

Fundação Centro de Análise, Pesquisa e Inovação Tecnológica.

Instituto de Ensino Superior FUCAPI

Sistemas de Informação

CALL CENTER AUTOMATIZADO: INTEGRANDO UM PABX BASEADO EM ASTERIX COM O GSAN

Autor: Marcos Roberto Garcia Bahiense Junior

Orientador: (M.Sc, Carlos Augusto Mar)

Manaus, AM

2015



Marcos Roberto Garcia Bahiense Junior

CALL CENTER AUTOMATIZADO: INTEGRANDO UM PABX BASEADO EM ASTERIX COM O GSAN

Monografia submetida ao curso de graduação em (Sistemas de Informação) do Instituto de Ensino Superior FUCAPI – CESF como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação. Área de concentração: Desenvolvimento e Análise de software.

Fundação Centro de Análise, Pesquisa e Inovação Tecnológica.

Instituto de Ensino Superior FUCAPI

Orientador: (M.Sc, Carlos Augusto Mar)

Manaus, AM

2015

Dedicatória.

Dedico este trabalho a todos aqueles que vão à luta sem medo de perder a batalha, que correm atrás dos seus sonhos e objetivos mesmo que pareça impossível alcança-los, a todos aqueles que fazem o bem ao próximo sem esperar recompensas e a todos aqueles que mesmo em tempos difíceis, acreditam que dias melhores estão por vir.

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, meu amigo Wellington que me fez enxergar o potencial deste trabalho e ao meu orientador Carlos Mar que acreditou que seria possível e muito ajudou na concretização deste trabalho..

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

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 de Conta, Informar Falta de Água e Solicitar Restabelecimento da Ligação, 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 agente intermediário responsável pela comunicação via protocolos SOAP e 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 redução de 20,46% dos registros de atendimentos diários.

Palavras-chaves: GSAN. Asterisk. Call Center.

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 agent responsible for communicating via SOAP protocols and 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 a reduction of 20.46% of daily attendance records.

Key-words: GSAN. Asterisk. Cell Center.

Lista de ilustrações

Figura 1 – Gráfico da Quantidade de Reclamações Mensais da CAJ	19
Figura 2 – Implantações do Sistema GSAN	25
Figura 3 – Arquitetura Detalhada do Sistema GSAN	29
Figura 4 – Exemplo de configuração do script build	31
Figura 5 – Execução do script de build	31
Figura 6 – Executando o sistema GSAN	32
Figura 7 – Acessando página inicial do Sistema	33
Figura 8 – Acessando página inicial da Interface WEB do Disc-OS	35
Figura 9 – Diagrama de implantação da solução	41
Figura 10 – Interface dos serviços automatizados.	44
Figura 11 – Declaração da servlet do WebService.	44
Figura 12 – Declaração do <i>EndPoint</i> dos serviços.	44
Figura 13 – Mapeamento dos serviços para consumo via Agi	45
Figura 14 – Geração do código fonte para consumo do WebService.	47
Figura 15 – Executando o sistema Integrador.	47
Figura 16 – Cadastro de um Ramal SIP.	48
Figura 17 – Diagrama do Fluxo da Unidade de Resposta Audível	50
Figura 18 – Softphone Zoiper sendo executado	51
Figura 19 – Configurar Ramal no Zoiper	52
Figura 20 – Média de atendimentos diários em companhias de saneamento	54
Figura 21 – Gráfico de avaliação dos serviços automatizados	59

Lista de tabelas

Tabela 1 – Principais Atributos do Imóvel	27
Tabela 2 – Tecnologias utilizadas	42

Lista de abreviaturas e siglas

AGI	Asterisk Gateway Interface
AMI	Asterisk Manager Interface
CAER	Companhia de Água e Esgotos de Roraima
CAERN	Companhia de Água e Esgotos do Rio Grande do Norte
CAJ	Companhia de Águas de Joinville
COMPESA	Companhia Pernambucana de Saneamento
CSS	Cascading Style Sheets
EAR	Enterprise Archive
EJB	Enterprise Java Beans
GPL	General Public Licence
GSAN	Sistema Integrado de Gestão de Serviços de Saneamento
IAX	Inter Asterisk eXchange
IDE	Integrated Development Environment
IP	Internet Protocol
JAXB	Java Architecture for XML Binding
JAX-WS	Java API for XML-Based Web Services
JDK	Java Development Kit
JEE	Java Enterprise Edition
JMS	Java Message Service
JRE	Java Runtime Environment
JSP	Java Server Pages
JVM	Java Virtual Machine
MC	Ministério das Cidades

MVC	Model View Controller
ORM	Object-relational Mapping
OS	Operating System
OS	Ordem de serviço
PA	Posto de Atendimento
PABX	Private Automatic Branch Exchange
PMSS	Programa de Modernização do Setor de Saneamento
QoS	Quality of Service
RA	Registro de Atendimento
SGBD	Sistema de Gerenciamento de Banco de Dados
SIP	Session Initiation Protocol
SNIS	Sistema Nacional de Informações do Setor de Saneamento
SNSA	Secretaria Nacional de Saneamento Ambiental
SOAP	Simple Object Access Protocol
URA	Unidade de Resposta Audível
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
WAV	WAVE
WEB	World Wide Web
WSDL	Web Services Description Language
XML	eXtensible Markup Language.

Sumário

	Introdução	17
0.1	Problema	17
0.2	Objetivo	18
0.2.1	Objetivo Geral	18
0.2.2	Objetivo Específico	18
0.3	Justificativa	19
0.4	Aspectos de Inovação	20
0.5	Trabalhos Relacionados	21
0.6	Metodo de Investigação	22
0.7	Estruturação da Monografia	22
1	FUNDAMENTAÇÃO TEÓRICA	25
	Fundamentação Teórica	25
1.1	GSAN	25
1.1.1	GSAN Conceitos	26
1.1.2	GSAN Arquitetura	28
1.1.3	GSAN Configuração	30
1.2	Asterisk	33
1.2.1	Asterisk Conceitos	34
1.2.2	Asterisk Instalação	34
2	REVISÃO BIBLIOGRÁFICA	37
	Revisão Bibliográfica	37
2.1	Tecnologias Utilizadas	37
2.1.1	GSAN - Detalhamento Técnico	37
2.1.2	Asterisk - Detalhamento Técnico	38
2.1.3	Simple Object Access Protocol	38
2.1.4	Asterisk Gateway Interface	38
2.1.5	WebServices	38
2.1.6	Unidade de Resposta Audível	39
2.1.7	Middleware	39
2.1.8	Disc-OS	39
2.1.9	Codec	39
3	PROCESSO DE INTEGRAÇÃO	41

Processo de Integração	41
3.1 Definição da Solução	41
3.2 Tecnologias Utilizadas	42
3.3 Etapas da Integração	42
3.3.1 Implementação de WebServices no GSAN	43
3.3.2 Desenvolver <i>Middleware</i>	45
3.3.3 Customizar o Asterisk	48
4 RESULTADOS ALCANÇADOS	53
Resultados Alcançados	53
4.1 Hardware da implantação	53
4.2 Estabelecer Métricas	53
4.3 Cenários de Testes	54
4.3.1 Obter 2ª Via de Conta	55
4.3.2 Informar Falta de Água	56
4.3.3 Solicitar Restabelecimento da Ligação de Água	57
4.4 Resultado dos Cenários de Teste	58
4.5 Apresentação dos Resultados Obtidos	58
5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	61
Considerações Finais e Trabalhos Futuros	61

Introdução

A Secretaria Nacional de Saneamento Ambiental do Ministério das Cidades (SNSA/MC) visando elevar o nível de desempenho e de 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, 2014) efetuou a contratação de uma empresa de tecnologia da informação brasileira para executar o projeto concebido. Nesse cenário, surge então o sistema GSAN (Sistema Integrado de Gestão de Serviços de Saneamento) 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, 2014). 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, 2014), 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, 2015), com intuito de apoiar o negócio, como é o caso do software Asterisk que implementa facilidades no uso de tecnologias como PABX (*Private Automatic Branch Exchange*) tanto para linhas telefônicas convencionais como também por VoIP (*Voice over Internet Protocol*) que utiliza a transmissão de voz sobre um rede IP (*Internet Protocol*) 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, resultando em um custo muito inferior a outras soluções do gênero.

0.1 Problema

Atualmente o GSAN, atende grande parte das companhias de saneamento brasileiras, como é o caso de companhias como por exemplo CAERN (Companhia de Água e Esgotos do Rio Grande do Norte), CAER (Companhia de Água e Esgotos de Roraima),

COMPESA (Companhia Pernambucana de Saneamento), MANAUS AMBIENTAL entre outras citadas no referencial teórico deste trabalho. 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 ao público, existem lacunas que ainda precisam ser atendidas de forma plena, principalmente organizações altamente demandadas pela população. Elas são 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, pois todo o atendimento de primeiro nível normalmente é 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 muitas 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. Diversos fatores contribuem para este cenário, assim como a dificuldade em manter um feedback rápido com o cliente, a falta de canais de comunicações flexíveis que permitam uma disponibilidade inclusive fora do horário comercial, a ineficiência na triagem dos atendimentos, são questões rotineiras enfrentadas nos cotidianos das empresas

0.2 Objetivo

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

0.2.1 Objetivo Geral

Desenvolver uma integração do sistema GSAN (Sistema Integrado de Gestão de Serviços de Saneamento) com uma ferramenta de PABX (*Private Automatic Branch Exchange*) que permita o atendimento automático de chamadas telefônicas via tecnologia VoIP para reduzir parte dos atendimentos destinados à solicitação de 2ª via de conta, falta de água e restabelecimento da ligação.

0.2.2 Objetivo Específico

Para alcance deste objetivo, fazem necessários os seguintes passos:

- Estabelecer métricas de atendimento, para os serviços automatizados com base em panoramas dos relatórios de análise regulatória.

- Implementar a integração entre o sistema GSAN com uma ferramenta de PABX.
- Implantar o sistema em ambiente isolado, para realizar o atendimento automático de serviços, como: Solicitação de 2ª. Via de conta, falta de água e reestabelecimento de ligação.
- Avaliar a solução quanto aos serviços automatizados.

0.3 Justificativa

A integração entre sistema GSAN com uma ferramenta de PABX será um experimento de cunho prático, realizado para atender a uma grande demanda do setor de saneamento brasileiro, que atualmente sofre com a dificuldade em fornecer uma comunicação efetiva e eficiente através de seu sistema de informação principal que atenda as expectativas dos clientes. Com base nas informações disponibilizadas, no Relatório de Análise Regulatória da Companhia de Águas de Joinville (CAJ) (AMAE, 2015) 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, conforme demonstrado na figura 1 abaixo:

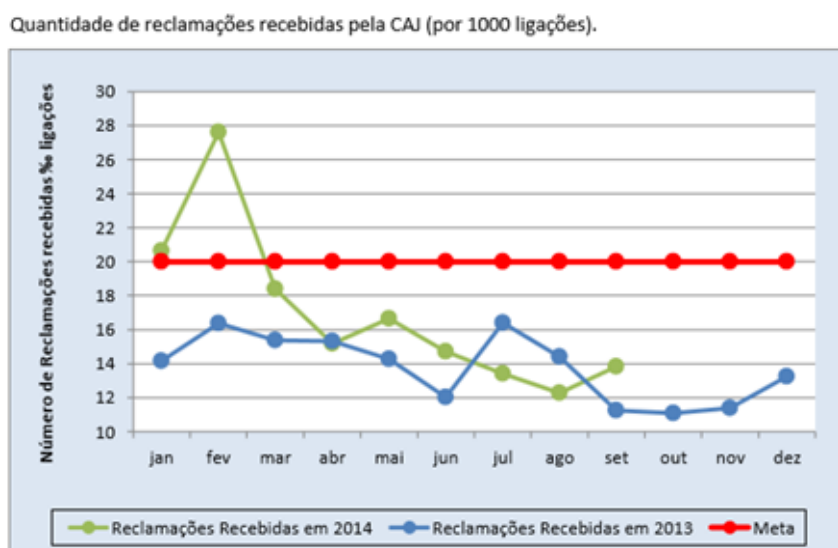


Figura 1 – Gráfico da Quantidade de Reclamações Mensais da CAJ

Fonte – Companhia Águas de Joinville, 2014

Em 2013 o indicador de Número de Reclamações apresentou uma média anual de 13,78 reclamações/mil ligações de água, com uma quantidade notória de reclamações

diárias se tornar custoso atender a todas solicitações individualmente utilizando somente atendentes sem que haja otimizações nos atendimentos, refletindo na média anual do tempo de espera das ligações para o *Call Center* da companhia que no ano de 2013 que apresentou o tempo de 75,2 segundos por atendimento, deixando evidente o quão necessário se faz adotar medidas de melhorias nos sistemas de *Call Center*. Conforme informações disponíveis no Sistema Nacional de Informações do Setor de Saneamento (SNIS) (SNIS, 2014), especificamente a Região Norte do país possui um dos piores índices de perda de faturamento do país, consequentemente isso gera lucros menores e ineficiência 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, será possível tornar o próprio sistema principal de uma empresa de saneamento o GSAN, capaz de suprir através dos recursos da ferramenta Asterisk a necessidade em disponibilizar de forma prática e padronizada o acesso a informações geradas e mantidas pela empresa, consequentemente possibilidade de redução de custo com a utilização de uma unidade de resposta audível para realizar o atendimento de primeiro nível. Propiciando ao cliente final um melhor e mais efetivo relacionamento com a empresa prestadora de serviços.

0.4 Aspectos de Inovação

O trabalho de pesquisa e desenvolvimento se trata de uma integração entre software totalmente distintos, com tecnologia *Open Source*, onde juntos serão capazes de atender à uma demanda existente no setor de saneamento brasileiro relacionado ao contexto de Atendimento ao Público. Com a integração entre o sistema GSAN com o software Asterisk, será possível transferir os atendimentos destinado a central de atendimento, para uma Unidade de Resposta Audível (URA), executando a triagem das solicitações de forma padronizada e organizada para as solicitações referentes aos seguintes tipos de serviços Obter 2ª via de Conta, Informar Falta de Água e Solicitar Restabelecimento de Ligação, será possível realizar de forma automatizada o atendimento, sem que haja intervenção humana durante o processo. Não há oficialmente uma versão publicada na comunidade do sistema GSAN capaz de atender esta demanda de forma igual ou superior, inovando em propor uma solução viável e eficiente de baixo custo, para possibilitando a melhoria no atendimento ao cliente, se destacando em permitir que o cliente possa realizar suas solicitações em qualquer horário do dia ou noite, sem ter que se locomover a empresa e acessar informações de débitos pendentes de forma rápida e padronizada.

0.5 Trabalhos Relacionados

Este trabalho de pesquisa e desenvolvimento se assemelha ao trabalho descrito por Guilherme, (Vieira, 2007) que também utilizou recursos do programa 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 dos diversos recursos do Asterisk, há divergência no objetivo onde este se destaca o fator de realizar uma integração com outro software, visando trazer uma 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 descreve 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 desenvolvido por Humberto (CAMPOS, 2007), utilizou os principais recursos do software Asterisk para realizar uma integração com um sistema externo que calcula os valores de cada ligação realizada, módulo chamada de “tarifador” além de exibir os valores em um hardware próprio, tal integração utilizou como referência tabelas em banco de dados para reconhecer eventos ocorridos e ações a serem tomadas, assemelhando-se a este trabalho pelo fato de utilizar recursos do Asterisk para disparar ações de sistemas externos, no entanto a forma de integração retratada acima se diferencia da forma adotada neste trabalho, que utiliza o protocolo de interface de serviço AGI disponibilizado para comunicação com sistemas externos, o próprio Asterisk irá disparar ações a serem realizadas por meio deste protocolo. Atualmente a empresa de saneamento Companhia Pernambucana de Saneamento (COMPESA, 2014) 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 a 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 *Digium*.

0.6 Metodo de Investigação

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

- Pesquisa bibliográfica para obter o embasamento teórico sobre funcionamento dos sistemas envolvidos
- Identificação das possíveis formas de integração entre ambos;
- Desenvolvimento da integração entre os sistemas;
- Realizar experimentações em ambiente isolado com cenários representando o atendimento de serviços de saneamento;
- Avaliação dos resultados obtidos nos experimentos quanto à redução da taxa de atendimento realizado por atendentes.

0.7 Estruturação da Monografia

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

Capítulo 2 – Fundamentação Teórica – Este capítulo tem como objetivo abordar alguns dos conceitos do saneamento brasileiro e como o sistema GSAN está construído para atendê-lo, quais os módulos que compõem o sistema e detalhar a arquitetura do sistema, assim como expor os conceitos que envolvem o software Asterisk.

Capítulo 3 – Revisão Bibliográfica – Será apresentado os principais conceitos utilizados como base no desenvolvimento deste trabalho.

Capítulo 4 – Processo de Integração – Trata-se da implementação realizada para integração entre os sistemas, apresentando as principais etapas para elaboração da comunicação entre os sistemas.

Capítulo 5 – Resultados Alcançados – Tem como característica a realização das experimentações e descrição dos resultado obtidos como resultado da integração entre o sistema GSAN com o *software* Asterisk.

Capítulo 6 – Considerações Finais e Trabalhos Futuros – Finalmente, no quinto capítulo, apresentam-se a conclusão e recomendações para trabalhos futuros, reunindo os comentários finais deste trabalho de pesquisa.

Cronograma

A seguir será apresentado o cronograma com o planejamento mensal do início e termino das atividades previstas para conclusão do trabalho de pesquisa.

Atividade	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Pesquisa bibliográfica.	●	●	-	-	-	-	-	-
Identificar formas de integração.	-	●	●	-	-	-	-	-
Implementar a integração.	-	-	●	●	●	-	-	-
Realizar experimentos.	-	-	-	-	●	●	●	-
Escrita da monografia.	●	●	●	●	●	●	●	●
Apresentação da Monografia.	-	-	-	-	-	-	-	●

1 Fundamentação Teórica

Neste capítulo será apresentado o sistema GSAN e software Asterisk, com a intenção de familiarizar o leitor com as notações que serão amplamente utilizadas no decorrer do trabalho.

1.1 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 2:

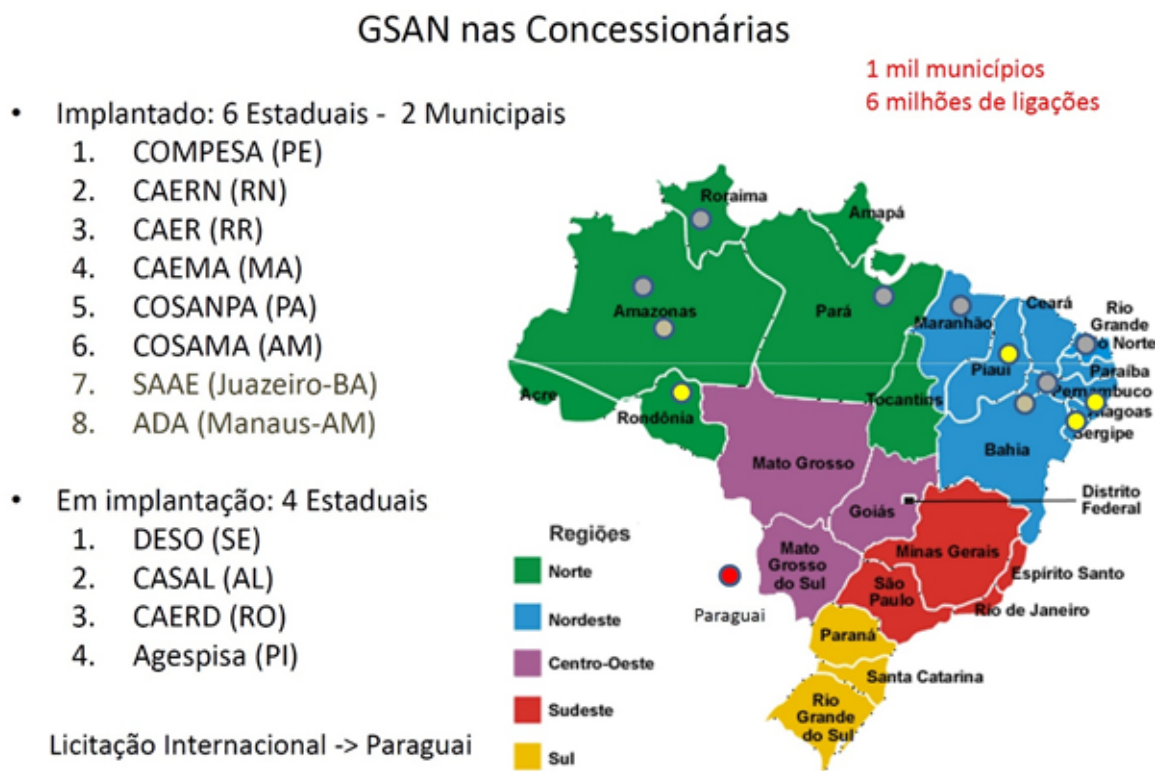


Figura 2 – Implantações do Sistema GSAN

Fonte – Portal do Software Público Brasileiro, 2014

Conforme demonstrado acima o sistema está em funcionamento em aproximadamente 1 mil municípios brasileiros, concentrado principalmente na região Nordeste e Norte 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.

1.1.1 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):

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

Tabela 1 – Principais Atributos do Imóvel

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 a 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.

1.1.2 GSAN Arquitetura

O Sistema GSAN foi desenvolvido fundamentalmente utilizando a plataforma JEE (*Java Enterprise Edition*), 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, *Enterprise Java Beans* (EJB), *Java Message Service* (JMS) API, *Java Server Pages* 2.1, entre outros. 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 3:

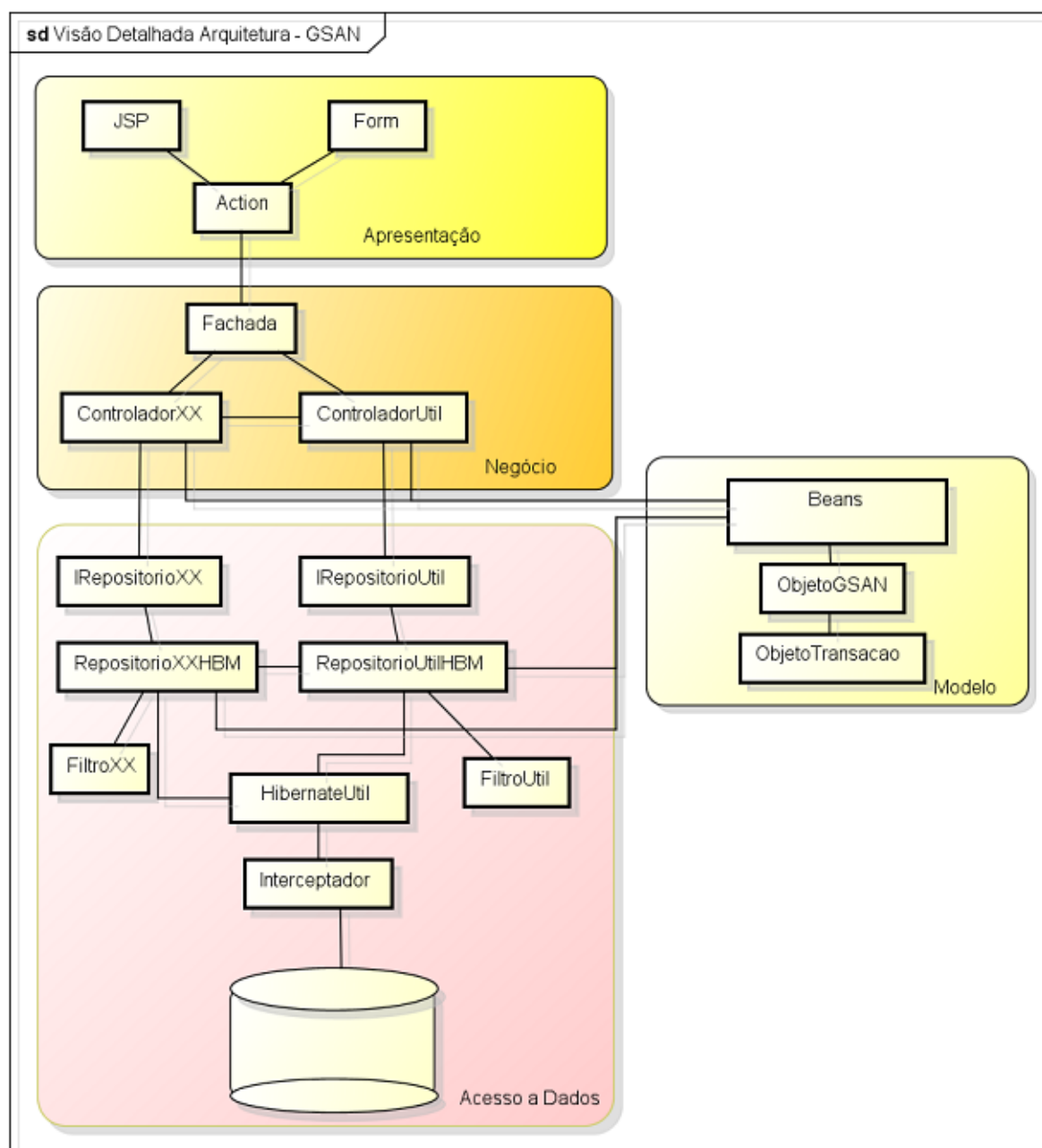


Figura 3 – Arquitetura Detalhada do Sistema GSAN

Fonte – Autoria Própria

A camada de apresentação utiliza recurso nativo da plataforma JEE para Web, sendo a JSP (*Java Server Pages*) 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*. 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* (EJB) versão 2.1. As classes de Repositório são classes da aplicação que utilizam o padrão de projeto *Singleton*, que garante a existência de apenas uma instância desse objeto no sistema, 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. Fazendo uso de padrões de projetos renomados, tornar o código mais organizado e entendível, facilitando futuras manutenções, a utilização do padrão MVC (*Model View Controller*) como estrutura principal faz com que exista isolamentos entre as camadas de Modelo, Visualização e Controler da aplicação, tornando a organização de pacotes bem estruturada, porém não houve um bom reaproveitamento de templates, por conta disso foram geradas muitas páginas específicas por funcionalidade e consequentemente muitos arquivos de configuração para o *framework* Struts que mapeia as requisições via arquivos no formato XML (*eXtensible Markup Language*).

1.1.3 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 para a comunidade de desenvolvedores e interessados. Com o código fonte em mãos será necessário o auxílio de uma IDE (*Integrated Development Environment*) 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 já possui alguns scripts de build para serem executados, no entanto é preciso configurar os locais adequados para geração do pacote, conforme segue o exemplo abaixo na figura 4:

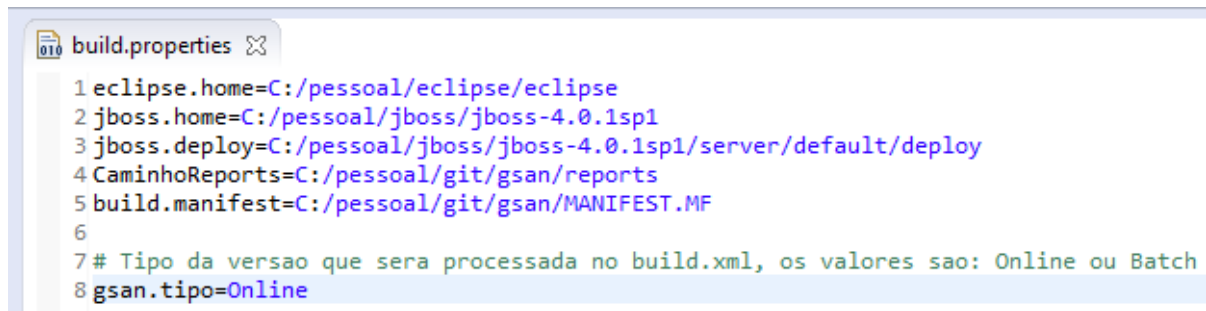


Figura 4 – Exemplo de configuração do script build

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 5:

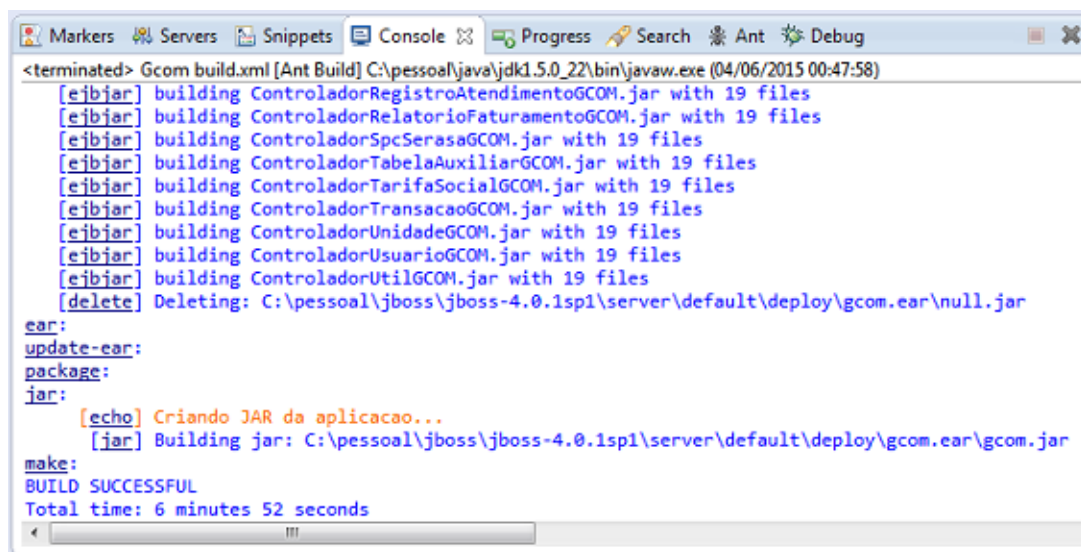
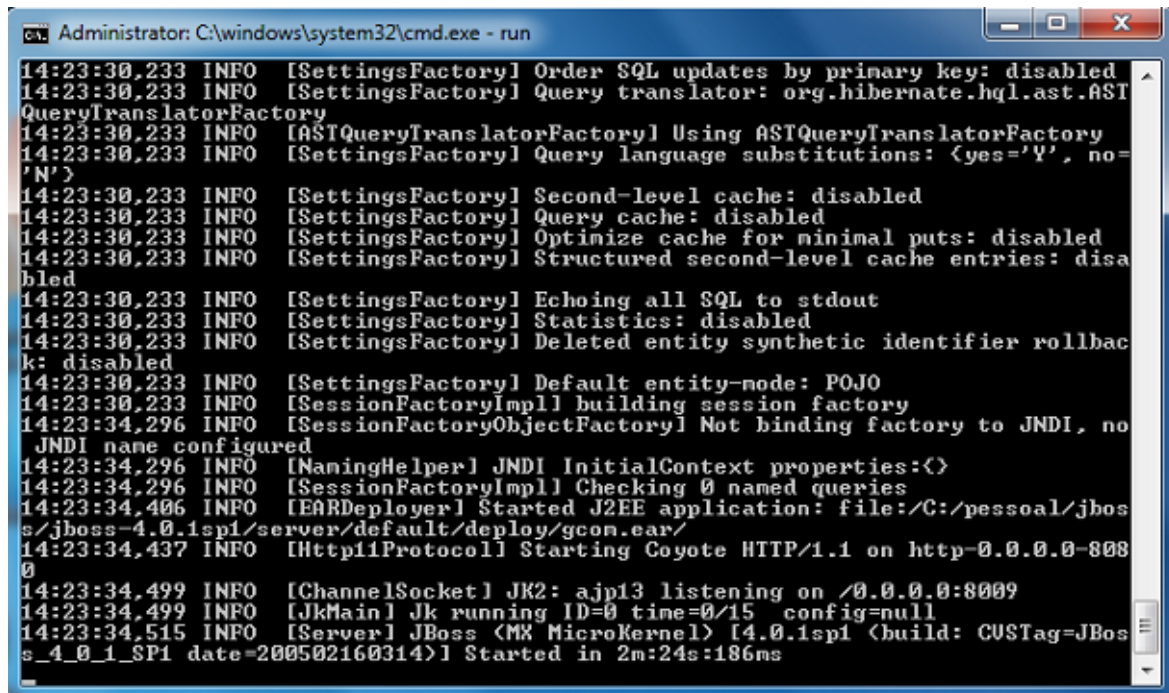


Figura 5 – Execução do script de build

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 6:



```

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-mode: 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/gcom.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

```

Figura 6 – Executando o sistema GSAN

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 7 abaixo:

A credencial de acesso criada por padrão é:

Login: admin

Senha: gcom

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

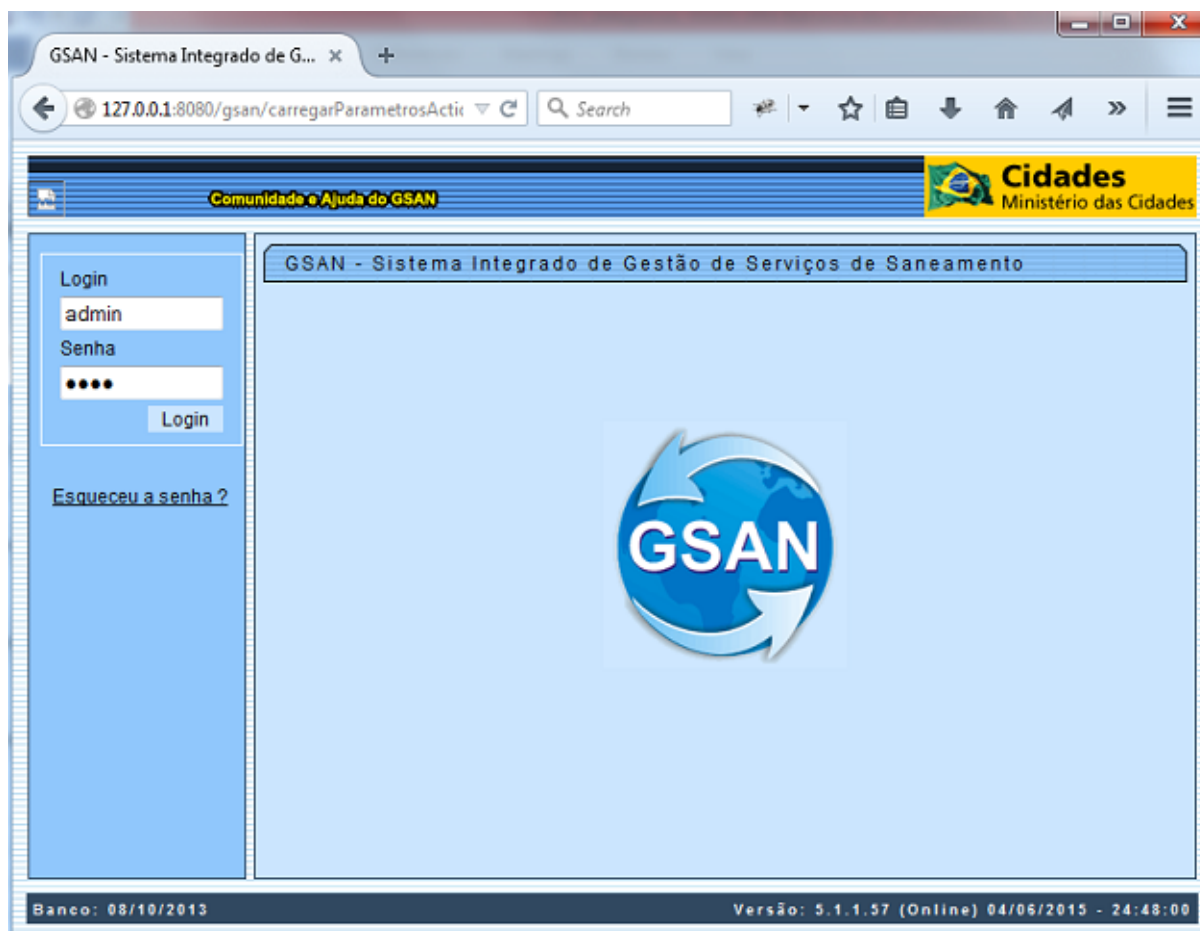


Figura 7 – Acessando página inicial do Sistema

Fonte – Autoria Própria

1.2 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 publica. Com tantos recursos disponíveis e sendo multiplataforma 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.

1.2.1 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 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” utilizando 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.

1.2.2 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 será 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 8 abaixo:

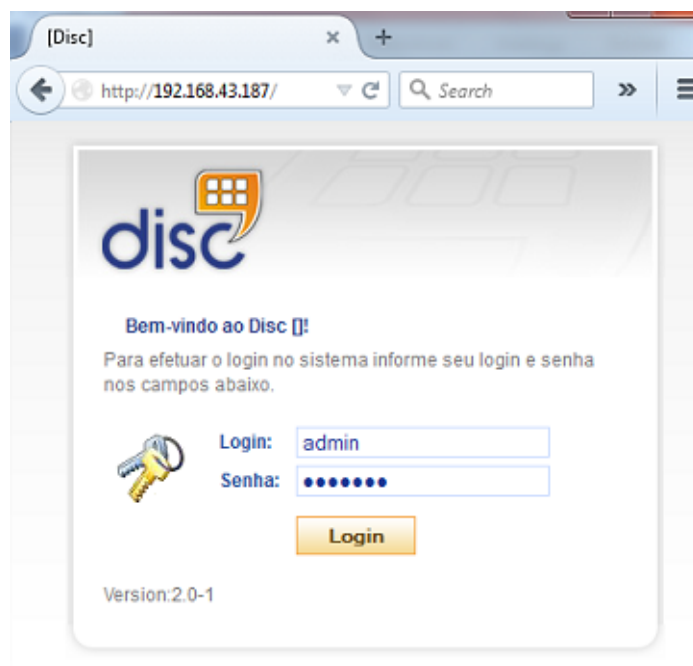


Figura 8 – Acessando página inicial da Interface WEB do Disc-OS

Fonte – Autoria Própria

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

Login: admin

Senha: disc-os

2 Revisão Bibliográfica

Neste capítulo será apresentado o conceito dos frameworks utilizados neste trabalho, além de justificar o motivo que levou a ser utilizados e destacar vantagens obtidas em seu uso.

2.1 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.

2.1.1 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:

Hibernate: 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. (Hibernate, 2014).

Apache Struts: Tem como característica principal a sua utilização na construção de modernos 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, 2015).

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, 2014), essencialmente o container *Enterprise Java Bean* (EJB), *Java Server Pages* (JSP) e *Servlets* que são executadas dentro de um servidor de aplicação Java EE.

2.1.2 Asterisk - Detalhamento Técnico

A ferramenta de código aberto Asterisk, 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 conferências, distribuição automática de chamadas (Asterisk, 2014), 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 GPL (*General Public Licence*), 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.

2.1.3 Simple Object Access Protocol

O protocolo SOAP (Simple Object Access Protocol) 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 HTTP (Hypertext Transfer Protocol) e RPC (Remote Procedure Call), possibilitando a utilização para realizar integrações entre softwares.

2.1.4 Asterisk Gateway Interface

Um dos protocolos de interface de comunicação com o software Asterisk chamado AGI (*Asterisk Gateway Interface*), tem o objetivo de possibilitar que o software consiga consumir recursos externos através de requisição semelhantes ao CGI (*Common Gateway Interface*) de servidores web, no entanto de forma inversa as requisições são originadas pelo próprio Asterisk (Asterisk-Java, 2015).

2.1.5 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 URI

(*Uniform Resource Identifier*) identifico a Web convencional, descritos e definidos usando XML (*Extensible Markup Language*) (WebService, 2015).

2.1.6 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 (Asterisk, 2014) e (VIEIRA, 2007).

2.1.7 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 (??)

2.1.8 Disc-OS

O Disc-OS refere-se a uma distribuição Linux (Cent-OS), customizada para utilização de PABX e PABX IP, abstrai toda a configuração de bibliotecas básicas e instalação do Asterisk (Disc-OS, 2015). Disponibiliza uma interface Web para configuração dos recursos, são algumas das vantagens, possuir disponível no idioma português brasileiro, além de tornar bem prático o processo de configurações.

2.1.9 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).

3 Processo de Integração

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

3.1 Definição da Solução

Após o estudo aprofundado sobre o sistema GSAN e software Asterisk foram identificadas diversas formas de realizar a integração, entre os sistemas pelo fato da existência de várias 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 SOAP e AGI para serem implementados por um *Middleware*, este será responsável em assumir o papel de intermediário entre os sistemas GSAN e Asterisk. Com o intuito de facilitar o entendimento da comunicação entre os sistemas, a seguir será exposto o diagrama de implantação da solução (figura 9) descrita acima, explanando os principais detalhes adotados nesta integração:

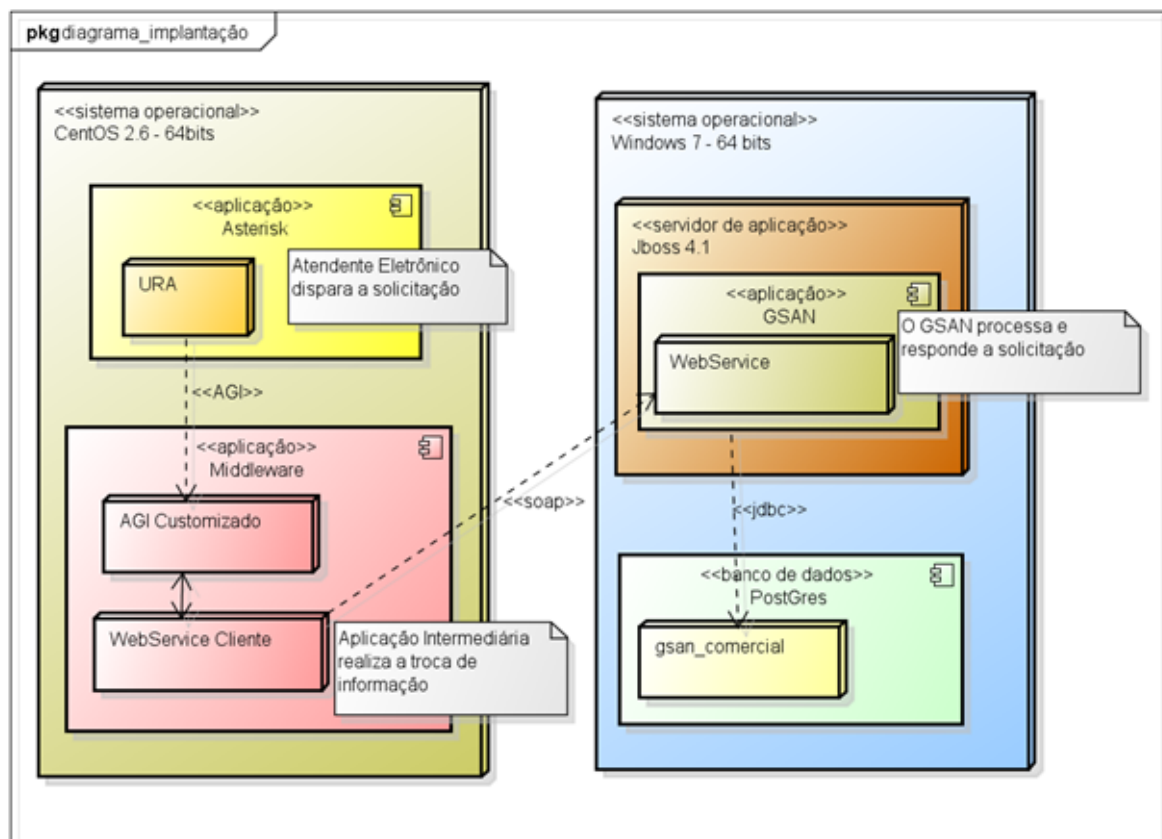


Figura 9 – Diagrama de implantação da solução

Fonte – Autoria Própria

Conforme ilustrado acima, o sistema GSAN irá prover uma interface de serviços na forma de WebServices utilizando o protocolo de comunicação SOAP (*Simple Object Access Protocol*), tais serviços serão consumidos através do *Middleware* intermediário denominado integrador que além de consumir os serviços do sistema GSAN, deverá também prover uma interface de serviços na forma utilizando o protocolo AGI, para então ser consumidos pelo Asterisk e responder a solicitação da Unidade de Resposta Audível.

3.2 Tecnologias Utilizadas

O processo de integração será 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.

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 XM.L	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

Tabela 2 – Tecnologias utilizadas

Fonte – Autoria Própria

3.3 Etapas da Integração

O processo de integração entre os sistemas será composto por três etapas principais, das quais serão necessários para compor a solução escolhida, conforme definido abaixo e descrito nas sessões seguintes:

- Implementar WebServices no Sistema GSAN
- Desenvolver Middleware

- Customizar o Asterisk

3.3.1 Implementação de WebServices no GSAN

A estratégia de desenvolver o WebService internamente ao sistema GSAN, foi adotada, visando a possibilidade de reaproveitamento de toda regra 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-WS (*Java API for XML-Based Web Services*) para serem consumidos por sistemas externos através de submissões de arquivos do tipo XML, definidas no padrão de comunicação SOAP. Para tratar de conversões de arquivos XML para objetos e vice-versa está sendo utilizada a especificação do padrão JAXB (*Java Architecture for XML Binding*), dessa forma estes serviços serão executados no mesmo servidor de aplicação. Na prática será 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 serialização 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 atender as necessidades do *Call Center* (figura 10):

Para definir a URL (*Uniform Resource Locator*) 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 11:

Feito isso, será preciso disponibilizar os novos serviços, definindo a classe concreta de implementação e o padrão de acesso que será adotado para a interface criada, para que

```

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 }

```

Figura 10 – Interface dos serviços automatizados.

Fonte – Autoria Própria

```

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

```

Figura 11 – Declaração da servlet do WebService.

Fonte – Autoria Própria

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 12:

```

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>

```

Figura 12 – Declaração do *EndPoint* dos serviços.

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

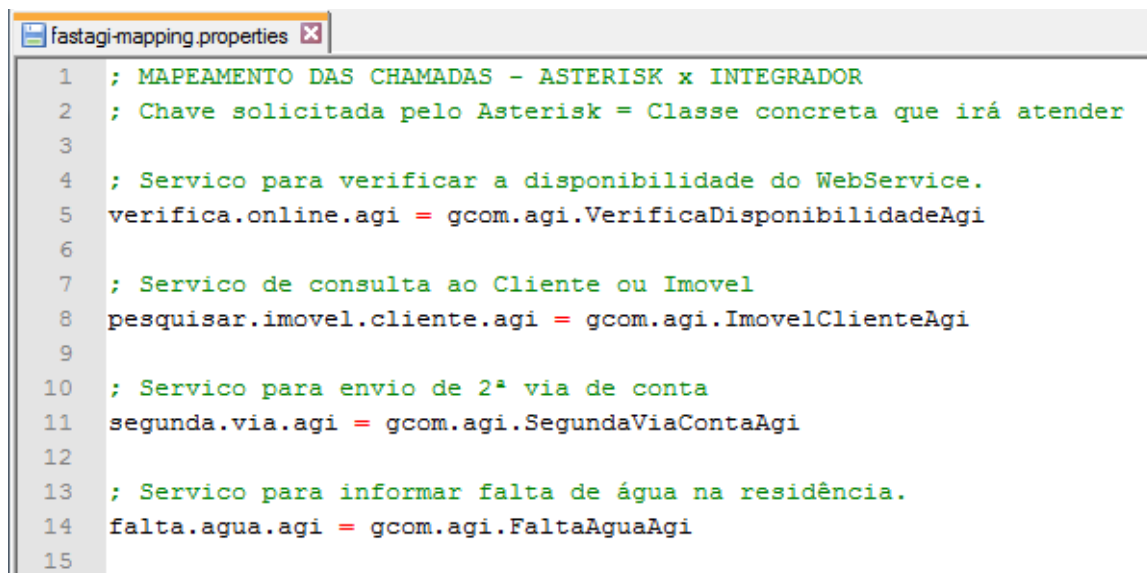
realizado da seguinte maneira:

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

Estando apto a ser integrado com sistemas externos.

3.3.2 Desenvolver *Middleware*

O *Middleware* intermediário denominado Integrador será responsável em receber as solicitações realizadas pela URA do Asterisk, após analisar deverá efetuar requisições ao *WebService* do sistema GSAN, que deverá retornar as informações esperadas, até então devolver as informações essenciais ao Asterisk. No cenário da integração proposta o sistema Integrador, deverá ser capaz de comunicar-se com o sistema Asterisk via protocolo AGI, para tanto foi utilizado um projeto *Open Source* chamada Asterisk-Java, que fornece bibliotecas para comunicação com o Asterisk utilizando os protocolos comuns ao sistema, no entanto será preciso mapear as chamadas que serão realizadas. Para cada novo serviço a ser disponibilizado, será necessário declarar uma chave de texto que será utilizada no Asterisk, representando o objeto a ser invocado, adicionados em um novo arquivo chamado *fastagi-mapping.properties* da seguinte maneira, conforme a figura 13 abaixo:



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

Figura 13 – Mapeamento dos serviços para consumo via Agi

Fonte – Autoria Própria

Feito isso o integrador já possuirá condições de recepcionar as solicitações originadas pelo *software* Asterisk utilizando o protocolo de comunicação AGI, e direcionar a chamada internamente para uma classe concreta que implemente o protocolo de comunicação utilizado, assim como a lógica de consumo aos novos serviços disponibilizados pelo *WebService* do sistema GSAN.

Para cada classe concreta a ser disponibilizada será necessário estender a classe *BaseAgiScript* disponibilizada pelo *framework* Asterisk-Java, após estender o comportamento será necessário implementar o método *service* obtido pela herança da *BaseAgiScript*, tal método será invocado quando a requisição for direcionada ao serviço mapeado, segue abaixo um exemplo de implementação utilizado, conforme algoritmo 1 abaixo:

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

Entrada: O primeiro parâmetro disponibiliza detalhes sobre a atual requisição, já o segundo contém informações e recursos para manipular o canal. (requisição, canal)

1. *atenderLigação()* ▷ *recepçionaaligação*
2. *digitoInformado* \leftarrow *canal[cliente]* ▷ *obtémdígitosinformadospelooriginador*
3. **se** *valorInformado* == *nulo* **então**
4. *tocarAudio(informar_valor)* ▷ *tocaráaudiodeaviso*
5. *canal[situacao]* \leftarrow *erro* ▷ *sinalizaparaocanalquehouveerro*
6. **senão**
7. *ws* \leftarrow *obterWebService()* ▷ *obténumainstânciadowebsevicecliente*
8. *retorno* \leftarrow *ws.pesquisarImovelOuCliente(valorInformado)* ▷ *fazarequisiçãopara*
9. *tratarRetorno(retorno)* ▷ *fazotratamentodoretornoobtido*
10. **fim**

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

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.

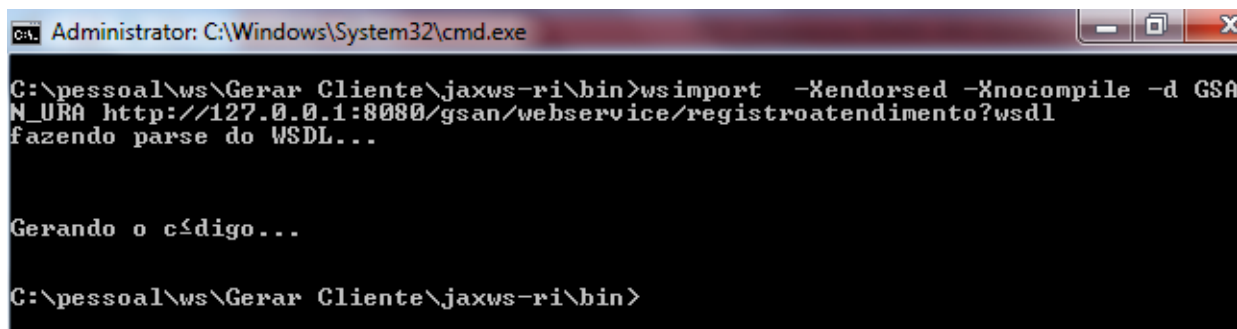


Figura 14 – Geração do código fonte para consumo do WebService.

Fonte – Autoria Própria

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 sistema Integrador deve ser disponibilizado em um terminal, que seja visível ao Asterisk, para que o mesmo consiga realizar requisições aos objetos mapeados anteriormente, neste trabalho o Integrador será executado no mesmo hospedeiro do Asterisk, conforme visto na figura 15 abaixo:

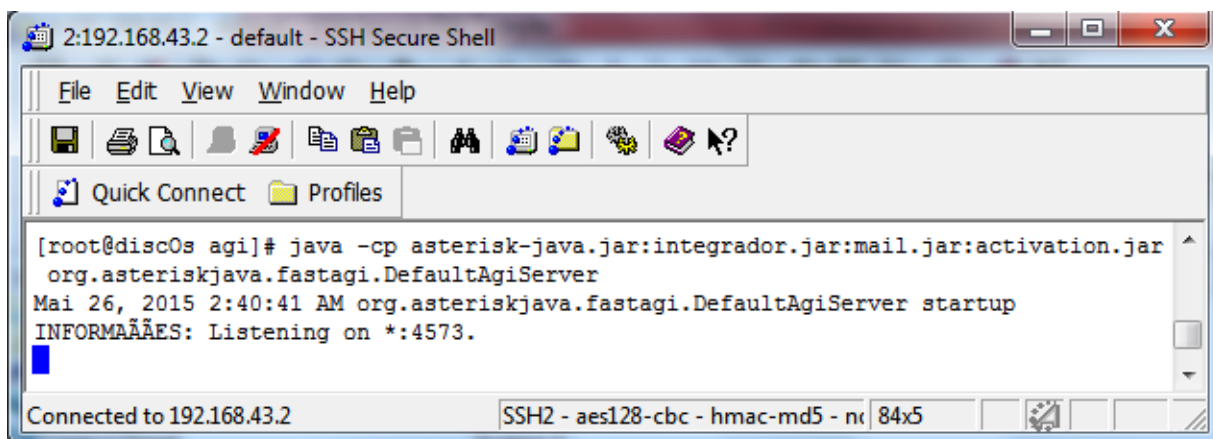


Figura 15 – Executando o sistema Integrador.

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 JVM (*Java Virtual Machine*) da plataforma Java, passando como *classpath* as seguintes bibliotecas:

- ***asterisk-java.jar***: biblioteca do framework utilizado para se comunicar com o Asterisk.
- ***integrador.jar***: o sistema integrador que integra as aplicações.

- **mail.jar** e **activation.jar**: ambos necessários no envio de email, para os casos de 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.

3.3.3 Customizar o 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 iremos configurar um ramal, utilizando a interface web do Disc-os, conforme exemplo abaixo na figura 16:

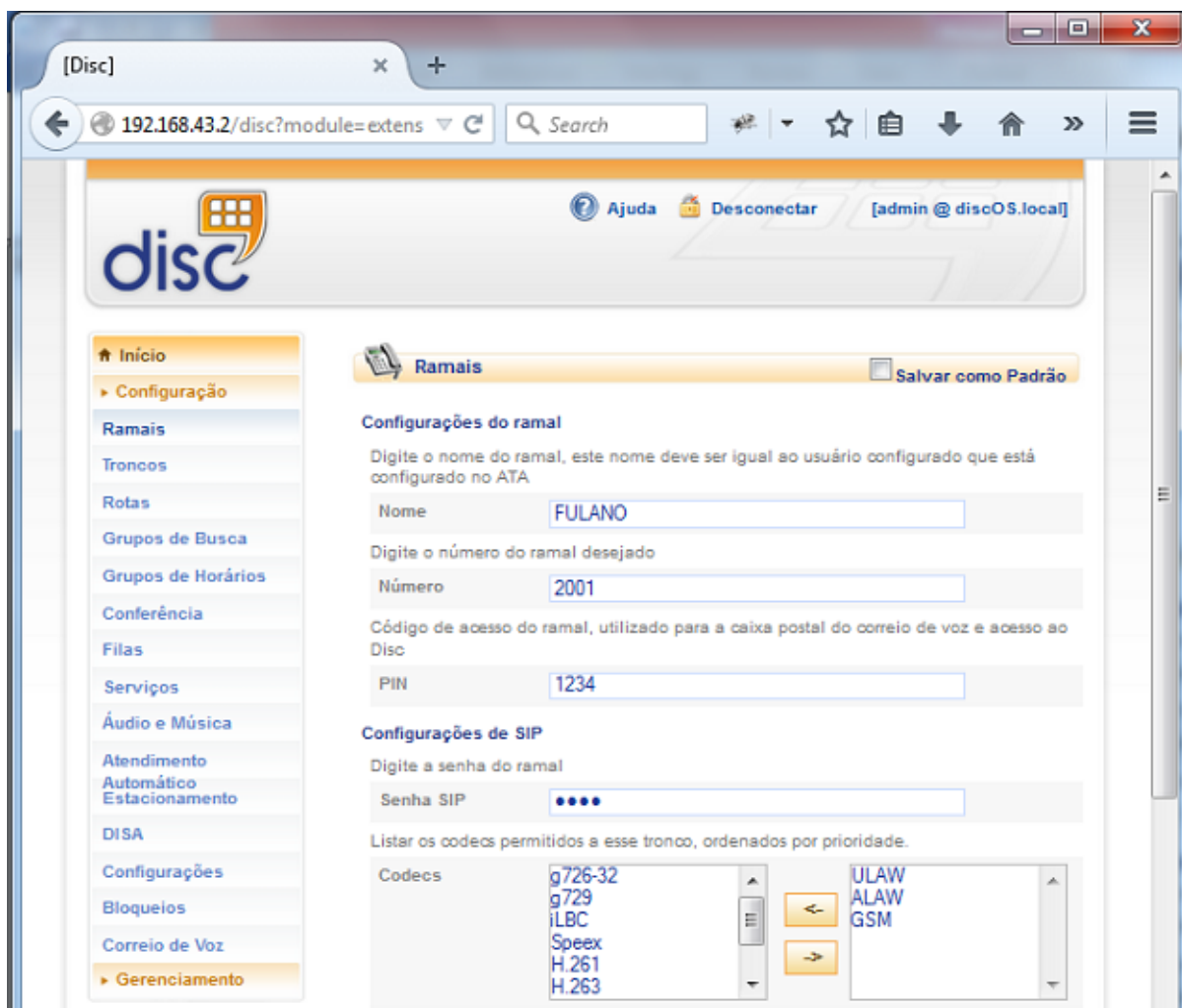


Figura 16 – Cadastro de um Ramal SIP.

Fonte – Autoria Própria

Conforme visto acima, 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 o seus devidos tratamentos, neste trabalho foi desenvolvido um fluxo de atendimento próprio, demonstrado na figura 17:

Dessa forma o cliente terá a opção de falar com o atendente logo no primeiro menu de atendimento e nos demais sub-fluxos, comportamento exigido por lei (BRASIL. Lei nº 8.078, Art.4), 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 abaixo a configuração utilizada para criação do contexto de PESQUISAR_CLIENTE utilizada para identificar o cliente, descrito no arquivo */etc/asterisk/*, conforme representado pelo algoritmo 2 abaixo:

Algoritmo 2 – Fluxo de identificação do cliente (Asterisk).

Entrada: Recebe a chamada atual com prioridade um.

1. *atenderLigação()* ▷ *recepçionaaligação*
2. *tocarAudio(beep)* ▷ *tocaráaudiodeaviso*
3. *tempo_limite_discagem* \leftarrow 3 ▷ *definitempolimitedeesperaentredígitos*
4. *tempo_limite_resposta* \leftarrow 7 ▷ *definitempolimitedeesperadoprimero digito*
5. *digitos* \leftarrow *lerDigitos()* ▷ *lerosdígitosinformados*
6. *canal[cliente]* \leftarrow *dígitos* ▷ *adicionaosdígitosnocanal*
7. *Agi(pesquisar.imovel.cliente.agi)* ▷ *fazachamadaagi*
8. **se** *canal[situacao]* == *sucesso* **então**
9. *gotoSucesso()* ▷ *redirecionaaligaçãoparasucesso*
10. **senão**
11. *gotoContextoAtendente()* ▷ *redirecionaaligaçãoparafalarcomatendente*

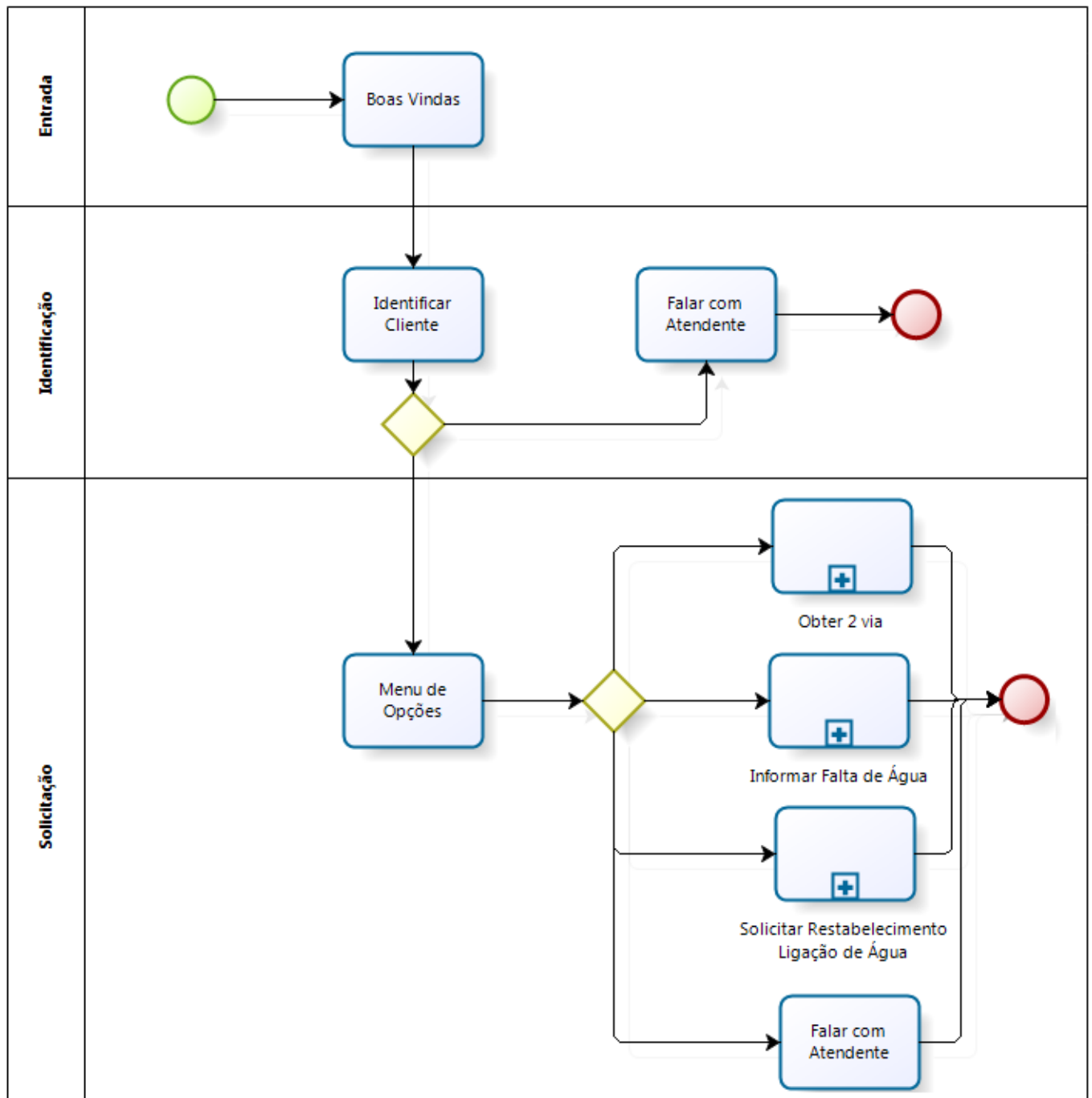


Figura 17 – Diagrama do Fluxo da Unidade de Resposta Audível

Fonte – Autoria Própria

12. fim

Neste trecho é declarado o contexto [PESQUISAR_CLIENTE], cuja prioridade 1 será anteder a ligação e logo em seguida emitir um áudio chamado beep, foi defini o tempo limite de espera para o primeiro dígito de 7 segundos e o tempo limite de espera para a discagem dos dígitos de 3 segundos, será lido os valores informados pelo Originador

da chamada e atribuída a variável `CLIENTE_IMOVEL`, após isto será realizada uma requisição via protocolo AGI para o *Middleware*, solicitando o serviço mapeando em `pesquisar.imovel.cliente.agi`, dependendo do retorno deste serviço será realizado um redirecionamento da ligação para um contexto específico, para consultar a configuração completa da URA encontra-se disponível nos anexos deste trabalho. Para tanto, será necessário customizar os arquivos de áudio para serem tocados, conforme as operações disponíveis forem acessadas. Para efetuar a gravação do áudio foi utilizado um recurso do Asterisk, discando para `*77`, existe um contexto que utiliza aplicativos nativos para realiza a gravação no formato WAV e o disponibiliza no diretório `/var/lib/asterisk/sounds/custom/` para utilização. Após as configurações básicas, será 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 Zoiper, conforme visto na figura 18:

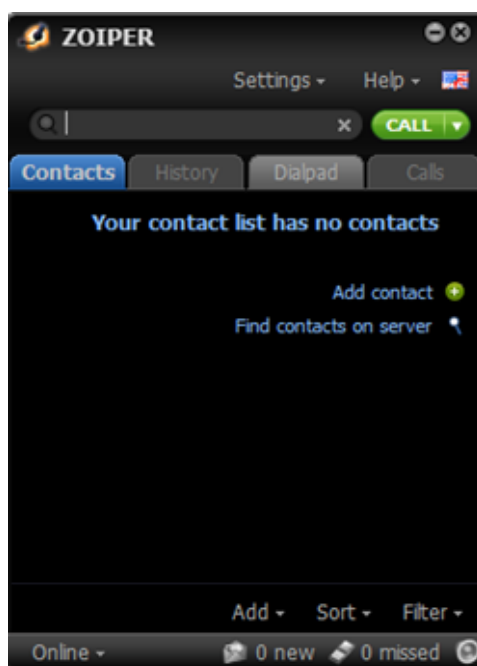
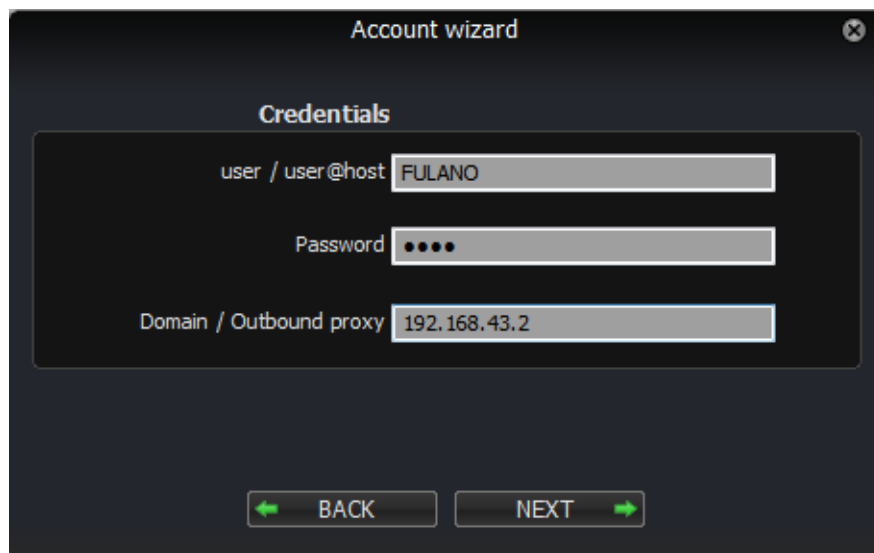


Figura 18 – Softphone Zoiper sendo executado

Fonte – Autoria Própria

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:

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



The image shows a dark-themed window titled "Account wizard" with a close button in the top right corner. Below the title is a section labeled "Credentials". Inside this section, there are three input fields: the first is labeled "user / user@host" and contains the text "FULANO"; the second is labeled "Password" and contains five black dots; the third is labeled "Domain / Outbound proxy" and contains the IP address "192.168.43.2". At the bottom of the window, there are two buttons: "BACK" with a green left-pointing arrow on its left, and "NEXT" with a green right-pointing arrow on its right.

Figura 19 – Configurar Ramal no Zoiper

Fonte – Autoria Própria

4 Resultados Alcançados

Será apresentado neste capítulo a execução dos experimentos sob a solução abordada nas seções anteriores, e exposto os resultados que foram alcançados.

4.1 Hardware da implantação

Na realização da implantação da solução proposta, foram utilizados os seguintes recursos de hardware:

Nº	SISTEMA OPER.	PROCESSADOR	MEMÓRIA	ARMAZENAMENTO
1	CentOS 2.6 64 bits	Intel Core i7	6GB	1TB
2	Windows 7 64 bits	Intel Core i5-3210M CPU@2.50GHz	8GB	300GB
3	Kali Linux 3.14 64 bits	Intel Core i7	2GB	250GB

Tabela 4 : Tabela de recursos utilizados.

Fonte: Autoria Própria

O recurso de número 1 é responsável em executar o software Asterisk paralelamente com Middleware desenvolvido, tornando o Call Center disponível a chamadas via VoIP.

O recurso de número 2 é responsável em executar o sistema GSAN através do servidor de aplicação Jboss 4.1, juntamente com o banco de dados Postgres.

O recurso de número 3 é responsável em simular chamadas VoIP para o Call Center (Asterisk).

Ambos os recursos estão conectados a uma rede local distribuída por um switch DLink 501, dessa forma cada recurso pode ser acessível na rede local. A rede local permite conexões de até 300 Mb/s conforme a especificação do Switch utilizado.

4.2 Estabelecer Métricas

De forma geral as companhias de saneamento atendem diariamente um acentuado volume de reclamações e solicitações de serviços a serem prestados, sejam para execução de serviços internos como serviços a serem prestados fora do ambiente da empresa, que normalmente exige um maior tempo para solucionar o atendimento. Neste trabalho foi definida a média de atendimento dos principais tipos de serviços disponíveis no sistema GSAN específicos para o módulo Atendimento ao Público que são registrados

diariamente, com base em informações obtidas através de relatórios regulatórios e entrevistas com funcionários da Companhia de Saneamento Manaus Ambiental, acerca dos atendimentos prestados nas Centrais de Atendimentos das empresas de saneamento, com base em análise das reclamações por tipos de serviço, foi construído o seguinte gráfico, conforme demonstrado na figura 20 abaixo:

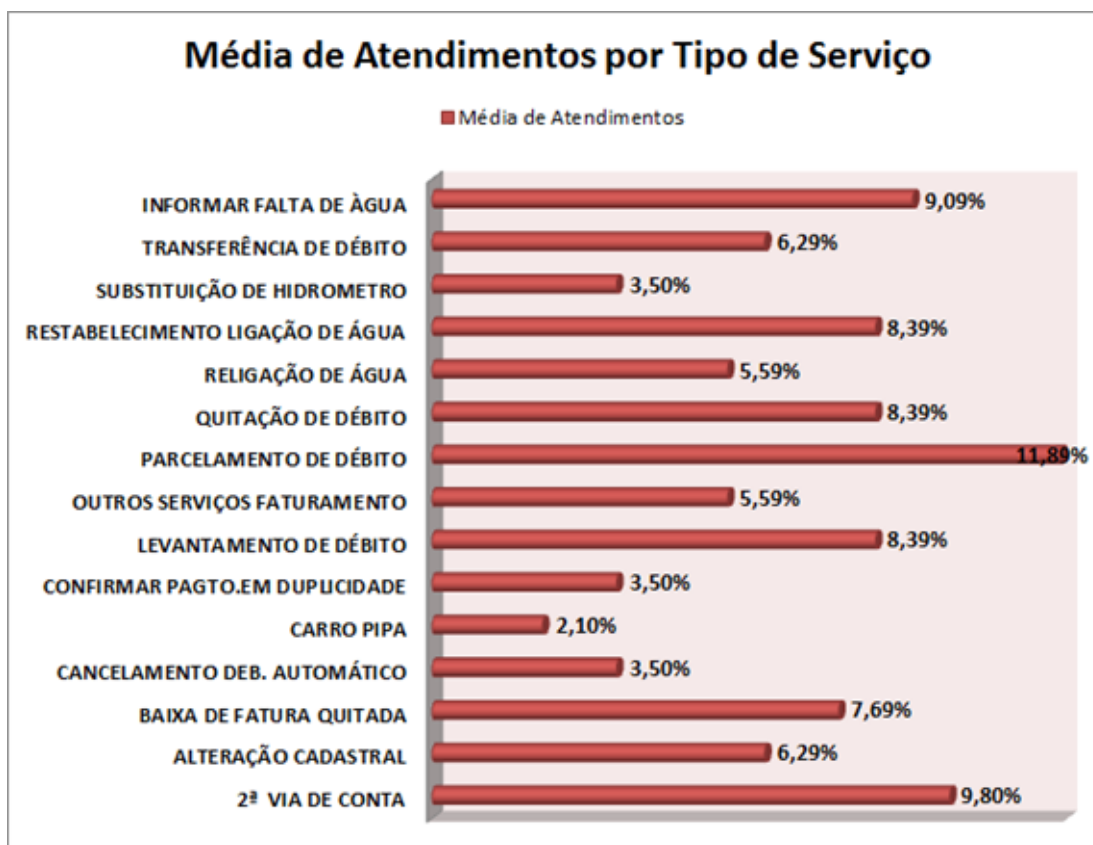


Figura 20 – Média de atendimentos diários em companhias de saneamento

Fonte – Autoria Própria

Com base na média de atendimentos realizados por tipos de serviços exposto acima, notamos que os serviços automatizados propostos neste trabalho representam o total de 27,28% dos atendimentos prestados diariamente nas centrais de atendimentos das companhias de saneamento, no entanto para que seja possível atender plenamente essa parcela dos atendimentos, foram propostos cenários de teste para averiguar o quão eficiente a solução está em solucionar os cenários.

4.3 Cenários de Testes

Os cenários de testes representam as possíveis situações cadastrais ou comportamentais apresentadas pelo cliente ao longo do atendimento, onde cada cenário deve ser totalmente atendimento para que se tenha êxito no atendimento e posteriormente a situação de **ATENDINDO**, caso contrário será obtido à situação de **NÃO ATENDINDO**,

caso o atendimento não seja concluído com sucesso o cenário deve conter uma observação, descrevendo o motivo ao qual levou a tal situação. Abaixo serão descritos os Cenários dos serviços propostos:

4.3.1 Obter 2ª Via de Conta

Abaixo serão descritos os casos de testes previstos para o serviço Obter 2ª Via de Conta;

CENÁRIO 1: Cliente devidamente cadastrado no Sistema, atualmente proprietário de um único imóvel que possui ativo a Ligação de Água e Esgoto, com pendência de duas contas. O Cliente deseja obter as duas contas em aberto.

SITUAÇÃO: ATENDENDO

OBSERVAÇÃO: Não aplicável

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 quatro contas em aberto. O Cliente deseja obter as quatro contas em aberto.

SITUAÇÃO: ATENDENDO

OBSERVAÇÃO: Não aplicável

CENÁRIO 3: Cliente devidamente cadastrado no Sistema, atualmente sendo responsável por três imóveis que possuem ativo a Ligação de Água e Esgoto, possuindo uma conta pendente para cada imóvel. O Cliente pretende obter a conta pendente de somente um imóvel específico.

SITUAÇÃO: ATENDENDO

OBSERVAÇÃO: Não aplicável

CENÁRIO 4: Cliente sem e-mail cadastrado no Sistema, atualmente sendo responsável por dois imóveis que possuem ativo a Ligação de Água e Esgoto, possuindo duas contas pendente para cada imóvel. O cliente pretende obter todas as contas em aberto dos dois imóveis.

SITUAÇÃO: NÃO ATENDENDO

OBSERVAÇÃO: Devido a falta do e-mail no cadastro do cliente não será possível enviar as contas pendentes ao mesmo, e também a solução inicialmente está tratando dos débitos de um imóvel por vez, sendo necessário redirecionar a ligação para o atendente concluir o atendimento, ou repetir a operação.

4.3.2 Informar Falta de Água

Abaixo serão descritos os casos de testes previstos para o serviço Informar Falta de Água;

CENÁRIO 1: Cliente devidamente cadastrado no Sistema, atualmente proprietário de um único 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 único imóvel.

SITUAÇÃO: ATENDINDO

OBSERVAÇÃO: Não aplicável

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.

SITUAÇÃO: ATENDINDO

OBSERVAÇÃO: Não aplicável

CENÁRIO 3: Cliente devidamente cadastrado no Sistema, atualmente responsável por três imóveis que possuem ativo a Ligação de Água, com uma conta pendente para cada imóvel. O Cliente deseja informar problemas no abastecimento de água para um imóvel específico ao qual o mesmo é responsável.

SITUAÇÃO: ATENDINDO

OBSERVAÇÃO: Não aplicável

CENÁRIO 4: Cliente devidamente cadastrado no Sistema, atualmente responsável por dois imóveis em locais diferentes, que possuem a Ligação de Água ativa, em recente situação de inadimplência. O Cliente deseja informar problemas no abastecimento de água para os dois imóveis ao qual o mesmo é responsável.

SITUAÇÃO: NÃO ATENDINDO

OBSERVAÇÃO: A solução inicialmente para Informar Falta de Água está tratando apenas um imóvel por vez, sendo necessário redirecionar a ligação para o atendente concluir o atendimento, ou repetir a operação.

4.3.3 Solicitar Restabelecimento da Ligação de Água

Abaixo serão 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.

SITUAÇÃO: ATENDENDO

OBSERVAÇÃO: Não aplicável

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.

SITUAÇÃO: ATENDENDO

OBSERVAÇÃO: Não aplicável

CENÁRIO 3: Cliente devidamente cadastrado no Sistema, atualmente responsável por dois imóveis, que possuem a Ligação de Água interrompida por corte, devido a falta de pagamento das dez contas em aberto para ambos. O Cliente deseja solicitar restabelecimento da ligação para os imóveis após realizar a negociação das faturas.

SITUAÇÃO: NÃO ATENDENDO

OBSERVAÇÃO: A solução inicialmente está tratando o restabelecimento da ligação de apenas um imóvel por vez, sendo necessário redirecionar a ligação para o atendente concluir o atendimento, ou repetir a operação.

CENÁRIO 4: Cliente com cadastro incompleto 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.

SITUAÇÃO: ATENDINDO

OBSERVAÇÃO: Não aplicável

4.4 Resultado dos Cenários de Teste

Após a apuração dos cenários descritos acima, chegamos aos seguintes resultados por serviço:

TABELAS PENDENTES :D

4.5 Apresentação dos Resultados Obtidos

Levando em consideração os resultados obtidos através da aplicação de todos os cenários de testes descritos acima sob os serviços desenvolvidos neste trabalho, observando os percentuais atingidos com base na média descrita dos principais Serviços do Atendimento ao Público, notamos que a solução proposta é capaz de reduzir em até 20,46%, dos registros de atendimentos diários, conforme visto abaixo na figura [21](#):

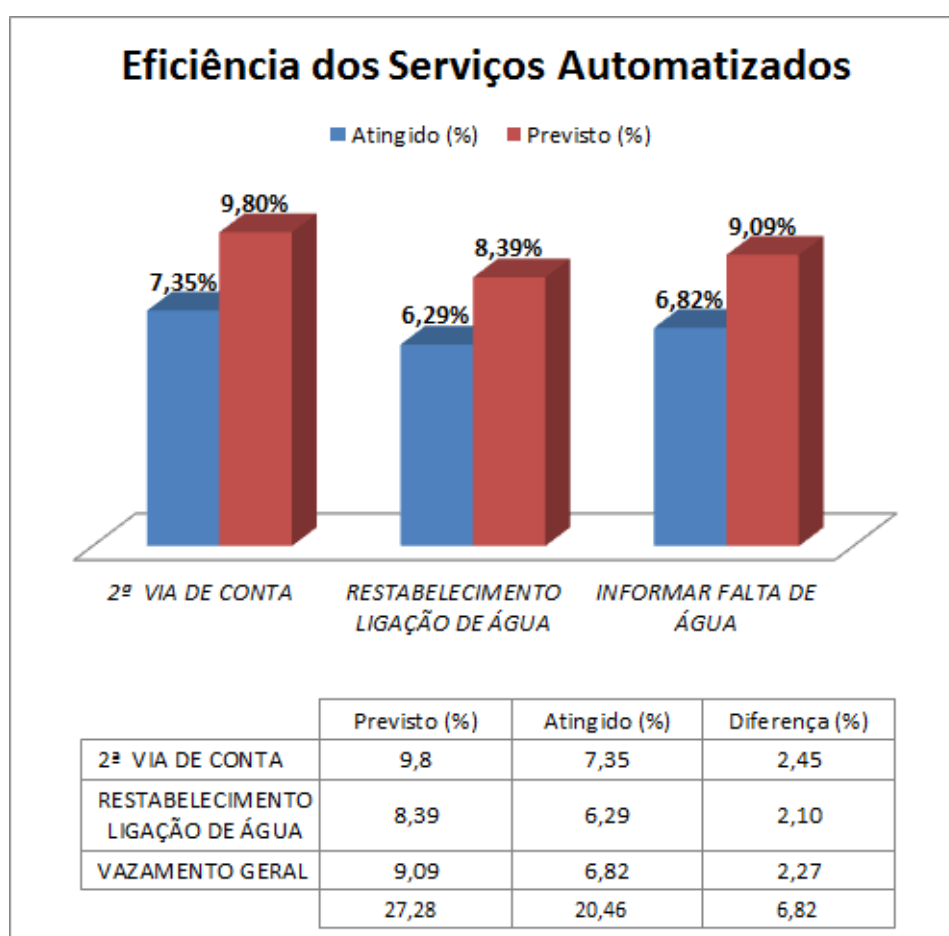


Figura 21 – Gráfico de avaliação dos serviços automatizados

Fonte – Autoria Própria

5 Considerações Finais e Trabalhos Futuros

A solução em *software* apresentada e desenvolvida se trata de uma proposta inicial que pode se tornar um módulo interno com grandes potenciais para integrações futuras, neste trabalho foi adotada a integração do sistema GSAN com uma central de atendimento automatizada através do *software* Asterisk, os novos serviços desenvolvidos possibilitam a reutilização para quaisquer outras soluções do gênero, o código fonte será disponibilizado na comunidade do GSAN, para que seja mantido e evoluído ao longo do tempo. Existem alguns recursos ou rotinas que podem ser criados para melhor aproveitamento e aprimoramento 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.
- Utilizar o Interceptador gerando estatísticas dos acessos realizados, coletando tempo de respostas e traçando perfis de cliente que mais realizam solicitações via *Call Center*.

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