanto a forma de integração retratada acima se diferencia da forma adotada neste trabalho, que utiliza a interface AGI disponibilizada para comunicação com sistemas externos, onde o próprio Asterisk irá disparar ações a serem realizadas por meio de um *Middleware* que implementa esta interface de comunicação.

Atualmente a empresa de saneamento Companhia Pernambucana de Saneamento (COMPESA, 2007) disponibilizou aos seus clientes o atendimento eletrônico por meio de URA, possibilitando a empresa realizar o atendimento destinado a central de atendimento, ou seja, o atendimento de primeiro nível, de forma automática e padronizada, propiciando também os direcionamentos entre ramais reais da empresa agilizando o atendimento e potencializando uma disponibilidade de 24 horas por 7 dias, com as informações à disposição dos clientes remotamente, porém a empresa não divulgou detalhes técnicos ou artefatos produzidos para realizar tal integração ou customização. Para auxílio na elaboração deste trabalho de pesquisa se fez de grande valia os detalhes apontados sobre o *software* Asterisk, principalmente a conceituação e protocolos disponibilizados para comunicação com sistemas externos, contidos no próprio *website* da Digium¹⁴.

¹⁴ Digium - Empresa criadora e mantenedora do Asterisk.

2.4 TECNOLOGIAS UTILIZADAS

No estudo para propor uma solução viável e consistente de integração entre sistemas, que seja realmente eficiente, é necessário entender todo o contexto em que está sendo operado o Sistema de Informação GSAN, visando identificar as informações mais relevantes acessadas pelo atendimento ao cliente, as principais dificuldades enfrentadas e os desafios que norteiam esse módulo do sistema.

2.4.1 SIMPLE OBJECT ACCESS PROTOCOL

O protocolo Simple Object Access Protocol (SOAP) tem o objetivo de possibilitar a troca de informações estruturadas em Linguagem de Marcação Extensível (XML), para sistemas distribuídos. A negociação e transmissão de mensagens foram baseadas em outros em outros protocolos de serviços como o HTTP¹⁵ e RPC¹⁶, possibilitando a utilização para realizar integrações entre softwares.

2.4.2 ASTERISK GATEWAY INTERFACE

O software Asterisk possui uma interface de comunicação chamado AGI¹⁷ (ASTERISK, 2014), que tem como objetivo prover uma maior flexibilidade para adaptar soluções de linguagens diferentes, com esta interface se torna possível a comunicação com recursos externos através de requisição semelhantes ao CGI¹⁸ de servidores web, onde as requisições são originadas pelo próprio Asterisk, existem diversos *frameworks* que implementam essa interface de comunicação, para este trabalho será utilizado o framework Asterisk-Java, por ser escrito sobre a plataforma Java e compatível com as tecnologias que foram utilizadas para a comunicação com o Webservice do sistema GSAN. O *framework* Asterisk-Java possui diversos recursos disponíveis para comunicação com a ferramenta Asterisk, a seguir pode ser visto alguns dos principais propostos pelo framework;

- Comunicação AGI - A classe BaseAgiScript.
- Comunicação HTTP - A classe ManagerConnection
- Ouvintes - As interfaces AsteriskServerListener e PropertyChangeListener.
- Controladores - A classes AsteriskServer e DefaultAsteriskServer

¹⁵ HTTP - *Hypertext Transfer Protocol*

¹⁶ RPC - *Remote Procedure Call*

¹⁷ AGI - *Asterisk Gateway Interface*

¹⁸ CGI - *Common Gateway Interface*

2.4.3 WEBSERVICES

O WebService se trata de uma solução que permite que sistemas diferentes se comuniquem através requisições de protocolo HTTP a recursos identificados por um URI¹⁹ identico a Web convencional, descritos e definidos usando XML²⁰.

2.4.4 UNIDADE DE RESPOSTA AUDÍVEL

A Unidade de Resposta Audível (URA) ou atendente eletrônico se trata de um software ou equipamento de *Call Center*, que possibilita o atendimento das ligações de forma automática, tal solução traz como benefício à padronização dos atendimentos e tem potencial para automatização dos atendimentos, com inúmeras possibilidades de customização através de integrações com sistemas externo (VIEIRA, 2007).

2.4.5 MIDDLEWARE

O *Middleware* ou intermediário se trata de uma camada de software responsável em mediar à comunicação de outros sistemas, utilizado normalmente em ambientes que tendem a utilizar plataformas, linguagens ou protocolos de comunicação diferentes nas trocas de informação, sendo um dos recursos adotados neste trabalho, tal recurso descrito por Almeida (2011).

2.4.6 DISC-OS

O Disc-OS refere-se a uma distribuição Linux chamada Cent-OS, customizada para utilização de PABX e PABX IP, abstrai toda a configuração de bibliotecas básicas e instalação do Asterisk (DARÚ, 2008). Disponibiliza uma interface Web para configuração dos principais recursos oque tornar bem prático o processo de configurações. Possui habilitado várias regras de *firewall* pré-configurado por questões de segurança.

2.4.7 CODEC

O codec (COder/DEcoder) se trata de processo de codificação e decodificação da voz humana a ser transmitidas entre a origem e destino em meio digital (VIEIRA, 2007).

2.4.8 JUNIT FRAMEWORK

O JUnit framework destina-se a garantia da qualidade do software, viabilizando a construção dos mais diversos tipos de testes através do seu arcabouço de recursos disponíveis, por ser software livre e adotado como padrão nas principais IDE de desenvolvimento Java, ganhou grande popularidade nas comunidades de desenvolvedores. O desenvolvimento de testes unitários tem sido adotado como métricas de qualidade na entrega do produto de software. O *framework* tornou a escrita de testes um processo fácil e intuitivo, fazendo uso

¹⁹ URI - *Uniform Resource Identifier*

²⁰ XML - *Extensible Markup Language*

de recurso da plataforma Java chamado *Annotation*, trouxe a possibilidade de padronizar os métodos de testes apenas adicionando anotações sobre os mesmo, segue abaixo algumas anotações comumente utilizadas;

- `@Test` - Anotação que representa um método de teste.
- `@Before` - Anotação indica que o método anotado será executado sempre antes de um método de teste.
- `@After` - Anotação indica que o método anotado será executado sempre após a um método de teste.

O JUnit possui um objeto chamado *Assert*, que contém uma série de validações possíveis para checagem do resultado esperado pelo teste.

3 MIDDLEWARE: GSAN E ASTERISK

Neste capítulo são apresentados os procedimentos para a integração dos sistemas envolvidos, expondo o detalhes de implementação e configuração da solução proposta.

3.1 ARQUITETURA DA SOLUÇÃO

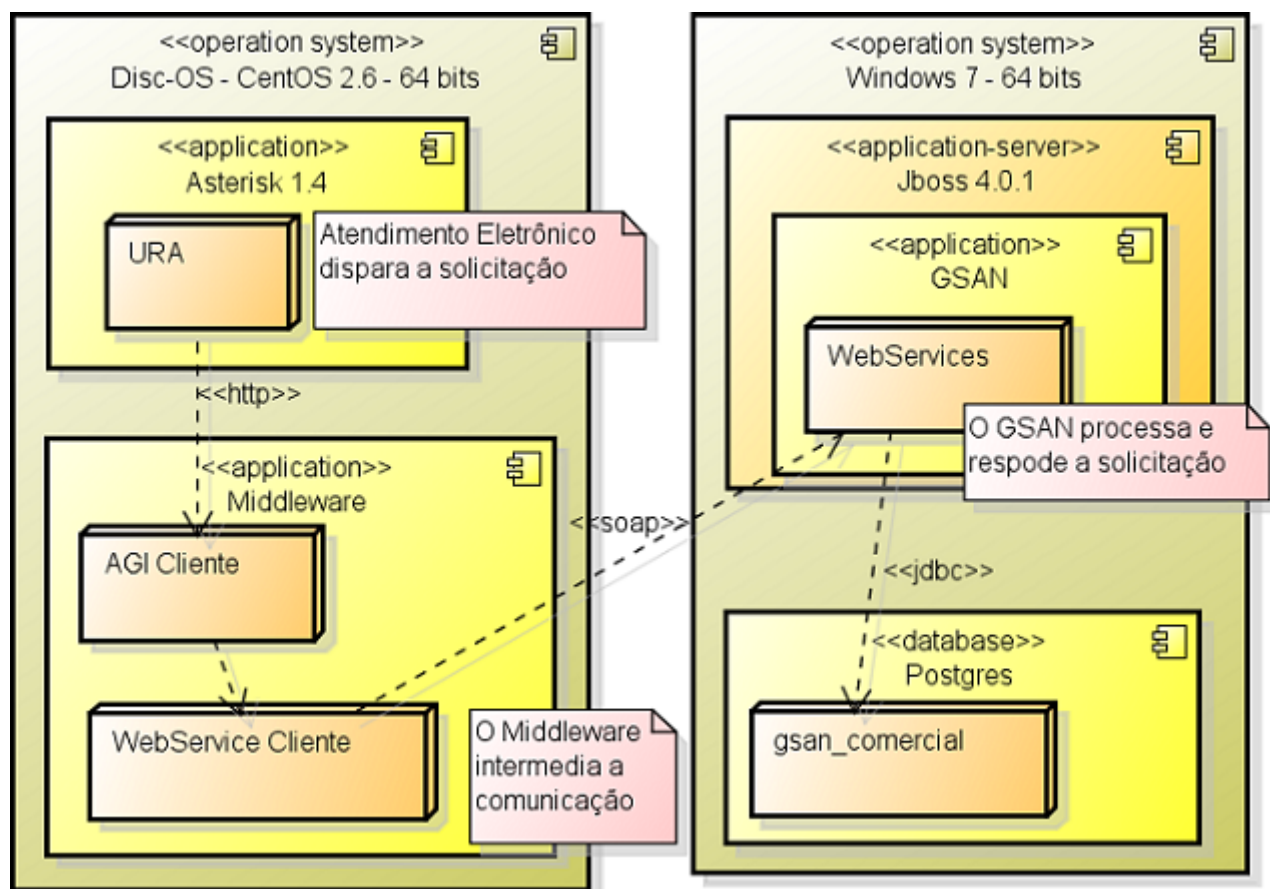
Após o estudo sobre o sistema GSAN e software Asterisk foram identificadas diversas formas de realizar a integração, entre os sistemas pelo fato da existência de vários protocolos possíveis de comunicação, no entanto a solução adotada será visando a reusabilidade, baixo custo de manutenção e o uso de tecnologias que já tenham uma maturidade no mercado. Foram escolhidos os protocolos SOAP¹ e AGI² para serem implementados por um *Middleware*, este é responsável em assumir o papel de intermediário entre os sistemas GSAN e Asterisk. O *Middleware* faz uso do protocolo SOAP para se comunicar com o sistema GSAN, já para a comunicação com o sistema *Asterisk* utiliza a interface de comunicação AGI, dessa forma o *Middleware* abstrai as disparidades entre as plataformas dos sistemas. No sistema Asterisk foi implementado o recurso chamado URA, Unidade de Resposta Audível tornando possível programar um fluxo pré-determinado de atendimento das solicitações recebidas por chamadas telefônicas.

Com o intuito de facilitar o entendimento da comunicação entre os sistemas, a seguir será exposto o diagrama de implantação da solução descrita acima, explanado os principais detalhes adotados nesta integração:

¹ SOAP - *Simple Object Access Protocol*

² AGI - *Asterisk Gateway Interface*

Figura 8 – Diagrama de implantação da solução



Fonte: Autoria Própria.

Conforme ilustrado acima na figura 8, o sistema GSAN prover uma interface de serviços na forma de *Webservices* utilizando o protocolo de comunicação SOAP³, tais serviços são consumidos através do *Middleware* intermediário que também prover uma interface AGI de serviços, para então serem consumidos pelo Asterisk e posteriormente responder as solicitações da Unidade de Resposta Audível.

O processo de integração é composto por tecnologias com paradigmas diferenciados, no entanto para melhorar o entendimento dos detalhes de compatibilidades adotados segue abaixo a tabela 2 das tecnologias utilizadas e versões correspondentes.

³ SOAP - *Simple Object Access Protocol*

Quadro 2 – Tecnologias utilizadas

Software	Finalidade	Versão
Java Platform, <i>Enterprise Edition</i> 5 (JEE)	Conjunto de Tecnologias e Serviços para implementar soluções da Plataforma Java com estabilidade, segurança e escalabilidade.	5
JBoss	Servidor de aplicação que implementa especificações JEE.	4.0.1 sp1
Hibernate	Framework utilizado para fazer o mapeamento objeto-relacional. É responsável pela camada de persistência.	3.1
PostgreSQL	Banco de dados relacional.	9.3.4
Apache Ant	Geração de <i>Enterprise Application Resources</i> (EAR) deploy's.	1.6.2
JasperReports	Tecnologia utilizada para criação de relatórios em PDF, HTML, XLS, CSV e XML.	1.2.2
Struts	Framework para controle de navegação e validação Web.	1.1
Disc-OS	Distribuição CentOS 2.6 que implementa interface web para a ferramenta Asterisk.	2.0-1
Asterisk	Software livre que permite a criação de PABX com diversos recursos.	1.4
Asterisk-Java	Framework para comunicação com o Asterisk via protocolo AGI.	1.0
JUnit	Framework para construção e execução de testes.	4.0

Fonte: Autoria Própria.

3.2 ETAPAS DA INTEGRAÇÃO

O processo de integração entre os sistemas está dividido em quatro etapas principais, das quais são necessárias para compor a solução escolhida, conforme definido abaixo e descrito nas sessões seguintes:

- Implementação de *WebServices* no Sistema GSAN.
- Implementação do *Middleware*.
- Customização do *software* Asterisk.
- Desenvolvimento da suíte de testes automatizados.

3.2.1 IMPLEMENTAÇÃO DE WEBSERVICES NO GSAN

A estratégia de desenvolver o Webservice internamente ao sistema GSAN, foi adotada visando a possibilidade de reaproveitamento das regras de negócio, entidade, controlados dentre os demais artefatos que fazem parte do sistema. Os novos serviços foram desenvolvidos sob tecnologias compatíveis com as utilizadas no sistema GSAN, utilizando a especificação JAX-WS⁴ para serem consumidos por sistemas externos através de submissões de arquivos do tipo XML, definidas no padrão de comunicação SOAP. Para

⁴ JAX-WS - *Java API for XML-Based Web Services*

tratar de conversões de arquivos XML para objetos e vice-versa está sendo utilizada a especificação do padrão JAXB⁵, dessa forma estes serviços são executados no mesmo servidor de aplicação. Na prática é necessário importar ao projeto as seguintes bibliotecas descritas abaixo;

Prover suporte a configuração do End-Points (Serviços);

- jaxws-api-2.2.jar
- jaxws-rt.jar
- jaxws-tools.jar

Fornecer suporte ao tratamento de serialização⁶ dos XML;

- jaxb-impl.jar
- jaxb-xjc.jar
- policy.jar
- stax-ex.jar
- stax2-api.jar
- streambuffer.jar
- woodstox-core-asl.jar

Após realizar a configuração das bibliotecas necessárias para implementação, foram definidos os seguintes novos serviços para automatizar os processos de Obtenção da 2ª via de conta, Informar Falta de Água e Solicitar Restabelecimento da Ligação;

isOnline: Realiza verificação de disponibilidade do sistema.

pesquisarImovelOuCliente: Obtém detalhes sobre um imóvel ou cliente cadastro no sistema GSAN.

obter2ViaConta: Gerar a 2ª via de conta pendente.

informarFaltaAgua: Formalizar junto ao sistema GSAN um Registro de Atendimento do tipo Falta de Água.

solicitarRestabelecimento: Formalizar junto ao sistema GSAN um Registro de Atendimento do tipo Solicitar Restabelecimento da Ligação de Água.

⁵ JAXB - *Java Architecture for XML Binding*

⁶ Serialização refere-se ao processo de conversão de um objeto em bytes.

A figura 9 abaixo, demonstra a declaração dos métodos do *WebService*.

Figura 9 – Interface dos serviços automatizados.

```

15 @WebService(serviceName = "WebServiceRegistroAtendimento")
16 public interface IWSRegistroAtendimento {
17
18     @WebMethod(action = "isOnline")
19     public String isOnline();
20
21     @WebMethod(action = "pesquisarImovelOuUsuario")
22     public Resposta pesquisarImovelOuUsuario(String idImovelCliente);
23
24     @WebMethod(action = "obter2ViaConta")
25     public Resposta obter2ViaConta(String idImovel);
26
27     @WebMethod(action = "informarFaltaAgua")
28     public Resposta informarFaltaAgua(String idImovel, String numeroRA);
29
30     @WebMethod(action = "solicitarRestabelecimento")
31     public Resposta solicitarRestabelecimento(String idImovel, String numeroRA);
32 }

```

Fonte: Autoria Própria.

Para definir a URL⁷ de acesso ao *WebService* será preciso atualizar o arquivo `web.xml`, declarando a servlet padrão definida na especificação do JAX-WS para esse propósito, conforme a figura 10:

Figura 10 – Declaração da servlet do Webservice.

```

1029
1030 <!-- Mapeamento da Servlet do WebService. -->
1031 <servlet>
1032     <servlet-name>WebServiceGSAN</servlet-name>
1033     <servlet-class>com.sun.xml.ws.transport.http.servlet.WSServlet</servlet-class>
1034 </servlet>
1035 <!-- Declarando o padrão a ser interceptado pela Servlet do WebService. -->
1036 <servlet-mapping>
1037     <servlet-name>WebServiceGSAN</servlet-name>
1038     <url-pattern>/webservice/*</url-pattern>
1039 </servlet-mapping>
1040

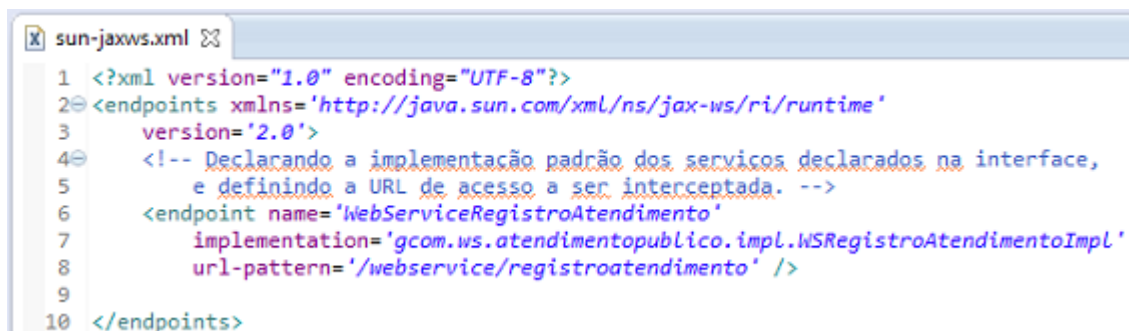
```

Fonte: Autoria Própria.

Feito isso, será preciso disponibilizar os novos serviços, definindo a Classe Concreta⁸ de implementação e o padrão de acesso que será adotado para a interface criada, para que as solicitações sejam interceptadas adequadamente redirecionadas ao *EndPoints* corretos, com isso será necessário criar o arquivo com a seguinte nomenclatura `sun-jaxws.xml`, localizado dentro do diretório `WEB_INF` da aplicação, segue abaixo o conteúdo do arquivo, conforme a figura 11:

⁷ URL - *Uniform Resource Locator*

⁸ Classe Concreta refere-se a classes que possuem atributos, métodos e construtores, passível de instanciação.

Figura 11 – Declaração do *EndPoint* dos serviços.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <endpoints xmlns='http://java.sun.com/xml/ns/jax-ws/ri/runtime'
3   version='2.0'>
4   <!-- Declarando a implementação padrão dos serviços declarados na interface,
5     e definindo a URL de acesso a ser interceptada. -->
6   <endpoint name='WebServiceRegistroAtendimento'
7     implementation='gcom.ws.atendimentopublico.impl.WSRegistroAtendimentoImpl'
8     url-pattern='/webservice/registroatendimento' />
9
10 </endpoints>
```

Fonte: Autoria Própria.

Realizado todos esses passos o sistema GSAN conseguirá disponibilizar os novos serviços declarados na interface de Registro de Atendimento, dessa forma o acesso será realizado da seguinte maneira:

http://<servidor>:<porta>/gsan/webservice/registroatendimento

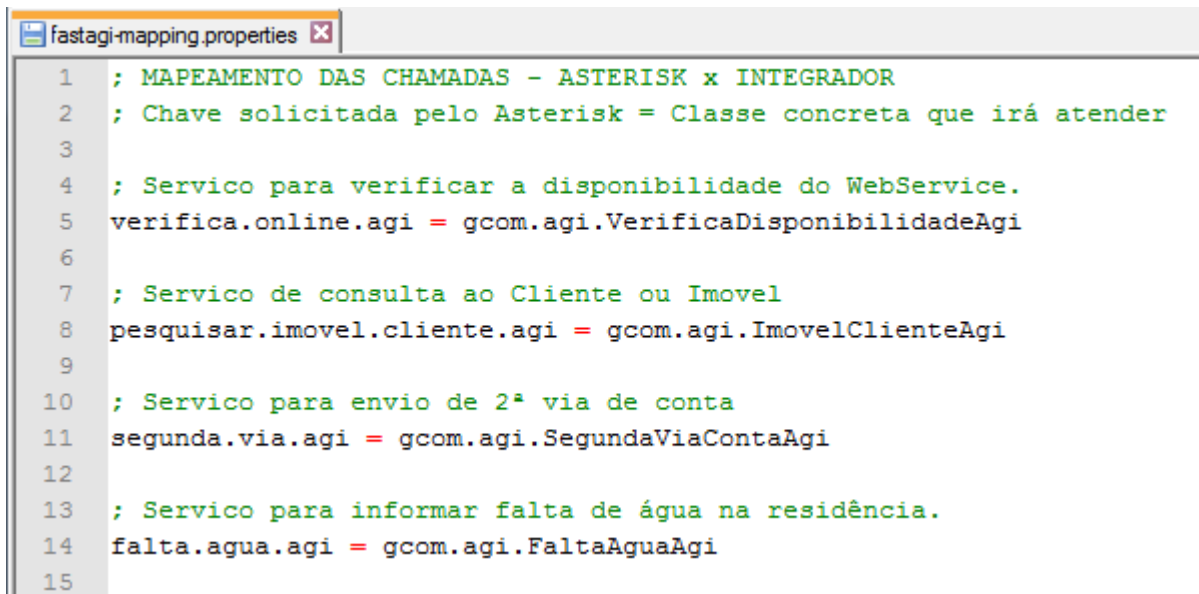
Estando apto a ser integrado com sistemas externos.

3.2.2 IMPLEMENTAÇÃO DO MIDDLEWARE

O *Middleware* intermediário denominado Integrador será responsável em receber as solicitações realizadas pela URA do Asterisk, após analisá-las deverá efetuar requisições ao *WebService* do sistema GSAN, a fim de obter as informações, logo em seguida tratá-las e então devolver as informações essenciais ao Asterisk. O *Middleware* é capaz de comunicar-se com os sistemas GSAN e Asterisk. A comunicação com o sistema GSAN é realizada por uma implementação cliente do *Webservice* disponibilizado, já a comunicação com o Asterisk é realizada por uma interface de comunicação chamada AGI, neste trabalho foi utilizado o *framework Open Source* chamado Asterisk-Java, que fornece recursos para comunicação com o Asterisk utilizando os protocolos comuns ao sistema.

Para cada novo serviço a ser tratado pelo *Middleware*, é necessário declarar uma chave de texto que será utilizada no Asterisk, que representa o objeto a ser invocado, adicionados em um novo arquivo chamado *fastagi-mapping.properties* da seguinte maneira, conforme a figura 12 abaixo:

Figura 12 – Mapeamento dos serviços para consumo via AGI



```

1 ; MAPEAMENTO DAS CHAMADAS - ASTERISK x INTEGRADOR
2 ; Chave solicitada pelo Asterisk = Classe concreta que irá atender
3
4 ; Serviço para verificar a disponibilidade do WebService.
5 verifica.online.agi = gcom.agi.VerificaDisponibilidadeAgi
6
7 ; Serviço de consulta ao Cliente ou Imovel
8 pesquisar.imovel.cliente.agi = gcom.agi.ImovelClienteAgi
9
10 ; Serviço para envio de 2ª via de conta
11 segunda.via.agi = gcom.agi.SegundaViaContaAgi
12
13 ; Serviço para informar falta de água na residência.
14 falta.agua.agi = gcom.agi.FaltaAguaAgi
15

```

Fonte: Autoria Própria.

Dessa forma o *Middleware* tem condições de recepcionar as solicitações originadas pelo *software* Asterisk utilizando a interface de comunicação AGI, e direcionar a chamada internamente para uma classe concreta que contém a lógica de consumo aos novos serviços disponibilizados pelo *WebService* do sistema GSAN.

Para cada classe concreta a ser disponibilizada é necessário estender a classe *BaseAgiScript* disponibilizada pelo *framework* Asterisk-Java, em seguida é preciso implementar o método *service* obtido pela herança da *BaseAgiScript*, tal método será invocado quando a requisição for direcionada ao serviço mapeado no arquivo *fastagi-mapping.properties*, segue abaixo um exemplo de implementação utilizado, conforme a figura 13 abaixo:

Figura 13 – Fluxo de Identificação de Cliente no Middleware.

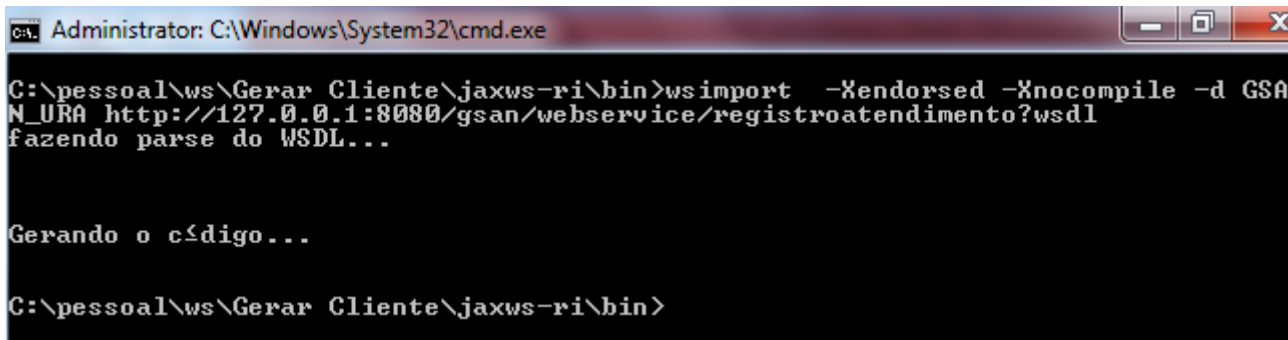
Algoritmo 1 Novo serviço de identificação do cliente (*Middleware*).

1. <i>atenderLigação()</i>	▷ Recepciona a liga
2. <i>digitoInformado</i> ← <i>canal[cliente]</i>	▷ Obtém os dígitos informados
3. if <i>valorInformado</i> <> null then	
4. <i>ws</i> ← <i>obterWebService()</i>	▷ Obtém a instância do WebService
5. <i>retorno</i> ← <i>ws.pesquisarImovelOuCliente(valorInformado)</i>	▷ Realiza a requisição
6. <i>tratarRetorno(retorno)</i>	▷ Verifica o retorno obtido
7. else	
8. <i>tocarAudio(informar_valor)</i>	▷ Tocar o áudio de aviso
9. <i>canal[situacao]</i> ← erro	▷ Sinaliza o erro
10. end if	

Fonte: Autoria Própria.

Visando facilitar o consumo dos novos serviços, será gerado o *WebService* cliente a partir do WSDL⁹ da interface do serviço disponibilizada no sistema GSAN, utilizando o software chamado WSIMPORT disponibilizado pela *Sun Microsystems*, conforme visto na figura 14:

Figura 14 – Geração do código fonte para consumo do Webservice.



```
Administrator: C:\Windows\System32\cmd.exe
C:\pessoal\ws\Gerar Cliente\jaxws-ri\bin>wsimport -Xendorsed -Xnocompile -d GSA
N_URA http://127.0.0.1:8080/gsan/webService/registroatendimento?wsdl
fazendo parse do WSDL...

Gerando o código...

C:\pessoal\ws\Gerar Cliente\jaxws-ri\bin>
```

Fonte: Autoria Própria.

Descrição dos parâmetros utilizados:

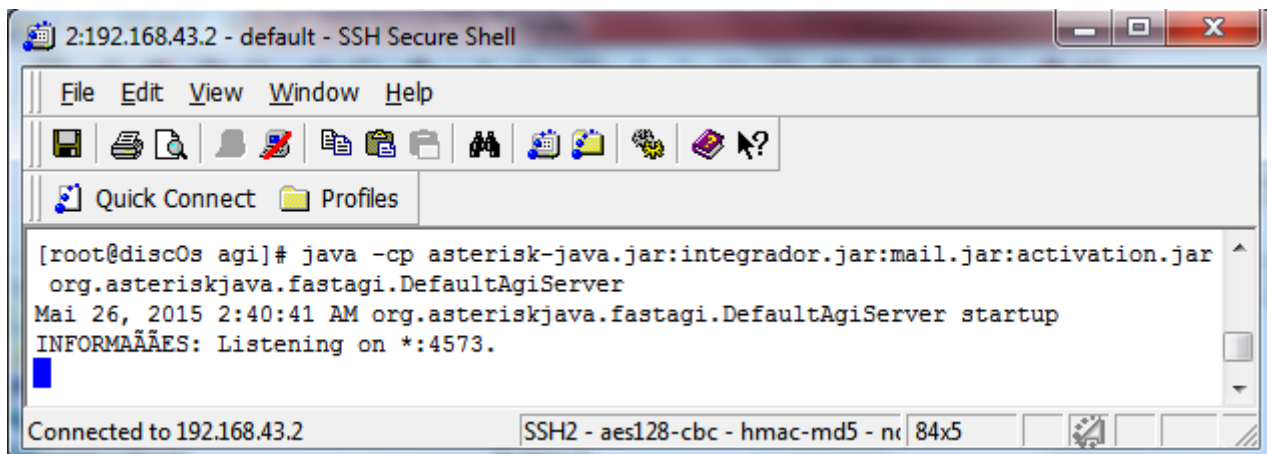
- ***Xendorsed***: Necessário para realizar a substituição dos padrões endossados do sistema, ou seja, utiliza a versão correta da JDK para realizar o *parser*.
- ***Xnocompile***: Utilizado para gerar o código fonte não compilado, ou seja, o arquivo na extensão .java.
- ***-d***: Informado para determinar o diretório para onde os arquivos devem ser gerados, no exemplo o diretório se chama "GSAN_URA".

Por último a URL que disponibiliza o WSDL dos serviços.

Com isso será gerado o código fonte de consumo dos serviços atualmente declarados na interface do *EndPoint*, para cada modificação na interface será necessário realizar o procedimento novamente. O *Middleware* deve ser disponibilizado em um terminal, que seja acessível ao Asterisk, para que o mesmo consiga realizar requisições aos objetos mapeados anteriormente, neste trabalho o *Middleware* foi executado no mesmo hospedeiro do Asterisk, conforme visto na figura 15 abaixo:

⁹ WSDL - *Web Services Description Language*

Figura 15 – Executando o sistema Integrador.



Fonte: Autoria Própria.

A instrução realizada acima executa a classe principal do *framework* Asterisk-Java a *org.asteriskjava.fastagi.DefaultAgiServer*, inicializando o serviço de comunicação via AGI, sob a JVM¹⁰ da plataforma Java, passando como *Classpath*¹¹ as seguintes bibliotecas:

- ***asterisk-java.jar***: biblioteca do *framework* utilizado para se comunicar com o Asterisk.
- ***integrador.jar***: o *Middleware* que integra as aplicações.
- ***mail.jar* e *activation.jar***: ambos necessários no envio de e-mail, para os casos de Obter 2ª via de conta.

Após estes passos o sistema Integrador já estará apto a receber solicitações do Asterisk e realizar requisições para o Sistema GSAN.

3.2.3 CUSTOMIZAÇÃO DO ASTERISK

A customização do Asterisk será fundamental para realizar a comunicação entre os sistemas, sendo responsável em disponibilizar de forma padronizada o acesso a URA no atendimento de primeiro nível de todas as chamadas recebidas, e disparar solicitações a recursos externos conforme a necessidade do cliente que efetuou a ligação. Primeiramente foi configurado um ramal, utilizando a interface web do Disc-OS, conforme exemplo a seguir na figura 16:

¹⁰ JVM - Java Virtual Machine

¹¹ Classpath refere-se a uma variável de ambiente que indica o local onde as dependências estão localizadas.

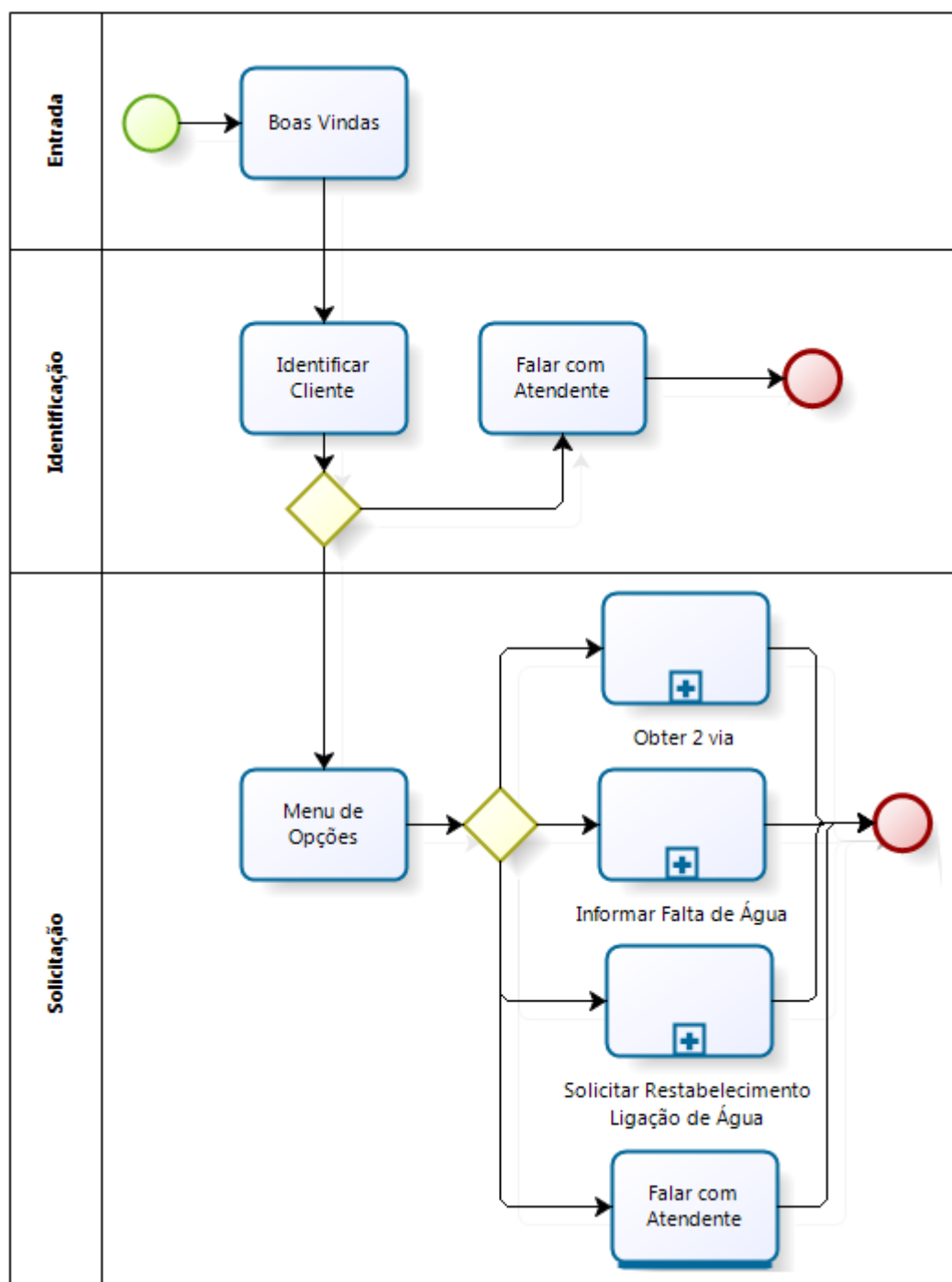
Figura 16 – Cadastro de um Ramal SIP.

The screenshot displays the Disc OS web interface for configuring SIP extensions. The browser window shows the URL `192.168.43.2/disc?module=extens`. The interface includes a sidebar menu on the left with categories like 'Início', 'Configuração', 'Ramais', 'Troncos', 'Rotas', 'Grupos de Busca', 'Grupos de Horários', 'Conferência', 'Filas', 'Serviços', 'Áudio e Música', 'Atendimento', 'Automático', 'Estacionamento', 'DISA', 'Configurações', 'Bloqueios', 'Correio de Voz', and 'Gerenciamento'. The main content area is titled 'Ramais' and features a 'Salvar como Padrão' button. Under 'Configurações do ramal', there are three input fields: 'Nome' (FULANO), 'Número' (2001), and 'PIN' (1234). Below this, the 'Configurações de SIP' section includes a 'Senha SIP' field (masked with dots) and a list of allowed codecs: g726-32, g729, iLBC, Speex, H.261, and H.263. There are also buttons for moving codecs between lists.

Fonte: Autoria Própria.

Conforme visto na figura 16, está sendo configurado o ramal de número 2001, com o nome de FULANO e definido alguns *codecs* que serão permitidos utilizar neste ramal, após isto o Disc-OS se encarregará de alterar o arquivo de propriedades *sip.conf* inserindo o ramal desejado com as características informadas. Para a configuração da Unidade de Resposta Audível, será preciso definir claramente quais serão as opções disponíveis e quais serão as possíveis saídas com os seus devidos tratamentos, neste trabalho foi desenvolvido um fluxo de atendimento próprio, demonstrado na figura 17:

Figura 17 – Diagrama do Fluxo da Unidade de Resposta Audível



Fonte: Autoria Própria.

Dessa forma o cliente tem a opção de falar com o atendente logo no primeiro menu de atendimento e nos demais sub fluxos, comportamento exigido conforme o artigo 4 da Lei nº 8.078 (BRASIL, 2008), que fixa normas gerais sobre o Serviço de Atendimento ao Consumidor, após realizar o passo de identificação o mesmo terá acesso as opções de serviços automatizados via integração caso ocorra tudo com êxito, caso não seja possível identificar o cliente será redirecionado para o Falar com Atendente. A interface *Web* do Disc-OS permite realizar a configuração da URA, de forma bem intuitiva, inclusive definir

tempos de espera, direcionamento para outros Ramais entre outras configurações que tornam o processo de configuração muito mais prático, no entanto para configurar o fluxo da URA realizando a comunicação via AGI com o *Middleware* será preciso realizar o procedimento manual de configuração, segue a configuração utilizada para criação do contexto de PESQUISAR_CLIENTE utilizada para identificar o cliente, descrito no arquivo */etc/asterisk/extensions.conf*, conforme representado pela figura 18 a seguir:

Figura 18 – Fluxo de Identificação de Cliente no Asterisk.

Algoritmo 2 Fluxo de identificação do cliente (Asterisk).

```

1. atenderLigação()                                ▷ recepciona a ligação
2. tocarAudio(beep)                                  ▷ tocar áudio de aviso
3. tempo_limite_discagem ← 3                          ▷ defini tempo limite de espera entre dígitos
4. tempo_limite_resposta ← 7                          ▷ defini tempo limite de espera do primeiro dígito
5. digitos ← lerDigitos()                            ▷ ler os dígitos informados
6. canal[cliente] ← digitos                          ▷ adiciona os dígitos no canal
7. Agi(pesquisar.imovel.cliente.agi)                  ▷ faz a chamada agi
8. if canal[situacao] == 'sucesso' then
9.   gotoSucesso()                                  ▷ redireciona a ligação para sucesso
10. else
11.   gotoContextoAtendente()                       ▷ redireciona a ligação para falar com atendente
12. end if

```

Fonte: Autoria Própria.

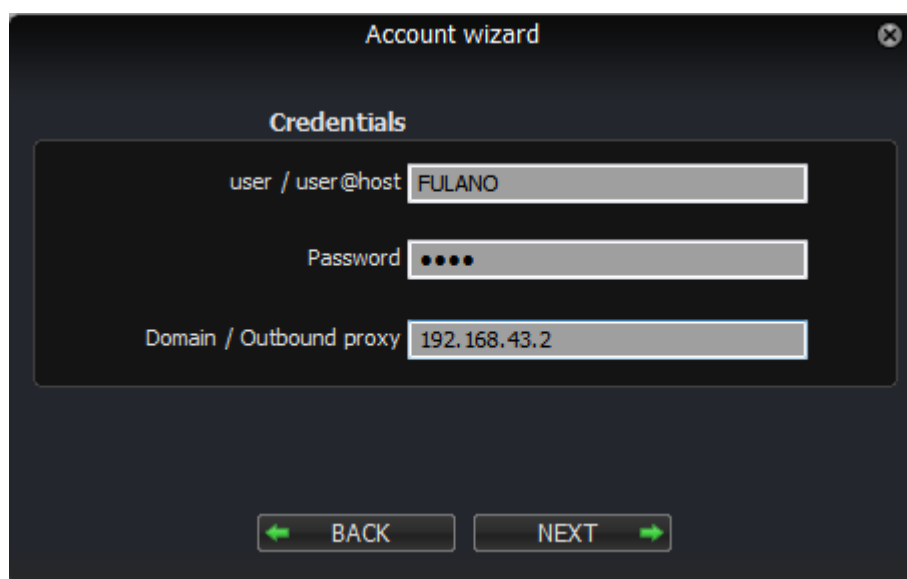
A figura 18 acima representa a rotina que será executada toda vez que um cliente ao entrar em contato com a Central de Atendimento e optar em se identificar, ao selecionar essa opção a URA por meio do contexto [PESQUISAR_CLIENTE] irá atender a ligação e logo em seguida emitir um áudio chamado "*beep*", após este som o cliente terá 7 segundos para informar o primeiro dígito e 3 segundos de esperar entre os dígitos, quando o cliente terminar de informar os dígitos de identificação o algoritmo irá atribuir os dígitos à variável chamada CLIENTE_IMOVEL e realizar o processo de requisição via interface de comunicação AGI para o *Middleware*, solicitando o serviço mapeando em "*pesquisar.imovel.cliente.agi*", caso o cliente seja identificado com sucesso a ligação será direcionada para o contexto de opções dos serviços automatizados, caso contrário a ligação será redirecionada para o atendente.

O fluxo de atendimento da URA, permite que seja configurado um arquivo de Áudio para interação com o cliente, neste trabalho foram criados os arquivos de áudio para serem tocados, conforme as operações disponíveis forem acessadas. A própria ferramenta Asterisk fornece uma forma de gravar um arquivo de áudio, discando para o número *77 é acionado um aplicativo nativo que começa a gravar o que for falado na chamada após o término da chamada é salvo o arquivo de áudio no diretório */var/lib/asterisk/sounds/custom/* para utilização.

Após as configurações básicas, é preciso ter um dispositivo para efetuar as chamadas e conseguir se comunicar com a URA, neste cenário existe um recurso chamado de *Softphone*, são programas que tornam o computador em um Ramal IP, possibilitando realizar e receber chamadas, com essa finalidade foi utilizado o Zoiper¹².

Para adicionar o ramal é preciso definir qual será o protocolo utilizado e inserir os dados configurados no Asterisk conforme visto na figura 19 abaixo:

Figura 19 – Configurar Ramal no Zoiper



Fonte: Autoria Própria.

Após realizar todos estes passos é possível acessar os serviços disponibilizados pela URA, através do Zoiper efetuar as chamadas para percorrer os fluxos previamente definidos.

3.2.4 DESENVOLVIMENTO DA SUÍTE DE TESTES AUTOMATIZADOS

A suíte de testes automatizados possui a responsabilidade de garantir o correto funcionamento da integração entre os sistemas envolvidos, ou seja prover um meio onde possa ser testado o Webservice do GSAN e o Middleware juntamente com o Asterisk. Neste cenário foram utilizados recursos do JUnit e Asterisk-Java. O *framework* JUnit 2.4.8 foi utilizado para criação dos cenários de teste, execução dos testes e identificação de falhas. Para estabelecer uma conexão com o Asterisk, criar e monitorar chamadas telefônicas com parametrização dinâmica foram utilizados recursos do *framework* Asterisk-Java ??.

A execução dos testes através do *framework* JUnit podem ser realizada a qualquer momento, seja manualmente pela IDE de desenvolvimento ou automatizada por algum ambiente de Integração Contínua¹³. Dessa forma é possível realizar a inspeção da qualidade

¹² Zoiper - Softphone que fornece recursos para simular ramais

¹³ Integração Contínua adotada na Engenharia de Software refere-se a uma prática de unir todo o trabalho desenvolvido várias vezes ao dia em uma estrutura principal.

do software, visando garantir o correto comportamento dos sistemas na construção de uma entrega ou *release*.

A suíte de teste é composta pelos seguintes classes;

- Constantes - Contém a parametrização necessária para configuração da conexão com a ferramenta Asterisk.
- Serviço - Permite efetuar o *Login*, *Logoff* e iniciar uma chamada telefônica.
- Testes - Implementa a lógica do cenário de teste.
- Utilitária - Realiza o registro de ouvintes em chamadas telefônicas.

Com o intuito de unir os recursos do *framework* JUnit com os do Asterisk-Java, cada classe de teste deve seguir um padrão no momento da implementação, conforme descrito a seguir:

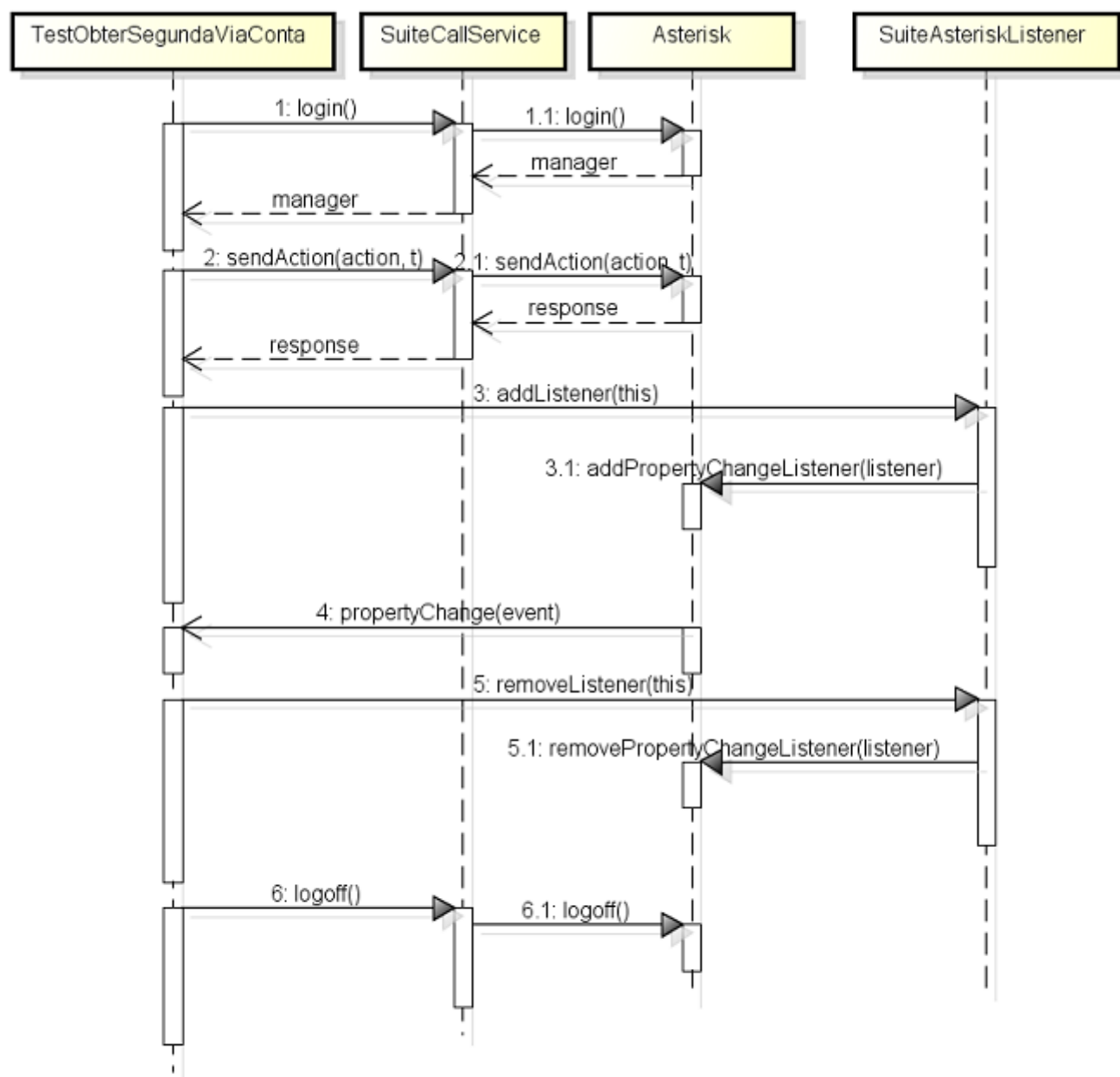
1. Configurar o método de Teste anotado com *@Test*.
2. Configurar o método de *Login* anotado com *@Before*.
3. Configurar o método de *Logoff* anotado com *@After*.
4. Implementar uma interface chamada *PropertyChangeListener*.

Por meio desta padronização na classe de teste, permite que a mesma seja executada pelo *framework* JUnit e se comunique com a ferramenta Asterisk por meio do *framework* Asterisk-Java.

O diagrama sequência demonstrado na figura 20 ilustra o funcionamento da classe de teste do serviço Obter 2ª via de conta, trabalhando em conjunto com a suíte de teste.

Esse modelo é atributo a todos os testes de integração desenvolvidos na suíte de testes, onde inicialmente é realizado o processo de Login junto a ferramenta Asterisk, em seguida é criada a chamada utilizando o método *sendAction*, e registrado o ouvinte através do método *addListener*, quando o Asterisk identifica mudança nas propriedades da chamada do ouvinte registrado, dispara o método *propertyChange* contendo os detalhes do evento, com esses detalhes o teste por determinar o ocorrência e sucesso ou error e posteriormente remove o registro de ouvinte e realiza o *Logoff*, chamando respectivamente os métodos *removeListener* e *Logoff*.

Figura 20 – Diagrama de sequência utilizando a suíte de teste.



Fonte: Autoria Própria.

Durante a execução de um cenário de teste, é necessário monitorar o comportamento da chamada ou canal internamente na ferramenta Asterisk, a fim de validar o retorno obtido pela requisição e posteriormente evidenciar a ocorrência do comportamento esperado ou de falhas. Com isso se faz necessário habilitar a conexão remota na ferramenta Asterisk. O arquivo `/etc/asterisk/manager.conf` contém as propriedades necessárias para serem configuradas, a seguir é demonstrada a configuração realizada para execução dos

testes automatizados:

[general] ▷ Contexto geral de conexão.
enabled=yes ▷ Habilitar a conexão remota.
port=5038 ▷ Define a porta.
displayconnects=yes ▷ Define a exibição das conexões no console.
permit=0.0.0.0/0.0.0.0 ▷ Define o endereço IP que será aceito.

Abaixo segue o exemplo de configuração para adição de um novo usuário para conexão remota:

[manager] ▷ Nome do novo usuário.
secret=pa55w0rd ▷ Senha do novo usuário
read=system,call,log,verbose,command,agent ▷ Define as permissões de leitura.
write=system,call,log,verbose,command,agent ▷ Define as permissões de escrita.
permit=0.0.0.0/0.0.0.0 ▷ Define o endereço IP que será aceito

A distribuição Disc-OS por padrão possui uma configuração de *firewall* bastante restritiva por questões de segurança, para que não ocorra rejeição nas solicitações realizadas ao Asterisk, será para fins de testes desabilitada as regras do *firewall*, utilizando o seguinte comando */etc/init.d/iptables stop*.

A suíte de testes contém a implementação dos testes de integração para os três serviços automatizados que são obter 2º via de conta, informar falta de água e solicitar restabelecimento da ligação de água, além de testes unitários para verificação da rotina de enviar e-mail e checagem da disponibilidade do *Webservice*. A vantagem principal obtida por meio da suíte de testes se trata da possibilidade de tornar o processo de identificação de falhas padronizado, já sua desvantagem se trata da complexidade na etapa de configuração.

4 DEMONSTRAÇÃO DA SUÍTE DE TESTES AUTOMATIZADOS

Será apresentado neste capítulo a demonstração da suíte de testes automatizados sob a solução abordada nas seções anteriores, e exposto os resultados que foram alcançados.

4.1 PREPARAÇÃO DO AMBIENTE DE TESTE

A demonstração da suíte de testes automatizados sobre a solução de integração dos sistemas, foram planejados em um ambiente controlado na elaboração e análise dos cenários de testes envolvidos.

Para que os cenários de testes sejam o mais semelhante ao ambiente de produção, os mesmos foram executados sobre uma chamada telefônica tendo como originador da ligação um cliente com suas características especificadas e descritas em cada cenário de teste.

O ambiente para execução dos testes ocorreu sobre um terminal cujo as especificações estão descritas abaixo:

Quadro 3 – Recursos computacionais utilizados

SISTEMA OPER.	PROCESSADOR	MEMÓRIA	ARMAZENAMENTO
Windows 7 64 bits	Intel Core i7-3520M CPU@2.9GHz	8GB DDR3	1TB 5400 rpm

Fonte: Autoria Própria.

Este terminal é responsável em executar ambos os sistemas envolvidos, no entanto como o software Asterisk está disponível somente para plataforma Linux, é preciso utilizar o recurso de máquina virtual através do software Virtual Box¹, para executar uma instância da distribuição Disc-Os conforme citado nas seções anteriores.

O terminal possui uma conexão de rede local ativa, para que seja atribuído um endereço IP ao sistema operacional hospedeiro e ao convidado.

Visando garantir o correto funcionamento e a padronização do ambiente de teste foram criados testes de integração reproduzindo uma chamada telefônica programaticamente simulando um ambiente real, para tanto foi necessário utilizar os seguintes recursos o *framework* JUnit na construção dos testes e o recurso nativo do Asterisk chamado *Local Channel*.

Dessa forma é possível criar um canal de comunicação com um contexto e extensão específica dentro do sistema Asterisk, ou seja permite acessar diretamente o ponto de integração entre os sistemas.

¹ Virtual Box - Software que provê formas de virtualização de terminais.

Foram criados contextos específicos para teste no software Asterisk, replicando os mesmos contextos utilizados no fluxo da URA visando isolar o comportamento que ocorre no momento da integração sem a necessidade de redirecionamentos entre serviços ou ramais, conforme visto abaixo:

Figura 21 – Declaração dos contextos de teste no Asterisk

```

499 [TESTE_PESQUISAR_CLIENTE]
500 exten => s,1,Answer()
501 exten => s,1,NoOp(## - Iniciando testes do servico: PESQUISAR CLIENTE - ##)
502 exten => s,n,NoOp(## - Parametros: Imovel:${CLIENTE_IMOVEL} - ##)
503 exten => s,n,Agi(aqi://192.168.43.187/pesquisar.imovel.cliente.aqi)
504 exten => s,n,NoOp(## - Resultado: ID-${ID_IMOVEL} || ST-${SITUACAO} || RA-${NUMERO_RA}
505
506 [TESTE_OBTER_SEGUNDA_VIA]
507 exten => s,1,Answer()
508 exten => s,1,NoOp(## - Iniciando testes do servico: OBTER SEGUNDA VIA - ##)
509 exten => s,n,NoOp(## - Parametros: Imovel:${ID_IMOVEL} - ##)
510 exten => s,n,Agi(aqi://192.168.43.187/segunda.via.aqi)
511 exten => s,n,NoOp(## - Resultado: ${ID_IMOVEL} || ${SITUACAO} - ###)
512
513 [TESTE_INFORMAR_FALTA_AGUA]
514 exten => s,1,Answer()
515 exten => s,1,NoOp(## - Iniciando testes do servico: INFORMAR FALTA AGUA - ##)
516 exten => s,n,NoOp(## - Parametros: Imovel:${ID_IMOVEL} - ##)
517 exten => s,n,Agi(aqi://192.168.43.187/falta.agua.aqi)
518 exten => s,n,NoOp(## - Resultado: ${ID_IMOVEL} || ${SITUACAO} - ###)
519
520 [TESTE_SOLICITAR_RESTABELECIMENTO]
521 exten => s,1,Answer()
522 exten => s,1,NoOp(## - Iniciando testes do servico: INFORMAR FALTA AGUA - ##)
523 exten => s,n,NoOp(## - Parametros: Imovel:${ID_IMOVEL} - ##)
524 exten => s,n,Agi(aqi://192.168.43.187/restabelecimento.ligacao.aqi)
525 exten => s,n,NoOp(## - Resultado: ${ID_IMOVEL} || ${SITUACAO} - ###)

```

Fonte: Autoria Própria.

4.2 CENÁRIOS DE TESTE

Nesta fase foram elaborados possíveis cenários de teste para os três serviços automatizados, são respectivamente Obter 2ª via de conta, Informar Falta de Água e Solicitar Restabelecimento da Ligação de Água. Cada serviço proposto possui um objetivo específico sempre vinculado a um imóvel e/ou cliente. Os cenários de testes representam as possíveis situações cadastrais fictícias e comportamentais vinculadas a um cliente, essencial para realizar um atendimento via sistema. Foram propostos 3 cenários de teste para cada serviço, visando assegurar o correto comportamento da solução. Abaixo estão descritos os cenários conforme cada serviço proposto:

4.2.1 OBTER 2ª VIA DE CONTA

Este serviço tem como objetivo checar se há alguma conta pendente referente ao imóvel informado, caso haja pendência o serviço deve submeter a conta pendente via e-mail para o cliente do imóvel.

Abaixo são descritos os casos de testes previstos para o serviço obter 2ª Via de Conta, foi cadastrado o mesmo endereço de e-mail para os 3 cenários.

CENÁRIO 1: Cliente devidamente cadastrado no Sistema, atualmente usuário de um imóvel que possui ativo a Ligação de Água e Esgoto, com uma conta em atraso. O Cliente deseja obter a conta em aberto, conforme demonstrado na figura 22.

Figura 22 – Obter 2ª via - Cenário de Teste 1

Dados do Imóvel

Imóvel:*

1368124

001.006.020.0010.002

Sit.Água: LIGADO

Sit.Esgoto: LIGADO

Sit.Lixo: DESLIGADO

Clientes

Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo
EDUALANO DE SOUZA	USUARIO/LOCATARIO	01/01/1900	(48)35334712		SIM

Contas

Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
07/2015	10/08/2015	9,49	6,64	0,00	0,00	0,00	0,00	16,13	0,81	NOR
Total	1 doc(s)	9,49	6,64	0,00	0,00	0,00	0,00	16,13	0,81	

Fonte: Autoria Própria.

CENÁRIO 2: Cliente devidamente cadastrado no Sistema, atualmente sendo usuário de um imóvel que possui a Ligação de Água cortada por falta de pagamento, atualmente com três contas em aberto. O Cliente deseja obter a conta de maior atraso, conforme demonstrado na figura 23.

Figura 23 – Obter 2ª via - Cenário de Teste 2

Dados do Imóvel										
Imóvel:*	B0864	001.002.010.0009.001								
Sit.Água:	CORTADO	Sit.Esgoto:	POTENCIAL	Sit.Lixo:	DESLIGADO					
Clientes										
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo					
LUIS HOMERO RAMOS	USUARIO/LOCATARIO	01/01/1900			SIM					
Contas										
Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
07/2013	10/08/2013	22,33	0,00	0,00	0,00	0,00	0,00	22,33	6,56	DIV
08/2013	10/09/2013	22,33	0,00	0,00	0,00	0,00	0,00	22,33	6,33	DIV
09/2013	10/10/2013	22,33	0,00	0,00	0,00	0,00	0,00	22,33	6,11	DIV
Total	3 doc(s)	66,99	0,00	0,00	0,00	0,00	0,00	66,99	19,00	

Fonte: Autoria Própria.

CENÁRIO 3: Cliente sem e-mail cadastrado no Sistema, atualmente sendo proprietário por dois imóveis que possuem ativo a Ligação de Água, possuindo várias contas pendentes para cada imóvel. O Cliente pretende obter todas as contas de todos os imóveis, conforme demonstrado na figura 24.

Figura 24 – Obter 2ª via - Cenário de Teste 3

Dados do Imóvel										
Imóvel:*	I3704001	001.001.028.0010.010								
Sit.Água:	LIGADO	Sit.Esgoto:	FACTIVEL	Sit.Lixo:	DESLIGADO					
Clientes										
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo					
ALCIBIADES JOSE SCALCO	PROPRIETARIO	03/06/2015	(48)96325450		SIM					
ALCIBIADES JOSE SCALCO	USUARIO/LOCATARIO	03/06/2015	(48)96325450		SIM					
Contas										
Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
06/2015	10/07/2015	36,51	0,00	0,00	25,60	0,00	0,00	62,11	3,78	NOR
07/2015	10/08/2015	36,51	0,00	0,00	25,60	0,00	0,00	62,11	3,14	NOR
08/2015	10/09/2015	36,51	0,00	0,00	25,60	0,00	0,00	62,11	2,50	NOR
Total	3 doc(s)	109,53	0,00	0,00	76,80	0,00	0,00	186,33	9,42	

Fonte: Autoria Própria.

4.2.2 INFORMAR FALTA DE ÁGUA

Este serviço tem como objetivo formalizar junto ao sistema GSAN um novo Registro de Atendimento referente a Falta de Água para o imóvel informado. Abaixo são descritos

os casos de testes previstos para o serviço Informar Falta de Água.

CENÁRIO 1: Cliente devidamente cadastrado no Sistema, atualmente locatário de um imóvel que possui ativo a Ligação de Água e Esgoto, com pendência de duas contas, com problemas no abastecimento de água. O Cliente pretende informar a falta de água para o imóvel, conforme demonstrado na figura 25.

Figura 25 – Informar Falta de Água - Cenário de Teste 1

Dados do Imóvel										
Imóvel:*	1276833	001.005.020.0017.001								
Sit.Água:	LIGADO	Sit.Esgoto:	POTENCIAL	Sit.Lixo:	DESLIGADO					
Clientes										
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo					
<u>SANTELMO AMARAL FERREIRA</u>	USUARIO/LOCATARIO	01/01/1900			SIM					
Contas										
Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
07/2015	10/08/2015	25,01	0,00	0,00	0,00	0,00	0,00	25,01	1,27	NOR
08/2015	10/09/2015	33,01	0,00	0,00	1,54	0,00	0,00	34,55	1,37	NOR
Total	2 doc(s)	58,02	0,00	0,00	1,54	0,00	0,00	59,56	2,64	

Fonte: Autoria Própria.

CENÁRIO 2: Cliente sem e-mail cadastrado no Sistema, atualmente usuário de um único imóvel que possui ativo a Ligação de Água, com situação de inadimplência, com problemas no abastecimento de água. O Cliente pretende informar a falta de água para o único imóvel, conforme demonstrado na figura 25.

Figura 26 – Informar Falta de Água - Cenário de Teste 2

Dados do Imóvel										
Imóvel:*	13702840	001.004.001.0001.003								
Sit.Água:	LIGADO	Sit.Esgoto:	FACTIVEL	Sit.Lixo:	DESLIGADO					
Clientes										
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo					
<u>LINENQUER ELIAS LENTZ</u>	PROPRIETARIO	18/12/2014	(48)96945443		SIM					
<u>LINENQUER ELIAS LENTZ</u>	USUARIO/LOCATARIO	18/12/2014	(48)96945443		SIM					
Contas										

Fonte: Autoria Própria.

CENÁRIO 3: Cliente devidamente cadastrado no Sistema, atualmente proprietário de dois imóveis, que possuem a Ligação de Água ativa, em situação de inadimplência. O Cliente deseja informar problemas no abastecimento de água do imóvel específico onde reside, conforme demonstrado na figura 27.

Figura 27 – Informar Falta de Água - Cenário de Teste 3

Dados do Imóvel

Imóvel:

Sit.Água: **LIGADO** Sit.Esgoto: **FACTIVEL** Sit.Lixo: **DESLIGADO**

Clientes					
Nome do Cliente	Tipo da Relação	Data Inicio Relação	Telefone	CPF/CNPJ	Ativo
<u>ALCIBIADES JOSE SCALCO</u>	PROPRIETARIO	03/06/2015	(48)96325450		SIM
<u>ALCIBIADES JOSE SCALCO</u>	USUARIO/LOCATARIO	03/06/2015	(48)96325450		SIM

Contas										
Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
<u>06/2015</u>	10/07/2015	36,51	0,00	0,00	<u>25,60</u>	0,00	0,00	62,11	<u>3,78</u>	NOR
<u>07/2015</u>	10/08/2015	36,51	0,00	0,00	<u>25,60</u>	0,00	0,00	62,11	<u>3,14</u>	NOR
<u>08/2015</u>	10/09/2015	36,51	0,00	0,00	<u>25,60</u>	0,00	0,00	62,11	<u>2,50</u>	NOR
Total	3 doc(s)	109,53	0,00	0,00	76,80	0,00	0,00	186,33	9,42	

Fonte: Autoria Própria.

4.2.3 SOLICITAR RESTABELECIMENTO DA LIGAÇÃO DE ÁGUA

Este serviço tem como objetivo formalizar junto ao sistema GSAN um novo Registro de Atendimento referente ao Restabelecimento da Ligação de Água para um imóvel informado. Abaixo são descritos os casos de testes previstos para o serviço Solicitar Restabelecimento da Ligação de Água.

CENÁRIO 1: Cliente devidamente cadastrado no Sistema, atualmente proprietário de um único imóvel que possui a Ligação de Água interrompida por corte, em situação recente de inadimplência. O Cliente deseja solicitar restabelecimento da ligação para o imóvel, conforme demonstrado na figura 28.

Figura 28 – Restabelecimento da Ligação de Água - Cenário de Teste 1

Dados do Imóvel					
Imóvel:*	1239723	001.002.034.0002.001			
Sit.Água:	CORTADO	Sit.Esgoto:	POTENCIAL	Sit.Lixo:	DESLIGADO
Clientes					
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo
SIMONE CARVALHO DA SILVA ZEFERINO	PROPRIETARIO	08/05/2014			SIM
SIMONE CARVALHO DA SILVA ZEFERINO	USUARIO/LOCATARIO	08/05/2014			SIM
Contas					

Fonte: Autoria Própria.

CENÁRIO 2: Cliente devidamente cadastrado no Sistema, atualmente usuário de um único imóvel que possui a Ligação de Água interrompida por corte, em situação recente de inadimplência. O Cliente deseja solicitar restabelecimento da ligação para o imóvel, conforme demonstrado na figura 29.

Figura 29 – Restabelecimento da Ligação de Água - Cenário de Teste 2

Dados do Imóvel					
Imóvel:*	80166	001.001.008.0022.001			
Sit.Água:	CORTADO	Sit.Esgoto:	POTENCIAL	Sit.Lixo:	DESLIGADO
Clientes					
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo
MARIA ISABEL CHAVES	USUARIO/LOCATARIO	01/01/1900			SIM
Contas					

Fonte: Autoria Própria.

CENÁRIO 3: Cliente devidamente cadastrado no Sistema, atualmente proprietário de um imóvel, que possui Ligação de Água interrompida por corte, devido a falta de pagamento das várias contas em aberto. O Cliente deseja solicitar restabelecimento da ligação para o imóvel sem realizar a negociação das faturas, conforme demonstrado na figura 30.

Figura 30 – Restabelecimento da Ligação de Água - Cenário de Teste 3

Dados do Imóvel

Imóvel:*

Sit.Água: CORTADO
Sit.Esgoto: FACTIVEL
Sit.Lixo: DESLIGADO

Clientes					
Nome do Cliente	Tipo da Relação	Data Início Relação	Telefone	CPF/CNPJ	Ativo
ADAIL RONSANI	PROPRIETARIO	23/10/2014	(48)99822370		SIM
ADAIL RONSANI	USUARIO/LOCATARIO	23/10/2014	(48)99822370		SIM

Contas										
Mês/Ano	Vencimento	Valor de Água	Valor de Esgoto	Valor de Lixo	Valor dos Débitos	Valor dos Créditos	Valor dos Impostos	Valor da Conta	Acrésc. Impont.	Sit.
01/2015	10/02/2015	25,01	0,00	0,00	0,00	0,00	0,00	25,01	2,77	NOR
02/2015	10/03/2015	25,01	0,00	0,00	1,71	0,00	0,00	26,72	2,69	NOR
03/2015	10/04/2015	25,01	0,00	0,00	0,00	0,00	0,00	25,01	2,28	NOR
04/2015	10/05/2015	25,01	0,00	0,00	0,00	0,00	0,00	25,01	2,03	NOR
05/2015	10/06/2015	25,01	0,00	0,00	0,00	0,00	0,00	25,01	1,77	NOR
Total	5 doc(s)	125,05	0,00	0,00	1,71	0,00	0,00	126,76	11,54	

Fonte: Autoria Própria.

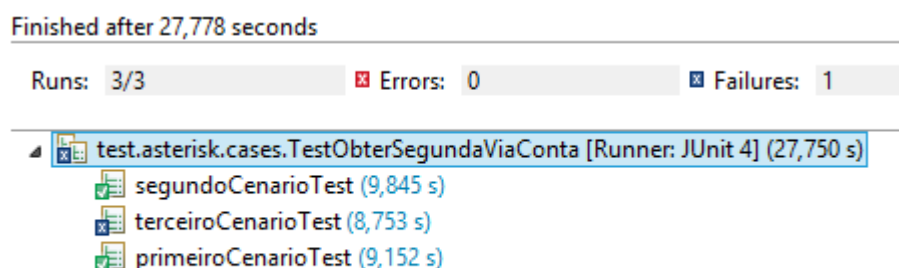
4.3 EXECUÇÃO DOS CENÁRIOS DE TESTE

A execução dos cenários propostos foi realizado utilizando o *framework* de teste JUnit citados na seção anterior, segue abaixo resultado;

4.3.1 OBTER 2ª VIA DE CONTA

Após a execução dos cenários de teste implementados para este serviço foi possível notar que 2/3 Cenários concluíram com sucesso e apenas 1/3 falhou em sua execução, a falha ocorreu no cenário 3 devido o cliente não possuir e-mail cadastrado no sistema, conforme demonstrado na figura 31.

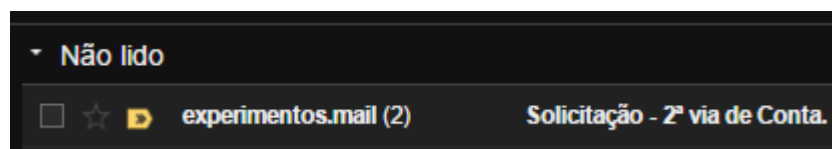
Figura 31 – Obter 2ª Via de Conta - Detalhes execução dos testes



Fonte: Autoria Própria.

Em cada cenário de teste o sistema deve enviar um e-mail contendo a 2ª via da conta solicitada, a figura 32 ilustra a caixa de entrada do E-mail configurado para cada cliente.

Figura 32 – Obter 2ª Via de Conta - E-mail recebido pelo cliente

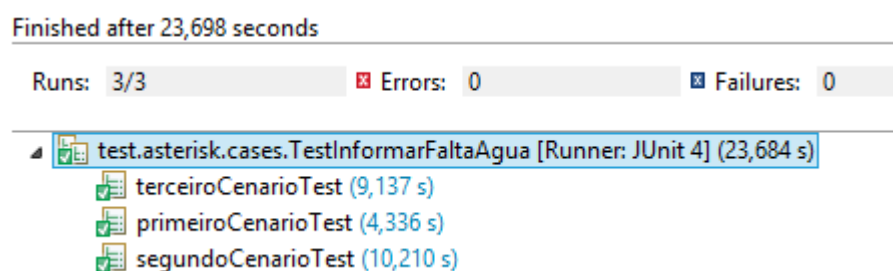


Fonte: Autoria Própria.

4.3.2 INFORMAR FALTA DE ÁGUA

Após a execução dos cenários de teste implementados para este serviço foi possível identificar que todos os 3 cenários previstos executaram com sucesso, conforme demonstrado na figura 33.

Figura 33 – Informar Falta de Água - Detalhes execução dos testes



Fonte: Autoria Própria.

Ao consulta o sistema GSAN, visando identificar se houve a ocorrência da formalização dos Registros de Atendimentos, o primeiro cenário cujo o identificador 1276833 representa a matrícula do imóvel, houve a geração do Registro de Atendimento de número 6594, conforme demonstrado na figura 34.

Figura 34 – Informar Falta de Água - RA gerado para o Cenário 1

Dados do Imóvel				
Imóvel:*	1276833		001.005.020.0017.001	
Situação de Água:	LIGADO	Situação de Esgoto:	POTENCIAL	
Dados Gerais do Registros de Atendimento				
Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
6594	6593	FALTA DE AGUA	11/11/2015	Pendente

Fonte: Autoria Própria.

Para o segundo cenário, cujo o identificador 13702840 representa a matrícula do imóvel, houve a geração do Registro de Atendimento de número 6598, conforme exposto na figura 35.

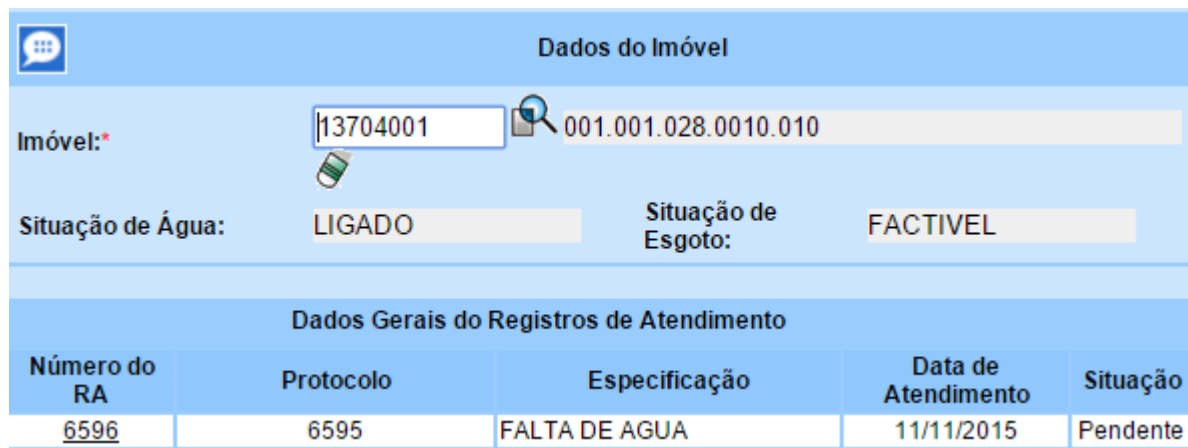
Figura 35 – Informar Falta de Água - RA gerado para o Cenário 2

Dados do Imóvel				
Imóvel:*	13702840		001.004.001.0001.003	
Situação de Água:	LIGADO	Situação de Esgoto:	FACTIVEL	
Dados Gerais do Registros de Atendimento				
Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
6598	6597	FALTA DE AGUA	11/11/2015	Pendente

Fonte: Autoria Própria.

Para o terceiro cenário, cujo o identificador 13704001 representa a matrícula do imóvel, também houve a geração do Registro de Atendimento de número 6596, conforme exposto na figura 36.

Figura 36 – Informar Falta de Água - RA gerado para o Cenário 3



Dados do Imóvel

Imóvel:

Situação de Água: Situação de Esgoto:

Dados Gerais do Registros de Atendimento

Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
6596	6595	FALTA DE AGUA	11/11/2015	Pendente

Fonte: Autoria Própria.

4.3.3 SOLICITAR RESTABELECIMENTO DA LIGAÇÃO DE ÁGUA

Após a execução dos cenários de teste implementados para este serviço foi possível identificar que todos os 3 cenários previsto executaram com sucesso, conforme demonstrado na figura 37.

Figura 37 – Restabelecimento da Ligação de Água - Detalhes execução dos testes

Finished after 19,909 seconds


Runs: 3/3 Errors: 0 Failures: 0

- test.asterisk.cases.TestSolicitarRestabelecimento [Runner: JUnit 4] (19,896 s)
 - segundoCenariotest (5,379 s)
 - primeiroCenariotest (10,223 s)
 - terceiroCenariotest (4,294 s)

Fonte: Autoria Própria.


Ao consulta o sistema GSAN, visando identificar se houve a ocorrência da formalização dos Registros de Atendimentos, o primeiro cenário cujo o identificar 1239723 representa a matrícula do imóvel, houve a geração do Registro de Atendimento de número 6604, conforme demonstrado na figura 38.

Figura 38 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 1



Dados do Imóvel

Imóvel:*



Situação de Água:

CORTADO

Situação de Esgoto:

POTENCIAL


Dados Gerais do Registros de Atendimento

Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
<u>6604</u>	6603	RESTABELIMENTO DA LIGACAO DE AGUA	11/11/2015	Pendente
<u>3326</u>	20141000002748	CORTE A PEDIDO	16/05/2014	Encerrado

Fonte: Autoria Própria.


Para o segundo cenário, cujo o identificador 80166 representa a matrícula do imóvel, houve a geração do Registro de Atendimento de número 6608, conforme exposto na figura 39.

Figura 39 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 2



Dados do Imóvel

Imóvel:*



Situação de Água:

LIGADO

Situação de Esgoto:

FACTIVEL

Dados Gerais do Registros de Atendimento

Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
<u>6598</u>	6597	FALTA DE AGUA	11/11/2015	Pendente

Fonte: Autoria Própria.

Para o terceiro cenário, cujo o identificador 13702572 representa a matrícula do imóvel, mesmo o cliente em situação de inadimplência houve a geração do Registro de Atendimento de número 6606, conforme exposto na figura 40.

Figura 40 – Restabelecimento da Ligação de Água - RA gerado para o Cenário 3

Dados do Imóvel	
Imóvel:*	13704001 001.001.028.0010.010
Situação de Água:	LIGADO
Situação de Esgoto:	FACTIVEL

Dados Gerais do Registros de Atendimento				
Número do RA	Protocolo	Especificação	Data de Atendimento	Situação
6596	6595	FALTA DE AGUA	11/11/2015	Pendente

Fonte: Autoria Própria.

Nos testes houve somente 1 falha conforme esperado no teste do Serviço Obter 2ª via de conta, onde o cliente não havia o e-mail cadastral, já nos outros testes a solução se comportou de forma estável pois todas as solicitações para Informar Falta de Água e Solicitar Restabelecimento da Ligação de Água foram atendidas e formalizadas junto ao sistema GSAN de forma automatizada.

Os detalhes sobre cada Registro de Atendimento mencionado acima podem ser consultados nas apêndices deste trabalho.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

5.1 CONSIDERAÇÕES FINAIS

A solução em *software* apresentada e desenvolvida trata-se de uma integração entre os sistemas GSAN e Asterisk, com o objetivo de automatizar os atendimentos destinados aos serviços Obter 2ª via de conta, Informar falta de água e Solicitar restabelecimento da ligação de água. Foi necessário desenvolver um *Middleware* para intermediar a comunicação entre os sistemas e posteriormente uma suíte de testes automatizadas para garantir o funcionamento da solução. Dessa forma possibilitou automatizar o atendimento dos serviços descritos acima utilizando uma Unidade de Resposta Audível com um fluxo pré-determinado e consequentemente realizar a padronização do fluxo de atendimento.

Levando em consideração os resultados obtidos através da aplicação de todos os cenários de testes descritos no capítulo anterior, sob os serviços desenvolvidos neste trabalho, é possível afirmar que a utilização da Unidade de Resposta Audível na realização dos atendimentos de primeiro nível, tem potencial para reduzir parte dos atendimentos destinados aos serviços Obter 2ª via de conta, Informar falta de água e Solicitar restabelecimento da ligação de água, além de favorecer a redução dos custos.

Toda solução apresentada será adicionada na comunidade do sistema GSAN, para que seja mantida e evoluída ao longo do tempo, servindo como contribuição para os utilizadores do software.

5.2 TRABALHOS FUTUROS

Existem alguns recursos ou rotinas que podem ser criados para melhor aproveitamento da solução proposta.

- Reconhecimento de voz, para os casos de falta de água, que ocorre em regiões fora do endereço do cliente, de forma que o cliente possa informar a Rua que ocorre o problema.
- Automatizar serviços como negociação de débitos do cliente para serem realizados através da URA.
- Automatizar o *feedback* dos serviços realizados, ou seja, a cada término de Ordem de Serviço o GSAN solicita do Asterisk que ele contate o cliente.
- Gerar estatísticas dos acessos realizados, coletando tempo de respostas e traçando perfis de cliente que mais realizam solicitações via *Call Center*.

- Realizar testes de performance utilizando a ferramenta *Sipp* para identificar a capacidade de resiliência da solução em situações adversas.
- Implantar a solução em um ambiente real.

Estes são cenários que podem contribuir para um melhor aproveitamento da solução, com potencial de redução ainda maior dos registros de atendimentos.

Referências

ALMEIDA, L. A. *Hydra: Desenvolvimento de uma aplicação ubíqua para redirecionamento de serviços no middleware uOS*. 48 p. — Universidade de Brasília, Brasília, 2011.

ASTERISK. 2014. Disponível em: <<http://www.voip-info.org/wiki/view/Asterisk+AGI>>. Acesso em: 03.08.2015.

BRASIL. *Regulamenta a Lei no 8.078, de 11 de setembro de 1990, para fixar normas gerais sobre o Serviço de Atendimento ao Consumidor - SAC*. Brasília, DF, 2008. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/decreto/d6523.htm>.

CAMPOS, H. V. A. de. *PBX-IP: Gerenciando Tecnologias de Telecomunicações*. 90 p. — Centro Universitário Positivo, Curitiba, 2007.

COMPANHIA PERNAMBUCANA DE SANEAMENTO. *Teleatendimento (Call Center)*. Recife, 2007. Disponível em: <<https://lojavirtual.compesa.com.br:8443/gsan/exibirCanaisAtendimentoCompesaAction.do?method=teleatendimento>>. Acesso em: 14.05.2014.

DARÚ, J. H. *Implantação do sistema DISC-OS*. 58 p. Monografia (Dissertação) — Sociedade Educacional de Santa Catarina, Joinville, 2008.

FOWLER, M. Model view controller. January 2003. Disponível em: <<http://martinfowler.com/eaCatalog/modelViewController.html>>. Acesso em: 10.5.2014.

JOINVILLE, A. M. de Regulacao dos Serviços de Agua e Esgotos de. *Relatorio de Analise Regulatoria da Companhia de Aguas de Joinville*. [S.l.], 2014. Disponível em: <<http://www.amae.sc.gov.br/wp-content/uploads/2014/12/Relatorio-de-Analise-Regulatoria-Agosto-2014.pdf>>. Acesso em: 14.4.2015.

MEIRELLES, F. S. Tecnologia de informação. 2014. Disponível em: <<http://easp.fgvsp.br/sites/easp.fgvsp.br/files/pesqti-gvcia2014ppt.pdf>>. Acesso em: 10.5.2014.

PORTAL DO SOFTWARE PÚBLICO BRASILEIRO. *Comunidade GSAN*. 2007. Disponível em: <http://www.portalsoftwarepublic.com.br/ver-comunidade?community_id=1593449>. Acesso em: 04.04.2014.

PROGRAMA DE MODERNIZAÇÃO DO SETOR DE SANEAMENTO (PMSS). Comunidade gsan. Abril 2015. Disponível em: <<http://www.pmss.gov.br/index.php/gsan>>. Acesso em: 09.05.2015.

SISTEMA NACIONAL DE INFORMAÇÕES SOBRE SANEAMENTO (SNIS). Diagnóstico dos serviços de Água e esgotos - 2013. Abril 2015. Disponível em: <<http://www.snis.gov.br/index.php/diagnostico-agua-e-esgotos/diagnostico-ae-2013>>. Acesso em: 09.05.2015.

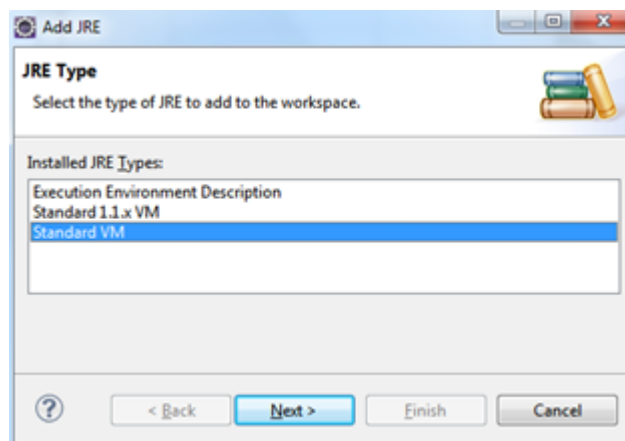
VIEIRA, G. *Módulo de integração VoIPWEB*. Monografia (Dissertação) — Universidade Federal de Santa Catarina, Florianópolis, 2007.

Apêndices

APÊNDICE A – Processo de configuração da JDK na IDE de desenvolvimento

O sistema GSAN roda sob a JDK (*Java Develop Kit*) 5, disponível atualmente no site da Oracle Coportarion, sendo necessário configurar a IDE com a versão correta para compilar o projeto, podemos configurar facilmente acessando a opção no menu superior *Window > Preferences > Java > Installed JREs* em seguida será preciso adicionar uma nova JVM (*Java Virtual Machine*) selecionando a opção Add e selecionar a opção Standard VM, conforme visto na figura ??;

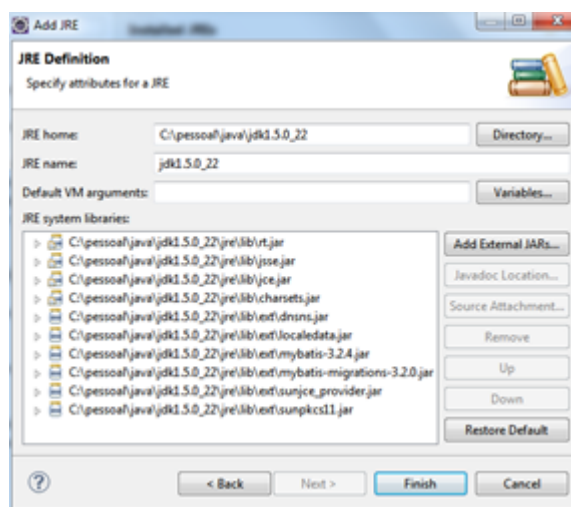
Selecionar Tipo de JVM.



Fonte: Autoria Própria.

Após a seleção é preciso confirmar a ação clicando em Next, na nova janela exibida o botão Directory... permite localizar o diretório onde está a versão do JDK que será utilizada, dessa forma deve ser selecionado o diretório raiz da versão, no exemplo representado por *C:/pessoal/java/jdk1.5.0_22*, conforme visto na figura ??;

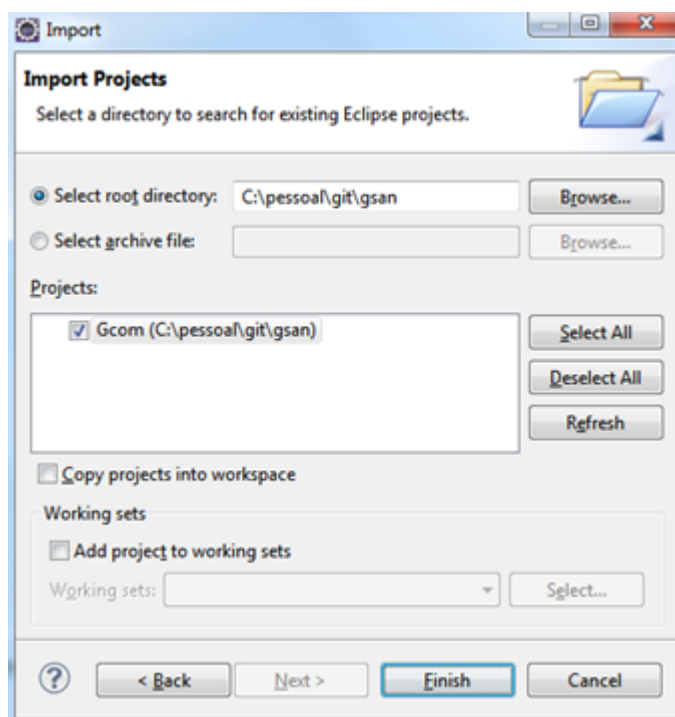
Adicionar nova JRE na IDE.



Fonte: Autoria Própria.

A própria IDE já preenche o JRE name, caso isso não ocorra será necessário preencher este campo de preferência com o nome da versão do JDK utilizada, realizado este passo a IDE terá condições de compilar as instruções contidas no fonte. Com isso será necessário importa o projeto no GSAN na IDE de desenvolvimento, selecionando a opção *File > Import...*, abrirá um janela onde deve ser selecionando a opção *General > Existing Projects into Workspace*, em seguida através do botão *Browser* será possível localizar o diretório onde está o fonte do sistema GSAN, conforme ilustrado na figura ??;

Importar Sistema GSAN na IDE.



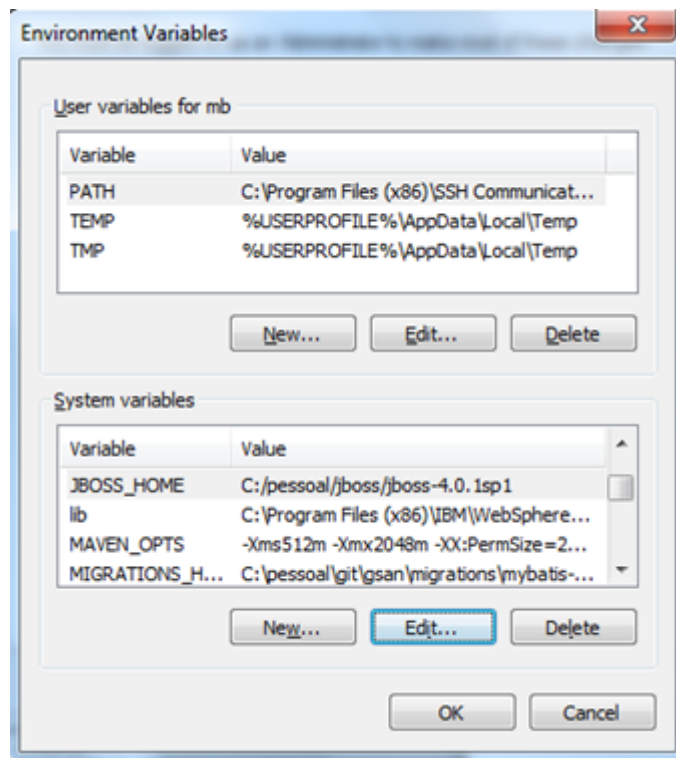
Fonte: Autoria Própria.

Feito isso precisa somente pressionar o botão de *Finish*, para confirmar a importação do projeto para a IDE de desenvolvimento.

APÊNDICE B – Configurar variável de ambiente do servidor de aplicação Jboss

Neste trabalho foi utilizado o projeto *Open Source Jboss Community* na versão 4.0.1, compatível com as tecnologias utilizadas no GSAN, para utilizá-lo será preciso declarar uma variável de ambiente no sistema operacional, por padrão nomeado de `JBOSS_HOME` contendo a localização do diretório do servidor de aplicação, no exemplo abaixo representado por `C:/pessoal/jboss/jboss-4.0.1sp1`, conforme visto na figura ??;

Adicionando variável de ambiente `JBOSS_HOME`.



Fonte: Autoria Própria.

Após a criação desta variável de ambiente, será preciso atualizar a variável de ambiente chamada `PATH` que reúne as variáveis utilizadas no sistema operacional, adicionando ao final o seguinte texto `%JBOSS_HOME%/bin;` para que o sistema operacional consiga localizar o diretório `/bin` do servidor de aplicação que contém o script de inicialização do servidor chamado `run.bat` para sistemas Windows e `run.sh` para sistemas Unix, neste script é possível modificar os parâmetros utilizados na execução do servidor de aplicação sob a JVM para a alocação de memória assim como habilitar a utilização de técnicas de debug remoto das aplicações.

APÊNDICE C – Configuração do Fluxo da URA

Segue abaixo a configuração do fluxo completo da URA, configurado no seguinte arquivo de configuração */etc/asterisk/extension.conf*;

```
; Declarando o contexto de boas vindas para executar o áudio 'inicial'
; o dígito representa a rotina associada abaixo:
; 1 – Contexto de identificação do Cliente.
; 2 – Falar com o atendente.
[disc-ivr-BOAS__VINDAS]
exten => s,1,Playback(custom/inicial)
exten => 1,1,Goto(disc-ivr-IDENTIFICACAO,7001,1)
exten => 2,1,Goto(FALAR__ATENDENTE,s,1)

; Declarando o contexto de identificação do cliente, executa o áudio
; 'identificacao' e redireciona para o contexto de PESQUISAR__CLIENTE
; executa o áudio 'pronto' e 'numero_ra', soletra o numero do RA e
; direciona para o Menu principal.
[disc-ivr-IDENTIFICACAO]
exten => 7001,1,Playback(custom/identificacao)
exten => 7001,2,Goto(PESQUISAR__CLIENTE,s,1)
exten => 7001,3,Playback(custom/pronto)
exten => 7001,4,Playback(custom/numero_ra)
exten => 7001,5,SayDigits($NUMERO__RA)
exten => 7001,6,Goto(disc-ivr-MENU,7002,1)

; Declara o context do Menu Principal, executa o áudio 'menu',
; o dígito representa a rotina associada abaixo:
; 1 – Serviço de 2ª via de Conta
; 2 – Serviço de Informar Falta de Água
; 3 – Serviço de Solicitar Restabelecimento da Ligação
; 4 – Falar com o atendente.
[disc-ivr-MENU]
exten => 7002,1,Playback(custom/menu)
exten => 1,1,Goto(disc-ivr-2VIA,7003,1)
exten => 2,1,Goto(disc-ivr-INFORMAR__FALTA__AGUA,7004,1)
```

```
exten => 3,1,Goto(disc-ivr-SOLICITAR_RESTABELECIMENTO,7005,1)
exten => 4,1,Goto(FALAR_ATENDENTE,s,1)
```

```
; Declarando o contexto de 2ª via, executa o áudio '2via'
; redireciona para o contexto de obter segunda via e
; desliga a ligação.
```

[disc-ivr-2VIA]

```
exten => 7003,1,Playback(custom/2via)
exten => 7003,2,Goto(OBTER_SEGUNDA_VIA,s,1)
exten => 7003,3,Hangup()
```

```
; Declarando o contexto disc-ivr-INFORMAR_FALTA_AGUA
; Executa o áudio 'falta_agua_abrir_chamado'
; Redireciona para o contexto INFORMAR_FALTA_AGUA
; Executa o áudio 'sucesso'
; Desliga a ligação
```

[disc-ivr-INFORMAR_FALTA_AGUA]

```
exten => 7004,1,Playback(custom/falta_agua_abrir_chamado)
exten => 7004,2,Goto(INFORMAR_FALTA_AGUA,s,1)
exten => 7004,3,Playback(custom/sucesso)
exten => 7004,4,Hangup()
```

```
; Declarando o contexto disc-ivr-SOLICITAR_RESTABELECIMENTO
; Executa o áudio 'restabelecimento'
; Redireciona para o contexto SOLICITAR_RESTABELECIMENTO
; Executa o áudio 'sucesso'
; Desliga a ligação
```

[disc-ivr-SOLICITAR_RESTABELECIMENTO]

```
exten => 7005,1,Playback(custom/restabelecimento)
exten => 7005,2,Goto(SOLICITAR_RESTABELECIMENTO,s,1)
exten => 7005,3,Playback(custom/sucesso)
exten => 7005,4,Hangup()
```

```
; Declarando o contexto 'PESQUISAR_CLIENTE'
; Atende a ligação
; Executa o áudio 'beep'
; Defini o tempo limite de espera entre a discagem dos dígitos
; Defini o tempo limite de espera do primeiro dígito
; Ler os dígitos informados
; Cria a variável 'CLIENTE_IMOVEL' recebendo os dígitos.
```

```
; Executa uma requisição Agi para '192.168.43.2/pesquisar.imovel.cliente.agi'
; Exibe o ID_IMOVEL
; Verifica a situação da requisição realizada
; Caso seja 'SUCESSO' redireciona para o contexto disc-ivr-IDENTIFICACAO
; Caso seja diferente de sucesso redireciona para FALAR_ATENDENTE
```

[PESQUISAR_CLIENTE]

```
exten => s,1,Answer()
exten => s,n,PlayBack(beep)
exten => s,n,Set(TIMEOUT(digit)=3)
exten => s,n,Set(TIMEOUT(response)=7)
exten => s,n,Read(NUMERO) ; LER OS DIGITOS
exten => s,n,Set(CLIENTE_IMOVEL=$NUMERO)
exten => s,n,Agi(agi://192.168.43.2/pesquisar.imovel.cliente.agi)
exten => s,n,NoOp($ID_IMOVEL)
exten => s,n,GotoIf($["$SITUACAO"== "SUCESSO"]?ok:falha)
exten => s,n(ok),Goto(disc-ivr-IDENTIFICACAO,7001,3)
exten => s,n(falha),Goto(FALAR_ATENDENTE,s,1)
```

```
; Declarando o contexto 'OBTER_SEGUNDA_VIA'
; Atende a ligação
; Executa uma requisição Agi para '192.168.43.2/segunda.via.agi'
; Printa o 'ID_IMOVEL'
; Verifica a situação da requisição realizada
; Caso seja 'SUCESSO' redireciona para o contexto 'disc-ivr-2VIA'
; Caso seja diferente de sucesso redireciona para 'FALAR_ATENDENTE'
```

[OBTER_SEGUNDA_VIA]

```
exten => s,1,Answer()
exten => s,n,Agi(agi://192.168.43.2/segunda.via.agi)
exten => s,n,NoOp($ID_IMOVEL)
exten => s,n,GotoIf($["$SITUACAO"== "SUCESSO"]?ok:falha)
exten => s,n(ok),Goto(disc-ivr-2VIA,7003,3)
exten => s,n(falha),Goto(FALAR_ATENDENTE,s,1)
```

```
; Declarando o contexto 'INFORMAR_FALTA_AGUA'
; Atende a ligação
; Executa uma requisição Agi para '192.168.43.2/falta.agua.agi'
; Verifica a situação da requisição realizada
; Caso seja 'SUCESSO' redireciona para o contexto
; 'disc-ivr-NFORMAR_FALTA_AGUA'
; Caso seja diferente de sucesso redireciona para 'FALAR_ATENDENTE'
```

[INFORMAR_FALTA_AGUA]

```
exten => s,1,Answer()
exten => s,n,Agi(agi://192.168.43.2/falta.agua.agi)
exten => s,n,GotoIf("$SITUACAO"== "SUCESSO"?ok:falha)
exten => s,n(ok),Goto(disc-ivr-INFORMAR_FALTA_AGUA,7004,3)
exten => s,n(falha),Goto(FALAR_ATENDENTE,s,1)
```

```
; Declarando o contexto 'SOLICITAR_RESTABELECIMENTO'
; Atende a ligação
; Executa uma requisição Agi para '192.168.43.2/restabelecimento.ligacao.agi'
; Verifica a situação da requisição realizada
; Caso seja 'SUCESSO' redireciona p/ o contexto
; 'disc-ivr-SOLICITAR_RESTABELECIMENTO'
; Caso seja diferente de sucesso redireciona para 'FALAR_ATENDENTE'
```

[SOLICITAR_RESTABELECIMENTO]

```
exten => s,1,Answer()
exten => s,n,Agi(agi://192.168.43.2/restabelecimento.ligacao.agi)
exten => s,n,GotoIf("$SITUACAO"== "SUCESSO"?ok:falha)
exten => s,n(ok),Goto(disc-ivr-SOLICITAR_RESTABELECIMENTO,7005,3)
exten => s,n(falha),Goto(FALAR_ATENDENTE,s,1)
```

```
; Declarando o contexto 'FALAR_ATENDENTE'
; Executa o áudio 'ligacao_redirecionada'
; Realiza uma ligação para o 'ATENDENTE'
```

[FALAR_ATENDENTE]

```
exten => s,1,PlayBack(custom/ligacao_redirecionada)
exten => s,2,Dial(SIP/ATENDENTE)
```

APÊNDICE D – Registros de Atendimentos Gerados

A execução dos cenários de testes resultaram na criação dos seguintes Registros de Atendimentos no sistema GSAN, a seguir pode ser consultado o detalhamento sobre os Registros de Atendimentos gerados para o serviço Informar Falta de Água.

Informar Falta de Água - Cenário 1 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6594 Situação do RA: PENDENTE
Número Manual:	000000659-9
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	9 FALTA DE AGUA
Especificação:	28 FALTA DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:29
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.

Informar Falta de Água - Cenário 2 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6598 Situação do RA: PENDENTE
Número Manual:	000000659-9
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	9 FALTA DE AGUA
Especificação:	28 FALTA DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:29
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.

Informar Falta de Água - Cenário 3 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6596 Situação do RA: PENDENTE
Número Manual:	000000659-9
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	9 FALTA DE AGUA
Especificação:	28 FALTA DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:29
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.

A seguir pode ser consultado os registros de atendimentos abertos pela execução da suíte de testes automatizados para o serviço Solicitar Restabelecimento da Ligação de

Água;

Solicitar Restabelecimento da Ligação de Água - Cenário 1 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6604 Situação do RA: PENDENTE
Número Manual:	000000660-2
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	3 CORTE/RELIGACAO/SUPRESSAO
Especificação:	83 RESTABELICIMENTO DA LIGACAO DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:38
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.

Solicitar Restabelecimento da Ligação de Água - Cenário 2 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6608 Situação do RA: PENDENTE
Número Manual:	000000660-2
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	3 CORTE/RELIGACAO/SUPRESSAO
Especificação:	83 RESTABELICIMENTO DA LIGACAO DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:38
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.

Solicitar Restabelecimento da Ligação de Água - Cenário 3 Detalhado.

Dados Gerais do Registro de Atendimento	
Número do RA:	6606 Situação do RA: PENDENTE
Número Manual:	000000660-2
Usuário que Abriu RA:	7 FABRICIO RODRIGUES DE OLIVEIRA
Tipo de Solicitação:	3 CORTE/RELIGACAO/SUPRESSAO
Especificação:	83 RESTABELICIMENTO DA LIGACAO DE AGUA
Tipo de Atendimento:	<input checked="" type="radio"/> on-line <input type="radio"/> manual
Data do Atendimento:	11/11/2015
Hora do Atendimento:	01:38
Tempo de Espera para Atendimento:	às
Data Prevista:	13/11/2015
Valor Sugerido:	0,00
Meio de Solicitação:	8 INTERNET
Unidade de Atendimento:	1 UNIDADE ORGANIZACIONAL
Unidade Atual:	1 UNIDADE ORGANIZACIONAL
Observação:	Atendimento realizado via URA.

Fonte: Autoria Própria.