

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
SISTEMAS DE INFORMAÇÃO**

**SISTEMA DE CONTROLE DE SUPORTE TÉCNICO
UTILIZANDO WEB SERVICE NA IMPRENSA OFICIAL DO
ESTADO DE SANTA CATARINA**

FÁBIO DE PAULA VIEIRA

FLORIANÓPOLIS, 2004

FÁBIO DE PAULA VIEIRA

SISTEMA DE CONTROLE DE SUPORTE TÉCNICO UTILIZANDO WEB
SERVICE NA IMPRENSA OFICIAL DO ESTADO DE SANTA
CATARINA

Monografia apresenta como Trabalho de Conclusão de Curso -TCC para obtenção do grau de Bacharel em Sistemas de Informação, no Centro Tecnológico - CTC.

Orientador: Prof. Dr. José Eduardo De Lucca

Co-Orientador: João Batista Bittencourt

Florianópolis, novembro de 2004

DEDICATÓRIA

Dedico este trabalho aos meus pais que, através do exemplo, me ensinaram a honrar nossos princípios e a lutar sempre.

AGRADECIMENTOS

Agradeço sinceramente a todos que me ajudaram neste trabalho, amigos que não mediram esforços para que esta etapa fosse vencida.

Agradeço, em especial, às pessoas mais importantes da minha vida: minha mãe, meu pai, minha irmã e minha namorada.

Agradeço, também, à minha pátria que pôde, apesar de suas dificuldades, me proporcionar um curso superior de alta qualidade.

RESUMO

Usuários de microcomputadores estão sujeitos a encontrar problemas na utilização dos equipamentos; isto é inerente ao uso de máquinas. Quando se trata de empresas, muitas vezes estes problemas permanecem pendentes por muito tempo até que o responsável tome conhecimento e providencie recursos para saná-lo, acarretando prejuízo no rendimento e na produção. Levando em conta a realidade do ambiente que será utilizado como objeto de análise - Imprensa Oficial do Estado de Santa Catarina - e suas características bem próprias, um sistema de controle de chamados técnicos de reparo é desenvolvido a fim de suprir as necessidades de maneira distribuída.

PALAVRAS-CHAVE: XML, Java, Web Service, Help Desk

SUMÁRIO

1 – Introdução.....	6
1.1 - Estrutura do Trabalho	7
2 – Caracterização do problema	8
2.1 - Ambiente	8
2.2 - Infra-estrutura	8
2.3 - Manutenção de equipamentos	11
3 – Estudos Realizados	15
3.1 - Estado da Arte	15
3.1.2 - Sistemas analisados	15
3.1.3 Considerações	25
3.2 – Tecnologias alternativas	25
3.2.1 - Arquiteturas baseadas em RPC	26
3.2.2 - Arquiteturas transacionais tipo HTTP	27
3.2.3 - Web Service	29
4 – Proposta de solução	45
4.1 - Processo de manutenção informatizado	45
4.2 - Uso de Web Service.....	51
5. Desenvolvimento do Sistema	53
5.1 Fase Planejar e Elaborar	53
5.1.1 Definição dos requisitos.....	54
5.1.2 Definição dos casos de uso	56
5.2 Fase Construir	66
5.2.1 Análise	66
5.2.2 Projeto	79
5.3 - Ferramentas utilizadas.....	90
5.4 Módulos implantados.....	92
5.5 Telas.....	93
Conclusão	97
Trabalhos futuros	98
Bibliografia	99
ANEXOS	100

1 INTRODUÇÃO

Devido ao avanço tecnológico e científico o panorama do trabalho sofreu significativas alterações em diversos contextos. Mesmo em atividades tradicionais como a agricultura, que outrora era rude e primitiva, é possível ter uma amostra do que vêm acontecendo. Antes, para produzir-se uma lavoura, necessitava-se de inúmeras pessoas para realizar o plantio. Hoje é fácil perceber que o campo está bastante equipado com máquinas capazes de plantar com um número mínimo de trabalhadores. É um exemplo de como as máquinas estão se tornando comum no trabalho.

Assim como na agricultura, existe hoje em todos os ramos de atividades e processos de produção uma vasta e crescente utilização de equipamentos de automação. Isto ocorre, basicamente, pela diminuição gradativa do custo dos equipamentos causados pelo aumento de suas produções.

O computador talvez seja o mais difundido equipamento de automação existente atualmente. Sua utilização é muito variada e pode ser encontrado tanto em pequenos estabelecimentos quanto em grandes centros de processamentos de dados.

Não é difícil crer que o próprio computador que surgiu para solucionar diversos problemas também cria outros tantos problemas. Ele está exposto, via de regra, aos mais diferentes usuários com os mais diversos estágios de conhecimento na área. Sendo assim, alguns problemas podem ser realmente problemas e outros problemas podem ser simplesmente mau uso pelo limitado conhecimento. Porém, no ponto de vista do suporte técnico, ambos serão problemas.

Empresas quase sempre contam com alguma forma de suporte técnico para resolver os problemas que lhes acontecem quase todos os dias. Se este suporte, quando dentro da própria empresa, trabalhar desprovido de algum método gerenciado, tende a ser confuso, ineficiente e não provedor de histórico dos chamados de reparo.

Como uma maneira de sistematizar este processo de manutenção e preservar o investimento em máquinas, um sistema de gerenciamento de manutenção se faz preciso.

Motivado pela crescente utilização da Internet aliada a programação distribuída, um sistema de manutenção nestes moldes se torna atrativo para empresas ou entidades que possuem unidades de atuação fisicamente separadas.

O Web Service é uma alternativa para ocupar este espaço. Ele é um modelo de computação distribuída recente que funciona fornecendo, basicamente, serviços remotos para serem utilizados por diferentes programas escritos para diferentes plataformas. Um servidor, conhecido por seu endereço, fornece serviços que estão especificados e publicados para algum cliente interessado em utilizá-lo. Desta forma, promovendo a interoperabilidade, a comunicação no sistema de manutenção pode ser estabelecida passando por cima das barreiras geográficas, de sistema operacional, de linguagem de programação e de protocolo de comunicação.

O sistema proposto por este trabalho gerencia (atribui tarefas para as pessoas responsáveis) e acompanha solicitações de serviço de manutenção até sua conclusão na Imprensa Oficial do Estado de Santa Catarina. O sistema também mantém os relatórios dos atendimentos, com as soluções aplicadas, para que possam vir a ser utilizados no futuro como uma base de conhecimento para soluções mais ágeis.

1.1 Estrutura do Trabalho

Este trabalho está dividido em seções que representam os pilares essenciais do desenvolvimento, partindo da idéia inicial e indo até a solução prática do problema. Depois da introdução do assunto há o segundo capítulo que trata do problema em questão e o ambiente de estudo. Em seguida, no terceiro capítulo são relatados os estudos realizados para o desenvolvimento do trabalho, tais como o estado da arte e as tecnologias relacionadas. O quarto capítulo aborda uma proposta para solução do problema e o quinto capítulo é referente ao desenvolvimento do sistema. Por fim existem as considerações finais.

2 CARACTERIZAÇÃO DO PROBLEMA

2.1 Ambiente

A Imprensa Oficial do Estado de Santa Catarina – IOESC – é uma empresa pública vinculada à Secretaria de Estado da Administração, que tem como objetivo divulgar os atos oficiais do Estado de Santa Catarina, produzindo obras gráficas, nas diversas formas, veiculando o Diário Oficial, o Diário da Justiça, materiais de campanha do governo e diversas outras publicações de interesse do Estado de Santa Catarina. Seu parque industrial, sua administração e a sua infra-estrutura de informática estão localizados na matriz, situada no bairro Saco dos Limões em Florianópolis, Capital do Estado.

A IOESC conta com uma unidade comercial no centro da cidade para facilitar o acesso de seus serviços aos cidadãos. Nesta unidade, chamada agência, é possível comprar edições dos jornais publicados, fazer assinaturas, fazer publicações, comprar livros, entre outras coisas.

A conexão à Internet é estabelecida pelo Centro de Informática e Automação do Estado de Santa Catarina – CIASC –, através de um anel de fibra ótica que interliga o próprio CIASC, a IOESC, a agência e diversas outras autarquias e entidades do Governo do Estado situadas na Capital.

Parte do conteúdo produzido pela IOESC é disponibilizada às pessoas através da Internet pelo Diário Oficial on-line que, por enquanto, contém apenas o Diário da Justiça, pois o projeto de implantação do Diário Oficial ainda não foi concluído.

2.2 Infra-estrutura

O ambiente de tecnologia de informação conta com os seguintes equipamentos:

- a. dois servidores IBM xSeries 232 com 512 megabytes de memória ram, dois processadores de 1.12 gigahertz e sistema operacional Windows 2000

- Advanced Server. Um deles é servidor Web, servidor de E-mail, DNS e domínio; o outro é servidor de banco de dados;
- b. dois servidores HP NetServ E 60 com 512 Megabytes de memória ram, dois processadores de 500 Megahertz e sistema operacional Windows 2000 Advanced Server. Um deles é servidor de aplicações internas (faturamento, estoque, etc.); o outro é servidor de antivírus;
 - c. firewall com configuração não fornecida;
 - d. uma Switch 3Com 2500;
 - e. uma Switch Micronet SP1659;
 - f. uma Switch Micronet SP 624K;
 - g. seis Hub's D-link DE 1824i;
 - h. quarenta estações de trabalho.

Nas figuras 2.1 e 2.2 as redes da sede e da agência respectivamente estão esboçadas.

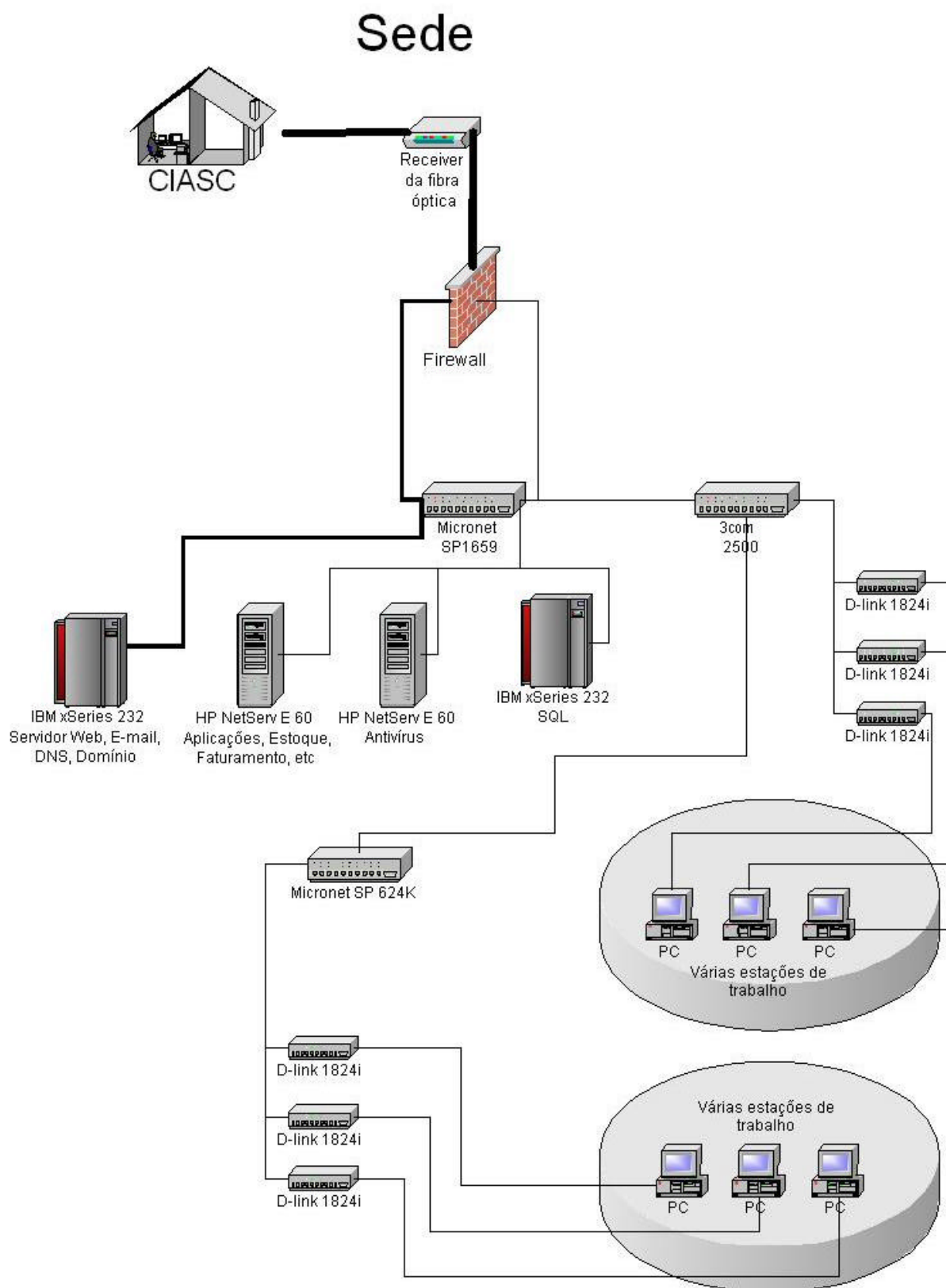


Figura 2.1 Esboço da rede da sede na sede

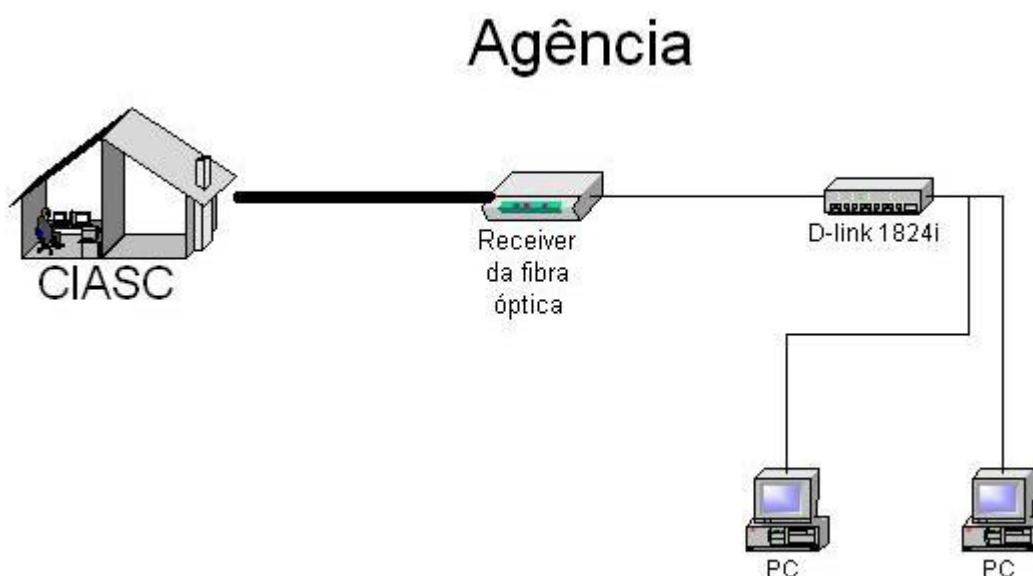


Figura 2.2 Esboço da rede da agência

2.3 Manutenção de equipamentos

A Imprensa Oficial do Estado de Santa Catarina conta, em sua estrutura, com um setor responsável pela área tecnológica da instituição: a SUATEC, sigla de Supervisão de Atualização Tecnológica. Este setor é encarregado de nortear e manter os recursos de automação e informação necessários para o andamento das atividades no âmbito da empresa. Ela está vinculada à Gerência de Planejamento da Produção - GEPLA.

Entre outras incumbências, esta supervisão é responsável pela realização da manutenção dos equipamentos de informática e outros equipamentos relacionados. Tais equipamentos são: microcomputadores, notebooks, servidores de rede, impressoras, scanners, estabilizadores, servidores de impressão, hubs, roteadores, aparelhos de fax, no-breaks, câmeras fotográficas digitais e zip drives.

O processo de manutenção de um equipamento começa quando um funcionário da empresa percebe algum problema no equipamento. Em seguida, o funcionário entra em contato com algum membro da SUATEC através do telefone ou por contato pessoal e relata os acontecimentos, quase sempre solicitando uma visita. O técnico do setor desloca-se até o local do equipamento, analisa os fatos e o

equipamento, tentando várias providências para solucionar o problema. Tendo solucionado, o episódio acaba. Porém, se esgotados todos os recursos e o problema persistir, o equipamento deve ser encaminhado à assistência técnica. Contudo, os pedidos de assistência técnica devem ser feitos mediante algumas normas do regimento interno da empresa.

Este processo que envolve a assistência técnica começa como um atendimento convencional citado anteriormente. A distinção ocorre quando o técnico da IOESC entende que não tem condições de resolver aquele problema e informa esta condição ao supervisor da SUATEC que determinará o destino daquele chamado. O processo continua através de um procedimento em que convites são enviados a, no mínimo, três empresas que disputarão uma concorrência de menor preço. Cada uma delas apresenta seu orçamento e a empresa que oferecer os serviços pelo menor preço ganha o direito de executar este trabalho. Depois de consertado pela assistência técnica externa, o equipamento retorna ao seu local original e o processo acaba.

É relevante ressaltar que a grande maioria dos chamados de assistência técnica ocorre por circunstâncias simples. São raros os problemas que exigem assistência especializada conforme mencionado. Porém eles existem.

A SUATEC é provida de três tipos de cargos: o supervisor – aquele que dirige e orienta o setor –, o técnico e o auxiliar técnico. Todos os cargos são capacitados para a execução de operações de manutenção, porém qualquer pedido de assistência técnica externo deve ser feito por intermédio do supervisor que solicita o reparo ao gerente da Gerência de Planejamento da Produção. Se aprovado, o processo prossegue como descrito anteriormente. Uma representação do processo pode ser observada na figura 2.3.

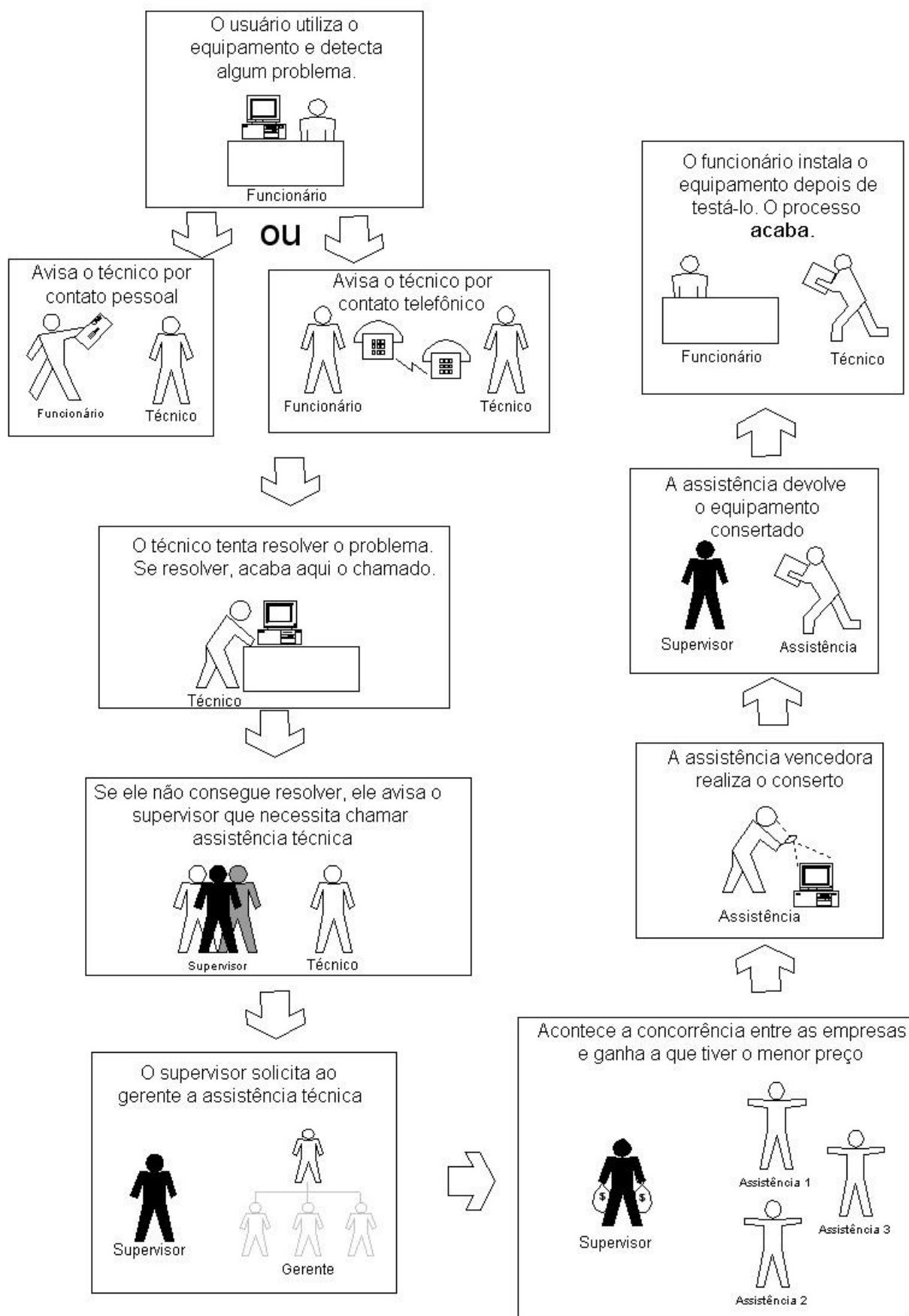


Figura 2.3 Representação do processo manual de abertura de chamado técnico.

A manutenção de um equipamento engloba tanto o hardware como o software. Abaixo estão listados os problemas mais comuns de dispositivos na empresa:

SOFTWARE	HARDWARE
<ul style="list-style-type: none"> ○ Problemas de usabilidade do sistema operacional em microcomputadores 	<ul style="list-style-type: none"> ○ Problemas de componentes físicos dos microcomputadores, tais como placas mães, placas de rede, placas de vídeo, placas de áudio, monitores de vídeo, teclado, mouse, disco rígido, unidades de discos removíveis entre outros.
<ul style="list-style-type: none"> ○ Problemas de usabilidades de programas de microcomputador, tais como Microsoft Office, programas de faturamento, estoque, emuladores, consoles, softwares de impressoras e scanners 	<ul style="list-style-type: none"> ○ Problemas de componentes periféricos, tais como cabos de rede, cabos elétricos, estabilizadores, no-breaks, impressoras e scanners
<ul style="list-style-type: none"> ○ Problemas de acesso ao e-mail, bem como a sua utilização 	<ul style="list-style-type: none"> ○ Problemas nos demais dispositivos de responsabilidade da SUATEC
<ul style="list-style-type: none"> ○ Problemas com microcomputadores infectados por vírus 	

Adiante, neste trabalho, é apresentada uma proposta de automação para este processo de manutenção observado, envolvendo as atividades descritas neste capítulo.

3 ESTUDOS REALIZADOS

3.1 Estado da Arte

A justificativa para a construção de um sistema de gerenciamento de chamados técnicos quando existem inúmeras soluções disponíveis para isto é quase uma questão de habilidade persuasiva. Naturalmente, existem vantagens de utilizar uma solução pronta, em que ela já foi suficientemente testada e, provavelmente, está bastante aprimorada e lapidada perante os percalços do uso. Por outro lado, a construção de um sistema capaz de gerenciar um chamado técnico também traz vantagens, pois o foco do desenvolvimento está totalmente voltado àquela determinada instituição que possui suas características próprias e específicas, proporcionando um sistema que atende a todas as necessidades existentes no ambiente sem acarretar mudanças bruscas de conduta na execução das tarefas diárias a que as pessoas envolvidas estão acostumadas.

Numa pesquisa realizada na Internet foi possível serem apontados alguns bons programas de “Help Desk”. Dentre os sistemas considerados, três deles foram selecionados com o objetivo de mostrar-se seu funcionamento e suas tecnologias agregadas.

3.1.2 Sistemas analisados

Existe uma vasta opção de sistemas do tipo “Help Desk” disponíveis pela Internet com a finalidade semelhante ao que se propõe neste trabalho. Cada qual dá um enfoque diferente à determinada característica do domínio do problema, mas a maioria cumpre o principal objetivo proposto em sistemas desta natureza: abrir e tratar um pedido de reparo.

Para uma amostra do que existe hoje em matéria de “Help Desk”, três sistemas foram selecionados para análise e, conseqüentemente, usados de inspiração. Nos momentos que precederam a escolha dos três softwares, foi possível notar muitas semelhanças, mas também muitas diferenças entre eles. Em meio a elas, podem-se destacar duas características marcantes. A primeira delas se

refere à simplicidade da interface e ao discernimento instantâneo das ferramentas; geralmente estes tipos de sistemas tendem a ser simples, mas ocorrem várias exceções. A segunda é referente à liberação do código: alguns são inteiramente livres para utilização e, no lado oposto, outros que cobram taxas para o seu uso. Duas características que pesam na maioria das escolhas.

Os três sistemas escolhidos foram o Web Help Desk, o SysAid e o ExDesk. A seguir a análise dos softwares.

Web Help Desk

O Web Help Desk é um sistema que integra as funcionalidades de “Help Desk” e de controle de patrimônio. Para utilizar o sistema o usuário precisa apenas de um acesso à Internet e de um navegador em seu computador, recurso presente na maioria dos sistemas operacionais mais populares. Esta característica mostra uma desejável preocupação com a independência de plataforma do lado cliente, que fica livre para utilizar a tecnologia desejada. Em aspectos gerais, possui uma interface amigável e com ferramentas de fácil acesso. Comparado a outros softwares, possui uma certa intuitividade nos comandos, principalmente aqueles situados do lado do usuário “menos experiente”, que é aquele que realiza os chamados de reparo. O sistema utiliza tecnologia Java, o que permite independência também de arquitetura do lado servidor, pois pode ser executado em qualquer servidor que tenha Java 2 SE instalado, como MacOS X Server 10.2 ou anteriores, Windows 2000 Server, Linux e Sun Solaris 8.

O usuário, ao fazer o login, tem acesso à sua interface e nela realiza as operações de abertura de chamado técnico. Estes chamados também podem ser feitos por e-mail.

O sistema considera uma série de fatores para designar um técnico a atender a um chamado. Algoritmos levantam informações a respeito do local de origem do chamado, o tipo de problema relatado e, com isto, determinam a melhor opção de técnico capaz de resolver aquele determinado problema registrado. Entre outras funcionalidades, destaca-se uma em que os usuários são mantidos informados pelo sistema através de e-mails sobre a situação do chamado aberto por eles, recebendo notificações à medida que os chamados sofrem alterações. Os técnicos são alertados sobre chamados que foram “esquecidos” por um certo período.

Em casos onde o cliente possui uma empresa com diversas filiais, o sistema disponibiliza meios de agrupar logicamente as unidades da corporação, fornecendo formas de controle centralizado e distinguido, quando for necessário, o tratamento de cada unidade.

O sistema de banco de dados é relacional, com suporte para OpenBase SQL 7, Microsoft SQL Server 2000, MySQL e Oracle 8i. Na base de dados é possível o armazenamento das informações de uma seção de “perguntas e respostas” mais freqüentes. Estas questões podem ser disponibilizadas aos usuários para que eles tentem resolver o problema por si próprios, caso deparem com uma pergunta relacionada com o seu problema .

Existem quatro tipos de usuários: cliente, contato admim, técnico e administrador.

Os usuários cliente e contato admim são muito parecidos em seus papéis no sistema: ambos podem abrir chamados, ver histórico de pedidos, ver as perguntas mais freqüentes e também fazer edição do perfil. O contato admin pode, além de tudo isto, ver o histórico de diferentes clientes e obter alguns tipos de relatórios específicos e autorizados a ele. Um exemplo de tela de abertura de chamado técnico na figura 3.1.

Web Help Desk™

Request History FAQs Profile Logout

Help Request

Problem Type: Hardware : Will not Boot

Subject: Ligar

Problem Detail: Não consigo ligar o computador

Attachments: Add File

Select Asset

Please select an asset from the "My Assets" list or choose a Model and Location below. You may also search for an asset using the asset number.

Asset No. 133 Search

Type: Desktop

Model: Dell GX260

Location: High School 2

Room: 100

Cancel Save

My Assets

No.	Model	Room
133	Apple PowerBook G4	100

1 item

Figura 3.1 Exemplo de abertura de chamado técnico

O cliente preenche os campos de descrição do problema e do equipamento para solicitar um atendimento.

O Técnico dispõe de uma interface mais ampla em opções. Como tela de partida, ele possui uma tabela de chamados abertos descrevendo rapidamente cada um destes chamados. Existe também um calendário de atividades que fornece uma visão geral das atividades a ele alocadas. Na figura 3.2 um exemplo de tela onde estão listados os chamados a ele designado.

Home Tickets Techs Clients Assets Parts Locations FAQs Reports											
My Tickets		My Calendar		Group Tickets		My Account					
New Ticket Download PDF Download TSV Priority: <input type="text"/> Status: <input type="text"/> Location: <input type="text"/> Max Display: <input type="text" value="100"/> Refresh: <input type="text" value="1 min"/>											
No.	Date	Updated	Status	Technician	Location	Department	Problem Type	Model	Asset No.	Client	
3819	09/13/04 06:30 PM	09/13/04 05:30 PM	Open	Tech	Zingeman's Dell	Social Security	IP Request			Client, Demo	
3877	09/08/04 10:39 PM	09/08/04 10:39 PM	Open	Tech	Chennai		Software Problem + Photoshop			Joy, Sujman	
3874	09/08/04 01:01 PM	09/08/04 01:01 PM	Open	Tech	Sydney - Co1	Social Security	Software Problem + Web Browser			Client, Demo	
3851	09/08/04 06:28 PM	09/08/04 06:28 PM	Open	Tech	High School 2	Social Security	IP Request			Client, Demo	
3783	09/08/04 03:07 PM	09/12/04 07:08 PM	Open	Tech	High School 2	Social Security	IP Request			Client, Demo	
3784	08/27/04 10:20 PM	08/27/04 10:20 PM	Open	Tech	Japan	Social Security	Software Problem + E-Mail			Client, Demo	
3782	08/27/04 03:34 PM	08/27/04 03:34 PM	Open	Tech	High School 2	Social Security	IP Request			Client, Demo	
3746	08/24/04 11:57 AM	08/28/04 11:41 PM	Open	Tech	High School 2	Social Security	IP Request			Client, Demo	
3742	08/24/04 08:38 AM	08/24/04 10:47 AM	Open	Tech			Software Problem + Word				
3732	08/23/04 06:05 AM	08/23/04 12:34 PM	Open	Tech	High School 2	IT	IP Request	ghMac	219	Alan, Rashedul	
1343	05/04/04 11:18 PM	08/22/04 11:47 AM	Open	Tech	Service Center	Admissions	Software Problem + Web Browser	eMac			
1340	05/04/04 10:47 AM	05/04/04 10:47 AM	Open	Tech	Chandler's Ford		Software Problem + E-Mail			dennison, p	
1332	05/04/04 05:37 AM	05/04/04 05:37 AM	Open	Tech	Dave Little's Bedroom		Software Problem + Web Browser			Client, Demo	
1321	05/03/04 08:57 AM	05/03/04 08:57 AM	Open	Tech	High School 2		Computer Problem + PC			Client, Demo	

Figura 3.2 Exemplo de lista de chamados

Além destas opções, ainda o técnico tem controle sobre os recursos administrados, ou seja, os equipamentos.

O administrador tem direitos iguais aos do técnico, acrescido de algumas outras funcionalidades, tais como as de verificar as condições dos chamados no momento, sendo capaz de alterar estes chamados e priorizá-los, caso seja necessário. Pode também gerenciar todos os registros na base de dados, como cadastro de usuários, de equipamentos, clientes, lugares gerenciados pelo sistema, etc. Um exemplo do ambiente do administrador na figura 3.3.

Home Tickets Techs Clients Assets Parts Locations FAQs Reports Preferences													
My Tickets			My Calendar			Group Tickets			My Account				
New Ticket Download PDF Download TSV Priority: <input type="text"/> Status: <input type="text"/> Location: <input type="text"/> Max Display: <input type="text"/> Refresh: <input type="text"/>													
No.	Date	Updated	Status	Priority	Alert Level	Due Date	Technician	Location	Room	Department	Problem Type	Problem Report	Client
1987	07/05/04 11:31 PM	08/05/04 12:55 PM	Open	Hot	Dr. schedule		Admin			IT	Software Problem - EMail	ada kuman po Laptop keluar asap	test test
2122	07/08/04 03:20 AM	09/02/04 07:51 AM	Waiting Part	By Fall	Dr. schedule	07/05/05 05:00 PM	Admin	Company1	class1	Academic Affairs	Printer Problem - Jam	xaddads	
3253	07/13/04 04:50 PM	07/13/04 04:50 PM	Open	Urgent	Dr. schedule	07/14/04 04:50 PM	Admin	High School 2	100	Social Security	Hardware - Will not Boot	Mac is dead: My Mac won't turn on	Client Demo
3258	07/14/04 12:32 AM	07/14/04 12:32 AM	Open	Urgent	Dr. schedule	07/14/04 05:00 PM	Admin	High School 2	100	Social Security	Printer Problem - Out of Toner	sdhdt: sdtstsd	Client Demo
3268	07/14/04 00:22 AM	08/02/04 02:51 PM	Open	High	Not completed	07/15/04 03:22 PM	Admin	High School 2	100	Social Security	Hardware - Will not Boot	disk error: sound from HD	Client Demo
3278	07/15/04 05:55 PM	07/15/04 05:05 PM	Open	Urgent	Dr. schedule	07/18/04 05:00 PM	Admin				Printer Problem - Out of Toner	Nothing prints on the paper	
3288	07/16/04 09:44 AM	09/02/04 09:39 AM	Open	Urgent	Not completed	07/19/04 09:44 AM	Admin	SAL		Social Security	Account Request - E-Mail	John Smith email: Joh wants small	
3303	07/16/04 07:28 AM	07/16/04 01:54 PM	Open	Medium	Not completed	07/21/04 11:28 AM	Admin	Quad Building		Social Security	Account Request - Password Change	bed req 1: this is a test	Client Demo
3310	07/20/04 01:23 AM	07/20/04 01:23 AM	Open	1-2 Weeks	Dr. schedule	09/02/04 05:00 PM	Admin			IT Dept	Computer Problem - Modem/usb - Satcom - Communications	this is the problem: this is the problem report: this is the problem report: this is the problem repo...	B, S
3311	07/20/04 03:57 AM	07/20/04 03:57 AM	Open	Medium	Not completed	07/22/04 11:00 AM	Admin	High School 2	100	Social Security	Hardware - Will not Boot	bed: www	Client Demo

Figura 3.3 Exemplo de ambiente do administrador

Em suma, o Web Help Desk é um sistema que satisfaz as necessidades de quem procura um bom Help Desk aliado a controle de patrimônio. Apresenta uma interface amigável e funcionalidades atraentes, como a de enviar e-mails à medida que os eventos acontecem. Importante reiterar que é suportado por diversas plataformas, tanto do lado servidor, porque é desenvolvido em Java, quanto no lado cliente que interage via navegador (browser).

SysAid

O SysAid é fornecido em três versões diferentes: um sistema de Help Desk, outro sistema de inventário de bens, e outro que combina estas duas funcionalidades. Ele pode ser instalado no ambiente da empresa ou simplesmente utilizando os serviços pela Internet através do Web Service da empresa Ilient, sua fornecedora. Funciona em ASP (Active Server Pages - ambiente para programação por scripts que executam no servidor e não no cliente onde o próprio servidor transforma os scripts em HTML padrão, possibilitando que qualquer navegador

acesse um site em ASP) e garante a comunicação entre os componentes via XML. Aceita ser instalado em servidores Windows, Linux e Solaris, e os clientes acessam via qualquer navegador.

Possui um recurso baseado em agentes que consiste em instalar estes nas máquinas para que eles mantenham sempre atualizada a base de dados do servidor com relação ao hardware e software instalados em cada computador. Além disso, o agente pode ser utilizado para fornecer acesso remoto a esses equipamentos.

As atividades de manutenção podem ser desempenhadas conforme a data de vencimento do chamado, a prioridade do chamado e a qual usuário se refere. Os problemas resolvidos são acumulados na base de dados que, posteriormente, formam uma base de conhecimento para consulta.

Possui basicamente dois tipos de usuários: o usuário final e o administrador. O usuário final pode fazer chamados, ter acesso a uma lista de problemas mais comuns e suas respectivas soluções, ver o histórico de chamados e editar seu perfil. Para abrir um chamado, o processo é parecido com o sistema apresentado anteriormente, que é bastante simples e evidente: o usuário digita uma categoria de problema, escolhe um título, faz uma descrição, caracteriza a urgência e informa a qual equipamento ele está se referindo.

O administrador utiliza as seguintes listas fornecidas pelo sistema para exercer suas atividades: lista dos últimos pedidos de chamados, lista dos chamados designados a ele, e a lista dos que aguardam decisão de distribuição de responsabilidade. O sistema oferece ainda uma visão geral da situação do sistema em relação a quantos equipamentos cadastrados existem e quais grupos de equipamentos estão disponíveis (agrupamento por categoria, tipo de equipamento, etc). O menu do administrador exibe opções onde é possível a organização destes grupos de equipamentos, a visão da lista dos equipamentos, a procura de equipamentos, a adição de equipamentos, a visão dos softwares instalados, a visão dos catálogos dos equipamentos e as listas dos fornecedores.

É o sistema de interface mais simples dos três analisados, mas atende a todas as necessidades de forma clara. Exemplos de interfaces nas figuras 3.4 e 3.5.

Submit Service Request

Category:

Title:

Description:

Urgency:

Asset:

Attachments:

Figura 3.4 Tela de abertura de chamado

SysAid HelpDesk

Welcome, Joel

Experiencing technical problems? Here, you can find solutions to common problems, and request support (by submitting "service requests") from SysAid administrators.

Please choose one of the following:





-  **Submit a service request.**
-  **Self service (view common problems and their solutions).**
-  **View your old service requests.**
-  **My Settings.**

Figura 3.5 Menu principal

O menu principal dispõe de poucas opções, sendo que uma delas induz o usuário a tentar achar o problema na lista dos problemas mais comuns.

Um dos principais atrativos deste sistema é sua simplicidade. O menu do usuário, por exemplo, fornece apenas quatro opções iniciais, incentivando o uso do software de Help Desk, problema freqüentemente confrontado em ambientes das empresas.

ExDesk

O ExDesk é um sistema de Help Desk destinado a pequenas e médias empresas. Baseado em ASP, ele disponibiliza acesso aos usuários através de um navegador convencional com uma interface também muito simples, o que facilita o uso. A empresa não disponibiliza versão para ser instalada em servidores da empresa do cliente interessado; a única forma de utilizar o sistema é fazendo um contrato com a empresa onde é realizado um cadastro do cliente, fornecendo-lhe uma conta de administrador e subcontas relacionadas, que são as contas de usuários finais do sistema.

Como se trata de um serviço remoto, ele está regido por um SLA (contrato de qualidade de serviço onde o contratado se compromete a realizar os serviços de uma forma satisfatória ao cliente, levando-se em conta o tempo de serviço disponível prestado, a assistência técnica ao usuário, entre outros itens relevantes).

O administrador pode ter acesso a trinta e cinco tipos de relatórios distintos podendo deixá-los no servidor da empresa (onde são replicados, evitando problemas de perda de informação) ou importá-los para armazenar na própria empresa por qualquer motivo. Entre outras coisas, o administrador é notificado a cada novo chamado realizado através de e-mail e Pager. Uma característica notável é a capacidade de armazenar todas as ações do sistema em arquivos log que facilita a auditoria. Alguns exemplos de telas da interface nas figuras 3.6 e 3.7.

Create a new trouble ticket

Trouble Tickets Menu

Date: 2002-07-30 Time: 09:29:03PM

User ID:

kazinski, ted (helpdesk)
 Kessen, Laraine (Laraine)
 Klitch, Tori (torik)
 Lewis, Grant (GrantLewis)
 Market, John (JackMKT)
 Mills, Lence (LanceMills)
 Parker, Ridley (Ridley58)
 sinistra, dean (network)
 uum, vac (vacuum)
 Zamanski, Jessi (JessiZ)
 zazula, Jacky (jackyZZ)

Email:

First Name:

Last Name:

Phone:

Assign Admin:

Trouble Type:

Trouble Priority:

Short Problem Description:

Full Problem Description:

Figura 3.6 Exemplo de formulário de criação de um chamado.

Trouble Ticket Log		
Date & Time	Action	By
2002-02-05 14:38	Submitted	John Smith
2002-02-05 14:39	Changed assigned admin from None to John Smith	John Smith
2002-07-30 21:21	Changed trouble type from helpdesk to procurement	John Smith
2002-07-30 21:21	Changed status from Submitted to In Progress	John Smith

Trouble Ticket Comments		
Date & Time	Comment	By
2002-07-30 21:21	Ordered new peripheral device, expected delivery is 2-3 weeks please check back later	John Smith

Figura 3.7 Exemplo de lista de chamados.

Depois do pedido feito, toda a ação executada pelo usuário administrador é registrada para que o chamado tenha seu histórico formado.

Baseado na Internet, este tipo de sistema carece de um bom contrato SLA para que possa ser utilizado. Para isto, a empresa garante que seus servidores estarão 99,95% do tempo disponível ao cliente. Para empresas de pequeno e médio porte talvez esta seja uma solução ideal, pois a empresa tem apenas os encargos mensais do contrato e, assim, não precisam manter um servidor e nem pessoas especializadas para a sustentação do serviço funcionado.

3.1.3 Considerações

Quando se trata de um sistema Help Desk, uma característica freqüentemente observada é a intuitividade para que sejam acessadas as funcionalidades do sistema. Entretanto, a simplicidade não é uma característica habitual e que deve ser intensamente considerada, visto que um sistema desta natureza é destinado a um universo muito variado de usuários e seus respectivos conhecimentos. Assim, um usuário com pouca prática em sistemas computacionais poderá se deparar com dificuldade em realizar uma operação.

Quando um sistema automatiza algum processo manual e consolidado dentro de uma empresa, ele tende a proporcionar baixa aceitação e até mesmo indolência por parte das pessoas à qual o sistema se destina. Portanto, a resistência ao uso pode se tornar um obstáculo a ser ultrapassado e que pode ser combatido através da simplicidade.

O desenvolvimento de um novo sistema de Help Desk direcionado às necessidades da IOESC é importante porque existem, naquela empresa, características peculiares e inerentes ao ambiente. Assim, é possível também oferecer um grau alto de simplicidade no acesso às ferramentas do sistema aos usuários, causando baixo impacto e pouca resistência nas pessoas às quais se destina.

Construir um novo sistema de suporte técnico e não utilizar um já existente é motivado pelo desejo de manter as características do processo às quais os funcionários estão acostumados e que o regulamento determina. Se desenvolvido, um sistema deve ser, por conseguinte, uma cópia do processo manual de abertura de chamado técnico, porém na sua variante automatizada. Assim, a penetração no ambiente corporativo deve minimizar os efeitos de rejeição e o resultado é um sistema mais adequado às necessidades restritas do ambiente.

3.2 Tecnologias alternativas

Existem algumas alternativas factíveis para um sistema de suporte técnico distribuído. Na realização deste trabalho, foram analisadas as seguintes alternativas para arquiteturas distribuídas (POTTS, 2003):

- arquiteturas baseadas em RPC, como CORBA, RMI e DCOM;
- arquiteturas transacionais tipo HTTP, como servlets, JSP, ASP, PHP e CGI;
- Web Service.

A seguir são apresentadas algumas informações para fundamentar-se a decisão da tecnologia a ser usada.

3.2.1 Arquiteturas baseadas em RPC

Este tipo de arquitetura funciona fundamentalmente por acesso remoto de uma aplicação cliente a uma aplicação servidora como se fosse uma entidade local. O lado cliente é programado com objetos e métodos como em sistemas convencionais e a própria arquitetura se encarrega da comunicação como o servidor.

Os programadores desenvolvem dois módulos: o stub e a estrutura. O stub é o módulo cujo código é executado no lado cliente, sendo responsável pela identificação de objetos e métodos que estão disponíveis no servidor, codificando e decodificando as mensagens que transitam entre eles para que elas possam ser entendidas através das interfaces. A estrutura tem objetivos muito semelhantes ao stub, recebendo requisições e retornando valores, mas é executada no lado servidor, fornecendo também informação sobre os serviços existentes disponíveis ao cliente.

As principais tecnologias envolvidas nesta arquitetura são CORBA, RMI e DCOM.

O CORBA, que significa Common Object Request Broker Architecture, foi concebido para construir aplicações cliente-servidor em ambientes heterogêneos, através da IDL (Interface Definition Language) que é uma interface utilizada para detalhar a estrutura dos objetos que serão transportados em um formato neutro de linguagem, e da ORB (Object Request Broker) que é o serviço responsável de localizar um objeto na rede, comunicar a requisição ao objeto, esperar os resultados e retornar este resultado ao cliente. O desenvolvedor recolhe a IDL de onde deseja obter o serviço e a executa através de um gerador de código disponível para sua linguagem específica, obtendo com isto o stub (ou a estrutura) específica escrita na sua linguagem. Depois desta etapa o programador prossegue escrevendo o programa.

Dentre as principais características pode-se destacar a capacidade de ser independente de plataforma e linguagem, a aptidão de admitir grande carga de transações por manter conexão persistente, e a necessidade de obter portas exclusivas para se comunicarem (dificultando a passagem por firewall).

O RMI (Remote Method Invocation) também é construído sob a arquitetura stub/estrutura e se trata de uma tecnologia muito semelhante ao CORBA, porém sem utilizar IDLs. Funciona unicamente entre arquiteturas Java, tanto no cliente como no servidor.

Dentre as características pode-se expor que tem de ser escrito exclusivamente em Java, necessita utilizar uma porta específica de comunicação na rede, e manipula grandes cargas de transação porque mantém conexão persistente.

O DCOM (Distributed Common Objetc Model) é a tecnologia da Microsoft para realização de chamadas remotas. Funciona basicamente igual às outras tecnologias RPCs mencionadas anteriormente: objetos são codificados em formato que podem ser transportados na rede e decodificados quando chegam no destinatário em formato específico ao da linguagem utilizada.

Muito embora suporte várias linguagens diferentes (C++, Visual Basic, C#, etc.), só funciona mediante plataforma Microsoft.

3.2.2 Arquiteturas transacionais tipo HTTP

Esta arquitetura é baseada no paradigma de servidor web: o código é executado em um servidor web e o cliente obtém acesso através do HTTP. O servidor responde a partir de uma requisição de operação feita e retorna com as respostas e os dados que foram requeridos por meio de textos HTML, em geral. Depois de concluída a transação, a conexão entre cliente e servidor é finalizada. Isto pode evidenciar uma certa desvantagem, pois se a conexão não é persistente, ela deve ser restabelecida a cada nova transação, perdendo-se muito tempo neste processo, porque pode haver necessidade de muitas transações para concluir uma determinada tarefa. Em contrapartida, a comunicação é efetivada através da porta 80, evitando coibição no firewall.

As principais tecnologias envolvidas nesta arquitetura são os servlets, JSP, ASP, PHP e CGI

O CGI (Common Gateway Interface) é um mecanismo onde um servidor web chama um programa (script de CGI) e passa para ele todos os dados que foram enviados juntamente com a requisição do programa. O programa CGI processa os dados e cria uma página contendo a resposta que é enviada ao usuário. Este mecanismo pode causar esgotamento de recursos do sistema porque, a cada solicitação, uma instância de CGI é criada, depois processada e destruída.

As principais características são: dificuldade de encontrá-los devido à ausência de serviço de diretórios, dificuldade de escrever clientes devido à falta de APIs específicas do serviço, dificuldade de interagir por não existir um formato de troca de dados-padrão aceitos, e disponibilidade em quase todo servidor web.

Os servlets/JSP (Java Server Pages) têm a mesma aptidão do CGI, porém sem comprometer os recursos da máquina, pois ao invés de ser executado como um programa, os servlets são multithreadados, ou seja, cada nova requisição chama uma instância leve de thread ao invés das instâncias iguais aos CGIs. Eles são escritos em linguagem Java e, por isto, herdam toda a capacidade de universalidade do Java.

As principais características: só pode ser escrito em Java, dificuldade de escrever clientes para os servlets porque não possuem uma API suficientemente documentada e específica do serviço, não existe troca de dados-padrão aceitos, e capacidade de executar em muitas plataformas de servidores web.

O ASP (Active Server Pages) – solução proveniente da Microsoft baseada em Visual Basic – e o PHP (Hypertext Preprocessor) – solução sustentada pelas comunidades Open Source para Linux e Unix – são essencialmente iguais em fundamentos. As duas soluções exigem que o desenvolvedor misture código e HTML para produzir os dados textuais resultantes. Possuem características semelhantes ao CGI e ao servlet no que diz respeito a encontrar serviços e saber como se comunicar com eles, porém são apropriadas para construir sistemas baseados em web para utilização de pessoas. O ASP pode ser executado em vários tipos de servidores web, mas são, na maioria dos casos, implantados em Windows. O PHP executa em inúmeros ambientes, mas é nos sistemas Unix onde ele se identifica mais.

Dentre as principais características podem-se destacar: ausência de serviço de diretórios, dificuldade de escrever clientes pela falta de uma API bem documentada específica do serviço e em que se basear (POTTS, KOPACK, 2003),

inexistência de padrão para troca de dados, facilidade de programação no servidor devido à simplicidade das linguagens relacionadas, e portabilidade a outros servidores que prestam suporte a essas linguagens (do lado cliente pode ser implementado em qualquer linguagem).

3.2.3 Web Service

Antes de entrar-se em detalhes sobre os Web Services, é conveniente mencionarem-se dois assuntos que, juntos, ajudam a entender a essência deste serviço. O primeiro é em relação às organizações responsáveis pelo controle das tomadas de decisões das normas e das versões de cada uma das principais especificações do Web Service. O segundo é em relação a uma tecnologia imprescindível para a independência de plataforma que o Web Services promove: o XML.

Controle das especificações

As três organizações mais importantes que fornecem as normas e recomendações dos Web Services são:

1. World Wide Web Consortium (W3C) – é a mais importante dessas organizações. É um consórcio que reúne catorze empresas interessadas em definir novas tecnologias para a Internet cujos produtos do trabalho são compilados por concordância das partes, gerando forte apoio às recomendações. Ela controla as especificações SOAP, WSDL, XML, XML Schema e HTTP (explicados em detalhes a seguir). Entre outras coisas, administra um documento que representa um ensaio de definição formal dos Web Services: o WS-Architecture, que ainda é muito imaturo e rudimentar.
2. Organization for the Advancement of Structured Information Standards (OASIS) – é uma associação global que promove o desenvolvimento de padrões para o comércio eletrônico. Ela controla as especificações UDDI, WS-Security entre outras.
3. Web Service Interoperability Organization (WS-I) – é uma organização com objetivo de promover a interoperabilidade dos Web Service entre as plataformas existentes (sistema operacional e linguagens de programação)

através de orientação, ensino de boas práticas e recursos capazes de fazer diferentes padrões funcionarem em conjunto.

XML

O acrônimo XML significa “Extensible Markup Language” ou linguagem de marcação de dados. Seu surgimento ocorreu para simplificar a tarefa de trocar documentos entre sistemas de computadores. O problema que ele resolve não consiste em transmitir simplesmente documentos em formato de texto de um computador para outro, mas sim tornar esses documentos transportados em documentos facilmente compreensíveis entre sistemas diferentes. Para isso, o XML define regras que determinam o formato das tags (espécie de etiqueta eletrônica para marcar alguma informação, exemplo <caneta>), dos caracteres permitidos, dos significados dos caracteres e do formato geral de um documento. Isto acontece de uma forma análoga a uma gramática que diz como se deve escrever um texto para que alguém possa ler, porém, não determina o significado das palavras empregadas no texto. Da mesma forma acontece em um documento XML. A gramática foi definida e o vocabulário é empregado de maneira tal que as partes interessadas conseguem se entender.

Não é uma nova linguagem. Na verdade trata-se de uma metalinguagem utilizada para marcação de dados onde é possível estruturar-se e classificar-se a informação no documento.

Este padrão foi aprovado pela W3C em fevereiro de 1998.

Regras do XML

Um documento XML é simples e deve possuir as seguintes regras gramaticais básicas:

- para cada tag inicial deverá existir uma tag final correspondente. Exemplo: <caneta> ... </caneta>;
- atributos precisam estar entre aspas. Exemplo: <caneta id="1">;
- tags devem estar acomodadas corretamente. Exemplo: uma seqüência de tags que começa com <a> e depois deve sempre acabar com depois .
- todo documento XML deve começar com <?xml version='1.0'?>

Um exemplo de documento XML na listagem 3.8.

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<!-- Documento de exemplo de XML-->
<ine>
  <aluno matricula="00238139">
    <nome> Fabio </nome>
    <sobrenome> Vieira </sobrenome>
  </aluno>
</ine>
```

Listagem 3.8 Um exemplo de documento XML

Os documentos que adotam estas regras são considerados consistentes. Os documentos inconsistentes retornam uma mensagem de erro quando analisados.

Componentes do XML

Os componentes do XML são:

- Documento XML: é o arquivo XML propriamente dito. É onde estão os dados que o documento XML possui.
- Parser XML: é um programa que lê o arquivo XML e traduz a informação contida nele para um programa interessado em processar a informação.
- Document Type Definition (DTD): um documento XML foi projetado para ser autocontido, utilizando o DTD para descrever a estrutura e a relação entre as tags do documento XML. Ele apresenta duas características inadequadas: não seguir a sintaxe XML e não possuir flexibilidade satisfatória para definir tipos de dados. Por isto é considerado obsoleto depois da publicação da especificação do esquema XML em 2001.
- Esquema XML: a exemplo do DTD, ele descreve a relação das tags no documento. Um parser XML compara um documento XML a seu respectivo esquema XML para garantir que as tags estão de acordo com as regras definidas. Os esquemas XML especificam, além dos elementos e atributos, os tipos de dados de um elemento e sua faixa de valores. Um arquivo de esquema é um arquivo XML convencional com as tags possuindo um prefixo xsd denotando que elas fazem parte de um esquema XML. A seguir, na

listagem 3.9, um exemplo de um documento XML e na listagem 3.10 seu esquema correspondente.

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<!-- Documento de exemplo de XML-->
<ine>
  <aluno matricula="00238139">
    <nome> Fabio </nome>
    <sobrenome> Vieira </sobrenome>
  </aluno>
</ine>
```

Listagem 3.9 O Documento

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Documento de exemplo de XML-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="aluno_ine" type="aluno"/>
  <xsd:complexType name="aluno">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="sobrenome" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Listagem 3.10 O esquema XML

O arquivo XML deve obedecer ao seu esquema XML, validando o documento.

- Namespaces: recurso para garantir que as tags sejam únicas dentro de um documento XML, evitando confundir as informações contidas. Levando-se em conta o princípio da modularidade, onde documentos XML podem conter outros documentos XML, é fácil perceber que documentos XML podem adquirir tags iguais. Um exemplo de tag que facilmente seria usada por diversos documentos XML é <istemas>. A melhor solução para este problema é definir prefixos que sejam seguramente únicos no mundo. Uma maneira de prefixar é utilizando uma URL (endereço). Exemplo: <www.inf.ufsc.br:sistemas>. Estaria quase garantido que nenhuma outra tag

seria igual. Porém, isto pode gerar tags muito grandes. Para resolver este problema surgem os namespaces que definem os apelidos que esses prefixos das tags vão assumir no documento. A palavra reservada “xmlns”, quando empregada, significa que um namespace está sendo definido como no exemplo: xmlns:meuatalho=”www.atalho.com.br”. Quando for utilizado, deverá ser: <meuatalho:minhatag> </meuatalho:minhatag>.

Web Service

O Web Service é um conjunto de aplicações que, juntas, são capazes de exercer atividades de caráter remoto com insumos heterogêneos e independentes de plataformas ou produtos de empresas específicas. Isto acontece com chamadas de procedimento remotas através de uma rede, como a Internet, que utiliza padrões neutros (HTML e XML).

Um cliente de Web Service não precisa saber qual sistema operacional o servidor utiliza e também não precisa saber em que linguagem de programação o serviço funciona. É semelhante ao uso de um navegador na Internet: o usuário só precisa saber o endereço para fazer uma requisição do conteúdo.

Um serviço web, como o Web Service, possui um modelo conceitual definido pelos seguintes papéis:

- o Provedor de serviço é a entidade que proporciona o serviço a ser prestado;
- o Consumidor de serviço é a entidade que consome o serviço disponível;
- o Registro de serviço é onde o provedor de serviço publica a suas funcionalidades e onde o cliente faz pesquisas para saber qual provedor possui o serviço que ele gostaria de contratar.

A figura 3.11 mostra a interação entre as entidades citadas.

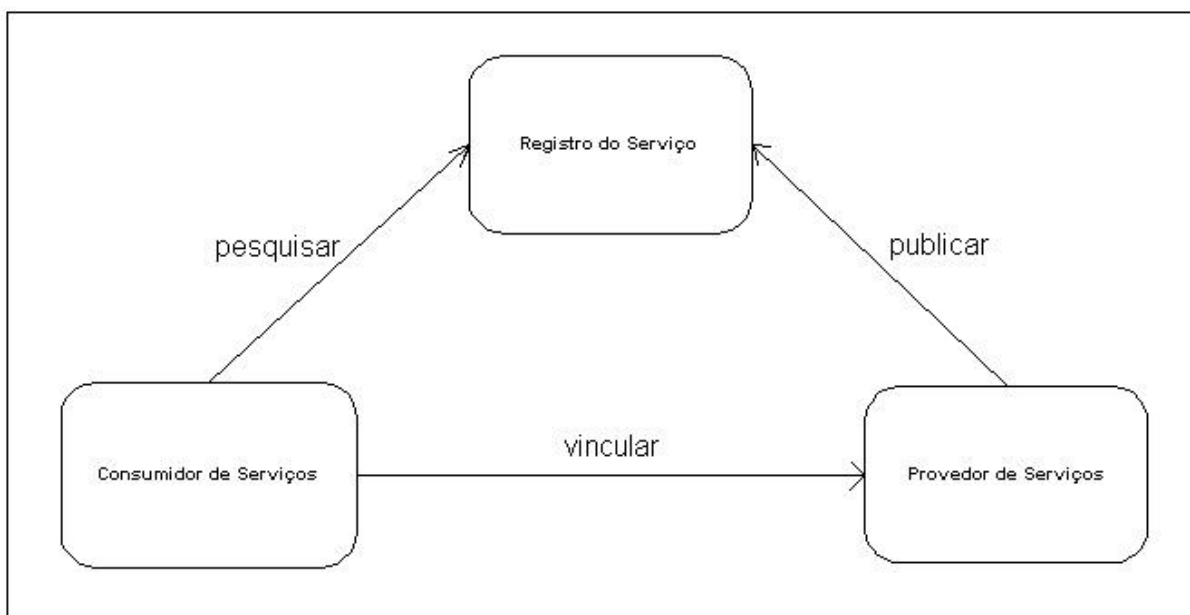


Figura 3.11 Interação entre Provedor, Consumidor e Registro

Fonte: [HENDRICKS, 2002], página 10

No ponto de vista de interoperabilidade, o Web Service pretende que qualquer aplicação seja virtualmente capaz de se comunicar com qualquer outra aplicação de software em qualquer outro lugar do planeta.

Os componentes que compõem as operações podem ser originados de programas elaborados por iniciativa do interessado ou utilizando alguma ferramenta já existente com licença livre de uso, ou comprado de fornecedores de softwares. Porém, esta grande possibilidade de interações proporcionada por esta livre escolha leva a crer que se torna difícil conciliar as partes. Para isto existem os principais padrões utilizados para nortear o Web Service.

Padrões do Web Service

Os padrões do Web Service são proporcionados por três processos básicos: SOAP, WSDL e UDDI. A pilha básica do Web Service é representada na figura a seguir. O texto do lado esquerdo de cada camada tem o objetivo de tornar inteligível o conceito citado no lado direito. Cada camada é construída sobre a camada precedente, como mostra a figura 3.12.

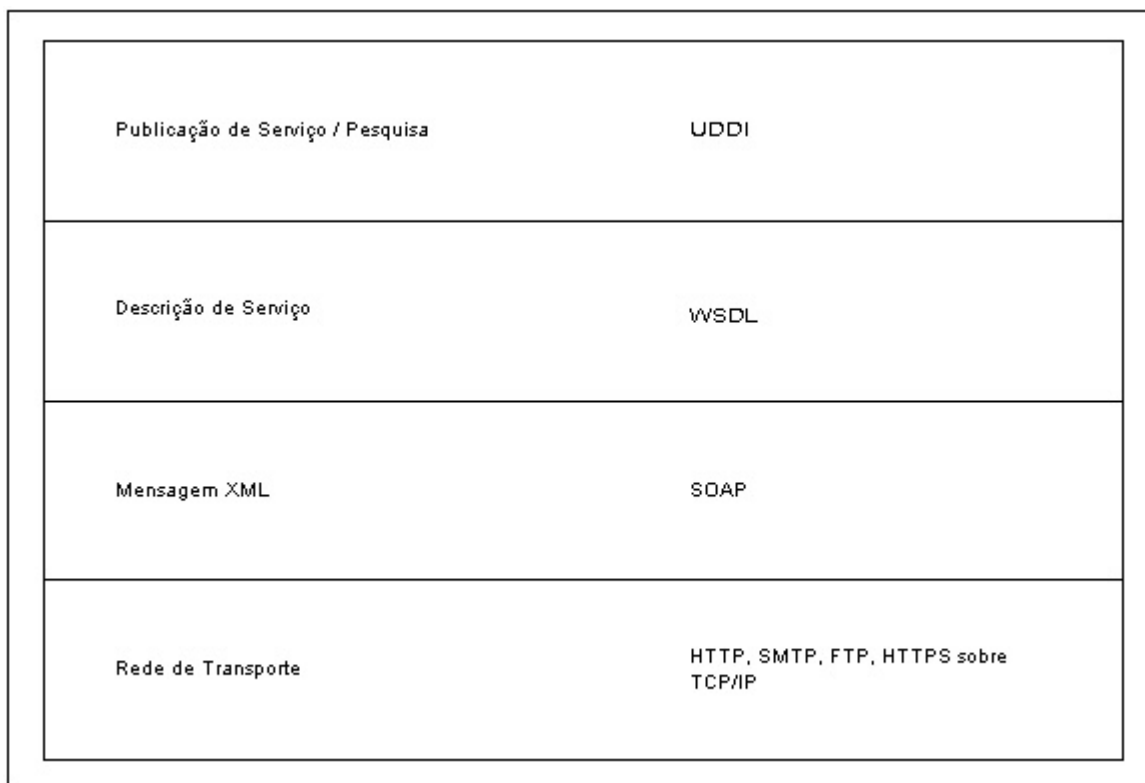


Figura 3.12 Pilha básica do Web Service

Fonte: [HENDRICKS, 2002], página 11

Agora que os principais padrões foram citados e superficialmente colocados no contexto dos Web Services, cada um deles é explicado a seguir.

HTTP

A camada de transporte é a camada mais baixa da pilha de serviço dos Web Services e é responsável simplesmente pelo transporte de uma informação do ponto A para o ponto B. O HTTP “hipertext transfer protocol”, que é controlado pela W3C, é o protocolo de transporte mais utilizado no Web Service por se tratar de um protocolo maduro e largamente utilizado. Dentre outras características, este protocolo atravessa com bastante facilidade os firewalls das instituições porque, via de regra, a porta que ele utiliza, a porta 80, está liberada para que os usuários da rede possam acessar páginas na Web. Contudo, os Web Services podem também ser utilizados com quaisquer outros protocolos, como, por exemplo, o FTP “file transfer protocol” ou o SMTP “Simple Mail Transfer Protocol”.

A finalidade do protocolo HTTP é carregar qualquer tipo de dado, dando suporte ao processo de requisição e resposta entre cliente e servidor. No caso dos Web Services, fará o transporte dos arquivos SOAP entre o consumidor e o provedor de serviço. Ele está na versão 1.1 e as atividades sobre ele foram encerradas por ser considerado estável.

SOAP

A origem do SOAP (Simple Object Access Protocol) é de uma tentativa de definir uma técnica de chamada de procedimento remoto de um computador para outro, chamado XML-RPC, elaborada por Dave Winer em uma empresa com o nome de UserLand. A Microsoft, a IBM e a IONA decidiram por melhorar e aprimorar o método XML-RPC, formando assim a especificação SOAP que utiliza padrões públicos vastamente utilizados, como o HTTP e o XML. Pouco tempo depois, foi submetida ao Consórcio W3C.

O SOAP é um documento de alta abstração, contendo uma especificação escrita num arquivo XML que serve para trocar informações entre dois programas sem limite de sistema operacional e linguagem de programação. Sua concepção também permite vinculá-lo a qualquer protocolo de transporte, tais como HTTP, FTP e SMTP.

A composição estrutural de uma mensagem SOAP é um envelope, um cabeçalho e um corpo. O envelope define o conteúdo, a descrição, a forma de processamento e os namespaces utilizados no arquivo da mensagem. O cabeçalho é opcional e serve para colocar informação adicional necessária. O corpo é o cerne, onde estão as informações sobre as chamadas dos métodos, os parâmetros a serem passados, as respostas da chamada e o relatório de falhas.

Seu funcionamento é resumidamente assim: o computador A envia um documento XML para um servidor B através do protocolo HTTP. O servidor Web de B recebe o arquivo e o encaminha para o seu processador SOAP que utiliza um parser de XML para analisar o documento. Depois de validar o arquivo XML pelo parser, o processador SOAP precisa legitimar este arquivo levando em conta a gramática SOAP e o esquema XML. Se ele passar por esta avaliação, o processador SOAP chama o Web Service no servidor B descrito no documento SOAP e passa os parâmetros que o documento contém. Depois do Web Service terminar o

processamento programado, ele criará uma resposta em um formato de mensagem SOAP que o servidor web vai remeter ao solicitador inicial. Quando chega ao computador A, o programa cliente HTTP repassa ao processador SOAP para que ele possa analisar o arquivo e validá-lo.

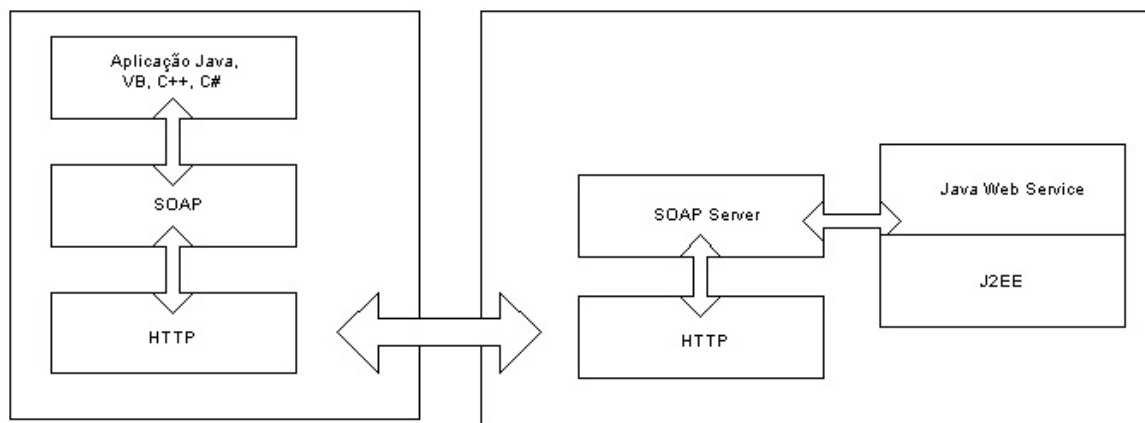


Figura 3.13 Comunicação utilizando protocolo SOAP

Fonte: [HENDRICKS, 2002], página 13

A gramática SOAP, como já comentado, é composta por duas partes obrigatórias, que são o corpo e o envelope e uma opcional que é o cabeçalho. Todas as tags de uma mensagem SOAP possuem o prefixo SOAP-ENV. A tag do envelope é <SOAP-ENV:Envelope>, a do cabeçalho <SOAP-ENV:Header> e a do corpo fica <SOAP-ENV:Body>. A figura 3.14 mostra uma representação gráfica de uma mensagem SOAP e a listagem 3.15 mostra um exemplo de mensagem SOAP real.

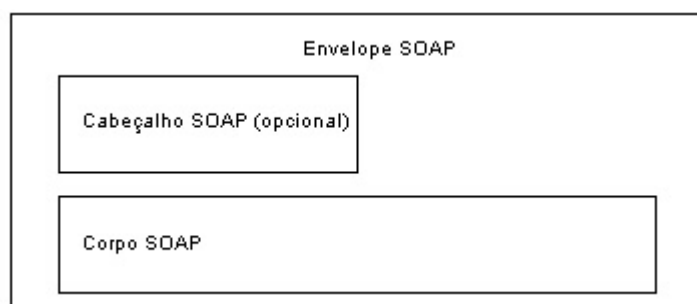


Figura 3.14 Exemplo gráfico de uma mensagem SOAP

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<!-- Documento de exemplo de envelope SOAP-->
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Header>
```

```

        <h:identify xmlns:h="http://www.inf.ufsc.br/~fabiov/header">fabiov@inf.ufsc.br</h:identify>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <m:GetStockQuote xmlns:m="http://www.inf.ufsc.br/~fabiov/getstockquote/">
            <m:ticker>UFSC</m:ticker>
        </m:GetStockQuote>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Listagem 3.15 Exemplo de mensagem SOAP

WSDL

No Web Service, todo serviço prestado por um servidor é descrito formalmente através de um documento XML chamado WSDL (Web Service Description Language). Esta é a interface do Web Service: a sua face. Um dos conteúdos deste documento são informações que servem de subsídio para alguém avaliar se o serviço prestado pelo respectivo Web Service é útil. O restante do conteúdo são informações de como fazer uma conexão a este Web Service. Desta forma, de posse de um documento WSDL, o usuário poderá acessar os serviços simplesmente analisando este documento. O WSDL foi desenvolvido pela Microsoft, IBM e Ariba, sendo, mais tarde, submetido ao W3C.

Os documentos WSDL são divididos em descrições concretas (elementos que servem para informar como conectar fisicamente o cliente ao serviço) e abstratas (elementos que descrevem a capacidade do Web Service).

Os elementos abstratos são:

- Type: os elementos contidos em <type> representam a definição dos tipos dos dados que o web Service lida. No caso de sua ausência na especificação, o significado é que o serviço utiliza apenas tipos de dados simples, tais como strings, inteiros, etc.
- Message: um potencial cliente, ao analisar este elemento, tem a informação necessária para deduzir o que o Web Service vai processar. Cada elemento <message> possui um ou mais elementos <part> (cada elemento <part> se refere a um tipo definido no elemento <type>).
- Operation: é como se fosse a chamada de um método em um sistema orientado a objetos conforme declarado pelos elementos <message>. As três mensagens permitidas são: mensagem de entrada (dados que o Web Service

confia receber), mensagem de saída (dados que o Web Service confia que enviará) e mensagem de falha (mensagens de erro). Existem quatro tipos de mensagens diferentes para realizar-se operação. São elas: operação unidirecional (cliente envia uma mensagem ao serviço sem obter uma resposta), solicitação-resposta (representa o tipo mais comum onde o cliente faz uma solicitação ao serviço e obtém uma mensagem de resposta como resultado), pedido-resposta (mensagem do serviço ao cliente, fazendo uma solicitação e obtendo uma resposta do cliente), e notificação (serviço envia uma mensagem ao cliente, mas não espera uma resposta).

Os elementos concretos são:

- Porttype: é o conjunto de todas as operações que um Web Service oferece.
- Binding: este elemento descreve os mecanismos de conexão usados pelo Web Service. Possui, basicamente, duas finalidades. A primeira é de criar uma ligação entre os elementos concretos e os abstratos do documento WSDL. A segunda é dispor de informações de endereço e protocolo de comunicação com o Web Service.
- Port: onde está definido o endereço e a porta de conexão do Web Service.
- Service: é onde estão contidas as portas que são apresentadas pelo documento WSDL. Como as portas não podem ser encadeadas, o elemento Service fornece este proveito, facilitando o esquema “saída de uma porta é a entrada da outra”. Diversos elementos <port> podem ser especificados dentro do elemento <service>.

Existe ainda o elemento Definitions (<definitions>) que representa a raiz do arquivo XML.

A figura 3.16 representa um documento WSDL.

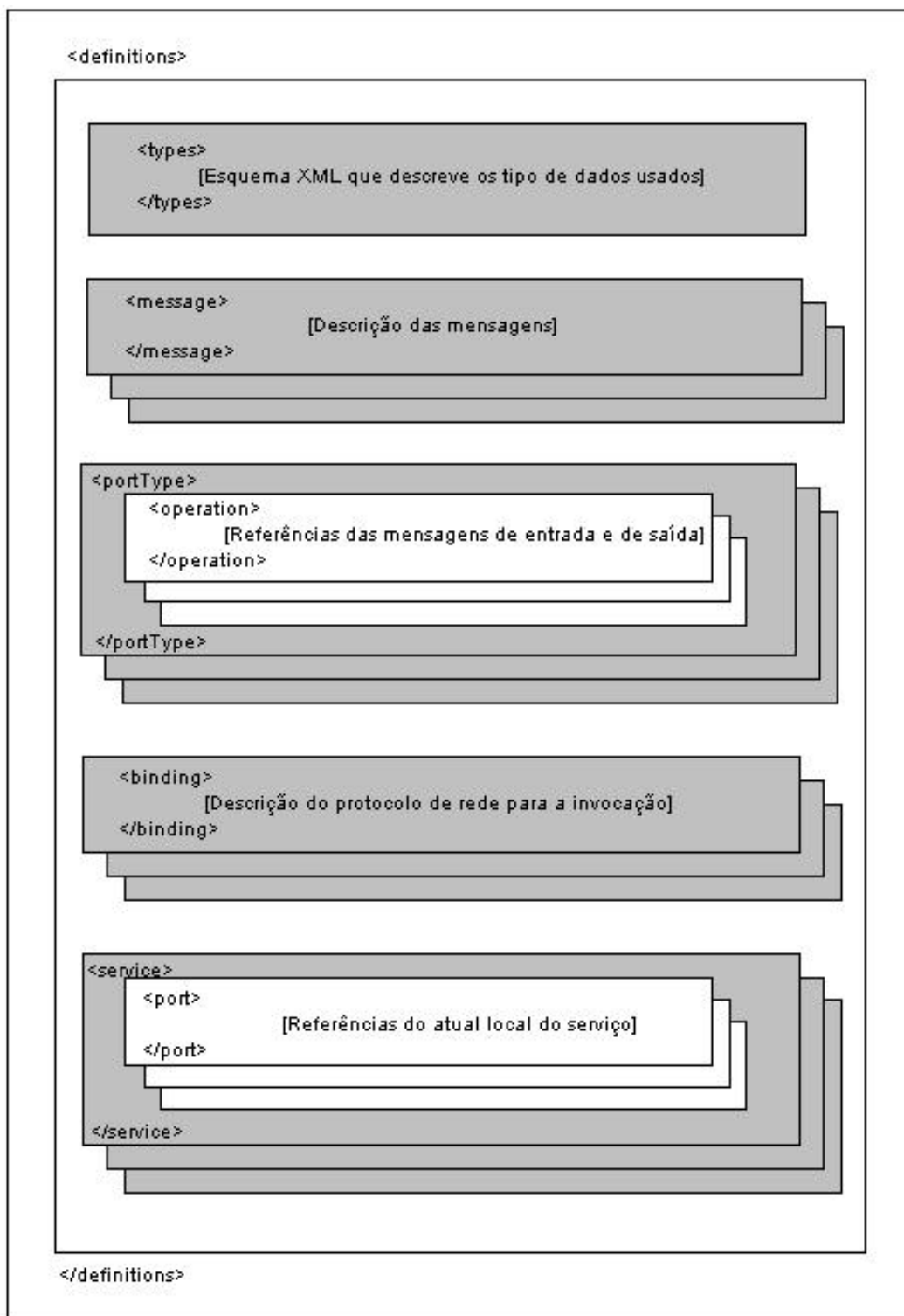


Figura 3.16 Representação de um documento WSDL

Fonte: [HENDRICKS, 2002], página 103

UDDI

Um provedor de serviços utiliza um arquivo WSDL para descrever sua empresa e seus serviços prestado. Ao invés de ter que distribuir este arquivo a cada cliente em potencial, ele pode publicá-lo em algum registro central que esteja disponível aos consumidores interessados.

Sinteticamente, UDDI (Universal Discovery, Description and Integration) é um padrão de registro onde se publicam e se procuram serviços de Web Service. Em outras palavras, é um mecanismo de descrição de serviços que, através de métodos simples, permite interagir em um registro central. Embora o UDDI se refira ao arquivo WSDL, isto não quer dizer que um arquivo WSDL esteja copiado fisicamente num registro UDDI. Ele, na verdade, fica em algum lugar na Web acessível por uma URL, sendo que esta última é copiada no registro UDDI.

Um servidor de registros UDDI é formado por um conjunto de registros onde cada registro informa a respeito de alguma entidade. Quando um registro é publicado em algum servidor de registro UDDI público, todos os outros servidores de registros públicos UDDI compartilharão esse registro no momento em que sincronizarem. Isto possibilita que uma pesquisa feita em um servidor de registro UDDI tenha o mesmo efeito de uma pesquisa feita em cada um dos outros servidores de registros.

O UDDI surgiu do interesse das empresas Ariba, IBM, Intel, Microsoft e SAP. Posteriormente, no ano de 2002, o grupo OASIS assumiu o controle através da UDDI.org através do qual diversas outras empresas apóiam e matem o projeto.

O UDDI baseia-se em quatro especificações. A primeira delas é referente a uma estrutura de dados que determina os tipos de dados que serão armazenados no registro UDDI. Esta estrutura é baseada em XML e pode ser, inclusive, um XML Schema. A segunda consiste nas especificações das APIs para acesso a um registro UDDI, tanto para consulta como para publicação. Elas são elaboradas em XML, depositadas em envelopes SOAP e enviadas por HTTP. Assim, garante que qualquer plataforma pode suportar. A terceira consiste nas especificações para replicação de registros entre os servidores. Por último, a quarta consiste na especificação do operador que define as políticas de segurança e de gerenciamento de dados.

A estrutura de dados de um registro UDDI imita um catalogo telefônico. Os dados num registro são estruturados da seguinte forma:

- White Pages: informação sobre nome, telefones, endereço e diversas informações sobre o negócio;
- Yellow Pages: índice de produtos e serviços;
- Green Pages: para informar sobre os serviços oferecidos e suas especificações técnicas para acessá-los.

Na figura 3.17, um exemplo de como funciona um registro UDDI.

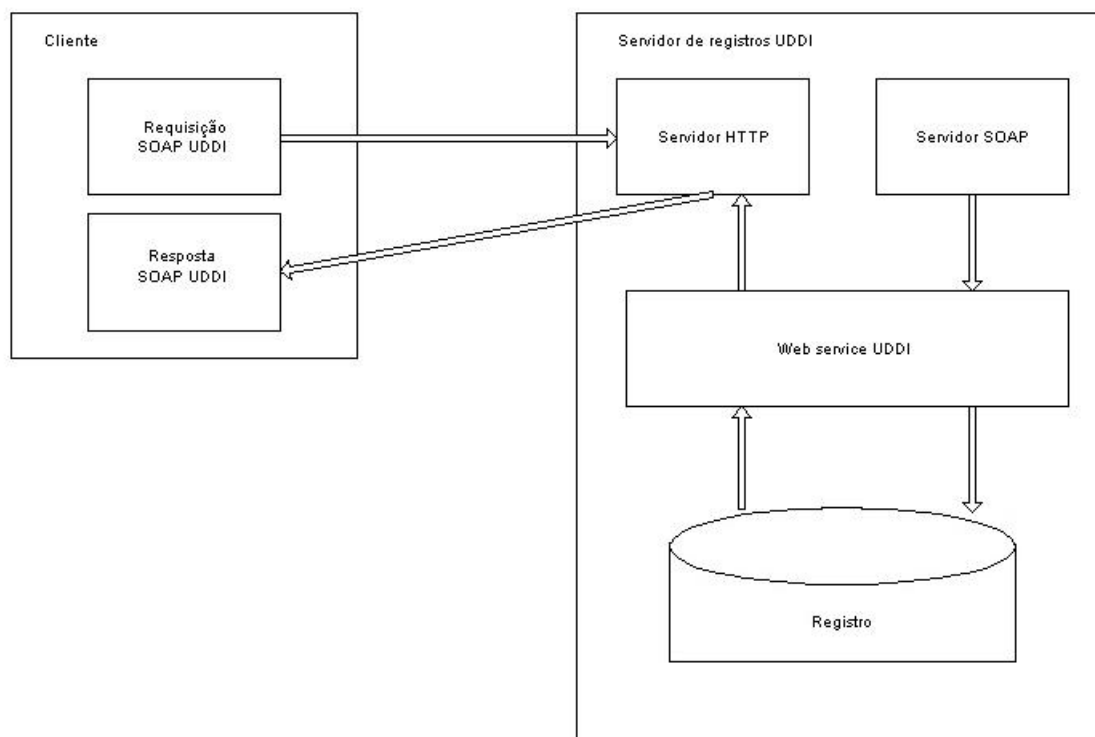


Figura 3.17 Funcionamento de um registro UDDI

Fonte: [POTTS, 2003], página 138

Os principais elementos contidos num registro UDDI são:

- tModel: especificação técnica de serviço num registro UDDI, como, por exemplo, uma interface de um serviço Web;
- BusinessEntity: contém informações sobre um negócio, tal como nome, endereço, etc.;
- BusinessService: descrição dos serviços oferecidos;
- BindingTemplate: Informações específicas sobre um serviço.

A figura 3.18 representa a decomposição de um documento WSDL em XML em um registro UDDI.

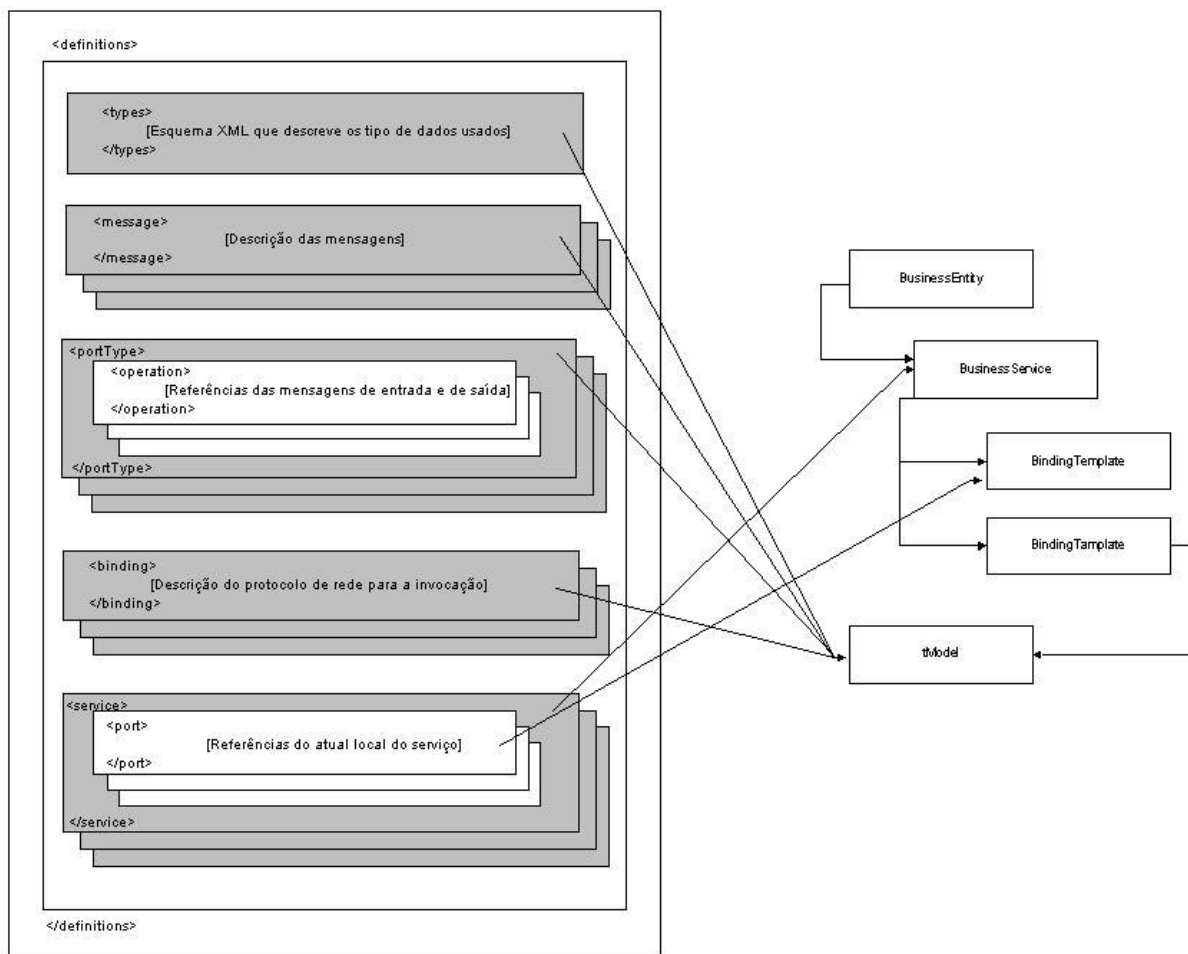


Figura 3.18 Decomposição de um documento WSDL no registro UDDI

Interação entre as tecnologias SOAP, WSDL e UDDI

As tecnologias que formam o Web Service interagem de forma a constituir um serviço único e compartilhado. A seguir, é descrito e mostrado através da figura 3.19 um processo hipotético de um Web Service em Java.

Existe uma seqüência de etapas macro que são:

1. Publica Informações WSDL:
 - a. o lado servidor gera um arquivo WSDL através de uma ferramenta ou manualmente;
 - b. o arquivo WSDL é publicado em um registro UDDI.
2. Obtém Informações WSDL:
 - a. o lado cliente, ou consumidor do serviço, procura o serviço que mais se ajusta às suas necessidades.
3. Mensagens SOAP:

- a. de posse do arquivo WSDL, o cliente tem condições de saber quais mensagens ele pode enviar ao Web Service;
- b. extrai do arquivo WSDL as informações de vínculo, neste caso, seria mensagens SOAP pelo protocolo de rede HTTP;
- c. ocorre a troca de mensagens SOAP entre o cliente e o Web Service.

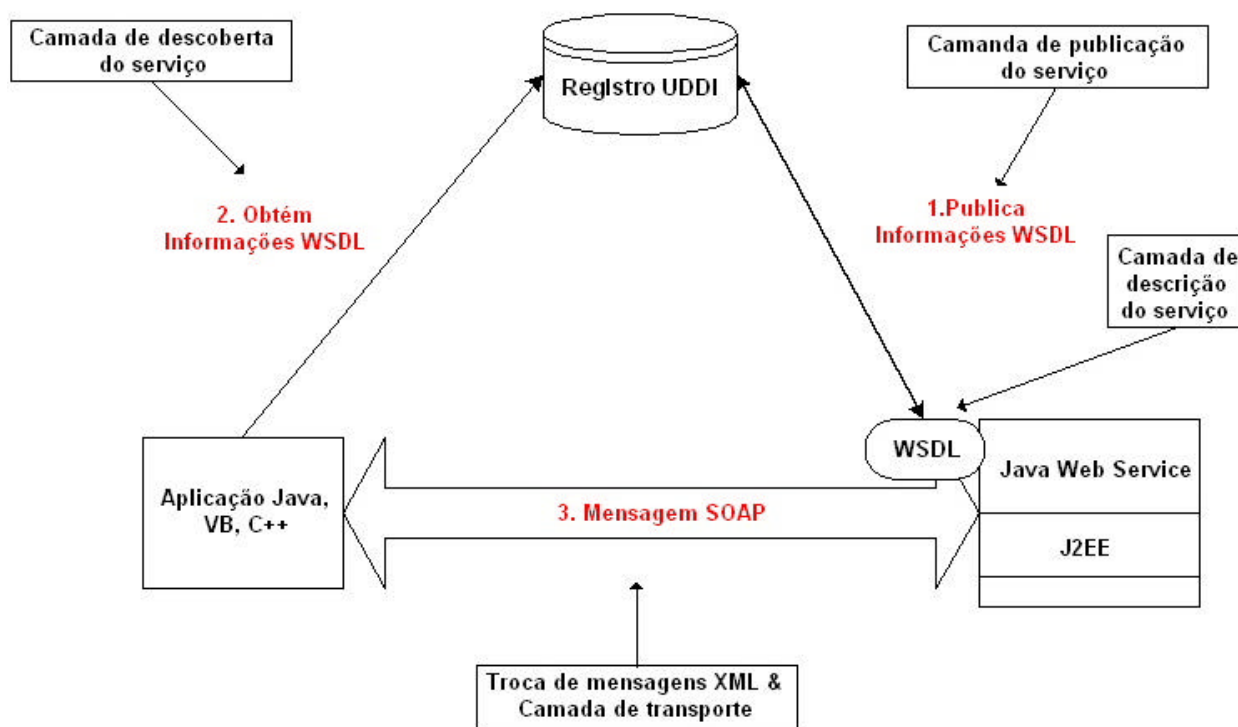


Figura 3.19 Processo de um Web Service em Java

Fonte [HENDRICKS, 2002], página 16

4 PROPOSTA DE SOLUÇÃO

4.1 Processo de manutenção informatizado

O objetivo de criar um sistema de gerenciamento de chamados técnicos não foi apresentar uma solução pronta e funcionalmente completa, mas sim dar o primeiro passo para que o processo de manutenção deixe de ser manual e passe a ser regulado por um sistema mais dinâmico.

O sistema é inspirado no processo convencional, descrito no capítulo dois, imitando diversos procedimentos que estão consolidados entre os funcionários e o regulamento da instituição. Desta forma, o sistema estará adequando as necessidades da IOESC e poderá, entre outras coisas, proporcionar no futuro, através do histórico dos chamados, uma base de dados capaz de gerar uma base de conhecimento.

O sistema deve gerenciar o processo de manutenção de equipamentos de informática até o fim, atribuindo tarefas às pessoas responsáveis, e mantendo o histórico dos relatórios produzidos a cada chamado.

Funcionamento

O sucesso de um software não necessariamente ocorre em virtude da quantidade de recursos que ele pode proporcionar. Às vezes o sucesso significa abranger unicamente o necessário, alcançando, assim, uma boa aceitação e a ampla utilização.

Desde o princípio da formulação deste projeto, uma das prioridades foi apresentar uma interface simples. Com este artifício pretende-se que o sistema produza uma baixa rejeição inicial por parte dos usuários relacionados com o processo de manutenção. A segunda prioridade foi moldar o sistema, considerando a metodologia de trabalho regulada pela empresa e convencionada pelos funcionários. Por fim, a terceira prioridade foi proporcionar um sistema que torne o processo de manutenção mais rápido comparado à maneira convencional.

Mais adiante, neste trabalho, no capítulo 5, é possível acompanhar o desenvolvimento analítico do ambiente onde são levantados os requisitos e as características do processo.

Nome

Foi dado ao sistema o nome “Chasque”. Esta palavra expressa um recado levado de uma pessoa a outra através de um terceiro elemento de confiança e de absoluta dignidade. Uma analogia ao sistema.

Usuários

O sistema é baseado em validação de usuários para liberar o acesso. Ele suporta três tipos de usuários: o Administrador, o Técnico e o Utente (usuário final). O usuário Utente, o mais limitado dos usuários, tem permissão apenas para abrir chamados técnicos e pesquisar outros chamados existentes. Esta característica ajuda na objetividade e evita o desvio da atenção do usuário para recursos secundários do sistema.

O usuário Técnico tem, além de todas as mesmas prerrogativas do usuário Utente, permissão de cadastrar equipamentos e de alterar alguns valores dos chamados. Estes valores são campos abertos para que o Técnico preencha, informando os procedimentos efetuados para consertar o equipamento.

O usuário com status de Administrador tem acesso irrestrito ao sistema. Ele pode exercer quaisquer funções citadas anteriormente, além de ter permissão de gerenciar as contas de usuários, alterar todos os chamados (daqui para frente intitulados de “tickets”) e controlar o cadastro dos equipamentos.

Cada tipo de usuário existente neste sistema sugere os cargos existentes hoje na SUATEC. Por exemplo: o usuário Administrador pressupõe-se que seja o Supervisor; o usuário Técnico pressupõe-se que seja algum técnico da SUATEC; e o usuário Utente presume-se que seja todo usuário final de equipamentos.

Dispositivos

Os dispositivos que fazem parte do universo em questão são: microcomputadores, notebooks, servidores, impressoras, scanners, estabilizadores, servidores de impressão, hubs, roteadores, switches, aparelhos de fax, no-breaks, câmeras fotográficas digitais e zip-drives.

Assistência externa

A exemplo do que acontece no processo convencional, quando existe a necessidade de assistência técnica externa, ela é derivada e relatada somente a partir do usuário Administrador (supervisor). Os procedimentos para este tipo de chamada, a consulta de preços e a autorização da gerência não estão no escopo deste sistema. Entretanto, informar qual empresa prestou o serviço e relatar os procedimentos de manutenção realizados por esta assistência técnica externa deve ser narrado pelo Administrador no sistema. Ele registra toda a conjuntura desta fração do processo de tratamento do ticket correspondente àquele chamado.

Ticket

O ticket é o elemento central do sistema. Ele representa cada chamado feito por algum usuário que necessita de assistência técnica. Nele está contida a descrição do problema, o vínculo com o equipamento, o vínculo com as pessoas envolvidas (técnico responsável e o autor do chamado) e outras informações importantes. Também é no ticket que o Técnico descreve os procedimentos efetuados para resolver o problema. Quando acaba um processo de manutenção, o ticket é arquivado.

Abandono de ticket

É natural que durante o tratamento de algum ticket o técnico chegue à conclusão de que não possui condições de resolver o problema. Procedendo este fato, ele estará dirigindo fatalmente este ticket ao Administrador do sistema que tomará providências acerca do que fazer. O Administrador jamais trata um ticket a não ser que ele seja abandonado. A partir daí, depois de analisar as descrições

contidas no ticket, ele tem duas opções: delegar este ticket a outro Técnico ou chamar a assistência técnica. Neste caso, o ticket fica sob sua tutela até que seja descrito o procedimento listado pela assistência.

Divisão do sistema

Em aplicações distribuídas um programa disponibiliza determinados métodos para que sejam invocados por outro programa remotamente. Assim, neste sistema proposto, os usuários usufruem funcionalidades em suas máquinas que na verdade são processadas no servidor.

O sistema é dividido em duas partes: servidor e console (instalados em todas máquinas clientes). No servidor todas as funcionalidades do sistema estão disponíveis e somente o usuário Administrador tem acesso. No console só as funcionalidades de abertura de chamado e de tratamento de chamado estão disponíveis, sendo que qualquer usuário tem acesso.

Categoria de problemas

Cada ticket é classificado pelo Técnico com uma característica de problema predefinida. Essas categorias foram levantadas pelos técnicos da SUATEC através de observações e têm caráter heurístico. As categorias de problemas são: vírus, sistema operacional, programas de trabalho, programas diversos, hd, memória, morosidade de processamento, hardware geral, monitor, Internet e e-mail.

Comportamento

Depois de introduzida as principais características do sistema que suporta o processo de abertura de chamado técnico é possível fazer-se um esboço do seu comportamento.

O usuário que possui algum problema técnico dirige-se a um computador e acessa o console da aplicação para relatar os acontecimentos. O chamado é transmitido a algum Técnico que tem o compromisso de encaminhar uma solução ao problema. Depois de deliberada a ocorrência, o Técnico relata o procedimento tomado no Ticket através de um console do sistema e arquiva o chamado.

Caso o Técnico não consiga resolver o problema, ele executa um processo do sistema chamado “abandono de chamado”. Com isto, o Administrador assume o chamado deixado pelo Técnico. De posse desse chamado, o Administrador decide se chama uma assistência externa ou se ele delega o chamado a outro Técnico. Se ele chamar a assistência técnica, o processo ocorre de forma idêntica ao processo manual até o momento de receber o equipamento consertado. Depois do equipamento reparado, o Administrador tem o dever de relatar os procedimentos tomados pela empresa prestadora de serviço.

É importante salientar que somente o Administrador pode interagir na parte servidora da aplicação, portanto, só ele pode executar as operações de manutenção de sistema. A figura 4.1 mostra como seria o processo informatizado. A figura 4.2 representa o esquema de funcionamento da aplicação cliente e a da aplicação servidora dentro e fora da IOESC (através da agência).

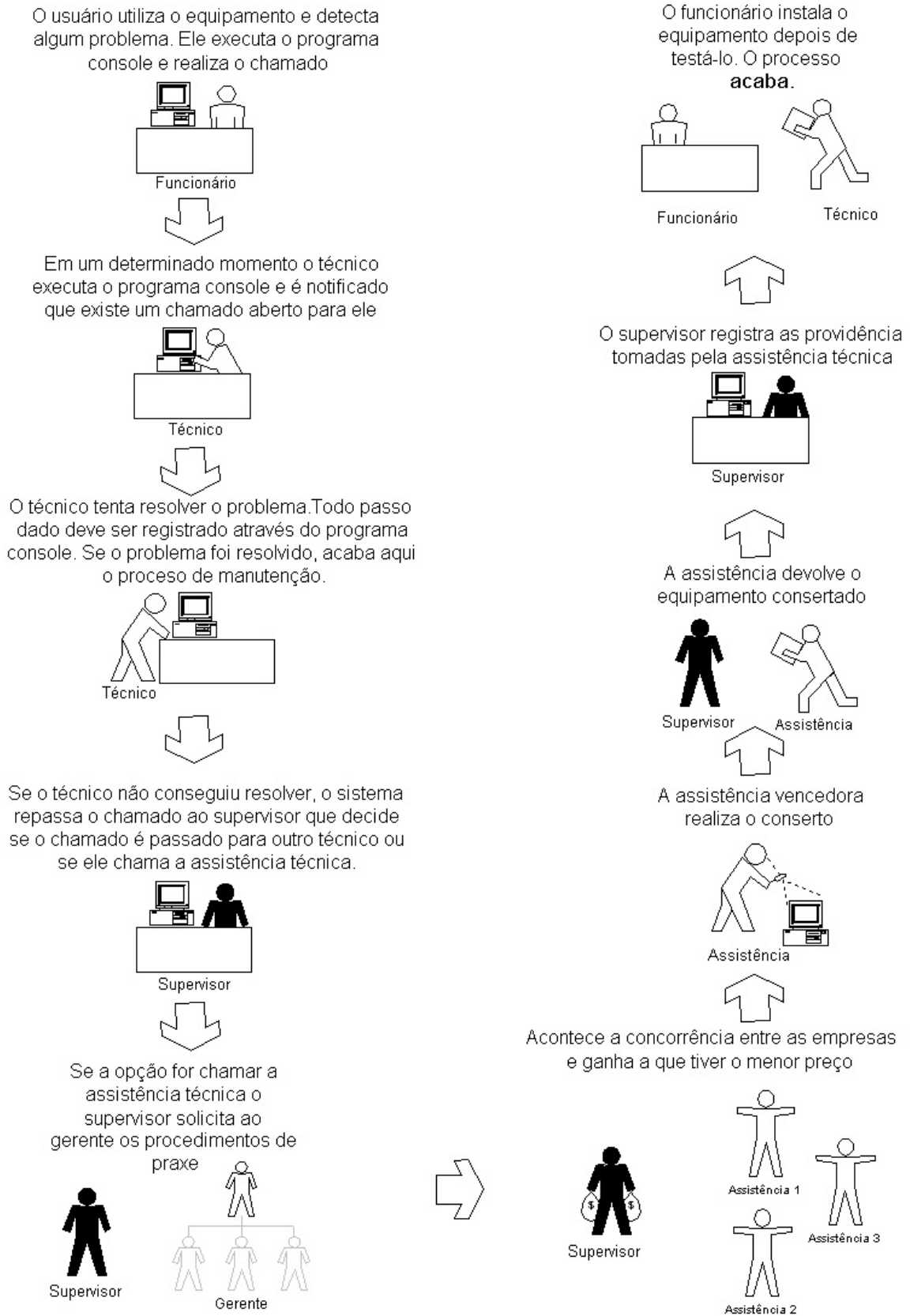


Figura 4.1 Representação do processo informatizado de abertura de chamado técnico

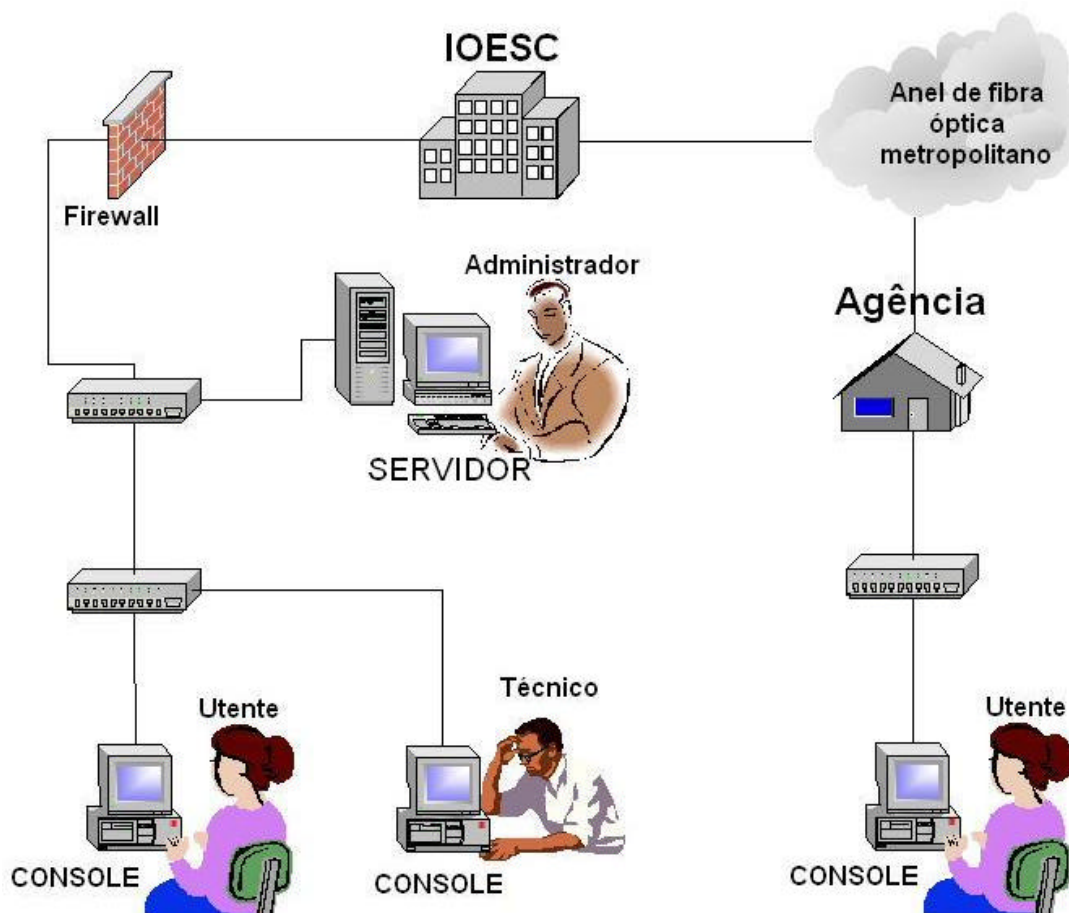


Figura 4.2 Aspecto da fragmentação do sistema

4.2 Uso de Web Service

Foram considerados alguns fatores que apontaram para o uso de Web Service ao invés de outra tecnologia neste trabalho dedicado à IOESC. O primeiro fator avaliado foi o uso de padrões abertos e amplamente aceitos que esta tecnologia envolve, sendo que nenhum investimento financeiro é necessário para desenvolver o sistema. Outro fator considerado é relativo a uma característica inerente a esta tecnologia: transparência ante firewalls, assumindo-se que o protocolo de comunicação entre as aplicações seja o protocolo SOAP, que é uma string XML transladada pela porta 80. Este recurso é útil tendo em vista a necessidade de comunicar a aplicação console na agência com a aplicação servidora na sede.

E, por fim, o Web Service poderá proporcionar, no futuro, uma integração mais estreita com os fornecedores de assistência técnica externa, através de uma interface exclusiva de funções que essas empresas poderão exercer no sistema. Ou também disponibilizar funções que produzam relatórios com informações táticas e técnicas convenientes para determinados departamentos do Governo do Estado interessados. Desta forma, essas instituições poderão utilizar seus sistemas legados e conectá-los ao Web Service da IOESC sem que qualquer migração de sistema ou atualização de versão cause problema de conectividade.

5 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do sistema foi elaborado utilizando-se a metodologia de UML. O acrônimo UML é uma abreviação de Linguagem de Modelagem Unificada, que na prática representa uma metodologia de notações para modelagem de sistemas orientados a objetos.

Um sistema precisa, antes de ser implementado, de uma análise detalhada dos requisitos do problema. Esta verdadeira investigação aplicada através de técnicas específicas ajuda a buscar e descobrir novos conceitos que auxiliarão os projetistas a determinarem soluções para a construção de um sistema mais robusto e reutilizável. Em resumo, consiste em avaliar o domínio do problema e, a partir daí, apresentar uma solução lógica sob o aspecto de objetos (coisas, conceitos ou entidades).

Neste trabalho, o desenvolvimento da análise se dá somente sob o domínio do problema. Ou seja, a análise do sistema levará em conta exclusivamente o núcleo do sistema que determina as principais atividades envolvidas e retrai as atividades periféricas e de sustentáculo.

Na avaliação do problema são dados três passos de nível macro: planejar e elaborar (onde são feitos os levantamentos dos requisitos e construções de protótipos), construir (construção prática do sistema) e instalar (implantar para o uso).

5.1 Fase Planejar e Elaborar

Um dos primeiros passos para a construção do sistema é a análise do problema e com isto derivarem-se conceitos significativos ao projeto. Esta análise preliminar é chamada de Fase Planejar e Elaborar. Durante esta fase, dois importantes processos são executados com o objetivo de fornecer a concepção inicial do sistema: definição dos requisitos e definição dos casos de uso. Os requisitos consistem em expor, declarativamente, as condições necessárias para atingir-se o objetivo maior, que é o sistema. Os casos de uso são descrições em prosa de determinados processos do domínio.

Neste trabalho, esta fase foi concebida conforme a seguir:

5.1.1 Definição dos requisitos

- Um texto com a visão geral

A finalidade deste projeto é produzir um sistema capaz de controlar a abertura de chamado técnico de reparos de equipamentos de informática para uma empresa.

- Cliente

Imprensa Oficial do Estado de Santa Catarina – IOESC –, empresa pública que tem por objetivo divulgar os atos oficiais do Estado de Santa Catarina.

- Objetivo

O objetivo principal é formalizar e automatizar o processo de abertura de chamados técnicos de reparo em equipamentos de informática, a fim de proporcionar agilidade e controle sobre os reparos. Mais especificamente, os objetivos são:

1. registrar pedidos de reparos;
2. formar um histórico;
3. atribuir trabalho às pessoas responsáveis;
4. acompanhar solicitação de serviços (avaliar processo) até a conclusão.

- Funções do sistema

Funções básicas:

Ref#	Função	Categoria
R 1.1	Cadastrar usuários (administrador, técnico e utente).	Evidente
R 1.2	Cadastrar dispositivos (equipamentos, computadores e periféricos).	Evidente
R 1.3	Cadastrar empresa de Assistência Técnica	Evidente
R 1.4	Efetuar autenticação de usuários no sistema.	Evidente
R 1.5	Registrar um pedido de reparo/manutenção (ticket).	Evidente

Ref#	Função	Categoria
R 1.6	Disponibilizar acesso aos tickets conforme autorização de cada tipo de usuário.	Oculto
R 1.7	Atribuir responsabilidade de reparo.	Oculto
R 1.8	Registrar os procedimentos de reparo até a conclusão.	Evidente
R 1.9	Registrar procedimentos de conserto de Assistência Técnica.	Evidente
R 1.10	Prover mecanismo de armazenamento persistente.	Oculto

Funções de registro de abertura de chamado:

Ref#	Função	Categoria
R 2.1	Registrar dados sobre o problema.	Evidente
R 2.2	Registrar dados sobre o dispositivo com problema.	Evidente

Funções de solucionar chamado:

Ref#	Função	Categoria
R 3.1	Registrar os passos dados pelo técnico para a solução do problema.	Evidente

Funções de administrar:

Ref#	Função	Categoria
R 4.1	Fazer um chamado externo à uma assistência técnica.	Evidente
R 4.2	Registrar os passos dados pela assistência técnica quando ela for chamada.	Evidente
R 4.3	Incluir, excluir usuários.	Evidente

O campo “Categoria” das tabelas possui dois tipos de valores: “Evidente”, que significa se a função pode ser percebida pelo usuário, ou “Oculto”, que significa que o usuário não percebe quando ela é executada.

5.1.2 Definição dos casos de uso

Os casos de uso são descrições narrativas de processos do domínio do problema. Neste tipo de casos de uso apresentados a seguir a descrição é feita em alto nível, ou seja, estão livres de tecnologias específicas. Eles servem apenas para dar uma idéia dos processos envolvidos. A construção de casos de uso está diretamente ligada a uma prévia descrição de requisitos e dependente desta , como a apresentada anteriormente.

Primeiramente, é necessária a definição de quais são os atores que iniciam processos e seus respectivos processos. No caso deste sistema, existem três atores: Utente, Técnico e Administrador. É importante salientar que existe uma hierarquia entre os usuários. O administrador pode executar quaisquer funções de Utente e Técnico. E Técnico pode executar quaisquer funções de Utente.

Cada caso de uso possui um nome, uma lista de atores relacionados, uma descrição do caso e uma classificação quanto à prioridade. Esta prioridade é uma classificação quanto ao grau de envolvimento do caso de uso com o contexto principal. A prioridade pode ser primária ou secundária.

A lista dos atores que iniciam processos e os respectivos processos é:

1. Utente:
 - a. efetuar autenticação;
 - b. abrir chamado.
2. Técnico:
 - a. tratar chamado;
 - b. cadastrar dispositivo.
3. Administrador:
 - a. tratar assistência;
 - b. cadastrar assistência;
 - c. cadastrar usuário.

Os casos de uso de alto nível são:

Caso de uso: Efetuar autenticação.

Ator: Utente ou Técnico ou Administrador.

Tipo: Primário.

Descrição: O usuário necessita identificar-se no sistema para efetuar alguma operação.

Caso de uso: Cadastrar dispositivo.

Ator: Utente ou Técnico ou Administrador.

Tipo: Secundário.

Descrição: O usuário efetua o cadastro de um equipamento ou computador ou periférico que ainda não foi cadastrado no sistema.

Caso de uso: Abrir chamado.

Ator: Utente ou Técnico ou Administrador.

Tipo: Primário.

Descrição: O usuário descreve um problema e informa em qual dispositivo o problema ocorreu e encaminha este chamado (ticket) para que ele seja atendido.

Caso de uso: Tratar chamado.

Ator: Técnico ou Administrador.

Tipo: Primário.

Descrição: O Técnico analisa o chamado (ticket) que lhe é delegado e toma as providências para solucionar o problema. Cada passo dado é devidamente registrado. Depois do problema resolvido, ele fecha o chamado (ticket) e os dados passam a fazer parte do histórico. Caso o Técnico julgue que não tem condições de resolver o problema, ele mantém o chamado aberto e informa que o problema não pode ser resolvido por ele, necessitando da assistência técnica.

Caso de uso: Cadastrar assistência.

Ator: Administrador.

Tipo: Secundário.

Descrição: O Administrador efetua o cadastro de alguma empresa de assistência técnica que ainda não foi cadastrado no sistema.

Caso de uso: Tratar assistência.

Ator: Administrador.

Tipo: Primário.

Descrição: O Administrador registra no chamado (ticket) as medidas tomadas pela assistência para a solução do problema.

Caso de uso: Cadastrar usuários.

Ator: Administrador.

Tipo: Secundário.

Descrição: O Administrador cadastra os usuários que ainda não fazem parte do sistema, devendo determinar que tipo de usuário cada um representa (Administrador, Técnico ou Utente).

Um Diagrama de caso de uso pode simplificar a recapitulação das atividades de cada ator. A seguir, o diagrama de casos de uso para este trabalho:

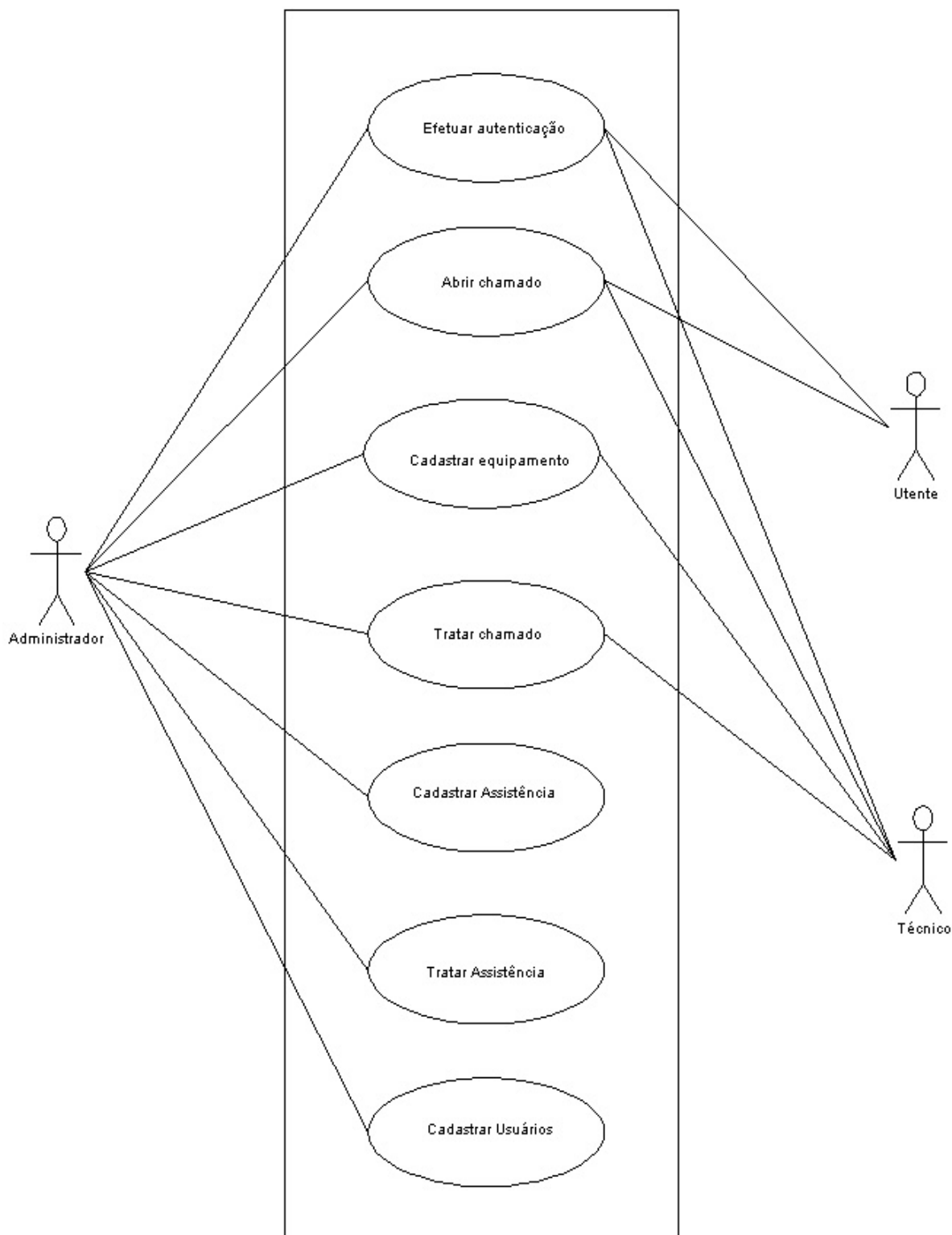


Figura 5.1 Diagrama de casos de uso

O principal objetivo de listar casos de uso de alto nível é fornecer descompromisso de detalhamento na análise dos processos do sistema. Agora que neste trabalho os casos de uso de alto nível já foram identificados é necessário

ampliar-se o entendimento dos processos por meio de casos de uso essenciais e expandidos. O termo essencial é acrescentado porque é expresso de uma forma ideal, e o termo expandido porque descreve o processo mais detalhadamente.

Os casos de uso expandidos são:

1. Efetuar autenticação

Caso de uso: Efetuar autenticação.

Ator: Utente ou Técnico ou Administrador.

Finalidade: Autorizar o usuário a acessar o sistema com os privilégios devidos.

Visão geral: Um usuário se dirige ao PAS (acrônimo de Ponto de Acesso ao Suporte, ou seja, local onde o usuário interage com o sistema) solicitando a entrada. O sistema autoriza a entrada e disponibiliza os devidos acessos.

Seqüência típica de Eventos:

Ação do ator	Resposta do Sistema
1. Este caso de uso começa quando um usuário deseja acessar o sistema.	
2. Ele se dirige ao PAS (Ponto de Acesso ao Suporte) mais próximo e solicita um acesso, enviando login e senha.	3. O sistema efetua a autenticação do usuário.
4. O usuário realiza as operações desejadas.	5. Sistema responde às operações.
6. O usuário solicita a saída.	7. O sistema concede a saída.

Seqüências alternativas:

Linha 3: O usuário não possui cadastro. Indicar erro.

2. Abrir chamado

Caso de uso: Abrir chamado.

Ator: Utente ou Técnico ou Administrador.

Finalidade: Abrir um chamado para reparo de um computador ou equipamento.

Visão geral: Um usuário percebe um problema com algum dispositivo. Para que ele seja atendido, ele acessa o sistema para registrar um pedido de chamado técnico de reparo. No término deste processo, o pedido fica em aberto, esperando que o problema seja solucionado.

Tipo: Primário e essencial.

Seqüência típica de Eventos:

Ação do ator	Resposta do Sistema
1. Este caso de uso começa quando um usuário percebe algum problema em algum dispositivo.	
2. O usuário acessa o sistema e solicita a abertura de um chamado.	3. O sistema provê um formulário contendo os campos para a abertura do chamado.
4. O usuário preenche o formulário informando, entre outras coisas, o problema e o número do patrimônio do dispositivo.	5. O sistema registra os dados.
	6. O sistema mantém o chamado aberto até que o problema seja solucionado.

Seqüência alternativa:

Linha 4: Se o dispositivo ainda não estiver cadastrado, o sistema deve fornecer meios para cadastrá-lo neste momento.

3. Tratar chamado

Caso de uso: Tratar chamado.

Ator: Técnico ou Administrador.

Finalidade: Registrar os passos dados para a solução do problema.

Visão geral: Um Técnico, ao acessar um sistema, é notificado que existe um chamado reservado a ele. O Técnico observa as informações contidas no

chamado e tenta resolver o problema. No decorrer do processo, o Técnico vai registrando os passos dados até o término que é a solução do problema ou o seu encaminhamento para a assistência técnica.

Tipo: Primário e essencial.

Seqüência típica de Eventos:

Ação do ator	Resposta do Sistema
1. Este caso de uso começa quando um técnico percebe que existe um chamado técnico em aberto para ele.	
2. O Técnico solicita atender o chamado.	3. O sistema retorna o Ticket.
4. O técnico analisa o Ticket.	
5. O Técnico toma providências com o objetivo de resolver o problema e informa os passos dados ao sistema.	6. O sistema registra os passos que o técnico enunciou.
7. Se o Técnico conseguiu resolver o problema, ver seção <i>Encerrar</i> .	
Se o Técnico ainda não conseguiu resolver o problema, ver seção <i>Manter aberto</i> .	
Se o Técnico não pode resolver o problema, ver seção <i>Manter aberto para assistência</i> .	

Seqüência alternativa:

Linha 2: O sistema informa que não existe chamado para ele. Então acaba o caso de uso.

Seção Encerrar:

Ação do ator	Resposta do sistema
1. O Técnico acaba de registrar seu último passo e o problema foi solucionado. Agora ele envia a	2. O sistema registra a informação.

informação que o chamado pode ser fechado.

3. O sistema encerra este chamado (ticket).

Seção Manter aberto:

Ação do ator

1. O Técnico registrou seu último passo possível até o momento e admite que ainda não consegue resolver o problema. Ele solicita que o ticket se mantenha aberto.

Resposta do sistema

2. O sistema registra a informação.

3. O sistema mantém o chamado aberto para o mesmo técnico.

Seção Manter aberto para assistência:

Ação do ator

1. O Técnico registrou seu último passo possível e determina que não tem condições de resolver o problema.

Resposta do sistema

2. O sistema registra a informação.

3. O sistema mantém o chamado aberto para algum Administrador tomar as providências.

4. Cadastrar assistência

Caso de uso: Cadastrar assistência.

Ator: Administrador.

Finalidade: Cadastrar uma empresa de assistência técnica.

Visão geral: O Administrador cadastra alguma empresa de assistência técnica no sistema.

Tipo: Secundário e essencial

Seqüência típica de Eventos:

Ação do ator	Resposta do Sistema
1. Este caso de uso começa quando um usuário deseja cadastrar uma empresa de assistência técnica.	
2. O usuário solicita cadastro de empresa de assistência técnica.	3. O sistema disponibiliza um formulário de cadastro de assistência técnica.
4. O usuário insere os dados da empresa.	5. O sistema registra as informações.

5. Tratar assistência

Caso de uso: Tratar assistência.

Ator: Administrador.

Finalidade: Registrar a avaliação dos serviços prestados e os passos dados pela a assistência técnica para resolver o problema.

Visão Geral: O Administrador, após a assistência solucionar o problema, registra uma avaliação do serviço prestado e os passos relevantes dados pela assistência técnica.

Tipo: Primário.

Seqüência típica de Eventos

Ação do ator	Resposta do Sistema
1. O Administrador relata as informações sobre o que a assistência fez para resolver os problemas deste chamado.	2. O sistema registra as informações.
3. O Administrador expõe uma avaliação sobre o serviço prestado.	4. O sistema registra a avaliação.

6. Cadastrar dispositivo

Caso de uso: Cadastrar dispositivo.

Ator: Utente ou Técnico ou Administrador.

Finalidade: Cadastrar os equipamentos da empresa.

Visão geral: O usuário, ao verificar que o equipamento não está cadastrado, poderá fazer o cadastro do equipamento.

Tipo: Secundário e essencial

Seqüência típica de Eventos:

Ação do ator	Resposta do sistema
1. O usuário solicita cadastro de um dispositivo.	2. O sistema disponibiliza um formulário de cadastro de dispositivo.
3. O usuário preenche os campos.	4. O sistema salva os dados em um meio persistente.

Seqüência alternativa:

Linha 4: O sistema informa que o dispositivo já está cadastrado.

7. Cadastrar usuário

Caso de uso: Cadastrar usuário.

Ator: Administrador.

Finalidade: Adicionar, remover e editar contas de usuários do sistema.

Visão geral: O Administrador cadastra o usuário que ainda não foi cadastrado e define o tipo de usuário.

Tipo: Secundário e essencial.

Seqüência típica de Eventos:

Ação do ator	Resposta do Sistema
1. O Administrador solicita ao sistema a opção de adicionar um usuário.	2. O sistema fornece um formulário para adicionar um usuário novo.
3. O Administrador insere os dados do usuário.	4. O sistema registra o novo usuário.

Seqüência alternativa:

Linha 2: O sistema informa que o usuário já está cadastrado.

5.2 Fase Construir

A Fase Construir de um projeto é cumprida através de ciclos de desenvolvimento onde cada um destes ciclos é desenvolvido relacionando um ou mais casos de usos expandidos. Estes casos de usos são tratados através de atividades de análise e de projeto. Durante a fase de análise ocorrem os procedimentos de descoberta e descrição de conceitos envolvidos no contexto do domínio do problema. Na fase de projeto são definidos os elementos lógicos de software que serão mais tarde implementados. No final dos ciclos, existe a fase de construção onde os elementos são implementados em alguma linguagem de programação orientada a objetos. A figura 5.1 mostra as atividades e os elementos que contribuem para o andamento de cada ciclo.

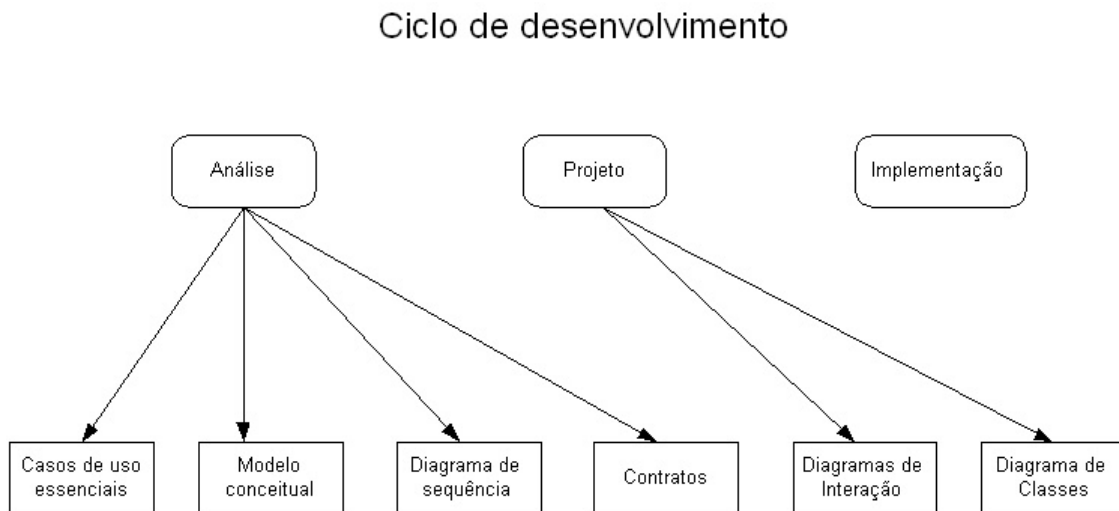


Figura 5.2 Elementos de um ciclo de desenvolvimento

O processo de construção em ciclo iterativo decorre sucessivamente, ou seja, a cada ciclo terminado o produto deste ciclo é acoplado ao que já foi feito e assim o desenvolvimento do sistema é feito gradativamente.

5.2.1 Análise

A fase de análise é assistida por metodologias de desenvolvimento muito úteis para se descobrirem conceitos relevantes no contexto. Até agora, os casos de

uso forneceram os processos do domínio. Nesta fase, o modelo conceitual fornecerá os conceitos e os termos; mais adiante, os diagramas de seqüência fornecerão os eventos e as operações do sistema; e, para finalizar, os contratos determinarão o que as operações devem cumprir. A seguir é descrita cada uma das metodologias aplicadas juntamente com o aproveitamento delas neste trabalho.

Modelo Conceitual

O Modelo Conceitual é um artefato que facilita o entendimento dos conceitos envolvidos no domínio do problema, familiarizando o analista com o vocabulário do sistema. Ele é uma estrutura estática que representa coisas (conceitos) do mundo real envolvidas no problema e que não pode ser confundido com componentes de software.

A procura de conceitos é realizada através de exame de todos os materiais levantados até este ponto da análise, e todos os conceitos são considerados candidatos a entrar no Modelo Conceitual. É interessante salientar que ele não define operações, apenas mostra conceitos, associações entre conceitos e atributos. O Modelo Conceitual a seguir já passou por diversas alterações ao longo do desenvolvimento e os atributos colocados nas entidades surgiram durante toda a fase de análise.

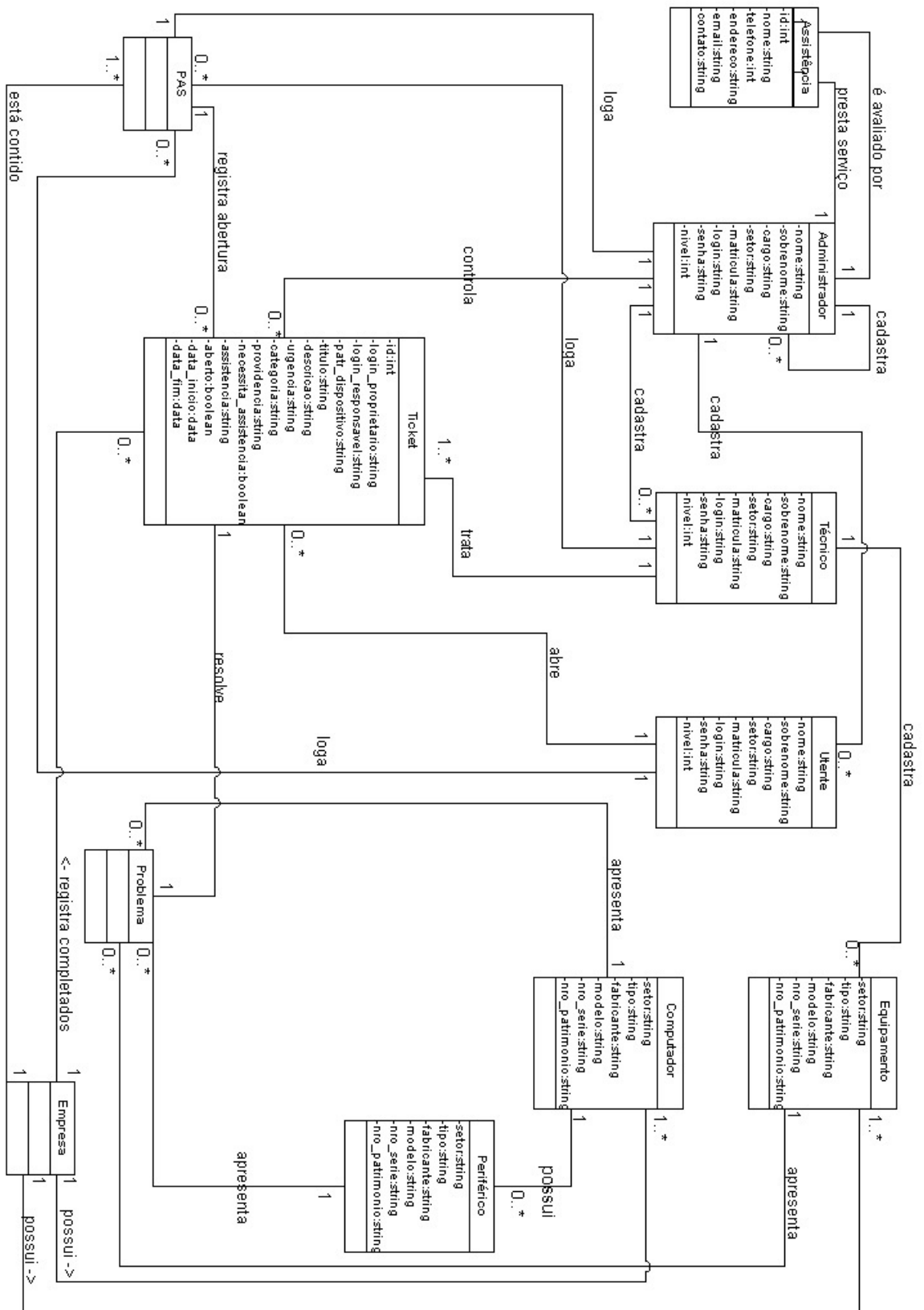


Figura 5.3 Modelo conceitual

Diagrama de Seqüência

Diagramas de seqüência ilustram os eventos descobertos nos casos de uso gerados pelos atores. Cada um destes eventos deve ser reconhecido e respondido pelo sistema, desencadeando vários processos em virtude dele. Ele é parte da investigação do sistema e está diretamente dependente dos casos de uso, porque é através deles que se podem deduzir as ações dos atores. É uma descrição de que o sistema faz ocultando como ele faz.

A seqüência típica de eventos de cada caso de uso sugere a interação dos atores com o sistema naquela determinada ocorrência; portanto, é útil gerar para cada caso de uso um diagrama de seqüência correspondente. Para esclarecer mais os processos, na medida do possível é interessante que se incluam parâmetros nas invocações feitas pelos atores.

A seguir, os diagramas de seqüência para cada caso de uso identificado neste trabalho.

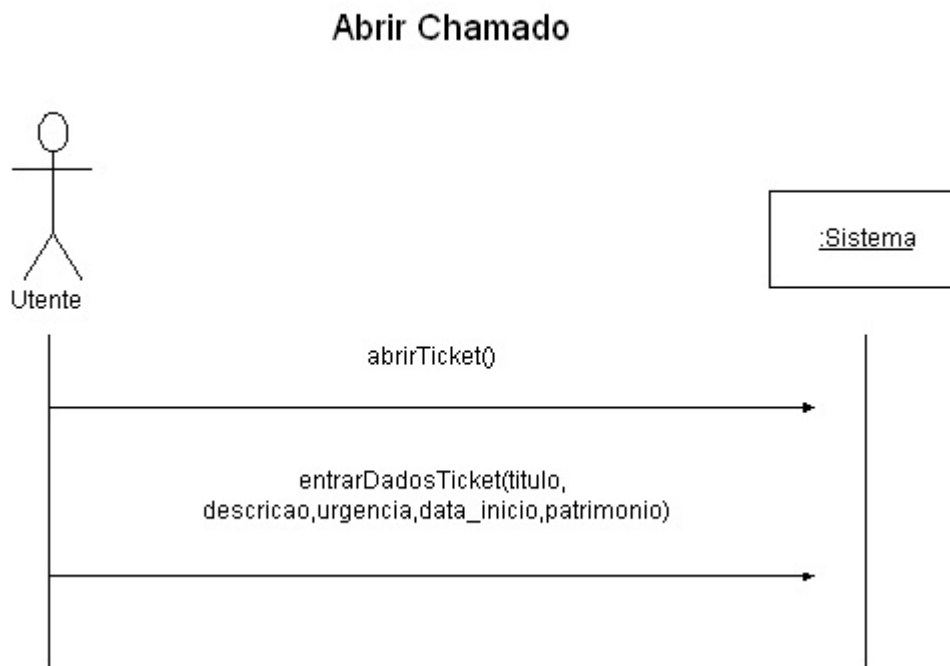


Figura 5.4 Diagrama de seqüência Abrir Chamado

Cadastrar assistência

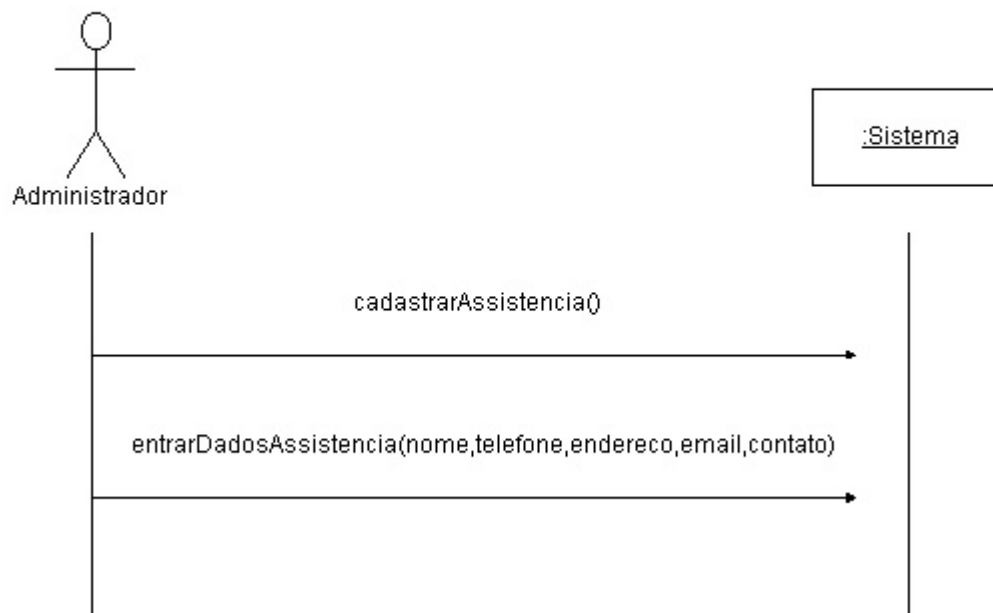


Figura 5.5 Diagrama de seqüência Cadastrar assistência

Cadastrar dispositivo

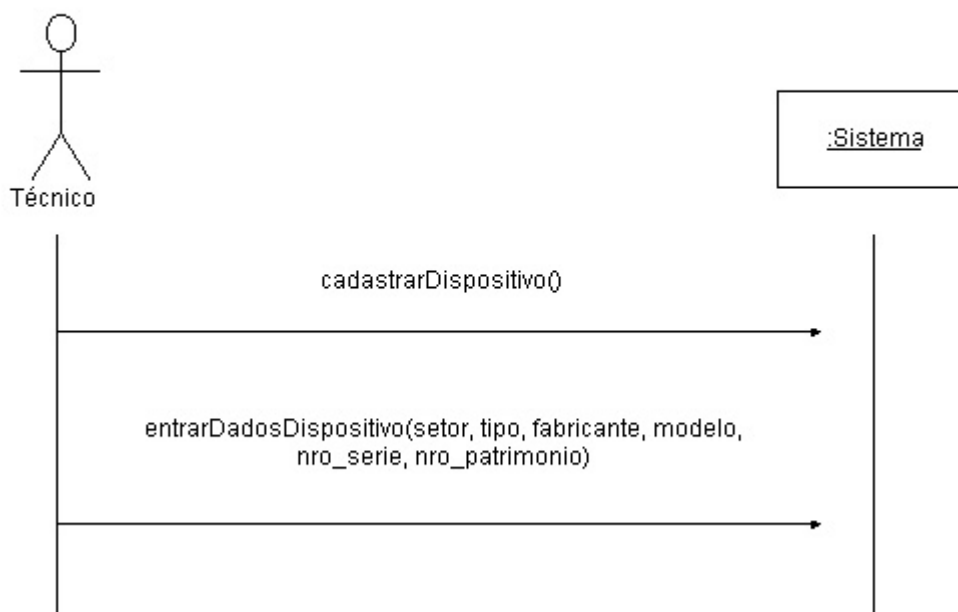


Figura 5.6 Diagrama de seqüência Cadastrar Dispositivo

Cadastrar usuário

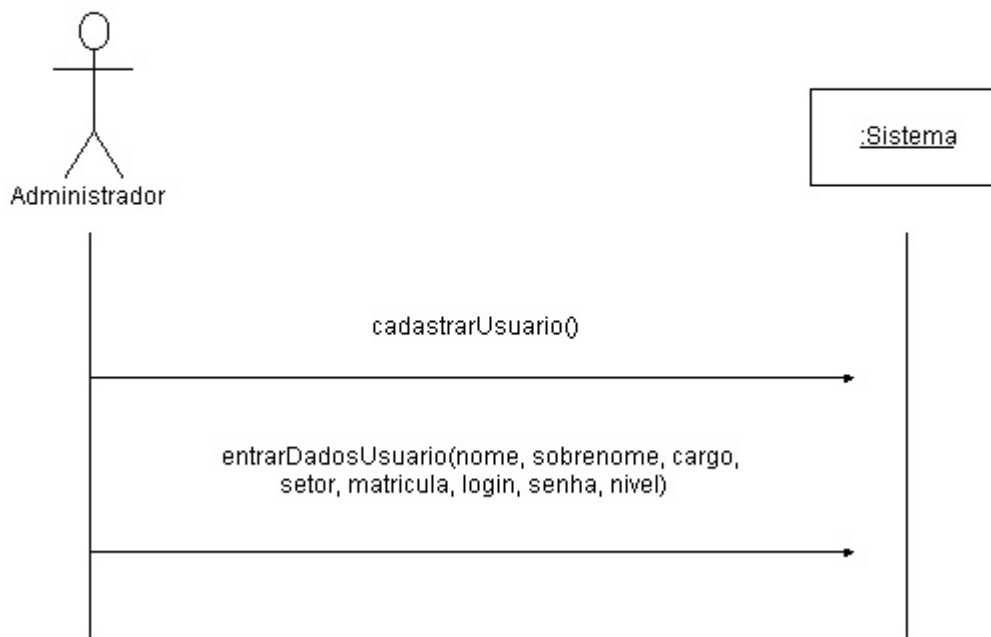


Figura 5.7 Diagrama de seqüência Cadastrar usuário

Efetuar autenticação



Figura 5.8 Diagrama de seqüência Efetuar autenticação

Tratar chamado com seção abandonar

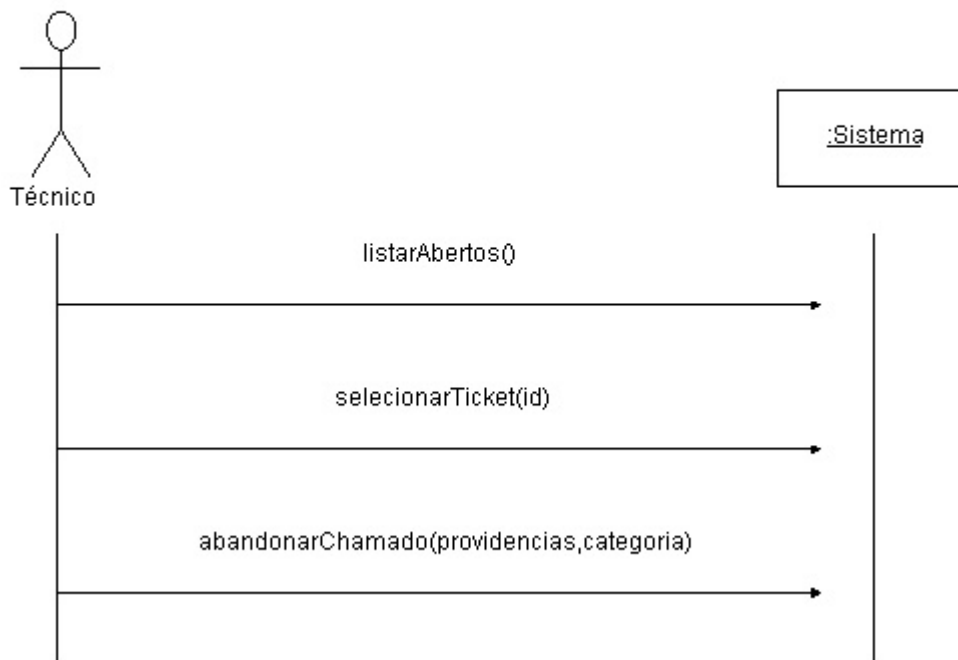


Figura 5.9 Diagrama de seqüência Tratar chamado com seção abandonar

Tratar chamado com seção Manter aberto

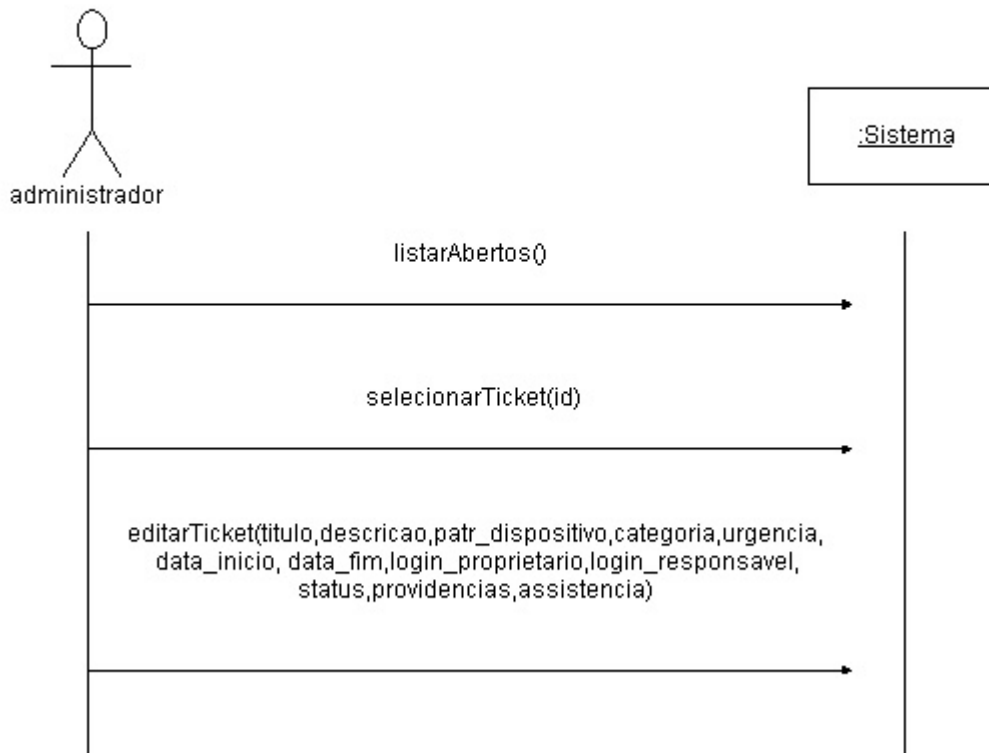


Figura 5.10 Diagrama de seqüência Tratar chamado com seção Manter aberto

Tratar chamado com seção encerrar

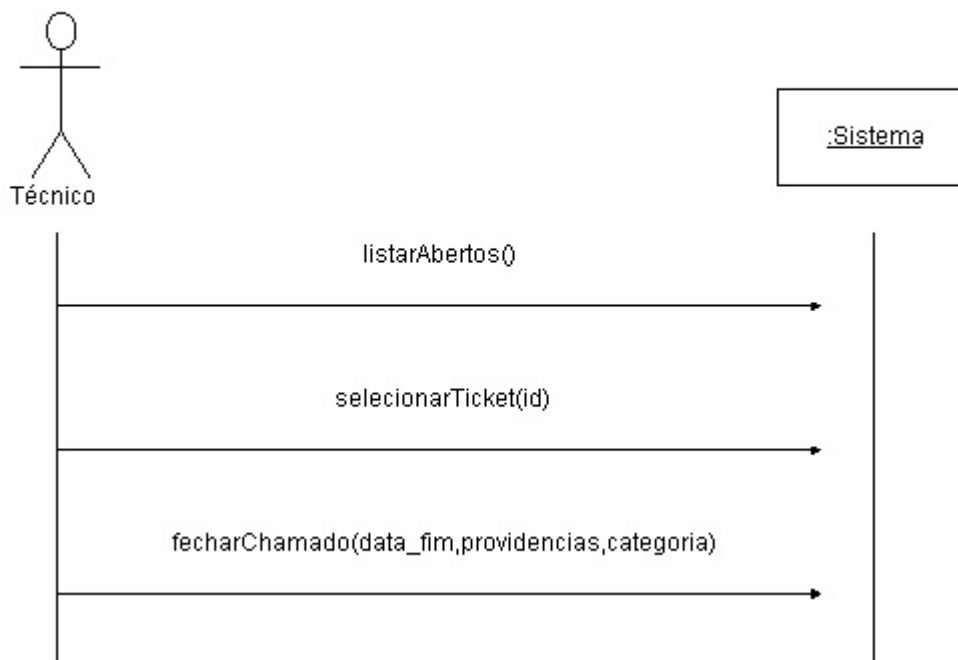


Figura 5.11 Diagrama de seqüência Tratar chamado com seção encerrar

Tratar chamado com seção Manter aberto

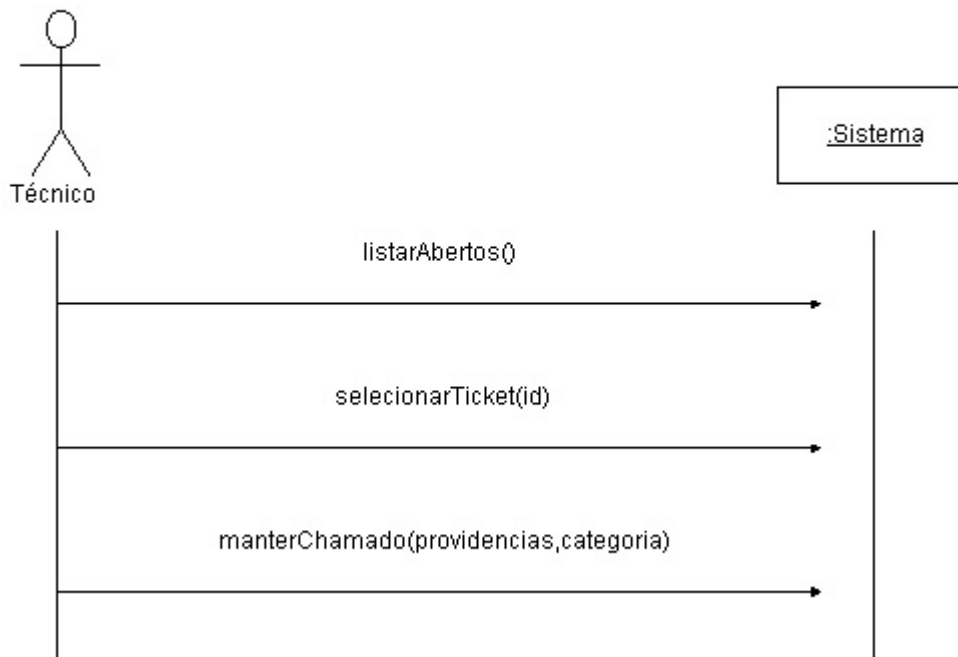


Figura 5.12 Diagrama de seqüência Tratar chamado com seção Manter aberto

Contratos

Os contratos são artifícios utilizados na análise para definição do comportamento do sistema perante as ações estimuladas pelos atores.

A criação de contratos é dependente da criação prévia do modelo conceitual e dos diagramas de seqüência do sistema. No seu desenvolvimento é necessário imaginar o sistema como uma “caixa-preta”, onde apenas é levando em conta “o que o sistema faz” e não “como ele faz”.

Em resumo, os contratos são documentos que descrevem as mudanças de estados do sistema após a invocação de uma operação, mostrando o que a operação se compromete em atingir.

Um contrato é gerado para cada ação contida no diagrama de seqüência onde estão descritos o nome do contrato (mesmo nome das operações do diagrama de seqüência), as responsabilidades (descrição informal do objetivo da operação), as pré-condições (condições necessárias para a execução da operação) e as pós-condições (estado do sistema após a operação terminada, sendo escrita sempre no passado).

A seguir estão disponíveis os contratos escritos para cada operação identificada neste trabalho.

1. abrirChamado(Utente)

Nome: abrirChamado(Utente)

Responsabilidades: Dar início ao processo de abrir o chamado.

Tipo: Sistema.

Pré-condições:

- Utente criado.

Pós-condições:

- um Ticket foi criado;
- uma associação entre Utente e Ticket foi criada.

2. acessarSistema(login,senha)

Nome: acessarSistema (login:string, senha:string).

Responsabilidades: Autorizar a entrada de um usuário no sistema e providenciar

os direitos de acesso.

Tipo: Sistema.

Exceções: Em caso de tentativa de acesso de usuário desconhecido, indicar erro.

Pré-condições:

- empresa iniciada;
- usuários criados;
- dispositivos criados;
- um PAS iniciado;
- usuário cadastrado.

Pós-condições:

- se foi identificado um usuário comum, um Utente foi criado;
- se foi identificado um usuário Técnico, um Técnico foi criado;
- se foi identificado um usuário Administrador, um Administrador foi criado;
- o usuário criado foi associado ao PAS.

3. cadastrarAssistencia(empresa)

Nome: cadastrarAssistencia(empresa)

Responsabilidades: Cadastrar uma empresa de assistência técnica no sistema.

Tipo: Sistema.

Pré-condições:

- um Administrador criado.

Pós -condições:

- uma Assistência foi criada;
- uma associação entre Administrador e Assistência foi criada;
- Assistência.empresa recebeu o valor empresa.

4. cadastrarDispositivo()

Nome: cadastrarDispositivo ()

Responsabilidades: Iniciar o processo de cadastro de um dispositivo.

Tipo: Sistema.

Pré-condições:

- Utente criado.

Pós-condições:

- um dispositivo foi criado;
- uma associação entre Utente e dispositivo foi estabelecida.

5. entrarDadosChamado(nro_patrimonio,titulo,descricao,urgencia,categoria)

Nome: entrarDadosChamado(nro_patrimonio:string, titulo:string, descricao:string, urgencia:urgencia, categoria:categoria)

Responsabilidades: Atribuir os dados para a abertura do chamado.

Tipo: Sistema.

Pré-condições:

- Ticket criado;
- Utente criado;
- associação entre Ticket e Utente realizada.

Pós-condições:

- Ticket.nro_patrimonio recebeu o valor dispositivo;
- Ticket.titulo.recebeu o valor titulo;
- Ticket.descricao recebeu o valor descricao;
- Ticket.urgencia recebeu o valor urgencia;
- Ticket.categoria recebeu o valor categoria;

6. entrarDadosDispositivo(setor, tipo, fabricante, modelo, nro_serie, nro_patrimonio, data_aquisicao, tempo_garantia, termo_garantia, fornecedor)

Nome: entrarDadosDispositivo(setor: string, tipo: tipo, fabricante: pessoa_juridica, modelo: string, nro_serie: string, nro_patrimonio: string, data_aquisicao: data, tempo_garantia: int, termo_garantia: string, fornecedor: pessoa_juridica)

Responsabilidades: Estabelecer os valores dos atributos de um dispositivo.

Tipo: Sistema.

Pré-condições:

- um Técnico foi criado;
- um Dispositivo foi criado;
- uma associação entre o Dispositivo e Técnico deve ter sido estabelecida.

Pós-condições:

- Dispositivo.setor recebeu o valor setor;
- Dispositivo.tipo recebeu o valor tipo;
- Dispositivo.fabricante recebeu o valor fabricante;
- Dispositivo.modelo recebeu o valor modelo;
- Dispositivo.nro_serie recebeu o valor nro_serie;
- Dispositivo.nro_patrimonio recebeu o valor nro_patrimonio;
- Dispositivo.data_aquisicao recebeu o valor data_aquisicao;
- Dispositivo.tempo_garantia recebeu o valor tempo_garantia;
- Dispositivo.termo_garantia recebeu o valor termo_garantia;
- Dispositivo.fornecedor recebeu o valor fornecedor.

7. excluirUsuario(nome)

Nome: excluirUsuario(nome)

Responsabilidades: Excluir um usuário da lista de usuários do sistema.

Tipo: Sistema.

Pré-condições:

- administrador criado.

Pós-condições:

- um Usuário foi criado;
- uma associação entre Usuário e Administrador foi criada.

8. fecharChamado(false)

Nome: fecharChamado(false).

Responsabilidades: Manter o chamado aberto.

Tipo: Sistema.

Pré-condições:

- Técnico criado;
- Ticket criado;
- associação entre Técnico e Ticket deve ter sido criada;
- pelo menos um valor de Ticket.providencias foi acrescentado.

Pós-condições:

- Ticket.fechado recebeu o valor false.

9. fecharChamado(true)

Nome: fecharChamado(false).

Responsabilidades: Manter o chamado aberto.

Tipo: Sistema.

Pré-condições:

- Técnico criado;
- Ticket criado;
- associação entre Técnico e Ticket deve ter sido criada;
- pelo menos um valor de Ticket.providencias foi acrescentado.

Pós-condições:

- Ticket.fechado recebeu o valor true.

10.inserirUsuario(nome, cargo, setor, matricula, login, senha)

Nome: inserirUsuario(nome: string, cargo: string, setor: string, matricula: string, login: string, senha: string)

Responsabilidades: Inserir um novo usuário no sistema.

Tipo: Sistema.

Pré-condições:

- Administrador criado.

Pós-condições:

- se era desejado um Utente para inserir, um Utente foi criado;
- se era desejado um Técnico para inserir, um Técnico foi criado;
- se era desejado um Administrador para inserir, um Administrador foi criado;
- uma associação entre Administrador e o novo Usuário foi criada;
- Usuário.nome recebeu o valor nome;
- Usuário.cargo recebeu o valor cargo;
- Usuário.setor recebeu o valor setor;
- Usuário.matricula recebeu o valor matricula;
- Usuário.login recebeu o valor login;
- Usuário.senha recebeu o valor senha.

11.registrarAvaliacao (avaliacao)

Nome: registrarAvaliacao(avaliacao).

Responsabilidades: Registrar a avaliação dos serviços prestados pela assistência técnica.

Tipo: Sistema.

Pré-condições:

- Administrador criado;
- Ticket criado;
- associação entre Ticket e Administrador realizada.

Pós-condições:

- Assistência foi criada;
- associação entre Administrador e Assistência foi criada;
- Assistência.avaliacao recebeu valor avaliacao.

12.registrarPasso(id,passo)

Nome: registrarPasso(id,passo).

Responsabilidades: Registrar os passos que o Técnico deu no decorrer do atendimento de um chamado (Ticket).

Tipo: Sistema.

Pré-condições:

- um Técnico criado;
- um Ticket criado.

Pós-condições:

- Ticket.providencias recebeu o valor passo.

13. sairSistema()

Nome: sairSistema()

Responsabilidades: Efetuar a saída do usuário do sistema.

Tipo: Sistema.

Pré-condições:

- Usuário foi criado.

Pós-condições:

- Todos objetos destruídos.

14. solicitarAssistencia(true)

Nome: solicitarAssistencia(true)

Responsabilidades: Firmar o chamado para que ele seja encaminhado à assistência.

Tipo: sistema.

Pré-condições:

- Técnico criado;
- Ticket criado;
- associação entre Ticket e Técnico estabelecida.

Pós-condições:

- Ticket.assistencia recebeu o valor true.

5.2.2 Projeto

Durante o ciclo de desenvolvimento só é possível passar para a fase de projeto depois da fase de análise estar completada. Daqui para frente todas as soluções tomadas serão consumidas através da implementação em uma linguagem de programação orientada a objetos.

O principal artefato desta fase é a criação dos diagramas de interação que ilustram como os objetos descobertos interagem para chegar à solução proposta nos contratos de operação, que por sua vez atendem aos requisitos. Ao término da construção destes diagramas, o próximo passo é a criação do diagrama de classes de projeto que resume as classes contidas no projeto e que serão implementadas em software.

A seguir é possível observar as soluções de projeto aplicadas neste trabalho.

Diagrama de Interação

Os contratos de operações, na fase de análise, apenas mostram o que deve resultar da interação entre os objetos. Agora os diagramas de interação ilustrarão estas interações através de mensagens entre objetos, utilizando como inspiração as pós-condições contidas nos contratos. Um diagrama deverá ser criado para cada operação descrita nos contratos.

Os diagramas de interação podem ser expressos através de dois tipos de diagramas: diagramas de colaboração e diagramas de seqüência (parecidos com aqueles utilizados no trabalho anteriormente). Neste trabalho será utilizado o diagrama de colaboração por ele ser tecnicamente mais expressivo e por ser capaz de traduzir informações para objetos do diagrama de classe com mais facilidade.

Na fase de criação do diagrama de colaboração (o tipo de diagrama de interação utilizado neste trabalho), serão atribuídas as responsabilidades de cada objeto que pertence ao sistema. Estas responsabilidades são funções que cada um deve exercer para realizar determinada operação. Porém, a distribuição destas responsabilidades não é feita a esmo, mas utilizando-se vários padrões, tais como os GRASP.

Um padrão é um par problema/solução que deve ser considerado toda vez que será atribuída uma responsabilidade de executar determinada tarefa por um objeto. Os padrões GRASP são padrões estabelecidos por desenvolvedores experientes em programação orientada a objetos para fornecer um repertório de princípios e soluções, norteando a construção de software. GRASP significa General

Responsibility Assignment Software Patterns, ou seja, necessidade de compreender princípios para projetar softwares com sucesso. Os cinco principais padrões GRASP de distribuição de responsabilidade são:

- Expert (Especialista): atribuir determinada responsabilidade a quem é o especialista da informação. Em outras palavras, aquele que tem melhor acesso à informação para executar determinado assunto deve fazê-lo.
- Creator (Criador): define qual melhor objeto para ser responsável pela criação de alguma instância de alguma classe.
- High Cohesion (Coesão Alta): relativo à preocupação de estabelecer funções a uma classe que não fuja do objetivo central. Ou seja, fazer com que a classe faça menos coisas não relacionadas com a responsabilidade desta determinada classe.
- Low Coupling (Acoplamento Fraco): estabelecer um equilíbrio de conexão de uma classe com as outras classes. Ou seja, o quanto uma classe é dependente das outras. Muita dependência não é desejável, pois para exercer-se qualquer modificação pode ser um trabalho muito complexo.
- Controller (Controlador): relativo à definição da responsabilidade de tratar as mensagens de evento do sistema a uma classe. Ou seja, aquelas mensagens originadas de fora do sistema.

Utilizando estes padrões, os diagramas de seqüência e as pós-condições dos contratos, os diagramas de colaboração foram os seguintes:

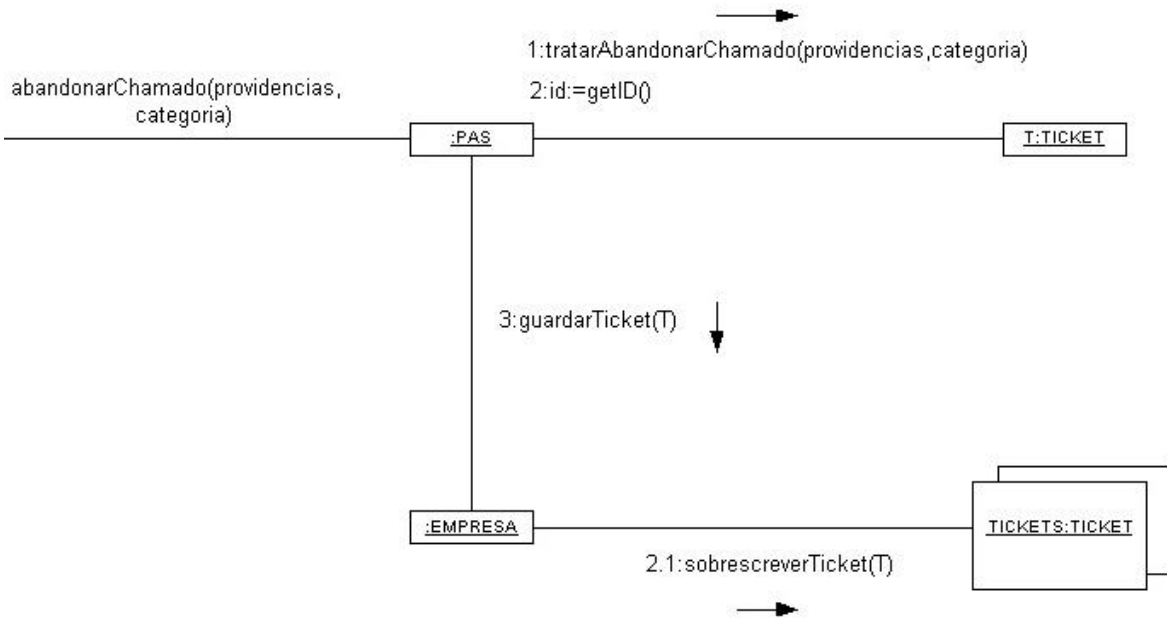


Figura 5.13 Diagrama de colaboração para a operação abandonarChamado

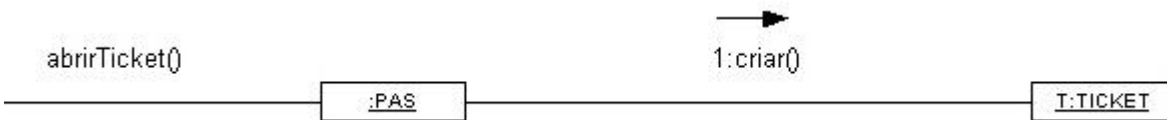


Figura 5.14 Diagrama de colaboração para a operação abrirTicket

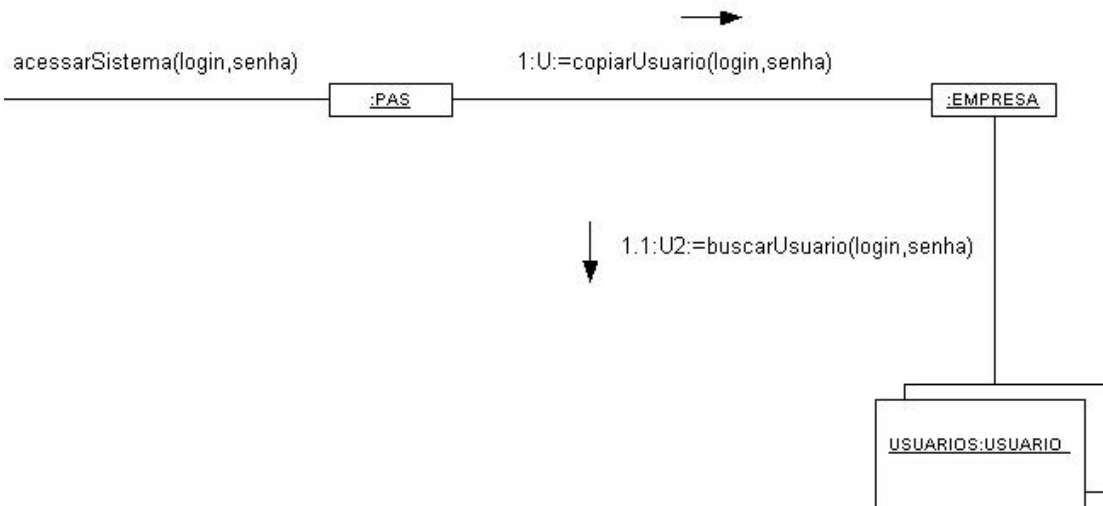


Figura 5.15 Diagrama de colaboração para a operação acessarSistema



Figura 5.16 Diagrama de colaboração para a operação cadastrarAssistencia



Figura 5.17 Diagrama de colaboração para a operação `cadastrarDispositivo`



Figura 5.18 Diagrama de colaboração para a operação `cadastrarUsuario`

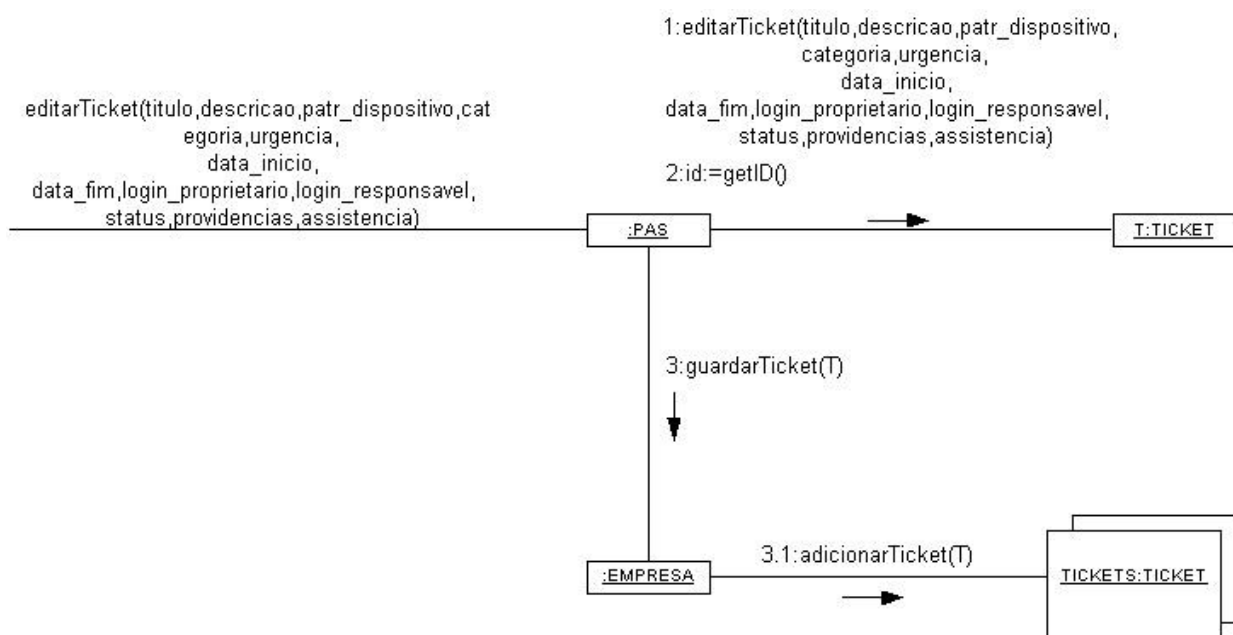


Figura 5.19 Diagrama de colaboração para a operação `editarTicket`

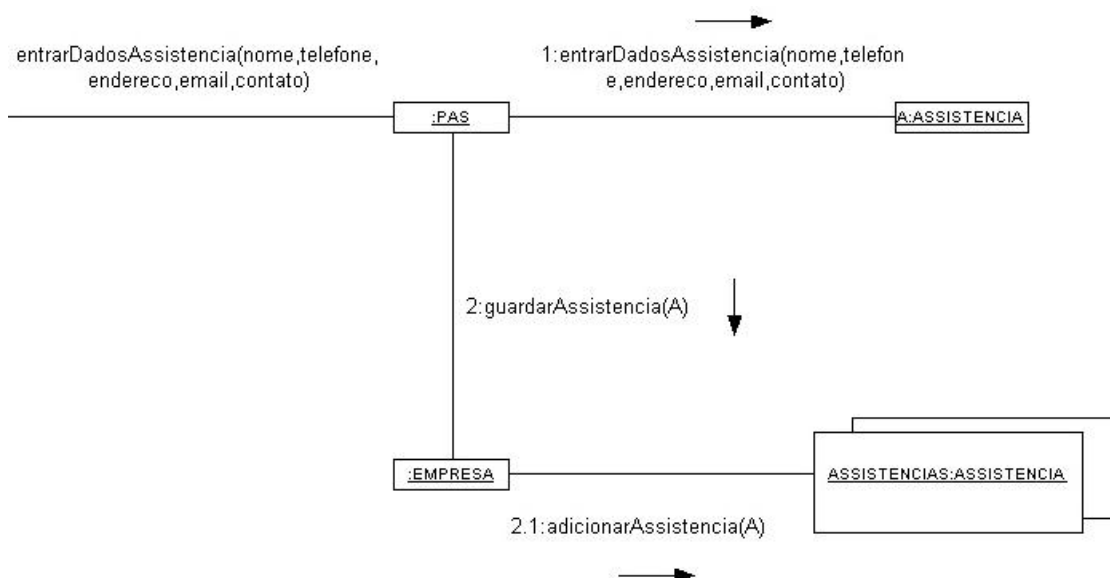


Figura 5.20 Diagrama de colaboração para a operação `entrarDadosAssistencia`

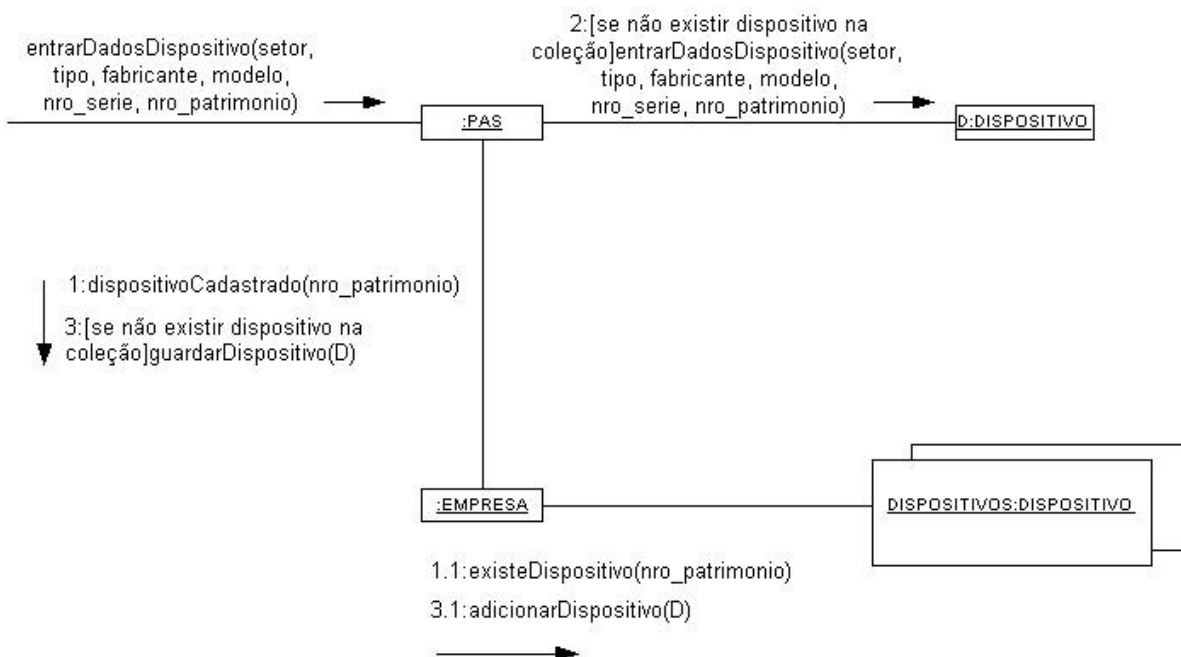


Figura 5.21 Diagrama de colaboração para a operação `entrarDadosDispositivo`

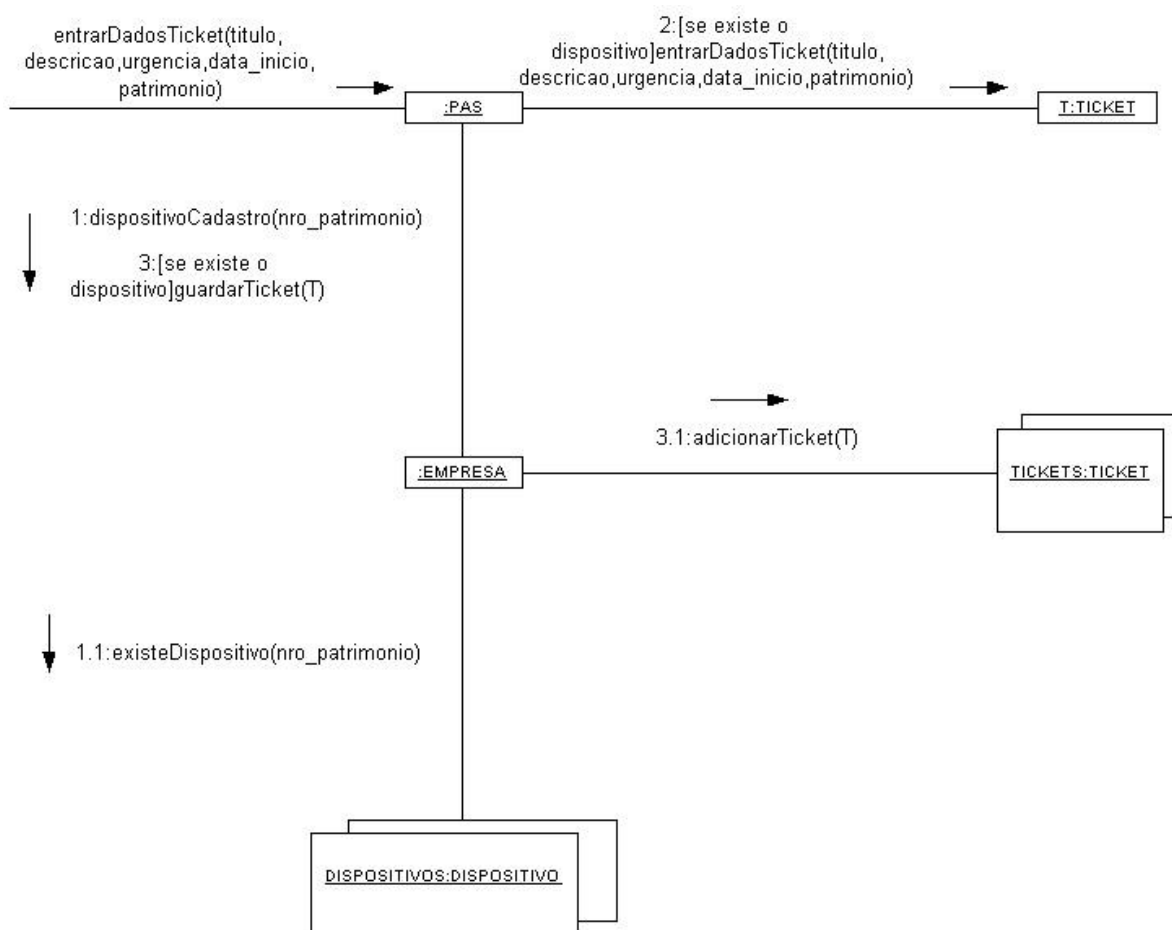


Figura 5.22 Diagrama de colaboração para a operação `entrarDadosTicket`

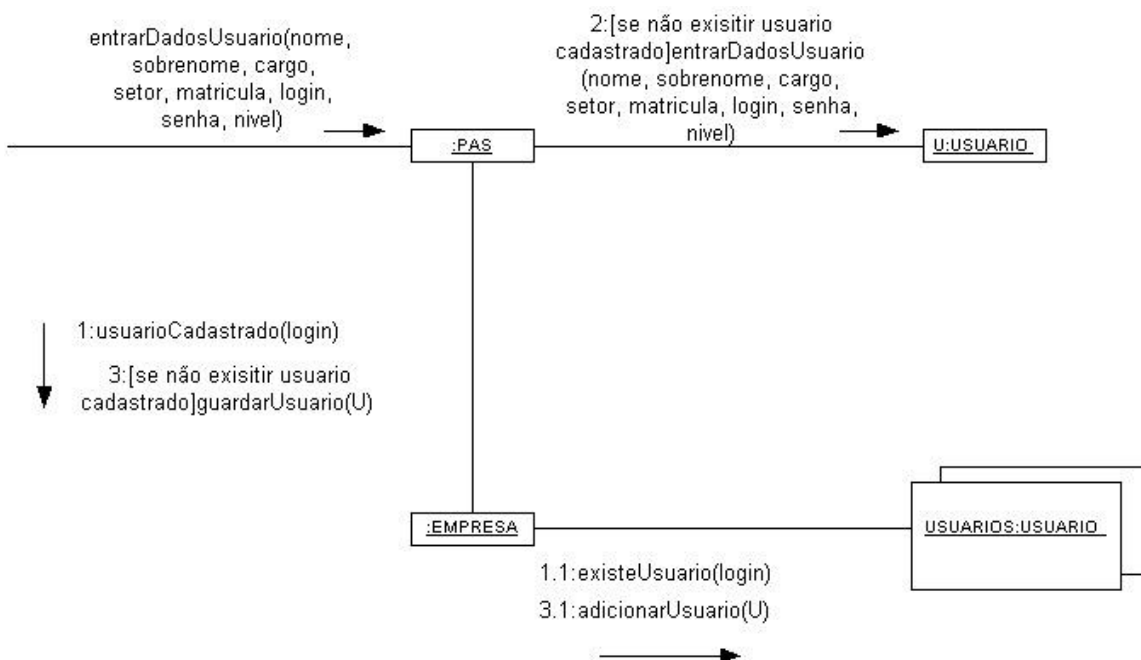


Figura 5.23 Diagrama de colaboração para a operação `entrarDadosUsuario`

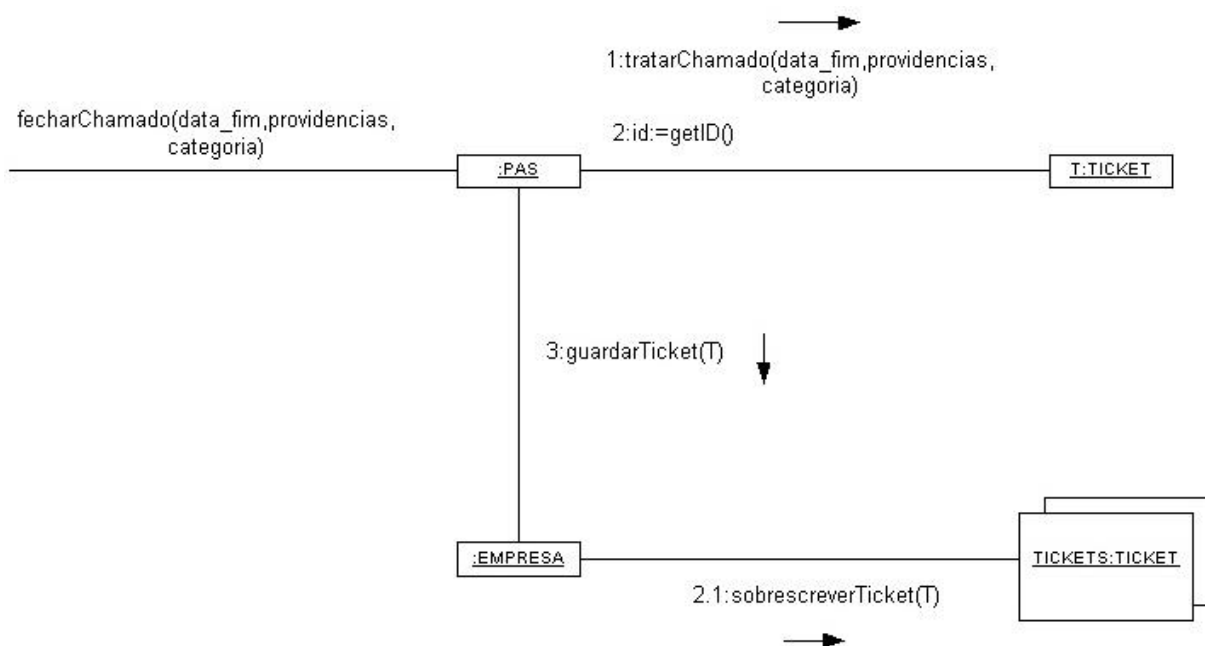


Figura 5.24 Diagrama de colaboração para a operação `fecharChamado`

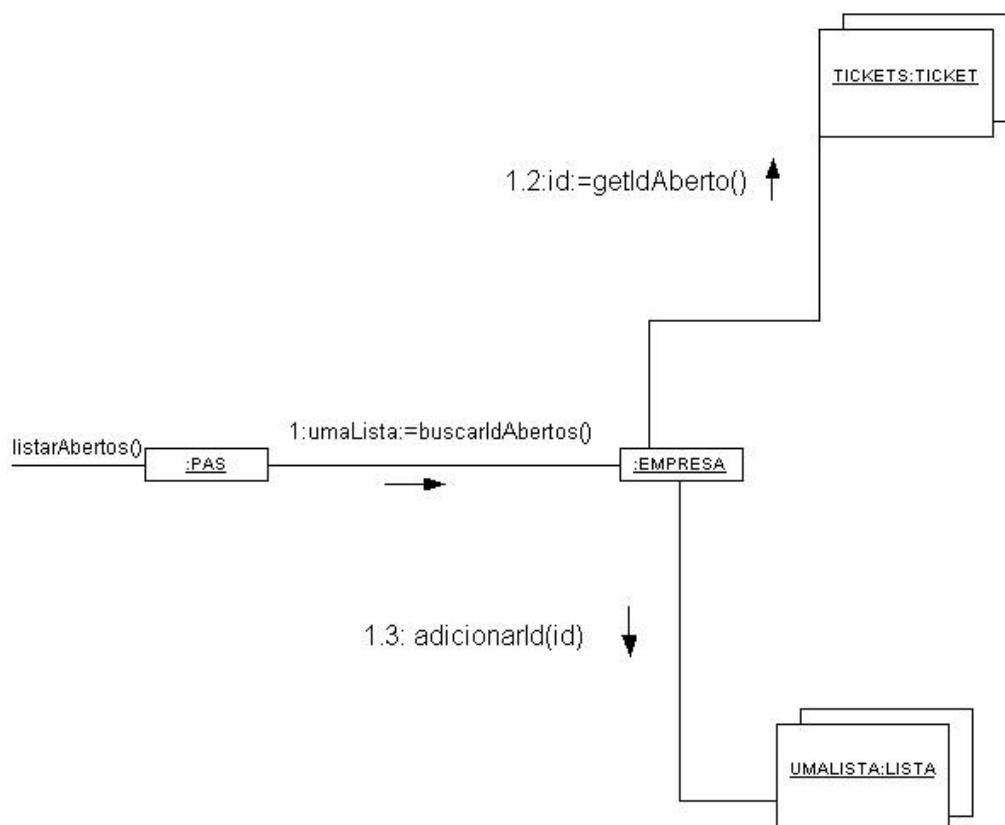


Figura 5.25 Diagrama de colaboração para a operação listarAbertos

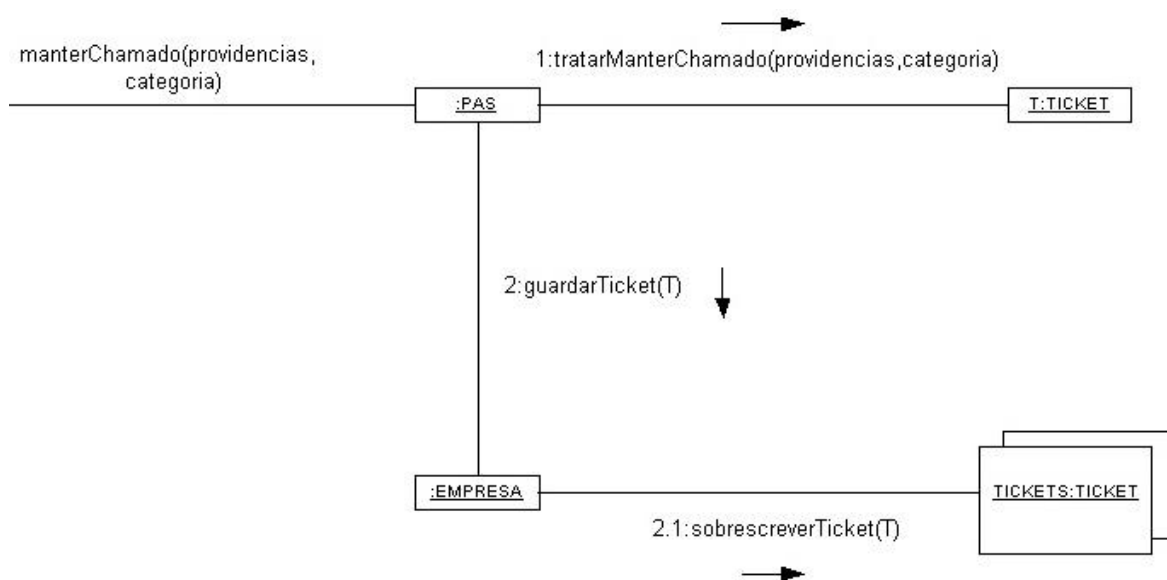


Figura 5.26 Diagrama de colaboração para a operação manterChamado

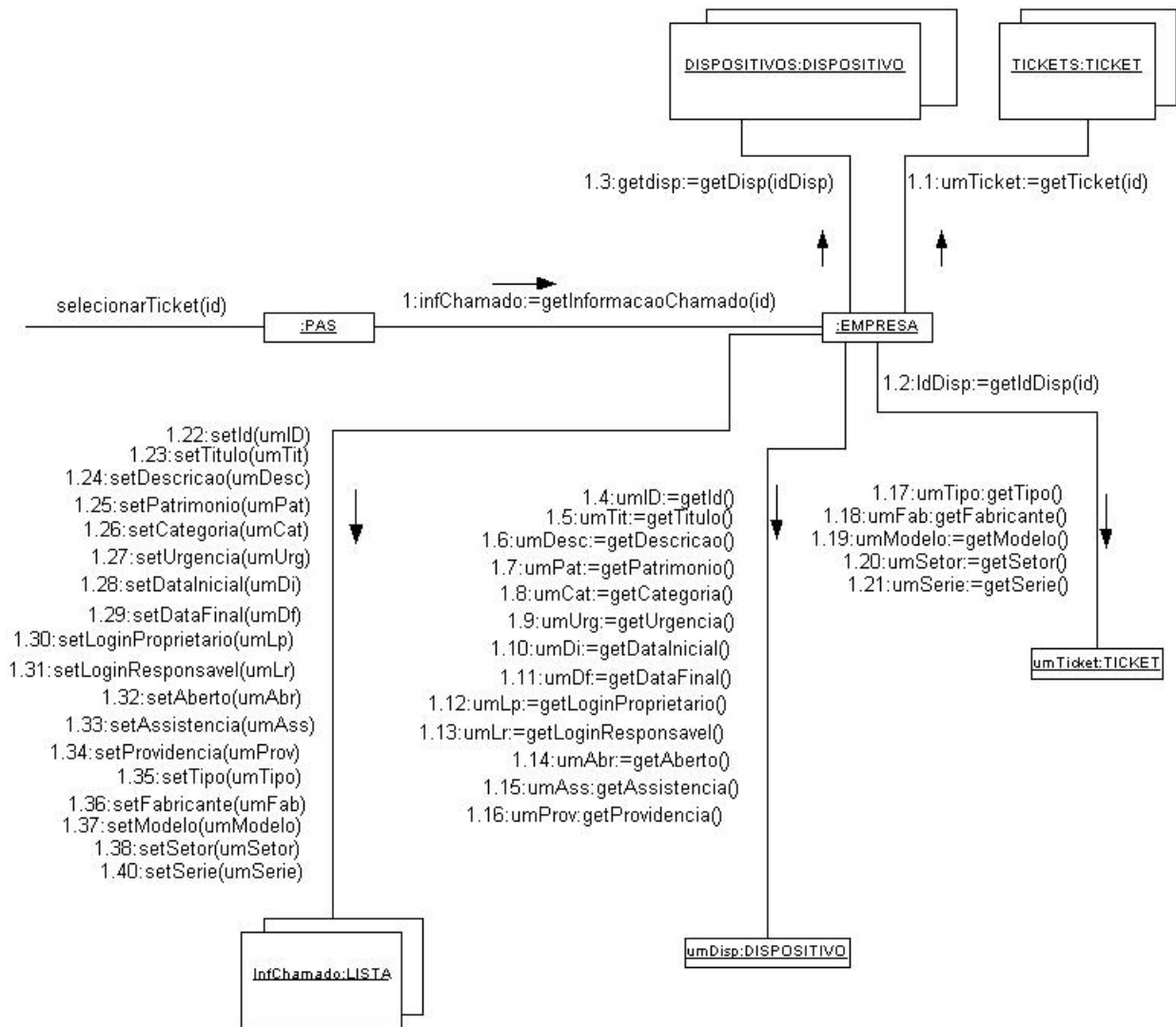


Figura 5.27 Diagrama de colaboração para a operação selecionarTicket

Diagrama de Classes

Depois de criados os diagramas de colaboração, é possível identificarem-se as classes que surgiram e seus respectivos métodos. De posse dessas informações é simples elaborar o diagrama de classes. A rigor, os dois diagramas surgem paralelamente, pois o esforço na criação do diagrama de colaboração é quase que o esforço total necessário para a definição das classes dos objetos no diagrama de classes.

Este diagrama é muito útil, pois ilustra globalmente as especificações das classes de softwares e suas interações numa aplicação. As principais informações contidas em um diagrama de classes são:

- o as próprias classes;

- as associações entre as classes;
- os atributos das classes e seus tipos;
- os métodos das classes;
- a navegabilidade entre as classes;
- e as dependências.

É importante mencionar que existe grande diferença entre o diagrama de classe e o modelo conceitual. O primeiro representa as classes de uma aplicação e o último define as entidades envolvidas no domínio do problema que não necessariamente se tornarão classes de software.

A seguir, na figura 5.28, o diagrama de classes deste trabalho.

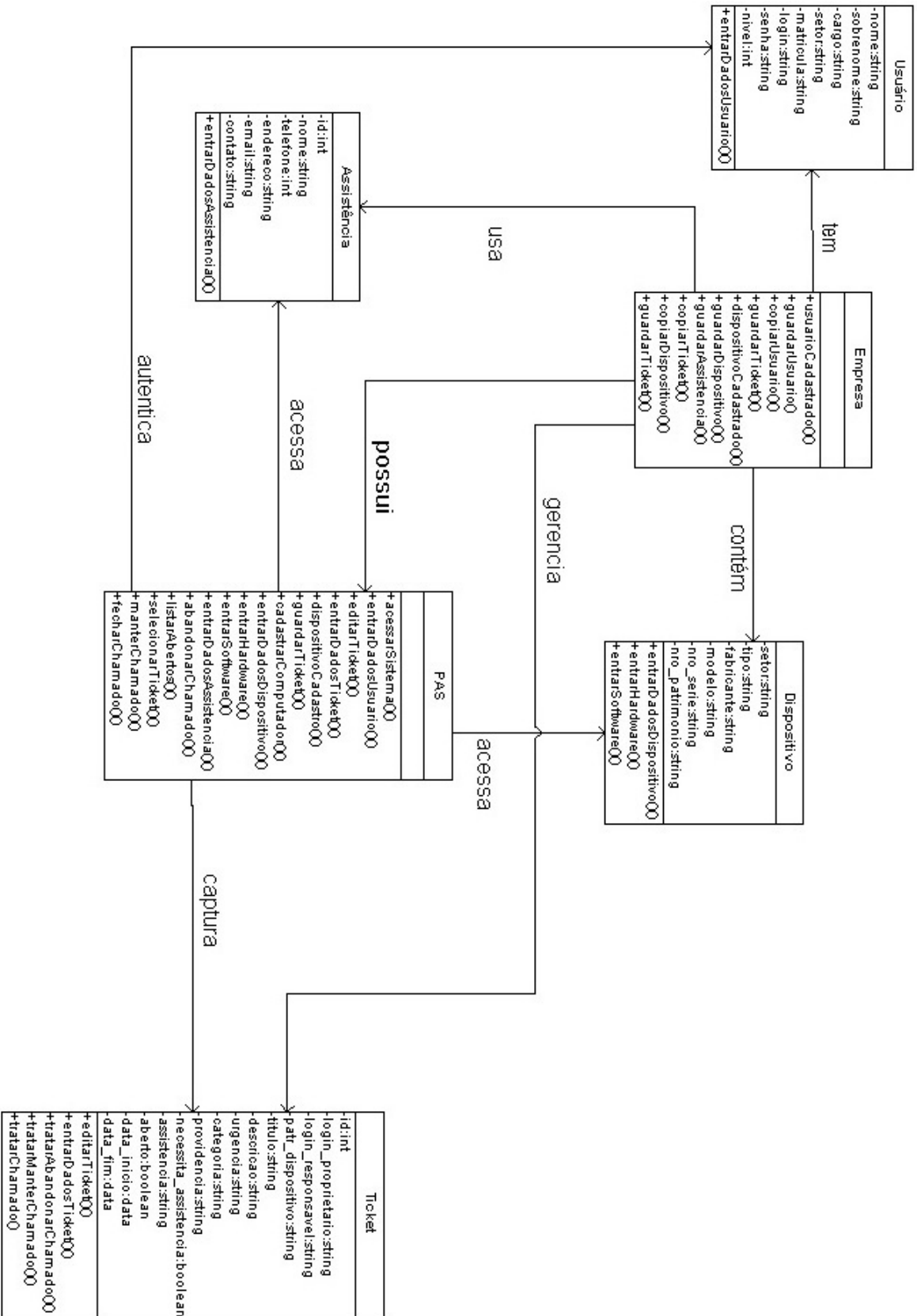


Figura 5.28 Diagrama de classes do projeto

5.3 Ferramentas utilizadas

O sistema é implementado em Java essencialmente porque a Sun Microsystems, sua desenvolvedora, dispõe de boa estrutura de apoio para construção de Web Services através dos pacotes disponibilizados. Também porque todos os pacotes são distribuídos livremente.

As ferramentas utilizadas neste trabalho estão descritas a seguir:

JSDK

Java Standard Developer Kit é o compilador Java, o interpretador Java, as principais APIs Java e o código fonte das APIs. A versão utilizada foi j2sdk1.4.2_04.

JWSDP

Java Web Service Developer Pack é o pacote de aplicações criado pela Sun Microsystems onde reúne várias tecnologias úteis para implementar Web Service em todo o ciclo de desenvolvimento. Neste pacote existem várias APIs e diversas ferramentas para a criação e publicação de serviços. A versão utilizada foi jwsdp - 1.0.01.

As APIs do pacote são:

- SAAJ (SOAP with Attachments API for Java) – módulo composto por diversas APIs existentes para manipular envelopes SOAP.
- JAXB (Java Architecture for XML Binding) – facilita o uso do XML compilando um esquema XML em uma ou mais classes Java, ou seja, elimina a necessidade de codificar classes para manipular XML.
- JAXM (Java API for XML Messaging) – fornece o código para criar mensagens SOAP.
- JAXP (Java API for XML Processing) – fornece uma interface padronizada para diferentes tipos de parsers XML, sendo utilizada para interpretar, criar e alterar mensagens.
- JAXR (Java API for XML Registries) – fornece uma API Java padrão para acessar diferentes tipos de registros XML (UDDI).
- JAX-RPC (Java API for XML-based RPC) – é a principal API para desenvolvimento do Web Service, pois é ela que incorpora

funcionalidade de modelo RPC baseadas em XML no código. É a API que será utilizada no trabalho.

A API JAX-RPC é o módulo responsável pela criação de Web Service nos moldes de RPC no Java e ajuda na criação do código de interface. As interfaces são as portas de entrada entre o cliente e o servidor. O código de interface do lado servidor chama-se "tie", e o código de interface do lado cliente chama-se "stubs".

Estes dois módulos são gerados por um conjunto de ferramentas chamadas wsdeploy (cria o tie) e wscompile (cria o stub).

O wsdeploy examina os métodos indicados que fornecerão os serviços do Web Service e cria várias classes que decodificam e codificam as mensagens SOAP trocadas com o cliente. Essa ferramenta também é responsável pela criação do arquivo WSDL, que descreve o serviço. Sua outra responsabilidade é a formação de um servlet de wrapper dentro do qual o serviço executará. Este servlet, juntamente com o código do serviço, compõe o arquivo WAR (Web Application Archive) que será implementado e estará disponível no servidor Tomcat. Desta forma, o módulo tie é criado.

O wscompile funciona examinando o arquivo WSDL do Web Service desejado. Imediatamente é gerado um conjunto de classes proxy que manipulará a comunicação entre o cliente e o serviço, ou seja, é criado o módulo stub. Assim, o programa cliente acessa os métodos remotos através do módulo stub como se tivesse implementado nele mesmo.

A figura 5.29 mostra a estrutura do JAX-RPC.

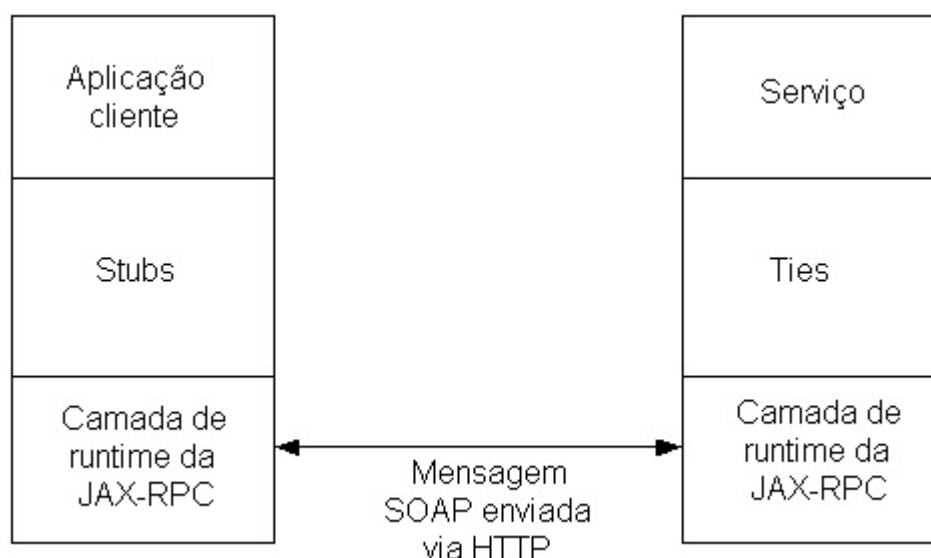


Figura 5.29 Estrutura do JAX-RPC .

Fonte: [POTTS, 2003], página 174.

Tomcat

O Tomcat é um servlet container onde são instalados os servlets de wrapper que tratam das requisições que o servidor vai receber. A versão utilizada foi Apache Tomcat 4.1.31.

Mysql

Mysql é um sistema de gerenciamento de banco de dados relacional, multiencadeado, baseado em comandos SQL (Structured Query Language) e de código fonte aberto. A versão utilizada foi Mysql-4.1.7.

O código fonte da aplicação está na seção de Anexos.

5.4 Módulos implantados

Módulos são partes do sistema divididas por contexto. Cada módulo trata de um tema identificado na fase de análise do problema, sendo que os elementos que compõem cada módulo são operações. Portanto, os módulos identificados e implementados foram: Chamado, Dispositivo, Usuário e Assistência.

Chamado

O Módulo Chamado trata de todas operações relacionadas com os processos de chamado técnico de reparo. As principais funções deste módulo são: “Abrir Chamado”, “Pesquisar Chamado” e “Tratar Chamado”.

Dispositivo

O Módulo Dispositivo trata de todas operações relacionadas com os processos envolvendo os dispositivos cadastrados no sistema. As principais funções deste módulo são: “Cadastrar Dispositivo” e “Pesquisar Dispositivo”.

Usuário

O Módulo Usuário trata das operações relacionadas com a administração dos usuários no sistema. As principais funções deste módulo são: “Cadastrar Usuário” e “Pesquisar Usuário”.

Assistência

O Módulo Assistência trata das operações relacionadas com a administração do cadastro das assistências técnicas contratadas pela IOESC. As principais funções deste módulo são: “Cadastro de Assistência” e “Pesquisa de Assistência”.

Os módulos compõem o conjunto de ferramentas que os usuários dispõem para utilizar o sistema. Naturalmente, cada módulo está disponível de acordo com a permissão do usuário.

É possível notar, nas cópias de telas do sistema apresentadas a seguir, que as opções do menu à esquerda estão distribuídas conforme as operações que compõem cada módulo.

5.5 Telas

Um dos objetivos do sistema é oferecer uma interface simples e estimulante para o usuário. Desta forma, optou-se por utilizar um menu à esquerda contendo todas as opções disponíveis ao usuário corrente. À direita há um espaço onde surge o formulário relativo à opção escolhida no menu. A figura 5.30 é uma cópia de tela da interface gráfica de um usuário Utente, realizando uma chamada de reparo.

The screenshot shows a window titled "Chasque - Console" with a dark blue header. The main area is light gray and contains a form titled "Abrir Chamado". On the left side, there are several sections: "Usuário: Fábio" with a "Trocar" button; "Chamado" with "Abrir" and "Pesquisar" buttons; "Senha" with a "Trocar" button; and "Visão:" with a radio button selected for "Usuário". The "Abrir Chamado" form has the following fields: "Título:" (text input), "Descrição:" (large text area), "Urgência:" (dropdown menu), "Data:" (split into "dia", "mes", and "ano" inputs), "Dispositivo" (text input), and "Patrimônio:" (text input). At the bottom of the form are "Submeter" and "Sair" buttons.

Figura 5.30 Exemplo de tela para usuário Utente realizando um chamado técnico

O usuário Técnico tem o dever de realizar o tratamento de um chamado designado a ele. Deste modo, o técnico solicita ao sistema a lista de chamados abertos que aguardam providências. Após obter a lista, o Técnico elege o chamado que será atendido primeiramente e assim requer que ele apareça na tela para realizar o tratamento. Com o chamado ampliado, ele deve completar as lacunas como na figura 5.31.

Chasque - Console

Técnico:
JB

Chamado

Listar abertos

Pesquisar

Dispositivo

Cadastrar

Pesquisar

Senha

Trocar

Visão:
 Usuário Técnico

Tratar chamado

Ticket n.º :
Título:
Descrição:
Dispositivo
Tipo:
Marca:
Modelo:
Setor:
Patrimônio:
N.º série:
Proprietário:
Urgência:
Data Abertura:
Data Conclusão: dia mes ano
Categoria do Problema:
Providências:

Submeter Fechar chamado Abandonar chamado Sair

Figura 5.31 Exemplo de tela para usuário Técnico realizando um tratamento.

O usuário Administrador procede diretamente no servidor e desempenha, entre outras atividades, o processo de tratamento de chamado envolvendo alguma assistência técnica externa. A figura 5.32 é uma cópia de tela de um usuário Administrador realizando um tratamento.

Chasque Administrador: **Verani**

Chamado

Abrir
Listar abertos
Pesquisar

Dispositivo

Cadastrar
Pesquisar

Usuário

Cadastrar
Listar
Pesquisar

Assistência

Cadastrar
Listar
Pesquisar

Senha

Trocar

Chamado

Ticket n.º :
Título:
Descrição:
Dispositivo
Tipo:
Marca:
Modelo:
Setor:
Patrimônio:
N.º série:
Categoria do Problema:
Urgência:
Data Abertura dia: mes: ano:
Data Conclusão: dia: mes: ano:
Proprietário:
Técnico Responsável:
Status:
Providências:
Assistência:

Submeter Fechar chamado Sair Excluir

Figura 5.32 Exemplo de tela para usuário Administrador realizando um tratamento.

CONCLUSÃO

O sistema proposto neste trabalho pretende auxiliar as pessoas envolvidas na manutenção de equipamentos no âmbito da IOESC. Desta forma, as atividades que apóiam este processo tornaram-se mais rápidas e organizadas comparadas ao panorama anterior, proporcionando aos usuários benefícios e aos administradores um conjunto de recursos para gerenciar o andamento das atividades. Além disso, os dados podem dar subsídio a uma futura base de conhecimento capaz de identificar novos problemas e ocorrências outrora escondidas em procedimentos inadequados.

Modelado para respeitar o regulamento e os procedimentos convencionais, necessários e habituais da IOESC, ele registra os pedidos de reparo, forma o histórico, auxilia no acompanhamento dos serviços e atribui responsabilidades. O elemento central, um ticket virtual, carrega consigo todas as informações pertinentes ao chamado que ele representa, simulando um bilhete de recado. É dotado de uma interface gráfica funcionalmente simples que favorece a adaptação, a introdução no ambiente e incentiva o uso através deste atributo causando baixo impacto e rejeição.

A utilização de Web Service e Java, através do pacote JWSDP distribuído pela Sun Microsystems, mais especificamente a API JAX-RPC que incorpora funcionalidade de modelo RPC baseadas em XML no código, foi motivada por características potencialmente úteis aos interesses da IOESC, tais como padrões abertos, flexibilidade e aptidão para a Internet. Desta forma, é possível a geração de baixos custos operacionais e, futuramente, integração de funcionalidades com parceiros comerciais externos, além de suportar novos modelos empresariais.

Ele atende às necessidades atuais do processo e, assim como todo sistema, está sujeito à defasagem tecnológica e a um período limitado de existência útil que compreende entre o momento de concepção e o surgimento das primeiras insuficiências funcionais devido às alterações que o próprio ambiente sofre, onde necessidades surgem e desaparecem sutilmente.

Assim, o produto deste trabalho tem por objetivo oferecer um recurso satisfatório às necessidades de manutenção de equipamentos da IOESC e contribuir, portanto, para o bom andamento das atividades que dependem destes equipamentos.

Trabalhos futuros

A finalidade deste trabalho foi dar os primeiros passos para que o processo de manutenção forneça elementos para novas iniciativas. Dentre os possíveis futuros trabalhos pode-se pautar uma expansão do sistema para dar suporte a mais tipos de dispositivos, saindo do limitado universo dos computadores e seus periféricos. Considerando que na IOESC existe uma variada concentração de máquinas em seu parque gráfico, é bastante razoável ampliar-se o sistema para suportar as peculiaridades de cada um dos equipamentos. Indo um pouco além, adicionar o controle da frota de veículos da empresa. Assim, o sistema estaria consideravelmente mais integrado às reais necessidades.

Outra sugestão é articular um console para dispositivos móveis – PDAs. Seria muito útil em situações onde os técnicos não possuem computadores para relatar as ocorrências, como é o caso de equipamentos no parque gráfico. Esta extensão do projeto é bastante facilitada porque a união entre os sistemas é favorecida através da interface Web Service que o servidor possui.

Por fim, para aperfeiçoar o atual processo informatizado de manutenção, seria conveniente integrar o sistema de manutenção da IOESC com sistemas oriundos dos fornecedores de assistência técnica. Assim, agilizaria o procedimento de tomada de preço, da solicitação de serviço e do fornecimento de relatórios referentes aos serviços prestados. A integração deveria ocorrer através de uma interface Web Service específica.

Ambicionando cada vez mais funcionalidades ao sistema, forma-se um ciclo de desenvolvimento contínuo capaz de gerar muitas novas soluções para procedimentos ineficazes que estão radicados na rotina de trabalho dos funcionários. Vagarosamente, vai-se tomando das pessoas determinadas funções árduas que são mais adequadas por meios informatizados.

BIBLIOGRAFIA

[POTTS, 2003] POTTS, Stephen; KOPACK, Mike. **Aprenda em 24 horas Web Services**. Rio de Janeiro: Campus, 2003.

[BECKER, 2001] BECKER, Aleksander Knabben. **Web Service e XML: novo paradigma da Programação Distribuída**. Florianópolis, UFSC, 2001.

[HENDRICKS, 2002] HENDRICKS, Marck; GALBRAITH, Ben; IRANI, Romin; MILBERNY, James; MODI, Tarak; TOST Andre; TOUSSAINT, Alex; BASHA, S. Jeelani; CABLE Scott. **Professional Java Web Service**. Rio de Janeiro: Alta Books, 2002.

[BASIURA, 2003] BASIURA, Russ et al. **Professional ASP.NET Web Service**. São Paulo: Pearson Education, 2003.

Apache Jakarta Tomcat

<http://jakarta.apache.org/tomcat/>

Mysql

<http://www.mysql.com/>

Java – Sun Microsystems

<http://java.sun.com/>

ANEXOS

Anexo 1 - Código fonte da aplicação

```
/*Assistencia.java*/

package suatec;

public class Assistencia {
    private int id;
    private String telefone;
    private String nome;
    private String endereco;
    private String email;
    private String contato;

    public Assistencia() {
    }

    public void setId(int umId) {
        id=umId;
    }

    public int getId() {
        return id;
    }

    public void setNome(String umNome) {
        nome=umNome;
    }

    public String getNome() {
        return nome;
    }

    public void setTelefone(String umTelefone) {
        telefone=umTelefone;
    }

    public String getTelefone() {
        return telefone;
    }

    public void setEndereco(String umEndereco) {
        endereco=umEndereco;
    }

    public String getEndereco() {
        return endereco;
    }

    public void setEmail(String umEmail) {
        email=umEmail;
    }

    public String getEmail() {
        return email;
    }
}
```

```

    }

    public void setContato(String umContato) {
        contato=umContato;
    }

    public String getContato() {
        return contato;
    }
}

/*****UsuarioDAO.java*****/

package suatec;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

import java.sql.ResultSet;

public class UsuarioDAO extends GenericDAO {
    protected static PreparedStatement psSelectUsuario;
    protected static PreparedStatement psUpdateUsuario;
    protected static PreparedStatement psInsertUsuario;
    protected static PreparedStatement psDeleteUsuario;
    protected static PreparedStatement psSelectTodosUsuarios;

    public UsuarioDAO(int indice) {
        super();
        setIndice(indice);
    }

    public void initialize() {
        try {
            psSelectUsuario = getConnection(indice).prepareStatement("
SELECT U.nome, U.sobrenome, U.cargo, U.setor, U.matricula, U.login,
U.senha, U.nivel FROM usuario U");
            psSelectTodosUsuarios =
getConnection(indice).prepareStatement(" SELECT U.nome, U.sobrenome,
U.cargo, U.setor, U.matricula, U.login, U.senha, U.nivel FROM usuario U");
            psUpdateUsuario = getConnection(indice).prepareStatement("
UPDATE usuario SET nome = ?, sobrenome = ?, cargo = ?, setor = ?, matricula
= ?, login = ?, senha = ?, nivel = ? WHERE login = ?");
            psInsertUsuario = getConnection(indice).prepareStatement("
INSERT INTO usuario (id, nome, sobrenome, cargo, setor, matricula, login,
senha, nivel) VALUES (?,?,?,?,?,?,?,?,?)");
            psDeleteUsuario = getConnection(indice).prepareStatement("
DELETE FROM usuario WHERE login = ?");
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList selectUsuarios(String nome) throws SQLException {
        ArrayList usuarios = new ArrayList();

```

```

Usuario usuario = null;
try {
    ResultSet rs = psSelectUsuario.executeQuery();
    int i = 0;
    while (rs.next()) {
        try {
            if (rs.getString("nome").compareTo(nome)==0) {
                usuario = new Usuario();
                usuario.setNome(rs.getString("nome"));
                usuario.setSobrenome(rs.getString("sobrenome"));
                usuario.setCargo(rs.getString("cargo"));
                usuario.setSetor(rs.getString("setor"));
                usuario.setMatricula(rs.getString("matricula"));
                usuario.setLogin(rs.getString("login"));
                usuario.setSenha(rs.getString("senha"));
                usuario.setNivel(rs.getInt("nivel"));
                usuarios.add(usuario);
                i++;
            }
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
catch(SQLException e) {
    e.printStackTrace();
}
return usuarios;
}

public ArrayList selectTodosUsuarios() throws SQLException {
    ArrayList usuarios = new ArrayList();
    Usuario usuario = null;
    try {
        ResultSet rs = psSelectUsuario.executeQuery();
        int i = 0;
        while (rs.next()) {
            try {
                usuario = new Usuario();
                usuario.setNome(rs.getString("nome"));
                usuario.setSobrenome(rs.getString("sobrenome"));
                usuario.setCargo(rs.getString("cargo"));
                usuario.setSetor(rs.getString("setor"));
                usuario.setMatricula(rs.getString("matricula"));
                usuario.setLogin(rs.getString("login"));
                usuario.setSenha(rs.getString("senha"));
                usuario.setNivel(rs.getInt("nivel"));
                usuarios.add(usuario);
                i++;
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
    }
    return usuarios;
}

```

```

}

public boolean updateUsuario(Usuario dtoUsuario) throws SQLException {
    boolean retorno = false;
    try {
        psUpdateUsuario.setString(1, dtoUsuario.getNome());
        psUpdateUsuario.setString(2, dtoUsuario.getSobrenome());
        psUpdateUsuario.setString(3, dtoUsuario.getCargo());
        psUpdateUsuario.setString(4, dtoUsuario.getSetor());
        psUpdateUsuario.setString(5, dtoUsuario.getMatricula());
        psUpdateUsuario.setString(6, dtoUsuario.getLogin());
        psUpdateUsuario.setString(7, dtoUsuario.getSenha());
        psUpdateUsuario.setInt(8, dtoUsuario.getNivel());
        psUpdateUsuario.setString(9, dtoUsuario.getLogin());
        retorno = psUpdateUsuario.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean insertUsuario(Usuario dtoUsuario) throws SQLException {
    boolean retorno = false;

    try {
        psInsertUsuario.setLong(1, dtoUsuario.getId());
        psInsertUsuario.setString(2, dtoUsuario.getNome());
        psInsertUsuario.setString(3, dtoUsuario.getSobrenome());
        psInsertUsuario.setString(4, dtoUsuario.getCargo());
        psInsertUsuario.setString(5, dtoUsuario.getSetor());
        psInsertUsuario.setString(6, dtoUsuario.getMatricula());
        psInsertUsuario.setString(7, dtoUsuario.getLogin());
        psInsertUsuario.setString(8, dtoUsuario.getSenha());
        psInsertUsuario.setInt(9, dtoUsuario.getNivel());
        retorno = psInsertUsuario.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean deleteUsuario(String login) throws SQLException {
    boolean retorno = false;
    try {
        psDeleteUsuario.setString(1, login);
        retorno = psDeleteUsuario.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public void setIndice(int indice) {
    this.indice = indice;
}

```



```

        public void setTabela(String tabela_) {
            this.tabela = tabela_;
        }
    }

/*****AssistenciaDAO.java*****/

package suatec;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

import java.sql.ResultSet;

public class AssistenciaDAO extends GenericDAO {
    protected static PreparedStatement psSelectAssistencia;
    protected static PreparedStatement psUpdateAssistencia;
    protected static PreparedStatement psInsertAssistencia;
    protected static PreparedStatement psDeleteAssistencia;
    protected static PreparedStatement psSelectTodasAssistencia;

    public AssistenciaDAO(int indice) {
        super();
        setIndice(indice);
    }

    public void initialize() {
        try {
            psSelectAssistencia = getConnection(indice).prepareStatement("
SELECT A.id, A.nome, A.telefone, A.endereco, A.email, A.contato FROM
assistencia A");
            psSelectTodasAssistencia =
getConnection(indice).prepareStatement(" SELECT A.id, A.nome, A.telefone,
A.endereco, A.email, A.contato FROM assistencia A");
            psUpdateAssistencia = getConnection(indice).prepareStatement("
UPDATE assistencia SET nome = ?, telefone = ?, endereco = ?, email = ?,
contato = ? WHERE nome = ?");
            psInsertAssistencia = getConnection(indice).prepareStatement("
INSERT INTO assistencia (id, nome, telefone, endereco, email, contato)
VALUES (?,?,?,?);");
            psDeleteAssistencia = getConnection(indice).prepareStatement("
DELETE FROM assistencia WHERE nome = ?");
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList selectAssistencias(String nome) throws SQLException {
        ArrayList assistencias = new ArrayList();
        Assistencia assistencia = null;
        try {
            ResultSet rs = psSelectAssistencia.executeQuery();
            int i = 0;
            while (rs.next()) {
                try {
                    if (rs.getString("nome").compareTo(nome)==0) {

```

```

        assistencia = new Assistencia();
        assistencia.setId(rs.getInt("id"));
        assistencia.setNome(rs.getString("nome"));
        assistencia.setTelefone(rs.getString("telefone"));
        assistencia.setEndereco(rs.getString("endereco"));
        assistencia.setEmail(rs.getString("email"));
        assistencia.setContato(rs.getString("contato"));
        assistencias.add(assistencia);
        i++;
    }
}
catch(Exception e) {
    e.printStackTrace();
}
}
catch(SQLException e) {
    e.printStackTrace();
}
return assistencias;
}

public ArrayList selectTodasAssistencias() throws SQLException {
    ArrayList assistencias = new ArrayList();
    Assistencia assistencia = null;
    try {
        ResultSet rs = psSelectAssistencia.executeQuery();
        int i = 0;
        while (rs.next()) {
            try {
                assistencia = new Assistencia();
                assistencia.setId(rs.getInt("id"));
                assistencia.setNome(rs.getString("nome"));
                assistencia.setTelefone(rs.getString("telefone"));
                assistencia.setEndereco(rs.getString("endereco"));
                assistencia.setEmail(rs.getString("email"));
                assistencia.setContato(rs.getString("contato"));
                assistencias.add(assistencia);
                i++;
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
    }
    return assistencias;
}

public boolean updateAssistencia(Assistencia dtoAssistencia) throws
SQLException {
    boolean retorno = false;

    try {
        psUpdateAssistencia.setString(1, dtoAssistencia.getNome());
        psUpdateAssistencia.setString(2, dtoAssistencia.getTelefone());
        psUpdateAssistencia.setString(3, dtoAssistencia.getEndereco());
        psUpdateAssistencia.setString(4, dtoAssistencia.getEmail());
    }
}

```

```

        psUpdateAssistencia.setString(5, dtoAssistencia.getContato());
        psUpdateAssistencia.setString(6, dtoAssistencia.getNome());
        retorno = psUpdateAssistencia.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean insertAssistencia(Assistencia dtoAssistencia) throws
SQLException {
    boolean retorno = false;
    try {
        psInsertAssistencia.setLong  (1, dtoAssistencia.getId());
        psInsertAssistencia.setString(2, dtoAssistencia.getNome());
        psInsertAssistencia.setString(3, dtoAssistencia.getTelefone());
        psInsertAssistencia.setString(4, dtoAssistencia.getEndereco());
        psInsertAssistencia.setString(5, dtoAssistencia.getEmail());
        psInsertAssistencia.setString(6, dtoAssistencia.getContato());
        retorno = psInsertAssistencia.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean deleteAssistencia(String nome) throws SQLException {
    boolean retorno = false;
    try {
        psDeleteAssistencia.setString(1, nome);
        retorno = psDeleteAssistencia.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public void setIndice(int indice) {
    this.indice = indice;
}

public void setTabela(String tabela_) {
    this.tabela = tabela_;
}
}

/*****ConnectionFactory.java*****/

package suatec;

import java.sql.Connection;

public class ConnectionManager {
    private MySqlConnection mysqlConnection;

    public Connection getConnection() {
        mysqlConnection = new MySqlConnection();
    }
}

```

```

        return mysqlConnection.getConnection();
    }

    public Connection getConnection(String pathProperties) {
        mysqlConnection = new MySqlConnection(pathProperties);
        return mysqlConnection.getConnection();
    }

    public static void main(String[] args) {
        ConnectionManager con = new ConnectionManager();
    }
}

/*****Dispositivo.java*****/

package suatec;

public class Dispositivo {
    private int id;
    private String setor;
    private String tipo;
    private String fabricante;
    private String modelo;
    private String nro_serie;
    private String nro_patrimonio;

    public Dispositivo() {
    }

    public void setId(int umId) {
        id=umId;
    }

    public int getId() {
        return id;
    }

    public void setSetor(String umSetor) {
        setor=umSetor;
    }

    public String getSetor() {
        return setor;
    }

    public void setTipo(String umTipo) {
        tipo=umTipo;
    }

    public String getTipo() {
        return tipo;
    }

    public void setFabricante(String umFabricante) {
        fabricante=umFabricante;
    }

    public String getFabricante() {
        return fabricante;
    }
}

```

```

public void setModelo(String umModelo) {
    modelo=umModelo;
}

public String getModelo() {
    return modelo;
}

public void setNro_serie(String umNro_serie) {
    nro_serie=umNro_serie;
}

public String getNro_serie() {
    return nro_serie;
}

public void setNro_patrimonio(String umNro_patrimonio) {
    nro_patrimonio=umNro_patrimonio;
}

public String getNro_patrimonio() {
    return nro_patrimonio;
}
}

/*****DispositivoDAO.java*****/

package suatec;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

import java.sql.ResultSet;

public class DispositivoDAO extends GenericDAO {
    protected static PreparedStatement psSelectDispositivo;
    protected static PreparedStatement psUpdateDispositivo;
    protected static PreparedStatement psInsertDispositivo;
    protected static PreparedStatement psDeleteDispositivo;
    protected static PreparedStatement psSelectTodosDispositivo;

    public DispositivoDAO(int indice) {
        super();
        setIndice(indice);
    }

    public void initialize() {
        try {
            psSelectDispositivo = getConnection(indice).prepareStatement("
SELECT D.id, D.setor, D.tipo, D.fabricante, D.modelo, D.nro_serie,
D.nro_patrimonio FROM dispositivo D");
            psSelectTodosDispositivo =
getConnection(indice).prepareStatement(" SELECT D.id, D.setor, D.tipo,
D.fabricante, D.modelo, D.nro_serie, D.nro_patrimonio FROM dispositivo D");
            psUpdateDispositivo = getConnection(indice).prepareStatement("
UPDATE dispositivo SET setor = ?, tipo = ?, fabricante = ?, modelo = ?,
nro_serie = ?, nro_patrimonio = ? WHERE nro_patrimonio = ?");

```

```

        psInsertDispositivo = getConnection(indice).prepareStatement("
INSERT INTO dispositivo (id, setor, tipo, fabricante, modelo, nro_serie,
nro_patrimonio) VALUES (?, ?, ?, ?, ?, ?, ?)");
        psDeleteDispositivo = getConnection(indice).prepareStatement("
DELETE FROM dispositivo WHERE nro_patrimonio = ?");
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

    public ArrayList selectDispositivos(String nro_patrimonio) throws
SQLException {
        ArrayList dispositivos = new ArrayList();
        Dispositivo dispositivo = null;
        try {
            ResultSet rs = psSelectDispositivo.executeQuery();
            int i = 0;
            while (rs.next()) {
                try {
                    if
(rs.getString("nro_patrimonio").compareTo(nro_patrimonio)==0) {
                        dispositivo = new Dispositivo();
                        dispositivo.setSetor(rs.getString("setor"));
                        dispositivo.setTipo(rs.getString("tipo"));

dispositivo.setFabricante(rs.getString("fabricante"));
                        dispositivo.setModelo(rs.getString("modelo"));

dispositivo.setNro_serie(rs.getString("nro_serie"));

dispositivo.setNro_patrimonio(rs.getString("nro_patrimonio"));
                        dispositivos.add(dispositivo);
                        i++;
                    }
                }
                catch(Exception e) {
                    e.printStackTrace();
                }
            }
        }
        catch(SQLException e) {
            e.printStackTrace();
        }
        return dispositivos;
    }
}

```

```

    public ArrayList selectTodosDispositivos() throws SQLException {
        ArrayList dispositivos = new ArrayList();
        Dispositivo dispositivo = null;
        try {
            ResultSet rs = psSelectDispositivo.executeQuery();
            int i = 0;
            while (rs.next()) {
                try {
                    dispositivo = new Dispositivo();
                    dispositivo.setSetor(rs.getString("setor"));
                    dispositivo.setTipo(rs.getString("tipo"));
                    dispositivo.setFabricante(rs.getString("fabricante"));
                    dispositivo.setModelo(rs.getString("modelo"));
                    dispositivo.setNro_serie(rs.getString("nro_serie"));

```

```

dispositivo.setNro_patrimonio(rs.getString("nro_patrimonio"));
                dispositivos.add(dispositivo);
                i++;
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
    }
    return dispositivos;
}

    public boolean updateDispositivo(Dispositivo dtoDispositivo) throws
SQLException {
        boolean retorno = false;
        try {
            psUpdateDispositivo.setString(1, dtoDispositivo.getSetor());
            psUpdateDispositivo.setString(2, dtoDispositivo.getTipo());
            psUpdateDispositivo.setString(3,
dtoDispositivo.getFabricante());
            psUpdateDispositivo.setString(4, dtoDispositivo.getModelo());
            psUpdateDispositivo.setString(5,
dtoDispositivo.getNro_serie());
            psUpdateDispositivo.setString(6,
dtoDispositivo.getNro_patrimonio());
            psUpdateDispositivo.setString(7,
dtoDispositivo.getNro_patrimonio());
            retorno = psUpdateDispositivo.execute();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return retorno;
    }

    public boolean insertDispositivo(Dispositivo dtoDispositivo) throws
SQLException {
        boolean retorno = false;
        try {
            psInsertDispositivo.setLong (1, dtoDispositivo.getId());
            psInsertDispositivo.setString(2, dtoDispositivo.getSetor());
            psInsertDispositivo.setString(3, dtoDispositivo.getTipo());
            psInsertDispositivo.setString(4,
dtoDispositivo.getFabricante());
            psInsertDispositivo.setString(5, dtoDispositivo.getModelo());
            psInsertDispositivo.setString(6,
dtoDispositivo.getNro_serie());
            psInsertDispositivo.setString(7,
dtoDispositivo.getNro_patrimonio());
            retorno = psInsertDispositivo.execute();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return retorno;
    }
}

```

```

    public boolean deleteDispositivo(String nro_patrimonio) throws
SQLException {
        boolean retorno = false;
        try {
            psDeleteDispositivo.setString(1, nro_patrimonio);
            retorno = psDeleteDispositivo.execute();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return retorno;
    }

    public void setIndice(int indice) {
        this.indice = indice;
    }

    public void setTabela(String tabela_) {
        this.tabela = tabela_;
    }
}

```

```

/*****Empresa.java*****/

```

```

package suatec;

import java.util.*;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

public class Empresa implements EmpresaInterface{
    ArrayList usuarios = new ArrayList();
    ArrayList assistencias = new ArrayList();
    ArrayList dispositivos = new ArrayList();
    ArrayList tickets = new ArrayList();
    UsuarioDAO usuarioDAO;
    AssistenciaDAO assistenciaDAO;
    DispositivoDAO dispositivoDAO;
    TicketDAO ticketDAO;

    public Empresa() {
        ConnectionManager con = new ConnectionManager();
        Connection connection = con.getConnection();
        int indice = GenericDAO.addConnection(connection);
        usuarioDAO = new UsuarioDAO(indice);
        assistenciaDAO = new AssistenciaDAO(indice);
        dispositivoDAO = new DispositivoDAO(indice);
        ticketDAO = new TicketDAO(indice);
        try {
            usuarios = usuarioDAO.selectTodosUsuarios();
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
        try {

```



```

        assistencias = assistenciaDAO.selectTodasAssistencias();
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        dispositivos = dispositivoDAO.selectTodosDispositivos();
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        tickets = ticketDAO.selectTodosTickets();
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
}

public boolean acessarSistema(String xlogin, String xsenha) {
    Usuario aux;
    boolean permitido = false;
    if (this.existeUsuarioCadastrado(xlogin)) {
        aux = copiarUsuario(xlogin);
        if (((aux.getLogin()).compareTo(xlogin) == 0) &&
            ((aux.getSenha()).compareTo(xsenha) == 0)) {
            permitido = true;
        }
    }
    return permitido;
}

public boolean criarUsuario(String xn,String xs,String xc,String
xset,String xm,String xl,String xsenh,int xnv) {
    boolean foiCriado;
    if (this.existeUsuarioCadastrado(xl)) {
        foiCriado = false;
    }
    else {
        Usuario umUsuario = new Usuario();
        umUsuario.setNome(xn);
        umUsuario.setSobrenome(xs);
        umUsuario.setCargo(xc);
        umUsuario.setSetor(xset);
        umUsuario.setMatricula(xm);
        umUsuario.setLogin(xl);
        umUsuario.setSenha(xsenh);
        umUsuario.setNivel(xnv);
        usuarios.add(umUsuario);
        foiCriado = true;
        boolean inseriu;
        try {
            inseriu = usuarioDAO.insertUsuario(umUsuario);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return foiCriado;
}
}

```

```

    public boolean criarDispositivo(String xpat, String xtip, String xset,
String xfab, String xmod, String xnro) {
        boolean foiCriado;
        if (this.existeDispositivoCadastrado(xpat)) {
            foiCriado = false;
        }
        else {
            Dispositivo umDispositivo = new Dispositivo();
            umDispositivo.setNro_patrimonio(xpat);
            umDispositivo.setTipo(xtip);
            umDispositivo.setSetor(xset);
            umDispositivo.setFabricante(xfab);
            umDispositivo.setModelo(xmod);
            umDispositivo.setNro_serie(xnro);
            dispositivos.add(umDispositivo);
            foiCriado = true;
            boolean inseriu;
            try {
                inseriu = dispositivoDAO.insertDispositivo(umDispositivo);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return foiCriado;
    }
}

```

```

    public boolean modificarDadosUsuario(String xn,String xs,String
xc,String xset,String xm,String xl,String xsenh,int xnv) {
        boolean foiModificado = false;
        if (this.existeUsuarioCadastrado(xl)){
            Usuario aux = new Usuario();
            foiModificado = true;
            aux.setNome(xn);
            aux.setSobrenome(xs);

            aux.setCargo(xc);
            aux.setSetor(xset);
            aux.setMatricula(xm);
            aux.setLogin(xl);
            aux.setSenha(xsenh);
            aux.setNivel(xnv);
            usuarios.set(this.posicaoListaUsuario(xl),aux);
            boolean atualizou;
            try {
                atualizou = usuarioDAO.updateUsuario(aux);
            }
            catch (SQLException e) {

                e.printStackTrace();
            }
        }
        return foiModificado;
    }
}

```

```

    public boolean modificarDadosDispositivo(String xpat,String
xtipo,String xsetor,String xfabr,String xmodelo,String xnrosérie) {
        boolean foiModificado = false;
        if (this.existeDispositivoCadastrado(xpat)){
            foiModificado = true;
            Dispositivo aux = new Dispositivo();
            aux.setNro_patrimonio(xpat);

```

```

        aux.setTipo(xtipo);
        aux.setSetor(xsetor);
        aux.setFabricante(xfabr);
        aux.setModelo(xmodelo);
        aux.setNro_serie(xnroserie);
        int posicaLista = this.posicaoListaDispositivo(xpat);
        dispositivos.set(posicaLista,aux);
        boolean atualizou;
        try {
            atualizou = dispositivoDAO.updateDispositivo(aux);
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return foiModificado;
}

public boolean existeUsuarioCadastrado(String login) {
    Usuario aux;
    boolean existe = false;
    for (int i=0;i<usuarios.size();i++) {
        aux = (Usuario)usuarios.get(i);
        if ((aux.getLogin()).compareTo(login)==0) {
            existe = true;
        }
    }
    return existe;
}

public boolean existeDispositivoCadastrado(String xpatt) {
    Dispositivo aux;
    boolean existe = false;
    for (int i=0;i<dispositivos.size();i++) {
        aux = (Dispositivo)dispositivos.get(i);
        if ((aux.getNro_patrimonio()).compareTo(xpatt)==0) {
            existe = true;
        }
    }
    return existe;
}

public ArrayList getListaLoginUsuariosCadastrados(){
    ArrayList listaLoginUsuariosCadastrados = new ArrayList();
    Usuario aux;
    String login;
    for (int i=0;i<usuarios.size();i++){
        aux = (Usuario)usuarios.get(i);
        login = aux.getLogin();
        listaLoginUsuariosCadastrados.add(login);
    }
    return listaLoginUsuariosCadastrados;
}

public ArrayList getListaDispositivosCadastrados(){
    ArrayList listaDispositivosCadastrados = new ArrayList();
    Dispositivo aux;
    String pat;
    for (int i=0;i<dispositivos.size();i++){
        aux = (Dispositivo)dispositivos.get(i);
        pat = aux.getNro_patrimonio();
    }
}

```

```

        listaDispositivosCadastrados.add(pat);
    }
    return listaDispositivosCadastrados;
}

public void excluirUsuario(String xlogin){
    Usuario aux;
    for (int i=0;i<usuarios.size();i++){
        aux=(Usuario)usuarios.get(i);
        if(aux.getLogin().compareTo(xlogin)==0){
            usuarios.remove(i);
            boolean delete;
            try {
                delete = usuarioDAO.deleteUsuario(xlogin);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

public void excluirDispositivo(String xnro_patr){
    Dispositivo aux;
    for (int i=0;i<dispositivos.size();i++){
        aux=(Dispositivo)dispositivos.get(i);
        if(aux.getNro_patrimonio().compareTo(xnro_patr)==0){
            dispositivos.remove(i);
            boolean delete;
            try {
                delete = dispositivoDAO.deleteDispositivo(xnro_patr);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

public ArrayList getInformacoesDispositivo(String xpatrr){
    ArrayList retorno = new ArrayList();
    Dispositivo aux;
    aux = this.copiarDispositivo(xpatrr);
    retorno.add(aux.getNro_patrimonio());
    retorno.add(aux.getTipo());
    retorno.add(aux.getSetor());
    retorno.add(aux.getFabricante());
    retorno.add(aux.getModelo());
    retorno.add(aux.getNro_serie());
    return retorno;
}

public ArrayList getInformacoesAssistencia(String xnomeass){
    ArrayList retorno = new ArrayList();
    Assistencia aux;
    aux = this.copiarAssistencia(xnomeass);
    retorno.add(aux.getNome());
    retorno.add(aux.getTelefone());
    retorno.add(aux.getEndereco());
    retorno.add(aux.getEmail());
    retorno.add(aux.getContato());
    return retorno;
}

```

```

}

public ArrayList getInformacoesUsuario(String xlogin){
    ArrayList retorno = new ArrayList();
    Usuario aux;
    aux = this.copiarUsuario(xlogin);
    retorno.add(aux.getNome());
    retorno.add(aux.getSobrenome());
    retorno.add(aux.getCargo());
    retorno.add(aux.getSetor());
    retorno.add(aux.getMatricula());
    retorno.add(aux.getLogin());
    retorno.add(aux.getSenha());
    Integer i = new Integer(aux.getNivel());
    retorno.add(i);
    return retorno;
}

public void troqueSenha(String l, String s) {
    Usuario aux = (Usuario)this.copiarUsuario(l);
    int indice = this.posicaoListaUsuario(l);
    aux.setSenha(s);
    usuarios.set(indice,aux);
}

public boolean existeAssistenciaCadastrada(String nome) {
    Assistencia aux;
    boolean existe = false;
    for (int i=0;i<assistencias.size();i++) {
        aux = (Assistencia)assistencias.get(i);
        if ((aux.getNome()).compareTo(nome)==0) {
            existe = true;
            return existe;
        }
    }
    return existe;
}

public boolean criarAssistencia(String xn,String xt,String xe,String
xm,String xc){
    boolean foiCriado;
    if (this.existeAssistenciaCadastrada(xn)) {
        foiCriado = false;
    }
    else {
        Assistencia umaAssistencia = new Assistencia();
        umaAssistencia.setNome(xn);
        umaAssistencia.setTelefone(xt);
        umaAssistencia.setEndereco(xe);
        umaAssistencia.setEmail(xm);
        umaAssistencia.setContato(xc);
        assistencias.add(umaAssistencia);
        foiCriado = true;
        boolean inseriu;
        try {
            inseriu = assistenciaDAO.insertAssistencia(umaAssistencia);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return foiCriado;
}

```

```

    }

    public boolean modificarAssistencia(String xn,String xt,String
xe,String xm,String xc){
        boolean foiModificada = false;
        Assistencia aux;
        String nome;
        for (int i=0;i<assistencias.size();i++){
            aux = (Assistencia)assistencias.get(i);
            nome = aux.getNome();
            if (nome.compareTo(xn)==0){
                aux.setTelefone(xt);
                aux.setEndereco(xe);
                aux.setEmail(xm);
                aux.setContato(xc);
                foiModificada = true;
                boolean atualizou;
                try {
                    atualizou = assistenciaDAO.updateAssistencia(aux);
                }
                catch (SQLException e) {
                    e.printStackTrace();
                }
                return foiModificada;
            }
        }
        return foiModificada;
    }

    public ArrayList getListaAssistenciasCadastradas(){
        ArrayList listaAssistenciasCadastradas = new ArrayList();
        Assistencia aux;
        String nome;
        for (int i=0;i<assistencias.size();i++){
            aux = (Assistencia)assistencias.get(i);
            nome = aux.getNome();
            listaAssistenciasCadastradas.add(nome);
        }
        return listaAssistenciasCadastradas;
    }

    public boolean excluirAssistencia(String xnome){
        Assistencia aux;
        String nome;
        boolean foiExcluido = false;
        for (int i=0;i<assistencias.size();i++){
            aux = (Assistencia)assistencias.get(i);
            nome = aux.getNome();
            if (nome.compareTo(xnome)==0){
                assistencias.remove(i);
                foiExcluido = true;
                boolean delete;
                try {
                    delete = assistenciaDAO.deleteAssistencia(xnome);
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
        return foiExcluido;
    }

```

```

    }

    public boolean criarTicket(String xtitulo,String xdescricao,String
xurgencia,String xdata_inicio,String xpatrimonio,String
xlogin_proprietario){
        Ticket aux = new Ticket();
        aux.setTitulo(xtitulo);
        aux.setDescricao(xdescricao);
        aux.setUrgencia(xurgencia);
        aux.setData_inicio(xdata_inicio);
        aux.setPatr_dispositivo(xpatrimonio);
        aux.setLogin_proprietario(xlogin_proprietario);
        aux.setLogin_responsavel(this.sortearResponsavel());
        aux.setAberto(true);
        aux.setNecessita_assistencia(false);
        aux.setId((tickets.size()));
        tickets.add(aux);
        boolean inseriu;
        try {
            inseriu = ticketDAO.insertTicket(aux);

        } catch (SQLException e) {

            e.printStackTrace();
        }
        return true;
    }

    public ArrayList getListaIdChamadosAbertos(){
        ArrayList lista = new ArrayList();
        Ticket aux;
        for (int i=0;i<tickets.size();i++){
            aux = (Ticket)tickets.get(i);

            if (aux.getAberto()){
                Integer ID = new Integer(aux.getId());
                lista.add(ID);
            }
        }
        return lista;
    }

    public boolean ehTecnico(String xlogin){
        boolean eh = false;
        Usuario aux;
        for (int i=0;i<usuarios.size();i++){
            aux = (Usuario)usuarios.get(i);
            if
((aux.getLogin().compareTo(xlogin)==0)&&(aux.getNivel())>=2)){
                eh = true;
                return eh;
            }
        }
        return eh;
    }

    public ArrayList getListaIdChamadosAbertosPendentes(){
        ArrayList lista = new ArrayList();
        Ticket aux;

        for (int i=0;i<tickets.size();i++){

```

```

        aux = (Ticket)tickets.get(i);
        if ((aux.getAberto()) && (aux.getNecessita_assistencia())){
            Integer ID = new Integer(aux.getId());
            lista.add(ID);
        }
    }
    return lista;
}

public boolean excluirTicket(int xindice){
    Ticket aux;
    boolean excluido = false;
    if (this.existeTicketCadastrado(xindice)){
        for (int i=0;i<tickets.size();i++){
            aux=(Ticket)tickets.get(i);
            if(aux.getId() == xindice){
                tickets.remove(i);
                excluido = true;
                boolean delete;
                try {
                    delete = ticketDAO.deleteTicket(xindice);
                } catch (SQLException e) {
                    e.printStackTrace();
                }
                return excluido;
            }
        }
    }
    return excluido;
}

public boolean existeTicketCadastrado(int xindice) {
    Ticket aux;
    boolean existe = false;
    for (int i=0;i<tickets.size();i++) {
        aux = (Ticket)tickets.get(i);
        if (aux.getId()==xindice) {
            existe = true;
            return existe;
        }
    }
    return existe;
}

public ArrayList getListaIdChamados(boolean ehPatrimonio, String nro){
    ArrayList lista = new ArrayList();
    Ticket aux;
    for (int i=0;i<tickets.size();i++){
        aux = (Ticket)tickets.get(i);
        if (ehPatrimonio){
            String xpatr = aux.getPatr_dispositivo();
            if (nro.compareTo(xpatr)==0){
                Integer ID = new Integer(aux.getId());
                lista.add(ID);
            }
        }else{
            int xid = aux.getId();
            int xnro = Integer.parseInt(nro);
            if (xid==xnro){
                Integer ID = new Integer(aux.getId());
                lista.add(ID);
            }
        }
    }
}

```



```

        }
    }
    return lista;
}

public ArrayList getInformacoesChamado(int xID){
    ArrayList retorno = new ArrayList();
    Ticket auxTicket;
    Dispositivo auxDispositivo;
    auxTicket = this.copiarTicket(xID);
    String patrr = auxTicket.getPatr_dispositivo();
    auxDispositivo = this.copiarDispositivo(patrr);
    retorno.add(auxDispositivo.getTipo());
    retorno.add(auxDispositivo.getFabricante());
    retorno.add(auxDispositivo.getModelo());
    retorno.add(auxDispositivo.getSetor());
    retorno.add(auxDispositivo.getNro_serie());
    retorno.add(auxTicket.getId()+"");
    retorno.add(auxTicket.getTitulo());
    retorno.add(auxTicket.getDescricao());
    retorno.add(auxTicket.getPatr_dispositivo());
    retorno.add(auxTicket.getCategoria());
    retorno.add(auxTicket.getUrgencia());
    retorno.add(auxTicket.getData_inicio());
    retorno.add(auxTicket.getData_fim());
    retorno.add(auxTicket.getLogin_proprietario());
    retorno.add(auxTicket.getLogin_responsavel());
    if (auxTicket.getAberto())
{retorno.add("true");}else{returno.add("false");}
    retorno.add(auxTicket.getAssistencia());
    retorno.add(auxTicket.getProvidencia());
    return retorno;
}

public boolean alteraChamado(int indice,String tit,String desc,String
p_disp,String cat,String urg,String d_ini,String d_fim,String l_prop,String
l_resp,boolean abe,String ass,String prov){
    boolean foiAlterado = false;
    if (this.existeTicketCadastrado(indice)){
        Ticket aux;
        aux = (Ticket)tickets.get(indice);
        aux.setTitulo(tit);
        aux.setDescricao(desc);
        aux.setPatr_dispositivo(p_disp);
        aux.setCategoria(cat);
        aux.setUrgencia(urg);
        aux.setData_inicio(d_ini);
        aux.setData_fim(d_fim);
        aux.setLogin_proprietario(l_prop);
        aux.setLogin_responsavel(l_resp);
        if (abe){aux.setAberto(true);}else{aux.setAberto(false);}
        aux.setAssistencia(ass);
        aux.setProvidencia(prov);
        tickets.set(indice,aux);
        foiAlterado = true;
        boolean atualizou;
        try {
            atualizou = ticketDAO.updateTicket(aux);
        }
        catch (SQLException e) {

```

```

        e.printStackTrace();
    }
}
return foiAlterado;
}

public boolean fecheChamado(int indice,String cat,String f_fim,String
prov,String ass,boolean abe){
    boolean foiFechado = false;
    if (this.existeTicketCadastrado(indice)){
        Ticket aux;
        aux = (Ticket)tickets.get(indice);
        aux.setCategoria(cat);
        aux.setData_fim(f_fim);
        aux.setProvidencia(prov);
        aux.setAssistencia(ass);
        aux.setAberto(abe);
        tickets.set(indice,aux);
        boolean atualizou;
        try {
            atualizou = ticketDAO.updateTicket(aux);
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
        foiFechado = true;
    }
    return foiFechado;
}

public boolean trateChamado(int indice,String cat,String prov,boolean
abe){
    boolean foiTratado = false;
    if (this.existeTicketCadastrado(indice)){
        Ticket aux;
        aux = (Ticket)tickets.get(indice);
        aux.setCategoria(cat);
        aux.setProvidencia(prov);
        aux.setAberto(abe);
        tickets.set(indice,aux);
        boolean atualizou;
        try {
            atualizou = ticketDAO.updateTicket(aux);
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
        foiTratado = true;
    }
    return foiTratado;
}

public boolean abandoneChamado(int indice,String cat,String
prov,boolean abe,boolean necass){
    boolean foiAbandonado = false;
    if (this.existeTicketCadastrado(indice)){
        Ticket aux;
        aux = (Ticket)tickets.get(indice);
        aux.setCategoria(cat);
        aux.setProvidencia(prov);
        aux.setAberto(abe);
    }
}

```

```

        aux.setNecessita_assistencia(necass);
        tickets.set(indice,aux);
        boolean atualizou;
        try {
            atualizou = ticketDAO.updateTicket(aux);
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
        foiAbandonado = true;
    }
    return foiAbandonado;
}

private int posicaoListaUsuario(String xxl){
    Usuario aux;
    int indice=-1;
    for (int i=0;i<usuarios.size();i++){
        aux = (Usuario)usuarios.get(i);
        if (aux.getLogin().compareTo(xxl)==0){
            indice=i;
        }
    }
    return indice;
}

private int posicaoListaDispositivo(String xxpat){
    Dispositivo aux;
    int indice=-1;
    for (int i=0;i<dispositivos.size();i++){
        aux = (Dispositivo)dispositivos.get(i);
        if (aux.getNro_patrimonio().compareTo(xxpat)==0){
            indice=i;
        }
    }
    return indice;
}

private Ticket copiarTicket(int ident){
    Ticket ticketRetorno = new Ticket();

    for (int i=0;i<tickets.size();i++){
        ticketRetorno = (Ticket)tickets.get(i);
        if ((ticketRetorno.getId()) == ident){
            return ticketRetorno;
        }else{
            ticketRetorno = null;
        }
    }
    return ticketRetorno;
}

private ArrayList getListaTecnicos(){
    ArrayList listaLoginTecnicos = new ArrayList();
    Usuario aux;
    for (int i=0;i<usuarios.size();i++){
        aux = (Usuario)usuarios.get(i);
        if (aux.getNivel() == 2){
            listaLoginTecnicos.add(aux.getLogin());
        }
    }
}

```

```

        return listaLoginTecnicos;
    }

    private String sortearResponsavel(){
        ArrayList lista = this.getListaTecnicos();
        int tamanhoListaTecnicos = lista.size();
        Random aleatorio = new Random();
        int numeroSorteado;
        String responsavel;
        if (tamanhoListaTecnicos != 0) {
            numeroSorteado = aleatorio.nextInt(tamanhoListaTecnicos);
            responsavel=(String)lista.get(numeroSorteado);
        }else{
            numeroSorteado = 0;
            responsavel = "root";
        }
        return responsavel;
    }

    private Usuario copiarUsuario(String xl){
        Usuario usuarioRetorno = new Usuario();
        for (int i=0;i<usuarios.size();i++){
            usuarioRetorno = (Usuario)usuarios.get(i);
            if ((usuarioRetorno.getLogin()).compareTo(xl) == 0){
                return usuarioRetorno;
            }else{
                usuarioRetorno = null;
            }
        }
        return usuarioRetorno;
    }

    private Dispositivo copiarDispositivo(String xnro_pat){
        Dispositivo dispositivoRetorno = new Dispositivo();
        for (int i=0;i<dispositivos.size();i++){
            dispositivoRetorno = (Dispositivo)dispositivos.get(i);
            if
            ((dispositivoRetorno.getNro_patrimonio()).compareTo(xnro_pat) == 0){
                return dispositivoRetorno;
            }else{
                dispositivoRetorno = null;
            }
        }
        return dispositivoRetorno;
    }

    private Assistencia copiarAssistencia(String xnomeasss){
        Assistencia assistenciaRetorno = new Assistencia();
        for (int i=0;i<assistencias.size();i++){
            assistenciaRetorno = (Assistencia)assistencias.get(i);
            if ((assistenciaRetorno.getNome()).compareTo(xnomeasss) == 0){
                return assistenciaRetorno;
            }else{
                assistenciaRetorno = null;
            }
        }
        return assistenciaRetorno;
    }
}

```

```
/******EmpresaInterface.java*****/
```

```
package suatec;
```

```
import java.util.*;
```

```
import java.rmi.Remote;
```

```
import java.rmi.RemoteException;
```

```
public interface EmpresaInterface {
    public boolean acessarSistema(String xlogin, String xsenha);
    public boolean criarUsuario(String xn,String xs,String xc,String
xset,String xm,String xl,String xsenh,int xnv);
    public boolean criarDispositivo(String xpat, String xtip, String xset,
String xfab, String xmod, String xnro);
    public boolean modificarDadosUsuario(String xn,String xs,String
xc,String xset,String xm,String xl,String xsenh,int xnv);
    public boolean modificarDadosDispositivo(String xpat,String
xtipo,String xsetor,String xfabr,String xmodelo,String xnroserie);
    public boolean existeUsuarioCadastrado(String login);
    public boolean existeDispositivoCadastrado(String xpat);
    public ArrayList getListaLoginUsuariosCadastrados();
    public ArrayList getListaDispositivosCadastrados();
    public void excluirUsuario(String xlogin);
    public void excluirDispositivo(String xnro_pat);
    public ArrayList getInformacoesDispositivo(String xpatrr);
    public ArrayList getInformacoesUsuario(String xlogin);
    public ArrayList getInformacoesAssistencia(String xnomeass);
    public void troqueSenha(String l, String s);
    public boolean existeAssistenciaCadastrada(String nome);
    public boolean criarAssistencia(String xn,String xt,String xe,String
xm,String xc);
    public boolean modificarAssistencia(String xn,String xt,String
xe,String xm,String xc);
    public ArrayList getListaAssistenciasCadastradas();
    public boolean excluirAssistencia(String xnome);
    public boolean criarTicket(String xtitulo,String xdescricao,String
xurgencia,String xdata_inicio,String xpatrimonio,String
xlogin_proprietario);
    public ArrayList getListaIdChamadosAbertos();
    public ArrayList getListaIdChamadosAbertosPendentes();
    public boolean excluirTicket(int xindice);
    public boolean existeTicketCadastrado(int xindice);
    public ArrayList getListaIdChamados(boolean ehPatrimonio, String nro);
    public ArrayList getInformacoesChamado(int xID);
    public boolean alteraChamado(int indice,String tit,String desc,String
p_disp,String cat,String urg,String d_ini,String d_fim,String l_prop,String
l_resp,boolean abe,String ass,String prov);
    public boolean fechaChamado(int indice,String cat,String f_fim,String
prov,String ass,boolean abe);
    public boolean trateChamado(int indice,String cat,String prov,boolean
abe);
    public boolean abandoneChamado(int indice,String cat,String
prov,boolean abe,boolean necass);
    public boolean ehTecnico(String xlogin);
}
```

```
/******GenericDAO.java*****/
```

```
package suatec;
```

```

import java.beans.Statement;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Vector;

public abstract class GenericDAO {
    protected String tabela;
    protected static Vector connections;
    protected int indice;

    public GenericDAO() {
        this.initialize();
    }

    public abstract void initialize();
    public abstract void setIndice(int indice);
    public abstract void setTabela(String tabela_);

    public static int addConnection(Connection connection) {
        if (connections == null) {
            connections = new Vector(5,2);
        }
        connections.add(connection);
        return connections.size()-1;
    }

    public static Connection getConnection(int indice) {
        return (Connection)connections.get(indice);
    }

    public static void removeConnection(int indice) {
        connections.remove(indice);
    }

    protected String montarListaCampos( ArrayList lista) {
        Iterator it = lista.iterator();
        String res = "";
        if (it.hasNext()) {
            res = (String)it.next();
            while (it.hasNext()) {
                res += ", " + (String)it.next();
            }
        }
        return res;
    }

    protected PreparedStatement finderSelect(Statement st, ArrayList
campos_chave, String condicao, Connection connection)
throws SQLException {
        String query = "SELECT " + montarListaCampos(campos_chave) +
" FROM " + tabela +
" WHERE " + condicao;
        return connection.prepareStatement(query);
    }
}

/*****MysqlConnection.java*****/

```

```

package suatec;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class MySqlConnection {
    private Connection      con;
    private String          dbUserName = "root";
    private String          dbPassword = "";
    private String          dbUrl      = "jdbc:mysql://localhost/ws";

    public MySqlConnection() {
        this.initialize();
    }

    public MySqlConnection(String pathProperties) {
        this.initialize();
    }

    protected void initialize() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(this.dbUrl, this.dbUserName,
this.dbPassword);
        }
        catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public Connection getConnection() {
        return con;
    }
}

```

```

/*****Pas.java*****/

```

```

package suatec;

import javax.swing.*;
import java.util.*;
public class Pas extends javax.swing.JFrame {

    Empresa ioesc;
    String usuarioLogado;

    public Pas() {
        initComponents();
        this.limparInterface();
        this.limparFrame();
        jPanelIdentificacao.setVisible(true);
        ioesc = new Empresa();
        jTextField1.setText("root");
        jPasswordField1.setText("root");
    }
}

```

```
private void limparInterface(){
    jPanelAbrirChamado.setVisible(false);
    jPanelListaDeChamados.setVisible(false);
    jPanelPesquisarChamado.setVisible(false);
    jPanelCadastroDispositivo.setVisible(false);
    jPanelPesquisarDispositivo.setVisible(false);
    jPanelListaDosDispositivos.setVisible(false);
    jPanelTrocarSenha.setVisible(false);
    jPanelCadastroUsuario.setVisible(false);
    jPanelListaDeUsuarios.setVisible(false);
    jPanelCadastroAssistencia.setVisible(false);
    jPanelListaDasAssistencias.setVisible(false);
    jPanelChamado.setVisible(false);
    jPanelIdentificacao.setVisible(false);
}

private void limparFrame(){
    jButton4.setVisible(false);
    jButton5.setVisible(false);
    jButton6.setVisible(false);
    jButton7.setVisible(false);
    jButton8.setVisible(false);
    jButton9.setVisible(false);
    jButton10.setVisible(false);
    jButton15.setVisible(false);
    jButton12.setVisible(false);
    jButton13.setVisible(false);
    jButton22.setVisible(false);
    botaoLogoff.setVisible(false);
    jLabel13.setVisible(false);
    jLabel15.setVisible(false);
    jLabel16.setVisible(false);
    jLabel17.setVisible(false);
    jLabel18.setVisible(false);
    jLabel19.setVisible(false);
}

private void disponibilizarFrame(){
    jButton4.setVisible(true);
    jButton5.setVisible(true);
    jButton6.setVisible(true);
    jButton7.setVisible(true);
    jButton8.setVisible(true);
    jButton9.setVisible(true);
    jButton10.setVisible(true);
    jButton15.setVisible(true);
    jButton12.setVisible(true);
    jButton13.setVisible(true);
    jButton22.setVisible(true);
    botaoLogoff.setVisible(true);
    jLabel13.setVisible(true);
    jLabel15.setVisible(true);
    jLabel16.setVisible(true);
    jLabel17.setVisible(true);
    jLabel18.setVisible(true);
    jLabel19.setVisible(true);
}

private void limparTelaCadastroUsuario()
{
```



```

        jTextField14.setText("");
        jTextField15.setText("");
        jTextField16.setText("");
        jTextField17.setText("");
        jTextField18.setText("");
        jTextField19.setText("");
        jPasswordField5.setText("");
        jPasswordField6.setText("");
        radioButton1.setSelected(false);
        radioButton2.setSelected(false);
        radioButton3.setSelected(false);
    }

    private void limparTelaCadastroAssistencia(){
        jTextField22.setText("");
        jTextField23.setText("");
        jTextField24.setText("");
        jTextField25.setText("");
        jTextField26.setText("");
    }

    private void limparTelaCadastroEquipamento(){
        jTextField4.setText("");
        jTextField8.setText("");
        jTextField9.setText("");
        jTextField10.setText("");
        jTextField11.setText("");
        jTextField12.setText("");
    }

    private void limparTelaAberturaChamado(){
        jTextField2.setText("");
        jTextArea1.setText("");
        jTextField7.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
    }

    private void limparTelaVerChamado(){
        jLabel61.setText("");
        jTextField27.setText("");
        jTextArea3.setText("");
        jTextField29.setText("");
        jTextField30.setText("");
        jTextField31.setText("");
        jTextField32.setText("");
        jTextField33.setText("");
        jTextField34.setText("");
        jTextField35.setText("");
        jCheckBox1.setContentAreaFilled(false);
        jTextArea2.setText("");
        jComboBox4.removeAllItems();
        jLabel20.setText("");
        jLabel32.setText("");
        jLabel46.setText("");
        jLabel47.setText("");
        jLabel49.setText("");
    }

    private void initComponents() { //GEN-BEGIN: initComponents
        jPanelIdentificacao = new javax.swing.JPanel();

```

```
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jPasswordField1 = new javax.swing.JPasswordField();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jLabel6 = new javax.swing.JLabel();
jButton7 = new javax.swing.JButton();
jButton8 = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jButton9 = new javax.swing.JButton();
jButton10 = new javax.swing.JButton();
jButton15 = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
jButton12 = new javax.swing.JButton();
jButton13 = new javax.swing.JButton();
botaoLogoff = new javax.swing.JButton();
jButton22 = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();
jPanelAbrirChamado = new javax.swing.JPanel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jTextArea1 = new javax.swing.JTextArea();
jTextField5 = new javax.swing.JTextField();
jTextField6 = new javax.swing.JTextField();
jButton16 = new javax.swing.JButton();
jButton17 = new javax.swing.JButton();
jTextField7 = new javax.swing.JTextField();
jPanelPesquisarChamado = new javax.swing.JPanel();
jButton19 = new javax.swing.JButton();
jButton20 = new javax.swing.JButton();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jComboBox6 = new javax.swing.JComboBox();
jTextField3 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jPanelListaDeChamados = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jButton18 = new javax.swing.JButton();
jLabel15 = new javax.swing.JLabel();
jComboBox5 = new javax.swing.JComboBox();
jPanelChamado = new javax.swing.JPanel();
jButton48 = new javax.swing.JButton();
jButton49 = new javax.swing.JButton();
jButton50 = new javax.swing.JButton();
jButton51 = new javax.swing.JButton();
jLabel59 = new javax.swing.JLabel();
jLabel60 = new javax.swing.JLabel();
jLabel61 = new javax.swing.JLabel();
jLabel62 = new javax.swing.JLabel();
```

```
jLabel63 = new javax.swing.JLabel();
jLabel64 = new javax.swing.JLabel();
jLabel65 = new javax.swing.JLabel();
jLabel66 = new javax.swing.JLabel();
jLabel67 = new javax.swing.JLabel();
jLabel68 = new javax.swing.JLabel();
jLabel69 = new javax.swing.JLabel();
jLabel70 = new javax.swing.JLabel();
jLabel71 = new javax.swing.JLabel();
jLabel72 = new javax.swing.JLabel();
jLabel73 = new javax.swing.JLabel();
jLabel74 = new javax.swing.JLabel();
jLabel75 = new javax.swing.JLabel();
jLabel76 = new javax.swing.JLabel();
jLabel77 = new javax.swing.JLabel();
jLabel78 = new javax.swing.JLabel();
jLabel79 = new javax.swing.JLabel();
jTextField27 = new javax.swing.JTextField();
jTextField29 = new javax.swing.JTextField();
jTextField30 = new javax.swing.JTextField();
jTextField31 = new javax.swing.JTextField();
jTextField32 = new javax.swing.JTextField();
jTextField33 = new javax.swing.JTextField();
jTextField34 = new javax.swing.JTextField();
jTextField35 = new javax.swing.JTextField();
jTextArea2 = new javax.swing.JTextArea();
jTextArea3 = new javax.swing.JTextArea();
jCheckBox1 = new javax.swing.JCheckBox();
jComboBox4 = new javax.swing.JComboBox();
jLabel20 = new javax.swing.JLabel();
jLabel32 = new javax.swing.JLabel();
jLabel46 = new javax.swing.JLabel();
jLabel47 = new javax.swing.JLabel();
jLabel49 = new javax.swing.JLabel();
jButton21 = new javax.swing.JButton();
jButton23 = new javax.swing.JButton();
jPanelCadastroDispositivo = new javax.swing.JPanel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jButton24 = new javax.swing.JButton();
jButton25 = new javax.swing.JButton();
jButton26 = new javax.swing.JButton();
jButton14 = new javax.swing.JButton();
jPanelPesquisarDispositivo = new javax.swing.JPanel();
jButton27 = new javax.swing.JButton();
jButton28 = new javax.swing.JButton();
jLabel28 = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
jTextField13 = new javax.swing.JTextField();
jPanelListaDosDispositivos = new javax.swing.JPanel();
```

```
jButton29 = new javax.swing.JButton();
jButton30 = new javax.swing.JButton();
jLabel30 = new javax.swing.JLabel();
jComboBox3 = new javax.swing.JComboBox();
jPanelCadastroUsuario = new javax.swing.JPanel();
jLabel35 = new javax.swing.JLabel();
jLabel36 = new javax.swing.JLabel();
jLabel37 = new javax.swing.JLabel();
jLabel38 = new javax.swing.JLabel();
jLabel39 = new javax.swing.JLabel();
jLabel40 = new javax.swing.JLabel();
jLabel41 = new javax.swing.JLabel();
jLabel42 = new javax.swing.JLabel();
jLabel43 = new javax.swing.JLabel();
jLabel44 = new javax.swing.JLabel();
jButton33 = new javax.swing.JButton();
jButton34 = new javax.swing.JButton();
jButton35 = new javax.swing.JButton();
jButton52 = new javax.swing.JButton();
jTextField14 = new javax.swing.JTextField();
jTextField15 = new javax.swing.JTextField();
jTextField16 = new javax.swing.JTextField();
jTextField17 = new javax.swing.JTextField();
jTextField18 = new javax.swing.JTextField();
jTextField19 = new javax.swing.JTextField();
jPasswordField5 = new javax.swing.JPasswordField();
jPasswordField6 = new javax.swing.JPasswordField();
jRadioButton1 = new javax.swing.JRadioButton();
jRadioButton2 = new javax.swing.JRadioButton();
jRadioButton3 = new javax.swing.JRadioButton();
jPanelListaDeUsuarios = new javax.swing.JPanel();
jLabel45 = new javax.swing.JLabel();
jButton36 = new javax.swing.JButton();
jButton37 = new javax.swing.JButton();
jComboBox1 = new javax.swing.JComboBox();
jPanelCadastroAssistencia = new javax.swing.JPanel();
jLabel48 = new javax.swing.JLabel();
jLabel50 = new javax.swing.JLabel();
jLabel51 = new javax.swing.JLabel();
jLabel53 = new javax.swing.JLabel();
jLabel54 = new javax.swing.JLabel();
jLabel55 = new javax.swing.JLabel();
jTextField22 = new javax.swing.JTextField();
jTextField23 = new javax.swing.JTextField();
jTextField24 = new javax.swing.JTextField();
jTextField25 = new javax.swing.JTextField();
jTextField26 = new javax.swing.JTextField();
jButton41 = new javax.swing.JButton();
jButton42 = new javax.swing.JButton();
jButton43 = new javax.swing.JButton();
jButton11 = new javax.swing.JButton();
jPanelListaDasAssistencias = new javax.swing.JPanel();
jLabel56 = new javax.swing.JLabel();
jButton44 = new javax.swing.JButton();
jButton45 = new javax.swing.JButton();
jComboBox2 = new javax.swing.JComboBox();
jPanelTrocarSenha = new javax.swing.JPanel();
jLabel31 = new javax.swing.JLabel();
jLabel33 = new javax.swing.JLabel();
jLabel34 = new javax.swing.JLabel();
jButton31 = new javax.swing.JButton();
```

```

jButton32 = new javax.swing.JButton();
jPasswordField3 = new javax.swing.JPasswordField();
jPasswordField4 = new javax.swing.JPasswordField();

getContentPane().setLayout(null);

setTitle("Cadastra");
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
        exitForm(evt);
    }
});

jPanelIdentificacao.setLayout(null);

jPanelIdentificacao.setPreferredSize(new java.awt.Dimension(600,
600));
jLabel1.setText("Login:");
jPanelIdentificacao.add(jLabel1);
jLabel1.setBounds(40, 50, 40, 15);

jLabel2.setText("Senha:");
jLabel2.setRequestFocusEnabled(false);
jPanelIdentificacao.add(jLabel2);
jLabel2.setBounds(40, 80, 40, 15);

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jTextField1);
jTextField1.setBounds(90, 50, 80, 21);

jPasswordField1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jPasswordField1ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jPasswordField1);
jPasswordField1.setBounds(90, 80, 80, 20);

jButton1.setLabel("OK");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jButton1);
jButton1.setBounds(40, 130, 80, 25);

jButton2.setLabel("Cancelar");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

```

```

jPanelIdentificacao.add(jButton2);
jButton2.setBounds(140, 130, 90, 25);

getContentPane().add(jPanelIdentificacao);
jPanelIdentificacao.setBounds(220, 40, 380, 320);

jPanelIdentificacao.getAccessibleContext().setAccessibleParent(jPanelIdentificacao);

jLabel3.setText("Administrador");
getContentPane().add(jLabel3);
jLabel3.setBounds(0, 0, 110, 15);

jLabel5.setText("Chamado");
getContentPane().add(jLabel5);
jLabel5.setBounds(0, 50, 70, 15);

jButton4.setLabel("Abrir");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

getContentPane().add(jButton4);
jButton4.setBounds(10, 70, 130, 20);

jButton5.setLabel("Listar Abertos");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});

getContentPane().add(jButton5);
jButton5.setBounds(10, 90, 130, 20);

jButton6.setLabel("Pesquisar");
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});

getContentPane().add(jButton6);
jButton6.setBounds(10, 110, 130, 20);

jLabel6.setText("Dispositivo");
getContentPane().add(jLabel6);
jLabel6.setBounds(0, 160, 80, 15);

jButton7.setLabel("Cadastrar");
jButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton7ActionPerformed(evt);
    }
});

getContentPane().add(jButton7);
jButton7.setBounds(10, 180, 130, 20);

```

```

jButton8.setLabel("Pesquisar");
jButton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton8ActionPerformed(evt);
    }
});

getContentPane().add(jButton8);
jButton8.setBounds(10, 200, 130, 20);

jLabel7.setText("Usu\u00e9rio");
getContentPane().add(jLabel7);
jLabel7.setBounds(0, 230, 60, 15);

jButton9.setLabel("Cadastrar");
jButton9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton9ActionPerformed(evt);
    }
});

getContentPane().add(jButton9);
jButton9.setBounds(10, 250, 130, 20);

jButton10.setLabel("Listar");
jButton10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton10ActionPerformed(evt);
    }
});

getContentPane().add(jButton10);
jButton10.setBounds(10, 270, 130, 20);

jButton15.setLabel("Trocar");
jButton15.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton15ActionPerformed(evt);
    }
});

getContentPane().add(jButton15);
jButton15.setBounds(10, 390, 130, 20);

jLabel8.setText("Assit\u00eancia");
getContentPane().add(jLabel8);
jLabel8.setBounds(0, 300, 60, 15);

jButton12.setLabel("Cadastrar");
jButton12.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton12ActionPerformed(evt);
    }
});

getContentPane().add(jButton12);
jButton12.setBounds(10, 320, 130, 20);

jButton13.setLabel("Listar");
jButton13.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton13ActionPerformed(evt);
        }
    });

    getContentPane().add(jButton13);
    jButton13.setBounds(10, 340, 130, 20);

    botaoLogoff.setText("Logoff");
    botaoLogoff.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoLogoffActionPerformed(evt);
        }
    });

    getContentPane().add(botaoLogoff);
    botaoLogoff.setBounds(10, 430, 130, 25);

    jButton22.setText("Pendentes");
    jButton22.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton22ActionPerformed(evt);
        }
    });

    getContentPane().add(jButton22);
    jButton22.setBounds(10, 130, 130, 20);

    jLabel9.setText("Senha");
    getContentPane().add(jLabel9);
    jLabel9.setBounds(0, 370, 80, 15);

    jPanelAbrirChamado.setLayout(null);

    jLabel10.setText("Abrir Chamado");
    jPanelAbrirChamado.add(jLabel10);
    jLabel10.setBounds(220, 10, 100, 15);

    jLabel11.setText("T\u00edtulo:");
    jPanelAbrirChamado.add(jLabel11);
    jLabel11.setBounds(50, 50, 50, 15);

    jLabel12.setText("Descri\u00e7\u00e3o:");
    jPanelAbrirChamado.add(jLabel12);
    jLabel12.setBounds(50, 80, 70, 15);

    jLabel13.setText("Urg\u00eancia:");
    jPanelAbrirChamado.add(jLabel13);
    jLabel13.setBounds(50, 180, 70, 15);

    jLabel14.setText("Data:");
    jPanelAbrirChamado.add(jLabel14);
    jLabel14.setBounds(50, 220, 40, 15);

    jLabel18.setText("Dispositivo");
    jPanelAbrirChamado.add(jLabel18);
    jLabel18.setBounds(50, 250, 90, 15);

    jLabel19.setText("Patrimonio:");
    jPanelAbrirChamado.add(jLabel19);
    jLabel19.setBounds(80, 270, 80, 15);

```



```

jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jTextField2);
jTextField2.setBounds(130, 50, 200, 20);

jPanelAbrirChamado.add(jTextArea1);
jTextArea1.setBounds(130, 80, 200, 70);

jPanelAbrirChamado.add(jTextField5);
jTextField5.setBounds(130, 220, 200, 21);

jTextField6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField6ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jTextField6);
jTextField6.setBounds(150, 270, 110, 21);

jButton16.setText("Criar");
jButton16.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton16ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jButton16);
jButton16.setBounds(100, 330, 110, 25);

jButton17.setText("Sair");
jButton17.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton17ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jButton17);
jButton17.setBounds(250, 330, 100, 25);

jPanelAbrirChamado.add(jTextField7);
jTextField7.setBounds(130, 170, 200, 21);

getContentPane().add(jPanelAbrirChamado);
jPanelAbrirChamado.setBounds(150, -10, 620, 550);

jPanelPesquisarChamado.setLayout(null);

jButton19.setText("Procurar");
jButton19.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton19ActionPerformed(evt);
    }
});

jPanelPesquisarChamado.add(jButton19);

```

```

jButton19.setBounds(120, 340, 120, 25);

jButton20.setText("Sair");
jButton20.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton20ActionPerformed(evt);
    }
});

jPanelPesquisarChamado.add(jButton20);
jButton20.setBounds(270, 340, 110, 25);

jLabel16.setText("Pesquisar Chamado");
jPanelPesquisarChamado.add(jLabel16);
jLabel16.setBounds(300, 20, 140, 15);

jLabel17.setText("Tipo:");
jPanelPesquisarChamado.add(jLabel17);
jLabel17.setBounds(100, 80, 70, 15);

jPanelPesquisarChamado.add(jComboBox6);
jComboBox6.setBounds(150, 80, 230, 21);

jPanelPesquisarChamado.add(jTextField3);
jTextField3.setBounds(150, 140, 230, 21);

jLabel4.setText("Patrimonio / N\u00b0 Ticket");
jPanelPesquisarChamado.add(jLabel4);
jLabel4.setBounds(10, 140, 140, 15);

getContentPane().add(jPanelPesquisarChamado);
jPanelPesquisarChamado.setBounds(150, 0, 620, 620);

jPanelListaDeChamados.setLayout(null);

jButton3.setText("Ver");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jPanelListaDeChamados.add(jButton3);
jButton3.setBounds(220, 200, 100, 25);

jButton18.setText("Sair");
jButton18.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton18ActionPerformed(evt);
    }
});

jPanelListaDeChamados.add(jButton18);
jButton18.setBounds(380, 200, 110, 25);

jLabel15.setText("Lista de Chamados");
jPanelListaDeChamados.add(jLabel15);
jLabel15.setBounds(290, 30, 150, 15);

jPanelListaDeChamados.add(jComboBox5);
jComboBox5.setBounds(60, 100, 550, 21);

```

```

getContentPane().add(jPanelListaDeChamados);
jPanelListaDeChamados.setBounds(150, 0, 620, 540);

jPanelChamado.setLayout(null);

jButton48.setText("Alterar");
jButton48.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton48ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton48);
jButton48.setBounds(160, 550, 130, 25);

jButton49.setText("Fechar Chamado");
jButton49.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton49ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton49);
jButton49.setBounds(160, 580, 130, 25);

jButton50.setText("Sair");
jButton50.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton50ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton50);
jButton50.setBounds(410, 550, 100, 25);

jButton51.setText("Excluir");
jButton51.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton51ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton51);
jButton51.setBounds(300, 580, 100, 25);

jLabel59.setText("Chamado");
jPanelChamado.add(jLabel59);
jLabel59.setBounds(210, 0, 70, 15);

jLabel60.setText("nro.");
jPanelChamado.add(jLabel60);
jLabel60.setBounds(10, 20, 50, 15);

jPanelChamado.add(jLabel61);
jLabel61.setBounds(50, 20, 60, 15);

jLabel62.setText("T\u00edtulo:");
jPanelChamado.add(jLabel62);
jLabel62.setBounds(10, 40, 50, 15);

```

```
jLabel63.setText("Descri\u00e7\u00e3o:");
jPanelChamado.add(jLabel63);
jLabel63.setBounds(10, 70, 70, 15);

jLabel64.setText("Dispositivo");
jPanelChamado.add(jLabel64);
jLabel64.setBounds(10, 150, 70, 15);

jLabel65.setText("Tipo:");
jPanelChamado.add(jLabel65);
jLabel65.setBounds(50, 170, 70, 15);

jLabel66.setText("Marca:");
jPanelChamado.add(jLabel66);
jLabel66.setBounds(50, 190, 70, 15);

jLabel67.setText("Modelo:");
jPanelChamado.add(jLabel67);
jLabel67.setBounds(50, 210, 70, 15);

jLabel68.setText("Setor:");
jPanelChamado.add(jLabel68);
jLabel68.setBounds(50, 230, 60, 15);

jLabel69.setText("Patrim\u00f4nio:");
jPanelChamado.add(jLabel69);
jLabel69.setBounds(50, 250, 80, 15);

jLabel70.setText("Nro. S\u00e9rie:");
jPanelChamado.add(jLabel70);
jLabel70.setBounds(50, 270, 80, 15);

jLabel71.setText("Data abertura:");
jPanelChamado.add(jLabel71);
jLabel71.setBounds(10, 330, 90, 15);

jLabel72.setText("Data conclus\u00e3o:");
jPanelChamado.add(jLabel72);
jLabel72.setBounds(10, 350, 150, 15);

jLabel73.setText("Propriet\u00e1rio:");
jPanelChamado.add(jLabel73);
jLabel73.setBounds(10, 370, 90, 15);

jLabel74.setText("T\u00e9cnico respons\u00e1vel:");
jPanelChamado.add(jLabel74);
jLabel74.setBounds(10, 400, 140, 15);

jLabel75.setText("Status:");
jPanelChamado.add(jLabel75);
jLabel75.setBounds(10, 420, 100, 15);

jLabel76.setText("Provid\u00eancias:");
jPanelChamado.add(jLabel76);
jLabel76.setBounds(10, 450, 120, 15);

jLabel77.setText("Assist\u00eancia:");
jPanelChamado.add(jLabel77);
jLabel77.setBounds(10, 510, 110, 15);

jLabel78.setText("Categoria do problema:");
```

```
jPanelChamado.add(jLabel78);
jLabel78.setBounds(10, 290, 150, 15);

jLabel79.setText("Urgencia:");
jPanelChamado.add(jLabel79);
jLabel79.setBounds(10, 310, 90, 15);

jPanelChamado.add(jTextField27);
jTextField27.setBounds(70, 40, 160, 20);

jPanelChamado.add(jTextField29);
jTextField29.setBounds(130, 250, 310, 21);

jPanelChamado.add(jTextField30);
jTextField30.setBounds(160, 290, 280, 21);

jPanelChamado.add(jTextField31);
jTextField31.setBounds(90, 310, 340, 21);

jPanelChamado.add(jTextField32);
jTextField32.setBounds(120, 330, 190, 21);

jPanelChamado.add(jTextField33);
jTextField33.setBounds(120, 350, 190, 21);

jPanelChamado.add(jTextField34);
jTextField34.setBounds(120, 370, 190, 21);

jPanelChamado.add(jTextField35);
jTextField35.setBounds(150, 400, 160, 21);

jPanelChamado.add(jTextArea2);
jTextArea2.setBounds(90, 450, 480, 50);

jPanelChamado.add(jTextArea3);
jTextArea3.setBounds(80, 70, 450, 80);

jCheckBox1.setText("Aberto");
jPanelChamado.add(jCheckBox1);
jCheckBox1.setBounds(80, 420, 110, 23);

jPanelChamado.add(jComboBox4);
jComboBox4.setBounds(90, 510, 230, 21);

jPanelChamado.add(jLabel20);
jLabel20.setBounds(90, 160, 170, 20);

jPanelChamado.add(jLabel32);
jLabel32.setBounds(100, 180, 160, 20);

jPanelChamado.add(jLabel46);
jLabel46.setBounds(110, 200, 160, 20);

jPanelChamado.add(jLabel47);
jLabel47.setBounds(100, 220, 190, 20);

jPanelChamado.add(jLabel49);
jLabel49.setBounds(120, 270, 140, 20);

jButton21.setText("Abandonar");
jButton21.addActionListener(new java.awt.event.ActionListener() {
```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton21ActionPerformed(evt);
        }
    });

    jPanelChamado.add(jButton21);
    jButton21.setBounds(410, 580, 100, 25);

    jButton23.setText("Tratar");
    jButton23.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton23ActionPerformed(evt);
        }
    });

    jPanelChamado.add(jButton23);
    jButton23.setBounds(300, 550, 100, 25);

    getContentPane().add(jPanelChamado);
    jPanelChamado.setBounds(150, 0, 640, 620);

    jPanelCadastroDispositivo.setLayout(null);

    jLabel21.setText("Cadastro de Dispositivo");
    jPanelCadastroDispositivo.add(jLabel21);
    jLabel21.setBounds(280, 10, 170, 15);

    jLabel22.setText("Nro. Patrimonio:");
    jPanelCadastroDispositivo.add(jLabel22);
    jLabel22.setBounds(20, 50, 140, 15);

    jLabel23.setText("Tipo:");
    jPanelCadastroDispositivo.add(jLabel23);
    jLabel23.setBounds(20, 80, 80, 15);

    jLabel24.setText("Setor:");
    jPanelCadastroDispositivo.add(jLabel24);
    jLabel24.setBounds(20, 110, 60, 15);

    jLabel25.setText("Fabricante:");
    jPanelCadastroDispositivo.add(jLabel25);
    jLabel25.setBounds(20, 140, 70, 15);

    jLabel26.setText("Modelo:");
    jPanelCadastroDispositivo.add(jLabel26);
    jLabel26.setBounds(20, 170, 60, 15);

    jLabel27.setText("Nro. S\u00e9rie:");
    jPanelCadastroDispositivo.add(jLabel27);
    jLabel27.setBounds(20, 200, 70, 15);

    jPanelCadastroDispositivo.add(jTextField4);
    jTextField4.setBounds(120, 50, 140, 21);

    jPanelCadastroDispositivo.add(jTextField8);
    jTextField8.setBounds(70, 80, 190, 21);

    jPanelCadastroDispositivo.add(jTextField9);
    jTextField9.setBounds(70, 110, 190, 21);

    jPanelCadastroDispositivo.add(jTextField10);

```

```

jTextField10.setBounds(100, 140, 160, 21);

jPanelCadastroDispositivo.add(jTextField11);
jTextField11.setBounds(80, 170, 180, 21);

jPanelCadastroDispositivo.add(jTextField12);
jTextField12.setBounds(100, 200, 160, 21);

jButton24.setText("Criar");
jButton24.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton24ActionPerformed(evt);
    }
});

jPanelCadastroDispositivo.add(jButton24);
jButton24.setBounds(20, 270, 130, 25);

jButton25.setText("Excluir");
jButton25.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton25ActionPerformed(evt);
    }
});

jPanelCadastroDispositivo.add(jButton25);
jButton25.setBounds(330, 270, 120, 25);

jButton26.setText("Sair");
jButton26.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton26ActionPerformed(evt);
    }
});

jPanelCadastroDispositivo.add(jButton26);
jButton26.setBounds(480, 270, 120, 25);

jButton14.setText("Modificar");
jButton14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton14ActionPerformed(evt);
    }
});

jPanelCadastroDispositivo.add(jButton14);
jButton14.setBounds(170, 270, 140, 25);

getContentPane().add(jPanelCadastroDispositivo);
jPanelCadastroDispositivo.setBounds(150, 0, 620, 540);

jPanelPesquisarDispositivo.setLayout(null);

jButton27.setText("Pesquisar");
jButton27.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton27ActionPerformed(evt);
    }
});

jPanelPesquisarDispositivo.add(jButton27);

```

```

jButton27.setBounds(80, 200, 130, 25);

jButton28.setText("Sair");
jButton28.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton28ActionPerformed(evt);
    }
});

jPanelPesquisarDispositivo.add(jButton28);
jButton28.setBounds(270, 200, 140, 25);

jLabel28.setText("Pesquisar dispositivo");
jPanelPesquisarDispositivo.add(jLabel28);
jLabel28.setBounds(200, 30, 160, 15);

jLabel29.setText("Nro. patrimonio:");
jPanelPesquisarDispositivo.add(jLabel29);
jLabel29.setBounds(20, 90, 120, 15);

jPanelPesquisarDispositivo.add(jTextField13);
jTextField13.setBounds(120, 90, 200, 21);

getContentPane().add(jPanelPesquisarDispositivo);
jPanelPesquisarDispositivo.setBounds(150, 0, 640, 620);

jPanelListaDosDispositivos.setLayout(null);

jButton29.setText("Ver");
jButton29.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton29ActionPerformed(evt);
    }
});

jPanelListaDosDispositivos.add(jButton29);
jButton29.setBounds(90, 170, 100, 25);

jButton30.setText("Sair");
jButton30.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton30ActionPerformed(evt);
    }
});

jPanelListaDosDispositivos.add(jButton30);
jButton30.setBounds(260, 170, 110, 25);

jLabel30.setText("Lista dos Dispositivos");
jPanelListaDosDispositivos.add(jLabel30);
jLabel30.setBounds(180, 20, 190, 15);

jPanelListaDosDispositivos.add(jComboBox3);
jComboBox3.setBounds(90, 100, 280, 21);

getContentPane().add(jPanelListaDosDispositivos);
jPanelListaDosDispositivos.setBounds(150, 0, 640, 620);

jPanelCadastroUsuario.setLayout(null);

jLabel35.setText("Cadastro Usu\u00e1rio");

```



```

jPanelCadastroUsuario.add(jLabel35);
jLabel35.setBounds(180, 20, 210, 15);

jLabel36.setText("Nome:");
jPanelCadastroUsuario.add(jLabel36);
jLabel36.setBounds(30, 60, 60, 15);

jLabel37.setText("Sobrenome:");
jPanelCadastroUsuario.add(jLabel37);
jLabel37.setBounds(30, 90, 80, 15);

jLabel38.setText("Cargo:");
jPanelCadastroUsuario.add(jLabel38);
jLabel38.setBounds(30, 120, 60, 15);

jLabel39.setText("Setor:");
jPanelCadastroUsuario.add(jLabel39);
jLabel39.setBounds(30, 150, 60, 15);

jLabel40.setText("Matr\u00edcula:");
jPanelCadastroUsuario.add(jLabel40);
jLabel40.setBounds(30, 180, 70, 15);

jLabel41.setText("Login:");
jPanelCadastroUsuario.add(jLabel41);
jLabel41.setBounds(30, 210, 70, 15);

jLabel42.setText("Senha:");
jPanelCadastroUsuario.add(jLabel42);
jLabel42.setBounds(30, 240, 70, 15);

jLabel43.setText("Repetir Senha:");
jPanelCadastroUsuario.add(jLabel43);
jLabel43.setBounds(30, 270, 90, 15);

jLabel44.setText("N\u00edvel:");
jPanelCadastroUsuario.add(jLabel44);
jLabel44.setBounds(30, 300, 60, 15);

jButton33.setText("Criar");
jButton33.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton33ActionPerformed(evt);
    }
});

jPanelCadastroUsuario.add(jButton33);
jButton33.setBounds(79, 400, 100, 25);

jButton34.setText("Excluir");
jButton34.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton34ActionPerformed(evt);
    }
});

jPanelCadastroUsuario.add(jButton34);
jButton34.setBounds(190, 400, 120, 25);

jButton35.setText("Sair");
jButton35.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton35ActionPerformed(evt);
        }
    });

    jPanelCadaastroUsuario.add(jButton35);
    jButton35.setBounds(450, 400, 110, 25);

    jButton52.setText("Modificar");
    jButton52.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton52ActionPerformed(evt);
        }
    });

    jPanelCadaastroUsuario.add(jButton52);
    jButton52.setBounds(320, 400, 120, 25);

    jPanelCadaastroUsuario.add(jTextField14);
    jTextField14.setBounds(100, 60, 180, 21);

    jPanelCadaastroUsuario.add(jTextField15);
    jTextField15.setBounds(120, 90, 160, 21);

    jPanelCadaastroUsuario.add(jTextField16);
    jTextField16.setBounds(90, 120, 190, 21);

    jPanelCadaastroUsuario.add(jTextField17);
    jTextField17.setBounds(90, 150, 190, 21);

    jPanelCadaastroUsuario.add(jTextField18);
    jTextField18.setBounds(100, 180, 180, 21);

    jPanelCadaastroUsuario.add(jTextField19);
    jTextField19.setBounds(100, 210, 180, 21);

    jPanelCadaastroUsuario.add(jPasswordField5);
    jPasswordField5.setBounds(100, 240, 180, 23);

    jPanelCadaastroUsuario.add(jPasswordField6);
    jPasswordField6.setBounds(120, 270, 160, 23);

    jButton1.setText("Usu\u00e9rio");
    jButton1.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jPanelCadaastroUsuario.add(jButton1);
    jButton1.setBounds(90, 300, 80, 23);

    jButton2.setText("T\u00e9cnico");
    jPanelCadaastroUsuario.add(jButton2);
    jButton2.setBounds(180, 300, 80, 23);

    jButton3.setText("Administrador");
    jPanelCadaastroUsuario.add(jButton3);
    jButton3.setBounds(280, 300, 120, 23);

```

```

getContentPane().add(jPanelCadastroUsuario);
jPanelCadastroUsuario.setBounds(150, 0, 640, 620);

jPanelListaDeUsuarios.setLayout(null);

jLabel45.setText("Lista de Usu\u00e1rios");
jPanelListaDeUsuarios.add(jLabel45);
jLabel45.setBounds(210, 10, 120, 15);

jButton36.setText("Ver");
jButton36.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton36ActionPerformed(evt);
    }
});

jPanelListaDeUsuarios.add(jButton36);
jButton36.setBounds(160, 200, 90, 25);

jButton37.setText("Sair");
jButton37.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton37ActionPerformed(evt);
    }
});

jPanelListaDeUsuarios.add(jButton37);
jButton37.setBounds(320, 200, 90, 25);

jPanelListaDeUsuarios.add(jComboBox1);
jComboBox1.setBounds(160, 90, 250, 20);

getContentPane().add(jPanelListaDeUsuarios);
jPanelListaDeUsuarios.setBounds(150, 0, 640, 620);

jPanelCadastroAssistencia.setLayout(null);

jLabel48.setText("Cadastro Assistencia");
jPanelCadastroAssistencia.add(jLabel48);
jLabel48.setBounds(200, 20, 170, 15);

jLabel50.setText("Nome:");
jPanelCadastroAssistencia.add(jLabel50);
jLabel50.setBounds(30, 100, 50, 15);

jLabel51.setText("Telefone:");
jPanelCadastroAssistencia.add(jLabel51);
jLabel51.setBounds(30, 130, 100, 15);

jLabel53.setText("Endere\u00e7o:");
jPanelCadastroAssistencia.add(jLabel53);
jLabel53.setBounds(30, 160, 90, 15);

jLabel54.setText("E-mail:");
jPanelCadastroAssistencia.add(jLabel54);
jLabel54.setBounds(30, 190, 80, 15);

jLabel55.setText("Contato:");
jPanelCadastroAssistencia.add(jLabel55);
jLabel55.setBounds(30, 220, 70, 15);

```

```

jPanelCadastroAssistencia.add(jTextField22);
jTextField22.setBounds(100, 100, 140, 21);

jPanelCadastroAssistencia.add(jTextField23);
jTextField23.setBounds(100, 130, 140, 21);

jPanelCadastroAssistencia.add(jTextField24);
jTextField24.setBounds(100, 160, 140, 21);

jPanelCadastroAssistencia.add(jTextField25);
jTextField25.setBounds(100, 190, 140, 21);

jPanelCadastroAssistencia.add(jTextField26);
jTextField26.setBounds(100, 220, 140, 21);

jButton41.setText("Criar");
jButton41.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton41ActionPerformed(evt);
    }
});

jPanelCadastroAssistencia.add(jButton41);
jButton41.setBounds(80, 310, 100, 25);

jButton42.setText("Excluir");
jButton42.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton42ActionPerformed(evt);
    }
});

jPanelCadastroAssistencia.add(jButton42);
jButton42.setBounds(370, 310, 110, 25);

jButton43.setText("Sair");
jButton43.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton43ActionPerformed(evt);
    }
});

jPanelCadastroAssistencia.add(jButton43);
jButton43.setBounds(510, 310, 110, 25);

jButton11.setText("Modificar");
jButton11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton11ActionPerformed(evt);
    }
});

jPanelCadastroAssistencia.add(jButton11);
jButton11.setBounds(210, 310, 130, 25);

getContentPane().add(jPanelCadastroAssistencia);
jPanelCadastroAssistencia.setBounds(150, 0, 640, 620);

jPanelListaDasAssistencias.setLayout(null);

jLabel56.setText("Assistencias");

```

```

jPanelListaDasAssistencias.add(jLabel56);
jLabel56.setBounds(200, 10, 110, 15);

jButton44.setText("Abrir");
jButton44.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton44ActionPerformed(evt);
    }
});

jPanelListaDasAssistencias.add(jButton44);
jButton44.setBounds(110, 230, 100, 25);

jButton45.setText("Sair");
jButton45.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton45ActionPerformed(evt);
    }
});

jPanelListaDasAssistencias.add(jButton45);
jButton45.setBounds(283, 230, 110, 25);

jPanelListaDasAssistencias.add(jComboBox2);
jComboBox2.setBounds(110, 100, 290, 21);

getContentPane().add(jPanelListaDasAssistencias);
jPanelListaDasAssistencias.setBounds(150, 0, 640, 620);

jPanelTrocarSenha.setLayout(null);

jLabel31.setText("Trocar senha");
jPanelTrocarSenha.add(jLabel31);
jLabel31.setBounds(230, 20, 150, 15);

jLabel33.setText("Nova senha:");
jPanelTrocarSenha.add(jLabel33);
jLabel33.setBounds(40, 110, 120, 15);

jLabel34.setText("Repetir nova senha:");
jPanelTrocarSenha.add(jLabel34);
jLabel34.setBounds(40, 150, 140, 15);

jButton31.setText("Submeter");
jButton31.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton31ActionPerformed(evt);
    }
});

jPanelTrocarSenha.add(jButton31);
jButton31.setBounds(129, 240, 120, 25);

jButton32.setText("Cancelar");
jButton32.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton32ActionPerformed(evt);
    }
});

jPanelTrocarSenha.add(jButton32);

```

```

jButton32.setBounds(297, 240, 130, 25);

jPanelTrocarSenha.add(jPasswordField3);
jPasswordField3.setBounds(130, 110, 170, 23);

jPanelTrocarSenha.add(jPasswordField4);
jPasswordField4.setBounds(150, 150, 150, 23);

getContentPane().add(jPanelTrocarSenha);
jPanelTrocarSenha.setBounds(150, 0, 640, 620);

java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setBounds((screenSize.width-800)/2, (screenSize.height-650)/2, 800,
650);
} //GEN-END: initComponents

private void jButton22ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton22ActionPerformed
    jComboBox5.removeAllItems();
    this.limparInterface();
    jPanelListaDeChamados.setVisible(true);
    ArrayList listaIdChamadosAbertosPendentes = new ArrayList();
    listaIdChamadosAbertosPendentes =
ioesc.getListaIdChamadosAbertosPendentes();
    for (int i=0;i<listaIdChamadosAbertosPendentes.size();i++){
        Integer aux = (Integer)listaIdChamadosAbertosPendentes.get(i);
        jComboBox5.addItem(aux);
    }
} //GEN-LAST:event_jButton22ActionPerformed

private void jButton23ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton23ActionPerformed
    int indice = Integer.parseInt(jLabel61.getText());
    if
((jTextField30.getText().compareTo("")==0) || (jTextArea2.getText().compareTo
(""))==0){
        JOptionPane.showMessageDialog(null,"Preencha 'Categoria' e
'Providencias'");
    }else{
        String cat = jTextField30.getText();
        String prov = jTextArea2.getText();
        boolean abe;
        if (jCheckBox1.isSelected()){abe=true;}else{abe=false;}
        ioesc.trateChamado(indice,cat,prov,abe);
        jPanelChamado.setVisible(false);
    }
} //GEN-LAST:event_jButton23ActionPerformed

private void jButton21ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton21ActionPerformed
    int indice = Integer.parseInt(jLabel61.getText());
    if
((jTextField30.getText().compareTo("")==0) || (jTextArea2.getText().compareTo
(""))==0){
        JOptionPane.showMessageDialog(null,"Preencha 'Categoria' e
'Providencias'.");
    }else{
        String cat = jTextField30.getText();
        String prov = jTextArea2.getText();
        boolean abe = true;

```

```

        boolean necass = true;
        ioesc.abandoneChamado(indice,cat,prov,abe,necass);
        jPanelChamado.setVisible(false);
    }
} //GEN-LAST:event_jButton21ActionPerformed

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton14ActionPerformed
    String p=jTextField4.getText();
    String t=jTextField8.getText();
    String s=jTextField9.getText();
    String f=jTextField10.getText();
    String m=jTextField11.getText();
    String n=jTextField12.getText();
    if ((ioesc.existeDispositivoCadastrado(p)) &&
(jTextField4.getText().compareTo("")!=0)) {
        ioesc.modificarDadosDispositivo(p,t,s,f,m,n);
        jPanelCadastroDispositivo.setVisible(false);
    }else{
        JOptionPane.showMessageDialog(null,"Usuário não existe e não
pode ser alterado.");
    }
    this.limparInterface();
} //GEN-LAST:event_jButton14ActionPerformed

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton11ActionPerformed
    String n = jTextField22.getText();
    String t = jTextField23.getText();
    String e = jTextField24.getText();
    String m = jTextField25.getText();
    String c = jTextField26.getText();
    if (ioesc.modificarAssistencia(n, t, e, m, c)){
        jPanelCadastroAssistencia.setVisible(false);
    }else{
        JOptionPane.showMessageDialog(null,"Nome alterado. Não consta
na lista das assistencias.");
    }
} //GEN-LAST:event_jButton11ActionPerformed

private void jButton52ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton52ActionPerformed
    String n=jTextField14.getText();
    String s=jTextField15.getText();
    String c=jTextField16.getText();
    String set=jTextField17.getText();
    String m=jTextField18.getText();
    String l=jTextField19.getText();
    String senha=jPasswordField5.getText();
    String conf_senh=jPasswordField6.getText();
    int nv;
    if (jRadioButton1.isSelected()){
        nv = 1;
    }else if (jRadioButton2.isSelected()){
        nv = 2;
    }else if (jRadioButton3.isSelected()){
        nv = 3;
    }else{
        nv =1;
    }
    if (senha.compareTo(conf_senh)==0)

```

```

        {
            if ((ioesc.existeUsuarioCadastrado(l)) &&
(1.compareTo("")!=0))
                {
                    ioesc.modificarDadosUsuario(n,s,c,set,m,l,senh,nv);
                    jPanelCadastroUsuario.setVisible(false);
                }else{
                    JOptionPane.showMessageDialog(null,"Usuário não existe e
não pode ser alterado.");
                }
            }else{
                JOptionPane.showMessageDialog(null,"Senhas não conferem");
            }
        }

    }//GEN-LAST:event_jButtonon52ActionPerformed

    private void botaoLogoffActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_botaoLogoffActionPerformed
    this.limparFrame();
    this.limparInterface();
    jPanelIdentificacao.setVisible(true);
    usuarioLogado = "";
} //GEN-LAST:event_botaoLogoffActionPerformed

    private void jButtonon1ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButtonon1ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jButtonon1ActionPerformed

    private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jPasswordField1ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jPasswordField1ActionPerformed

    private void jButton51ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton51ActionPerformed
        int indice = Integer.parseInt(jLabel61.getText());
        if (ioesc.existeTicketCadastrado(indice)){
            ioesc.excluirTicket(indice);
            this.limparTelaVerChamado();
            jPanelChamado.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Ticket não consta no Banco
de Dados.");
        }
    } //GEN-LAST:event_jButton51ActionPerformed

    private void jButton50ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton50ActionPerformed
        jPanelChamado.setVisible(false);
        this.limparTelaVerChamado();
    } //GEN-LAST:event_jButton50ActionPerformed

    private void jButton49ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton49ActionPerformed
        int indice = Integer.parseInt(jLabel61.getText());
        if
((jTextField30.getText().compareTo("")==0) || (jTextField33.getText().compare
To("")==0) || (jTextArea2.getText().compareTo("")==0)) {

```



```

        JOptionPane.showMessageDialog(null,"Preencha 'Categoria, 'Data
de conclusão' e 'Providencias'");
    }else{
        String cat = jTextField30.getText();
        String f_fim = jTextField33.getText();
        String prov = jTextArea2.getText();
        String ass = (String)jComboBox4.getSelectedItem();
        boolean abe = false;
        ioesc.fecheChamado(indice,cat,f_fim,prov,ass,abe);
        jPanelChamado.setVisible(false);
    }
}
} //GEN-LAST:event_jButton49ActionPerformed

private void jButton48ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton48ActionPerformed
    String p_disp = jTextField29.getText();
    if (ioesc.existeDispositivoCadastrado(p_disp)){
        int indice = Integer.parseInt(jLabel61.getText());
        String tit = jTextField27.getText();
        String desc = jTextArea3.getText();
        String cat = jTextField30.getText();
        String urg = jTextField31.getText();
        String d_ini = jTextField32.getText();
        String d_fim = jTextField33.getText();
        String l_prop = jTextField34.getText();
        String l_resp = jTextField35.getText();
        boolean abe;
        if (jCheckBox1.isSelected()){abe = true;}else{abe = false;}
        String ass = ((String)jComboBox4.getSelectedItem());
        String prov = jTextArea2.getText();

        ioesc.alteraChamado(indice,tit,desc,p_disp,cat,urg,d_ini,d_fim,l_prop,l_res
p,abe,ass,prov);
        jPanelChamado.setVisible(false);
    }else{
        JOptionPane.showMessageDialog(null,"Infelizmente não existe
este dispositivo cadastrado.");
    }
} //GEN-LAST:event_jButton48ActionPerformed

private void jButton44ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton44ActionPerformed
    jPanelListaDasAssistencias.setVisible(false);
    jPanelCadastroAssistencia.setVisible(true);
    this.limparTelaCadastroAssistencia();

    String nomeass = (String) jComboBox2.getSelectedItem();

    ArrayList aAssistencia = new ArrayList();

    aAssistencia = ioesc.getInformacoesAssistencia(nomeass);

    jTextField22.setText(nomeass);
    jTextField23.setText((String)aAssistencia.get(1));
    jTextField24.setText((String)aAssistencia.get(2));
    jTextField25.setText((String)aAssistencia.get(3));
    jTextField26.setText((String)aAssistencia.get(4));
} //GEN-LAST:event_jButton44ActionPerformed

private void jButton45ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton45ActionPerformed

```

```

        jPanelListaDasAssistencias.setVisible(false);
    }//GEN-LAST:event_jButton45ActionPerformed

    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton13ActionPerformed
        jComboBox2.removeAllItems();
        this.limparInterface();
        jPanelListaDasAssistencias.setVisible(true);
        ArrayList assistenciasCadastradas = new ArrayList();
        assistenciasCadastradas = ioesc.getListaAssistenciasCadastradas();
        for (int i=0;i<assistenciasCadastradas.size();i++){
            String aux = (String)assistenciasCadastradas.get(i);
            jComboBox2.addItem(aux);
        }
    }//GEN-LAST:event_jButton13ActionPerformed

    private void jButton43ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton43ActionPerformed
        jPanelCadastroAssistencia.setVisible(false);
        jPanelListaDasAssistencias.setVisible(false);
    }//GEN-LAST:event_jButton43ActionPerformed

    private void jButton42ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton42ActionPerformed
        if (ioesc.excluirAssistencia(jTextField22.getText())){
            jPanelCadastroAssistencia.setVisible(false);
        }else
        {
            JOptionPane.showMessageDialog(null,"Nome alterado. Não consta
na lista das assistencias.");
        }
    }

    }//GEN-LAST:event_jButton42ActionPerformed

    private void jButton41ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton41ActionPerformed
        String n = jTextField22.getText();
        String t = jTextField23.getText();
        String e = jTextField24.getText();
        String m = jTextField25.getText();
        String c = jTextField26.getText();
        if (ioesc.existeAssistenciaCadastrada(n)){
            JOptionPane.showMessageDialog(null,"Assistencia não pode ser
cadastrada.");
        }else{
            ioesc.criarAssistencia(n, t, e, m, c);
            jPanelCadastroAssistencia.setVisible(false);
        }
    }
    }//GEN-LAST:event_jButton41ActionPerformed

    private void jButton12ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton12ActionPerformed
        this.limparInterface();
        this.limparTelaCadastroAssistencia();
        jPanelCadastroAssistencia.setVisible(true);
    }//GEN-LAST:event_jButton12ActionPerformed

    private void jButton37ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton37ActionPerformed
        jPanelListaDeUsuarios.setVisible(false);
    }//GEN-LAST:event_jButton37ActionPerformed

```

```

private void jButton36ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton36ActionPerformed
    this.limparTelaCadastroUsuario();
    String login_usuario = (String) jComboBox1.getSelectedItemAt();
    jPanelListaDeUsuarios.setVisible(false);
    jPanelCadastroUsuario.setVisible(true);
    ArrayList oUsuario = new ArrayList();
    oUsuario = ioesc.getInformacoesUsuario(login_usuario);
    jTextField14.setText((String)oUsuario.get(0));
    jTextField15.setText((String)oUsuario.get(1));
    jTextField16.setText((String)oUsuario.get(2));
    jTextField17.setText((String)oUsuario.get(3));
    jTextField18.setText((String)oUsuario.get(4));
    jTextField19.setText((String)oUsuario.get(5));
    jPasswordField5.setText((String)oUsuario.get(6));
    jPasswordField6.setText((String)oUsuario.get(6));
    int nn = (Integer) oUsuario.get(7).intValue();
    if (nn==1){
        jButton1.setSelected(true);
    }else{
        if (nn==2){
            jButton2.setSelected(true);
        }else{
            if (nn==3){
                jButton3.setSelected(true);
            }
        }
    }
} //GEN-LAST:event_jButton36ActionPerformed

private void jButton35ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton35ActionPerformed
    jPanelCadastroUsuario.setVisible(false);
} //GEN-LAST:event_jButton35ActionPerformed

private void jButton34ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton34ActionPerformed
    String l=jTextField19.getText();
    if ((l.compareTo("")!=0)&&(ioesc.existeUsuarioCadastrado(l))){
        jPanelCadastroUsuario.setVisible(false);
        ioesc.excluirUsuario(l);
        this.limparTelaCadastroUsuario();
    }else{
        JOptionPane.showMessageDialog(null,"Não foi possível
excluir.");
    }
} //GEN-LAST:event_jButton34ActionPerformed

private void jButton33ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton33ActionPerformed
    String n=jTextField14.getText();
    String s=jTextField15.getText();
    String c=jTextField16.getText();
    String set=jTextField17.getText();
    String m=jTextField18.getText();
    String l=jTextField19.getText();
    String senh=jPasswordField5.getText();
    String conf_senh=jPasswordField6.getText();
    int nv;
    if (jRadioButton1.isSelected()){

```

```

        nv = 1;
    }else if (jRadioButton2.isSelected()){
        nv = 2;
    }else if (jRadioButton3.isSelected()){
        nv = 3;
    }else{
        nv =1;
    }

    if (senh.compareTo(conf_senh)==0){
        if ((ioesc.existeUsuarioCadastrado(l)==false) &&
(l.compareTo("")!=0)){
            ioesc.criarUsuario(n,s,c,set,m,l,senh,nv);
            jPanelCadastroUsuario.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Tente novamente.
Usuário cadastrado ou login vazio.");
        }
    }else{
        JOptionPane.showMessageDialog(null,"Senhas não conferem.");
    }
} //GEN-LAST:event_jButton33ActionPerformed

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton9ActionPerformed
    this.limparInterface();
    jPanelCadastroUsuario.setVisible(true);
    this.limparTelaCadastroUsuario();
} //GEN-LAST:event_jButton9ActionPerformed

private void jButton32ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton32ActionPerformed
    jPanelTrocarSenha.setVisible(false);
} //GEN-LAST:event_jButton32ActionPerformed

private void jButton31ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton31ActionPerformed
    jPanelTrocarSenha.setVisible(false);
    if
(jPasswordField3.getText().compareTo(jPasswordField4.getText())==0){
        ioesc.troqueSenha(usuarioLogado,jPasswordField3.getText());
    }
} //GEN-LAST:event_jButton31ActionPerformed

private void jButton15ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton15ActionPerformed
    this.limparInterface();
    jPanelTrocarSenha.setVisible(true);
} //GEN-LAST:event_jButton15ActionPerformed

private void jButton29ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton29ActionPerformed

    jPanelListaDosDispositivos.setVisible(false);
    jPanelCadastroDispositivo.setVisible(true);
    this.limparTelaCadastroEquipamento();

    String nro_patrr = (String) jComboBox3.getSelectedItemAt();

    ArrayList oDispositivo = new ArrayList();
    oDispositivo = ioesc.getInformacoesDispositivo(nro_patrr);

```

```

        jTextField4.setText(nro_patrr);
        jTextField8.setText((String)oDispositivo.get(1));
        jTextField9.setText((String)oDispositivo.get(2));
        jTextField10.setText((String)oDispositivo.get(3));
        jTextField11.setText((String)oDispositivo.get(4));
        jTextField12.setText((String)oDispositivo.get(5));
    }//GEN-LAST:event_jButton29ActionPerformed

    private void jButton30ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton30ActionPerformed
        jPanelListaDosDispositivos.setVisible(false);
        this.limparInterface();
    }//GEN-LAST:event_jButton30ActionPerformed

    private void jButton27ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton27ActionPerformed
        String nrodisp = jTextField13.getText();
        JComboBox3.removeAllItems();
        this.limparInterface();

        if (ioesc.existeDispositivoCadastrado(nrodisp)){
            JComboBox3.addItem(nrodisp);
            jPanelPesquisarDispositivo.setVisible(false);
            jPanelListaDosDispositivos.setVisible(true);
        }else{
            this.limparInterface();
            JOptionPane.showMessageDialog(null,"Dispositivo não
cadastrado.");
        }
    }//GEN-LAST:event_jButton27ActionPerformed

    private void jButton28ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton28ActionPerformed
        jPanelPesquisarDispositivo.setVisible(false);
        this.limparInterface();
    }//GEN-LAST:event_jButton28ActionPerformed

    private void jButton26ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton26ActionPerformed
        jPanelCadastroDispositivo.setVisible(false);
        this.limparInterface();
    }//GEN-LAST:event_jButton26ActionPerformed

    private void jButton25ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton25ActionPerformed
        String patr=jTextField4.getText();
        if
((patr.compareTo("")!=0)&&(ioesc.existeDispositivoCadastrado(patr))){
            jPanelCadastroDispositivo.setVisible(false);
            ioesc.excluirDispositivo(patr);
            this.limparTelaCadastroEquipamento();
        }else{
            JOptionPane.showMessageDialog(null,"Não foi possível
excluir.");
        }
    }//GEN-LAST:event_jButton25ActionPerformed

    private void jButton24ActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButton24ActionPerformed

```

```

String p = jTextField4.getText();
String t = jTextField8.getText();
String s = jTextField9.getText();
String f = jTextField10.getText();
String m = jTextField11.getText();
String ns = jTextField12.getText();
if ((ioesc.existeDispositivoCadastrado(p)) || (p.compareTo("")==0)){
    this.limparInterface();
    JOptionPane.showMessageDialog(null,"Dispositivo NÃO pode ser
cadastrado.");
}
else{
    ioesc.criarDispositivo(p,t,s,f,m,ns);
}
jPanelCadastroDispositivo.setVisible(false);
this.limparInterface();
} //GEN-LAST:event_jButton24ActionPerformed

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton7ActionPerformed
    this.limparInterface();
    this.limparTelaCadastroEquipamento();
    jPanelCadastroDispositivo.setVisible(true);
} //GEN-LAST:event_jButton7ActionPerformed

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton6ActionPerformed
    this.limparInterface();
    jPanelPesquisarChamado.setVisible(true);
    jTextField3.setText("");
    jComboBox6.removeAllItems();
    jComboBox6.addItem("");
    jComboBox6.addItem("Patrimonio");
    jComboBox6.addItem("Nro Ticket");
} //GEN-LAST:event_jButton6ActionPerformed

private void jButton20ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton20ActionPerformed
    jPanelPesquisarChamado.setVisible(false);
} //GEN-LAST:event_jButton20ActionPerformed

private void jButton19ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton19ActionPerformed
    jComboBox5.removeAllItems();
    this.limparInterface();
    jPanelPesquisarChamado.setVisible(true);
    String valorPesquisa = jTextField3.getText();

    if
(((String)jComboBox6.getSelectedItem()).compareTo("Patrimonio")==0){
        jPanelListaDeChamados.setVisible(true);
        ArrayList listaIdChamados = new ArrayList();
        listaIdChamados = ioesc.getListaIdChamados(true,valorPesquisa);
        for (int i=0;i<listaIdChamados.size();i++){
            Integer aux = (Integer)listaIdChamados.get(i);
            jComboBox5.addItem(aux);
        }
        jPanelPesquisarChamado.setVisible(false);
    }
    else{
        if (((String)jComboBox6.getSelectedItem()).compareTo("Nro
Ticket")==0){
            jPanelListaDeChamados.setVisible(true);

```

```

        ArrayList listaIdChamados = new ArrayList();
        listaIdChamados =
ioesc.getListaIdChamados(false, valorPesquisa);
        for (int i=0;i<listaIdChamados.size();i++){
            Integer aux = (Integer)listaIdChamados.get(i);
            jComboBox5.addItem(aux);
        }
        jPanelPesquisarChamado.setVisible(false);
    }else{
        JOptionPane.showMessageDialog(null, "Selecione um tipo de
pesquisa.");
    }
}
} //GEN-LAST:event_jButton19ActionPerformed

private void jButton18ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton18ActionPerformed
    jPanelListaDeChamados.setVisible(false);
} //GEN-LAST:event_jButton18ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton3ActionPerformed
    jComboBox4.removeAllItems();
    this.limparInterface();
    this.limparTelaVerChamado();
    jPanelChamado.setVisible(true);

    ArrayList oChamado = new ArrayList();

    if (((String)jComboBox5.getSelectedItem()).compareTo("")==0){
        JOptionPane.showMessageDialog(null, "Nenhum chamado
selecionado.");
    }else{
        Integer ID_inteiro = (Integer) jComboBox5.getSelectedItem();
        int ID = (ID_inteiro.intValue());

        oChamado = ioesc.getInformacoesChamado(ID);

        jLabel20.setText((String)oChamado.get(0));
        jLabel32.setText((String)oChamado.get(1));
        jLabel46.setText((String)oChamado.get(2));
        jLabel47.setText((String)oChamado.get(3));
        jLabel49.setText((String)oChamado.get(4));

        jLabel61.setText((String)oChamado.get(5));
        jTextField27.setText((String)oChamado.get(6));
        jTextArea3.setText((String)oChamado.get(7));
        jTextField29.setText((String)oChamado.get(8));
        jTextField30.setText((String)oChamado.get(9));
        jTextField31.setText((String)oChamado.get(10));
        jTextField32.setText((String)oChamado.get(11));
        jTextField33.setText((String)oChamado.get(12));
        jTextField34.setText((String)oChamado.get(13));
        jTextField35.setText((String)oChamado.get(14));
        if (((String)oChamado.get(15)).compareTo("true")==0)
{jCheckBox1.setSelected(true);}else{jCheckBox1.setSelected(false);}
        jComboBox4.addItem((String)oChamado.get(16));
        jTextArea2.setText((String)oChamado.get(17));

        ArrayList assistenciasCadastradas = new ArrayList();

```

```

        assistenciasCadastradas =
ioesc.getListaAssistenciasCadastradas();
        for (int i=0;i<assistenciasCadastradas.size();i++){
            String ass = (String)assistenciasCadastradas.get(i);
            jComboBox4.addItem(ass);
        }
    }
} //GEN-LAST:event_jButton3ActionPerformed

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton5ActionPerformed
    jComboBox5.removeAllItems();
    this.limparInterface();
    jPanelListaDeChamados.setVisible(true);
    ArrayList listaIdChamadosAbertos = new ArrayList();
    listaIdChamadosAbertos = ioesc.getListaIdChamadosAbertos();
    for (int i=0;i<listaIdChamadosAbertos.size();i++){
        Integer aux = (Integer)listaIdChamadosAbertos.get(i);
        jComboBox5.addItem(aux);
    }
} //GEN-LAST:event_jButton5ActionPerformed

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton17ActionPerformed
    jPanelAbrirChamado.setVisible(false);
} //GEN-LAST:event_jButton17ActionPerformed

private void jButton16ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton16ActionPerformed
    String p = jTextField6.getText();
    String t = jTextField2.getText();
    String d = jTextArea1.getText();
    String u = jTextField7.getText();
    String dt = jTextField5.getText();
    if (ioesc.existeDispositivoCadastrado(p)){
        ioesc.criarTicket(t,d,u,dt,p,usuarioLogado);
        jPanelAbrirChamado.setVisible(false);
        this.limparInterface();
    }else{
        JOptionPane.showMessageDialog(null,"Dispositivo não está
cadastrado. Entre em contato com a SUATEC.");
    }
} //GEN-LAST:event_jButton16ActionPerformed

private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jTextField6ActionPerformed

} //GEN-LAST:event_jTextField6ActionPerformed

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jTextField2ActionPerformed

} //GEN-LAST:event_jTextField2ActionPerformed

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton10ActionPerformed
    jComboBox1.removeAllItems();
    this.limparInterface();
    jPanelListaDeUsuarios.setVisible(true);
    ArrayList usuariosCadastrados = new ArrayList();
    usuariosCadastrados = ioesc.getListaLoginUsuariosCadastrados();
}

```



```

        for (int i=0;i<usuariosCadastrados.size();i++){
            String aux = (String)usuariosCadastrados.get(i);
            jComboBox1.addItem(aux);
        }
    }//GEN-LAST:event_jButton10ActionPerformed

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton8ActionPerformed
        this.limparInterface();
        jPanelPesquisarDispositivo.setVisible(true);
        jTextField13.setText("");
    } //GEN-LAST:event_jButton8ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton4ActionPerformed
        this.limparInterface();
        this.limparTelaAberturaChamado();
        jPanelAbrirChamado.setVisible(true);
    } //GEN-LAST:event_jButton4ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton1ActionPerformed
        String l=jTextField1.getText();
        String s=jPasswordField1.getText();
        if (ioesc.acessarSistema(l,s)){
            jPanelIdentificacao.setVisible(false);
            this.disponibilizarFrame();
            usuarioLogado = l;
        }else{
            JOptionPane.showMessageDialog(null,"Login e Senha não
            conferem!");
        }
        jTextField1.setText("");
        jPasswordField1.setText("");
    } //GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton2ActionPerformed
        System.exit(0);
    } //GEN-LAST:event_jButton2ActionPerformed

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jTextField1ActionPerformed

    } //GEN-LAST:event_jTextField1ActionPerformed

    private void exitForm(java.awt.event.WindowEvent evt) { //GEN-
    FIRST:event_exitForm
        System.exit(0);
    } //GEN-LAST:event_exitForm

    public static void main(String args[]) {
        new Pas().show();
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton botaoLogoff;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton10;

```

```
private javax.swing.JButton jButton11;
private javax.swing.JButton jButton12;
private javax.swing.JButton jButton13;
private javax.swing.JButton jButton14;
private javax.swing.JButton jButton15;
private javax.swing.JButton jButton16;
private javax.swing.JButton jButton17;
private javax.swing.JButton jButton18;
private javax.swing.JButton jButton19;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton20;
private javax.swing.JButton jButton21;
private javax.swing.JButton jButton22;
private javax.swing.JButton jButton23;
private javax.swing.JButton jButton24;
private javax.swing.JButton jButton25;
private javax.swing.JButton jButton26;
private javax.swing.JButton jButton27;
private javax.swing.JButton jButton28;
private javax.swing.JButton jButton29;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton30;
private javax.swing.JButton jButton31;
private javax.swing.JButton jButton32;
private javax.swing.JButton jButton33;
private javax.swing.JButton jButton34;
private javax.swing.JButton jButton35;
private javax.swing.JButton jButton36;
private javax.swing.JButton jButton37;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton41;
private javax.swing.JButton jButton42;
private javax.swing.JButton jButton43;
private javax.swing.JButton jButton44;
private javax.swing.JButton jButton45;
private javax.swing.JButton jButton48;
private javax.swing.JButton jButton49;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton50;
private javax.swing.JButton jButton51;
private javax.swing.JButton jButton52;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JButton jButton9;
private javax.swing.JCheckBox jCheckBox1;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JComboBox jComboBox3;
private javax.swing.JComboBox jComboBox4;
private javax.swing.JComboBox jComboBox5;
private javax.swing.JComboBox jComboBox6;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
```



```
private javax.swing.JLabel jLabel76;
private javax.swing.JLabel jLabel77;
private javax.swing.JLabel jLabel78;
private javax.swing.JLabel jLabel79;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanelAbrirChamado;
private javax.swing.JPanel jPanelCadastroAssistencia;
private javax.swing.JPanel jPanelCadastroDispositivo;
private javax.swing.JPanel jPanelCadastroUsuario;
private javax.swing.JPanel jPanelChamado;
private javax.swing.JPanel jPanelIdentificacao;
private javax.swing.JPanel jPanelListaDasAssistencias;
private javax.swing.JPanel jPanelListaDeChamados;
private javax.swing.JPanel jPanelListaDeUsuarios;
private javax.swing.JPanel jPanelListaDosDispositivos;
private javax.swing.JPanel jPanelPesquisarChamado;
private javax.swing.JPanel jPanelPesquisarDispositivo;
private javax.swing.JPanel jPanelTrocarSenha;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JPasswordField jPasswordField3;
private javax.swing.JPasswordField jPasswordField4;
private javax.swing.JPasswordField jPasswordField5;
private javax.swing.JPasswordField jPasswordField6;
private javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton2;
private javax.swing.JRadioButton jRadioButton3;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
private javax.swing.JTextArea jTextArea3;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField22;
private javax.swing.JTextField jTextField23;
private javax.swing.JTextField jTextField24;
private javax.swing.JTextField jTextField25;
private javax.swing.JTextField jTextField26;
private javax.swing.JTextField jTextField27;
private javax.swing.JTextField jTextField29;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField30;
private javax.swing.JTextField jTextField31;
private javax.swing.JTextField jTextField32;
private javax.swing.JTextField jTextField33;
private javax.swing.JTextField jTextField34;
private javax.swing.JTextField jTextField35;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
```

```

private javax.swing.JTextField jTextField9;
// End of variables declaration//GEN-END:variables

}

/*****PasClient.java*****/

package suatec;

import javax.swing.*;
import java.util.*;
import javax.xml.rpc.Stub;
import java.lang.String;

public class PasClient extends javax.swing.JFrame {
    Empresa ioesc;
    String usuarioLogado;

    public PasClient() {
        initComponents();
        this.limparInterface();
        this.limparFrame();
        jPanelIdentificacao.setVisible(true);
        Stub stub = createProxy();
        EmpresaInterface ioesc = (EmpresaInterface) stub;
    }

    private void limparInterface(){
        jPanelAbrirChamado.setVisible(false);
        jPanelListaDeChamados.setVisible(false);
        jPanelPesquisarChamado.setVisible(false);
        jPanelCadastroDispositivo.setVisible(false);
        jPanelPesquisarDispositivo.setVisible(false);
        jPanelListaDosDispositivos.setVisible(false);
        jPanelTrocarSenha.setVisible(false);
        jPanelChamado.setVisible(false);
        jPanelIdentificacao.setVisible(false);
    }

    private void limparFrame(){
        labelTecnico.setVisible(false);
        labelUtente.setVisible(false);
        jLabel5.setVisible(false);
        botaoAbrir.setVisible(false);
        botaoListarAbertos.setVisible(false);
        botaoPesquisar.setVisible(false);
        jLabel6.setVisible(false);
        botaoCadastrar.setVisible(false);
        botaoPesquisarDispositivo.setVisible(false);
        botaoTrocarSenha.setVisible(false);
        botaoLogoff.setVisible(false);
        jLabel9.setVisible(false);
        jToggleButton1.setVisible(false);
    }

    private void disponibilizarFrameUtente(){
        labelTecnico.setVisible(false);
        labelUtente.setVisible(true);
        jLabel5.setVisible(true);
        botaoAbrir.setVisible(true);
        botaoListarAbertos.setVisible(false);
    }

```

```

        botaoPesquisar.setVisible(true);
        jLabel6.setVisible(false);
        botaoCadastrar.setVisible(false);
        botaoPesquisarDispositivo.setVisible(false);
        botaoTrocarSenha.setVisible(true);
        botaoLogoff.setVisible(true);
        jLabel9.setVisible(true);
        jButton1.setVisible(false);
    }

    private void disponibilizarFrameTecnico(){
        labelTecnico.setVisible(true);
        labelUtente.setVisible(false);
        jLabel5.setVisible(true);
        botaoAbrir.setVisible(false);
        botaoListarAbertos.setVisible(true);
        botaoPesquisar.setVisible(true);
        jLabel6.setVisible(true);
        botaoCadastrar.setVisible(true);
        botaoPesquisarDispositivo.setVisible(true);
        botaoTrocarSenha.setVisible(true);
        botaoLogoff.setVisible(true);
        jLabel9.setVisible(true);
        jButton1.setVisible(true);
    }

    private void disponibilizarFrameTecnico_Utente(){
        labelTecnico.setVisible(false);
        labelUtente.setVisible(true);
        jLabel5.setVisible(true);
        botaoAbrir.setVisible(true);
        botaoListarAbertos.setVisible(false);
        botaoPesquisar.setVisible(true);
        jLabel6.setVisible(false);
        botaoCadastrar.setVisible(false);
        botaoPesquisarDispositivo.setVisible(false);
        botaoTrocarSenha.setVisible(true);
        botaoLogoff.setVisible(true);
        jLabel9.setVisible(true);
        jButton1.setVisible(true);
    }

    private void limparTelaCadastroEquipamento(){
        jTextField4.setText("");
        jTextField8.setText("");
        jTextField9.setText("");
        jTextField10.setText("");
        jTextField11.setText("");
        jTextField12.setText("");
    }

    private void limparTelaAberturaChamado(){
        jTextField2.setText("");
        jTextArea1.setText("");
        jTextField7.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
    }

    private void limparTelaVerChamado(){
        jLabel61.setText("");
    }

```

```

        jTextField27.setText("");
        jTextArea3.setText("");
        jTextField29.setText("");
        jTextField30.setText("");
        jTextField31.setText("");
        jTextField32.setText("");
        jTextField33.setText("");
        jTextField34.setText("");
        jTextField35.setText("");
        jCheckBox1.setContentAreaFilled(false);
        jTextArea2.setText("");
        jComboBox4.removeAllItems();
        jLabel20.setText("");
        jLabel32.setText("");
        jLabel46.setText("");
        jLabel47.setText("");
        jLabel49.setText("");
    }

private void initComponents() { //GEN-BEGIN: initComponents
    jPanelIdentificacao = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jPasswordField1 = new javax.swing.JPasswordField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    labelTecnico = new javax.swing.JLabel();
    labelUtente = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    botaoAbrir = new javax.swing.JButton();
    botaoListarAbertos = new javax.swing.JButton();
    botaoPesquisar = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    botaoCadastrar = new javax.swing.JButton();
    botaoPesquisarDispositivo = new javax.swing.JButton();
    botaoTrocarSenha = new javax.swing.JButton();
    botaoLogoff = new javax.swing.JButton();
    jLabel9 = new javax.swing.JLabel();
    jToggleButton1 = new javax.swing.JToggleButton();
    jPanelAbrirChamado = new javax.swing.JPanel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    jLabel18 = new javax.swing.JLabel();
    jLabel19 = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jTextArea1 = new javax.swing.JTextArea();
    jTextField5 = new javax.swing.JTextField();
    jTextField6 = new javax.swing.JTextField();
    jButton16 = new javax.swing.JButton();
    jButton17 = new javax.swing.JButton();
    jTextField7 = new javax.swing.JTextField();
    jPanelPesquisarChamado = new javax.swing.JPanel();
    jButton19 = new javax.swing.JButton();
    jButton20 = new javax.swing.JButton();
    jLabel16 = new javax.swing.JLabel();
    jLabel17 = new javax.swing.JLabel();
    jComboBox6 = new javax.swing.JComboBox();

```

```
jTextField3 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jPanelListaDeChamados = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jButton18 = new javax.swing.JButton();
jLabel115 = new javax.swing.JLabel();
jComboBox5 = new javax.swing.JComboBox();
jPanelChamado = new javax.swing.JPanel();
jButton48 = new javax.swing.JButton();
jButton49 = new javax.swing.JButton();
jButton50 = new javax.swing.JButton();
jButton51 = new javax.swing.JButton();
jLabel59 = new javax.swing.JLabel();
jLabel60 = new javax.swing.JLabel();
jLabel61 = new javax.swing.JLabel();
jLabel62 = new javax.swing.JLabel();
jLabel63 = new javax.swing.JLabel();
jLabel64 = new javax.swing.JLabel();
jLabel65 = new javax.swing.JLabel();
jLabel66 = new javax.swing.JLabel();
jLabel67 = new javax.swing.JLabel();
jLabel68 = new javax.swing.JLabel();
jLabel69 = new javax.swing.JLabel();
jLabel70 = new javax.swing.JLabel();
jLabel71 = new javax.swing.JLabel();
jLabel72 = new javax.swing.JLabel();
jLabel73 = new javax.swing.JLabel();
jLabel74 = new javax.swing.JLabel();
jLabel75 = new javax.swing.JLabel();
jLabel76 = new javax.swing.JLabel();
jLabel77 = new javax.swing.JLabel();
jLabel78 = new javax.swing.JLabel();
jLabel79 = new javax.swing.JLabel();
jTextField27 = new javax.swing.JTextField();
jTextField29 = new javax.swing.JTextField();
jTextField30 = new javax.swing.JTextField();
jTextField31 = new javax.swing.JTextField();
jTextField32 = new javax.swing.JTextField();
jTextField33 = new javax.swing.JTextField();
jTextField34 = new javax.swing.JTextField();
jTextField35 = new javax.swing.JTextField();
jTextArea2 = new javax.swing.JTextArea();
jTextArea3 = new javax.swing.JTextArea();
jCheckBox1 = new javax.swing.JCheckBox();
jComboBox4 = new javax.swing.JComboBox();
jLabel20 = new javax.swing.JLabel();
jLabel32 = new javax.swing.JLabel();
jLabel46 = new javax.swing.JLabel();
jLabel47 = new javax.swing.JLabel();
jLabel49 = new javax.swing.JLabel();
jButton21 = new javax.swing.JButton();
jButton23 = new javax.swing.JButton();
jPanelCadastroDispositivo = new javax.swing.JPanel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
```



```

jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jButton24 = new javax.swing.JButton();
jButton25 = new javax.swing.JButton();
jButton26 = new javax.swing.JButton();
jButton14 = new javax.swing.JButton();
jPanelPesquisarDispositivo = new javax.swing.JPanel();
jButton27 = new javax.swing.JButton();
jButton28 = new javax.swing.JButton();
jLabel28 = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
jTextField13 = new javax.swing.JTextField();
jPanelListaDosDispositivos = new javax.swing.JPanel();
jButton29 = new javax.swing.JButton();
jButton30 = new javax.swing.JButton();
jLabel30 = new javax.swing.JLabel();
jComboBox3 = new javax.swing.JComboBox();
jPanelTrocarSenha = new javax.swing.JPanel();
jLabel31 = new javax.swing.JLabel();
jLabel33 = new javax.swing.JLabel();
jLabel34 = new javax.swing.JLabel();
jButton31 = new javax.swing.JButton();
jButton32 = new javax.swing.JButton();
jPasswordField3 = new javax.swing.JPasswordField();
jPasswordField4 = new javax.swing.JPasswordField();

getContentPane().setLayout(null);

setTitle("Cadastra");
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
        exitForm(evt);
    }
});

jPanelIdentificacao.setLayout(null);

jPanelIdentificacao.setPreferredSize(new java.awt.Dimension(600,
600));

jLabel1.setText("Login:");
jPanelIdentificacao.add(jLabel1);
jLabel1.setBounds(40, 50, 40, 15);

jLabel2.setText("Senha:");
jLabel2.setRequestFocusEnabled(false);
jPanelIdentificacao.add(jLabel2);
jLabel2.setBounds(40, 80, 40, 15);

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jTextField1);
jTextField1.setBounds(90, 50, 80, 21);

```

```

        jPasswordField1.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jPasswordField1ActionPerformed(evt);
            }
        });

jPanelIdentificacao.add(jPasswordField1);
jPasswordField1.setBounds(90, 80, 80, 20);

jButton1.setLabel("OK");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jButton1);
jButton1.setBounds(40, 130, 80, 25);

jButton2.setLabel("Cancelar");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jPanelIdentificacao.add(jButton2);
jButton2.setBounds(140, 130, 90, 25);

getContentPane().add(jPanelIdentificacao);
jPanelIdentificacao.setBounds(220, 40, 380, 320);

jPanelIdentificacao.getAccessibleContext().setAccessibleParent(jPanelIdenti
ficacao);

labelTecnico.setText("T\u00e9cnico");
getContentPane().add(labelTecnico);
labelTecnico.setBounds(0, 0, 110, 15);

labelUtente.setText("Usu\u00e1rio");
getContentPane().add(labelUtente);
labelUtente.setBounds(0, 0, 80, 15);

jLabel5.setText("Chamado");
getContentPane().add(jLabel5);
jLabel5.setBounds(0, 70, 70, 15);

botaoAbrir.setLabel("Abrir");
botaoAbrir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botaoAbrirActionPerformed(evt);
    }
});

getContentPane().add(botaoAbrir);
botaoAbrir.setBounds(10, 90, 130, 20);

botaoListarAbertos.setLabel("Listar Abertos");
botaoListarAbertos.addActionListener(new
java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoListarAbertosActionPerformed(evt);
        }
    });

    getContentPane().add(botaoListarAbertos);
    botaoListarAbertos.setBounds(10, 90, 130, 20);

    botaoPesquisar.setLabel("Pesquisar");
    botaoPesquisar.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoPesquisarActionPerformed(evt);
        }
    });

    getContentPane().add(botaoPesquisar);
    botaoPesquisar.setBounds(10, 110, 130, 20);

    jLabel6.setText("Dispositivo");
    getContentPane().add(jLabel6);
    jLabel6.setBounds(0, 160, 80, 15);

    botaoCadastrar.setLabel("Cadastrar");
    botaoCadastrar.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoCadastrarActionPerformed(evt);
        }
    });

    getContentPane().add(botaoCadastrar);
    botaoCadastrar.setBounds(10, 180, 130, 20);

    botaoPesquisarDispositivo.setLabel("Pesquisar");
    botaoPesquisarDispositivo.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoPesquisarDispositivoActionPerformed(evt);
        }
    });

    getContentPane().add(botaoPesquisarDispositivo);
    botaoPesquisarDispositivo.setBounds(10, 200, 130, 20);

    botaoTrocarSenha.setLabel("Trocar");
    botaoTrocarSenha.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoTrocarSenhaActionPerformed(evt);
        }
    });

    getContentPane().add(botaoTrocarSenha);
    botaoTrocarSenha.setBounds(10, 390, 130, 20);

    botaoLogoff.setText("Logoff");
    botaoLogoff.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botaoLogoffActionPerformed(evt);
        }
    });

```

```

    });

    getContentPane().add(botaoLogoff);
    botaoLogoff.setBounds(10, 435, 130, 20);

    jLabel9.setText("Senha");
    getContentPane().add(jLabel9);
    jLabel9.setBounds(0, 370, 80, 15);

    jButton1.setText("Usu\u00e9rio");
    jButton1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

    getContentPane().add(jButton1);
    jButton1.setBounds(10, 540, 130, 20);

    jPanelAbrirChamado.setLayout(null);

    jLabel10.setText("Abrir Chamado");
    jPanelAbrirChamado.add(jLabel10);
    jLabel10.setBounds(220, 10, 100, 15);

    jLabel11.setText("T\u00edtulo:");
    jPanelAbrirChamado.add(jLabel11);
    jLabel11.setBounds(50, 50, 50, 15);

    jLabel12.setText("Descri\u00e7\u00e3o:");
    jPanelAbrirChamado.add(jLabel12);
    jLabel12.setBounds(50, 80, 70, 15);

    jLabel13.setText("Urg\u00eancia:");
    jPanelAbrirChamado.add(jLabel13);
    jLabel13.setBounds(50, 180, 70, 15);

    jLabel14.setText("Data:");
    jPanelAbrirChamado.add(jLabel14);
    jLabel14.setBounds(50, 220, 40, 15);

    jLabel18.setText("Dispositivo");
    jPanelAbrirChamado.add(jLabel18);
    jLabel18.setBounds(50, 250, 90, 15);

    jLabel19.setText("Patrim\u00f3nio:");
    jPanelAbrirChamado.add(jLabel19);
    jLabel19.setBounds(80, 270, 80, 15);

    jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});

    jPanelAbrirChamado.add(jTextField2);
    jTextField2.setBounds(130, 50, 200, 20);

    jPanelAbrirChamado.add(jTextArea1);
    jTextArea1.setBounds(130, 80, 200, 70);

```

```

jPanelAbrirChamado.add(jTextField5);
jTextField5.setBounds(130, 220, 200, 21);

jTextField6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField6ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jTextField6);
jTextField6.setBounds(150, 270, 110, 21);

jButton16.setText("Criar");
jButton16.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton16ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jButton16);
jButton16.setBounds(100, 330, 110, 25);

jButton17.setText("Sair");
jButton17.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton17ActionPerformed(evt);
    }
});

jPanelAbrirChamado.add(jButton17);
jButton17.setBounds(250, 330, 100, 25);

jPanelAbrirChamado.add(jTextField7);
jTextField7.setBounds(130, 170, 200, 21);

getContentPane().add(jPanelAbrirChamado);
jPanelAbrirChamado.setBounds(150, -10, 620, 550);

jPanelPesquisarChamado.setLayout(null);

jButton19.setText("Procurar");
jButton19.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton19ActionPerformed(evt);
    }
});

jPanelPesquisarChamado.add(jButton19);
jButton19.setBounds(120, 340, 120, 25);

jButton20.setText("Sair");
jButton20.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton20ActionPerformed(evt);
    }
});

jPanelPesquisarChamado.add(jButton20);
jButton20.setBounds(270, 340, 110, 25);

```

```

jLabel16.setText("Pesquisar Chamado");
jPanelPesquisarChamado.add(jLabel16);
jLabel16.setBounds(300, 20, 140, 15);

jLabel17.setText("Tipo:");
jPanelPesquisarChamado.add(jLabel17);
jLabel17.setBounds(100, 80, 70, 15);

jPanelPesquisarChamado.add(jComboBox6);
jComboBox6.setBounds(150, 80, 230, 21);

jPanelPesquisarChamado.add(jTextField3);
jTextField3.setBounds(150, 140, 230, 21);

jLabel14.setText("Patrimonio / N\u00b0 Ticket");
jPanelPesquisarChamado.add(jLabel14);
jLabel14.setBounds(10, 140, 140, 15);

getContentPane().add(jPanelPesquisarChamado);
jPanelPesquisarChamado.setBounds(150, 0, 620, 620);

jPanelListaDeChamados.setLayout(null);

jButton3.setText("Ver");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jPanelListaDeChamados.add(jButton3);
jButton3.setBounds(220, 200, 100, 25);

jButton18.setText("Sair");
jButton18.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton18ActionPerformed(evt);
    }
});

jPanelListaDeChamados.add(jButton18);
jButton18.setBounds(380, 200, 110, 25);

jLabel15.setText("Lista de Chamados");
jPanelListaDeChamados.add(jLabel15);
jLabel15.setBounds(290, 30, 150, 15);

jPanelListaDeChamados.add(jComboBox5);
jComboBox5.setBounds(60, 100, 550, 21);

getContentPane().add(jPanelListaDeChamados);
jPanelListaDeChamados.setBounds(150, 0, 620, 540);

jPanelChamado.setLayout(null);

jButton48.setText("Alterar");
jButton48.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton48ActionPerformed(evt);
    }
});

```

```

jPanelChamado.add(jButton48);
jButton48.setBounds(160, 550, 130, 25);

jButton49.setText("Fechar Chamado");
jButton49.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton49ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton49);
jButton49.setBounds(160, 580, 130, 25);

jButton50.setText("Sair");
jButton50.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton50ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton50);
jButton50.setBounds(410, 550, 100, 25);

jButton51.setText("Excluir");
jButton51.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton51ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton51);
jButton51.setBounds(300, 580, 100, 25);

jLabel59.setText("Chamado");
jPanelChamado.add(jLabel59);
jLabel59.setBounds(210, 0, 70, 15);

jLabel60.setText("nro.");
jPanelChamado.add(jLabel60);
jLabel60.setBounds(10, 20, 50, 15);

jPanelChamado.add(jLabel61);
jLabel61.setBounds(50, 20, 60, 15);

jLabel62.setText("T\u00edtulo:");
jPanelChamado.add(jLabel62);
jLabel62.setBounds(10, 40, 50, 15);

jLabel63.setText("Descri\u00e7\u00e3o:");
jPanelChamado.add(jLabel63);
jLabel63.setBounds(10, 70, 70, 15);

jLabel64.setText("Dispositivo");
jPanelChamado.add(jLabel64);
jLabel64.setBounds(10, 150, 70, 15);

jLabel65.setText("Tipo:");
jPanelChamado.add(jLabel65);
jLabel65.setBounds(50, 170, 70, 15);

```

```
jLabel66.setText("Marca:");
jPanelChamado.add(jLabel66);
jLabel66.setBounds(50, 190, 70, 15);

jLabel67.setText("Modelo:");
jPanelChamado.add(jLabel67);
jLabel67.setBounds(50, 210, 70, 15);

jLabel68.setText("Setor:");
jPanelChamado.add(jLabel68);
jLabel68.setBounds(50, 230, 60, 15);

jLabel69.setText("Patrimonio:");
jPanelChamado.add(jLabel69);
jLabel69.setBounds(50, 250, 80, 15);

jLabel70.setText("Nro. S\u00e9rie:");
jPanelChamado.add(jLabel70);
jLabel70.setBounds(50, 270, 80, 15);

jLabel71.setText("Data abertura:");
jPanelChamado.add(jLabel71);
jLabel71.setBounds(10, 330, 90, 15);

jLabel72.setText("Data conclus\u00e3o:");
jPanelChamado.add(jLabel72);
jLabel72.setBounds(10, 350, 150, 15);

jLabel73.setText("Propriet\u00e1rio:");
jPanelChamado.add(jLabel73);
jLabel73.setBounds(10, 370, 90, 15);

jLabel74.setText("T\u00e9cnico respons\u00e1vel:");
jPanelChamado.add(jLabel74);
jLabel74.setBounds(10, 400, 140, 15);

jLabel75.setText("Status:");
jPanelChamado.add(jLabel75);
jLabel75.setBounds(10, 420, 100, 15);

jLabel76.setText("Providencias:");
jPanelChamado.add(jLabel76);
jLabel76.setBounds(10, 450, 120, 15);

jLabel77.setText("Assistencia:");
jPanelChamado.add(jLabel77);
jLabel77.setBounds(10, 510, 110, 15);

jLabel78.setText("Categoria do problema:");
jPanelChamado.add(jLabel78);
jLabel78.setBounds(10, 290, 150, 15);

jLabel79.setText("Urgencia:");
jPanelChamado.add(jLabel79);
jLabel79.setBounds(10, 310, 90, 15);

jPanelChamado.add(jTextField27);
jTextField27.setBounds(70, 40, 160, 20);

jPanelChamado.add(jTextField29);
jTextField29.setBounds(130, 250, 310, 21);
```



```

jPanelChamado.add(jTextField30);
jTextField30.setBounds(160, 290, 280, 21);

jPanelChamado.add(jTextField31);
jTextField31.setBounds(90, 310, 340, 21);

jPanelChamado.add(jTextField32);
jTextField32.setBounds(120, 330, 190, 21);

jPanelChamado.add(jTextField33);
jTextField33.setBounds(120, 350, 190, 21);

jPanelChamado.add(jTextField34);
jTextField34.setBounds(120, 370, 190, 21);

jPanelChamado.add(jTextField35);
jTextField35.setBounds(150, 400, 160, 21);

jPanelChamado.add(jTextArea2);
jTextArea2.setBounds(90, 450, 480, 50);

jPanelChamado.add(jTextArea3);
jTextArea3.setBounds(80, 70, 450, 80);

jCheckBox1.setText("Aberto");
jPanelChamado.add(jCheckBox1);
jCheckBox1.setBounds(80, 420, 110, 23);

jPanelChamado.add(jComboBox4);
jComboBox4.setBounds(90, 510, 230, 21);

jPanelChamado.add(jLabel20);
jLabel20.setBounds(90, 160, 170, 20);

jPanelChamado.add(jLabel32);
jLabel32.setBounds(100, 180, 160, 20);

jPanelChamado.add(jLabel46);
jLabel46.setBounds(110, 200, 160, 20);

jPanelChamado.add(jLabel47);
jLabel47.setBounds(100, 220, 190, 20);

jPanelChamado.add(jLabel49);
jLabel49.setBounds(120, 270, 140, 20);

jButton21.setText("Abandonar");
jButton21.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton21ActionPerformed(evt);
    }
});

jPanelChamado.add(jButton21);
jButton21.setBounds(410, 580, 100, 25);

jButton23.setText("Tratar");
jButton23.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton23ActionPerformed(evt);
    }
});

```

```

    }
});

jPanelChamado.add(jButton23);
jButton23.setBounds(300, 550, 100, 25);

getContentPane().add(jPanelChamado);
jPanelChamado.setBounds(150, 0, 640, 620);

jPanelCadastroDispositivo.setLayout(null);

jLabel21.setText("Cadastro de Dispositivo");
jPanelCadastroDispositivo.add(jLabel21);
jLabel21.setBounds(280, 10, 170, 15);

jLabel22.setText("Nro. Patrimonio:");
jPanelCadastroDispositivo.add(jLabel22);
jLabel22.setBounds(20, 50, 140, 15);

jLabel23.setText("Tipo:");
jPanelCadastroDispositivo.add(jLabel23);
jLabel23.setBounds(20, 80, 80, 15);

jLabel24.setText("Setor:");
jPanelCadastroDispositivo.add(jLabel24);
jLabel24.setBounds(20, 110, 60, 15);

jLabel25.setText("Fabricante:");
jPanelCadastroDispositivo.add(jLabel25);
jLabel25.setBounds(20, 140, 70, 15);

jLabel26.setText("Modelo:");
jPanelCadastroDispositivo.add(jLabel26);
jLabel26.setBounds(20, 170, 60, 15);

jLabel27.setText("Nro. S\u00e9rie:");
jPanelCadastroDispositivo.add(jLabel27);
jLabel27.setBounds(20, 200, 70, 15);

jPanelCadastroDispositivo.add(jTextField4);
jTextField4.setBounds(120, 50, 140, 21);

jPanelCadastroDispositivo.add(jTextField8);
jTextField8.setBounds(70, 80, 190, 21);

jPanelCadastroDispositivo.add(jTextField9);
jTextField9.setBounds(70, 110, 190, 21);

jPanelCadastroDispositivo.add(jTextField10);
jTextField10.setBounds(100, 140, 160, 21);

jPanelCadastroDispositivo.add(jTextField11);
jTextField11.setBounds(80, 170, 180, 21);

jPanelCadastroDispositivo.add(jTextField12);
jTextField12.setBounds(100, 200, 160, 21);

jButton24.setText("Criar");
jButton24.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton24ActionPerformed(evt);
    }
});

```

```

    }
  });

  jPanelCadastroDispositivo.add(jButton24);
  jButton24.setBounds(20, 270, 130, 25);

  jButton25.setText("Excluir");
  jButton25.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton25ActionPerformed(evt);
    }
  });

  jPanelCadastroDispositivo.add(jButton25);
  jButton25.setBounds(330, 270, 120, 25);

  jButton26.setText("Sair");
  jButton26.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton26ActionPerformed(evt);
    }
  });

  jPanelCadastroDispositivo.add(jButton26);
  jButton26.setBounds(480, 270, 120, 25);

  jButton14.setText("Modificar");
  jButton14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton14ActionPerformed(evt);
    }
  });

  jPanelCadastroDispositivo.add(jButton14);
  jButton14.setBounds(170, 270, 140, 25);

  getContentPane().add(jPanelCadastroDispositivo);
  jPanelCadastroDispositivo.setBounds(150, 0, 620, 540);

  jPanelPesquisarDispositivo.setLayout(null);

  jButton27.setText("Pesquisar");
  jButton27.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton27ActionPerformed(evt);
    }
  });

  jPanelPesquisarDispositivo.add(jButton27);
  jButton27.setBounds(80, 200, 130, 25);

  jButton28.setText("Sair");
  jButton28.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton28ActionPerformed(evt);
    }
  });

  jPanelPesquisarDispositivo.add(jButton28);
  jButton28.setBounds(270, 200, 140, 25);

```

```

jLabel28.setText("Pesquisar dispositivo");
jPanelPesquisarDispositivo.add(jLabel28);
jLabel28.setBounds(200, 30, 160, 15);

jLabel29.setText("Nro. patrimonio:");
jPanelPesquisarDispositivo.add(jLabel29);
jLabel29.setBounds(20, 90, 120, 15);

jPanelPesquisarDispositivo.add(jTextField13);
jTextField13.setBounds(120, 90, 200, 21);

getContentPane().add(jPanelPesquisarDispositivo);
jPanelPesquisarDispositivo.setBounds(150, 0, 640, 620);

jPanelListaDosDispositivos.setLayout(null);

jButton29.setText("Ver");
jButton29.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton29ActionPerformed(evt);
    }
});

jPanelListaDosDispositivos.add(jButton29);
jButton29.setBounds(90, 170, 100, 25);

jButton30.setText("Sair");
jButton30.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton30ActionPerformed(evt);
    }
});

jPanelListaDosDispositivos.add(jButton30);
jButton30.setBounds(260, 170, 110, 25);

jLabel30.setText("Lista dos Dispositivos");
jPanelListaDosDispositivos.add(jLabel30);
jLabel30.setBounds(180, 20, 190, 15);

jPanelListaDosDispositivos.add(jComboBox3);
jComboBox3.setBounds(90, 100, 280, 21);

getContentPane().add(jPanelListaDosDispositivos);
jPanelListaDosDispositivos.setBounds(150, 0, 640, 620);

jPanelTrocarSenha.setLayout(null);

jLabel31.setText("Trocar senha");
jPanelTrocarSenha.add(jLabel31);
jLabel31.setBounds(230, 20, 150, 15);

jLabel33.setText("Nova senha:");
jPanelTrocarSenha.add(jLabel33);
jLabel33.setBounds(40, 110, 120, 15);

jLabel34.setText("Repetir nova senha:");
jPanelTrocarSenha.add(jLabel34);
jLabel34.setBounds(40, 150, 140, 15);

jButton31.setText("Submeter");

```

```

jButton31.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton31ActionPerformed(evt);
    }
});

jPanelTrocarSenha.add(jButton31);
jButton31.setBounds(129, 240, 120, 25);

jButton32.setText("Cancelar");
jButton32.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton32ActionPerformed(evt);
    }
});

jPanelTrocarSenha.add(jButton32);
jButton32.setBounds(297, 240, 130, 25);

jPanelTrocarSenha.add(jPasswordField3);
jPasswordField3.setBounds(130, 110, 170, 23);

jPanelTrocarSenha.add(jPasswordField4);
jPasswordField4.setBounds(150, 150, 150, 23);

getContentPane().add(jPanelTrocarSenha);
jPanelTrocarSenha.setBounds(150, 0, 640, 620);

java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setBounds((screenSize.width-800)/2, (screenSize.height-650)/2, 800,
650);
} //GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButton1ActionPerformed
    if
(jToggleButton1.isSelected()){this.disponibilizarFrameTecnico_Utente();}else
{this.disponibilizarFrameTecnico();}
} //GEN-LAST:event_jButton1ActionPerformed

private void jButton23ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton23ActionPerformed
    int indice = Integer.parseInt(jLabel61.getText());
    if
((jTextField30.getText().compareTo("")==0)|| (jTextArea2.getText().compareTo
(""))==0){
        JOptionPane.showMessageDialog(null,"Preencha 'Categoria' e
'Providencias'");
    }else{
        String cat = jTextField30.getText();
        String prov = jTextArea2.getText();
        boolean abe;
        if (jCheckBox1.isSelected()){abe=true;}else{abe=false;}
        ioesc.trateChamado(indice, cat, prov, abe);
        jPanelChamado.setVisible(false);
    }
} //GEN-LAST:event_jButton23ActionPerformed

private void jButton21ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton21ActionPerformed

```

```

        int indice = Integer.parseInt(jLabel61.getText());
        if
((jTextField30.getText().compareTo("")==0)|| (jTextArea2.getText().compareTo
(""))==0)){
            JOptionPane.showMessageDialog(null,"Preencha 'Categoria' e
'Providencias'.");
        }else{
            String cat = jTextField30.getText();
            String prov = jTextArea2.getText();
            boolean abe = true;
            boolean necass = true;
            ioesc.abandoneChamado(indice,cat,prov,abe,necass);
            jPanelChamado.setVisible(false);
        }
    }//GEN-LAST:event_jButtonon21ActionPerformed

    private void jButtonon14ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonon14ActionPerformed
        String p=jTextField4.getText();
        String t=jTextField8.getText();
        String s=jTextField9.getText();
        String f=jTextField10.getText();
        String m=jTextField11.getText();
        String n=jTextField12.getText();
        if ((ioesc.existeDispositivoCadastrado(p)) &&
(jTextField4.getText().compareTo("")!=0)) {
            ioesc.modificarDadosDispositivo(p,t,s,f,m,n);
            jPanelCadastroDispositivo.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Usuário não existe e não
pode ser alterado.");
        }
        this.limparInterface();
    }//GEN-LAST:event_jButtonon14ActionPerformed

    private void botaoLogoffActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_botaoLogoffActionPerformed
        this.limparFrame();
        this.limparInterface();
        jPanelIdentificacao.setVisible(true);
        usuarioLogado = "";
    }//GEN-LAST:event_botaoLogoffActionPerformed

    private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jPasswordField1ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jPasswordField1ActionPerformed

    private void jButtonon51ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonon51ActionPerformed
        int indice = Integer.parseInt(jLabel61.getText());
        if (ioesc.existeTicketCadastrado(indice)){
            ioesc.excluirTicket(indice);
            this.limparTelaVerChamado();
            jPanelChamado.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Ticket não consta no Banco
de Dados.");
        }
    }//GEN-LAST:event_jButtonon51ActionPerformed

```

```

        private void jButton50ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton50ActionPerformed
        jPanelChamado.setVisible(false);
        this.limparTelaVerChamado();
    } //GEN-LAST:event_jButton50ActionPerformed

        private void jButton49ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton49ActionPerformed
        int indice = Integer.parseInt(jLabel61.getText());
        if
    ((jTextField30.getText().compareTo("")==0) || (jTextField33.getText().compare
    To("")==0) || (jTextArea2.getText().compareTo("")==0)){
            JOptionPane.showMessageDialog(null,"Preencha 'Categoria, 'Data
    de conclusão' e 'Providencias'");
        }else{
            String cat = jTextField30.getText();
            String f_fim = jTextField33.getText();
            String prov = jTextArea2.getText();
            String ass = (String)jComboBox4.getSelectedItem();
            boolean abe = false;
            ioesc.fecheChamado(indice,cat,f_fim,prov,ass,abe);
            jPanelChamado.setVisible(false);
        }
    } //GEN-LAST:event_jButton49ActionPerformed

        private void jButton48ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton48ActionPerformed
        String p_disp = jTextField29.getText();
        if (ioesc.existeDispositivoCadastrado(p_disp)){
            int indice = Integer.parseInt(jLabel61.getText());
            String tit = jTextField27.getText();
            String desc = jTextArea3.getText();
            String cat = jTextField30.getText();
            String urg = jTextField31.getText();
            String d_ini = jTextField32.getText();
            String d_fim = jTextField33.getText();
            String l_prop = jTextField34.getText();
            String l_resp = jTextField35.getText();
            boolean abe;
            if (jCheckBox1.isSelected()){abe = true;}else{abe = false;}
            String ass = ((String)jComboBox4.getSelectedItem());
            String prov = jTextArea2.getText();

            ioesc.alteraChamado(indice,tit,desc,p_disp,cat,urg,d_ini,d_fim,l_prop,l_res
            p,abe,ass,prov);
            jPanelChamado.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Infelizmente não existe
            este dispositivo cadastrado.");
        }
    } //GEN-LAST:event_jButton48ActionPerformed

        private void jButton32ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton32ActionPerformed
        jPanelTrocarSenha.setVisible(false);
    } //GEN-LAST:event_jButton32ActionPerformed

        private void jButton31ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton31ActionPerformed
        jPanelTrocarSenha.setVisible(false);
    }

```

```

        if
(jPasswordField3.getText().compareTo(jPasswordField4.getText())==0){
            ioesc.troqueSenha(usuarioLogado,jPasswordField3.getText());
        }
    }//GEN-LAST:event_jButton31ActionPerformed

    private void botaoTrocarSenhaActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_botaoTrocarSenhaActionPerformed
        this.limparInterface();
        jPanelTrocarSenha.setVisible(true);
    } //GEN-LAST:event_botaoTrocarSenhaActionPerformed

    private void jButton29ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton29ActionPerformed

        jPanelListaDosDispositivos.setVisible(false);
        jPanelCadastroDispositivo.setVisible(true);
        this.limparTelaCadastroEquipamento();

        String nro_patrr = (String) jComboBox3.getSelectedItemAt();

        ArrayList oDispositivo = new ArrayList();
        oDispositivo = ioesc.getInformacoesDispositivo(nro_patrr);

        jTextField4.setText(nro_patrr);
        jTextField8.setText((String)oDispositivo.get(1));
        jTextField9.setText((String)oDispositivo.get(2));
        jTextField10.setText((String)oDispositivo.get(3));
        jTextField11.setText((String)oDispositivo.get(4));
        jTextField12.setText((String)oDispositivo.get(5));
    } //GEN-LAST:event_jButton29ActionPerformed

    private void jButton30ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton30ActionPerformed
        jPanelListaDosDispositivos.setVisible(false);
        this.limparInterface();
    } //GEN-LAST:event_jButton30ActionPerformed

    private void jButton27ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton27ActionPerformed
        String nrodisp = jTextField13.getText();
        jComboBox3.removeAllItems();
        this.limparInterface();

        if (ioesc.existeDispositivoCadastrado(nrodisp)){
            jComboBox3.addItem(nrodisp);
            jPanelPesquisarDispositivo.setVisible(false);
            jPanelListaDosDispositivos.setVisible(true);
        }else{
            this.limparInterface();
            JOptionPane.showMessageDialog(null,"Dispositivo não
cadastrado.");
        }
    } //GEN-LAST:event_jButton27ActionPerformed

    private void jButton28ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton28ActionPerformed
        jPanelPesquisarDispositivo.setVisible(false);
        this.limparInterface();
    } //GEN-LAST:event_jButton28ActionPerformed

```



```

        private void jButton26ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_jButton26ActionPerformed
            jPanelCadastroDispositivo.setVisible(false);
            this.limparInterface();
        } //GEN-LAST:event_jButton26ActionPerformed

        private void jButton25ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_jButton25ActionPerformed
            String patr=jTextField4.getText();
            if
            ((patr.compareTo("")!=0)&&(ioesc.existeDispositivoCadastrado(patr))) {
                jPanelCadastroDispositivo.setVisible(false);
                ioesc.excluirDispositivo(patr);
                this.limparTelaCadastroEquipamento();
            } else {
                JOptionPane.showMessageDialog(null,"Não foi possível
excluir.");
            }
        } //GEN-LAST:event_jButton25ActionPerformed

        private void jButton24ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_jButton24ActionPerformed
            String p = jTextField4.getText();
            String t = jTextField8.getText();
            String s = jTextField9.getText();
            String f = jTextField10.getText();
            String m = jTextField11.getText();
            String ns = jTextField12.getText();
            if ((ioesc.existeDispositivoCadastrado(p)) || (p.compareTo("")==0)) {
                this.limparInterface();
                JOptionPane.showMessageDialog(null,"Dispositivo NÃO pode ser
cadastrado.");
            } else {
                ioesc.criarDispositivo(p,t,s,f,m,ns);
            }
            jPanelCadastroDispositivo.setVisible(false);
            this.limparInterface();
        } //GEN-LAST:event_jButton24ActionPerformed

        private void botaoCadastrarActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_botaoCadastrarActionPerformed
            this.limparInterface();
            this.limparTelaCadastroEquipamento();
            jPanelCadastroDispositivo.setVisible(true);
        } //GEN-LAST:event_botaoCadastrarActionPerformed

        private void botaoPesquisarActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_botaoPesquisarActionPerformed
            this.limparInterface();
            jPanelPesquisarChamado.setVisible(true);
            jTextField3.setText("");
            jComboBox6.removeAllItems();
            jComboBox6.addItem("");
            jComboBox6.addItem("Patrimonio");
            jComboBox6.addItem("Nro Ticket");
        } //GEN-LAST:event_botaoPesquisarActionPerformed

        private void jButton20ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_jButton20ActionPerformed
            jPanelPesquisarChamado.setVisible(false);

```

```

} //GEN-LAST:event_jButton20ActionPerformed

private void jButton19ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton19ActionPerformed

    jComboBox5.removeAllItems();
    this.limparInterface();
    jPanelPesquisarChamado.setVisible(true);
    String valorPesquisa = jTextField3.getText();

    if
(((String)jComboBox6.getSelectedItem()).compareTo("Patrimonio")==0){
        jPanelListaDeChamados.setVisible(true);
        ArrayList listaIdChamados = new ArrayList();
        listaIdChamados = ioesc.getListaIdChamados(true,valorPesquisa);
        for (int i=0;i<listaIdChamados.size();i++){
            Integer aux = (Integer)listaIdChamados.get(i);
            jComboBox5.addItem(aux);
        }
        jPanelPesquisarChamado.setVisible(false);
    }else{
        if (((String)jComboBox6.getSelectedItem()).compareTo("Nro
Ticket")==0){
            jPanelListaDeChamados.setVisible(true);
            ArrayList listaIdChamados = new ArrayList();
            listaIdChamados =
ioesc.getListaIdChamados(false,valorPesquisa);
            for (int i=0;i<listaIdChamados.size();i++){
                Integer aux = (Integer)listaIdChamados.get(i);
                jComboBox5.addItem(aux);
            }
            jPanelPesquisarChamado.setVisible(false);
        }else{
            JOptionPane.showMessageDialog(null,"Selecione um tipo de
pesquisa.");
        }
    }
} //GEN-LAST:event_jButton19ActionPerformed

private void jButton18ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton18ActionPerformed
    jPanelListaDeChamados.setVisible(false);
} //GEN-LAST:event_jButton18ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton3ActionPerformed

    jComboBox4.removeAllItems();
    this.limparInterface();
    this.limparTelaVerChamado();
    jPanelChamado.setVisible(true);

    ArrayList oChamado = new ArrayList();

    Integer ID_inteiro = (Integer) jComboBox5.getSelectedItem();
    int ID = (ID_inteiro.intValue());

    oChamado = ioesc.getInformacoesChamado(ID);

```

```

jLabel20.setText((String)oChamado.get(0));
jLabel32.setText((String)oChamado.get(1));
jLabel46.setText((String)oChamado.get(2));
jLabel47.setText((String)oChamado.get(3));
jLabel49.setText((String)oChamado.get(4));

jLabel61.setText((String)oChamado.get(5));
jTextField27.setText((String)oChamado.get(6));
jTextArea3.setText((String)oChamado.get(7));
jTextField29.setText((String)oChamado.get(8));
jTextField30.setText((String)oChamado.get(9));
jTextField31.setText((String)oChamado.get(10));
jTextField32.setText((String)oChamado.get(11));
jTextField33.setText((String)oChamado.get(12));
jTextField34.setText((String)oChamado.get(13));
jTextField35.setText((String)oChamado.get(14));
if (((String)oChamado.get(15)).compareTo("true")==0)
{jCheckBox1.setSelected(true);}else{jCheckBox1.setSelected(false);}
jComboBox4.addItem((String)oChamado.get(16));
jTextArea2.setText((String)oChamado.get(17));

ArrayList assistenciasCadastradas = new ArrayList();
assistenciasCadastradas = ioesc.getListaAssistenciasCadastradas();
for (int i=0;i<assistenciasCadastradas.size();i++){
    String ass = (String)assistenciasCadastradas.get(i);
    jComboBox4.addItem(ass);
}
} //GEN-LAST:event_jButton3ActionPerformed

private void
botaoListarAbertosActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_botaoListarAbertosActionPerformed
    jComboBox5.removeAllItems();
    this.limparInterface();
    jPanelListaDeChamados.setVisible(true);
    ArrayList listaIdChamadosAbertos = new ArrayList();
    listaIdChamadosAbertos = ioesc.getListaIdChamadosAbertos();
    for (int i=0;i<listaIdChamadosAbertos.size();i++){
        Integer aux = (Integer)listaIdChamadosAbertos.get(i);
        jComboBox5.addItem(aux);
    }
} //GEN-LAST:event_botaoListarAbertosActionPerformed

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton17ActionPerformed
    jPanelAbrirChamado.setVisible(false);
} //GEN-LAST:event_jButton17ActionPerformed

private void jButton16ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton16ActionPerformed
    String p = jTextField6.getText();
    String t = jTextField2.getText();
    String d = jTextArea1.getText();
    String u = jTextField7.getText();
    String dt = jTextField5.getText();
    if (ioesc.existeDispositivoCadastrado(p)){
        ioesc.criarTicket(t,d,u,dt,p,usuarioLogado);
        jPanelAbrirChamado.setVisible(false);
        this.limparInterface();
    }else{

```

```

        JOptionPane.showMessageDialog(null,"Dispositivo não está
cadastrado. Entre em contato com a SUATEC.");
    }
} //GEN-LAST:event_jButton16ActionPerformed

private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jTextField6ActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_jTextField6ActionPerformed

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jTextField2ActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_jTextField2ActionPerformed

private void
botaoPesquisarDispositivoActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_botaoPesquisarDispositivoActionPerformed
    this.limparInterface();
    jPanelPesquisarDispositivo.setVisible(true);
    jTextField13.setText("");
} //GEN-LAST:event_botaoPesquisarDispositivoActionPerformed

private void botaoAbrirActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_botaoAbrirActionPerformed
    this.limparInterface();
    this.limparTelaAberturaChamado();
    jPanelAbrirChamado.setVisible(true);
} //GEN-LAST:event_botaoAbrirActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
    String l=jTextField1.getText();
    String s=jPasswordField1.getText();
    if (ioesc.acessarSistema(l,s)){
        jPanelIdentificacao.setVisible(false);
        if
(ioesc.ehTecnico(l)){this.disponibilizarFrameTecnico();}else{this.disponibi
lizarFrameUtente();jToggleButton1.setVisible(false);}
        usuarioLogado = l;
    }else{
        JOptionPane.showMessageDialog(null,"Login e Senha não
conferem!");
    }
    jTextField1.setText("");
    jPasswordField1.setText("");
} //GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
    System.exit(0);
} //GEN-LAST:event_jButton2ActionPerformed

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jTextField1ActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_jTextField1ActionPerformed

/***** Exit the Application *****/
private void exitForm(java.awt.event.WindowEvent evt) { //GEN-
FIRST:event_exitForm

```

```

        System.exit(0);
    } //GEN-LAST:event_exitForm

    public static void main(String args[]) {
        new PasClient().show();
    }

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JButton botaoAbrir;
    private javax.swing.JButton botaoCadastrar;
    private javax.swing.JButton botaoListarAbertos;
    private javax.swing.JButton botaoLogoff;
    private javax.swing.JButton botaoPesquisar;
    private javax.swing.JButton botaoPesquisarDispositivo;
    private javax.swing.JButton botaoTrocarSenha;
    private javax.swing.JButton jButtonon1;
    private javax.swing.JButton jButtonon14;
    private javax.swing.JButton jButtonon16;
    private javax.swing.JButton jButtonon17;
    private javax.swing.JButton jButtonon18;
    private javax.swing.JButton jButtonon19;
    private javax.swing.JButton jButtonon2;
    private javax.swing.JButton jButtonon20;
    private javax.swing.JButton jButtonon21;
    private javax.swing.JButton jButtonon23;
    private javax.swing.JButton jButtonon24;
    private javax.swing.JButton jButtonon25;
    private javax.swing.JButton jButtonon26;
    private javax.swing.JButton jButtonon27;
    private javax.swing.JButton jButtonon28;
    private javax.swing.JButton jButtonon29;
    private javax.swing.JButton jButtonon3;
    private javax.swing.JButton jButtonon30;
    private javax.swing.JButton jButtonon31;
    private javax.swing.JButton jButtonon32;
    private javax.swing.JButton jButtonon48;
    private javax.swing.JButton jButtonon49;
    private javax.swing.JButton jButtonon50;
    private javax.swing.JButton jButtonon51;
    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JComboBox jComboBox3;
    private javax.swing.JComboBox jComboBox4;
    private javax.swing.JComboBox jComboBox5;
    private javax.swing.JComboBox jComboBox6;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;

```

```
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel46;
private javax.swing.JLabel jLabel47;
private javax.swing.JLabel jLabel49;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel59;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel60;
private javax.swing.JLabel jLabel61;
private javax.swing.JLabel jLabel62;
private javax.swing.JLabel jLabel63;
private javax.swing.JLabel jLabel64;
private javax.swing.JLabel jLabel65;
private javax.swing.JLabel jLabel66;
private javax.swing.JLabel jLabel67;
private javax.swing.JLabel jLabel68;
private javax.swing.JLabel jLabel69;
private javax.swing.JLabel jLabel70;
private javax.swing.JLabel jLabel71;
private javax.swing.JLabel jLabel72;
private javax.swing.JLabel jLabel73;
private javax.swing.JLabel jLabel74;
private javax.swing.JLabel jLabel75;
private javax.swing.JLabel jLabel76;
private javax.swing.JLabel jLabel77;
private javax.swing.JLabel jLabel78;
private javax.swing.JLabel jLabel79;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanelAbrirChamado;
private javax.swing.JPanel jPanelCadastroDispositivo;
private javax.swing.JPanel jPanelChamado;
private javax.swing.JPanel jPanelIdentificacao;
private javax.swing.JPanel jPanelListaDeChamados;
private javax.swing.JPanel jPanelListaDosDispositivos;
private javax.swing.JPanel jPanelPesquisarChamado;
private javax.swing.JPanel jPanelPesquisarDispositivo;
private javax.swing.JPanel jPanelTrocarSenha;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JPasswordField jPasswordField3;
private javax.swing.JPasswordField jPasswordField4;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
private javax.swing.JTextArea jTextArea3;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField27;
private javax.swing.JTextField jTextField29;
```

```

private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField30;
private javax.swing.JTextField jTextField31;
private javax.swing.JTextField jTextField32;
private javax.swing.JTextField jTextField33;
private javax.swing.JTextField jTextField34;
private javax.swing.JTextField jTextField35;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
private javax.swing.JToggleButton jButton1;
private javax.swing.JLabel labelTechnico;
private javax.swing.JLabel labelUtente;
// End of variables declaration//GEN-END:variables

private static Stub createProxy(){
    return (Stub) (new MinhaEmpresa_Impl().getEmpresaInterfacePort());
}

}

/*****Ticket.java*****/
package suatec;

import java.util.*;
public class Ticket {
    private int id;
    private String patr_dispositivo;
    private String data_inicio;
    private String data_fim;
    private String login_proprietario;
    private String login_responsavel;
    private String titulo;
    private String descricao;
    private String urgencia;
    private String categoria;
    private String providencia;
    private String assistencia;
    private boolean necessita_assistencia;
    private boolean aberto;

    public Ticket() {
    }

    public void setId(int umId) {
        id=umId;
    }

    public int getId() {
        return id;
    }

    public void setPatr_dispositivo(String umPatr_dispositivo) {
        patr_dispositivo=umPatr_dispositivo;
    }

    public String getPatr_dispositivo() {

```

```
        return patr_dispositivo;
    }

    public void setData_inicio(String umaData_inicio) {
        data_inicio=umaData_inicio;
    }

    public String getData_inicio() {
        return data_inicio;
    }

    public void setData_fim(String umaData_fim) {
        data_fim=umaData_fim;
    }

    public String getData_fim() {
        return data_fim;
    }

    public void setLogin_proprietario(String umLogin_proprietario) {
        login_proprietario=umLogin_proprietario;
    }

    public String getLogin_proprietario() {
        return login_proprietario;
    }

    public void setLogin_responsavel(String umLogin_responsavel) {
        login_responsavel=umLogin_responsavel;
    }

    public String getLogin_responsavel() {
        return login_responsavel;
    }

    public void setTitulo(String umTitulo) {
        titulo=umTitulo;
    }

    public String getTitulo() {
        return titulo;
    }

    public void setDescricao(String umaDescricao) {
        descricao=umaDescricao;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setUrgencia(String umaUrgencia) {
        urgencia=umaUrgencia;
    }

    public String getUrgencia() {
        return urgencia;
    }

    public void setCategoria(String umaCategoria) {
        categoria=umaCategoria;
    }
```



```

    }

    public String getCategoria() {
        return categoria;
    }

    public void setProvidencia(String umaProvidencia) {
        providencia=umaProvidencia;
    }

    public String getProvidencia() {
        return providencia;
    }

    public void setAssistencia(String umaAssistencia) {
        asistencia=umaAssistencia;
    }

    public String getAssistencia() {
        return asistencia;
    }

    public void setNecessita_assistencia(boolean umaNecessita_assistencia)
{
        necessita_assistencia=umaNecessita_assistencia;
    }

    public boolean getNecessita_assistencia() {
        return necessita_assistencia;
    }

    public void setAberto(boolean umAberto) {
        aberto=umAberto;
    }

    public boolean getAberto() {
        return aberto;
    }
}

```

```

/*****TicketDAO.java*****/

```

```

package suatec;

```

```

import java.sql.Connection;

```

```

import java.sql.PreparedStatement;

```

```

import java.sql.SQLException;

```

```

import java.util.ArrayList;

```

```

import java.sql.ResultSet;

```

```

public class TicketDAO extends GenericDAO {
    protected static PreparedStatement psSelectTicket;
    protected static PreparedStatement psSelectTodosTicket;
    protected static PreparedStatement psUpdateTicket;
    protected static PreparedStatement psInsertTicket;
    protected static PreparedStatement psDeleteTicket;

    public TicketDAO(int indice) {
        super();
    }
}

```

```

        setIndice(indice);
    }

    public void initialize() {
        try {
            psSelectTicket = getConnection(indice).prepareStatement("
SELECT T.id, T.login_proprietario, T.login_responsavel, T.patr_dispositivo,
T.titulo, T.descripcion, T.urgencia, T.categoria, T.providencia,
T.necesita_asistencia, T.asistencia, T.aberto, T.data_inicio, T.data_fim
FROM ticket T");
            psSelectTodosTicket = getConnection(indice).prepareStatement("
SELECT T.id, T.login_proprietario, T.login_responsavel, T.patr_dispositivo,
T.titulo, T.descripcion, T.urgencia, T.categoria, T.providencia,
T.necesita_asistencia, T.asistencia, T.aberto, T.data_inicio, T.data_fim
FROM ticket T");
            psUpdateTicket = getConnection(indice).prepareStatement("
UPDATE ticket SET login_proprietario = ?, login_responsavel = ?,
patr_dispositivo = ?, titulo = ?, descripcion = ?, urgencia = ?, categoria =
?, providencia = ?, necesita_asistencia = ?, asistencia = ?, abierto = ?,
data_inicio = ?, data_fim = ? WHERE id = ?");
            psInsertTicket = getConnection(indice).prepareStatement("
INSERT INTO ticket (id, login_proprietario, login_responsavel,
patr_dispositivo, titulo, descripcion, urgencia, categoria, providencia,
necesita_asistencia, asistencia, abierto, data_inicio, data_fim ) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?)");
            psDeleteTicket = getConnection(indice).prepareStatement("
DELETE FROM ticket WHERE id = ?");
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList selectTickets(int abierto) throws SQLException {
        ArrayList tickets = new ArrayList();
        Ticket ticket = null;
        try {
            ResultSet rs = psSelectTicket.executeQuery();
            int i = 0;
            while (rs.next()) {
                try {
                    if (rs.getInt("abierto") == abierto) {
                        ticket = new Ticket();
                        ticket.setId(rs.getInt("id"));

ticket.setLogin_proprietario(rs.getString("login_proprietario"));

ticket.setLogin_responsavel(rs.getString("login_responsavel"));

ticket.setPatr_dispositivo(rs.getString("patr_dispositivo"));
                        ticket.setTitulo(rs.getString("titulo"));
                        ticket.setDescripcion(rs.getString("descripcion"));
                        ticket.setUrgencia(rs.getString("urgencia"));
                        ticket.setCategoria(rs.getString("categoria"));
                        ticket.setProvidencia(rs.getString("providencia"));
                        if (rs.getInt("necesita_asistencia")==0)
{ticket.setNecesita_asistencia(false);} else
{ticket.setNecesita_asistencia(true);}
                        ticket.setAsistencia(rs.getString("asistencia"));
                    }
                }
            }
        }
    }

```

```

        if (rs.getInt("aberto")==0)
{ticket.setAberto(false);} else {ticket.setAberto(true);}
        ticket.setData_inicio(rs.getString("data_inicio"));
        ticket.setData_fim(rs.getString("data_fim"));
        tickets.add(ticket);
        i++;
    }
}
    catch(Exception e) {
        e.printStackTrace();
    }
}
}
catch(SQLException e) {
    e.printStackTrace();
}
return tickets;
}

public ArrayList selectTodosTickets() throws SQLException {
    ArrayList tickets = new ArrayList();
    Ticket ticket = null;
    try {
        ResultSet rs = psSelectTicket.executeQuery();
        int i = 0;
        while (rs.next()) {
            try {
                ticket = new Ticket();
                ticket.setId(rs.getInt("id"));

ticket.setLogin_proprietario(rs.getString("login_proprietario"));

ticket.setLogin_responsavel(rs.getString("login_responsavel"));

ticket.setPatr_dispositivo(rs.getString("patr_dispositivo"));
                ticket.setTitulo(rs.getString("titulo"));
                ticket.setDescricao(rs.getString("descricao"));
                ticket.setUrgencia(rs.getString("urgencia"));
                ticket.setCategoria(rs.getString("categoria"));
                ticket.setProvidencia(rs.getString("providencia"));
                if (rs.getInt("necessita_assistencia")==0)
{ticket.setNecessita_assistencia(false);} else
{ticket.setNecessita_assistencia(true);}
                ticket.setAssistencia(rs.getString("assistencia"));
                if (rs.getInt("aberto")==0) {ticket.setAberto(false);}
            else {ticket.setAberto(true);}
                ticket.setData_inicio(rs.getString("data_inicio"));
                ticket.setData_fim(rs.getString("data_fim"));
                tickets.add(ticket);
                i++;
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
    }
    return tickets;
}
}

```

```

public boolean updateTicket(Ticket ticket) throws SQLException {
    boolean retorno = false;
    try {
        psUpdateTicket.setString(1, ticket.getLogin_proprietario());
        psUpdateTicket.setString(2, ticket.getLogin_responsavel());
        psUpdateTicket.setString(3, ticket.getPatr_dispositivo());
        psUpdateTicket.setString(4, ticket.getTitulo());
        psUpdateTicket.setString(5, ticket.getDescricao());
        psUpdateTicket.setString(6, ticket.getUrgencia());
        psUpdateTicket.setString(7, ticket.getCategoria());
        psUpdateTicket.setString(8, ticket.getProvidencia());
        if
(ticket.getNecessita_assistencia()==false){psUpdateTicket.setInt(9,0);}else
{psUpdateTicket.setInt(9,1);}
        psUpdateTicket.setString(10, ticket.getAssistencia());
        if
(ticket.getAberto()==false){psUpdateTicket.setInt(11,0);}else{psUpdateTicke
t.setInt(11,1);}
        psUpdateTicket.setString(12, ticket.getData_inicio());
        psUpdateTicket.setString(13, ticket.getData_fim());
        psUpdateTicket.setLong (14, ticket.getId());
        retorno = psUpdateTicket.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean insertTicket(Ticket ticket) throws SQLException {
    boolean retorno = false;
    try {
        psInsertTicket.setLong (1, ticket.getId());
        psInsertTicket.setString(2, ticket.getLogin_proprietario());
        psInsertTicket.setString(3, ticket.getLogin_responsavel());
        psInsertTicket.setString(4, ticket.getPatr_dispositivo());
        psInsertTicket.setString(5, ticket.getTitulo());
        psInsertTicket.setString(6, ticket.getDescricao());
        psInsertTicket.setString(7, ticket.getUrgencia());
        psInsertTicket.setString(8, ticket.getCategoria());
        psInsertTicket.setString(9, ticket.getProvidencia());
        if
(ticket.getNecessita_assistencia()==false){psInsertTicket.setInt(10,0);}els
e{psInsertTicket.setInt(10,1);}
        psInsertTicket.setString(11,ticket.getAssistencia());
        if
(ticket.getAberto()==false){psInsertTicket.setInt(12,0);}else{psInsertTicke
t.setInt(12,1);}
        psInsertTicket.setString(13,ticket.getData_inicio());
        psInsertTicket.setString(14,ticket.getData_fim());
        retorno = psInsertTicket.execute();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return retorno;
}

public boolean deleteTicket(long id) throws SQLException {

```

```

        boolean retorno = false;
        try {
            psDeleteTicket.setLong(1, id);
            retorno = psDeleteTicket.execute();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return retorno;
    }

    public void setIndice(int indice) {
        this.indice = indice;
    }

    public void setTabela(String tabela_) {
        this.tabela = tabela_;
    }
}

/*****Usuario.java*****/

package suatec;

public class Usuario
{
    private int id;
    private String nome;
    private String sobrenome;
    private String cargo;
    private String setor;
    private String matricula;
    private String login;
    private String senha;

    private int nivel;

    public Usuario() {
    }

    public void setNome(String umNome) {
        nome=umNome;
    }

    public String getNome() {
        return nome;
    }

    public void setSobrenome(String umSobrenome) {
        sobrenome=umSobrenome;
    }

    public String getSobrenome() {
        return sobrenome;
    }

    public void setCargo(String umCargo) {
        cargo=umCargo;
    }
}

```

```
    }

    public String getCargo() {
        return cargo;
    }

    public void setSetor(String umSetor) {
        setor=umSetor;
    }

    public String getSetor() {
        return setor;
    }

    public void setMatricula(String umaMatricula) {
        matricula=umaMatricula;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setLogin(String umLogin) {
        login=umLogin;
    }

    public String getLogin() {
        return login;
    }

    public void setSenha(String umaSenha) {
        senha=umaSenha;
    }

    public String getSenha() {
        return senha;
    }

    public void setNivel(int umNivel) {
        nivel=umNivel;
    }

    public int getNivel() {
        return nivel;
    }

    public void setId(int id_) {
        id=id_;
    }

    public int getId() {
        return id;
    }
}
```

Anexo 2 - Artigo referente ao trabalho

Sistema de controle de suporte técnico utilizando Web Service na Imprensa Oficial do Estado de Santa Catarina

Fábio de Paula Vieira

fabiov@inf.ufsc.br

Sistemas de Informação – Universidade Federal de Santa Catarina

Orientador: José Eduardo De Lucca

***Resumo.** Diante da crescente inclusão de equipamentos de informática nas mais diferentes modalidades de trabalho na Imprensa Oficial do Estado de Santa Catarina, um sistema se tornou preciso para gerenciar a rotina que envolve o processo de manutenção de equipamentos. Atendendo às características específicas do ambiente, um sistema distribuído utilizando Web Service se mostrou atraente e vantajoso, uma vez que utiliza padrões abertos e tem características interoperáveis.*

***Abstract.** Due to an increasingly use of computing equipments at Imprensa Oficial do Estado de Santa Catarina, a maintenance management system became mandatory. Dealing with an idiosyncratic environment, a distributed system using Web Services presents advantages once it uses open standards and interoperability.*

1. Introdução

O avanço tecnológico e científico alterou o panorama em quase todas as atividades e processos de produção. Uma evidência disto é a crescente utilização de equipamentos de automação no ambiente de trabalho das pessoas, proporcionado essencialmente pela diminuição gradativa do custo dos equipamentos devido à ampliação de suas produções.

O computador e seus periféricos se tornaram talvez os dispositivos de automação mais difundidos existentes hoje. Sua utilização é muito variada e pode ser encontrado desde pequenos estabelecimentos comerciais a grandes centros de processamentos de dados. Assim sendo, como qualquer outra máquina, pode apresentar problema.

Devido ao crescente uso da Internet e da programação distribuída, um sistema de manutenção utilizando estes recursos se mostra atrativo para empresas ou entidades que possuem unidades de atuação fisicamente separadas.

O Web Service, modelo de computação distribuída recente que utiliza padrões abertos, funciona como um provedor de serviços remotos que pode ser utilizado por diferentes programas escritos para diferentes plataformas. Um servidor, conhecido por um endereço, fornece serviços que estão especificados e publicados para algum cliente interessado utilizá-lo.

O sistema proposto e objeto de estudo em questão tem por objetivo gerenciar, atribuir tarefas para as pessoas responsáveis e acompanhar as solicitações de serviço de manutenção até sua conclusão. O sistema também mantém relatórios de atendimento, com as soluções aplicadas, para que possam vir a ser utilizado no futuro como uma base de conhecimento para soluções mais ágeis.

2. IOESC

A Imprensa Oficial do Estado de Santa Catarina – IOESC – é uma empresa pública vinculada a Secretaria de Estado da Administração que tem por objetivo divulgar os atos oficiais do Estado de Santa Catarina, através de obras gráficas em diversas modalidades, publicando o Diário Oficial, o Diário da Justiça, materiais de campanha do governo e diversas outras publicações de interesse do Estado.

A Imprensa conta com uma unidade de atuação no centro da cidade onde representa a empresa e facilita o acesso aos serviços prestados aos cidadãos. A administração, o parque gráfico, e a infra-estrutura de informática estão localizados na matriz no bairro Saco dos Limões em Florianópolis, capital do Estado.

Parte do conteúdo produzido pela IOESC é disponibilizado às pessoas pela Internet através do Diário Oficial on-line que, por enquanto, conta apenas com o Diário da Justiça, pois o projeto de implantação do Diário Oficial ainda não foi concluído.

A Imprensa Oficial do Estado de Santa Catarina possui um setor responsável pela área tecnológica da instituição: a SUATEC, sigla de Supervisão de Atualização Tecnológica. Este setor é encarregado de manter os recursos de automação e informação da empresa. Ela está vinculada à Gerência de Planejamento da Produção - GEPLA.

Esta supervisão é responsável em realizar a manutenção dos equipamentos de informática e outros equipamentos relacionados. Tais equipamentos são: microcomputadores, notebooks, servidores de rede, impressoras, scanners, estabilizadores, servidores de impressão, hubs, roteadores, aparelhos de fax, no-breaks, câmeras fotográficas digitais e zip drives.

3. Web Service e ferramentas utilizadas

Aplicações distribuídas são programas que executam tarefas em diferentes computadores realizando troca de informação entre eles a partir de uma rede. Uma das alternativas para uso deste tipo de tecnologia são os mecanismos RPC (Remote Procedure Call) onde um conjunto de regras e aplicações produzem sistemas que realizam chamadas de procedimentos remotos transparente ao usuário. Alguns destes mecanismos de chamada remota de procedimento são CORBA, RMI, Web Services entre outros.

O CORBA (Common Object Request Broker Architecture) é uma especificação de arquitetura distribuída para ambientes heterogêneos que se comunicam utilizando IDL e ORB. A IDL (Interface, Definition Language, ou Linguagem de Definição de Dados) é uma interface utilizada para detalhar a estrutura dos objetos que serão transportados em um formato neutro de linguagem. A ORB (Object Request Broker) é o serviço responsável de localizar um objeto na rede, comunicar a requisição ao objeto, esperar os resultados e retornar este resultado ao cliente.

O RMI (Remote Method Invocation) é um mecanismo de RPC para programação orientada a objetos desenvolvidos exclusivamente para o Java, onde realiza chamada de métodos remotamente semelhante ao CORBA porém sem utilizar IDL.

O Web Service é uma solução onde um serviço é publicado, localizado e utilizado através de uma rede, como a Internet, onde as comunicações entre as aplicações distribuídas ou serviços interativos ocorrem por mensagens escritas em formato XML, utilizando HTTP na maioria das vezes como protocolo de comunicação.

Os serviços de um servidor Web Service, as chamadas de procedimentos remotos, podem ser acessado por qualquer aplicação escrita em qualquer linguagem e executando em qualquer plataforma. Sua interoperabilidade se dá pelo uso de padrões abertos, tais como o XML, SOAP, UDDI, WSDL e o HTTP.

O XML (Extensible Markup Language) é um documento de texto estruturado utilizado para descrever informações; o SOAP (Simple Object Access Protocol) é um protocolo baseado em XML para comunicação entre as aplicações e serviço; o WSDL (Web Service Definition Language) é um documento XML com as descrições dos serviços de um determinado Web Service; o UDDI (Universal Description, Discovery and Integration) é uma infra-estrutura pública de diretório utilizada para publicar e localizar os serviços disponíveis por um Web Service e descritas no documento WSDL; e o HTTP é o protocolo de transporte mais utilizado.

O ciclo de vida de um Web Service ocorre com [BASIURA,2003]:

1. Criação do Web Service;

2. Divulgação do serviço em um registro UDDI;
3. localização do serviço;
4. obtenção da descrição do serviço através da WSDL;
5. criação do Proxy do cliente;
6. chamada do web service através do SOAP.

O pacote utilizado para o desenvolvimento do Web Service neste trabalho foi o JWSDP 1.0.01 (Java Web Service Developer Pack), pacote desenvolvido pela Sun Microsystems, que reúne um conjunto de aplicações que permitem desenvolver, testar e publicar aplicações distribuídas utilizando conceitos XML. As APIs deste pacote para criação e publicação dos serviços são:

- SAAJ (SOAP with Attachments API for Java) – módulo composto por diversas APIs existentes para manipular envelopes SOAP;
- JAXB (Java Architecture for XML Binding) – facilita o uso do XML compilando um esquema XML em uma ou mais classes Java, ou seja, elimina a necessidade de codificar classes para manipular XML;
- JAXM (Java API for XML Messaging) – fornece o código para criar mensagens SOAP;
- JAXP (Java API for XML Processing) – fornece uma interface padronizada para diferentes tipos de parsers XML, sendo utilizada para interpretar, criar e alterar mensagens;
- JAXR (Java API for XML Registries) – fornece uma API Java padrão para acessar diferentes tipos de registros XML (UDDI);
- JAX-RPC (Java API for XML-based RPC) – é a principal API para desenvolvimento do Web Service, pois é ela que incorpora funcionalidade de modelo RPC baseadas em XML no código. É a API que será utilizada no trabalho.

A API JAX-RPC é o módulo responsável pela criação de Web Service nos moldes de RPC no Java, auxiliando na criação do código de interface. As interfaces são as portas de entrada dos clientes e servidores. O código de interface do lado servidor chama-se “tie” e o código de interface do lado cliente chama-se “stubs”. Estes dois módulos são gerados por ferramentas denominadas wsdeploy (que cria o tie e o arquivo WSDL) e wscompile (que cria o stub de acordo com a especificação no WSDL). Depois dos códigos tie gerados, a aplicação é publicada num Web Container, que neste trabalho foi o servidor TOMCAT 4.1.31.

O Mysql 4.1.7 foi o sistema de gerenciamento de banco de dados relacional utilizado.

4. Manutenção de equipamentos

O processo de manutenção de um equipamento na IOESC começa quando um funcionário da empresa percebe algum problema no equipamento e entra em contato com algum membro da SUATEC através do telefone ou por contato pessoal solicitando atendimento. O técnico do setor vai ao local do equipamento, analisa e toma providências para solucionar o problema. Se ele conseguir solucionar o caso encerra. Entretanto, se o problema persistir, o técnico informa ao supervisor da SUATEC que determinará o destino daquele problema. Pode acontecer basicamente duas coisas: o supervisor aciona outro técnico para resolver o problema, ou realiza um processo de concorrência de menor preço entre empresas convidadas para consertar o equipamento. Depois do equipamento consertado pela assistência técnica, ele retorna ao seu local original.

Os principais problemas observados no processo de manutenção de equipamento são:

- ✍ Informalidade – não é feito nenhum documento formal solicitando o pedido de reparo;
- ✍ Pessoal – qualquer pedido de reparo é realizado pessoalmente;
- ✍ Moroso – pode ocorrer demora no processo em ocasiões de esquecimento;
- ✍ Sujeito à informação distorcida – um recado de algum funcionário pode chegar ao técnico de forma descaracterizada;
- ✍ Volátil – nenhum relatório é produzido no processo de manutenção, sendo que a informação é perdida.

Neste processo de manutenção três cargos estão envolvidos: o supervisor – que dirige e orienta o setor; o técnico e o auxiliar técnico. Todos os cargos são capacitados para executar operações de manutenção, porém os pedidos de assistência externa devem ser feitos somente por intermédio do supervisor que solicitará o reparo para a gerência.

Os problemas mais freqüentes observados na empresa são mostrados na tabela a seguir.

SOFTWARE	HARDWARE
<ul style="list-style-type: none"> ○ Problemas de usabilidade do sistema operacional em microcomputadores. 	<ul style="list-style-type: none"> ○ Problemas de componentes físicos dos microcomputadores, tais como placas mães, placas de rede, placas de vídeo, placas de áudio, monitores de vídeo, teclado, mouse, disco rígido, unidades de discos removíveis entre outros.
<ul style="list-style-type: none"> ○ Problemas de usabilidades de programas de microcomputador tais como Microsoft Office, programas de faturamento, estoque, emuladores, 	<ul style="list-style-type: none"> ○ Problemas de componentes periféricos tais como cabos de rede, cabos elétricos, estabilizadores, no-breaks, impressoras e scanners.

SOFTWARE	HARDWARE
consoles, softwares de impressoras e scanners.	
○ Problemas de acesso ao e-mail, bem como a sua utilização.	○ Problemas nos demais dispositivos de responsabilidade da SUATEC.
○ Problemas com microcomputadores infectados por vírus.	

Tabela de problemas observados

5. Sistema

O objetivo de criar um sistema de gerenciamento de chamados técnicos é dar o primeiro passo para que o processo de manutenção deixe de ser manual e passe a ser mais dinâmico e impessoal. O processo deve parecer-se com o processo convencional, com as mesmas características e regras imitando vários procedimentos que estão consolidados e respeitam o regulamento da instituição.

Basicamente três fatores foram permanentemente considerados no planejamento do sistema. O primeiro foi oferecer uma interface simples e funcional para que o sistema produza uma baixa rejeição inicial por parte dos usuários relacionados com o processo de manutenção. O segundo fator foi o de respeitar os procedimentos convencionais e regulados pelo regimento. Por fim, o terceiro fator prioritário foi tornar o sistema igualmente rápido para realizar um chamado comparado à maneira convencional.

A aplicação é dividida em duas partes, sendo elas o cliente (console) e o servidor. No servidor é possível ter acesso a todas ferramentas que o sistema suporta. No console somente as ferramentas relacionadas com o processo de abertura e tratamento do chamado.

O sistema suporta três tipos de usuários: o Administrador, o Técnico e o Utente (o usuário final). O usuário Utente tem permissão apenas para abrir chamados técnicos e pesquisar outros chamados existentes. Esta restrição ajuda na objetividade e evita o desvio da atenção do usuário para recursos secundários do sistema.

O usuário Técnico tem, além de todas as mesmas prerrogativas do usuário Utente, permissão de cadastrar equipamentos e de alterar alguns valores dos chamados conforme suas permissões.

O usuário Administrador tem acesso irrestrito ao sistema. Ele pode exercer quaisquer funções citadas anteriormente, além de ter permissão de gerenciar as contas de usuários, alterar todos os chamados (daqui para frente intitulados de “tickets”) e controlar o cadastrado dos equipamentos.

É importante evidenciar que somente o Administrador consegue interagir na parte servidora da aplicação, portanto, só ele tem acesso às operações de manutenção de sistema. Na parte console qualquer usuário tem acesso, entretanto as funcionalidades oferecidas são somente aquelas que os usuários Técnico e Utente possuem.

Cada tipo de usuário neste sistema é sugestivo aos cargos existentes hoje na SUATEC. Por exemplo: o usuário Administrador pressupõe-se que seja o Supervisor; o usuário Técnico pressupõe-se que seja algum técnico da SUATEC; e o usuário Utente presume-se que seja os demais funcionários da empresa.

O ticket é o elemento central do sistema. Ele representa cada chamado feito por algum usuário que necessita de assistência técnica. Nele está contida a descrição do problema, o vínculo com o equipamento, o vínculo com as pessoas envolvidas (técnico responsável e o autor do chamado) e outras informações importantes. Também é no ticket que o Técnico descreve os procedimentos efetuados para resolver o problema. Quando acaba um processo de manutenção o ticket é arquivado.

O ciclo de vida do sistema inicia-se quando um usuário percebe algum problema técnico em determinado equipamento e dirige-se a um computador para acessar a aplicação console para relatar os acontecimentos. Este chamado depois de criado é delegado a algum Técnico que fica compromissado em encaminhar uma solução ao problema. Depois de deliberada a ocorrência, o Técnico relata os procedimentos tomados para resolver o problema no Ticket também através de um console do sistema. Se o Técnico não conseguir resolver o problema, ele realiza um processo do sistema chamado “abandono de chamado”. O resultado deste processo é o repasse da responsabilidade do Ticket ao Administrador que assume o chamado deixado pelo Técnico. De posse desse chamado o Administrador decide se chama uma assistência externa ou se ele delega o chamado a outro Técnico. Se ele chamar a assistência técnica, o processo ocorre semelhante ao processo convencional, realizando concorrência entre empresas, sendo que ao receber o equipamento consertado o Administrador tem o dever de relatar os procedimentos tomados pela empresa prestadora do serviço no Ticket.

6. Conclusão

Determinados procedimentos existentes nas empresas necessitam de uma metodologia de trabalho bem aplicada e, muitas vezes, de uma ferramenta de apoio. Durante suas atividades os funcionários podem estabelecer determinadas atitudes inadequadas e, com isto, contrair comportamentos diferentes daqueles desejados pelas pessoas que administram as tarefas. Se bem organizadas e definidas, as funções exercidas pelos funcionários tendem a produzir resultados

mais positivos, corretos e que menos distorcem a realidade. Assim sendo, estabelecendo no ambiente uma ferramenta capaz de controlar as atividades é possível fixar um procedimento padrão e gerar material para exames analíticos de comportamento e identificação de procedimentos inadequados realizados.

Quando um sistema busca preencher esta lacuna, ele enfrenta importantes barreiras, tais como aceitação, adaptação ao meio e boa definição de seus objetivos. Estabelecer uma nova metodologia de trabalho para as pessoas é uma iniciativa que necessita cuidado e instigação por parte dos responsáveis. Deste modo, é preciso certificar-se que a finalidade da ferramenta foi absorvida pelos funcionários e ficar atento às informações provenientes das pessoas envolvidas no processo.

O desenvolvimento de um sistema de gerenciamento de manutenção direcionado exclusivamente às necessidades da IOESC foi importante, considerando-se que naquela empresa existem peculiaridades e atributos intrínsecos àquele realidade. Portanto, mantendo as características do processo aos quais os funcionários estão acostumados, e respeitando as normas do regulamento, o sistema está adequadamente harmonizado naquilo que se espera. O sistema oferece um alto grau de simplicidade no acesso às ferramentas aos usuários, proporcionando baixo impacto e pouca resistência nas pessoas à qual se destina, considerando-se que muitos funcionários não estão familiarizados com programas de computadores em geral.

O sistema é, por conseguinte, uma cópia do processo convencional de abertura de chamado técnico, que oferece vantagens para os gerenciadores e para os funcionários que utilizam o serviço de manutenção da empresa. Entre outros benefícios, pode-se destacar a formação de histórico que proporciona informações sobre os procedimentos realizados, agilidade no processo de manutenção, controle das atividades dos técnicos, tempo gasto para as soluções do problema, entre outras coisas.

Utilizar Web Service proporcionou benefícios importantes, um deles foi conseguir chamadas de procedimentos remotos do sistema atravessando o firewall sem que fosse preciso realizar alterações na configuração, e nas normas de segurança estabelecidas.

Outras contribuições do Web Services são de caráter econômico e tecnológico, visto que sua composição é de padrões abertos e interoperáveis, aceitos mundialmente. Sendo independente de plataforma, possibilita estender futuramente funcionalidades úteis para a empresa, como uma interface de interação com fornecedores de assistência técnica ou até mesmo algum outro órgão de governo interessado em manter informações a respeito do processo de manutenção da IOESC. Com Web Service o sistema está favorável a expansões e migrações de softwares, causando menos impacto e inconveniência aos usuários.

7. Referências Bibliográficas

[BASIURA,2003]BASIURA, Russ et al. **Professional ASP.NET Web Service**. São Paulo: Pearson Education, 2003.

[POTTS,2003]POTTS, Stephen; KOPACK, Mike. **Aprenda em 24 horas Web Services**. Rio de Janeiro: Campus, 2003.

[BECKER,2001]BECKER, Aleksander Knabben. **Web Service e XML: novo paradigma da Programação Distribuída**. Florianópolis, UFSC, 2001.

[HENDRICKS,2002]HENDRICKS, Marck; GALBRAITH, Ben; IRANI, Romin; MILBERNY, James; MODI, Tarak; TOST Andre; TOUSSAINT, Alex; BASHA, S. Jeelani; CABLE Scott. **Profissional Java Web Service**. Rio de Janeiro: Alta Books, 2002.