

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística
Curso de Sistemas de Informação

**DESENVOLVIMENTO E ESTUDO DE UMA FERRAMENTA
PARA GERÊNCIA DE SLA APLICADO NA REDE NACIONAL DE
ENSINO E PESQUISA**

Rodrigo José Bertoldi Pescador
Paulo Alexandre de Oliveira Brandtner

Florianópolis

2012

Rodrigo José Bertoldi Pescador
Paulo Alexandre de Oliveira Brandtner

**DESENVOLVIMENTO E ESTUDO DE UMA FERRAMENTA
PARA GERÊNCIA DE SLA APLICADO NA REDE NACIONAL DE
ENSINO E PESQUISA**

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do grau de Bacharel em Sistemas
de Informação.

Orientador: Prof. Dr. Carlos Becker Westphall

Florianópolis

2012

Rodrigo José Bertoldi Pescador

Paulo Alexandre de Oliveira Brandtner

**DESENVOLVIMENTO E ESTUDO DE UMA FERRAMENTA
PARA GERÊNCIA DE SLA APLICADO NA REDE NACIONAL DE
ENSINO E PESQUISA**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo Curso de Sistemas de Informação.

Florianópolis, 18 de dezembro de 2012.

Banca Examinadora:

Prof. Dr. Carlos Becker Westphall

Orientador

Profa. Dra. Carla Merkle Westphall

Edison Tadeu Lopes Melo

Resumo

Um acordo de nível de serviço é um contrato firmado entre o fornecedor de uma determinada solução e um cliente. Ele basicamente estabelece pontos e termos capazes de serem medidos de forma que os serviços prestados possam ser acompanhados e observados qualitativamente.

A "onda da terceirização" faz deste instrumento um importante aliado na contratação de um serviço, uma vez que o cliente pode cobrar o que foi previamente fixado em contrato e monitorar a qualidade do que está sendo oferecido. Isto faz com que o fornecedor trabalhe de forma correta e alinhada às necessidades do cliente.

O modelo de negócio praticado pela RNP exige a existência de SLA para o cumprimento da qualidade de serviço. Para cada instituição cliente é contratado um ou mais enlaces de dados junto às operadoras para realizar sua interligação ao PoP mais próximo. O PoP então é responsável pela geração de relatórios mensais informando, principalmente, a disponibilidade do serviço prestado.

Este trabalho, em uma primeira etapa, tem como objetivo o estudo de métricas e suas formas de medição para criar acordos de SLA que sejam aplicados ao modelo de negócio da RNP. Na segunda etapa, será desenvolvida uma aplicação WEB para realizar o controle destes acordos anteriormente definidos e permitir a geração de relatórios gerenciais para tomadas de decisão.

Palavras-chave: SLA, SLM, Gerência de Redes, Rede Nacional de Ensino e Pesquisa, Medições de SLA

Lista de Figuras

Figura 1 - Representação do modelo OSI em camadas.....	19
Figura 2 - Comparação entre arquitetura OSI e TCP/IP	24
Figura 3 - Arquitetura geral de uma solução de gerência.....	26
Figura 4 - Solicitação de informação através de SNMP de um gerente a um agente.....	28
Figura 5 - Envio de informação do agente ao gerente (<i>trap</i>).	28
Figura 6 - Grupos da MIB-II	31
Figura 7 - Árvore MIB parcial a partir da raiz. Fonte: Contessa & Polina.....	32
Figura 8 - Estrutura básica de uma requisição AJAX.	45
Figura 9 - Coleta de medições.....	49
Figura 10 – Interface de cadastro de usuários.....	57
Figura 11 - Tela de cadastro de acordos	58
Figura 12 - Tela de cadastro de monitoramentos	59
Figura 13 – Interface de cadastro de manutenções programadas.....	61
Figura 14 - Esquema do banco (parte 1)	62
Figura 15 - Esquema do banco (parte 2)	63
Figura 16 – Exemplo de resultado obtido por um monitoramento com evento de queda.	64
Figura 17 – Exemplo de resultado obtido por um monitoramento com disponibilidade de 100%.	65

Figura 18 - Arquitetura geral de uma solução de gerência (LOPES 2003) ...	276
Figura 19 - Coleta das medições do sistema de controle de SLA	278
Figura 20 - Tela de cadastro de manutenções programadas	281
Figura 21 - Exemplo de resultado obtido por um monitoramento com evento de queda.	282
Figura 22 - Exemplo de resultado obtido por um monitoramento com disponibilidade de 100%.	282

Lista de Tabelas

Tabela 1 - Descrição dos objetos da MIB	32
Tabela 2 - Métricas vs. formas de coleta dos dados.....	54
Tabela 3 - OID's utilizados pelo sistema	55
Tabela 4 - Métricas vs. formas de coleta dos dados.....	280
Tabela 5 - OID's utilizados pelo sistema	280

Lista de Abreviaturas

DoS - Department of Defense

FTP - File Transfer Protocol

HTTP - Hypertext Transfer Protocol

ICMP - Internet Control Message Protocol

ISO - International Organization for Standardization

ITU-T - International Telecommunication Union - Telecommunication
Standardization

MIB - Management Information Base

OSI - Open Systems Interconnection

PoP - Point of Presence

PoP-SC - Ponto de Presença da RNP em Santa Catarina

RNP - Rede Nacional de Ensino e Pesquisa

RTT - Round Trip Time

SLA - Service Level Agreement

SLM - Service Level - Service Level Management

SMTP - Simple Mail Transfer Protocol

SSH - Secure Shell

SNMP - Simple Network Management Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

WEB - World Wide Web

Sumário

Resumo	5
Lista de Tabelas	8
Lista de Abreviaturas	9
1. Introdução.....	12
1.1 Proposta do trabalho	13
1.2 Objetivos	13
1.2.1 Objetivos específicos	13
1.3 Justificativa.....	14
1.4 Áreas de conhecimento.....	15
1.5 Organização do trabalho	16
2. Gerenciamento de redes	17
2.1 OSI.....	18
2.1.1 Modelo de camadas.....	18
2.1.2 TCP/IP	21
2.1.3 SNMP.....	24
2.1.4 Management Information Base (<i>MIB</i>)	30
3. Gerenciamento e acordos de nível de serviço: SLM & SLA	32
3.1 Introdução	33
3.2 SLM.....	33
3.2.1 Por que se interessar por SLM?	35
3.3 SLA.....	37
4. Tecnologias	39
4.1 MySQL	39
4.2 HTML.....	41
4.3 CSS (Cascading Style Sheets).....	42
4.4 PHP.....	42
4.5 AJAX	44
5. Proposta do sistema.....	46
5.1 Ambiente atual no PoP-SC/RNP	47
5.2 Novo ambiente de gerenciamento do SLA	48
5.2.1 Acordos (métricas).....	50
5.2.2 Formas de medição e coleta.....	53
5.3 Principais funcionalidades do sistema.....	55
5.3.1 Cadastro de usuários.....	56
5.3.2 Cadastro de acordos.....	57
5.3.3 Cadastro de monitoramentos.....	58
5.3.4 Registro e acompanhamento de eventos.....	59

5.3.5	Geração de relatórios.....	60
5.3.6	Manutenções programadas	60
5.4	Modelagem do banco de dados	62
5.5	Resultados	63
6.	Considerações finais e trabalhos futuros.....	66
7.	Referências bibliográficas.....	68
8.	Apêndice.....	72

1. Introdução

Hoje em dia é difícil encontrar uma empresa que não esteja ligada à Internet. Para isso, elas contratam planos de empresas prestadoras de serviços de telecomunicações (provedores) que possam fornecer a tecnologia necessária para seu acesso à rede.

Com o aumento da utilização das redes em seus negócios e pelo fato delas, em muitos casos, terem se tornado um item indispensável ao seu funcionamento, as empresas precisam ter garantido um sistema tolerante a falhas, ou que pelo menos estas falhas sejam resolvidas em um curto período de tempo de modo a evitar paralisações que possam causar algum transtorno para a empresa, seja ele qual for.

Diante disto, a grande preocupação das empresas prestadoras de serviço de telecomunicações hoje em dia é a "qualidade percebida pelo cliente". E a estratégia adotada pelos provedores de telecomunicações para atuar com sucesso sobre este tema é a obtenção do acordo de Nível de Serviço (SLA - *Service Level Agreement*) entre clientes e provedores. (PIMENTEL, 2008). Segundo recomendação E.860 da ITU-T, o SLA é o resultado de uma negociação entre duas ou mais partes, no caso o cliente e o provedor do serviço, que tem como objetivo alcançar um entendimento comum em relação a entrega do serviço: seus níveis de qualidade, responsabilidades, prioridades, etc. O principal objetivo do SLA é prover informações do serviço prestado de forma objetiva e quantificável.

1.1 Proposta do trabalho

O trabalho a ser desenvolvido possui duas etapas. Na primeira etapa, o principal objetivo é o estudo de acordos e formas de implementação para realização das medições das métricas do SLA voltadas às necessidades da Rede Nacional de Ensino e Pesquisa. Já na segunda etapa, temos a criação e desenvolvimento de um sistema de gerência e medição de SLA, onde o administrador pode controlar objetos gerenciados, acordos, gerar relatórios e acompanhar chamados abertos junto às prestadoras de serviço.

A ferramenta é baseada em interface *Web* e irá possuir um agente responsável por coletar os dados referentes às medições das métricas aos objetos cadastrados.

1.2 Objetivos

De maneira geral, o objetivo do trabalho é realizar um estudo sobre métricas e formas de se gerenciar um SLA e propor o desenvolvimento de um ambiente para monitoração e gerência de Acordos de Nível de Serviço aplicado às necessidades observadas na RNP.

No desenvolvimento do sistema serão utilizadas somente tecnologias de software livre.

1.2.1 Objetivos específicos

- Estudo e definição das métricas que serão utilizadas nos acordos de SLA voltados à rede;
- Criação de um agente para monitoramento das métricas estabelecidas;
- Desenvolvimento de visualizador para geração de relatórios gerenciais dos acordos em monitoramento;
- Implantação do sistema inicialmente no PoP-SC/RNP e posteriormente em outros PoPs da RNP interessados;
- Prover informações suficientes de modo que seja possível cobrar do prestador de serviço o que foi definido no SLA;
- Permitir o controle, acompanhamento e histórico de chamados abertos junto às prestadoras de serviço;

1.3 Justificativa

A RNP possui diversos Pontos de Presença espalhados pelo Brasil. Estes Pontos de Presença são provedores de serviços, tendo como principal função fornecer conectividade à rede acadêmica e a Internet para as diversas Instituições de Ensino e Pesquisa nas respectivas unidades da federação.

Mensalmente a RNP exige que os PoPs enviem um relatório de SLA dos serviços prestados aos clientes, tendo como principal métrica a ser fornecida a disponibilidade do enlace. Ferramentas como Nagios¹ e Zabbix² (ambas *open source*), voltadas à gerência de rede, não possuem um módulo flexível e dedicado ao gerenciamento de acordos SLA. Basicamente o que fazem é a medição da disponibilidade de um *link*, através de *ping*, por exemplo. Além disso, geram um relatório simples baseado em perdas de pacotes, sem relacionar eventos ocorridos.

Fornecer outras métricas de SLA para os enlaces, e não só a disponibilidade, também é um importante insumo para o gerenciamento da qualidade do serviço prestado.

1.4 Áreas de conhecimento

- Programação orientada à WEB;
- Gerência de redes;
- Gerenciamento de Níveis de Serviço (SLM);

¹ Nagios: Ferramenta para gerência de rede. Disponível em: <http://www.nagios.org>.

² Zabbix: Software voltado à gerência de redes. Disponível em: www.zabbix.com.

- Acordos e métricas de nível de serviço (SLA);
- Funcionamento e configuração de dispositivos de rede (roteadores e switches);
- Ambiente LINUX;

1.5 Organização do trabalho

O trabalho é dividido em seis capítulos:

- Capítulo 1: Introdução ao tema SLA, objetivos e justificativa do trabalho a ser desenvolvido.
- Capítulo 2: Revisão bibliográfica dos protocolos de comunicação de rede e protocolos de gerência.
- Capítulo 3: Detalhamento sobre o tema gerenciamento de acordos de nível de serviço (SLM), acordos de nível de serviços (SLA) e sua importância.
- Capítulo 4: Apresenta as tecnologias utilizadas no desenvolvimento do sistema.
- Capítulo 5: Descreve a situação atual do gerenciamento dos acordos de SLA no PoP-SC e também toda a infraestrutura e detalhamento do novo ambiente desenvolvido, métricas, geração de relatórios.
- Capítulo 6: Aqui serão apresentadas as considerações finais e aprendizados e os trabalhos futuros relacionados ao tema.

2. Gerenciamento de redes

As redes de computadores e seus recursos associados possuem tamanha importância para uma organização que estes componentes praticamente não podem parar. Em outras palavras, isso significa que as falhas e degradação dos serviços devem ser cada vez menores.

STALLINGS (1998 apud LOPES, 2003, p. 17) diz que o objetivo da gerência de redes é monitorar e controlar os elementos da rede (físicos e lógicos), de forma a garantir determinado nível de qualidade do serviço. Para isto, os gerentes e administradores de rede são auxiliados por sistemas de gerenciamento de redes.

Um sistema de gerenciamento de rede pode ser definido como um conjunto de ferramentas de monitoração e controle dos dispositivos da rede, onde existe uma interface com informações sobre o estado da rede.

CLEMM (2007) explica que o gerenciamento de rede refere-se às atividades, métodos, procedimentos e ferramentas pertinentes a operação, administração, manutenção e provisionamento de sistemas de redes. A operação é a responsável por manter a rede funcionando em condições adequadas, de forma a detectar problemas o mais rapidamente possível e de preferência antes do cliente ser atingido. Já a administração deve estar preocupada em manter e controlar os recursos e como eles são distribuídos pela rede de forma a ter o ambiente sob controle. A manutenção está preocupada em realizar reparos e upgrades na infraestrutura, sejam eles corretivos ou preventivos. Por fim, o provisionamento é

encarregado de configurar os recursos da rede para suportarem determinada aplicação.

2.1 OSI

2.1.1 Modelo de camadas

Quando as primeiras redes de computadores foram criadas, somente equipamentos de um mesmo fabricante poderiam se comunicar, ou seja, não havia interação entre os protocolos utilizados. Desta forma, as empresas optavam somente por uma solução, como por exemplo IBM.

No início dos anos 80, a *ISO* criou um grupo voltado às discussões sobre a padronização e interoperabilidade de hardware e protocolos envolvidos na comunicação de dados. Como resultado desta ação, em 1984 foi criado o modelo em camadas chamado OSI que tinha como o objetivo permitir a comunicação e troca de informação entre os equipamentos independentemente de seu fabricante.

Segundo FILIPPETTI (2008) este modelo é dividido em sete camadas subdivididas em duas categorias: superior e inferior (Figura 1).

As camadas superiores (transporte, sessão, apresentação e aplicação) lidam com assuntos relacionados à aplicação e são, geralmente, implementadas em software.

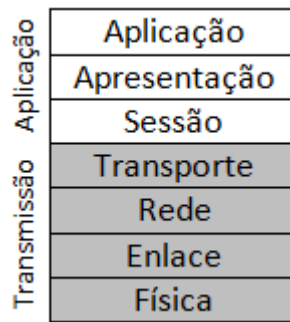


Figura 1 - Representação do modelo OSI em camadas.

Fonte: FILIPPETTI (2008)

De acordo com Grigonis (2000), cada uma das camadas possui as seguintes funções:

- **Camada física:** responsável por ativar, manter e desativar a conexão física para a transferência de bit entre diferentes dispositivos.
- **Camada de enlace:** responsável pelos procedimentos e protocolos para o funcionamento dos canais de comunicação, além de fornecer detecção de erros, correção/sincronização para o nível físico. Neste nível os computadores se comunicam através de endereços MAC das placas de rede.
- **Camada de rede:** trata como os dados são encaminhados a partir do computador de origem para o computador de destino, enviando-o na

direção certa ao longo dos nós intermediários. A camada de rede faz roteamento e encaminhamento de pacotes entre redes.

- **Camada de transporte:** define as regras para troca de informações e gerencia o controle de fluxo e entrega (determinar se todos os pacotes chegaram) e verificação de erros/recuperação entre as redes. É a garantia de que todos os dados foram transferidos de forma íntegra.
- **Camada de sessão:** intermediador entre hardware e software. A camada de transporte lida como um fluxo de dados é dirigido, mas é a camada de sessão que diz quando os dados podem fluir. É responsável pelo estabelecimento, gerenciamento e finalização de sessões entre o transmissor e o receptor.
- **Camada de apresentação:** converte o formato do dado recebido pela camada superior (aplicação) em um formato comum ao protocolo. A camada de apresentação também se preocupa com métodos de criptografia de dados e segurança de dados e algoritmos de compressão.
- **Camada de aplicação:** disponibiliza os recursos para a comunicação entre a máquina destino e o usuário da aplicação. Essa camada contém funções para aplicações de serviços, como transferência de arquivos, acesso remoto a arquivos e terminais virtuais.

2.1.1.1 Áreas Funcionais do Gerenciamento

A *ISO* definiu um modelo de gerenciamento de redes estruturado em 5 áreas. Conhecido por FCAPS, são divididas em:

- **Gerenciamento de desempenho:** Avalia medidas como vazão, latência, disponibilidade afim de realizar ajustes que se façam necessários.
- **Gerenciamento de falhas:** Localiza e isola possíveis falhas na rede. Reconfiguração da rede ou troca de peças defeituosas.
- **Gerenciamento de configuração:** Controle dos recursos de hardware e software. Acompanha as mudanças e altera as configurações de recursos da rede.
- **Gerenciamento de contabilização:** Acompanhamento dos recursos de rede visando a cobrança destes ou regulamentação de seu uso.
- **Gerenciamento de segurança:** Monitoramento e controle de acesso à rede e às informações dos objetos gerenciáveis. Procura garantir que a política de segurança da rede seja cumprida.

2.1.2 TCP/IP

O modelo TCP/IP é originado antes mesmo do conhecimento do modelo *OSI* proposto pela *ISO*, sendo que foi inicialmente criado para atender à demanda de conexão entre diversos projetos do Departamento de Defesa dos Estados Unidos.

A necessidade de uma rede que funcionasse em qualquer situação, mesmo em meio a guerras, e que cruzasse diversas formas de meio físico (satélite, ondas, cabo óptico, etc) foi o que motivou os americanos a desenvolver este modelo que é a base onde a Internet se desenvolveu.

Com a popularização da Internet, os usuários que quisessem se juntar à rede deveriam utilizar um mesmo protocolo para se comunicarem, neste caso, o protocolo TCP/IP. Na prática, hoje a maioria absoluta das redes se apoiam no TCP/IP para seu funcionamento.

Uma curiosidade é que o modelo TCP/IP não foi pensado e criado para ser um modelo de referência, mas tinha como objetivo solucionar um problema específico e importante. Acabou por se tornar um padrão de implementação para todos os fabricantes de dispositivos de rede.

2.1.2.1 Camadas do TCP/IP

Apesar de algumas camadas do TCP/IP possuírem o mesmo nome de camadas do modelo *OSI*, suas características são inerentes a cada um dos protocolos e podem induzir ao erro se não forem observadas cuidadosamente. A Figura 2 ilustra a comparação referenciando os dois modelos.

STALLINGS (2007) coloca o número de camadas do TCP/IP como sendo cinco descrevendo cada uma da seguinte maneira:

- **Física:** responsável pelas especificações das características do meio de transmissão da rede, origem dos sinais, taxa de dados, etc.

- **Acesso à Rede:** camada preocupada em relação a troca de dados entre sistemas finais e a rede a qual estão conectados. A tecnologia usada depende da rede, que pode ser frame relay, ethernet, etc. Uma vez que as funções de acesso a rede estão separadas em uma camada, o próximo nível não precisa se preocupar com as características específicas das redes utilizadas. Esta camada trata basicamente o roteamento de dados entre dois sistemas que estão conectados em uma mesma rede.
- **Internet:** a função desta camada é permitir o roteamento de dados e a comunicação entre dispositivos que estejam fisicamente ligados a redes distintas. O protocolo é implementado não só no sistema final mas também em roteadores.
- **Transporte:** geralmente as aplicações necessitam a troca de informações de uma maneira confiável, ou seja, uma garantia de que dados transmitidos em um ponto A cheguem da mesma forma e ordem em um ponto B. Uma vez que este requisito é independente da aplicação, faz sentido existir uma camada exclusiva para este fim que compartilhe as funcionalidades de controle. o *TCP* é a forma mais confiável para transmissão de informações.

- **Aplicação:** este nível é o suporte necessário para várias aplicações e contém os protocolos de alto nível como SSH, FTP, HTTP, SMTP, etc. Suas funções se assemelham muito as do modelo *ISO*.

OSI	TCP/IP
Aplicação	Aplicação
Apresentação	
Sessão	
Transporte	Transporte
Rede	Internet
Enlace de Dados	Acesso à Rede
Física	Física

Figura 2 - Comparação entre arquitetura OSI e TCP/IP

Fonte: STALLINGS (2007)

2.1.3 SNMP

2.1.3.1 Funcionamento

O protocolo SNMP é um protocolo de gerência definido a nível de aplicação e é utilizado para obter informações de servidores SNMP - agentes espalhados em

uma rede. Os dados são obtidos através de requisições de um gerente a um ou mais agentes utilizando os serviços do protocolo de transporte UDP para enviar e receber suas mensagens através da rede. Dentre as variáveis que podem ser requisitadas estão a MIB II ou MIB privada de cada fabricante.

O gerenciamento da rede através do SNMP permite o acompanhamento simples e fácil do estado, em tempo real, da rede, podendo ser utilizado para gerenciar diferentes tipos de sistemas. (DIAS & JUNIOR).

O protocolo SNMP define um padrão de acesso aos dados, mas não especifica quais são estes dados. Sendo assim, ele permite a criação de novas informações de acordo com a necessidade de cada equipamento ou fabricante. Na verdade, o que o SNMP faz quando solicita uma informação é consultar a MIB do equipamento.

Existem basicamente dois grupos de operação no protocolo SNMP: recuperação e alteração da informação. O primeiro permite a obtenção (leitura) de um dado ou valor de determinado objeto da MIB, o segundo permite a alteração (escrita) de um dado ou valor do objeto.

Hoje, todos os equipamentos de rede possuem suporte ao protocolo SNMP.

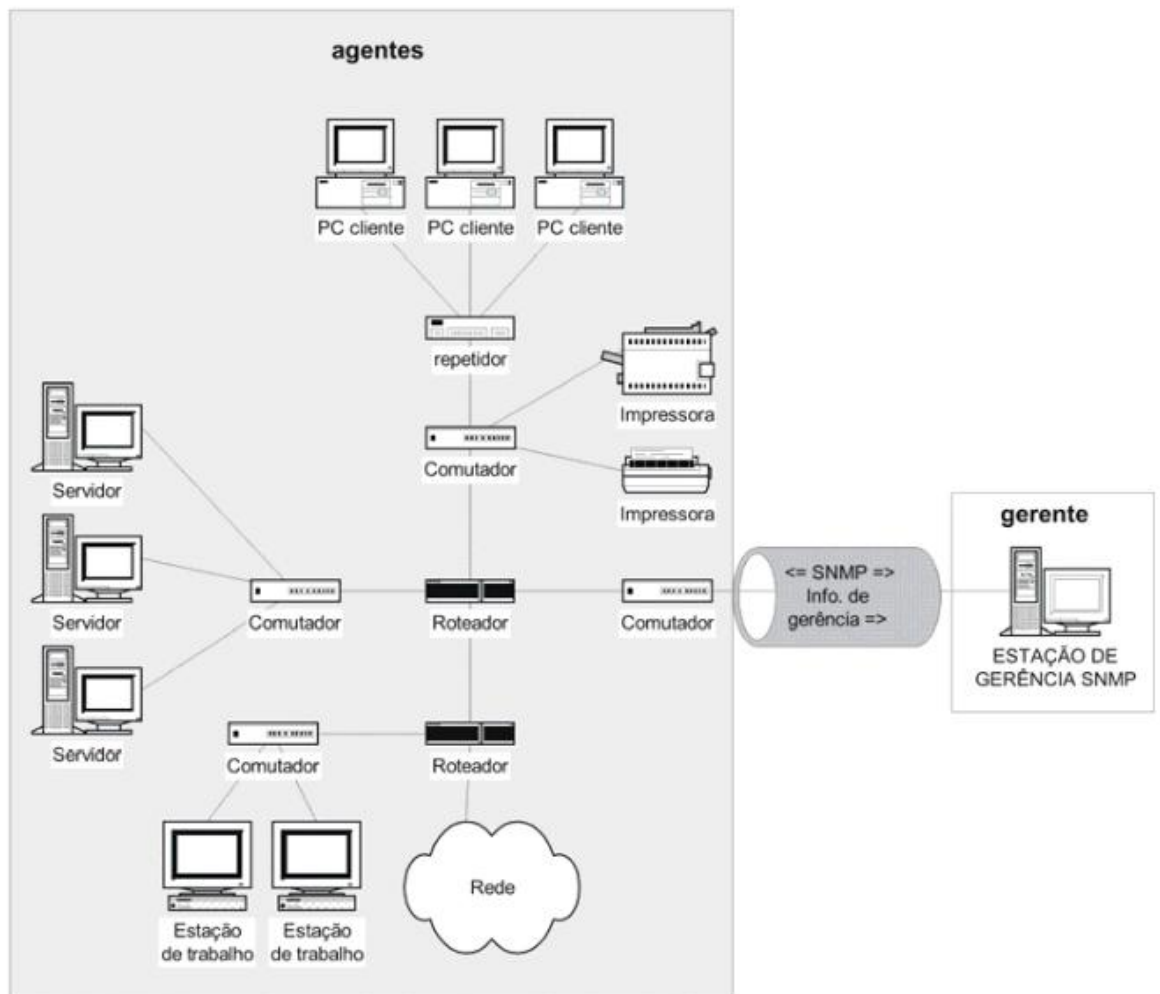


Figura 3 - Arquitetura geral de uma solução de gerência.

Fonte: LOPES (2003)

Agente: é o processo responsável pela resposta das informações quando solicitadas pelo gerente. Mantém a base de informações disponíveis no equipamento (*MIB*). Pode estar localizado em um roteador, computador, switch, etc.

Gerente: software executado em uma estação que realiza as solicitações de informações aos agentes, geração de relatórios e alertas.

2.1.3.2 Operações disponíveis

O protocolo SNMP define um conjunto de 5 comandos para operação, conforme relata DIAS & JUNIOR (2001). Comandos *get* e *get-next* buscam a informação a partir da MIB de um agente. *Get-response* é o comando executado pelo agente retornando informações solicitadas pelo gerente através de *get* e *get-next*. *Set-request* permite a alteração de variáveis em uma MIB. Já a *trap* é utilizada pelo agente para enviar informações sem que o gerente as tenha solicitado.

- **Get:** este comando é utilizado pelo gerente para solicitar informações aos agentes SNMP na rede. Adicionalmente, é possível inserir na requisição um *OID* da MIB para retorno da informação específica.
- **Get-next:** atua da mesma forma que o comando *get*, porém neste caso quando a resposta de uma solicitação anterior é recebida, o gerente faz um novo *get-next* para obter a informação da variável seguinte da MIB. Este processo só finaliza quando é retornado um erro informando que não há mais objetos disponíveis naquela MIB.
- **Get-response:** através deste comando o agente retorna informações ao gerente sobre requisições feitas.

- **Set-request:** permite a alteração de parâmetros e variáveis nos agentes, sendo que um uso bastante comum é a alteração na forma como um dispositivo é configurado.
- **Trap:** o dispositivo gerenciado (agente) envia uma informação para o gerente sem que este a tenha solicitado. Se utilizado, um roteador, por exemplo, pode alertar o gerente sobre uma falha de hardware.

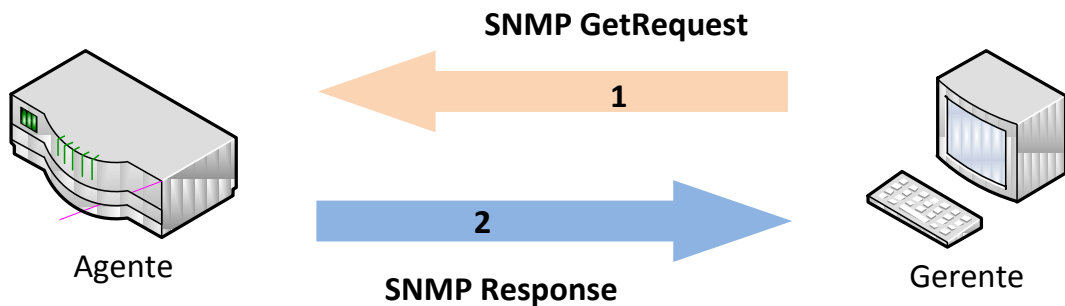


Figura 4 - Solicitação de informação através de SNMP de um gerente a um agente.

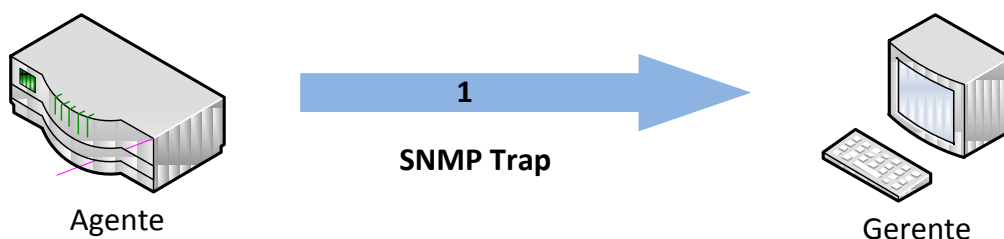


Figura 5 - Envio de informação do agente ao gerente (*trap*).

2.1.3.3 Versões do SNMP

Desde a criação do protocolo SNMP diversas modificações foram realizadas procurando aprimorar a forma de seu funcionamento, com destaque para melhorias no quesito segurança na recuperação das informações. Atualmente na versão 3, é um protocolo estável, confiável e seguro para o gerenciamento de redes.

Em sua primeira implementação, chamada SNMPv1 e criada em 1988, era criticado por seus mecanismos de segurança frágeis e vulneráveis. Para obtenção de informações, bastava informar a senha (comunidade), que era transmitida em texto puro. Além disso, não conseguia manipular muitos dados em uma única requisição.

Já em sua segunda versão, denominada SNMPv2, trouxe algumas melhorias em relação a versão anterior, como por exemplo a adição de um novo tipo de dado chamado 64 bit counter, que permite a leitura de grandes valores - enquanto que na versão 1 está limitado a 32 bit counter. Uma outra característica a ser destacada é a possibilidade de se obter uma maior quantidade de dados em uma única requisição através do novo operador *get bulk*. Questões de segurança, porém, não foram resolvidos e continuam com as mesmas falhas do SNMPv1.

Por fim, o SNMPv3 é considerada a versão mais atual do protocolo. De acordo com CLEMM (2007), foram introduzidos parâmetros que finalmente tornam este protocolo seguro através da autenticação dos solicitantes e encriptação das mensagens. Ainda segundo o autor, quando um agente recebe uma requisição

SNMP, é possível determinar se o gerente em questão é autorizado e se a mensagem não foi adulterada ao longo do caminho.

De uma maneira geral, a versão 3 do SNMP é uma reescrita de sua versão anterior, focando em melhorias de segurança. Até mesmo as operações são idênticas em ambas as versões.

2.1.4 Management Information Base (*MIB*)

A *MIB* é a base de informações de gerenciamento. O agente é capaz de responder ao gerente consultas SNMP sobre o conjunto de informações contidas em sua *MIB*.

Constituída por uma estrutura em árvore (Figura 7Figura 7 - Árvore MIB parcial a partir da raiz.) contendo as variáveis de gerência de um determinado equipamento, a *MIB* define para cada variável um identificador único denominado *OID*, formado por um inteiro não negativo. (CONTESSA & POLINA).

Segundo DIAS & JUNIOR (2001), na MIB-II os objetos estão divididos em 10 grupos (Figura 6Figura 6 - Grupos da MIB-II). A MIB-II define as informações padrões para gerenciamento de redes TCP/IP.

Grupo	Informação
system (1)	informações básicas do sistema
interfaces (2)	interfaces de rede
at (3)	tradução de endereços
ip (4)	protocolo ip
icmp (5)	protocolo icmp
tcp (6)	protocolo tcp
udp (7)	protocolo udp
egp (8)	protocolo egp
transmission (10)	meios de transmissão
snmp (11)	protocolo snmp

Figura 6 - Grupos da MIB-II

Para cada objeto existem atributos como Object-Type, Sintax, Access, Status e Description. A Tabela 1 **Tabela 1 - Descrição dos objetos da MIB** apresenta a utilização de cada um destes atributos.

Um módulo MIB patenteado é elaborado pelo fabricante para gerenciar elementos em particular considerados importantes, mas que não fazem parte dos padrões descritos pelo protocolo.

Atributo	Descrição
OBJECT-TYPE	Descreve o nome do objeto. É único e permite a coleta das informações por nome.
SYNTAX	Define o tipo de valor de determinado objeto, entre eles: INTEGER, GAUGE, COUNTER, TIMESTAMP, OCTET STRING, OBJECT IDENTIFIER, NULL, DisplayString
ACCESS	Define a forma de acesso ao dado, podendo ser do tipo Read-only, Read-write, Write-only, Not-accessible
STATUS	A MIB padrão define um conjunto de objetos a serem implementados e que devem estar presentes. Este campo indica o estado de implementação do objeto. Mandatory : Este objeto deve ser implementado no agente. Optional : O objeto pode ser implementado no agente. Obsolete : Este objeto não é mais implementado no agente, está em desuso.

DESCRIPTION	Contém informações em formato de texto descrevendo o uso do objeto.
-------------	---

Tabela 1 - Descrição dos objetos da MIB

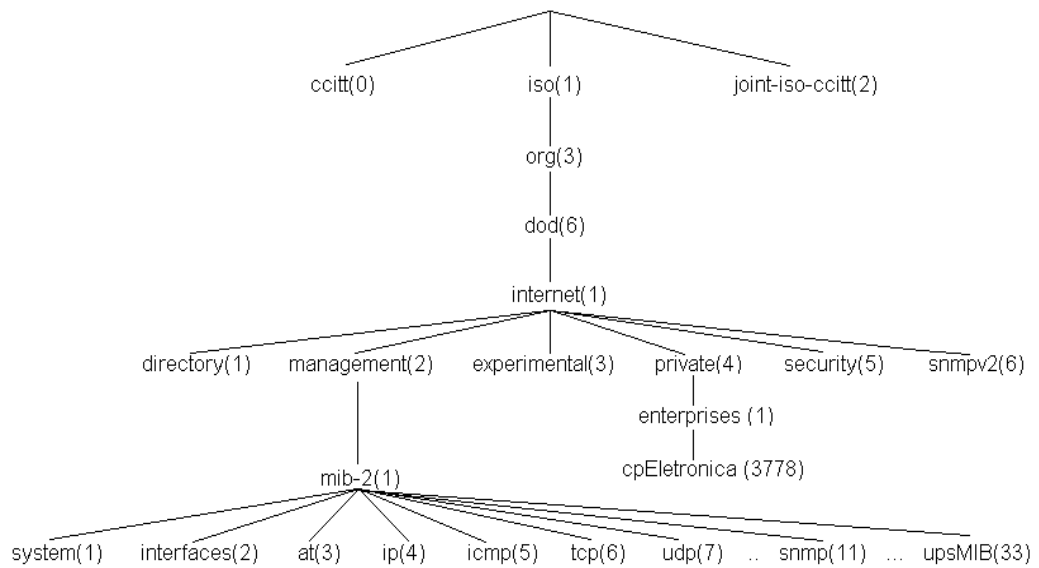


Figura 7 - Árvore MIB parcial a partir da raiz. Fonte: Contessa & Polina

3. Gerenciamento e acordos de nível de serviço: SLM & SLA

Neste capítulo são apresentados e descritos os Acordos de Nível de Serviços - SLA, e o Gerenciamento destes Níveis de Serviços - SLM.

3.1 Introdução

A crescente terceirização no mercado de telecomunicações vem alterando a relação entre seus compradores e fornecedores de serviços. Um dos pontos importantes nesta mudança é a aplicação de SLAs em contratos. Desta forma, as empresas possuem garantias, que devem ser cumpridas por seus fornecedores, de níveis de qualidade e prazos.

Segundo a recomendação E.860 da ITU-T (2002), o SLA é um acordo estabelecido entre o provedor e seu cliente para definir formalmente o nível de qualidade que deve ser atingido na prestação de um serviço. O objetivo é tornar a medição da qualidade objetiva e quantificável, introduzindo meios para mensurar a expectativa do cliente e, portanto, para gerir a qualidade percebida por ele.

A gestão da qualidade percebida pelo cliente só é possível a partir do uso adequado de ferramentas de TI pelos provedores de telecomunicações.

A aplicação do SLA ainda precisa se desenvolver, uma vez que a maioria das companhias que adotam algum tipo de SLA o fazem de forma reativa, apresentando métricas somente de disponibilidade e desempenho. Isto pode ser explicado pela falta de ferramentas automatizadas que permitam fazer uma gestão efetiva dos acordos de nível de serviço.

3.2 SLM

O SLM (*Service Level Management*) é o conjunto de processos e procedimentos que visam garantir o cumprimento dos SLAs. É responsável pelo planejamento e coordenação dos acordos. Além disso, deve garantir a qualidade requerida para os serviços através do acompanhamento dos relatórios SLA e cruzamento dos parâmetros estabelecidos em contrato.

O instrumento de controle do SLM é o Acordo de Nível de Serviço (SLA), um contrato celebrado entre empresa ou setor de TI e seus clientes, que define os parâmetros de capacidade do sistema, o desempenho da rede, tempo de resposta global necessário para atender aos objetivos empresariais, que também especifica um processo para avaliar e informar à qualidade do serviço prestado pela TI e descreve a compensação devida ao cliente, se ela não cumpriu com o prometido. (LEÃO, 2002).

A partir da detecção de uma violação, descobre-se qual foi o elemento que causou o problema, como uma linha de comunicação, uma aplicação ou um banco de dados muito carregado. A análise fim-a-fim leva a um isolamento rápido das causas dos rompimentos de SLA para que o ambiente possa ser avaliado e dimensionado para atender as demandas de seus usuários em relação aos serviços oferecidos. (PEREIRA, 2004).

Esta solução deve monitorar as informações que passam sobre a infraestrutura de TI gerando histórico sobre elas, de modo a permitir a atuação em tempo real em caso de anormalidades. Assim, o SLA pode ser cumprido de forma eficiente.

O gerenciamento de nível de serviço é uma questão de sobrevivência. Segundo STURM & MORRIS & JANDER (2001), é raro encontrar uma empresa bem sucedida que venda seus produtos na forma do "pegar ou largar". Em vez disso, as empresas se esforçam e investem para definir claramente as capacidades e limitações do produto que estão vendendo.

Em suma, o SLM serve principalmente para assegurar o cumprimento do SLA de forma a prevenir e reduzir os impactos negativos na qualidade dos serviços, manter e melhorar a qualidade dos serviços contratados/oferecidos e avaliar as mudanças de estratégias.

3.2.1 Por que se interessar por SLM?

Para STURM & MORRIS & JANDER (2001), existem seis motivos que fundamentam a necessidade de implementação de gerenciamento de nível de serviço em uma organização.

- **Satisfação do cliente:** é importante um diálogo entre as partes, assim a empresa poderá entender as necessidades reais e exigências do cliente. Uma vez estipulado um nível de serviço aceitável, é possível avaliar o desempenho do que é oferecido.
- **Administração das expectativas:** é normal que o usuário tenha, com o passar do tempo, expectativas de níveis de qualidade maiores com o

serviços prestados. Uma vez que o SLA é documentado, este funciona como um mecanismo de retenção e os compromissos a serem cumpridos estão contidos no documento. É possível alterar ou renegociar os acordos SLA para se adequarem com as expectativas crescentes dos usuários, porém é provável que deverá haver aporte de recursos financeiros.

- **Controle de recursos:** através do monitoramento das métricas é possível se antecipar a qualquer necessidade de nova capacidade de recursos ou tomar ações para resolver degradações nos serviços oferecidos.
- **Marketing dos serviços:** ao assegurar níveis contínuos de qualidade e tempo de resposta adequados ao SLA, a organização passa a ser conhecida como boa prestadora de serviços.
- **Gestão dos custos:** através do gerenciamento do nível de serviço é possível alocar os recursos de forma mais adequada. Por exemplo, se não existem objetivos definidos e estipulados para determinado serviço, possivelmente haverá alocação excessiva de recursos de pessoal, equipamentos, capacidade para suprir uma demanda que, na maior parte do tempo, pode estar ociosa.

- **Estratégia defensiva:** a aplicação do SLM traz uma segurança à empresa prestadora do serviço. Uma vez que existe claramente a documentação dos objetivos, não há espaço para dúvidas em relação ao atendimento ou não das definições propostas.

3.3 SLA

Quando uma organização contrata serviços de um provedor, são necessárias algumas garantias para a empresa que está contratando o serviço, e também para a prestadora do serviço. Esta garantia é fornecida através de um contrato chamado de Contrato de Nivel de Serviço - SLA (*Service Level Agreement*). O SLA é o ponto de referência para o controle da qualidade a ser fornecida, ou recebida por uma empresa (SILVA, 2005).

Ele é descrito como uma declaração de expectativas e obrigações que existem no relacionamento de negócio entre duas organizações e seu consumidor. Esta declaração, muitas vezes chamada de contrato, especifica os níveis de qualidade de serviço que o fornecedor se compromete em disponibilizar, além de cláusulas legais e quais as consequências para cada parte se houver descumprimento de deveres. (RIGHI *et al.*, 2006).

Características como confiabilidade, disponibilidade, velocidade e tempo de restauração de falhas também são geralmente abordadas no contexto de um SLA. (PEREIRA, 2004).

Esse tipo de acordo foi inicialmente desenvolvido para o fornecimento de serviços internos das empresas, sendo que os pioneiros foram os serviços de informática. A seguir, foram utilizados para a regulamentação da conectividade de rede entre grandes provedores do mesmo nível e, posteriormente, nos serviços externos, onde passaram a ser tratados por meio de contratos, com as especificações de níveis de serviços requeridas pelo cliente. (FRANKE, 2010).

Segundo RIGHI, *et al.* (2006), tornaram-se populares devido principalmente à terceirização da infra-estrutura de comunicação pelas organizações.

Finalmente, um SLA oferece uma linguagem comum para comunicação entre duas comunidades diferentes: os prestadores de serviço e os clientes. A documentação de um entendimento mútuo obtido durante o processo de negociação do acordo de nível de serviço deve oferecer clareza aos negócios acordados. (LEÃO, 2002).

Para monitorar o cumprimento desse negócio é necessário que as informações sobre um determinado serviço sejam coletadas e tratadas, gerando-se relatórios dos níveis de serviços, os quais possuem valores reais dos parâmetros em um determinado período (hora, dia, mês, ano, etc.). (PEREIRA, 2004).

Segundo Godbolt (2002), os SLAs devem:

- Descrever a amplitude e tipo de serviço fornecido.
- Definir os recursos disponíveis para suporte.

- Clarificar a compreensão e o acordo de todos os partidos na expectativa comum.
- Fornecer expectativas realistas em relação aos serviços.
- Ser simples e despojados de linguagem complexa.
- Focar-se essencialmente em necessidades do negócio.
- Concentrar-se em assuntos importantes.
- Encorajar parcerias para que os clientes colaborem com os fornecedores de modo a melhorar os serviços dentro da comunidade.
- Estabelecer um fórum de discussão e negociação.

4. Tecnologias

Para desenvolver a ferramenta, utilizamos algumas tecnologias associadas à programação, que serão apresentadas neste capítulo.

4.1 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional (*relational database management system - RDBMS*) poderoso e muito rápido. Um

banco de dados permite armazenar, pesquisar, classificar e recuperar dados de forma eficiente. (Welling & Thomson, 2005).

O Servidor de MySQL controla o acesso aos dados para assegurar que vários usuários possam trabalhar com os dados ao mesmo tempo, fornecer acesso rápido aos dados e assegurar que somente usuários autorizados obtenham acesso. Portanto, o MySQL é um servidor multiusuário e multiencadeado (ou *multithreaded*). (Welling & Thomson, 2005).

Ele é uma alternativa atrativa porque, mesmo possuindo uma tecnologia complexa de banco de dados, seu custo é bastante baixo. Tem como destaque suas características de velocidade, escalabilidade e confiabilidade, o que vem fazendo com que ele seja adotado por departamentos de TI (Tecnologia da Informação), desenvolvedores web e vendedores de pacotes de softwares. (Niederauer, 2008).

Segundo Niederauer (2008), a seguir são listadas algumas vantagens do MySQL:

- Número ilimitado de utilização por usuários simultâneos;
- Capacidade de manipulação de tabelas com mais de 50.000.000 de registros;
- Fácil e eficiente controle de privilégios de usuários;
- Alta velocidade de execução de comandos;

Uma das características mais importantes destacadas está no fato dele ter seu código aberto, sobre a licença GNU General Public Licence (GPL). Este fato faz com que as pessoas possam modificar livremente o MySQL conforme suas

necessidades, e melhorar o produto de domínio público através de contribuições de experiências. (PACHEV, 2003).

Algumas das utilizações em ambientes corporativos incluem: web sites dinâmicos, log de uso, Data Warehousing, bases integradas em aplicativos e em dispositivos embutidos. (PACHEV, 2003).

4.2 HTML

HTML é uma linguagem de marcação (estruturação) de *hyperlinks* e *layouts* de documentos. Ela define a sintaxe e posição de instruções especiais que não são exibidas ao usuário, mas dizem como o conteúdo do documento deve ser mostrado, incluindo, texto, imagens e outras mídias suportadas. (KENNEDY & MUSCIANO, 2002).

A linguagem também determina como tornar um documento interativo, através de *links* especiais, que conectam outras partes do documento, outros documentos ou recursos da Internet. (KENNEDY & MUSCIANO, 2002).

Quem especifica as regras do HTML é o W3C (*World Wide Web Consortium*), um consórcio internacional composto por representantes de companhias de tecnologias. Ele tenta estabelecer padrões entre as companhias produtoras de navegadores *Web* através de comitês de membros interessados na tecnologia. Estabelece também outros padrões *Web*, como por exemplo, as folhas de estilo (CSS, *Cascading Style Sheets*). (PFAFFENBERGER, 2004).

4.3 CSS (Cascading Style Sheets)

É uma linguagem de estilo desenvolvida pela W3C em 1996, com o propósito de simplificar e melhorar o *design* das páginas *Web*. (MANSFIELD, 2005).

Pode ser utilizada para controlar os aspectos básicos do estilo de um documento escrito em linguagem de marcação, tais como fontes e cores. Entre seus principais fatores, podemos citar as suas propriedades, que determinam como um elemento deve ser mostrado no navegador. (DUCKETT, 2005).

O CSS provê métodos para controlar como os documentos serão apresentados em outras mídias além dos navegadores tradicionais na tela, tais como impressoras ou dispositivos móveis. Além disso, possui regras para especificar a apresentação não-visual de documentos, por exemplo, como o texto deve soar quando lido por um assistente de leitura de documentos. (ROBBINS, 2007).

Ele também aumenta a eficiência no carregamento das páginas *Web* permitindo separar o conteúdo que determina os estilos da linguagem de marcação sem ter que reescrever todas as propriedades novamente em cada um dos arquivos HTML, podendo determinar como cabeçalhos, parágrafos e textos padrões devem ser posicionados, qual seu tamanho e cor, por exemplo, em apenas uma parte do código. (MANSFIELD, 2005).

4.4 PHP

O PHP é uma linguagem de criação de scripts do lado do servidor que foi projetada especificamente para a Web. (WELLING & THOMSON, 2005).

Segundo Niederauer (2008), o PHP é uma das linguagens de programação mais utilizadas na web para a criação de páginas dinâmicas. Suas principais características são:

- Gratuito e com código aberto: o arquivo de instalação pode ser obtido gratuitamente no site <http://www.php.net>. Além disso, o PHP é um software com código-fonte aberto.
- Embutido no HTML: o HTML e o PHP podem ser misturados. Você pode começar a escrever em PHP, de repente escrever um trecho em HTML, depois voltar para o PHP e assim por diante.
- Baseado no servidor: quando você acessa uma página PHP através de seu navegador, todo o código PHP é executado no servidor e somente o resultado final é exibido para o usuário. Portanto, o navegador exibe a página já processada, sem consumir recursos de seu computador.
- Bancos de dados: diversos bancos de dados são suportados pelo PHP, ou seja, o PHP possui código que executa funções de cada um. Entre eles, temos MySQL, PostgreSQL, SQLite, InterBase,

Oracle, SQL Server, entre outros que oferecem suporte à linguagem SQL (*Structured Query Language*).

A linguagem é muito flexível, não é limitada apenas à saída HTML. Qualquer tipo de documento pode ser gerado, PHP tem suporte integrado para trabalhar com arquivos PDF, imagens GIF, JPG e PNG, e arquivos Flash. (LERDORF *et al.*, 2006).

Ele roda em todos os principais sistemas operacionais, desde distribuições de Unix, incluindo Linux, FreeBSD e Solaris, até Windows e MAC OS X. Pode ser usado com os servidores web mais importantes, incluindo Apache e Microsoft IIS. (LERDORF *et al.*, 2006).

Alguns dos principais concorrentes do PHP são Perl, Microsoft ASP.NET, JavaServer Pages (JSP) e ColdFusion. Em comparação a esses produtos, o PHP tem muitas vantagens, incluindo, alto desempenho. interfaces para muitos sistemas diferentes de banco de dados, bibliotecas integradas para muitas tarefas comuns da web, facilidade de aprender e utilizar, ótimo suporte orientado a objetos, portabilidade, disponibilidade de código-fonte, disponibilidade de suporte. (Welling & Thomson, 2005).

4.5 AJAX

AJAX é um acrônimo para Javascript Assíncrono e XML (Asynchronous JavaScript and XML) e essencialmente oferece uma técnica para que o código

javascript faça chamadas em *background* para o servidor e recupere informações quando necessário. (DARIE *et al.*, 2006).

Basicamente, o que o AJAX faz é usar objetos do tipo *XMLHttpRequest* baseados em Javascript para disparar requisições assíncronas (sem a necessidade de recarregar a página). Fazendo uso dessa tecnologia, as aplicações *Web* podem receber e enviar informações ao servidor, esperar que ele processe-as e então alterar dinamicamente a página em questão, sem que o usuário tenha que sair dela ou perder o foco no que estava fazendo. O servidor pode retornar qualquer tipo de dado ou informação que o *script* requisitar. (BABIN, 2007).

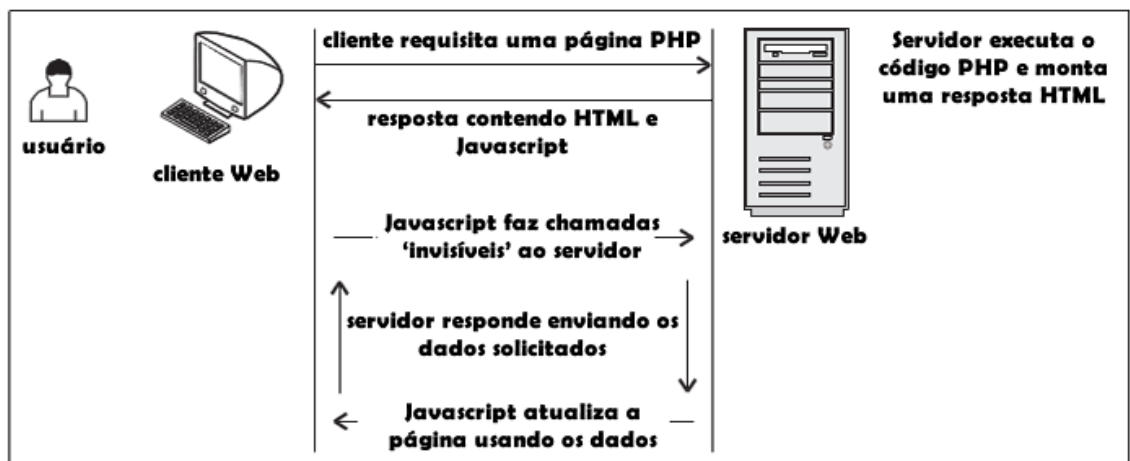


Figura 8 - Estrutura básica de uma requisição AJAX.

Segundo ZAKAS (2006), são identificados alguns princípios chaves das aplicações AJAX, entre eles:

- **Tráfego minimizado:** Aplicações AJAX enviam e requisitam o mínimo de informações, diminuindo o tráfego entre cliente e servidor, contribuindo para a objetividade do código.
- **Interatividade:** Introduz diferentes tipos de modelos de interação entre o usuário e a página, ao contrário do 'clique e espere' das páginas tradicionais.
- **Rapidez:** Toda a comunicação após o carregamento da página inicial deve ser gerenciado pelo AJAX, assim evitando a atualização dela, que prejudicando a experiência do usuário.
- **Usabilidade:** Desenvolvida com o intuito de facilitar a utilização pelo usuário e melhorar a experiência dele com a navegação.

Resumindo, o AJAX é uma forma de balancear as funcionalidades do cliente e do servidor, permitindo que os dois se comuniquem em segundo plano enquanto o usuário executa uma ação. (DARIE *et al*, 2006).

5. Proposta do sistema

Neste capítulo apresentaremos o ambiente para gerência de SLA que foi proposto e desenvolvido de acordo com os estudos realizados. Para uma melhor

organização, dividiremos esta seção em três partes. Na primeira abordaremos o ambiente de medições de SLA atualmente em uso no PoP-SC, suas características e limitações. Já na segunda parte, trataremos sobre a nova ferramenta de gerência SLA, o novo ambiente proposto, novas funcionalidades, formas de medição e métricas. Na terceira parte serão apresentadas algumas interações do sistema e a forma como ele foi estruturado.

5.1 Ambiente atual no PoP-SC/RNP

O cenário atual e a forma como as medições de SLA são realizadas no PoP-SC deixam um pouco a desejar quanto a sua confiabilidade e formas de captação das medições. O software, que foi desenvolvido em meados de 2004 como um trabalho de conclusão de curso, atualmente faz uso somente de probes ICMP (ping) para verificação das métricas, o bloqueio de pacotes ICMP ou então um nível de processamento elevado no dispositivo alvo poderiam influenciar nos resultados obtidos, uma vez que poderia haver descarte destes pacotes. Além disso, o uso exclusivo de ping como forma de medição limita a coleta de outras métricas como utilização de banda, erros de interface, etc. Também não é possível aferir as métricas do SLA através do estado das portas físicas dos equipamentos alvo, por exemplo, pois para isto seria necessário a utilização do protocolo SNMP.

Um ponto importante em uma ferramenta de SLA é a capacidade de gerar um relatório detalhado sobre os problemas encontrados e detectados ao longo de um

período de um acordo. O relatório gerado pela ferramenta atual indica os períodos e o tempo total de uma indisponibilidade, por exemplo, porém não permite a correlação destes eventos registrados pelo sistema com informações obtidas pelos operadores da rede. Em um típico caso onde o operador tem ciência da falta de energia elétrica em um equipamento, ele poderia adicionar um comentário na queda detectada pelo sistema e, opcionalmente, associá-la a uma manutenção emergencial para que esta parada seja descontada do cálculo total do SLA.

Os dados obtidos pela ferramenta de SLA utilizada atualmente suprem a necessidade mínimas de informações, porém determinados aspectos podem ser melhorados. Através da experiência da equipe técnica do PoP-SC, foi possível identificar lacunas e propor melhorias para o sistema de medições conforme veremos no próximo capítulo.

5.2 Novo ambiente de gerenciamento do SLA

Como falado na seção anterior, a ferramenta desenvolvida busca resolver e tratar situações inexistentes ou melhorar módulos presentes no atual sistema de medições, principalmente quanto a forma de obtenção das informações de medição e geração do relatório dos acordos.

A estrutura geral do funcionamento da ferramenta desenvolvida pode ser observada na Figura 9.

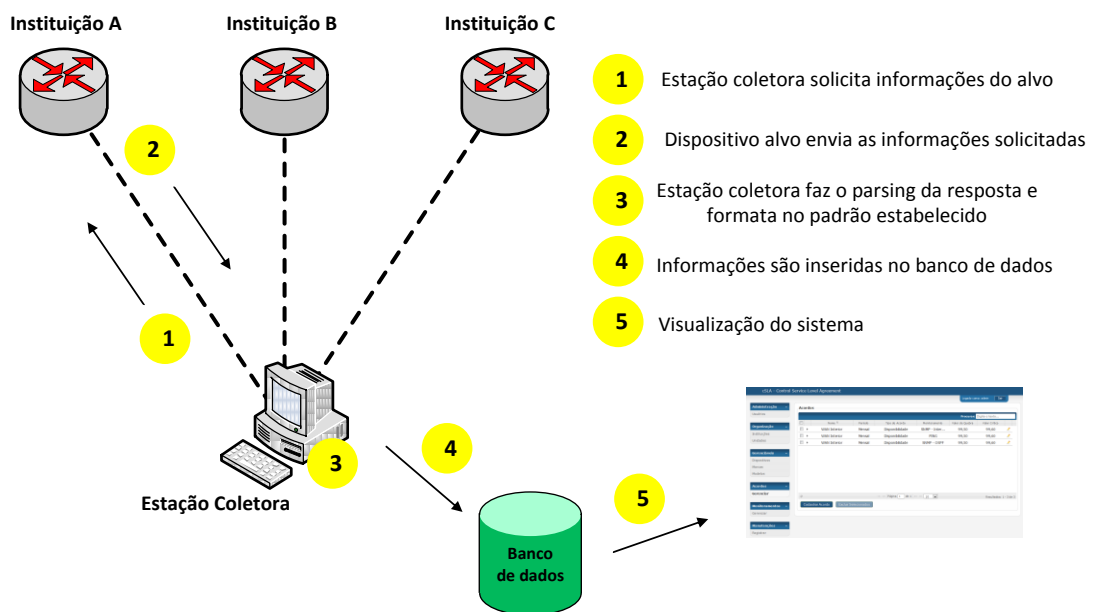


Figura 9 - Coleta de medições

Nota-se uma máquina gerente centralizada responsável pela coleta e armazenamento de todas as informações. A obtenção e tratamento dos dados de medições passam por ela, sendo assim o núcleo de tudo.

O funcionamento do sistema se dá basicamente da seguinte forma:

1. A estação gerente dispara a coleta das informações contra um determinado alvo;
2. O dispositivo alvo responde, ou não, a esta requisição com as informações solicitadas;
3. A informação recebida é tratada e é feito o *parsing* da mensagem para se adequar ao modelo de dados do sistema;
4. Uma vez que os dados estão padronizados, eles são inseridos no banco de dados para posterior recuperação;

Estes dados são obtidos, via de regra, a cada um minuto por um script desenvolvido em PHP que percorre todos os dispositivos alvo cadastrados nos acordos requisitando seus estados e demais informações necessárias para cálculo das métricas.

Um ponto bastante interessante a ser destacado é a forma como o sistema dispara a coleta dos dados junto aos equipamentos. Tudo é feito de forma paralela e independente, evitando, assim, a interferência de um processo em outro. Caso qualquer processo falhe ou trave, os demais não serão afetados e o programa continua a ser executando seu ciclo normal. A linha abaixo mostra a configuração do crontab do sistema Linux para chamado do script de monitoramento.

```
*/* * * * * /var/www/application/scripts/monitoring.sh > /dev/null 2>&1
```

Um segundo fator de destaque é que o software suporta o protocolo IPv6 tanto na máquina coletora quanto para a configuração dos dispositivos alvo para realização das medições do SLA, ou seja, é possível configurar para que os alvos sejam alcançados através de IPv4 e IPv6.

5.2.1 Acordos (métricas)

Em um SLA voltado a rede de computadores é possível estabelecer diversas métricas para verificação da qualidade do serviço prestado. O dia a dia na operação e administração de redes nos mostra alguns pontos importantes que devem ser observados prioritariamente neste processo. Desta forma, as métricas implementadas para utilização no ambiente são detalhadas abaixo.

- **Disponibilidade**

Este indicador descreve o percentual de tempo que para um dado período de um acordo, o enlace, incluindo o equipamento alvo, ficou disponível e em operação, podendo ser calculado através da fórmula abaixo.

$$ID = [(Pa - Si + Tm) / Pa] * 100$$

Onde:

ID = índice de disponibilidade do acordo (em %).

Pa = período do acordo (tipicamente um mês) em minutos

Si = total de inoperância do serviço durante o período do acordo em minutos

Tm = intervalo de tempo onde existe intersecção entre quedas detectadas pelo sistema e um período de manutenção programada

- **Perda de pacotes**

Esta métrica indica a relação entre pacotes enviados e recebidos para um determinado dispositivo alvo e é expressada em porcentagem.

$$\frac{PP = Pe - Pr}{Pe}$$

Onde:

PP = perda de pacotes

Pe = pacotes enviados

Pr = pacotes recebidos

- **Latência**

A latência, também chamada de *RTT (round trip time)*, é o somatório do tempo de ida e volta de um pacote de um ponto A para um ponto B. Caso um ponto A não envie um pacote ou o ponto B não devolva uma resposta, o valor da medição é considerado desconhecido.

- **Jitter (médio)**

O *jitter* é a variação do atraso da entrega de pacotes sucessivos na rede entre um ponto A e um ponto B. Para medir o jitter, basta obter a soma da diferença de tempo entre pacotes seguidos em uma rede e dividir pelo total de amostras - 1.

Por exemplo, em uma medição com 5 amostras são coletados os seguintes tempos: 12, 15, 20, 11, 10, respectivamente. Tirando a diferença de tempo entre os pacotes temos:

$$12 \text{ a } 15 = 3 \quad | \quad 15 \text{ a } 20 = 5 \quad | \quad 20 \text{ a } 11 = 9 \quad | \quad 11 \text{ a } 10 = 1$$

A soma entre as diferenças das medições é 18. Então o *jitter* é $18 / 4$, ou 4,5 ms.

- **Banda utilizada:**

Representa a quantidade de dados passantes por um enlace em um determinado tempo. Caso um valor definido seja atingido e se mantenha por um certo período, existe a quebra do acordo. A unidade de medida típica é bits.

- **Erros de interface:**

Representa o número de pacotes recebidos com erros em uma interface e que não poderão ser repassados às camadas superiores.

5.2.2 Formas de medição e coleta

Agora que as métricas disponíveis para utilização no sistema foram apresentadas, é importante saber de que forma elas podem ser medidas. Para a obtenção dos dados, seja qual for a métrica, utilizamos basicamente a ferramenta *fping* e o protocolo SNMP. DZUBIN (2007), mantenedor atual do aplicativo, explica

que o *fping* é um programa que utiliza Internet Control Message Protocol para determinar se um dispositivo alvo está disponível, trazendo algumas facilidades quando utilizado com scripts, pois sua saída é facilmente parseada. As funcionalidade conhecidas na implementação tradicional no *ping* estão disponíveis também no *fping*.

A

Tabela 2 relaciona a métrica e suas possíveis formas de coleta dos dados.

Métrica	Formas de coleta dos dados
Disponibilidade	ping, snmp interface, snmp ospf
Perda de pacotes	ICMP
Latência	ICMP
Jitter	ICMP
Banda utilizada	snmp interface
Erros de interface	snmp interface

Tabela 2 - Métricas vs. formas de coleta dos dados

Para a métrica de disponibilidade, conforme a Tabela 2 mostra, é possível realizar as consultas dos dados de três maneiras diferentes: ping, snmp interface e snmp ospf. Já para métricas de perda de pacotes, latência e jitter está disponível somente a coleta através de ICMP. Por último, para coleta de banda utilizada e erros de interface somente o SNMP está disponível.

No caso das métricas que utilizam SNMP, os OID's necessários estão relacionados na Tabela 3.

SNMP Interface (ifOperStatus)	1.3.6.1.2.1.2.2.1.8
SNMP OSPF (ospfNbrState)	1.3.6.1.2.1.14.10.1.6
Banda utilizada (ifHCInOctets)	1.3.6.1.2.1.31.1.1.1.6
Banda utilizada (ifHCOctets)	1.3.6.1.2.1.31.1.1.1.10
Erros de interface (ifInErrors)	1.3.6.1.2.1.2.2.1.14
Erros de interface (ifOutErrors)	1.3.6.1.2.1.2.2.1.20

Tabela 3 - OID's utilizados pelo sistema

Em um primeiro momento existirá somente uma máquina coletora centralizada (Figura 9), porém o sistema está projetado para permitir a coleta dos dados de forma distribuída via web service. Isto permitirá verificar o SLA de diferentes pontos em redes distintas.

5.3 Principais funcionalidades do sistema

A seguir serão apresentadas algumas das principais funcionalidades do sistema, entre elas o cadastro de usuários, registro e acompanhamento de eventos, manutenções programadas e geração de relatórios.

5.3.1 Cadastro de usuários

Os usuários do sistema podem ser classificados em três tipos: administradores do sistema, administradores de unidades e usuários 'somente leitura'. Isso permite um melhor gerenciamento do acesso ao software, definindo padrões de utilização distintos que podem vir a ser úteis dependendo da situação.

Enquanto que um usuário administrador do sistema pode realizar todas as tarefas, incluindo o acesso aos *logs*, cadastro de outros administradores e demais funcionalidades, um usuário administrador de unidade pode realizar consultas, gerar relatórios, cadastrar e efetuar alterações de cadastros relacionados à sua própria unidade.

Em contrapartida, um usuário 'somente leitura' pode verificar a situação de acordos, gerar relatórios e verificar a situação de equipamentos, mas tem seu acesso restrito à parte cadastral. Essa subdivisão permite a descentralização da administração, deixando a cargo das instituições o controle de acesso ao sistema.

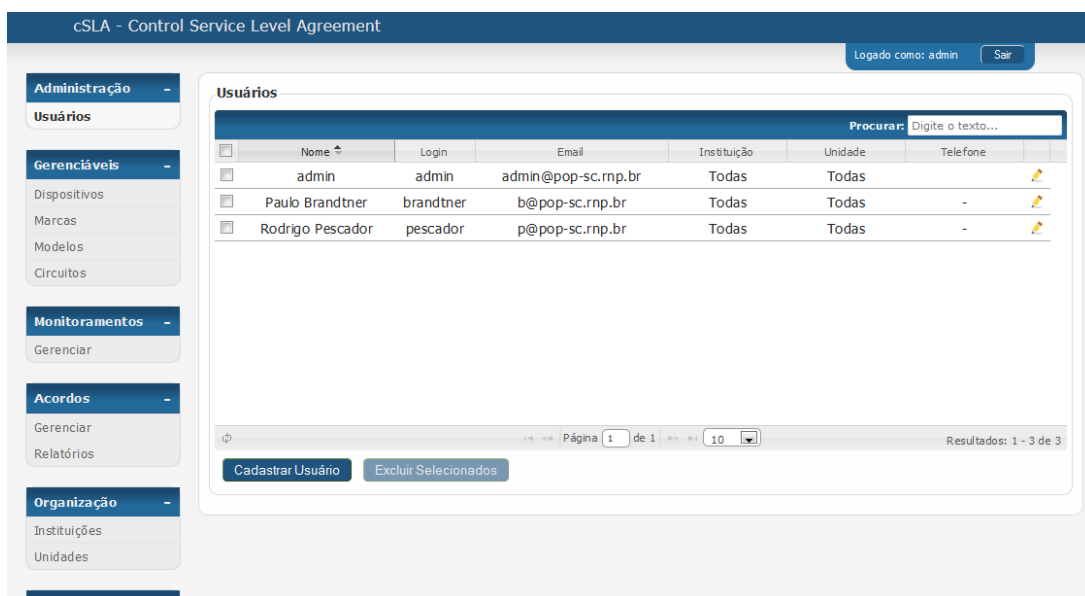


Figura 10 – Interface de cadastro de usuários

5.3.2 Cadastro de acordos

Após cadastrar os usuários e objetos gerenciáveis, é necessário definir e cadastrar os acordos que serão monitorados pelo sistema. Define-se então o tipo, a periodicidade, a forma pela qual os dados serão coletados e os valores crítico e de quebra do acordo.

A princípio prevêem-se periodicidades diárias, semanais, quinzenais, mensais, semestrais e anuais e os seguintes tipos de acordos: disponibilidade, perda de pacotes, latência, jitter médio, banda utilizada e erros de interface.

A forma de coleta dos dados se dará inicialmente através de três maneiras: ICMP, SNMP Interface e SNMP OSPF.

Com os parâmetros definidos e cadastrados, a próxima etapa compreende o cadastro dos monitoramentos que serão analisados dentro de cada acordo.

The screenshot shows a web application interface for 'cSLA - Control Service Level Agreement'. The top navigation bar includes the title and a user status 'Logado como: admin' with a 'Sair' button. A left sidebar contains a menu with categories: 'Administração' (Usuários), 'Gerenciáveis' (Dispositivos, Marcas, Modelos, Circuitos), 'Acordos' (Gerenciar), 'Monitoramentos' (Gerenciar), and 'Organização' (Instituições, Unidades). The main content area is titled 'Acordos' and contains a sub-section 'CADASTRO DE ACORDOS'. This form includes fields for 'Nome*', 'Tipo de Acordo*' (dropdown), 'Tipo de Monitoramento*' (dropdown), and 'Período*' (dropdown). Below these are fields for 'Descrição', 'Valor de Quebra', and 'Valor Crítico'. At the bottom of the form are 'Cadastrar' and 'Voltar' buttons.

Figura 11 - Tela de cadastro de acordos

5.3.3 Cadastro de monitoramentos

Para poder estabelecer o atual estado dos dispositivos que devem ser monitorados dentro de um acordo, são criados os chamados 'monitoramentos', os quais são associados a acordos previamente definidos e identificam o dispositivo onde deve ser feita a coleta de dados.

Para efetuar essa coleta, scripts que rodam paralelamente realizam consultas nesses equipamentos através do método escolhido e armazenam a informação a respeito deles no banco de dados para posterior geração de relatórios.

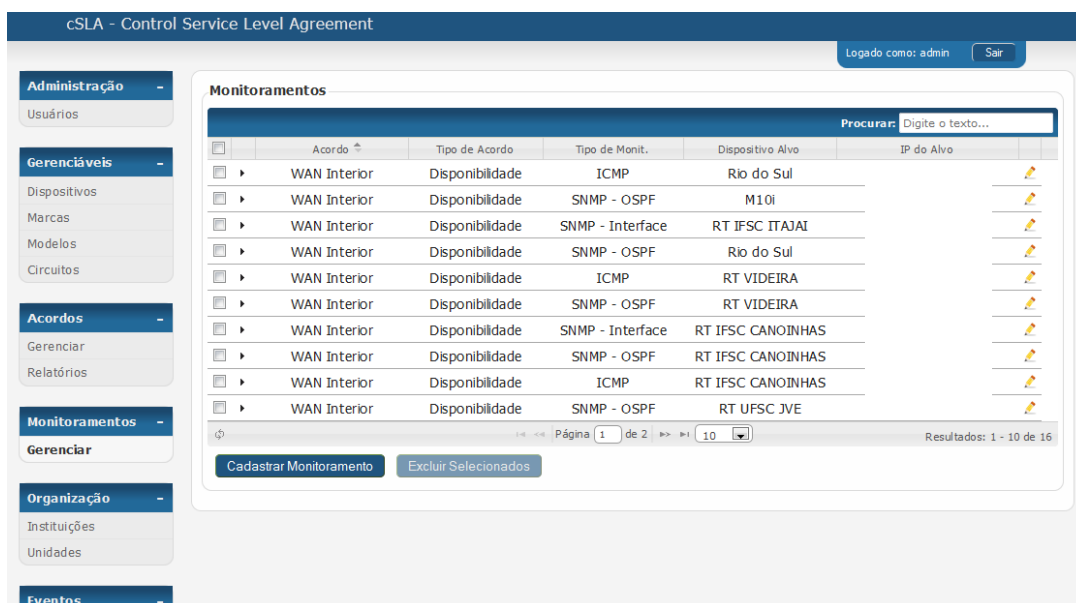


Figura 12 - Tela de cadastro de monitoramentos

5.3.4 Registro e acompanhamento de eventos

Durante a execução dos scripts para identificação dos estados dos dispositivos, são gerados os chamados eventos, os quais também ficam armazenados no banco de dados para consulta posterior e podem ser associados às quedas que ocorreram durante determinado período.

Uma vez definidos os eventos, o sistema permite a inclusão de informações a respeito deles. É possível descrevê-los, informando o possível motivo do evento, como, por exemplo, registrar a abertura de chamado para reparo junto ao prestador de serviço ou informando uma possível falha no fornecimento de energia elétrica. Além disso, o software permite relacionar um número de abertura de ticket junto à operadora para tornar mais fácil a sua localização.

Depois de determinado o evento, ainda é possível incluir comentários quando da mudança de estado do mesmo, notificando os envolvidos e responsáveis da atual situação e permitindo o acompanhamento da resolução do problema.

5.3.5 Geração de relatórios

Outra funcionalidade prevista pelo sistema é o módulo de geração de relatórios, sendo possível definir parâmetros como tipo de acordo, tipo de monitoramento, unidade, métrica ou um conjunto deles.

O usuário poderá selecionar o intervalo de tempo para o qual deseja gerar o documento e esse contemplará as informações referentes aos eventos ocorridos no período, como quedas, abertura de chamados e outras anotações relevantes.

Uma das ideias propostas pelo módulo é gerar um documento basicamente pronto no formato DOC, permitindo que o usuário utilize ferramentas de edição de arquivos de texto e faça as alterações necessárias e observações pertinentes.

Para isso, serão utilizados templates que auxiliarão na automatização e confecção do relatório dos acordos definidos, que muitas vezes são periodicamente solicitados pelo cliente.

5.3.6 Manutenções programadas

Observa-se geralmente nas unidades uma necessidade de realização de manutenções preventivas em alguns de seus dispositivos, sendo que esses podem fazer parte de acordos que estão sendo monitorados pelo sistema.

Essas paradas, chamadas de manutenções programadas, não podem e não devem afetar os valores coletados pelo software para definir uma possível quebra de acordo, portanto devem ser desconsideradas no momento da geração do relatório.

Tendo em vista essa necessidade, o aplicativo permite cadastrar essas paradas, informando o motivo e período no qual o dispositivo pode ficar indisponível e esse período será descontado do cálculo total do SLA.

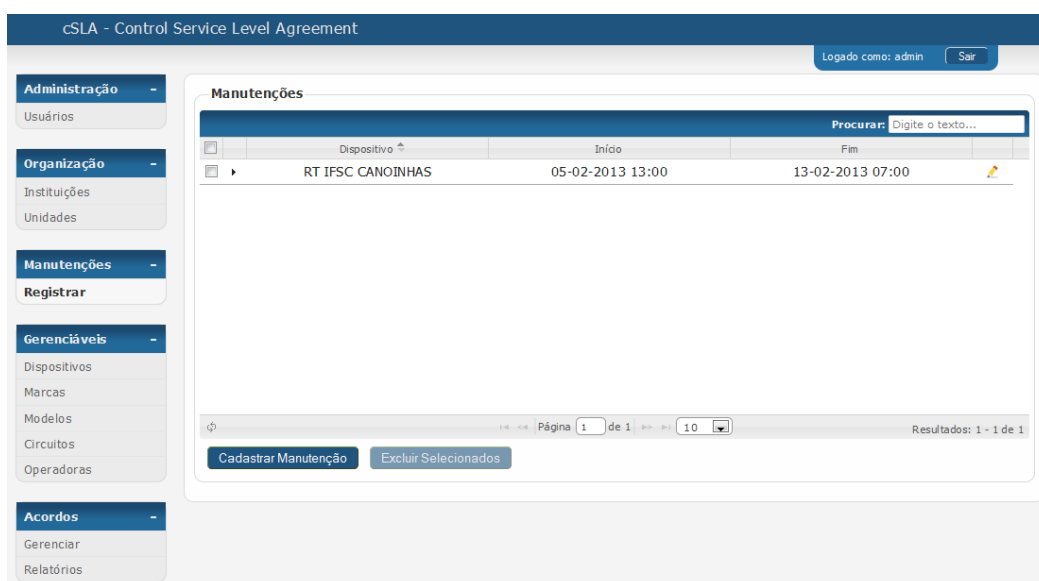


Figura 13 – Interface de cadastro de manutenções programadas

5.4 Modelagem do banco de dados

As Figura 14 e Figura 15 representam as modelagens do banco de dados utilizado no sistema.

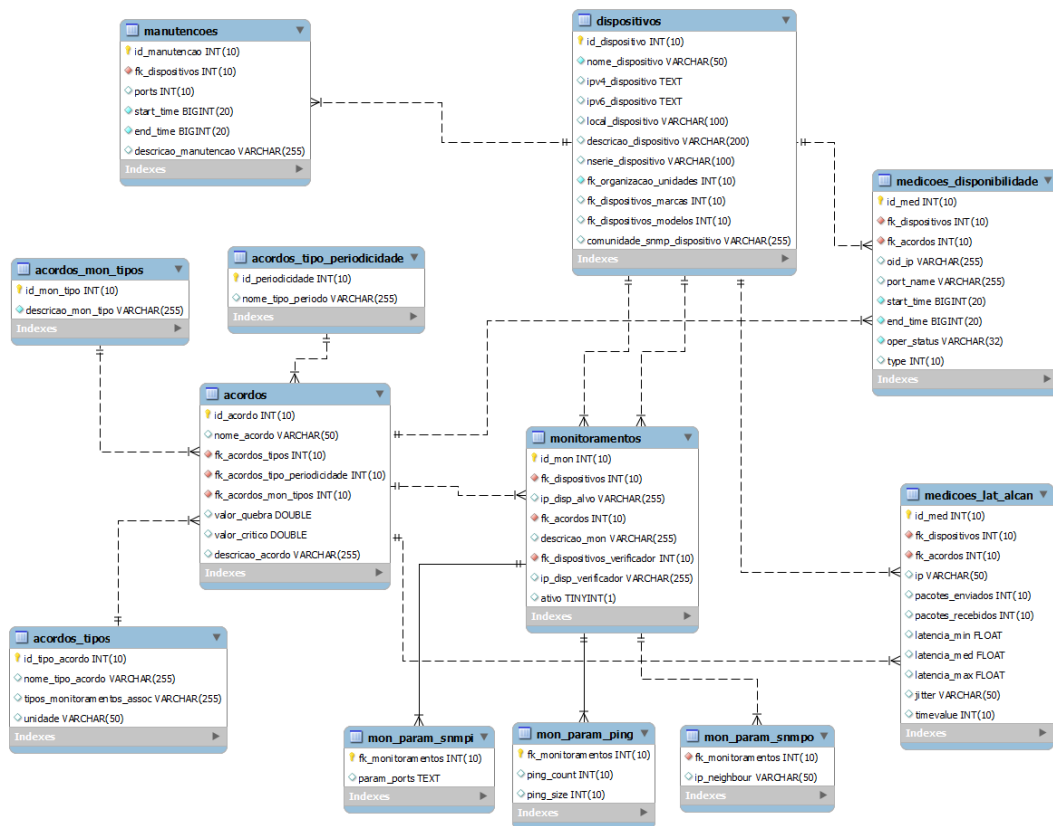


Figura 14 - Esquema do banco (parte 1)

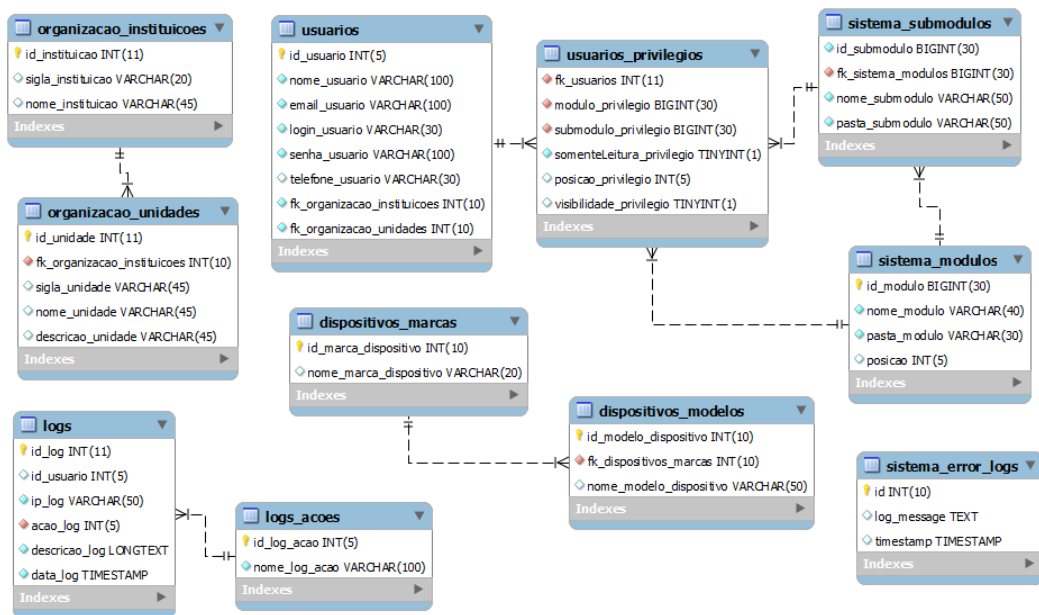


Figura 15 - Esquema do banco (parte 2)

5.5 Resultados

A interface para visualização dos resultados ainda está em fase de desenvolvimento, portanto os dados obtidos ainda não podem ser mostrados através de capturas de telas, mas durante o desenvolvimento do trabalho alguns monitoramentos de teste foram cadastrados e foi possível acompanhar a coleta de informações através dos scripts.

Nas Figura 16 e Figura 17 podemos observar um esboço do resultado obtido por dois exemplos de monitoramentos, um com uma queda registrada e outro com uma interface apresentando tempo de uptime de 100%.

Analisando a Figura 16 e Figura 17, verificamos que no mesmo intervalo de tempo, para duas portas distintas de um mesmo dispositivo, obtivemos resultados diferentes.

No primeiro caso, para a porta lsq-1/3/0.5, a disponibilidade foi de 99,79% e foi registrada uma queda de 01h49min no dia 16 de fevereiro de 2013. No segundo caso, não foram registradas quedas, portanto, o índice de disponibilidade dessa interface foi de 100%.

Através do sistema de eventos previamente mencionado, o operador da instituição pode descrever e associar informações à indisponibilidade ocorrida e no momento da geração do relatório essas informações podem ser levantadas permitindo um melhor entendimento da causa da queda pelo cliente.

Período: 15-01-2013 00:00:00 a 20-02-2013 13:00:00
DISPOSITIVO: M10i
PORTA: lsq-1/3/0.5
UPTIME: 876:07:00 ==> 99.79%
DOWNTIME: 01:49:00
INDEFINIDO: 00:04:00

QUEDAS
16-02-2013 19:09:02 -> 16-02-2013 20:58:02

Figura 16 – Exemplo de resultado obtido por um monitoramento com evento de queda.

Período: 15-01-2013 00:00:00 a 20-02-2013 13:00:00
DISPOSITIVO: M10i
PORTA: ge-0/0/1
UPTIME: 877:58:00 ==> 100.00%
INDEFINIDO: 00:02:00

Figura 17 – Exemplo de resultado obtido por um monitoramento com disponibilidade de 100%.

É importante ressaltar que pelo fato das medições serem realizadas de minuto em minuto, temos o que chamamos de período indefinido ou tempo de incerteza, período esse no qual não é possível avaliar com precisão o atual status da interface. Se houve mudança de estado entre uma medição e outra, é contabilizado um minuto de tempo de incerteza que para efeitos de cálculos é considerado como disponível.

Esses resultados, juntamente com as informações registradas nos sistema de eventos e outras informações pertinentes, somadas aos gráficos, também em fase de desenvolvimento, compõem os relatórios que são gerados pelo software para verificação de quebra de acordos e a manutenção da qualidade do nível de serviço pela operadora, podendo servir para o cliente acompanhar de forma mais organizada se o acordo está sendo cumprido e se há penalidades ou multas a serem pagas por parte da prestadora do serviço.

6. Considerações finais e trabalhos futuros

Pessoas envolvidas na área de tecnologia podem perceber a crescente necessidade de se adotar um SLA para aferir e garantir a qualidade dos serviços prestados. Para um controle eficiente destes acordos, se faz necessária uma ferramenta integrada que meça e gere os respectivos indicadores em forma de relatórios. A ferramenta desenvolvida apresenta ao administrador as informações necessárias à aplicação do SLA, porém cabe a organização definir de que forma estes dados serão usados para a manutenção de um bom nível de prestação dos serviços (seja por multas ou por descontos em serviços).

O sistema é de grande auxílio para o acompanhamento dos acordos estabelecidos, mas não basta inserir um valor fictício de nível de serviço e esperar que ele seja cumprido. Como citado ao longo do trabalho, o SLA é a negociação entre duas ou mais partes que devem chegar a um entendimento comum em relação a qualidade da entrega do serviço, responsabilidades, etc. Sendo assim, deve haver um esforço dos dois lados para que os objetivos sejam alcançados.

Para uma grande organização como a RNP, onde existe a contratação de muitos serviços de muitos prestadores diferentes, gerenciar cada um destes acordos é uma tarefa que dispende tempo. Qualquer ponto que possa ser automatizado tem um ganho significativo no processo. É possível dizer que a ferramenta desenvolvida irá trazer, principalmente, uma grande economia de tempo na confecção dos relatórios, uma vez que seu processo é praticamente todo automatizado através da

geração de templates editáveis. A principal arma de um SLA é um relatório bem elaborado e detalhado sobre os eventos e acontecimentos.

Por fim, vale destacar que no mercado não existem ferramentas de uso livre com as funcionalidades aqui apresentadas.

Roadmap de implementação de funcionalidades próximas:

- Criação de módulo para cálculo automático de valor de multa baseado nas medições realizadas pelo sistema;
- Geração de alertas aos operadores da rede informando acordos que estejam próximos da quebra do contrato;
- Relatório de eficiência das manutenções realizadas (verifica se uma janela de manutenção foi marcada muito grande em relação o serviço efetuado);
- Console para equipe de operação acompanhar o estado dos acordos;
- Novos tipos de acordos (uptime de equipamentos, uso de CPU, etc.);
- Coleta de medições distribuída por agentes remotos;

7. Referências bibliográficas

BABIN, Lee. **Beginning Ajax with PHP From Novice to Professional**. California: Apress, 253p, 2007.

CONTESSA, Diego Fraga; POLINA, Everton Rafael. **Gerenciamento de Equipamentos Usando o Protocolo SNMP**. Artigo – Departamento de Pesquisa e Desenvolvimento - CP Eletrônica S.A. Porto Alegre, RS.

DARIE, Cristian; BRINZAREA, Bogdan; CHERECHES-TOSA, Filip; BUCICA, Mihai. **Building Responsive Web Applications AJAX and PHP**. Birmingham: Packt, 273p, 2006.

DIAS, Beethovem Z.; JUNIOR, Nilton A. **Protocolo de Gerenciamento SNMP**. Nota técnica CBPF-NT-006/01.

DUCKETT, Jon. **Accessible XHTML and CSS Web Sites Problem - Design - Solution**. Indiana, Wiley Publishing, 458p, 2005.

FARIA, José C. G. **Ambiente de monitoração de SLA para redes**. Florianópolis, 2004. Trabalho de conclusão de curso - Universidade Federal de Santa Catarina, Centro Tecnológico.

FILIPPETTI, Marco Aurélio. **CCNA 4.1 Guia Completo de Estudo**. Florianópolis: Visual Books, 2008. 480p.

CLEMM, Alexander. **Network Management Fundamentals**. Indianapolis, Cisco Press, 510p, 2007.

STALLINGS, William. **Data and Computer Communications**. Upper Saddle River, Pearson Education, 878p, 2007.

FRANKE, Hans Alberto. **Uma Abordagem de acordo de nível de serviço para computação em nuvem**. 97 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2010.

GODBOLT, Shane. **Why service level agreements? An overview of their origins and best practice**: produção científica internacional. Inform: The newsletter of IFM healthcare. v. 13, n. 3, p. 1-2, 2002.

GRIGONIS, Richard. **Computer Telephony Encyclopedia**. New York: CMP Books, 2000. 563p.

KENNEDY, Bill; MUSCIANO, Chuck; **HTML & XHTML: The Definitive Guide**. 5. ed. California: O'Reilly, 670p, 2002.

LEÃO, Maximilliam M. **Um estudo sobre a concepção e desenvolvimento do acordo de nível de serviço para ambientes de redes de computadores**. Florianópolis, 2002. 127 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

LERDORF, Rasmus; MACINTYRE, Peter; TATROE, Kevin. **Programming PHP**. 2. ed. California: O'Reilly, 540p, 2006.

LOPES, Raquel V. *et al.* **Melhores Práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campus, 2003.

ITU-T – **Telecommunication standardization sector of ITU**. Recomendação E.860. [S.I.]: Framework of a Service Level Agreement, 2002.

NIEDERAUER, Juliano. **Guia de Consulta Rápida: Integrando PHP 5 com MySQL**. 2. ed. São Paulo: Novatec, 2008.

MANSFIELD, Richard. **CSS Web Design For Dummies**. Indiana, Wiley Publishing, 362p, 2005.

PACHEV, Alexander Sasha. **MySQL Enterprise Solutions**. Indianapolis: Wiley Publishing, 2003. 398p.

PEREIRA, Mateus C. **Análise de desempenho em redes Wireless ad-hoc e estabelecimento de um acordo de nível de serviço pró-ativo**. 2004. 130 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

PIMENTEL, Adda Regina de Oliveira; QUINTELLA, Heitor Luiz Murat de Meirelles. **A importância do SLA no setor de telecomunicações**. Rev. Eletrônica Produção & Engenharia. v. 1, n. 1, p. 1-12, set./dez. 2008.

PAFFENBERGER, Brian; SCHAFER, Steven M.; WHITE, Charles; KAROW, Bill. **HTML, XHTML, and CSS Bible**. 3. ed. Indiana: Wiley Publishing, 790p, 2004.

RIGHI, Rafael R.;PELLISSARI, Felipe R.; WESTPHALL, Carlos B. **Especificação e validação de métricas para acordos de níveis de serviço orientados à segurança**. Artigo – Universidade Federal de Santa Catarina. Florianópolis, SC, 2004.

ROBBINS, Jennifer N. **Learning Web Design, A Beginner's Guide to (X)HTML, Style Sheets, and Web Graphics**. 3. ed. California, O'Reilly, 464p, 2007.

SILVA, Marco Antônio da. **Um SLA para VoIP e seu mapeamento em uma rede DiffServ/MPLS**. Florianópolis, 2005. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

STALLINGS, W. **SNMP, SNMPv2, SNMPv3 and RMON1 and 2**. 3. ed. Addison-Wesley, 1998.

WELLING, Luke; THOMSON, Laura. **PHP e MYSQL Desenvolvimento Web**. 3ª ed. Rio de Janeiro: Elsevier, 2005. 754p.

ZAKAS, Nicholas C.; McPEAK, Jeremy; FAWCETT, Joe. **Professional Ajax**. Indiana, Wiley Publishing, 380p, 2006.

STURM, Rick; MORRIS, Wayne; JANDER, Mary. **Service Level Management**. Campus, 2001. 272p.

Gerenciamento Integrado de Modems via SNMP: Um Estudo de Caso.

Disponível em
<<http://www.ufrgs.br/niee/eventos/SBC/2000/pdf/semish/semi011.pdf>>. Acessado em
03/11/2012 15:27h.

DZUBIN, Thomas. 2007. Disponível em <<http://fping.sourceforge.net>>.
Acessado em 10/12/2012.

8. Apêndice

Código fonte:

```
<?php
//Modulos:
define("ID_MODULO_SISTEMA", 1);

define("ID_SUBMODULO_ADMIN", 1);
define("ID_SUBMODULO_ORGANIZACAO", 2);
define("ID_SUBMODULO_GERENCIAVEIS", 4);
define("ID_SUBMODULO_ACORDOS", 8);
define("ID_SUBMODULO_MONITORAMENTOS", 16);
define("ID_SUBMODULO_MANUTENCOES", 32);
define("ID_SUBMODULO_EVENTOS", 64);

//Central:
define("ID_SUBMODULO_ADMIN_USUARIOS", 1);
define("ID_SUBMODULO_ORGANIZACAO_INSTITUICOES", 2);
define("ID_SUBMODULO_ORGANIZACAO_UNIDADES", 4);
define("ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS", 8);
define("ID_SUBMODULO_GERENCIAVEIS_MARCAS", 16);
define("ID_SUBMODULO_GERENCIAVEIS_MODELOS", 32);
define("ID_SUBMODULO_ACORDOS_GERENCIAR", 64);
define("ID_SUBMODULO_MONITORAMENTOS_GERENCIAR", 128);
define("ID_SUBMODULO_MANUTENCOES_REGISTRAR", 256);
define("ID_SUBMODULO_ACORDOS_RELATORIOS", 512);
define("ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS", 1024);
define("ID_SUBMODULO_EVENTOS_REGISTRAR", 2048);
define("ID_SUBMODULO_EVENTOS_ACOMPANHAMENTO", 4096);
define("ID_SUBMODULO_GERENCIAVEIS_OPERADORAS", 8192);

//Template:
define("TITULO_PADRAO", "cSLA");
```



```

define("TEMPLATE_GLOBAL", "global.php");
define("TEMPLATE_CABECALHO", "header.php");
define("TEMPLATE_CORPO", "body.php");
define("TEMPLATE_RODAPE", "footer.php");
define("TEMPLATE_MENU", "menu.php");
define("TEMPLATE_ACESSO_NEGADO", "unauthorized.php");
define("TEMPLATE_ERRO", "error.php");

define("CAMINHO_PUBLICO", realpath(dirname(__FILE__) . '/../../public/'));
define("CAMINHO_RAIZ", realpath(dirname(__FILE__) . '/../'));
define("CAMINHO_IMAGENS", CAMINHO_PUBLICO . "/_img/");

define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../'));
define('INCLUDE_PATH', APPLICATION_PATH . "/include/");
define('CONTROLLER_PATH', APPLICATION_PATH . "/controle/");

if (isset($_SERVER['REQUEST_URI'])) {
    $ROOT_URI = explode('/public', $_SERVER['REQUEST_URI']);
    $ROOT_URI[0] = '/public/';
    define('URI_RAIZ', 'http://' . htmlentities($_SERVER['SERVER_NAME']) . $ROOT_URI);
    define("URI_CONFIG", URI_RAIZ . "config/");
    define("URI_CSS", URI_RAIZ . "_css/");
    define("URI_JS", URI_RAIZ . "_js/");
    define("URI_IMAGENS", URI_RAIZ . "_img/");
    define("URI_TEMPLATE", URI_RAIZ . "template/");
    define("URI_SOBRE", URI_RAIZ . "sobre/");
    define("URI_ERRO", URI_RAIZ);

    define("URI_ORGANIZACAO", URI_RAIZ . "organizacao/");
    define("URI_ORGANIZACAO_INSTITUICOES", URI_ORGANIZACAO . "instituicoes/");
    define("URI_ORGANIZACAO_UNIDADES", URI_ORGANIZACAO . "unidades/");

    define("URI_ADMIN", URI_RAIZ . "admin/");
    define("URI_ADMIN_USUARIOS", URI_ADMIN . "usuarios/");
    define("URI_ADMIN_LOGS", URI_ADMIN . "logs/");

    define("URI_GERENCIAVEIS", URI_RAIZ . "gerenciaveis/");
    define("URI_GERENCIAVEIS_DISPOSITIVOS", URI_GERENCIAVEIS . "dispositivos/");
    define("URI_GERENCIAVEIS_MARCAS", URI_GERENCIAVEIS . "marcas/");
    define("URI_GERENCIAVEIS_MODELOS", URI_GERENCIAVEIS . "modelos/");
    define("URI_GERENCIAVEIS_CIRCUITOS", URI_GERENCIAVEIS . "circuitos/");
    define("URI_GERENCIAVEIS_OPERADORAS", URI_GERENCIAVEIS . "operadoras/");

    define("URI_ACORDOS", URI_RAIZ . "acordos/");
    define("URI_ACORDOS_GERENCIAR", URI_ACORDOS . "gerenciar/");
    define("URI_ACORDOS_RELATORIOS", URI_ACORDOS . "relatorios/");

    define("URI_MONITORAMENTOS", URI_RAIZ . "monitoramentos/");
    define("URI_MONITORAMENTOS_GERENCIAR", URI_MONITORAMENTOS . "gerenciar/");

    define("URI_MANUTENCOES", URI_RAIZ . "manutencoes/");
    define("URI_MANUTENCOES_REGISTRAR", URI_MANUTENCOES . "registrar/");

    define("URI_EVENTOS", URI_RAIZ . "eventos/");
    define("URI_EVENTOS_REGISTRAR", URI_EVENTOS . "registrar/");
    define("URI_EVENTOS_ACOMPANHAMENTO", URI_EVENTOS . "registrar/");

    if (isset($_SERVER['SERVER_ADDR'])) {
        define('SERVER_IP', $_SERVER['SERVER_ADDR']); }
}

```

```

    }

?>

<?php
/*
 * Arquivo de Configuração cSLA
 * -----
 * SÓ ALTERE SE SOUBER O QUE ESTIVER FAZENDO
 */

require_once("constantes.php");
require_once(INCLUDE_PATH . "/Logger.php");
require_once(INCLUDE_PATH . "/ErrorHandler.php");
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/funcoes.php");

/* Informações gerais sobre a interface */

global $config;
$config["versao"] = "0.3 - 25/02/2013";

/* Configurações do MySQL */
$config["mysql_ip"] = "";
$config["mysql_usuario"] = "";
$config["mysql_senha"] = '';
$config["mysql_base"] = "";

$config['firephp_log_enabled'] = true;
$config['database_log_enabled'] = true;
$config['file_log_enabled'] = true;
$config['file_log_location'] = "/var/log/csla/csla.log";

/*ERROR LEVELS
E_ERROR:                // 1 //
E_WARNING:              // 2 //
E_PARSE:                // 4 //
E_NOTICE:               // 8 //
E_STRICT:               // 2048 //
E_DEPRECATED:          // 8192 //
ALL ERRORS:             // 32767 //
*/

$config['database_log_level'] = 7;
$config['file_log_level'] = 32767;

//Caminho MIB para descrição de interfaces
$config['ifDescr'] = "iso.3.6.1.2.1.2.2.1.2";

$config['ifOperStatus'] = "iso.3.6.1.2.1.2.2.1.8";
$config['ifOperStatus_expectedValue'] = "1";

$config['OSPF'] = ".1.3.6.1.2.1.14.10.1.6";
$config['OSPF_expectedMinValue'] = "5";

set_error_handler("myErrorHandler");

```

```

error_reporting(0);
ini_set('display_errors',0);

?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleAcordo{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleAcordo();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
     JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_ACORDOS_GERENCIAR)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /**
         * Verifica os dados do parâmetro enviado
         */
        $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
        $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
        $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
        $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
        $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

        $response = new stdClass();

        /**
         * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
         query para filtrar os dados
         */
        $params = array();

```

```

$where = "";
if(isset($search) && $search != ""){
    $response->filtro = true;
    $search = normalize($search);
    $where = " WHERE nome_acordo LIKE ?
                OR descricao_mon_tipo LIKE ?
                OR nome_tipo_periodo LIKE ?
                OR nome_tipo_acordo LIKE ?
                OR valor_critico LIKE ?
                OR valor_quebra LIKE ?";

    $params = array('ssdds',
        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"));
}

$result = $this->DB->execPreparedSQL("SELECT COUNT(*) as count FROM acordos a
                                JOIN
acordos_tipo_periodicidade atp ON a.`fk_acordos_tipo_periodicidade` =
atp.`id_periodicidade`
                                JOIN
acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo`
                                JOIN
`acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo` $where", $params);

if(!empty($result)){
    $count = $result[0]['count'];
    if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

    if ($page > $total_pages){ $page = $total_pages; }
    if ($limit < 0){ $limit = 0; }
    $start = $limit * $page - $limit;
    if ($start < 0){ $start = 0; }

    $order = "";
    if(isset($sidx) && $sidx!=""){
        $order = ($sord=='asc')?"ASC":"DESC";
        $order = "ORDER BY $sidx $order";
    }

    $result = $this->DB->execPreparedSQL("SELECT id_acordo, nome_acordo,
nome_tipo_periodo, nome_tipo_acordo,
    descricao_mon_tipo, valor_quebra, valor_critico
    FROM acordos a
    JOIN `acordos_tipo_periodicidade` atp ON a.`fk_acordos_tipo_periodicidade` =
atp.`id_periodicidade`
    JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo`
    JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo` $where $order
LIMIT $start,$limit", $params);

```

```

$response->page = $page;
$response->total = $total_pages;
$response->records = $count;

$i = 0;
foreach ($result as $res){

    $response->rows[$i]['id'] = $res['id_acordo'];
    $response->rows[$i]['cell'] = array(
        format_title_trim($res['nome_acordo'],15),
        format_title_trim($res['nome_tipo_periodo'],15),
        format_title_trim($res['nome_tipo_acordo'],25),
        format_title_trim($res['descricao_mon_tipo'],12),
        str_replace(".",",",sprintf("%.2f",$res['valor_quebra'])),
        str_replace(".",",",sprintf("%.2f",$res['valor_critico'])),
        "<img class='edit' src='\" . URI_IMAGENS . \"edit.png'
border='0' />"
    );
    $i++;
}

}
exit(json_encode($response));
}

public function add_update(){

if (!$this->auth->check(ID_SUBMODULO_ACORDOS_GERENCIAR)) {
    exit(json_encode(array('status' => 'unauthorized')));
}

$invalid_fields = array();

foreach ($_REQUEST as $key => $value) {
    if (is_array($value)) {
        $value_new = array();
        foreach ($value as $chave=>$value_item) {
            if ($value_item != ""){
                $value_new[$chave] =
anti_injection($value_item);
            }
        }
        $$key = $value_new;
    } else {
        $$key = anti_injection($value);
    }
}

if (!$nome_acordo){
    $invalid_fields['nome_acordo'] = "Nome inválido.";
}

if (isset($fk_acordos_tipos) && !$fk_acordos_tipos){
    $invalid_fields['fk_acordos_tipos'] = "Tipo Inválido.";
}

if (!$fk_acordos_tipo_periodicidade){
    $invalid_fields['fk_acordos_tipo_periodicidade'] = "Período
Inválido.";
}

```

```

    }

    if (isset($fk_acordos_mon_tipos) && !$fk_acordos_mon_tipos){
        $invalid_fields['fk_acordos_mon_tipos'] = "Tipo de Monitoramento
Inválido.";
    }

    $valor_quebra = str_replace(",", ".", $valor_quebra);
    $valor_critico = str_replace(",", ".", $valor_critico);

    if (!is_numeric($valor_quebra)){
        $invalid_fields['valor_quebra'] = "Valor de Quebra Inválido.";
    }

    if (!is_numeric($valor_critico)){
        $invalid_fields['valor_critico'] = "Valor Crítico Inválido.";
    }

    if (!empty($invalid_fields)){
        exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
    }

    $arrayInfo = array();
    if ($acao == 'cadastrar'){
        $arrayInfo["nome_acordo"] = $nome_acordo;
        $arrayInfo["fk_acordos_tipos"] = $fk_acordos_tipos;
        $arrayInfo["fk_acordos_mon_tipos"] = $fk_acordos_mon_tipos;
        $arrayInfo["fk_acordos_tipo_periodicidade"] =
$fk_acordos_tipo_periodicidade;
        $arrayInfo["valor_quebra"] = $valor_quebra;
        $arrayInfo["valor_critico"] = $valor_critico;
        $arrayInfo["descricao_acordo"] = $descricao_acordo;
    } else {
        $result = $this->DB->execPreparedSQL("SELECT * FROM acordos WHERE
id_acordo = ?", array('i', $this->DB->escape($id)));
        $valorAntigo["nome_acordo"] = $result[0]['nome_acordo'];
        $valorAntigo["fk_acordos_tipo_periodicidade"] =
$result[0]["fk_acordos_tipo_periodicidade"];
        $valorAntigo["valor_quebra"] = $result[0]["valor_quebra"];
        $valorAntigo["valor_critico"] = $result[0]["valor_critico"];
        $valorAntigo["descricao_acordo"] = $result[0]["descricao_acordo"];

        if ($valorAntigo["nome_acordo"] != $nome_acordo) {
            $arrayInfo["nome_acordo"] = $nome_acordo; }
        if ($valorAntigo["fk_acordos_tipo_periodicidade"] !=
$fk_acordos_tipo_periodicidade) { $arrayInfo["fk_acordos_tipo_periodicidade"] =
$fk_acordos_tipo_periodicidade; }
        if ($valorAntigo["valor_quebra"] != $valor_quebra) {
            $arrayInfo["valor_quebra"] = $valor_quebra; }
        if ($valorAntigo["valor_critico"] != $valor_critico) {
            $arrayInfo["valor_critico"] = $valor_critico; }
        if ($valorAntigo["descricao_acordo"] != $descricao_acordo) {
            $arrayInfo["descricao_acordo"] = $descricao_acordo; }
    }

    }

    $log = "";

```

```

$params = array();
$posicoes = array();
$update_query = array();

if ($acao == 'cadastrar'){
    $action = "CADASTRO DE ACORDO";
    $action_text = "cadastrado";

    $params[] = $this->DB->getTypeLabel($arrayInfo);
    $log = "<b>Atributos cadastrados:</b><br />";
    foreach ($arrayInfo as $key => $valor) {
        $log .= $key . ": " . $valor . "<br />";
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("INSERT INTO acordos
(`".implode("`", $chaves_sql).`) VALUES ("`.implode("`", $posicoes).`)", $params, true);

} elseif ($acao == 'alterar') {
    if (!empty($arrayInfo)) {
        $action = "ALTERAÇÃO DE ACORDO";
        $action_text = "alterado";

        $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

        $log .= "<b>Acordo alterado:<br/>
{$valorAntigo["nome_acordo"]}<br />";
        $log .= "<b>Atributos alterados: </b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $update_query[] = "`$key` = ?";
            $params[] = $this->DB->escape($valor);
        }
        $params[] = $this->DB->escape($id);
        $result = $this->DB->execPreparedSQL("UPDATE acordos SET
".implode("`", $update_query).` WHERE id_acordo = ?", $params, true);

    } else {
        exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
    }
} else {
    exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
}

if (!$this->Logger->logACTION($action, $log))
return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

exit(json_encode(array('status' => 'success', 'msg' => "Acordo $action_text
com sucesso !")));

}

public function delete(){

```

```

        if (!$this->auth->check(ID_SUBMODULO_ACORDOS_GERENCIAR)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }

        $ids = anti_injection($_REQUEST["ids"]);
        $log = "<b>Acordos deletados:</b><br/>";

        $params = array();
        $posicoes = array();
        $params[] = $this->DB->getTypeLabel($ids);
        foreach ($ids as $key=>$valor) {
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("SELECT nome_acordo FROM acordos WHERE
id_acordo IN (".implode(",",$posicoes).")", $params);

        foreach ($result as $res){ $log .= "{$res['nome_acordo']}<br/>"; }

        $result = $this->DB->execPreparedSQL("DELETE FROM acordos WHERE id_acordo IN
(implode(",",$posicoes).")", $params, true);

        if (!$this->Logger->logACTION("EXCLUSÃO DE ACORDO", $log))
            return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

        exit(json_encode(array('status'=>'success','msg'=>'Acordo(s) excluído(s) com
sucesso!')));
    }

    public function showInfo(){

        if (!$this->auth->check(ID_SUBMODULO_ACORDOS_GERENCIAR)) {
            exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !")));
        }

        $result = $this->DB->execPreparedSQL("SELECT id_acordo, nome_acordo,
nome_tipo_periodo, nome_tipo_acordo, descricao_mon_tipo,
valor_quebra, valor_critico, descricao_acordo
FROM acordos a
JOIN `acordos_tipo_periodicidade` atp ON a.`fk_acordos_tipo_periodicidade` =
atp.`id_periodicidade`
JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo`
JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo`
WHERE id_acordo = ?", array("i", $this->DB->escape($_REQUEST["id"])));

        if (empty($result)){
            exit(json_encode(array('status' => 'error')));
        } else {
            foreach ($result as $res) {
                $response = "<div style='display:inline'>";
            }
        }
    }

```



```

        $response .= "<strong>Nome:</strong></h4>";
        $.res["nome_acordo"]."<br />";
        $response .= "<strong>Tipo:</strong></h4>";
        $.res["nome_tipo_acordo"]."<br />";
        $response .= "<strong>Tipo de Monitoramento:</strong></h4>";
        $.res["descricao_mon_tipo"]."<br />";
        $response .= "<strong>Período:</strong></h4>";
        $.res["nome_tipo_periodo"]."<br />";
        $response .= "<strong>Valor de Quebra:</strong></h4>";
        $.str_replace(".", "", sprintf("%.2f", $res['valor_quebra']))."%<br />";
        $response .= "<strong>Valor Crítico:</strong></h4>";
        $.str_replace(".", "", sprintf("%.2f", $res['valor_critico']))."%<br />";
        $response .= "<strong>Descrição:</strong></h4>";
        $.res["descricao_acordo"]."<br />";
        $response .= "</div>";
        $response .= "<div class='clear'></div>";
    }

    exit(json_encode(array("status"=>"success", "info"=>$response)));
}

}

public function getInfo(){
    $result = $this->DB->execPreparedSQL("SELECT * FROM acordos a JOIN
`acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo` WHERE id_acordo = ?",
array("i", $this->DB->escape($_REQUEST["id"])));
    if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
}

public function getTiposMon(){
    $id_acordo = (isset($_REQUEST['select'])) ? $this->DB-
>escape($_REQUEST['select']) : ("");

    $result = $this->DB->execPreparedSQL("SELECT tipos_monitoramentos_assoc,
unidade FROM acordos_tipos WHERE id_tipo_acordo = ?", array("i", $id_acordo));
    $tipos_assoc = explode("|", $result[0]['tipos_monitoramentos_assoc']);
    $unidade = $result[0]['unidade'];

    $tipos = "";
    $result = $this->DB->execPreparedSQL("SELECT id_mon_tipo, descricao_mon_tipo
FROM acordos_mon_tipos WHERE id_mon_tipo IN (".implode(", ", $tipos_assoc).")", array());
    foreach ($result as $res){
        $tipos .= "<option
value='{ $res['id_mon_tipo'] }'>". $res['descricao_mon_tipo'] . "</option>";
    }

    $retorno = array("options" => $tipos, "unidade" => $unidade);
    exit (json_encode($retorno));
}

}
?>
<?php

```

```

/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleCircuito{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleCircuito();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
    JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /*
         * Verifica os dados do parâmetro enviado
         */
        $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
        $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
        $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
        $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
        $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

        $response = new stdClass();

        /*
         * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
        query para filtrar os dados
         */
        $params = array();
        $where = "";
        if(isset($search) && $search != ""){
            $response->filtro = true;
            $search = normalize($search);
            $where = " WHERE designacao_circuito LIKE ?
                OR nome_dispositivo LIKE ?

```

```

                OR nome_operadora LIKE ?
                OR port_neighbour LIKE ?";
$params = array('ssss',
$this->DB->escape("%$search%"),
$this->DB->escape("%$search%"),
$this->DB->escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT COUNT(*) AS count FROM circuitos
c
LEFT
JOIN dispositivos d ON c.`fk_dispositivos`=d.`id_dispositivo`
LEFT
JOIN operadoras o ON c.`fk_operadoras`=o.`id_operadora` $where", $params);

    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }

        $result = $this->DB->execPreparedSQL("SELECT id_circuito,
designacao_circuito, nome_dispositivo, nome_operadora, port_neighbour
FROM circuitos c
LEFT JOIN dispositivos d ON c.`fk_dispositivos`=d.`id_dispositivo`
LEFT JOIN operadoras o ON c.`fk_operadoras`=o.`id_operadora` $where $order LIMIT
$start,$limit", $params);

        $response->page = $page;
        $response->total = $total_pages;
        $response->records = $count;

        $i = 0;
        foreach ($result as $res){

            if (strpos($res['port_neighbour'], "::-") {
                $temp = (explode("::",$res['port_neighbour']));
                $port_neighbour = $temp[1];
            } else {
                $port_neighbour = $res['port_neighbour'];
            }

            $response->rows[$i]['id'] = $res['id_circuito'];
            $response->rows[$i]['cell'] = array(
                format_title_trim($res['designacao_circuito'],35),
                format_title_trim($res['nome_dispositivo'],20),

```



```

    } elseif ($tipo_circuito == 2){
        if (empty($ip_neighbour)){
            $invalid_fields['ip_neighbour'] = "É necessário definir
o IP do neighbour.";
        } else if (!validaIP($ip_neighbour)) {
            $invalid_fields['ip_neighbour'] = "IP do neighbour
inválido.";
        } else {
            $port_neighbour = $ip_neighbour;
        }
    }
}

if (!$fk_operadoras){
    $invalid_fields['fk_operadoras'] = "Operadora inválida.";
}

if (!$fk_dispositivos){
    $invalid_fields['fk_dispositivos'] = "Dispositivo alvo inválido.";
}

if ($acao == 'cadastrar'){
    $where = "designacao_circuito = ?";
    $params = array('s', $this->DB->escape($designacao_circuito));
} elseif ($acao == 'alterar') {
    $where = "designacao_circuito = ? and id_circuito <> ?";
    $params = array('si', $this->DB->escape($designacao_circuito), $this-
>DB->escape($id));
}

$result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM circuitos
WHERE $where", $params);
if ($result[0]['count'] >0){ $invalid_fields['designacao_circuito'] =
"Designação já existente."; }

if ($tipo_circuito){
    if ($acao == 'cadastrar'){
        $where = "fk_dispositivos = ? and port_neighbour = ? and
fk_operadoras = ?";
        $params = array('isi', $this->DB->escape($fk_dispositivos),
$this->DB->escape($port_neighbour), $this->DB->escape($fk_operadoras));
    } elseif ($acao == 'alterar') {
        $where = "fk_dispositivos = ? and port_neighbour = ? and
fk_operadoras = ? and id_circuito <> ?";
        $params = array('isii', $this->DB->escape($fk_dispositivos),
$this->DB->escape($port_neighbour), $this->DB->escape($fk_operadoras), $this->DB-
>escape($id));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
circuitos m WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['designacao_circuito'] =
"Já existe circuito cadastrado com esses parâmetros."; }
}

if (!empty($invalid_fields)){

```

```

        exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
    }

    $arrayInfo = array();
    $arrayInfoExtra = array();
    if ($acao == 'cadastrar'){

        $arrayInfo["designacao_circuito"] = $designacao_circuito;
        $arrayInfo["fk_dispositivos"] = $fk_dispositivos;
        $arrayInfo["ip_disp_alvo"] = $ip_disp_alvo;
        $arrayInfo["tipo_circuito"] = $tipo_circuito;
        $arrayInfo["port_neighbour"] = $port_neighbour;
        $arrayInfo["fk_operadoras"] = $fk_operadoras;

    } else {

        $result = $this->DB->execPreparedSQL("SELECT * FROM circuitos c
WHERE id_circuito = ?", array('i', $this->DB->escape($id)));

        $valorAntigo["designacao_circuito"] =
$result[0]["designacao_circuito"];
        $valorAntigo["fk_dispositivos"] = $result[0]['fk_dispositivos'];
        $valorAntigo["ip_disp_alvo"] = $result[0]['ip_disp_alvo'];
        $valorAntigo["tipo_circuito"] = $result[0]['tipo_circuito'];
        $valorAntigo["port_neighbour"] = $result[0]['port_neighbour'];
        $valorAntigo["fk_operadoras"] = $result[0]['fk_operadoras'];

    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){

        $action = "CADASTRO DE CIRCUITO";
        $action_text = "cadastrado";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO circuitos
(`".implode("``,`",$chaves_sql).`) VALUES (".implode(",",$posicoes).")", $params, true);

    } elseif ($acao == 'alterar') {

        if (!empty($arrayInfo)) {

            $action = "ALTERAÇÃO DE CIRCUITO";

```

```

        $action_text = "alterado";

        $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

        $log .= "<b>Atributos alterados: </b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $update_query[] = "`$key` = ?";
            $params[] = $this->DB->escape($valor);
        }

        $params[] = $this->DB->escape($id);

        if (!empty($update_query)){
            $result = $this->DB->execPreparedSQL("UPDATE circuitos
SET ".implode(", ", $update_query)." WHERE id_circuito = ?", $params, true);
        }

        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Circuito $action_text
com sucesso !")));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Circuitos deletados:</b><br />";
    $log .= implode(",",$ids)."<br />";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("DELETE FROM circuitos WHERE id_circuito
IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE CIRCUITO", $log))

```

```

        return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

        exit(json_encode(array('status'=>'success', 'msg'=>'Circuito(s) excluído(s)
com sucesso!'))));
    }

    public function showInfo(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS)) {
            exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !")));
        }

        $result = $this->DB->execPreparedSQL("SELECT * FROM circuitos c
JOIN dispositivos d ON c.`fk_dispositivos`=d.`id_dispositivo`
JOIN operadoras o ON c.`fk_operadoras`=o.`id_operadora`
WHERE id_circuito = ?", array("i", $this->DB->escape($_REQUEST["id"])));

        if (empty($result)){
            exit(json_encode(array('status' => 'error')));
        } else {
            foreach ($result as $res) {

                $designacao_circuito = ($res['designacao_circuito']) ?
$res['designacao_circuito'] : "-";
                $nome_dispositivo = $res['nome_dispositivo'];
                $nome_operadora = $res['nome_operadora'];
                $port_neighbour = $res['port_neighbour'];

                $response = "<div style='display:inline'>";
                $response .= "<strong>Designação do Circuito:</strong></h4>";
                $response .= "<strong>Nome do Dispositivo:</strong></h4>";
                $response .= "<strong>Nome da Operadora:</strong></h4>";
                $response .= "<strong>Porta ou IP do Neighbour:</strong></h4>";
                $response .= "</div>";
                $response .= "<div class='clear'></div>";
            }

            exit(json_encode(array("status"=>"success", "info"=>$response)));
        }
    }

    public function getInfo(){
        $result = $this->DB->execPreparedSQL("SELECT * FROM circuitos c
JOIN dispositivos d ON c.`fk_dispositivos`=d.`id_dispositivo`
JOIN operadoras o ON c.`fk_operadoras`=o.`id_operadora`
WHERE id_circuito = ?", array("i", $this->DB->escape($_REQUEST["id"])));

```

LEFT

LEFT


```

        if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
    }

}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleDispositivo{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleDispositivo();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /*
         * Verifica os dados do parâmetro enviado
         */
        $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
        $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
        $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
        $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
        $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

        $response = new stdClass();

        /*
         * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
query para filtrar os dados

```

```

*/
$params = array();
$where = "";
if(isset($search) && $search != ""){
    $response->filtro = true;
    $search = normalize($search);
    $where = " WHERE nome_dispositivo LIKE ?
                OR ipv4_dispositivo LIKE ?
                OR ipv6_dispositivo LIKE ?
                OR CONCAT(sigla_instituicao,' - ',
sigla_unidade) LIKE ?";
    $params = array('ssss',
    $this->DB->escape("%$search%"),
    $this->DB->escape("%$search%"),
    $this->DB->escape("%$search%"),
    $this->DB->escape("%$search%"));
}

$result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos d

JOIN organizacao_unidades ou ON d.`fk_organizacao_unidades` = ou.`id_unidade`

JOIN organizacao_instituicoes oi ON ou.`fk_organizacao_instituicoes` =
oi.`id_instituicao` $where", $params);

if(!empty($result)){
    $count = $result[0]['count'];
    if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

    if ($page > $total_pages){ $page = $total_pages; }
    if ($limit < 0){ $limit = 0; }
    $start = $limit * $page - $limit;
    if ($start < 0){ $start = 0; }

    $order = "";
    if(isset($sidx) && $sidx!=""){
        $order = ($sord=='asc')?"ASC":"DESC";
        $order = "ORDER BY $sidx $order";
    }

    $result = $this->DB->execPreparedSQL("SELECT id_dispositivo,
nome_dispositivo, ipv4_dispositivo, ipv6_dispositivo,

CONCAT(sigla_instituicao,' - ', sigla_unidade) AS concat_inst_unidade_dispositivo

FROM dispositivos d

JOIN organizacao_unidades ou ON d.`fk_organizacao_unidades` = ou.`id_unidade`

JOIN organizacao_instituicoes oi ON ou.`fk_organizacao_instituicoes` =
oi.`id_instituicao` $where $order LIMIT $start,$limit", $params);

    $response->page = $page;
    $response->total = $total_pages;
    $response->records = $count;

    $i = 0;

```

```

        foreach ($result as $res){
            $res['ipv4_dispositivo'] = $res['ipv4_dispositivo'] ?
$res['ipv4_dispositivo'] : "-";
            $res['ipv6_dispositivo'] = $res['ipv6_dispositivo'] ?
$res['ipv6_dispositivo'] : "-";

            $response->rows[$i]['id'] = $res['id_dispositivo'];
            $response->rows[$i]['cell'] = array(
                format_title_trim($res['nome_dispositivo'],20),
                str_replace("|","\n",$res['ipv4_dispositivo']),
                str_replace("|","\n",$res['ipv6_dispositivo']),
                format_title_trim($res['concat_inst_unidade_dispositivo'],15),
                "<img class='edit' src='\" . URI_IMAGENS . \"edit.png'
border='0' />"
            );
            $i++;
        }
    }
    exit(json_encode($response));
}

public function add_update(){
    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($_REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $chave=>$value_item) {
                if ($value_item != ""){
                    $value_new[$chave] =
anti_injection($value_item);
                }
            }
            $$key = $value_new;
        } else {
            $$key = anti_injection($value);
        }
    }

    if (!$nome_dispositivo){
        $invalid_fields['nome_dispositivo'] = "Nome inválido.";
    }

    if (!$fk_organizacao_unidades){
        $invalid_fields['fk_organizacao_unidades'] = "Unidade inválida.";
    }

    if ((!$enderecos_ipv4) && (!$enderecos_ipv6)) {
        $invalid_fields['enderecosipv4_0'] = "";
        $invalid_fields['enderecosipv6_0'] = "Pelos menos um endereço IPv4 ou
IPv6 necessário.";
    } else {

```

```

        if ($enderecos_ipv4){
            foreach($enderecos_ipv4 as $pos=>$ipv4){
                if (!validaIP($ipv4)){
                    $invalid_fields["enderecosipv4_$pos"] =
"Endereço IPv4 - ".$pos+1)." Inválido.";
                }
            }
        }
        if ($enderecos_ipv6){
            foreach($enderecos_ipv6 as $pos=>$ipv6){
                if (!validaIP($ipv6)){
                    $invalid_fields["enderecosipv6_$pos"] =
"Endereço IPv6 - ".$pos+1)." Inválido.";
                }
            }
        }
    }

    if ($acao == 'cadastrar'){
        $where = "nome_dispositivo = ?";
        $params = array('s', $this->DB->escape($nome_dispositivo));
    } elseif ($acao == 'alterar') {
        $where = "nome_dispositivo = ? and id_dispositivo <> ?";
        $params = array('si', $this->DB->escape($nome_dispositivo), $this-
>DB->escape($id));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['nome_dispositivo'] = "Nome já
existente."; }

    if (!empty($invalid_fields)){
        exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
    }

    $arrayInfo = array();
    if ($acao == 'cadastrar'){

        $arrayInfo["nome_dispositivo"] = $nome_dispositivo;
        $arrayInfo["fk_organizacao_unidades"] = $fk_organizacao_unidades;

        $arrayInfo["local_dispositivo"] = ($local_dispositivo) ?
$local_dispositivo : "";
        $arrayInfo["descricao_dispositivo"] = ($descricao_dispositivo) ?
$descricao_dispositivo : "";
        $arrayInfo["nserie_dispositivo"] = ($nserie_dispositivo) ?
$nserie_dispositivo : "";
        $arrayInfo["comunidade_snmp_dispositivo"] =
($comunidade_snmp_dispositivo) ? $comunidade_snmp_dispositivo : "";
        $arrayInfo["fk_dispositivos_marcas"] = ($fk_dispositivos_marcas) ?
$fk_dispositivos_marcas : 0;
        $arrayInfo["fk_dispositivos_modelos"] = ($fk_dispositivos_modelos) ?
$fk_dispositivos_modelos : 0;

        if ($enderecos_ipv4) { natsort($enderecos_ipv4);
$arrayInfo["ipv4_dispositivo"] = implode("|",$enderecos_ipv4); }

```

```

        if ($enderecos_ipv6) { natsort($enderecos_ipv6);
$arrayInfo["ipv6_dispositivo"] = implode("|",$enderecos_ipv6); }

    } else {

        $result = $this->DB->execPreparedSQL("SELECT * FROM dispositivos
WHERE id_dispositivo = ?", array('i', $this->DB->escape($id)));
        $valorAntigo["nome_dispositivo"] = $result[0]['nome_dispositivo'];
        $valorAntigo["local_dispositivo"] = $result[0]['local_dispositivo'];
        $valorAntigo["ipv4_dispositivo"] = $result[0]['ipv4_dispositivo'];
        $valorAntigo["ipv6_dispositivo"] = $result[0]['ipv6_dispositivo'];
        $valorAntigo["descricao_dispositivo"] =
$result[0]['descricao_dispositivo'];
        $valorAntigo["nserie_dispositivo"] =
$result[0]['nserie_dispositivo'];
        $valorAntigo["fk_organizacao_unidades"] =
$result[0]['fk_organizacao_unidades'];
        $valorAntigo["fk_dispositivos_marcas"] =
$result[0]['fk_dispositivos_marcas'];
        $valorAntigo["fk_dispositivos_modelos"] =
$result[0]['fk_dispositivos_modelos'];
        $valorAntigo["comunidade_snmp_dispositivo"] =
$result[0]['comunidade_snmp_dispositivo'];

        $enderecos_ipv4_antigos = ($result[0]["ipv4_dispositivo"]) ?
explode("|", $result[0]["ipv4_dispositivo"]) : array();
        $enderecos_ipv6_antigos = ($result[0]["ipv6_dispositivo"]) ?
explode("|", $result[0]["ipv6_dispositivo"]) : array();

        if (!isset($enderecos_ipv4)) $enderecos_ipv4 = array();
        if (!isset($enderecos_ipv6)) $enderecos_ipv6 = array();

        natsort($enderecos_ipv4);
        natsort($enderecos_ipv6);

        if ($valorAntigo["nome_dispositivo"] != $nome_dispositivo) {
$arrayInfo["nome_dispositivo"] = $nome_dispositivo; }
        if ($valorAntigo["local_dispositivo"] != $local_dispositivo) {
$arrayInfo["local_dispositivo"] = $local_dispositivo; }
        if ($valorAntigo["descricao_dispositivo"] != $descricao_dispositivo)
{ $arrayInfo["descricao_dispositivo"] = $descricao_dispositivo; }
        if ($valorAntigo["nserie_dispositivo"] != $nserie_dispositivo) {
$arrayInfo["nserie_dispositivo"] = $nserie_dispositivo; }
        if ($valorAntigo["fk_organizacao_unidades"] !=
$fk_organizacao_unidades) { $arrayInfo["fk_organizacao_unidades"] =
$fk_organizacao_unidades; }
        if ($valorAntigo["fk_dispositivos_marcas"] !=
$fk_dispositivos_marcas) { $arrayInfo["fk_dispositivos_marcas"] = $fk_dispositivos_marcas;
}
        if ($valorAntigo["fk_dispositivos_modelos"] !=
$fk_dispositivos_modelos) { $arrayInfo["fk_dispositivos_modelos"] =
$fk_dispositivos_modelos; }
        if ($valorAntigo["comunidade_snmp_dispositivo"] !=
$comunidade_snmp_dispositivo) { $arrayInfo["comunidade_snmp_dispositivo"] =
$comunidade_snmp_dispositivo; }

        if (array_diff($enderecos_ipv4, $enderecos_ipv4_antigos) ||
array_diff($enderecos_ipv4_antigos, $enderecos_ipv4)){
            $arrayInfo["ipv4_dispositivo"] = implode("|",$enderecos_ipv4);

```

```

    }

    if (array_diff($enderecos_ipv6, $enderecos_ipv6_antigos) ||
array_diff($enderecos_ipv6_antigos, $enderecos_ipv6)){
        $arrayInfo["ipv6_dispositivo"] = implode("|",$enderecos_ipv6);
    }
}

$log = "";

$params = array();
$posicoes = array();
$update_query = array();

if ($acao == 'cadastrar'){
    $action = "CADASTRO DE DISPOSITIVO";
    $action_text = "cadastrado";

    $params[] = $this->DB->getTypeLabel($arrayInfo);
    $log = "<b>Atributos cadastrados:</b><br />";
    foreach ($arrayInfo as $key => $valor) {
        $log .= $key . ": " . $valor . "<br />";
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("INSERT INTO dispositivos
(`".implode("``,`",$chaves_sql).`) VALUES (".implode(",",$posicoes).")", $params, true);

} elseif ($acao == 'alterar') {
    if (!empty($arrayInfo)) {
        $action = "ALTERAÇÃO DE DISPOSITIVO";
        $action_text = "alterado";

        $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

        $log .= "<b>Dispositivo alterado:<br />
{$valorAntigo["nome_dispositivo"]}<br />";
        $log .= "<b>Atributos alterados: </b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $update_query[] = "`$key` = ?";
            $params[] = $this->DB->escape($valor);
        }
        $params[] = $this->DB->escape($id);
        $result = $this->DB->execPreparedSQL("UPDATE dispositivos SET
".implode(",",$update_query).` WHERE id_dispositivo = ?", $params, true);

    } else {
        exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
    }
} else {
    exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
}
}

```

```

        if (!$this->Logger->logACTION($action, $log))
            return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

        exit(json_encode(array('status' => 'success', 'msg' => "Dispositivo
$action_text com sucesso !")));
    }

    public function delete(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }

        $ids = anti_injection($_REQUEST["ids"]);
        $log = "<b>Dispositivos deletados:</b><br/>";

        $error = false;
        $error_msg = "Alguns dispositivos não foram excluídos pois há monitoramentos
associados:<br/>";
        $params = array();
        $posicoes = array();
        foreach ($ids as $key=>$valor) {

            $result = $this->DB->execPreparedSQL("SELECT nome_mon_disponibilidade
FROM mon_disponibilidade

WHERE fk_dispositivos = ? OR fk_dispositivos_verificador = ?", array("ii", $this-
>DB->escape($valor), $this->DB->escape($valor)));

            if (empty($result)){
                $ids_validos[] = $valor;
            } else {
                $error = true;
                $mon_assoc[] = $result[0]['nome_mon_disponibilidade'];
            }
        }

        if (isset($ids_validos)){
            $params[] = $this->DB->getTypeLabel($ids_validos);
            foreach ($ids_validos as $valor) {
                $posicoes[] = "?";
                $params[] = $this->DB->escape($valor);
            }

            $result = $this->DB->execPreparedSQL("SELECT nome_dispositivo FROM
dispositivos WHERE id_dispositivo IN (".implode(",",$posicoes).")", $params);

            foreach ($result as $res){ $log .= "{$res['nome_dispositivo']}<br/>";
        }

        $result = $this->DB->execPreparedSQL("DELETE FROM dispositivos WHERE
id_dispositivo IN (".implode(",",$posicoes).")", $params, true);

        if (!$this->Logger->logACTION("EXCLUSÃO DE DISPOSITIVO", $log))
            return json_encode(array("status"=>"error","msg"=>"Erro no banco de
dados.<br /> A operação foi efetuada mas não logada!"));
        }
    }

```

```

        if ($error){
            array_unique($mon_assoc);
            foreach ($mon_assoc as $mon) {
                $error_msg .= "- $mon<br/>";
            }
            exit(json_encode(array('status'=>"error", 'msg'=>"$error_msg")));
        } else {
            exit(json_encode(array('status'=>"success", 'msg'=>'Dispositivo(s)
excluído(s) com sucesso!'))));
        }
    }

    public function showInfo(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS)) {
            exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !"))));
        }

        $result = $this->DB->execPreparedSQL("SELECT nome_marca_dispositivo,
nome_modelo_dispositivo, nserie_dispositivo, comunidade_snmp_dispositivo,

        local_dispositivo, descricao_dispositivo, sigla_instituicao, sigla_unidade,
nome_unidade

        dispositivos d

        dispositivos_modelos dmo ON d.`fk_dispositivos_modelos` = dmo.`id_modelo_dispositivo`
        dispositivos_marcas dma ON d.`fk_dispositivos_marcas` = dma.`id_marca_dispositivo`
        organizacao_unidades ou ON d.`fk_organizacao_unidades` = ou.`id_unidade`
        organizacao_instituicoes oi ON ou.`fk_organizacao_instituicoes` = oi.`id_instituicao`

        WHERE id_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["id"])));

        if (empty($result)){
            exit(json_encode(array('status' => 'error')));
        } else {
            foreach ($result as $res) {

                $res["local_dispositivo"] = ($res["local_dispositivo"]) ?
($res["local_dispositivo"]) : "-";
                $res["descricao_dispositivo"] =
($res["descricao_dispositivo"]) ? ($res["descricao_dispositivo"]) : "-";
                $res["nome_marca_dispositivo"] =
($res["nome_marca_dispositivo"]) ? ($res["nome_marca_dispositivo"]) : "-";
                $res["nome_modelo_dispositivo"] =
($res["nome_modelo_dispositivo"]) ? ($res["nome_modelo_dispositivo"]) : "-";
                $res["nserie_dispositivo"] = ($res["nserie_dispositivo"]) ?
($res["nserie_dispositivo"]) : "-";
                $res["comunidade_snmp_dispositivo"] =
($res["comunidade_snmp_dispositivo"]) ? ($res["comunidade_snmp_dispositivo"]) : "-";

                $response = "<div style='display:inline'>";
            }
        }
    }
}

```



```

        $response .= "<strong>Instituição:</strong></h4>";
        "$res["sigla_instituicao"]."<br />";
        $response .= "<strong>Unidade:</strong></h4>";
        "$res["sigla_unidade"]." - "$res["nome_unidade"]."<br />";
        $response .= "<strong>Local:</strong></h4>";
        "$res["local_dispositivo"]."<br />";
        $response .= "<strong>Descrição:</strong></h4>";
        "$res["descricao_dispositivo"]."<br />";
        $response .= "<strong>Marca:</strong></h4>";
        "$res["nome_marca_dispositivo"]."<br />";
        $response .= "<strong>Modelo:</strong></h4>";
        "$res["nome_modelo_dispositivo"]."<br />";
        $response .= "<strong>Nº. Série:</strong></h4>";
        "$res["nserie_dispositivo"]."<br />";
        $response .= "<strong>Comunidade SNMP:</strong></h4>";
        "$res["comunidade_snmp_dispositivo"]."<br />";
        $response .= "</div>";
        $response .= "<div class='clear'></div>";
    }

    exit(json_encode(array("status"=>"success", "info"=>$response)));
}

}

public function getInfo(){
    $result = $this->DB->execPreparedSQL("SELECT * FROM dispositivos WHERE
id_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["id"])));
    if (empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
}

public function getIps(){
    $id_dispositivo = (isset($_REQUEST['select'])) ? $this->DB->
escape($_REQUEST['select']) : ("");
    $ips = "";

    $result = $this->DB->execPreparedSQL("select ipv4_dispositivo,
ipv6_dispositivo from dispositivos WHERE id_dispositivo = ? order by ipv4_dispositivo",
array('i', $this->DB->escape($id_dispositivo)));
    foreach ($result as $res){
        $ips4 = explode("|", $res["ipv4_dispositivo"]);
        $ips6 = explode("|", $res["ipv6_dispositivo"]);
        foreach ($ips4 as $ip){
            $ips .= "<option value='$ip'>$ip</option>";
        }
        foreach ($ips6 as $ip){
            $ips .= "<option value='$ip'>$ip</option>";
        }
    }

    $retorno = array("options" => $ips);
    exit (json_encode($retorno));
}

public function getPortas(){

```

```

        global $config;

        $select_all = (isset($_REQUEST["select_all"]) && $_REQUEST["select_all"] ==
0) ? false : true;
        $tipo = (isset($_REQUEST["tipo"])) ? $_REQUEST["tipo"] : "";

        if (validaIP($_REQUEST["ip"])){
            $ip = $this->DB->escape($_REQUEST["ip"]);
        } else {
            exit(json_encode(array("status"=>"error", "msg"=>"IP do dispositivo
alvo inválido.")));
        }

        $portas_monitoradas = array();
        $verifica_portas = false;
        if ($_REQUEST["id"]){
            $verifica_portas = true;

            if($tipo == "monitoramentos"){
                $result = $this->DB->execPreparedSQL("SELECT param_ports FROM
monitoramentos m

                LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`

                WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"])));
                if (!empty($result)){
                    $portas = $result[0]['param_ports'];
                    $portas_monitoradas = explode("||", $portas);
                } else {
                    exit('<p class="error">ID Inválido.</p>');
                }
            } elseif($tipo == "circuitos") {
                $result = $this->DB->execPreparedSQL("SELECT port_neighbour
FROM circuitos c

                WHERE id_circuito = ?", array("i", $this->DB->escape($_REQUEST["id"])));
                if (!empty($result)){
                    $portas_monitoradas[] = $result[0]['port_neighbour'];
                } else {
                    exit('<p class="error">ID Inválido.</p>');
                }
            }
        }

        $result = $this->DB->execPreparedSQL("SELECT comunidade_snmp_dispositivo FROM
dispositivos d

        WHERE id_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["dispositivo"])));
        if (!empty($result)){ $comunidade =
$result[0]['comunidade_snmp_dispositivo']; } else { exit('<p class="error">ID
Inválido.</p>'); }

        $response = `usr/bin/snmpwalk -t 1 -v 2c -c $comunidade $ip
{$config['ifDescr']}`;
        if ($response != null){
            $response = str_replace(" ", "", trim($response));
            $resp = explode("\n", $response);
            $form = "";
            if ($select_all){

```

```

        $form = "<div class='checkbox_label' style='float:left;'>
                <div class='checkbox' id='check_all_div'>
                    <input type='checkbox'
id='check_all' name='check_all' class='check_all'>
                </div>
                <label for='check_all'>Selecionar Todas</label>
                </div>
                <div class='clear'></div><br/>";
    } else {
        $form = "<label style='margin-left:15px'>Selecione uma
porta.</label><br/><br/>";
    }
    $portas_disponiveis = array();
    foreach($resp as $info){
        if (preg_match("/unrouted|null/i", $info)){
            continue;
        }
        $info = str_replace("${config['ifDescr']}.", "", trim($info));
        $val = explode("=", $info);
        $valor = $val[0];
        preg_match("/:\"(.*)\"/", $val[1], $matches);
        $string = $matches[1];
        $value = "${valor}::${string}";
        $portas_disponiveis[] = $value;
        $checked = "";
        if (in_array($value,$portas_monitoradas)) { $checked =
"checked='checked'"; }
        $form .= "<div class='checkbox_label' style='float:left;'>
                <div class='checkbox' id='$string'>
                    <input type='checkbox' $checked
id='$string' name='ports[]' value='$value'>
                </div>
                <label for='$string'>$string</label>
                </div>";
    }

    $portas_indisponiveis = "";
    if ($verifica_portas){
        $portas_nd =
array_diff($portas_monitoradas,$portas_disponiveis);
        if (!empty($portas_nd)){
            foreach ($portas_nd as $pt){
                $pt = explode(":", $pt);
                $portas_indisponiveis .= "${pt[1]}<br/>";
            }
        }
    }

    exit(json_encode(array("msg"=>"$form", "ports_nd"=>"$portas_indisponiveis")));
    } else {
        exit(json_encode(array("status"=>"error", "msg"=>"Não houve resposta
do dispositivo.")));
    }
}
?>

```

```

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleInstituicao{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleInstituicao();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
     JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_INSTITUCOES)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /*
         * Verifica os dados do parâmetro enviado
         */
        $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
        $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
        $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
        $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
        $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

        $response = new stdClass();

        /*
         * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
         query para filtrar os dados
         */
        $params = array();
        $where = "";
        if(isset($search) && $search != ""){
            $response->filtro = true;
            $search = normalize($search);

```

```

        $where = " WHERE sigla_instituicao LIKE ? OR nome_instituicao LIKE ?
";
        $params = array('ss', $this->DB->escape("%$search%"), $this->DB-
>escape("%$search%"));
    }
    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
organizacao_instituicoes $where", $params);
    /*
    * Caso não ocorrer erros durante a busca, configura o total de páginas dado
o número de dados limite por página
    */
    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }

        $result = $this->DB->execPreparedSQL("SELECT id_instituicao,
sigla_instituicao, nome_instituicao

FROM organizacao_instituicoes $where $order LIMIT $start,$limit", $params);

        $response->page = $page;
        $response->total = $total_pages;
        $response->records = $count;

    /*
    * Configura a tabela na variável response para retornar para JQgrid
da página
    */
    $i = 0;
    foreach ($result as $res){
        $response->rows[$i]['id'] = $res['id_instituicao'];
        $response->rows[$i]['cell'] = array(
            format_title_trim($res['sigla_instituicao'],15),
            format_title_trim($res['nome_instituicao'],40),
            "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"

        );
        $i++;
    }
    }
    exit(json_encode($response));
}

public function add_update(){

    if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_INSTITUICOES)) {

```

```

        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($_REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $value_item) {
                $value_new[] = anti_injection($value_item);
            }
            $$key = array_sum($value_new);
        } else {
            $$key = anti_injection($value);
        }
    }

    if (!$sigla_instituicao){
        $invalid_fields['sigla_instituicao'] = "Sigla inválida.";
    }

    if (!$nome_instituicao){
        $invalid_fields['nome_instituicao'] = "Nome inválido.";
    }

    if ($acao == 'cadastrar'){
        $where = "sigla_instituicao = ?";
        $params = array('s', $this->DB->escape("$sigla_instituicao"));
    } elseif ($acao == 'alterar') {
        $where = "sigla_instituicao = ? and id_instituicao <> ?";
        $params = array('si', $this->DB->escape("$sigla_instituicao"), $this->DB->escape("$id"));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
    organizacao_instituicoes WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['sigla_instituicao'] = "Sigla já
    existente."; }

    if (!empty($invalid_fields)){
        exit(json_encode(array('status' => 'error', 'errors' =>
    $invalid_fields)));
    }

    $arrayInfo = array();
    if ($acao == 'cadastrar'){
        $arrayInfo["sigla_instituicao"] = $sigla_instituicao;
        $arrayInfo["nome_instituicao"] = $nome_instituicao;
    } else {
        $result = $this->DB->execPreparedSQL("SELECT sigla_instituicao,
    nome_instituicao FROM organizacao_instituicoes WHERE id_instituicao = ?", array('i', $this->DB->escape("$id")));
        $valorAntigo["sigla_instituicao"] = $result[0]['sigla_instituicao'];
        $valorAntigo["nome_instituicao"] = $result[0]['nome_instituicao'];

        if ($valorAntigo["sigla_instituicao"] != $sigla_instituicao) {
            $arrayInfo["sigla_instituicao"] = $sigla_instituicao; }
        if ($valorAntigo["nome_instituicao"] != $nome_instituicao) {
            $arrayInfo["nome_instituicao"] = $nome_instituicao; }
    }

```

```

    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE INSTITUIÇÃO";
        $action_text = "cadastrada";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO
organizacao_instituicoes (`".implode("`", $chaves_sql)."` ) VALUES
("".implode("`,`", $posicoes)."`)", $params, true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo)) {
            $action = "ALTERAÇÃO DE INSTITUIÇÃO";
            $action_text = "alterada";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

            $log .= "<b>Instituição alterada:<br/>
{$valorAntigo["sigla_instituicao"]}<br />";
            $log .= "<b>Atributos alterados: </b><br />";
            foreach ($arrayInfo as $key => $valor) {
                $log .= $key . ": " . $valor . "<br />";
                $update_query[] = "`$key` = ?";
                $params[] = $this->DB->escape($valor);
            }
            $params[] = $this->DB->escape($id);
            $result = $this->DB->execPreparedSQL("UPDATE
organizacao_instituicoes SET ".implode(" ", $update_query)." WHERE id_instituicao = ?",
$params, true);

        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

```

```

        exit(json_encode(array('status' => 'success', 'msg' => "Instituição
$action_text com sucesso !"))));
    }

    public function delete(){

        if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_INSTITUICOES)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }

        $ids = anti_injection($_REQUEST["ids"]);
        $log = "<b>Instituições deletadas:</b><br/>";

        $params = array();
        $posicoes = array();
        $params[] = $this->DB->getTypeLabel($ids);
        foreach ($ids as $key=>$valor) {
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("SELECT sigla_instituicao FROM
organizacao_instituicoes WHERE id_instituicao IN (".implode(",",$posicoes).")", $params);

        foreach ($result as $res){ $log .= "{$res['sigla_instituicao']}<br/>"; }

        $result = $this->DB->execPreparedSQL("DELETE FROM organizacao_instituicoes
WHERE id_instituicao IN (".implode(",",$posicoes).")", $params, true);

        if (!$this->Logger->logACTION("EXCLUSÃO DE INSTITUIÇÃO", $log))
            return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

        exit(json_encode(array('status'=>'success','msg'=>'Instituição excluída com
sucesso!'))));
    }

    public function getInfo(){

        $result = $this->DB->execPreparedSQL("SELECT * FROM organizacao_instituicoes
WHERE id_instituicao = ?", array("i", $this->DB->escape($_REQUEST["id"])));
        if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }

    }

}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleManutencao{
    protected $db;
    private static $instance;

```



```

/**
 * Método responsável por retornar uma instância da classe
 */
public static function getInstance() {
    if (!self::$instance) {
        self::$instance = new ControleManutencao();
    }
    return self::$instance;
}

/**
 * Construtor da classe
 */
public function __construct(){
    $this->DB = DB::getInstance();
    $this->Logger = Logger::getInstance();
    $this->auth = new Auth();
}

/**
 * Método responsável pela criação da tabela de usuários para montagem de uma
JQgrid.
 */
public function getTableData(){

    if (!$this->auth->check(ID_SUBMODULO_MANUTENCOES_REGISTRAR)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }
    /*
     * Verifica os dados do parâmetro enviado
     */
    $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
    $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
    $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
    $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
    $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

    $response = new stdClass();

    /*
     * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
query para filtrar os dados
     */
    $params = array();
    $where = "";
    if(isset($search) && $search != ""){
        $response->filtro = true;
        $search = normalize($search);
        $where = " WHERE nome_dispositivo LIKE ?";
        $params = array('sis',
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
manutencoes m

```

```

JOIN dispositivos d ON m.`fk_dispositivos`=d.`id_dispositivo`

$where", $params);

    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }

        $result = $this->DB->execPreparedSQL("SELECT id_manutencao,
nome_dispositivo, start_time, end_time

FROM manutencoes m

JOIN dispositivos d ON m.`fk_dispositivos`=d.`id_dispositivo`

$where $order LIMIT $start,$limit", $params);

        $response->page = $page;
        $response->total = $total_pages;
        $response->records = $count;

        $i = 0;
        foreach ($result as $res){

            $start_time = date("d-m-Y H:i", $res['start_time']);
            $end_time = date("d-m-Y H:i", $res['end_time']);

            $response->rows[$i]['id'] = $res['id_manutencao'];
            $response->rows[$i]['cell'] = array(
                format_title_trim($res['nome_dispositivo'],30),
                format_title_trim($start_time,17),
                format_title_trim($end_time,17),
                "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"
            );
            $i++;
        }
    }
    exit(json_encode($response));
}

public function add_update(){

    if (!$this->auth->check(ID_SUBMODULO_MANUTENCOES_REGISTRAR)) {

```

```

        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($_REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $chave=>$value_item) {
                if ($value_item != ""){
                    $value_new[$chave] =
anti_injection($value_item);
                }
            }
            $$key = $value_new;
        } else {
            $$key = anti_injection($value);
        }
    }

    if (!$fk_dispositivos){
        $invalid_fields['fk_dispositivos'] = "Dispositivo alvo inválido.";
    }

    if (!$inicio_data_manutencao || !validaData($inicio_data_manutencao)){
        $invalid_fields['inicio_data_manutencao'] = "Data de início
inválida.";
    }

    if (!$inicio_hora_manutencao || !validaTime($inicio_hora_manutencao)){
        $invalid_fields['inicio_hora_manutencao'] = "Hora de início
inválida.";
    }

    if (!$fim_data_manutencao || !validaData($fim_data_manutencao)){
        $invalid_fields['fim_data_manutencao'] = "Data de fim inválida.";
    }

    if (!$fim_hora_manutencao || !validaTime($fim_hora_manutencao)){
        $invalid_fields['fim_hora_manutencao'] = "Hora de fim inválida.";
    }

    if ($inicio_data_manutencao && $inicio_hora_manutencao &&
    $fim_data_manutencao && $fim_hora_manutencao){
        $inicio_data_manutencao = str_replace("/", "-",
    $inicio_data_manutencao);
        $fim_data_manutencao = str_replace("/", "-", $fim_data_manutencao);

        $tempo_inicio = strtotime("$inicio_data_manutencao
    $inicio_hora_manutencao");
        $tempo_fim = strtotime("$fim_data_manutencao $fim_hora_manutencao");

        if ($tempo_inicio > $tempo_fim){
            $invalid_fields['fim_data_manutencao'] = "";
            $invalid_fields['fim_hora_manutencao'] = "Data e hora finais
    devem ser maiores que as iniciais.";
        }
    }
}

```

```

        if (!$ip_dispositivo){
            $invalid_fields['ip_dispositivo'] = "IP do dispositivo alvo
inválido.";
        }

        if (empty($ports)){
            $invalid_fields['portas_field'] = "É necessário selecionar pelo menos
uma porta.";
        }

        if (!$descricao_manutencao){
            $invalid_fields['descricao_manutencao'] = "Descrição inválida.";
        }

        if (!empty($invalid_fields)){
            exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
        }

        $arrayInfo = array();
        $arrayInfoExtra = array();
        if ($acao == 'cadastrar'){

            $arrayInfo["fk_dispositivos"] = $fk_dispositivos;
            $arrayInfo["ip_dispositivo"] = $ip_dispositivo;
            $arrayInfo["descricao_manutencao"] = $descricao_manutencao;
            $arrayInfo["start_time"] = $tempo_inicio;
            $arrayInfo["end_time"] = $tempo_fim;

            if (isset($ports)){
                $arrayInfo["param_ports"] = "";
                natsort($ports);
                foreach ($ports as $port){
                    $arrayInfo["param_ports"] .= "{$port}||";
                }
                $arrayInfo["param_ports"] = substr($arrayInfo["param_ports"],
0, -2);
            }

        } else {

            $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m
LEFT JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`
LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`
WHERE id_mon = ?", array('i', $this->DB->escape($id)));

            $valorAntigo["nome_mon"] = $result[0]['nome_mon'];
            $valorAntigo["fk_mon_tipos"] = $result[0]['fk_mon_tipos'];
            $valorAntigo["descricao_mon"] = $result[0]["descricao_mon"];
            $valorAntigo["fk_dispositivos"] = $result[0]['fk_dispositivos'];
            $valorAntigo["fk_dispositivos_verificador"] =
$result[0]['fk_dispositivos_verificador'];
            $valorAntigo["ip_dispositivo_alvo"] =
$result[0]['ip_dispositivo_alvo'];

```

```

                $valorAntigo["ping_count"] = isset($result[0]['ping_count']) ?
$result[0]['ping_count'] : "";
                $valorAntigo["ping_size"] = isset($result[0]['ping_size']) ?
$result[0]['ping_size'] : "";

                if ($valorAntigo["nome_mon"] != $nome_mon) { $arrayInfo["nome_mon"] =
$nome_mon; }

                if ($valorAntigo["fk_mon_tipos"] != $fk_mon_tipos) {
                    exit(json_encode(array('status' => 'error', 'msg' => 'Tipo de
monitoramento não pode ser alterado!')));
                }

                if ($valorAntigo["descricao_mon"] != $descricao_mon) {
$arrayInfo["descricao_mon"] = $descricao_mon; }
                if ($valorAntigo["fk_dispositivos"] != $fk_dispositivos) {
$arrayInfo["fk_dispositivos"] = $fk_dispositivos; }
                if ($valorAntigo["fk_dispositivos_verificador"] !=
$fk_dispositivos_verificador) { $arrayInfo["fk_dispositivos_verificador"] =
$fk_dispositivos_verificador; }
                if ($valorAntigo["ip_dispositivo_alvo"] != $ip_dispositivo_alvo) {
$arrayInfo["ip_dispositivo_alvo"] = $ip_dispositivo_alvo; }

                if (!isset($ping_count)) { $ping_count = 0; }
                if (!isset($ping_size)) { $ping_size = 0; }

                $portas_antigas = isset($result[0]["param_ports"]) ? explode("||",
$result[0]["param_ports"]) : array();

                if (!isset($ports)) { $ports = array(); } else { natsort($ports); }

                if ($fk_mon_tipos == 1){
                    if ($valorAntigo["ping_count"] != $ping_count) {
$arrayInfoExtra["ping_count"] = $ping_count; }
                    if ($valorAntigo["ping_size"] != $ping_size) {
$arrayInfoExtra["ping_size"] = $ping_size; }
                } else if ($fk_mon_tipos == 2){
                    if (array_diff($portas_antigas, $ports) || array_diff($ports,
$portas_antigas)){
                        $arrayInfoExtra["param_ports"] = "";
                        foreach ($ports as $port){
                            $arrayInfoExtra["param_ports"] .= "{$port}||";
                        }
                        $arrayInfoExtra["param_ports"] =
substr($arrayInfoExtra["param_ports"], 0, -2);
                    }
                }
            }

            $log = "";

            $params = array();
            $posicoes = array();
            $update_query = array();

            if ($acao == 'cadastrar'){

                $action = "CADASTRO DE MANUTENÇÃO";

```

```

$action_text = "cadastrada";

$params[] = $this->DB->getTypeLabel($arrayInfo);
$log = "<b>Atributos cadastrados:</b><br />";
foreach ($arrayInfo as $key => $valor) {
    $log .= $key . ": " . $valor . "<br />";
    $chaves_sql[] = $key;
    $posicoes[] = "?";
    $params[] = $this->DB->escape($valor);
}

$result = $this->DB->execPreparedSQL("INSERT INTO manutencoes
(`".implode("`,"",$chaves_sql).`) VALUES (".implode("",$posicoes).")", $params, true);
} elseif ($acao == 'alterar') {

    if (!empty($arrayInfo) || !empty($arrayInfoExtra)) {

        $action = "ALTERAÇÃO DE MONITORAMENTO";
        $action_text = "alterado";

        $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

        $log .= "<b>Monitoramento alterado:<br/>
{$valorAntigo["nome_mon"]}<br />";
        $log .= "<b>Atributos alterados: </b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $update_query[] = "`$key` = ?";
            $params[] = $this->DB->escape($valor);
        }

        $params[] = $this->DB->escape($id);
        $this->DB->autocommit(false);

        if (!empty($update_query)){
            $result = $this->DB->execPreparedSQL("UPDATE
monitoramentos SET ".implode("",$update_query)." WHERE id_mon = ?", $params, true);
        }

        if (!empty($arrayInfoExtra)){
            $paramsX[] = $this->DB->getTypeLabel($arrayInfoExtra,
$update = true);

            foreach ($arrayInfoExtra as $keyX => $valorX) {
                $log .= $keyX . ": " . $valorX . "<br />";
                $update_queryX[] = "`$keyX` = ?";
                $paramsX[] = $this->DB->escape($valorX);
            }
            $paramsX[] = $this->DB->escape($id);
            $result = $this->DB->execPreparedSQL("UPDATE $table SET
".implode("",$update_queryX)." WHERE fk_monitoramentos = ?", $paramsX, true);
        }

        $this->DB->commit();
        $this->DB->autocommit(true);

    } else {
        exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
    }
}

```

```

    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!'))));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Monitoramento
$action_text com sucesso !"))));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_MANUTENCOES_REGISTRAR)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Monitoramentos deletados:</b><br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("SELECT * FROM manutencoes WHERE
id_manutencao IN (".implode(",",$posicoes).")", $params);

    foreach ($result as $res){ $log .= "{$res['descricao_manutencao']}<br/>"; }

    $result = $this->DB->execPreparedSQL("DELETE FROM manutencoes WHERE
id_manutencao IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE MANUTENÇÃO", $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status'=>'success', 'msg'=>'Manutenção excluída com
sucesso!'))));
}

public function showInfo(){

    if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
        exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !"))));
    }

    $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m
LEFT
JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`

```

```

JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`
WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"]));

if (empty($result)){
    exit(json_encode(array('status' => 'error')));
} else {
    foreach ($result as $res) {

        $res["nome_mon"] = ($res["nome_mon"]) ? ($res["nome_mon"]) :
"-";

        $response = "<div style='display:inline'>";
        $response .= "<strong>Nome:</strong></h4>";
"$res["nome_mon"]."<br />";
        $response .= "</div>";
        $response .= "<div class='clear'></div>";
    }

    exit(json_encode(array("status"=>"success", "info"=>$response)));
}

}

public function getInfo(){
    $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m
LEFT JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`
LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`

WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"]));
    if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
}

public function getPortas(){
    global $config;

    if (validaIP($_REQUEST["ip"])){
        $ip = $this->DB->escape($_REQUEST["ip"]);
    } else {
        exit(json_encode(array("status"=>"error", "msg"=>"IP do dispositivo
alvo inválido.")));
    }

    $portas_monitoradas = array();
    $verifica_portas = false;
    if ($_REQUEST["id"]){
        $verifica_portas = true;
        $result = $this->DB->execPreparedSQL("SELECT param_ports FROM
monitoramentos m

LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`

WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"]));

```



```

        if (!empty($result)){
            $portas = $result[0]['param_ports'];
            $portas_monitoradas = explode("||", $portas);
        } else {
            exit('<p class="error">ID Inválido.</p>');
        }
    }

    $result = $this->DB->execPreparedSQL("SELECT comunidade_snmp_dispositivo FROM
dispositivos d

    WHERE id_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["dispositivo"]));
    if (!empty($result)){ $comunidade =
$result[0]['comunidade_snmp_dispositivo']; } else { exit('<p class="error">ID
Inválido.</p>'); }

    //echo "/usr/bin/snmpwalk -t 3 -v 2 -c $comunidade $ip {$config['ifDescr']}";
    $response = `/usr/bin/snmpwalk -t 1 -v 2c -c $comunidade $ip
{$config['ifDescr']}`;
    if ($response != null){
        $response = str_replace(" ", "", trim($response));
        $resp = explode("\n", $response);
        $form = "";
        $form = "<div class='checkbox_label' style='float:left;'>
id='check_all_div'>
                                <div class='checkbox'
id='check_all' name='check_all' class='check_all'>
                                    <input type='checkbox'
                                        </div>
                                        <label for='check_all'>Selecionar
Todas</label>
                                </div>
                                <div class='clear'></div><br/>";

        $portas_disponiveis = array();
        foreach($resp as $info){
            $info = str_replace("{$config['ifDescr']}.", "", trim($info));
            $val = explode("=", $info);
            $valor = $val[0];
            preg_match("/:\\"(.*)\"/", $val[1], $matches);
            $string = $matches[1];
            $value = "{$valor}::{$string}";
            $portas_disponiveis[] = $value;
            $checked = "";
            if (in_array($value, $portas_monitoradas)) { $checked =
"checked='checked'"; }

            $form .= "<div class='checkbox_label' style='float:left;'>
                                <div class='checkbox' id='$string'>
                                    <input type='checkbox' $checked
id='$string' name='ports[]' value='$value'>
                                </div>
                                <label for='$string'>$string</label>
                                </div>";
        }

        $portas_indisponiveis = "";
        if ($verifica_portas){
            $portas_nd =
array_diff($portas_monitoradas, $portas_disponiveis);

```

```

        if (!empty($portas_nd)){
            foreach ($portas_nd as $pt){
                $pt = explode("::", $pt);
                $portas_indisponiveis .= "{$pt[1]}<br/>";
            }
        }
    }

    exit(json_encode(array("msg"=>"$form", "ports_nd"=>"$portas_indisponiveis")));
} else {
    exit(json_encode(array("status"=>"error", "msg"=>"Não houve resposta
do dispositivo.")));
}
}
}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleMarca{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleMarca();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
    JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MARCAS)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /**
         * Verifica os dados do parâmetro enviado
         */
    }
}

```

```

$page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
$limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
$idx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
$sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
$search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

$response = new stdClass();
/*
 * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
query para filtrar os dados
 */
$params = array();
$where = "";
if(isset($search) && $search != ""){
    $response->filtro = true;
    $search = normalize($search);
    $where = " WHERE nome_marca_dispositivo LIKE ? ";
    $params = array('s', $this->DB->escape("%$search%"));
}
$result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos_marcas $where", $params);

/*
 * Caso não ocorrer erros durante a busca, configura o total de páginas dado
o número de dados limite por página
 */
if(!empty($result)){
    $count = $result[0]['count'];
    if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

    if ($page > $total_pages){ $page = $total_pages; }
    if ($limit < 0){ $limit = 0; }
    $start = $limit * $page - $limit;
    if ($start < 0){ $start = 0; }

    $order = "";
    if(isset($idx) && $idx!=""){
        $order = ($sord=='asc')?"ASC":"DESC";
        $order = "ORDER BY $idx $order";
    }

    $result = $this->DB->execPreparedSQL("SELECT id_marca_dispositivo,
nome_marca_dispositivo
FROM dispositivos_marcas $where $order LIMIT $start,$limit", $params);

    $response->page = $page;
    $response->total = $total_pages;
    $response->records = $count;

/*
 * Configura a tabela na variável response para retornar para JQgrid
da página
 */
$i = 0;
foreach ($result as $res){
    $response->rows[$i]['id'] = $res['id_marca_dispositivo'];
}

```

```

        $response->rows[$i]['cell'] = array(
            format_title_trim($res['nome_marca_dispositivo'],40),
            "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"
        );
        $i++;
    }
}
exit(json_encode($response));
}

public function add_update(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MARCAS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $value_item) {
                $value_new[] = anti_injection($value_item);
            }
            $$key = array_sum($value_new);
        } else {
            $$key = anti_injection($value);
        }
    }

    $nome_marca_dispositivo = mb_strtoupper($nome_marca_dispositivo);

    if (!$nome_marca_dispositivo){
        $invalid_fields['nome_marca_dispositivo'] = "Nome Inválido.";
    }

    if ($acao == 'cadastrar'){
        $where = "nome_marca_dispositivo = ?";
        $params = array('s', $this->DB->escape("$nome_marca_dispositivo"));
    } elseif ($acao == 'alterar') {
        $where = "nome_marca_dispositivo = ? and id_marca_dispositivo <> ?";
        $params = array('si', $this->DB->escape("$nome_marca_dispositivo"),
$this->DB->escape($id));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos_marcas WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['nome_marca_dispositivo'] =
"Nome já existente."; }

    if (!empty($invalid_fields)){
        exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
    }

    $arrayInfo = array();
    if ($acao == 'cadastrar'){

```

```

        $arrayInfo["nome_marca_dispositivo"] = $nome_marca_dispositivo;
    } else {
        $result = $this->DB->execPreparedSQL("SELECT nome_marca_dispositivo
FROM dispositivos_marcas WHERE id_marca_dispositivo = ?", array('i', $this->DB-
>escape($id)));
        $valorAntigo["nome_marca_dispositivo"] =
$result[0]['nome_marca_dispositivo'];

        if ($valorAntigo["nome_marca_dispositivo"] !=
$nome_marca_dispositivo) { $arrayInfo["nome_marca_dispositivo"] = $nome_marca_dispositivo;
}
    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE MARCA";
        $action_text = "cadastrada";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO dispositivos_marcas
(`".implode("`",`",$chaves_sql).`) VALUES ("`.implode("`",`",$posicoes).`)", $params, true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo)) {
            $action = "ALTERAÇÃO DE MARCA";
            $action_text = "alterada";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

            $log .= "<b>Marca alterada:<br/>
{$valorAntigo["nome_marca_dispositivo"]}<br />";
            $log .= "<b>Atributos alterados: </b><br />";
            foreach ($arrayInfo as $key => $valor) {
                $log .= $key . ": " . $valor . "<br />";
                $update_query[] = "`$key` = ?";
                $params[] = $this->DB->escape($valor);
            }
            $params[] = $this->DB->escape($id);
            $result = $this->DB->execPreparedSQL("UPDATE
dispositivos_marcas SET ".implode("`",`",$update_query).` WHERE id_marca_dispositivo = ?",
$params, true);

        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    }
}

```

```

    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!'))));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Marca $action_text
com sucesso !"))));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MARCAS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Marcas deletadas:</b><br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("SELECT nome_marca_dispositivo FROM
dispositivos_marcas WHERE id_marca_dispositivo IN (".implode(",",$posicoes).")", $params);

    foreach ($result as $res){ $log .= "{$res['nome_marca_dispositivo']}<br/>"; }

    $result = $this->DB->execPreparedSQL("DELETE FROM dispositivos_marcas WHERE
id_marca_dispositivo IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE MARCA", $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status'=>'success','msg'=>'Marca(s) excluída(s) com
sucesso!'))));
}

public function getInfo(){

    $result = $this->DB->execPreparedSQL("SELECT * FROM dispositivos_marcas WHERE
id_marca_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["id"])));
    if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }

}

public function getMarcas(){

```

```

        $select = (isset($_REQUEST['select'])) ? $this->DB->escape($_REQUEST['select']) : ("");

        $marcas = "";

        $result = $this->DB->execPreparedSQL("SELECT id_marca_dispositivo,
nome_marca_dispositivo FROM dispositivos_marcas order by nome_marca_dispositivo", array());
        if (!empty($result)){
            foreach ($result as $res){
                if ($select == $res['id_marca_dispositivo']){
                    $marcas .= "<option selected='selected'
value='{$res['id_marca_dispositivo']}'>".$res['nome_marca_dispositivo']."</option>";
                } else {
                    $marcas .= "<option
value='{$res['id_marca_dispositivo']}'>".$res['nome_marca_dispositivo']."</option>";
                }
            }
        } else {
            $marcas .= "<option value=''>Nenhuma marca cadastrada</option>";
        }

        $retorno = array("marcas" => $marcas);
        exit (json_encode($retorno));
    }
}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleModelo{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleModelo();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
     JQgrid.
     */

```

```

public function getTableData(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MODELOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }
    /*
     * Verifica os dados do parâmetro enviado
     */
    $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
    $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
    $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
    $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
    $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

    $response = new stdClass();

    /*
     * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
     query para filtrar os dados
     */
    $params = array();
    $where = "";
    if(isset($search) && $search != ""){
        $response->filtro = true;
        $search = normalize($search);
        $where = " WHERE nome_marca_dispositivo LIKE ? OR
nome_modelo_dispositivo LIKE ? ";
        $params = array('ssss', $this->DB->escape("%$search%"), $this->DB-
>escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos_modelos dmo
dispositivos_marcas dma
dmo.`fk_dispositivos_marcas`=dma.`id_marca_dispositivo` $where", $params);

    /*
     * Caso não ocorrer erros durante a busca, configura o total de páginas dado
     o número de dados limite por página
     */
    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }
    }
}

```



```

        $result = $this->DB->execPreparedSQL("SELECT id_modelo_dispositivo,
nome_marca_dispositivo, nome_modelo_dispositivo FROM dispositivos_modelos dmo

JOIN dispositivos_marcas dma
ON
dmo.`fk_dispositivos_marcas`=dma.`id_marca_dispositivo` $where $order LIMIT $start,$limit",
$params);

$response->page = $page;
$response->total = $total_pages;
$response->records = $count;

/*
 * Configura a tabela na variável response para retornar para JQgrid
da página
 */
$i = 0;
foreach ($result as $res){
    $response->rows[$i]['id'] = $res['id_modelo_dispositivo'];
    $response->rows[$i]['cell'] = array(
        format_title_trim($res['nome_marca_dispositivo'],15),
        format_title_trim($res['nome_modelo_dispositivo'],20),
        "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"
    );
    $i++;
}
}
exit(json_encode($response));
}

public function add_update(){
    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MODELOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($_REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $value_item) {
                $value_new[] = anti_injection($value_item);
            }
            $$key = array_sum($value_new);
        } else {
            $$key = anti_injection($value);
        }
    }

    $nome_modelo_dispositivo = mb_strtoupper($nome_modelo_dispositivo);

    if (!$fk_dispositivos_marcas){
        $invalid_fields['fk_dispositivos_marcas'] = "Marca inválida.";
    }

    if (!$nome_modelo_dispositivo){

```

```

        $invalid_fields['nome_modelo_dispositivo'] = "Nome inválido.";
    }

    if ($acao == 'cadastrar'){
        $where = "nome_modelo_dispositivo = ?";
        $params = array('s', $this->DB->escape($nome_modelo_dispositivo));
    } elseif ($acao == 'alterar') {
        $where = "nome_modelo_dispositivo = ? and id_modelo_dispositivo <>
?";
        $params = array('si', $this->DB->escape($nome_modelo_dispositivo),
$this->DB->escape($id));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
dispositivos_modelos WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['nome_modelo_dispositivo'] =
"Nome já existente."; }

    if (!empty($invalid_fields)){
        exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
    }

    $arrayInfo = array();
    if ($acao == 'cadastrar'){
        $arrayInfo["nome_modelo_dispositivo"] = $nome_modelo_dispositivo;
        $arrayInfo["fk_dispositivos_marcas"] = $fk_dispositivos_marcas;
    } else {
        $result = $this->DB->execPreparedSQL("SELECT * FROM
dispositivos_modelos WHERE id_modelo_dispositivo = ?", array('i', $this->DB->escape($id)));
        $valorAntigo["nome_modelo_dispositivo"] =
$result[0]['nome_modelo_dispositivo'];
        $valorAntigo["fk_dispositivos_marcas"] =
$result[0]['fk_dispositivos_marcas'];

        if ($valorAntigo["nome_modelo_dispositivo"] !=
$nome_modelo_dispositivo) { $arrayInfo["nome_modelo_dispositivo"] =
$nome_modelo_dispositivo; }
        if ($valorAntigo["fk_dispositivos_marcas"] !=
$fk_dispositivos_marcas) { $arrayInfo["fk_dispositivos_marcas"] = $fk_dispositivos_marcas;
}
    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE MODELO";
        $action_text = "cadastrado";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
        }
    }

```

```

        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("INSERT INTO
dispositivos_modelos (`".implode("`", $chaves_sql)."`") VALUES
("".implode("",$posicoes)."$", $params, true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo)) {
            $action = "ALTERAÇÃO DE MODELO";
            $action_text = "alterado";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

            $log .= "<b>Modelo alterado:<br/>
{$valorAntigo["nome_modelo_dispositivo"]}<br />";
            $log .= "<b>Atributos alterados: </b><br />";
            foreach ($arrayInfo as $key => $valor) {
                $log .= $key . ": " . $valor . "<br />";
                $update_query[] = "`$key` = ?";
                $params[] = $this->DB->escape($valor);
            }
            $params[] = $this->DB->escape($id);
            $result = $this->DB->execPreparedSQL("UPDATE
dispositivos_modelos SET ".implode(" ", $update_query)." WHERE id_modelo_dispositivo = ?",
$params, true);

        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
    }

    if (!$this->Logger->logACTION($action, $log))
return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Modelo $action_text
com sucesso !")));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_MODELOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Modelos deletados:</b><br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
    }
}

```

```

        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("SELECT nome_modelo_dispositivo FROM
dispositivos_modelos WHERE id_modelo_dispositivo IN (".implode(",",$posicoes).")",
$params);

    foreach ($result as $res){ $log .= "{$res['nome_modelo_dispositivo']}<br/>";
}

    $result = $this->DB->execPreparedSQL("DELETE FROM dispositivos_modelos WHERE
id_modelo_dispositivo IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE MODELO", $log))
return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status'=>'success','msg'=>'Modelo(s) excluído(s) com
sucesso!')));
    break;
}

    public function getInfo(){

        $result = $this->DB->execPreparedSQL("SELECT * FROM dispositivos_modelos
WHERE id_modelo_dispositivo = ?", array("i", $this->DB->escape($_REQUEST["id"])));
        if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }

    }

    public function getModelos(){

        $id_marca = (isset($_REQUEST['select'])) ? $this->DB-
>escape($_REQUEST['select']) : ("");

        $modelos = "<option value=''></option>";
        $result = $this->DB->execPreparedSQL("SELECT id_modelo_dispositivo,
nome_modelo_dispositivo FROM dispositivos_modelos WHERE fk_dispositivos_marcas = ? order by
nome_modelo_dispositivo", array("i", $id_marca));
        foreach ($result as $res){
            $modelos .= "<option
value='{$res['id_modelo_dispositivo']}'>".$res['nome_modelo_dispositivo']."</option>";
        }

        $retorno = array("options" => $modelos);
        exit (json_encode($retorno));
    }
}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleMonitoramento{
    protected $db;
    private static $instance;

```

```

/**
 * Método responsável por retornar uma instância da classe
 */
public static function getInstance() {
    if (!self::$instance) {
        self::$instance = new ControleMonitoramento();
    }
    return self::$instance;
}

/**
 * Construtor da classe
 */
public function __construct(){
    $this->DB = DB::getInstance();
    $this->Logger = Logger::getInstance();
    $this->auth = new Auth();
}

/**
 * Método responsável pela criação da tabela de usuários para montagem de uma
JQgrid.
 */
public function getTableData(){

    if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }
    /*
     * Verifica os dados do parâmetro enviado
     */
    $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
    $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
    $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
    $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
    $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

    $response = new stdClass();

    /*
     * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
query para filtrar os dados
     */
    $params = array();
    $where = "";
    if(isset($search) && $search != ""){
        $response->filtro = true;
        $search = normalize($search);
        $where = " WHERE nome_acordo LIKE ?
                                OR d.nome_dispositivo LIKE ?
                                OR nome_tipo_acordo LIKE ?
                                OR descricao_mon_tipo LIKE ?
                                OR ip_disp_alvo LIKE ?";

        $params = array('sssss',
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"),

```

```

        $this->DB->escape("%$search%"),
        $this->DB->escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT COUNT(*) AS count FROM
monitoramentos m
JOIN acordos a ON m.`fk_acordos`=a.`id_acordo`
JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo`
JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo`
JOIN dispositivos d ON m.`fk_dispositivos`=d.`id_dispositivo`
JOIN dispositivos d2 ON m.`fk_dispositivos_verificador`=d2.`id_dispositivo` $where",
$params);

    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }

        $result = $this->DB->execPreparedSQL("SELECT id_mon, nome_acordo,
descricao_mon_tipo, nome_tipo_acordo,
d.nome_dispositivo, d2.nome_dispositivo as nome_dispositivo_verificador,
ip_disp_alvo
FROM monitoramentos m
LEFT JOIN acordos a ON m.`fk_acordos`=a.`id_acordo`
LEFT JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo`
LEFT JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo`
LEFT JOIN dispositivos d ON m.`fk_dispositivos`=d.`id_dispositivo`
LEFT JOIN dispositivos d2 ON m.`fk_dispositivos_verificador`=d2.`id_dispositivo`
$where $order LIMIT $start,$limit", $params);

        $response->page = $page;
        $response->total = $total_pages;
        $response->records = $count;

        $i = 0;
        foreach ($result as $res){

```

```

        $response->rows[$i]['id'] = $res['id_mon'];
        $response->rows[$i]['cell'] = array(
            format_title_trim($res['nome_acordo'],35),
            format_title_trim($res['nome_tipo_acordo'],20),
            format_title_trim($res['descricao_mon_tipo'],20),
            format_title_trim($res['nome_dispositivo'],20),
            format_title_trim($res['ip_disp_alvo'],15),
            "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"
        );
        $i++;
    }
}
exit(json_encode($response));
}

public function add_update(){
    if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $invalid_fields = array();

    foreach ($_REQUEST as $key => $value) {
        if (is_array($value)) {
            $value_new = array();
            foreach ($value as $chave=>$value_item) {
                if ($value_item != ""){
                    $value_new[$chave] =
anti_injection($value_item);
                }
            }
            $$key = $value_new;
        } else {
            $$key = anti_injection($value);
        }
    }

    if (!$ip_disp_alvo){
        $invalid_fields['ip_disp_alvo'] = "IP do dispositivo alvo inválido.";
    }

    if ($acao == "alterar"){
        $fk_acordos = $fk_acordos_altera;
    }

    if (!$fk_acordos){
        $invalid_fields['fk_acordos'] = "Acordo inválido.";
    } else {
        $tmp_fk_acordos = explode("###", $fk_acordos);
        $fk_mon_tipos = $tmp_fk_acordos[1];
        if ($fk_mon_tipos == 1){
            if ($ping_count != "" && !is_numeric($ping_count)){
                $invalid_fields['ping_count'] = "Quantidade de pacotes
inválida.";
            }
        }
    }
}

```

```

    }
    if ($ping_size != "" && !is_numeric($ping_size)){
        $invalid_fields['ping_size'] = "Tamanho do pacote
inválido.";
    }
} elseif ($fk_mon_tipos == 2){
    if (empty($ports)){
        $invalid_fields['portas_field'] = "É necessário
selecionar pelo menos uma porta.";
    }
} elseif ($fk_mon_tipos == 3){
    if (empty($ip_neighbour)){
        $invalid_fields['ip_neighbour'] = "É necessário definir
o IP do neighbour.";
    } else if (!validaIP($ip_neighbour)) {
        $invalid_fields['ip_neighbour'] = "IP do neighbour
inválido.";
    }
}
}
}

if (!$fk_dispositivos){
    $invalid_fields['fk_dispositivos'] = "Dispositivo alvo inválido.";
}

if (!$fk_dispositivos_verificador){
    $invalid_fields['fk_dispositivos_verificador'] = "Dispositivo
verificador inválido.";
}

if ($fk_acordos){
    if ($fk_mon_tipos == 1 || $fk_mon_tipos == 2) {
        if ($acao == 'cadastrar'){
            $where = "fk_dispositivos = ? and ip_disp_alvo = ? and
fk_acordos = ? and fk_dispositivos_verificador = ?";
            $params = array('isii', $this->DB-
>escape($fk_dispositivos), $this->DB->escape($ip_disp_alvo), $this->DB-
>escape($fk_acordos), $this->DB->escape($fk_dispositivos_verificador));
        } elseif ($acao == 'alterar') {
            $where = "fk_dispositivos = ? and ip_disp_alvo = ? and
fk_acordos = ? and fk_dispositivos_verificador = ? and id_mon <> ?";
            $params = array('isiii', $this->DB-
>escape($fk_dispositivos), $this->DB->escape($ip_disp_alvo), $this->DB-
>escape($fk_acordos), $this->DB->escape($fk_dispositivos_verificador), $this->DB-
>escape($id));
        }
    } elseif ($fk_mon_tipos == 3) {
        if ($acao == 'cadastrar'){
            $where = "fk_dispositivos = ? and ip_disp_alvo = ? and
fk_acordos = ? and fk_dispositivos_verificador = ? and ip_neighbour = ?";
            $params = array('isiis', $this->DB-
>escape($fk_dispositivos), $this->DB->escape($ip_disp_alvo), $this->DB-
>escape($fk_acordos), $this->DB->escape($fk_dispositivos_verificador), $this->DB-
>escape($ip_neighbour));
        } elseif ($acao == 'alterar') {
            $where = "fk_dispositivos = ? and ip_disp_alvo = ? and
fk_acordos = ? and fk_dispositivos_verificador = ? and ip_neighbour = ? and id_mon <> ?";

```



```

        $params = array('isiii', $this->DB-
>escape($fk_dispositivos), $this->DB->escape($ip_disp_alvo), $this->DB-
>escape($fk_acordos), $this->DB->escape($fk_dispositivos_verificador), $this->DB-
>escape($ip_neighbour), $this->DB->escape($id));
    }
}

$result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
monitoramentos m

LEFT JOIN mon_param_snmpo mpso ON m.`id_mon`=mpso.`fk_monitoramentos`

WHERE $where", $params);
    if ($result[0]['count'] >0){ $invalid_fields['fk_acordos'] = "Já
existe monitoramento cadastrado com esses parâmetros para esse acordo."; }

}

if (!empty($invalid_fields)){
    exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
}

$arrayInfo = array();
$arrayInfoExtra = array();
if ($acao == 'cadastrar'){

    $arrayInfo["descricao_mon"] = $descricao_mon;
    $arrayInfo["fk_acordos"] = $tmp_fk_acordos[0];
    $arrayInfo["fk_dispositivos"] = $fk_dispositivos;
    $arrayInfo["fk_dispositivos_verificador"] =
$fk_dispositivos_verificador;
    $arrayInfo["ip_disp_alvo"] = $ip_disp_alvo;

    if (isset($ping_count)) { $arrayInfoExtra["ping_count"] =
$ping_count; }

    if (isset($ping_size)) { $arrayInfoExtra["ping_size"] = $ping_size; }

    if (isset($ports)){
        $arrayInfoExtra["param_ports"] = "";
        natsort($ports);
        foreach ($ports as $port){
            $arrayInfoExtra["param_ports"] .= "{$port}||";
        }
        $arrayInfoExtra["param_ports"] =
substr($arrayInfoExtra["param_ports"], 0, -2);
    }

    if (isset($ip_neighbour)) { $arrayInfoExtra["ip_neighbour"] =
$ip_neighbour; }

} else {

    $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m

LEFT JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`

LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`

```

```

LEFT JOIN mon_param_snmpo mpso ON m.`id_mon`=mpso.`fk_monitoramentos`

WHERE id_mon = ?", array('i', $this->DB->escape($id)));

    $valorAntigo["descricao_mon"] = $result[0]["descricao_mon"];
    $valorAntigo["fk_dispositivos"] = $result[0]['fk_dispositivos'];
    $valorAntigo["fk_dispositivos_verificador"] =
$result[0]['fk_dispositivos_verificador'];
    $valorAntigo["ip_disp_alvo"] = $result[0]['ip_disp_alvo'];

    $valorAntigo["ping_count"] = isset($result[0]['ping_count']) ?
$result[0]['ping_count'] : "";
    $valorAntigo["ping_size"] = isset($result[0]['ping_size']) ?
$result[0]['ping_size'] : "";

    $valorAntigo["ip_neighbour"] = isset($result[0]['ip_neighbour']) ?
$result[0]['ip_neighbour'] : "";

    if ($valorAntigo["descricao_mon"] != $descricao_mon) {
$arrayInfo["descricao_mon"] = $descricao_mon; }
    if ($valorAntigo["fk_dispositivos"] != $fk_dispositivos) {
$arrayInfo["fk_dispositivos"] = $fk_dispositivos; }
    if ($valorAntigo["fk_dispositivos_verificador"] !=
$fk_dispositivos_verificador) { $arrayInfo["fk_dispositivos_verificador"] =
$fk_dispositivos_verificador; }
    if ($valorAntigo["ip_disp_alvo"] != $ip_disp_alvo) {
$arrayInfo["ip_disp_alvo"] = $ip_disp_alvo; }

    if (!isset($ping_count)) { $ping_count = 0; }
    if (!isset($ping_size)) { $ping_size = 0; }

    $portas_antigas = isset($result[0]["param_ports"]) ? explode("||",
$result[0]["param_ports"]) : array();

    if (!isset($ports)) { $ports = array(); } else { natsort($ports); }

$result = $this->DB->execPreparedSQL("SELECT fk_acordos_mon_tipos
FROM monitoramentos m

JOIN acordos a ON a.`id_acordo`=m.`fk_acordos`

WHERE id_mon = ?", array('i', $this->DB->escape($id)));
    $fk_mon_tipos = $result[0]['fk_acordos_mon_tipos'];

    if ($fk_mon_tipos == 1){
        if ($valorAntigo["ping_count"] != $ping_count) {
$arrayInfoExtra["ping_count"] = $ping_count; }
        if ($valorAntigo["ping_size"] != $ping_size) {
$arrayInfoExtra["ping_size"] = $ping_size; }
    } else if ($fk_mon_tipos == 2){
        if (array_diff($portas_antigas, $ports) || array_diff($ports,
$portas_antigas)){
            $arrayInfoExtra["param_ports"] = "";
            foreach ($ports as $port){
                $arrayInfoExtra["param_ports"] .= "{$port}||";
            }
            $arrayInfoExtra["param_ports"] =
substr($arrayInfoExtra["param_ports"], 0, -2);

```

```

    }
    } else if ($fk_mon_tipos == 3){
        if ($valorAntigo["ip_neighbour"] != $ip_neighbour) {
$arrayInfoExtra["ip_neighbour"] = $ip_neighbour; }
        }
    }

$log = "";

$params = array();
$posicoes = array();
$update_query = array();

switch ($fk_mon_tipos) {
    case 1:
        $table = 'mon_param_ping';
        break;
    case 2:
        $table = 'mon_param_snmpi';
        break;
    case 3:
        $table = 'mon_param_snmpo';
        break;
}

if ($acao == 'cadastrar'){

    $action = "CADASTRO DE MONITORAMENTO";
    $action_text = "cadastrado";

    $params[] = $this->DB->getTypeLabel($arrayInfo);
    $log = "<b>Atributos cadastrados:</b><br />";
    foreach ($arrayInfo as $key => $valor) {
        $log .= $key . ": " . $valor . "<br />";
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $this->DB->autocommit(false);
    $result = $this->DB->execPreparedSQL("INSERT INTO monitoramentos
(`".implode("`",`",$chaves_sql).`) VALUES ("`.implode("`",$posicoes).`)", $params, true);
    $mon_id = $this->DB->last_id();

    if (!empty($arrayInfoExtra)){

        $arrayInfoExtra['fk_monitoramentos'] = $mon_id;
        $paramsX[] = $this->DB->getTypeLabel($arrayInfoExtra);
        foreach ($arrayInfoExtra as $keyX => $valorX) {
            $log .= $keyX . ": " . $valorX . "<br />";
            $chaves_sqlX[] = $keyX;
            $posicoesX[] = "?";
            $paramsX[] = $this->DB->escape($valorX);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO $table
(`".implode("`",`",$chaves_sqlX).`) VALUES ("`.implode("`",$posicoesX).`)", $paramsX, true);
    }
}

```

```

        $this->DB->commit();
        $this->DB->autocommit(true);
    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo) || !empty($arrayInfoExtra)) {
            $action = "ALTERAÇÃO DE MONITORAMENTO";
            $action_text = "alterado";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

            $log .= "<b>Atributos alterados: </b><br />";
            foreach ($arrayInfo as $key => $valor) {
                $log .= $key . ": " . $valor . "<br />";
                $update_query[] = "`$key` = ?";
                $params[] = $this->DB->escape($valor);
            }

            $params[] = $this->DB->escape($id);
            $this->DB->autocommit(false);

            if (!empty($update_query)){
                $result = $this->DB->execPreparedSQL("UPDATE
monitoramentos SET ".implode(", ", $update_query)." WHERE id_mon = ?", $params, true);
            }

            switch ($fk_mon_tipos) {
                case 1:
                    $table = 'mon_param_ping';
                    break;
                case 2:
                    $table = 'mon_param_snmpi';
                    break;
                case 3:
                    $table = 'mon_param_snmpo';
                    break;
            }

            if (!empty($arrayInfoExtra)){
                $paramsX[] = $this->DB->getTypeLabel($arrayInfoExtra,
$update = true);

                foreach ($arrayInfoExtra as $keyX => $valorX) {
                    $log .= $keyX . ": " . $valorX . "<br />";
                    $update_queryX[] = "`$keyX` = ?";
                    $paramsX[] = $this->DB->escape($valorX);
                }
                $paramsX[] = $this->DB->escape($id);
                $result = $this->DB->execPreparedSQL("UPDATE $table SET
".implode(", ", $update_queryX)." WHERE fk_monitoramentos = ?", $paramsX, true);
            }

            $this->DB->commit();
            $this->DB->autocommit(true);
        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    }
}

```

```

    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!'))));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Monitoramento
$action_text com sucesso !"))));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Monitoramentos deletados:</b><br/>";
    $log .= implode(",",$ids)."<br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("DELETE FROM monitoramentos WHERE id_mon
IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE MONITORAMENTO", $log))
        return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status'=>'success','msg'=>'Monitoramento excluído com
sucesso!'))));
}

public function showInfo(){

    if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
        exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !"))));
    }

    $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m
LEFT
JOIN acordos a ON m.`fk_acordos`=a.`id_acordo`
LEFT
JOIN dispositivos d ON m.`fk_dispositivos`=d.`id_dispositivo`
LEFT
JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`
LEFT
JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`

```

```

JOIN mon_param_snmpo mpso ON m.`id_mon`=mpso.`fk_monitoramentos` LEFT
JOIN `acordos_tipo_periodicidade` atp ON a.`fk_acordos_tipo_periodicidade` = LEFT
atp.`id_periodicidade`
JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos` = amt.`id_mon_tipo` LEFT
JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo` LEFT

WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"]));

if (empty($result)){
    exit(json_encode(array('status' => 'error')));
} else {
    foreach ($result as $res) {

        $descricao = ($res['descricao_mon']) ? $res['descricao_mon'] :
"-";

        $nome_acordo = $res['nome_acordo'];
        $descricao_mon_tipo = $res['descricao_mon_tipo'];
        $nome_tipo_acordo = $res['nome_tipo_acordo'];
        $params_extras = "";
        $ports = "";

        if ($res['ping_count']){
            $params_extras .= "<strong>Número de
Pacotes:</strong></h4> {$res['ping_count']} <br />";
            $params_extras .= "<strong>Tamanho do
Pacote:</strong></h4> {$res['ping_size']} bytes <br />";
        } elseif ($res['param_ports']){

            $params_extras .= "<strong>Portas: </strong></h4>";

            $tmp_ports = explode("||", $res['param_ports']);
            foreach ($tmp_ports as $oid_port){
                $port = explode("::",$oid_port);
                $ports .= $port[1].", ";
            }

            $params_extras .= substr($ports,0,-2);
            $params_extras .= "<br />";

        } elseif ($res['ip_neighbour']){
            $params_extras .= "<strong>IP do
Neighbour:</strong></h4> {$res['ip_neighbour']} <br />";
        }

        $response = "<div style='display:inline'>";
        $response .= "<strong>Descrição do
Monitoramento:</strong></h4> $descricao <br />";
        $response .= "<strong>Tipo de Monitoramento:</strong></h4>
$descricao_mon_tipo <br />";
        $response .= "<strong>Nome do Acordo:</strong></h4>
$nome_acordo <br />";
        $response .= "<strong>Tipo de Acordo:</strong></h4>
$nome_tipo_acordo <br />";
        $response .= "$params_extras";
        $response .= "</div>";
    }
}

```

```

        $response .= "<div class='clear'></div>";
    }

    exit(json_encode(array("status"=>"success","info"=>$response)));
}

}

public function getInfo(){
    $result = $this->DB->execPreparedSQL("SELECT * FROM monitoramentos m
LEFT JOIN acordos a ON m.`fk_acordos`=a.`id_acordo`
LEFT JOIN mon_param_ping mpp ON m.`id_mon`=mpp.`fk_monitoramentos`
LEFT JOIN mon_param_snmpi mpsi ON m.`id_mon`=mpsi.`fk_monitoramentos`
LEFT JOIN mon_param_snmpo mpso ON m.`id_mon`=mpso.`fk_monitoramentos`
WHERE id_mon = ?", array("i", $this->DB->escape($_REQUEST["id"])));
    if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
}

}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleOperadora{
protected $db;
private static $instance;

/**
 * Método responsável por retornar uma instância da classe
 */
public static function getInstance() {
    if (!self::$instance) {
        self::$instance = new ControleOperadora();
    }
    return self::$instance;
}

/**
 * Construtor da classe
 */
public function __construct(){
    $this->DB = DB::getInstance();
    $this->Logger = Logger::getInstance();
    $this->auth = new Auth();
}

/**
 * Método responsável pela criação da tabela de usuários para montagem de uma
JQgrid.

```

```

*/
public function getTableData(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_OPERADORAS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }
    /*
    * Verifica os dados do parâmetro enviado
    */
    $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
    $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
    $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
    $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
    $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

    $response = new stdClass();

    /*
    * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
    query para filtrar os dados
    */
    $params = array();
    $where = "";
    if(isset($search) && $search != ""){
        $response->filtro = true;
        $search = normalize($search);
        $where = " WHERE nome_operadora LIKE ?
                    OR resp_abertura_chamado LIKE ?
                    OR tel_abertura_chamado LIKE ?";

        $params = array('sss',
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"),
            $this->DB->escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT COUNT(*) AS count FROM
operadoras o $where", $params);

    if(!empty($result)){
        $count = $result[0]['count'];
        if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

        if ($page > $total_pages){ $page = $total_pages; }
        if ($limit < 0){ $limit = 0; }
        $start = $limit * $page - $limit;
        if ($start < 0){ $start = 0; }

        $order = "";
        if(isset($sidx) && $sidx!=""){
            $order = ($sord=='asc')?"ASC":"DESC";
            $order = "ORDER BY $sidx $order";
        }

        $result = $this->DB->execPreparedSQL("SELECT id_operadora,
nome_operadora, resp_abertura_chamado, tel_abertura_chamado FROM operadoras o
$where $order LIMIT $start,$limit", $params);

```



```

$response->page = $page;
$response->total = $total_pages;
$response->records = $count;

$i = 0;
foreach ($result as $res){

    $res['resp_abertura_chamado'] =
($res['resp_abertura_chamado']) ? $res['resp_abertura_chamado'] : "-";
    $res['tel_abertura_chamado'] = ($res['tel_abertura_chamado'])
? $res['tel_abertura_chamado'] : "-";

    $response->rows[$i]['id'] = $res['id_operadora'];
    $response->rows[$i]['cell'] = array(
    format_title_trim($res['nome_operadora'],35),
    format_title_trim($res['resp_abertura_chamado'],20),
    format_title_trim($res['tel_abertura_chamado'],20),
    "<img class='edit' src='" . URI_IMAGENS . "edit.png'
border='0' />"
    );
    $i++;
}

}
exit(json_encode($response));
}

public function add_update(){

if (!$this->auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
    exit(json_encode(array('status' => 'unauthorized')));
}

$invalid_fields = array();

foreach ($_REQUEST as $key => $value) {
    if (is_array($value)) {
        $value_new = array();
        foreach ($value as $chave=>$value_item) {
            if ($value_item != ""){
                $value_new[$chave] =
anti_injection($value_item);
            }
        }
        $$key = $value_new;
    } else {
        $$key = anti_injection($value);
    }
}

if (!$nome_operadora){
    $invalid_fields['nome_operadora'] = "Nome inválido.";
}

if ($tel_abertura_chamado && strlen($tel_abertura_chamado) < 14){
    $invalid_fields['tel_abertura_chamado'] = "Telefone de abertura de
chamado inválido.";
}
}

```

```

        if ($tel_escalamento_nivel_1 && strlen($tel_escalamento_nivel_1) < 14){
            $invalid_fields['tel_escalamento_nivel_1'] = "Telefone de
escalamento nível 1 inválido.";
        }

        if ($tel_escalamento_nivel_2 && strlen($tel_escalamento_nivel_2) < 14){
            $invalid_fields['tel_escalamento_nivel_2'] = "Telefone de
escalamento nível 2 inválido.";
        }

        if ($acao == 'cadastrar'){
            $where = "nome_operadora = ?";
            $params = array('s', $this->DB->escape($nome_operadora));
        } elseif ($acao == 'alterar') {
            $where = "nome_operadora = ? and id_operado1ra <> ?";
            $params = array('si', $this->DB->escape($nome_operadora), $this->DB->
>escape($id));
        }

        $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
operadoras WHERE $where", $params);
        if ($result[0]['count'] >0){ $invalid_fields['nome_operadora'] = "Nome já
existente."; }

        if (!empty($invalid_fields)){
            exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
        }

        $arrayInfo = array();
        if ($acao == 'cadastrar'){

            $arrayInfo["nome_operadora"] = $nome_operadora;

            $arrayInfo["resp_abertura_chamado"] = ($resp_abertura_chamado) ?
$resp_abertura_chamado : "";
            $arrayInfo["tel_abertura_chamado"] = ($tel_abertura_chamado) ?
$tel_abertura_chamado : "";
            $arrayInfo["resp_escalamento_nivel_1"] =
($resp_escalamento_nivel_1) ? $resp_escalamento_nivel_1 : "";
            $arrayInfo["tel_escalamento_nivel_1"] =
($tel_escalamento_nivel_1) ? $tel_escalamento_nivel_1 : "";
            $arrayInfo["resp_escalamento_nivel_2"] =
($resp_escalamento_nivel_2) ? $resp_escalamento_nivel_2 : "";
            $arrayInfo["tel_escalamento_nivel_2"] =
($tel_escalamento_nivel_2) ? $tel_escalamento_nivel_2 : "";

        } else {

            $result = $this->DB->execPreparedSQL("SELECT * FROM operadoras WHERE
id_operadora = ?", array('i', $this->DB->escape($id)));
            $valorAntigo["nome_operadora"] = $result[0]['nome_operadora'];
            $valorAntigo["resp_abertura_chamado"] =
$result[0]['resp_abertura_chamado'];
            $valorAntigo["tel_abertura_chamado"] =
$result[0]['tel_abertura_chamado'];
            $valorAntigo["resp_escalamento_nivel_1"] =
$result[0]['resp_escalamento_nivel_1'];

```

```

        $valorAntigo["tel_escalonamento_nivel_1"] =
$result[0]['tel_escalonamento_nivel_1'];
        $valorAntigo["resp_escalonamento_nivel_2"] =
$result[0]['resp_escalonamento_nivel_2'];
        $valorAntigo["tel_escalonamento_nivel_2"] =
$result[0]['tel_escalonamento_nivel_2'];

        if ($valorAntigo["resp_abertura_chamado"] != $resp_abertura_chamado)
{ $arrayInfo["resp_abertura_chamado"] = $resp_abertura_chamado; }
        if ($valorAntigo["tel_abertura_chamado"] != $tel_abertura_chamado) {
$arrayInfo["tel_abertura_chamado"] = $tel_abertura_chamado; }
        if ($valorAntigo["resp_escalonamento_nivel_1"] !=
$resp_escalonamento_nivel_1) { $arrayInfo["resp_escalonamento_nivel_1"] =
$resp_escalonamento_nivel_1; }
        if ($valorAntigo["tel_escalonamento_nivel_1"] !=
$tel_escalonamento_nivel_1) { $arrayInfo["tel_escalonamento_nivel_1"] =
$tel_escalonamento_nivel_1; }
        if ($valorAntigo["resp_escalonamento_nivel_2"] !=
$resp_escalonamento_nivel_2) { $arrayInfo["resp_escalonamento_nivel_2"] =
$resp_escalonamento_nivel_2; }
        if ($valorAntigo["tel_escalonamento_nivel_2"] !=
$tel_escalonamento_nivel_2) { $arrayInfo["tel_escalonamento_nivel_2"] =
$tel_escalonamento_nivel_2; }

    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE OPERADORA";
        $action_text = "cadastrada";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO operadoras
(`".implode("``,`",$chaves_sql).`) VALUES (".implode("`,`",$posicoes).`)", $params, true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo)) {
            $action = "ALTERAÇÃO DE OPERADORA";
            $action_text = "alterada";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

            $log .= "<b>Dispositivo alterado:<br/>
{$valorAntigo["nome_operadora"]}<br />";
            $log .= "<b>Atributos alterados: </b><br />";
            foreach ($arrayInfo as $key => $valor) {

```

```

        $log .= $key . ": " . $valor . "<br />";
        $update_query[] = "`$key` = ?";
        $params[] = $this->DB->escape($valor);
    }
    $params[] = $this->DB->escape($id);
    $result = $this->DB->execPreparedSQL("UPDATE operadoras SET
".implode(", ", $update_query)." WHERE id_operadora = ?", $params, true);

        } else {
            exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
        }
    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
    }

    if (!$this->Logger->logACTION($action, $log))
        return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status' => 'success', 'msg' => "Operadora
$action_text com sucesso !")));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_OPERADORAS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Monitoramentos deletados:</b><br/>";
    $log .= implode(", ", $ids)."<br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("DELETE FROM operadoras WHERE
id_operadora IN (" . implode(", ", $posicoes) . ")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE OPERADORA", $log))
        return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

    exit(json_encode(array('status'=>'success', 'msg'=>'Operadora excluída com
sucesso!')));
}

public function showInfo(){

    if (!$this->auth->check(ID_SUBMODULO_GERENCIAVEIS_OPERADORAS)) {

```

```

        exit(json_encode(array('status' => 'unauthorized', 'msg'=>"Não
Autorizado !"))));
    }

    $result = $this->DB->execPreparedSQL("SELECT * FROM operadoras o
WHERE id_operadora = ?", array("i", $this->DB->escape($_REQUEST["id"])));

    if (empty($result)){
        exit(json_encode(array('status' => 'error')));
    } else {
        foreach ($result as $res) {

            $resp_escalamento_nivel_1 =
$res['resp_escalamento_nivel_1'];
            $tel_escalamento_nivel_1 =
$res['tel_escalamento_nivel_1'];
            $resp_escalamento_nivel_2 =
$res['resp_escalamento_nivel_2'];
            $tel_escalamento_nivel_2 =
$res['tel_escalamento_nivel_2'];

            $response = "<div style='display:inline'>";
            $response .= "<strong>Escalonamento Nível 1:</strong></h4>
$resresp_escalamento_nivel_1 $tel_escalamento_nivel_1 <br />";
            $response .= "<strong>Escalonamento Nível 2:</strong></h4>
$resresp_escalamento_nivel_2 $tel_escalamento_nivel_2 <br />";
            $response .= "</div>";
            $response .= "<div class='clear'></div>";
        }

        exit(json_encode(array("status"=>"success","info"=>$response)));
    }
}

public function getInfo(){
    $result = $this->DB->execPreparedSQL("SELECT * FROM operadoras o
WHERE id_operadora = ?", array("i", $this->DB->escape($_REQUEST["id"])));
    if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
}

}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleUnidade{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe

```

```

    */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleUnidade();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
    public function __construct(){
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
        $this->auth = new Auth();
    }

    /**
     * Método responsável pela criação da tabela de usuários para montagem de uma
     JQgrid.
     */
    public function getTableData(){

        if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_UNIDADES)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }
        /*
         * Verifica os dados do parâmetro enviado
         */
        $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
        $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
        $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
        $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
        $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

        $response = new stdClass();

        /*
         * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
         query para filtrar os dados
         */
        $params = array();
        $where = "";
        if(isset($search) && $search != ""){
            $response->filtro = true;
            $search = normalize($search);
            $where = " WHERE sigla_instituicao LIKE ? OR sigla_unidade LIKE ? OR
nome_unidade LIKE ? OR descricao_unidade LIKE ? ";

            //joao etc

            //where (nome = like '%search1' or centro = like '%search1') and (
nome = like '%search2%' or centro = like '%search2%') and ....

            $params = array('ssss', $this->DB->escape("%$search%"), $this->DB-
>escape("%$search%"), $this->DB->escape("%$search%"));
        }
    }

```

```

        $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
organizacao_unidades ou
                                                                    JOIN
organizacao_instituicoes oi
                                                                    ON
ou.`fk_organizacao_instituicoes`=oi.`id_instituicao` $where", $params);

        /*
        * Caso não ocorrer erros durante a busca, configura o total de páginas dado
o número de dados limite por página
        */
        if(!empty($result)){
            $count = $result[0]['count'];
            if ($count > 0) { $total_pages = ceil($count / $limit); } else {
$total_pages = 0; }

            if ($page > $total_pages){ $page = $total_pages; }
            if ($limit < 0){ $limit = 0; }
            $start = $limit * $page - $limit;
            if ($start < 0){ $start = 0; }

            $order = "";
            if(isset($sidx) && $sidx!=""){
                $order = ($sord=='asc')?"ASC":"DESC";
                $order = "ORDER BY $sidx $order";
            }

            $result = $this->DB->execPreparedSQL("SELECT id_unidade,
sigla_instituicao, sigla_unidade, nome_unidade, descricao_unidade
FROM organizacao_unidades ou JOIN organizacao_instituicoes oi
                                                                    ON
ou.`fk_organizacao_instituicoes`=oi.`id_instituicao` $where $order LIMIT $start,$limit",
$params);

            $response->page = $page;
            $response->total = $total_pages;
            $response->records = $count;

            /*
            * Configura a tabela na variável response para retornar para JQgrid
da página
            */
            $i = 0;
            foreach ($result as $res){

                $descricao = ($res["descricao_unidade"]) ?
$res["descricao_unidade"] : "-";

                $response->rows[$i]['id'] = $res['id_unidade'];
                $response->rows[$i]['cell'] = array(
                    format_title_trim($res['sigla_unidade'],15),
                    format_title_trim($res['nome_unidade'],20),
                    format_title_trim($res['sigla_instituicao'],15),
                    format_title_trim($descricao,40),
                    "<img class='edit' src='\" . URI_IMAGENS . \"edit.png'
border='0' />"

                );
                $i++;
            }

```

```

    }
}
exit(json_encode($response));
}

public function add_update(){

if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_UNIDADES)) {
    exit(json_encode(array('status' => 'unauthorized')));
}

$invalid_fields = array();

foreach ($_REQUEST as $key => $value) {
    if (is_array($value)) {
        $value_new = array();
        foreach ($value as $value_item) {
            $value_new[] = anti_injection($value_item);
        }
        $$key = array_sum($value_new);
    } else {
        $$key = anti_injection($value);
    }
}

if (!$sigla_unidade){
    $invalid_fields['sigla_unidade'] = "Sigla inválida.";
}

if (!$nome_unidade){
    $invalid_fields['nome_unidade'] = "Nome inválido.";
}

if (!$fk_organizacao_instituicoes){
    $invalid_fields['fk_organizacao_instituicoes'] = "Instituição
inválida.";
}

if ($acao == 'cadastrar'){
    $where = "sigla_unidade = ? and fk_organizacao_instituicoes = ?";
    $params = array('si', $this->DB->escape($sigla_unidade), $this->DB->
escape($fk_organizacao_instituicoes));
} elseif ($acao == 'alterar') {
    $where = "sigla_unidade = ? and id_unidade <> ? and
fk_organizacao_instituicoes = ?";
    $params = array('sii', $this->DB->escape($sigla_unidade), $this->DB->
escape($id), $this->DB->escape($fk_organizacao_instituicoes));
}

$result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
organizacao_unidades WHERE $where", $params);
if ($result[0]['count'] >0){ $invalid_fields['sigla_unidade'] = "Sigla já
existente."; }

if (!empty($invalid_fields)){
    exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
}
}
}

```



```

$arrayInfo = array();
if ($acao == 'cadastrar'){
    $arrayInfo["sigla_unidade"] = $sigla_unidade;
    $arrayInfo["nome_unidade"] = $nome_unidade;
    $arrayInfo["descricao_unidade"] = $descricao_unidade;
    $arrayInfo["fk_organizacao_instituicoes"] =
$fk_organizacao_instituicoes;
    } else {
        $result = $this->DB->execPreparedSQL("SELECT * FROM
organizacao_unidades WHERE id_unidade = ?", array('i', $this->DB->escape($id)));
        $valorAntigo["sigla_unidade"] = $result[0]['sigla_unidade'];
        $valorAntigo["nome_unidade"] = $result[0]['nome_unidade'];
        $valorAntigo["descricao_unidade"] = $result[0]['descricao_unidade'];
        $valorAntigo["fk_organizacao_instituicoes"] =
$result[0]['fk_organizacao_instituicoes'];

        if ($valorAntigo["sigla_unidade"] != $sigla_unidade) {
            $arrayInfo["sigla_unidade"] = $sigla_unidade; }
        if ($valorAntigo["nome_unidade"] != $nome_unidade) {
            $arrayInfo["nome_unidade"] = $nome_unidade; }
        if ($valorAntigo["descricao_unidade"] != $descricao_unidade) {
            $arrayInfo["descricao_unidade"] = $descricao_unidade; }
        if ($valorAntigo["fk_organizacao_instituicoes"] !=
$fk_organizacao_instituicoes) { $arrayInfo["fk_organizacao_instituicoes"] =
$fk_organizacao_instituicoes; }
    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE UNIDADE";
        $action_text = "cadastrada";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }

        $result = $this->DB->execPreparedSQL("INSERT INTO
organizacao_unidades (`".implode("`", $chaves_sql)."`") VALUES
(`".implode("`,`", $posicoes)."`)", $params, true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo)) {
            $action = "ALTERAÇÃO DE UNIDADE";
            $action_text = "alterada";

            $params[] = $this->DB->getTypeLabel($arrayInfo, $update =
true);

```

```

        $log .= "<b>Unidade alterada:<br/>
{$valorAntigo["sigla_unidade"]}<br />";
        $log .= "<b>Atributos alterados: </b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $update_query[] = "`$key` = ?";
            $params[] = $this->DB->escape($valor);
        }
        $params[] = $this->DB->escape($id);
        $result = $this->DB->execPreparedSQL("UPDATE
organizacao_unidades SET ".implode(",",$update_query)." WHERE id_unidade = ?", $params,
true);

    } else {
        exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
    }
    } else {
        exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
    }
}

if (!$this->Logger->logACTION($action, $log))
return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

exit(json_encode(array('status' => 'success', 'msg' => "Unidade $action_text
com sucesso !")));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_ORGANIZACAO_UNIDADES)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Unidades deletadas:</b><br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("SELECT sigla_unidade FROM
organizacao_unidades WHERE id_unidade IN (".implode(",",$posicoes).")", $params);

    foreach ($result as $res){ $log .= "{$res['sigla_unidade']}<br/>"; }

    $result = $this->DB->execPreparedSQL("DELETE FROM organizacao_unidades WHERE
id_unidade IN (".implode(",",$posicoes).")", $params, true);

    if (!$this->Logger->logACTION("EXCLUSÃO DE UNIDADE", $log))
return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));
}

```

```

        exit(json_encode(array('status'=>'success', 'msg'=>'Unidade(s) excluída(s) com
sucesso!'))));
        break;
    }

    public function getInfo(){

        $result = $this->DB->execPreparedSQL("SELECT * FROM organizacao_unidades
WHERE id_unidade = ?", array("i", $this->DB->escape($_REQUEST["id"])));
        if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
        break;

    }

    public function getUnidades(){

        $id_unidade = (isset($_REQUEST['select'])) ? $this->DB-
>escape($_REQUEST['select']) : ("");

        $unidades = "<option value='*'>Todas</option>";
        $result = $this->DB->execPreparedSQL("SELECT id_unidade, sigla_unidade,
nome_unidade FROM organizacao_unidades WHERE fk_organizacao_instituicoes = ? order by
sigla_unidade", array("i", $id_unidade));
        foreach ($result as $res){
            $unidades .= "<option
value='{ $res['id_unidade'] }'>{ $res['sigla_unidade'] } [ { $res['nome_unidade'] } ]</option>";
        }

        $retorno = array("options" => $unidades);
        exit (json_encode($retorno));
    }
}
?>

<?php
/**
 * Módulo responsável pelo controle do usuário.
 */

class ControleUsuario{
    protected $db;
    private static $instance;

    /**
     * Método responsável por retornar uma instância da classe
     */
    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new ControleUsuario();
        }
        return self::$instance;
    }

    /**
     * Construtor da classe
     */
}

```

```

public function __construct(){
    $this->DB = DB::getInstance();
    $this->Logger = Logger::getInstance();
    $this->auth = new Auth();
}

/**
 * Método responsável pela criação da tabela de usuários para montagem de uma
JQgrid.
 */
public function getTableData(){

    if (!$this->auth->check(ID_SUBMODULO_ADMIN_USUARIOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }
    /*
     * Verifica os dados do parâmetro enviado
     */
    $page = isset($_REQUEST['page']) ? $_REQUEST['page'] : null;
    $limit = isset($_REQUEST['rows']) ? $_REQUEST['rows'] : null;
    $sidx = isset($_REQUEST['sidx']) ? $_REQUEST['sidx'] : null;
    $sord = isset($_REQUEST['sord']) ? $_REQUEST['sord'] : null;
    $search = (isset($_REQUEST['search']) && $_REQUEST['search']) ?
$_REQUEST['search'] : null;

    $response = new stdClass();

    /*
     * Se tiver sido enviado um parâmetro de busca, coloca a cláusula WHERE na
query para filtrar os dados
     */
    $params = array();
    $where = "";
    if(isset($search) && $search != ""){
        $response->filtro = true;
        $search = normalize($search);
        $where = " WHERE nome_usuario LIKE ? OR email_usuario LIKE ? OR
login_usuario LIKE ? OR telefone_usuario LIKE ? ";
        $params = array('ssss', $this->DB->escape("%$search%"), $this->DB-
>escape("%$search%"), $this->DB->escape("%$search%"), $this->DB->escape("%$search%"));
    }

    $result = $this->DB->execPreparedSQL("SELECT count(*) as count

FROM usuarios u

LEFT JOIN organizacao_instituicoes oi ON
u.`fk_organizacao_instituicoes`=oi.`id_instituicao`

LEFT JOIN organizacao_unidades ou ON u.`fk_organizacao_unidades`=ou.`id_unidade`
$where", $params);

    /*
     * Caso não ocorrer erros durante a busca, configura o total de páginas dado
o número de dados limite por página
     */
    if(!empty($result)){
        $count = $result[0]['count'];
    }
}

```

```

$total_pages = 0; }
if ($count > 0) { $total_pages = ceil($count / $limit); } else {

if ($page > $total_pages){ $page = $total_pages; }
if ($limit < 0){ $limit = 0; }
$start = $limit * $page - $limit;
if ($start < 0){ $start = 0; }

$order = "";
if(isset($sidx) && $sidx!=""){
    $order = ($sord=='asc')?"ASC":"DESC";
    $order = "ORDER BY $sidx $order";
}

$result = $this->DB->execPreparedSQL("SELECT id_usuario,
nome_usuario, email_usuario, login_usuario, telefone_usuario, sigla_instituicao,
sigla_unidade

FROM usuarios u

LEFT JOIN organizacao_instituicoes oi ON
u.`fk_organizacao_instituicoes`=oi.`id_instituicao`

LEFT JOIN organizacao_unidades ou ON u.`fk_organizacao_unidades`=ou.`id_unidade`
$where $order LIMIT $start,$limit", $params);

$response->page = $page;
$response->total = $total_pages;
$response->records = $count;

/*
 * Configura a tabela na variável response para retornar para JQgrid
da página
 */
$i = 0;
foreach ($result as $res){

    $res['sigla_instituicao'] = $res['sigla_instituicao'] ?
$res['sigla_instituicao'] : "Todas";
    $res['sigla_unidade'] = $res['sigla_unidade'] ?
$res['sigla_unidade'] : "Todas";
    $res['telefone_usuario'] = $res['telefone_usuario'] ?
$res['telefone_usuario'] : "-";

    $response->rows[$i]['id'] = $res['id_usuario'];
    $response->rows[$i]['cell'] = array(
    format_title_trim($res['nome_usuario'],20),
    format_title_trim($res['login_usuario'],15),
    format_title_trim($res['email_usuario'],20),
    format_title_trim($res['sigla_instituicao'],15),
    format_title_trim($res['sigla_unidade'],15),
    format_title_trim($res['telefone_usuario'],15),
    "<img class='edit' src='' . URI_IMAGENS .
"edit.png' border='0' />"

    );
    $i++;
}
}

```

```

        exit(json_encode($response));
    }

    public function add_update(){

        if (!$this->auth->check(ID_SUBMODULO_ADMIN_USUARIOS)) {
            exit(json_encode(array('status' => 'unauthorized')));
        }

        $invalid_fields = array();

        foreach ($_REQUEST as $key => $value) {
            if (is_array($value)) {
                $value_new = array();
                foreach ($value as $chave=>$value_item) {
                    if ($value_item != ""){
                        $value_new[$chave] =
anti_injection($value_item);
                    }
                }
                $$key = $value_new;
            } else {
                $$key = anti_injection($value);
            }
        }

        if (!$nome_usuario){
            $invalid_fields['nome_usuario'] = "Nome inválido.";
        }

        if(strlen($login_usuario) < 4){
            $invalid_fields['login_usuario'] = "O login deve possuir mais de 4
caracteres.";
        }

        if (strlen($senha_usuario) < 4) {
            $invalid_fields['senha_usuario'] = "A senha deve conter 4 caracteres
ou mais.";
        }

        if (!validaEmail($email_usuario)) {
            $invalid_fields['email_usuario'] = "E-mail inválido.";
        }

        if ($acao == 'cadastrar'){
            $where = "login_usuario = ?";
            $params = array('s', $this->DB->escape($nome_usuario));
        } elseif ($acao == 'alterar') {
            $where = "login_usuario = ? and id_usuario <> ?";
            $params = array('si', $this->DB->escape($nome_usuario), $this->DB-
>escape($id));
        }

        $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM usuarios
WHERE $where", $params);
        if ($result[0]['count'] >0){ $invalid_fields['login_usuario'] = "Login já
existente."; }
    }

```

```

        if (!empty($invalid_fields)){
            exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
        }

        $somaPriv = 0;
        $somaSomenteLeitura = 0;

        $mudanca_privilegios = false;
        $privilegios_novos = array();
        if (!isset($somenteLeitura)) $somenteLeitura = array();

        if(isset($privilegios)){
            for($i=0;$i<sizeof($privilegios);$i++){
                $sl = 0;
                $temp = explode("_",$privilegios[$i]);
                $somaPriv += $temp[1];
                if(in_array($privilegios[$i],$somenteLeitura)){
                    $sl = 1;
                    $somaSomenteLeitura += $temp[1];
                }
                $privilegios_modulo = $temp[0];
                $privilegios_submodulo = $temp[1];

                $privilegios_novos[$privilegios_modulo."_".$privilegios_submodulo] = $sl;
            }
        }

        $arrayInfo = array();
        if ($acao == 'cadastrar'){

            $arrayInfo["nome_usuario"] = $nome_usuario;
            $arrayInfo["email_usuario"] = $email_usuario;
            $arrayInfo["login_usuario"] = $login_usuario;
            $arrayInfo["senha_usuario"] = md5($senha_usuario);
            $arrayInfo["fk_organizacao_instituicoes"] =
$fk_organizacao_instituicoes;
            $arrayInfo["fk_organizacao_unidades"] = $fk_organizacao_unidades;

            if ($telefone_usuario) { $arrayInfo["telefone_usuario"] =
$telefone_usuario; }

        } else {

            $result = $this->DB->execPreparedSQL("SELECT * FROM usuarios WHERE
id_usuario = ?", array('i', $this->DB->escape($id)));
            $valorAntigo["nome_usuario"] = $result[0]['nome_usuario'];
            $valorAntigo["email_usuario"] = $result[0]['email_usuario'];
            $valorAntigo["login_usuario"] = $result[0]['login_usuario'];
            $valorAntigo["telefone_usuario"] = $result[0]['telefone_usuario'];
            $valorAntigo["fk_organizacao_instituicoes"] =
$result[0]['fk_organizacao_instituicoes'];
            $valorAntigo["fk_organizacao_unidades"] =
$result[0]['fk_organizacao_unidades'];

            if ($valorAntigo["nome_usuario"] != $nome_usuario) {
                $arrayInfo["nome_usuario"] = $nome_usuario; }
            if ($valorAntigo["email_usuario"] != $email_usuario) {
                $arrayInfo["email_usuario"] = $email_usuario; }
        }
    }
}

```

```

        if ($valorAntigo["login_usuario"] != $login_usuario) {
$arrayInfo["login_usuario"] = $login_usuario; }
        if ($valorAntigo["telefone_usuario"] != $telefone_usuario) {
$arrayInfo["telefone_usuario"] = $telefone_usuario; }
        if ($valorAntigo["fk_organizacao_instituicoes"] !=
$fk_organizacao_instituicoes) { $arrayInfo["fk_organizacao_instituicoes"] =
$fk_organizacao_instituicoes; }
        if ($valorAntigo["fk_organizacao_unidades"] !=
$fk_organizacao_unidades) { $arrayInfo["fk_organizacao_unidades"] =
$fk_organizacao_unidades; }

        if ($senha_usuario != "xxxxxx") { $arrayInfo["senha_usuario"] =
md5($senha_usuario); }

        $privilegios_antigos = $this->getPrivilegios();
        if (array_diff_assoc($privilegios_novos, $privilegios_antigos) ||
array_diff_assoc($privilegios_antigos, $privilegios_novos)){
            $mudanca_privilegios = true;
        }

        if ($valorAntigo["login_usuario"] == "admin" && $login_usuario !=
"admin") {
            $invalid_fields['login_usuario'] = "O login do usuário admin
não pode ser alterado.";
        }

        if (!array_key_exists('1_1', $privilegios_novos) ||
$privilegios_novos['1_1'] == 1){
            $invalid_fields['login_usuario'] = "O usuário admin precisa
ter permissão para o módulo Usuários.";
        }

        if (!empty($invalid_fields)){
            exit(json_encode(array('status' => 'error', 'errors' =>
$invalid_fields)));
        }
    }

    $log = "";

    $params = array();
    $posicoes = array();
    $update_query = array();

    if ($acao == 'cadastrar'){
        $action = "CADASTRO DE USUÁRIO";
        $action_text = "cadastrado";

        $params[] = $this->DB->getTypeLabel($arrayInfo);
        $log = "<b>Atributos cadastrados:</b><br />";
        foreach ($arrayInfo as $key => $valor) {
            $log .= $key . ": " . $valor . "<br />";
            $chaves_sql[] = $key;
            $posicoes[] = "?";
            $params[] = $this->DB->escape($valor);
        }
    }

```



```

        if ($somaPriv) { $log .= "Soma Privilégios" . ": " . $somaPriv . "<br
/>"; }

        if ($somaSomenteLeitura) { $log .= "Soma SomenteLeitura" . ": " .
$somaSomenteLeitura . "<br />"; }

        $this->DB->autocommit(false);
        $result = $this->DB->execPreparedSQL("INSERT INTO usuarios
(`".implode("`",`",$chaves_sql).`) VALUES (".implode(",",$posicoes).")", $params, true);
        $user_id = $this->DB->last_id();

        if(isset($privilegios_novos)){
            foreach ($privilegios_novos as $priv=>$sl){
                $temp = explode("_",$priv);
                $privilegios_modulo = $temp[0];
                $privilegios_submodulo = $temp[1];
                $result = $this->DB->execPreparedSQL("INSERT INTO
usuarios_privilegios
(`fk_usuarios`,`modulo_privilegio`,`submodulo_privilegio`,`somenteLeitura_privilegio`)
VALUES (?, ?, ?, ?)", array("iiii", $this->DB->escape($user_id), $this->DB-
>escape($privilegios_modulo), $this->DB->escape($privilegios_submodulo), $this->DB-
>escape($sl)), true);
            }
        }

        $this->DB->commit();
        $this->DB->autocommit(true);

    } elseif ($acao == 'alterar') {
        if (!empty($arrayInfo) || $mudanca_privilegios) {
            $action = "ALTERAÇÃO DE USUÁRIO";
            $action_text = "alterado";

            $log .= "<b>Usuário alterado:<br/>
{$valorAntigo["nome_usuario"]}<br />";
            $log .= "<b>Atributos alterados: </b><br />";

            $this->DB->autocommit(false);

            if (!empty($arrayInfo)){
                $params[] = $this->DB->getTypeLabel($arrayInfo, $update
= true);

                foreach ($arrayInfo as $key => $valor) {
                    $log .= $key . ": " . $valor . "<br />";
                    $update_query[] = "`$key` = ?";
                    $params[] = $this->DB->escape($valor);
                }
                $params[] = $this->DB->escape($id);
                $result = $this->DB->execPreparedSQL("UPDATE usuarios
SET ".implode(", ", $update_query). " WHERE id_usuario = ?", $params, true);
            }

            if ($mudanca_privilegios) {

                if ($somaPriv) { $log .= "Soma Privilégios" . ": " .
$somaPriv . "<br />"; }

                if ($somaSomenteLeitura) { $log .= "Soma
SomenteLeitura" . ": " . $somaSomenteLeitura . "<br />"; }
                $result = $this->DB->execPreparedSQL("DELETE FROM
usuarios_privilegios WHERE fk_usuarios = ?", array("i", $this->DB->escape($id)), true);

```

```

        foreach ($privilegios_novos as $priv=>$sl){
            $temp = explode("_",$priv);
            $privilegios_modulo = $temp[0];
            $privilegios_submodulo = $temp[1];
            $result = $this->DB->execPreparedSQL("INSERT
INTO usuarios_privilegios
(`fk_usuarios`,`modulo_privilegio`,`submodulo_privilegio`,`somenteLeitura_privilegio`)
VALUES (?, ?, ?, ?)", array("iiii", $this->DB->escape($id), $this->DB-
>escape($privilegios_modulo), $this->DB->escape($privilegios_submodulo), $this->DB-
>escape($sl)), true);
        }

        $this->DB->commit();
        $this->DB->autocommit(true);

    } else {
        exit(json_encode(array('status' => 'warning', 'msg' => 'Nenhum
campo foi modificado!')));
    }
} else {
    exit(json_encode(array('status' => 'error', 'msg' => 'Ação
Inválida!')));
}

if (!$this->Logger->logACTION($action, $log))
return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

exit(json_encode(array('status' => 'success', 'msg' => "Usuário $action_text
com sucesso !")));
}

public function delete(){

    if (!$this->auth->check(ID_SUBMODULO_ADMIN_USUARIOS)) {
        exit(json_encode(array('status' => 'unauthorized')));
    }

    $ids = anti_injection($_REQUEST["ids"]);
    $log = "<b>Usuários deletados:</b><br/>";

    $params = array();
    $posicoes = array();
    $params[] = $this->DB->getTypeLabel($ids);
    foreach ($ids as $key=>$valor) {
        $posicoes[] = "?";
        $params[] = $this->DB->escape($valor);
    }

    $result = $this->DB->execPreparedSQL("SELECT nome_usuario FROM usuarios WHERE
id_usuario IN (".implode(",",$posicoes).")", $params);

    foreach ($result as $res){ $log .= "{$res['nome_usuario']}<br/>"; }

    $result = $this->DB->execPreparedSQL("DELETE FROM usuarios WHERE id_usuario
IN (".implode(",",$posicoes).")", $params, true);
}

```

```

        if (!$this->Logger->logACTION("EXCLUSÃO DE USUÁRIO", $log))
            return json_encode(array("status"=>"error", "msg"=>"Erro no banco de dados.<br
/> A operação foi efetuada mas não logada!"));

        exit(json_encode(array('status'=>'success', 'msg'=>'Usuário(s) excluído(s) com
sucesso!'))));
    }

    public function getInfo(){
        $result = $this->DB->execPreparedSQL("SELECT * FROM usuarios WHERE id_usuario
= ?", array("i", $this->DB->escape($_REQUEST["id"])));
        if (!empty($result)){ return $result[0]; } else { exit('<p class="error">ID
Inválido.</p>'); }
    }

    public function getPrivilegios(){
        $privilegios = array();
        $result = $this->DB->execPreparedSQL("SELECT modulo_privilegio,
submodulo_privilegio, somenteLeitura_privilegio FROM usuarios_privilegios WHERE fk_usuarios
= ?", array("i", $this->DB->escape($_REQUEST["id"])));
        if (!empty($result)){
            foreach ($result as $res){

                $privilegios[$res['modulo_privilegio']."_".$res['submodulo_privilegio']] =
(int)$res['somenteLeitura_privilegio'];
            }
        }
        return $privilegios;
    }
}
?>

<?php
require_once(APPLICATION_PATH ."/config/padrao.php");

class Auth {

    private $DB;

    public function __construct() {
        $this->DB = DB::getInstance();
        $this->Logger = Logger::getInstance();
    }

    public function login($login_usuario, $login_senha, $chk, $imgVerificacao = false) {

        /* Verifica se o usuário preencheu os campos */
        if (!$login_usuario || !$login_senha){
            return json_encode(array("status"=>"warning", "msg"=>"Atenção:
Preencha os campos."));
        }

        /* Código de verificação para dificultar roubo de sessão */
        $codigo_verificacao = md5($_SERVER['HTTP_USER_AGENT'] . "%cH3Ck#c0d3%");

        /* Verifica o código */
        if ($codigo_verificacao != $chk)

```

```

return json_encode(array("status"=>"error", "msg"=>"Erro: Autenticação
abortada."));

/* Sessão é válida */
session_start();

/* Verifica o número de tentativas de login */
if (isset($_SESSION['tentativas']) && ($_SESSION['tentativas'] >= 3)) {
    if (isset($imgVerificacao) && $imgVerificacao!='') {
        include(dirname(__FILE__) . "/vImage.php");
        $vImage = new vImage();
        $vImage->loadCodes();
        if (!$vImage->checkCode()){
            return
json_encode(array("error"=>"tentativas", "status"=>"error", "msg"=>"Codigo de verificação
incorreto."));
        }
    } else {
        if (isset($imgVerificacao) && $imgVerificacao=='') {
            return
json_encode(array("error"=>"tentativas", "status"=>"warning", "msg"=>"Preencha o campo de
verificação."));
        }
    }
    return
json_encode(array("error"=>"tentativas", "status"=>"warning", "msg"=>"Excedeu o número de
tentativas."));
}

/* Pesquisa os dados do usuário com o email e senha passados */
$result = $this->DB->execPreparedSQL("SELECT * FROM usuarios WHERE
login_usuario = ? and senha_usuario=MD5(?)", array('ss', $this->DB->escape($login_usuario),
$this->DB->escape($login_senha)));

/* Verifica se retornou algum resultado */
if (empty($result)) {

    $_SESSION['tentativas'] = isset($_SESSION['tentativas']) ?
$_SESSION['tentativas']+1 : 0; //Incrementa número de tentativas de login

    $log_descricao = "Tentativa de login: " . $login_usuario;

    if (!$this->Logger->logACTION("TENTATIVA DE LOGIN", $log_descricao))
return json_encode(array("status"=>"error", "msg"=>"Erro no banco de
dados.<br /> A operação foi cancelada!"));

    # LOG - Fim #
$error = (isset($_SESSION['tentativas']) && ($_SESSION['tentativas']
>= 3)) ? "tentativas" : null;
return json_encode(array("error"=>"$error",
"status"=>"error", "msg"=>"Usuário ou senha inválido(s)."));
}

/* Caso os dados estejam corretos, armazena as variáveis na session */
$_SESSION['tentativas'] = 0;
$_SESSION["HTTP_USER_AGENT"] = md5($_SERVER['HTTP_USER_AGENT']);
$_SESSION["id_usuario"] = $result[0]['id_usuario'];
$_SESSION["nome_usuario"] = $result[0]['nome_usuario'];
$_SESSION["login_usuario"] = $login_usuario;

```

```

        $id = $_SESSION["id_usuario"];

        $result = $this->DB->execPreparedSQL("SELECT count(*) as count FROM
usuarios_privilegios WHERE fk_usuarios = ?", array('i', $this->DB->escape($id)));
        if (isset($result[0]["count"]) && $result[0]["count"] == 0){
            $this->logout();
            return json_encode(array("status"=>"warning","msg"=>"Você não possui
privilégio para acessar o sistema.</br>Por favor contate o administrador!"));
        }

        # LOG - Início #
        $log_descricao = "Login: " . $login_usuario . " - Nome: " .
$_SESSION["nome_usuario"];

        if (!$this->Logger->logACTION("LOGIN", $log_descricao))
            return json_encode(array("status"=>"error","msg"=>"Erro no banco de dados.<br
/> A operação foi cancelada!"));

        return json_encode(array("status"=>"success","msg"=>"Autenticado"));
    }

    public function logout() {
        if ($this->check()) {
            session_unset(); //Libera todas as variáveis de sessão
            session_destroy(); //Destroi a Session
        }
    }

    public function check($submodule = false) {

        if (!isset($_SESSION)) {
            session_start();
        }

        if ($submodule) {
            if (!is_numeric($submodule)) {
                trigger_error("ID do submodulo é inválido");
            }

            $id = isset($_SESSION["id_usuario"]) ? $_SESSION["id_usuario"] : "";
            if (!$id){return false;}

            $result = $this->DB->execPreparedSQL("SELECT * FROM
usuarios_privilegios WHERE fk_usuarios = ? AND submodulo_privilegio = ?", array('ii',
$this->DB->escape($id), $this->DB->escape($submodule)));
            if (empty($result)) return false;

            global $somenteleitura;
            $somenteleitura = $result[0]['somenteleitura_privilegio'];

            return true;
        }
        else {
            if (isset($_SESSION["id_usuario"])) {
                return true;
            }
            return false;
        }
    }

```

```

    }
}

<?php
require_once(dirname(__FILE__) . '/../config/padrao.php');

class DB {

    private $connection;
    private $host;
    private $user;
    private $password;
    private $database;
    private static $instance;

    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new DB();
        }
        return self::$instance;
    }

    private function __construct() {
        global $config;
        $this->host = $config['mysql_ip'];
        $this->user = $config['mysql_usuario'];
        $this->password = $config['mysql_senha'];
        $this->database = $config['mysql_base'];
        $this->connect();
    }

    public function __destruct() {
        $this->disconnect();
    }

    public function connect() {
        $this->connection = new mysqli($this->host, $this->user, $this->password,
$this->database);
        if (mysqli_connect_errno()) {
            echo "Problema com o banco de dados. Contate o administrador do
sistema.";
            exit();
        }
    }

    public function disconnect() {
        if (is_resource($this->connection)) {
            $this->connection->close();
        }
    }

    public function query($query) {
        if (!$result = $this->connection->query($query)) {
            trigger_error("MySQLi Query: " . $this->connection->error);
        }
        return $result;
    }
}

```

```

public function escape($var){
    if (!$this->connection->ping()) { $this->connect(); }

    if (get_magic_quotes_gpc()) {
        $var = stripslashes($var);
    }

    return utf8_decode(anti_injection($this->connection-
>real_escape_string($var)));
}

public function autocommit($boolean){
    if (!$this->connection->ping()) { $this->connect(); }
    $this->connection->autocommit($boolean);
}

public function commit(){
    if (!$this->connection->ping()) { $this->connect(); }
    $this->connection->commit();
}

public function rollback(){
    if (!$this->connection->ping()) { $this->connect(); }
    $this->connection->rollback();
}

public function last_id(){
    if (!$this->connection->ping()) { $this->connect(); }
    return $this->connection->insert_id;
}

public function getError(){
    return $this->connection->error;
}

//Usage:
//
//execSQL($sql, $parameters, $close);
//
//$sql = Statement to execute;
//$parameters = array of type and values of the parameters (if any)
//$close = true to close $stmt (in inserts) false to return an array with the
values;
//
//Examples:
//
//execPreparedSQL("SELECT * FROM table WHERE id = ?", array('i', $id), false);
//execPreparedSQL("SELECT * FROM table", array(), false);
//execPreparedSQL("INSERT INTO table(id, name) VALUES (?,?)", array('ss', $id,
$name), true);

//Character    Description
//i    corresponding variable has type integer
//d    corresponding variable has type double
//s    corresponding variable has type string
//b    corresponding variable is a blob and will be sent in packets

function execPreparedSQL($sql, $params, $close = false){

```

```

$params_text = print_r($params, true);

if (!$this->connection->ping()) { die(trigger_error("Houve uma desconexão com
o banco de dados. Repita a operação.")); }

$stmt = $this->connection->prepare($sql) or die(trigger_error("Failed to
prepared the statement: ".mysqli_error($this->connection)." -- Last Query: $sql -- Params:
$params_text"));

if ($params){
    call_user_func_array(array($stmt, 'bind_param'),$this-
>refValues($params));
}

$stmt->execute();
if ($stmt->errno) {
    trigger_error("MySQLi Query: [$stmt->errno] " . $this->connection-
>error. " -- Last Query: '$sql' -- Params: $params_text");
}

$results = array();

if($close){
    $result = $this->connection->affected_rows;
} else {
    $meta = $stmt->result_metadata();

    while ( $field = $meta->fetch_field() ) {
        $parameters[] = &$row[$field->name];
    }

    call_user_func_array(array($stmt, 'bind_result'),$this-
>refValues($parameters));

    while ( $stmt->fetch() ) {
        $x = array();
        foreach( $row as $key => $val ) {
            $x[$key] = utf8_encode($val);
        }
        $results[] = $x;
    }

    $result = $results;
}

$stmt->close();

return $result;
}

function refValues($arr){
    if (strnatcmp(PHP_VERSION(),'5.3') >= 0) //Reference is required for PHP 5.3+
    {
        $refs = array();
        foreach($arr as $key => $value)
            $refs[$key] = &$arr[$key];
        return $refs;
    }
}

```



```

        return $arr;
    }

function getTypeLabel(array $args, $update = false)
{
    $return = '';
    foreach ($args as $input) {
        if (is_int($input)) $return .= 'i';
        else if (is_double($input) || is_float($input)) $return .= 'd';
        else if (is_string($input)) $return .= 's';
        else $return .= 'b';
    }

    if ($update){ $return .= 'i'; }
    return $return;
}
}

?>

<?php

require_once(dirname(__FILE__) . '/../config/constantes.php');
require_once(APPLICATION_PATH . "/config/padiao.php");

function myErrorHandler($errno, $errstr, $errfile, $errline)
{
    $Logger = Logger::getInstance();
    $error_msg = ReturnErrorType($errno).": $errstr -- $errfile -- $errline";
    $Logger->log($error_msg, $errno);
    /* Don't execute PHP internal error handler */
    //header("Location: ".URI_ERRO.'error.php');
    exit('<p class="error">Ocorreu um erro inesperado.<br />Entre em contato com o
administrador do sistema.</p>');
}

register_shutdown_function('shutDownFunction');
function shutDownFunction() {

    $Logger = Logger::getInstance();
    $error = error_get_last();
    if ($error){
        $error_msg = ReturnErrorType($error['type']).": {$error['message']} --
{$error['file']} -- {$error['line']}";
        $Logger->log($error_msg, 1);
        //header("Location: ".URI_ERRO.'error.php');
    }

}

function ReturnErrorType($type)
{
    switch($type)
    {
        case E_ERROR:           return 'E_ERROR'; // 1 //
        case E_WARNING:        return 'E_WARNING'; // 2 //
        case E_PARSE:          return 'E_PARSE'; // 4 //
        case E_NOTICE:         return 'E_NOTICE'; // 8 //
        case E_CORE_ERROR:     return 'E_CORE_ERROR'; // 16 //
    }
}

```

```

        case E_CORE_WARNING:          return 'E_CORE_WARNING'; // 32 //
        case E_CORE_ERROR:           return 'E_COMPILE_ERROR'; // 64 //
        case E_COMPILE_WARNING:      return 'E_COMPILE_WARNING'; // 128 //
        case E_USER_ERROR:           return 'E_USER_ERROR'; // 256 //
        case E_USER_WARNING:         return 'E_USER_WARNING'; // 512 //
        case E_USER_NOTICE:          return 'E_USER_NOTICE'; // 1024 //
        case E_STRICT:                return 'E_STRICT'; // 2048 //
        case E_RECOVERABLE_ERROR:     return 'E_RECOVERABLE_ERROR'; // 4096 //
        case E_DEPRECATED:           return 'E_DEPRECATED'; // 8192 //
        case E_USER_DEPRECATED:      return 'E_USER_DEPRECATED'; // 16384 //
    }
    return $type;
}
}

<?php

function imprimir_erro($message = "") {
    ob_clean();
    require_once(dirname(__FILE__) . "/TemplateHandler.php");
    $template->set_content_file('admin/usuarios/usuarios.php');
    $template = new TemplateHandler();
    $template->set_body_file(TEMPLATE_GLOBAL);
    $template->set_title(TITULO_PADRAO . " - Erro");
    $template->show();
}

function unauthorized() {
    if (!headers_sent()) {
        header("Status: 403");
        ob_clean();
        require_once(dirname(__FILE__) . "/TemplateHandler.php");
        $template = new TemplateHandler();
        try {
            $template->set_body_file(TEMPLATE_ACESSO_NEGADO);
            $template->set_title(TITULO_PADRAO . " - Não autorizado");
            $template->show();
        }
        catch (Exception $e) {
            exit(1);
        }
    }
    else {
        exit("Não autorizado");
    }
}

function anti_injection($dados) {

    if (is_array($dados)){
        $data = array();
        foreach($dados as $key=>$dado){
            $dado =
preg_replace("/(from|select|update|insert|delete|where|union|drop table|show tables|--
|\\\\)/", "", $dado);
            $dado = trim($dado);
            $dado = strip_tags($dado);
            $dado = addslashes($dado);
            $data[$key] = $dado;
        }
    }
}

```

```

    }
    return $data;
} else {
    //Remove palavras que contenham sintaxe SQL:
    $dado = preg_replace("/(from|select|update|insert|delete|where|union|drop
table|show tables|--|\\\\\\)/", "", $dados);
    $dado = trim($dado); //Limpa espaços vazios no início e fim da string
    $dado = strip_tags($dado); //Remove tags HTML e PHP
    $dado = addslashes($dado); //Adiciona barras invertidas a uma string
    return $dado;
}
}

function format_title_trim($text, $max_length) {
    if (strlen($text) > $max_length) {
        return "<span title='$text'>".mb_substr($text, 0, $max_length, 'UTF-8') .
"...<span>";
    } else {
        return $text;
    }
}

function normalize($string) {
    $table = array(
        'Š'=>'S', 'š'=>'s', 'Đ'=>'Dj', 'đ'=>'dj', 'Ž'=>'Z', 'ž'=>'z', 'Č'=>'C', 'č'=>'c',
'Ć'=>'C', 'ć'=>'c',
        'À'=>'A', 'Á'=>'A', 'Â'=>'A', 'Ã'=>'A', 'Ä'=>'A', 'Å'=>'A', 'Æ'=>'A', 'Ç'=>'C',
'È'=>'E', 'É'=>'E',
        'Ê'=>'E', 'Ë'=>'E', 'Ì'=>'I', 'Í'=>'I', 'Î'=>'I', 'Ï'=>'I', 'Ñ'=>'N', 'Ò'=>'O',
'Ó'=>'O', 'Ô'=>'O',
        'Õ'=>'O', 'Ö'=>'O', 'Ø'=>'O', 'Ù'=>'U', 'Ú'=>'U', 'Û'=>'U', 'Ü'=>'U', 'Ý'=>'Y',
'Þ'=>'B', 'ß'=>'Ss',
        'à'=>'a', 'á'=>'a', 'â'=>'a', 'ã'=>'a', 'ä'=>'a', 'å'=>'a', 'æ'=>'a', 'ç'=>'c',
'è'=>'e', 'é'=>'e',
        'ê'=>'e', 'ë'=>'e', 'ì'=>'i', 'í'=>'i', 'î'=>'i', 'ï'=>'i', 'ð'=>'o', 'ñ'=>'n',
'ò'=>'o', 'ó'=>'o',
        'ô'=>'o', 'õ'=>'o', 'ö'=>'o', 'ø'=>'o', 'ù'=>'u', 'ú'=>'u', 'û'=>'u', 'ý'=>'y',
'ÿ'=>'y', 'þ'=>'b',
        'ÿ'=>'y', 'Ŕ'=>'R', 'ŕ'=>'r',
    );
    return strtr($string, $table);
}

function validaIP($ip) {
    if(filter_var($ip, FILTER_VALIDATE_IP) === FALSE) { return false; } else { return
true; }
}

function validaEmail($email){
    return preg_match("/^[[:alnum:]]+[a-z0-9_-]*@[a-z0-9_-]+\.[a-z]{2,4}$/i", $email);
}

function secToTime($segundos) {
    $h = intval($segundos/3600);
    $m = intval($segundos/60-$h*60);
    $s = intval($segundos-($h*3600+$m*60));

    return sprintf('%02d:%02d:%02d', $h, $m, $s);
}

```

```

function validaData($date) {
    list($dd,$mm,$yy)=explode("/", $date);
    if (is_numeric($yy) && is_numeric($mm) && is_numeric($dd))
    {
        return checkdate($mm,$dd,$yy);
    }
    return false;
}

function validaTime($time){
    return preg_match("/(0?\d|1[0-2]):(0\d|[0-5]\d)/i", $time);
}

?>

<?php
require_once(INCLUDE_PATH ."/DB.php");
require_once(INCLUDE_PATH ."/FirePHPCore/fb.php");
require_once(INCLUDE_PATH ."/PHPMailer/class.phpmailer.php");

class Logger {

    private static $DB;
    private $mail;
    private $date;
    private static $instance;

    public static function getInstance() {
        if (!self::$instance) {
            self::$instance = new Logger();
        }
        return self::$instance;
    }

    private function __construct() {
        $this->date = Date("d-m-Y H:i:s");
    }

    public function log($message, $lvl = 0) {

        global $config;

        ob_start();

        if ($config['firephp_log_enabled']) {
            $this->logtoFB($message);
        }

        if (is_array($message)){
            $message = print_r($message,true);
        }

        if ($config['file_log_enabled'] && (($lvl & $config['file_log_level']) ||
        $lvl == 0)) {
            $this->logtoFILE($message);
        }
    }
}

```

```

        if ($config['database_log_enabled'] && ($lvl &
$config['database_log_level'])) {
            $this->logtoDATABASE($message);
        }
    }

    public function logACTION($action, $message){
        if (!mysqli_connect_errno()) {
            self::$DB = DB::getInstance();
            $message = $this->decode($message);
            try {
                $res = self::$DB->execPreparedSQL("select id_log_acao FROM
`logs_acoes` WHERE nome_log_acao = ?", array('s', normalize($action)));
                if (!empty($res)){
                    $id_acao = $res[0]['id_log_acao'];
                    $iduser = isset($_SESSION["id_usuario"]) ?
$_SESSION["id_usuario"]: null;
                    $ip = $_SERVER["REMOTE_ADDR"];
                    $res = self::$DB->execPreparedSQL("INSERT INTO `logs`
(id_usuario, ip_log, acao_log, descricao_log) VALUES (?, ?, ?, ?)", array('isis', $iduser,
$ip, $id_acao, $message), true);
                    return true;
                } else {
                    $this->logtoFILE("WARNING: logACTION: $action ==>
id_log_acao não encontrado.");
                }
            } catch (Exception $e) {
                $this->logtoFILE("ERRO: logACTION: ".$e->getMessage());
            }
        }
        return false;
    }

    protected function logtoFB($message){
        FB::log($message);
    }

    protected function logtoDATABASE($message){
        if (!mysqli_connect_errno()) {
            self::$DB = DB::getInstance();
            $message = $this->decode($message);
            self::$DB->execPreparedSQL("INSERT INTO `sistema_error_logs`
(log_message) VALUES (?)", array('s', $message), true);
        }
    }

    protected function logtoFILE($message){
        global $config;
        $fp = fopen($config['file_log_location'], 'a+');
        fwrite($fp, $this->date." ==> ".$this->decode($message)."\n");
        fclose($fp);
    }

    protected function decode($message){
        return ($this->is_utf8($message)) ? utf8_decode($message) : $message;
    }

    protected function is_utf8($string) {

```

```

return preg_match('%^(?:
    [\x09\x0A\x0D\x20-\x7E]          # ASCII
    | [\xC2-\xDF][\x80-\xBF]       # non-overlong 2-byte
    | \xE0[\xA0-\xBF][\x80-\xBF]   # excluding overlongs
    | [\xE1-\xEC\xEE\xEF][\x80-\xBF]{2} # straight 3-byte
    | \xED[\x80-\x9F][\x80-\xBF]   # excluding surrogates
    | \xF0[\x90-\xBF][\x80-\xBF]{2} # planes 1-3
    | [\xF1-\xF3][\x80-\xBF]{3}    # planes 4-15
    | \xF4[\x80-\x8F][\x80-\xBF]{2} # plane 16
)*%xs', $string);
}
?>

<?php

require_once("Auth.php");

class TemplateHandler {

    private $global_file;
    private $header_file;
    private $body_file;
    private $footer_file;
    private $content_file;
    private $menu_file;
    private $modules_tabs_file;
    private $title;
    private $css_files;
    private $js_files;
    private $module;
    private $sub_module;
    private $sub_module_item;
    private $auth;
    private $authenticated_only;

    public function TemplateHandler() {
        $this->global_file = TEMPLATE_GLOBAL;
        $this->header_file = TEMPLATE_CABECALHO;
        $this->body_file = TEMPLATE_CORPO;
        $this->footer_file = TEMPLATE_RODAPE;
        $this->menu_file = TEMPLATE_MENU;
        $this->title = TITULO_PADRAO;
        $this->css_files = array();
        $this->js_files = array();
        $this->module = 0;
        $this->sub_module = 0;
        $this->sub_module_item = 0;
        $this->authenticated_only = false;
        $this->auth = new Auth();
    }

    public function show() {
        if ($this->authenticated_only && !$this->auth->check($this->sub_module_item))
        {
            $this->body_file = TEMPLATE_ACESSO_NEGADO;
        }
        if ($this->template_file_exists($this->global_file)) {
            include(CAMINHO_RAIZ . "/template/" . $this->global_file);
        }
    }
}

```

```

    }
    else {
        throw new Exception("TemplateHandler: Arquivo do template global não
encontrado.");
    }
}

public function show_header() {
    if ($this->template_file_exists($this->header_file)) {
        include(CAMINHO_RAIZ . "/template/" . $this->header_file);
    }
}

public function show_body() {
    if ($this->template_file_exists($this->body_file)) {
        include(CAMINHO_RAIZ . "/template/" . $this->body_file);
    }
}

public function show_footer() {
    if ($this->template_file_exists($this->footer_file)) {
        include(CAMINHO_RAIZ . "/template/" . $this->footer_file);
    }
}

public function show_content() {
    if ($this->public_file_exists($this->content_file)) {
        include(CAMINHO_PUBLICO . "/" . $this->content_file);
    }
}

public function show_menu() {
    if ($this->template_file_exists($this->menu_file)) {
        include(CAMINHO_RAIZ . "/template/" . $this->menu_file);
    }
}

public function import_css() {
    if ($this->css_files) {
        foreach($this->css_files as $css_file) {
            if ($this->is_external($css_file)) {
                echo '<link rel="stylesheet" type="text/css" href="'.
$css_file . '" />';
            }
            else {
                echo '<link rel="stylesheet" type="text/css" href="'.
URI_CSS . $css_file . '" />';
            }
        }
    }
}

public function import_js() {
    if ($this->js_files) {
        foreach($this->js_files as $js_file) {
            if ($this->is_external($js_file)) {
                echo '<script type="text/javascript" src="' . $js_file
. '></script>';
            }
        }
    }
}

```

```

        else {
            echo '<script type="text/javascript" src="'. URI_JS .
$js_file . '></script>';
        }
    }
}

public function is_external($file) {
    if (substr($file, 0, 4) == "http" || substr($file, 0, 3) == "www") {
        return true;
    }
    return false;
}

private function template_file_exists($file) {
    if ($file) {
        return file_exists(CAMINHO_RAIZ . "/template/" . $file);
    }
    return false;
}

private function public_file_exists($file) {
    if ($file) {
        return file_exists(CAMINHO_PUBLICO . "/" . $file);
    }
    return false;
}

public function get_header_file() {
    return $this->header_file;
}

public function set_header_file($header_file) {
    if ($this->template_file_exists($header_file)) {
        $this->header_file = $header_file;
    }
    else {
        trigger_error("TemplateHandler: Arquivo de cabeçalho não existe: " .
CAMINHO_RAIZ . "/" . $header_file);
    }
}

public function get_body_file() {
    return $this->body_file;
}

public function set_body_file($body_file) {
    if ($this->template_file_exists($body_file)) {
        $this->body_file = $body_file;
    }
    else {
        trigger_error("TemplateHandler: Arquivo do corpo não existe: " .
CAMINHO_RAIZ . "/" . $body_file);
    }
}

public function get_footer_file() {
    return $this->footer_file;
}

```



```

    }

    public function set_footer_file($footer_file) {
        if ($this->template_file_exists($footer_file)) {
            $this->footer_file = $footer_file;
        }
        else {
            trigger_error("TemplateHandler: Arquivo do cabeçalho não existe: " .
CAMINHO_RAIZ . "/" . $footer_file);
        }
    }

    public function get_content_file() {
        return $this->content_file;
    }

    public function set_content_file($content_file) {
        if ($this->public_file_exists($content_file)) {
            $this->content_file = $content_file;
        }
        else {
            trigger_error("TemplateHandler: Arquivo do conteúdo não existe: " .
CAMINHO_PUBLICO . "/" . $content_file);
        }
    }

    public function get_menu_file() {
        return $this->menu_file;
    }

    public function set_menu_file($menu_file) {
        if ($this->template_file_exists($menu_file)) {
            $this->menu_file = $menu_file;
        }
        else {
            trigger_error("TemplateHandler: Arquivo do menu não existe: " .
CAMINHO_RAIZ . "/" . $menu_file);
        }
    }

    public function get_title() {
        return $this->title;
    }

    public function set_title($title) {
        $this->title = $title;
    }

    public function get_css_files() {
        return $this->css_files;
    }

    public function set_css_files($css_files) {
        if (is_array($css_files)) {
            $this->css_files = $css_files;
        }
        else {
            throw new Exception("TemplateHandler: Os arquivos CSS devem ser um
array.");
        }
    }

```

```

    }
}

public function get_js_files() {
    return $this->js_files;
}

public function set_js_files($js_files) {
    if (is_array($js_files)) {
        $this->js_files = $js_files;
    }
    else {
        trigger_error("TemplateHandler: Os arquivos JS devem ser um array.");
    }
}

public function get_module() {
    return $this->module;
}

public function set_module($module) {
    if (is_numeric($module)) {
        $this->module = $module;
    }
    else {
        trigger_error("TemplateHandler: O ID do módulo deve ser um número
inteiro");
    }
}

public function get_sub_module() {
    return $this->sub_module;
}

public function set_sub_module($sub_module) {
    if (is_numeric($sub_module)) {
        $this->sub_module = $sub_module;
    }
    else {
        trigger_error("TemplateHandler: O ID do submódulo deve ser um número
inteiro");
    }
}

public function get_sub_module_item() {
    return $this->sub_module_item;
}

public function set_sub_module_item($sub_module_item) {
    if (is_numeric($sub_module_item)) {
        $this->sub_module_item = $sub_module_item;
    }
    else {
        trigger_error("TemplateHandler: O ID do item do submódulo deve ser um
número inteiro");
    }
}

public function is_authenticated_only() {

```

```

        return $this->authenticated_only;
    }

    public function set_authenticated_only($authenticated_only) {
        if (is_bool($authenticated_only)) {
            $this->authenticated_only = $authenticated_only;
        }
        else {
            trigger_error("TemplateHandler: Somente autenticado deve ser um
booleano");
        }
    }
}

<?php

require_once(dirname(__FILE__) . '/../config/padrao.php');
require_once(INCLUDE_PATH . "DB.php");

$DB = DB::getInstance();
$Logger = Logger::getInstance();

$Logger->log("Executando script de monitoramento...");

$result = $DB->execPreparedSQL("SELECT id_acordo, fk_dispositivos, ip_disp_alvo,
ping_count,
param_ports, ip_neighbour, comunidade_snmp_dispositivo
fk_acordos_mon_tipos, ping_size,
FROM monitoramentos m
LEFT JOIN dispositivos d ON
m.`fk_dispositivos`=d.`id_dispositivo`
LEFT JOIN acordos a ON
m.`fk_acordos`=a.`id_acordo`
LEFT JOIN mon_param_ping mpp ON
m.`id_mon`=mpp.`fk_monitoramentos`
LEFT JOIN mon_param_snmpi mpsi ON
m.`id_mon`=mpsi.`fk_monitoramentos`
LEFT JOIN mon_param_snmpo mpso ON
m.`id_mon`=mpso.`fk_monitoramentos`", array());

foreach ($result as $mon) {

    $id_dis = $mon['fk_dispositivos'];
    $tipo = $mon['fk_acordos_mon_tipos'];
    $ip_alvo = $mon['ip_disp_alvo'];
    $id_acordo = $mon['id_acordo'];

    switch ($tipo){

        case 1:
            //processar o ping
            $ping_count = ($mon['ping_count']) ? $mon['ping_count'] : 10;
            $ping_size = ($mon['ping_size']) ? $mon['ping_size'] : 1000;
            system("/usr/bin/php /var/www/application/scripts/pingget.php
\"\".$id_dis.\"\" \"\".$tipo.\"\" \"\".$ip_alvo.\"\" \"\".$ping_count.\"\" \"\".$ping_size.\"\"
\"\".$id_acordo.\"\"> /dev/null &");

            break;
    }
}

```

```

        case 2:
            $comunidade = $mon['comunidade_snmp_dispositivo'];
            $ports = explode("|", $mon['param_ports']);
            foreach ($ports as $port) {
                $tmp = explode(":", $port);
                $port = $tmp[0];
                $port_name = $tmp[1];
                system("/usr/bin/php /var/www/application/scripts/snmpget.php
\"\".\"$id_dis.\"\" \"\".\"$tipo.\"\" \"\".\"$port.\"\" \"\".\"$port_name.\"\" \"\".\"$comunidade.\"\"
\"\".\"$ip_alvo.\"\" \"\".\"$id_acordo.\"\"> /dev/null &");
            }
            break;

        case 3:
            //processar o ospf
            $comunidade = $mon['comunidade_snmp_dispositivo'];
            $ip_neighbour = $mon['ip_neighbour'];
            system("/usr/bin/php /var/www/application/scripts/ospfget.php
\"\".\"$id_dis.\"\" \"\".\"$tipo.\"\" \"\".\"$comunidade.\"\" \"\".\"$ip_alvo.\"\" \"\".\"$id_acordo.\"\"
\"\".\"$ip_neighbour.\"\" > /dev/null &");
            break;
    }
}

$Logger->log("Execução do script de monitoramento finalizada.");

?>

<?php
require_once(dirname(__FILE__) . '/../config/padrao.php');
require_once(INCLUDE_PATH . "DB.php");

$DB = DB::getInstance();
$Logger = Logger::getInstance();

$id_dis = $argv[1];
$tipo = $argv[2];
$comunidade = $argv[3];
$ip = $argv[4];
$acordo = $argv[5];
$ip_neighbour = $argv[6];

$time = time();

$arrayInfo['fk_dispositivos'] = $id_dis;
$arrayInfo['fk_acordos'] = $acordo;
$arrayInfo['oid_ip'] = $ip;
$arrayInfo['end_time'] = $time;
$arrayInfo['type'] = $tipo;
$arrayInfo['port_name_neighbour'] = $ip_neighbour;

$response = `usr/bin/snmpget -t 1 -v 2c -c $comunidade $ip
${config['OSPF']}.ip_neighbour.0`;
$status_op = "NA";

```

```

if (preg_match("/No Such Instance|Timeout/i", $response)){
    $Logger->log("ERRO: snmpget.php Dispositivo: $id_dis IP: $ip --> $response");
} else if (!$response) {
    $status_op = "UNKNOWN";
} else {
    $response = preg_match("/:(.*)/", $response, $matches);
    $value = trim($matches[1]);
    if ($value >= $config['OSPF_expectedMinValue']){
        $status_op = 'up';
    } else {
        $status_op = 'down';
    }
}

$result = $DB->execPreparedSQL("SELECT id_med, oper_status, end_time FROM
medicoes_disponibilidade ms where fk_dispositivos = ? and fk_acordos = ? and type = ? and
oid_ip = ? and port_name_neighbour = ? order by end_time desc", array("iiiiss", $DB-
>escape($id_dis), $DB->escape($acordo), $DB->escape($tipo), $DB->escape($ip), $DB-
>escape($ip_neighbour));
$last_status = (isset($result[0]['oper_status'])) ? ($result[0]['oper_status']) : "";
$last_time = (isset($result[0]['end_time'])) ? ($result[0]['end_time']) : "";
$id_med = (isset($result[0]['id_med'])) ? ($result[0]['id_med']) : "";

$arrayInfo['oper_status'] = "$status_op";

if (!$last_status || $last_status != $status_op){

    #INSERIR NOVA ENTRADA
    $arrayInfo['start_time'] = $time;

    $params[] = $DB->getTypeLabel($arrayInfo);
    foreach ($arrayInfo as $key => $valor) {
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $DB->escape($valor);
    }

    $result = $DB->execPreparedSQL("INSERT INTO medicoes_disponibilidade
(`.implode(",`,`,$chaves_sql).`) VALUES (".implode(",`,`,$posicoes).`)", $params, true);

} else {

    #ATUALIZAR A ENTRADA ANTIGA
    //$result = $DB->execPreparedSQL("UPDATE medicoes_disponibilidade SET end_time = ?
WHERE fk_dispositivos = ? and fk_acordos = ? and type = ? and oid_ip = ? and
port_name_neighbour = ? and end_time = ?", array("iiiissi", $DB->escape($time), $DB-
>escape($id_dis), $DB->escape($acordo), $DB->escape($tipo), $DB->escape($ip), $DB-
>escape($ip_neighbour), $DB->escape($last_time)), true);
    $result = $DB->execPreparedSQL("UPDATE medicoes_disponibilidade SET end_time = ?
WHERE id_med = ?", array("ii", $DB->escape($time), $DB->escape($id_med)), true);
}

?>

<?php

require_once(dirname(__FILE__) . '/../config/padrao.php');
require_once(INCLUDE_PATH . "DB.php");

```

```

$DB = DB::getInstance();
$Logger = Logger::getInstance();

$id_dis = $argv[1];
$tipo = $argv[2];
$ip = $argv[3];
$ping_count = $argv[4];
$ping_size = $argv[5];
$acordo = $argv[6];

$ipversion = (strpos($ip, ":") ? 6 : "");
$time = time();

$arrayInfo['fk_dispositivos'] = $id_dis;
$arrayInfo['end_time'] = $time;
$arrayInfo['type'] = $tipo;
$arrayInfo['oid_ip'] = $ip;
$arrayInfo['fk_acordos'] = $acordo;

//$Logger->log("bin/fping$ipversion -c $ping_count -b $ping_size $ip");
$response = `/usr/bin/fping$ipversion -c $ping_count -b $ping_size $ip 2>&1`;
$status_op = "down";

$tmp_resp = explode("\n", $response);
$jitter_count = 0;
$jitter_total = 0;
$last_ping = 0;
$jitter_final = "NA";
foreach ($tmp_resp as $line){
    $tmp_line = explode(" ", $line);
    if (preg_match("/\[i", $line)){
        if (!$last_ping){
            $last_ping = $tmp_line[5];
        } else {
            $jitter_total += abs($tmp_line[5] - $last_ping);
            $last_ping = $tmp_line[5];
            $jitter_count++;
        }
    } elseif (preg_match("/xmt/rcv/%loss/i", $line)){
        $pacotes = explode("/", str_replace(",","",$tmp_line[4]));
        $pct_enviados = $pacotes[0];
        $pct_recebidos = $pacotes[1];
        if (isset($tmp_line[7])) {
            $status_op = "up";
            $latencias = explode("/",$tmp_line[7]);
            $latmin = $latencias[0];
            $latmed = $latencias[1];
            $latmax = $latencias[2];
        }
    }
}

if ($jitter_count > 0){
    $jitter_final = sprintf("%.2f",$jitter_total/$jitter_count);
}

$arrayInfoExtra['fk_dispositivos'] = $id_dis;
$arrayInfoExtra['fk_acordos'] = $acordo;

```

```

$arrayInfoExtra['ip'] = $ip;
$arrayInfoExtra['pacotes_enviados'] = (isset($pct_enviados)) ? $pct_enviados : 0;
$arrayInfoExtra['pacotes_recebidos'] = (isset($pct_recebidos)) ? $pct_recebidos : 0;
$arrayInfoExtra['latencia_min'] = (isset($latmin)) ? $latmin : 0;
$arrayInfoExtra['latencia_med'] = (isset($latmed)) ? $latmed : 0;
$arrayInfoExtra['latencia_max'] = (isset($latmax)) ? $latmax : 0;
$arrayInfoExtra['jitter'] = $jitter_final;
$arrayInfoExtra['timevalue'] = $time;

$result = $DB->execPreparedSQL("SELECT id_med, oper_status, end_time FROM
medicoes_disponibilidade ms where fk_dispositivos = ? and fk_acordos = ? and type = ? and
oid_ip = ? order by end_time desc", array("iiis", $DB->escape($id_dis), $DB-
>escape($acordo), $DB->escape($tipo), $DB->escape($ip)));
$last_status = (isset($result[0]['oper_status'])) ? ($result[0]['oper_status']) : "";
$last_time = (isset($result[0]['end_time'])) ? ($result[0]['end_time']) : "";
$id_med = (isset($result[0]['id_med'])) ? ($result[0]['id_med']) : "";

$arrayInfo['oper_status'] = "$status_op";

if (!$last_status || $last_status != $status_op){

    #INSERIR NOVA ENTRADA
    $arrayInfo['start_time'] = $time;

    $params[] = $DB->getTypeLabel($arrayInfo);
    foreach ($arrayInfo as $key => $valor) {
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $DB->escape($valor);
    }

    $result = $DB->execPreparedSQL("INSERT INTO medicoes_disponibilidade
(`".implode("`",`", $chaves_sql).`) VALUES (`".implode("`", $posicoes).`)", $params, true);

} else {

    #ATUALIZAR A ENTRADA ANTIGA
    $result = $DB->execPreparedSQL("UPDATE medicoes_disponibilidade SET end_time = ?
WHERE id_med = ?", array("ii", $DB->escape($time), $DB->escape($id_med)), true);
}

$paramsX[] = $DB->getTypeLabel($arrayInfoExtra);
foreach ($arrayInfoExtra as $key => $valor) {
    $chaves_sqlX[] = $key;
    $posicoesX[] = "?";
    $paramsX[] = $DB->escape($valor);
}
$result = $DB->execPreparedSQL("INSERT INTO medicoes_lat_alcan
(`".implode("`",`", $chaves_sqlX).`) VALUES (`".implode("`", $posicoesX).`)", $paramsX, true);

?>

<?php

require_once(dirname(__FILE__) . '/../config/padrao.php');
require_once(INCLUDE_PATH . "DB.php");

$DB = DB::getInstance();
$Logger = Logger::getInstance();

```

```

$id_dis = $argv[1];
$tipo = $argv[2];
$port = $argv[3];
$port_name = $argv[4];
$comunidade = $argv[5];
$ip = $argv[6];
$acordo = $argv[7];

$time = time();

$arrayInfo['fk_dispositivos'] = $id_dis;
$arrayInfo['fk_acordos'] = $acordo;
$arrayInfo['oid_ip'] = $port;
$arrayInfo['port_name_neighbour'] = $port_name;
$arrayInfo['end_time'] = $time;
$arrayInfo['type'] = $tipo;

$response = `usr/bin/snmpget -t 1 -v 2c -c $comunidade $ip
${config['ifOperStatus']}.${port}`;
$status_op = "NA";

if (preg_match("/No Such Instance/i", $response)){
    $Logger->log("ERRO: snmpget.php Dispositivo: $id_dis Port: $port Port Name:
$port_name IP: $ip --> $response");
} else if (!$response) {
    $status_op = "UNKNOWN";
} else {
    $response = preg_match("/:(.*)/", $response, $matches);
    $value = trim($matches[1]);
    if ($value == $config['ifOperStatus_expectedValue']){
        $status_op = 'up';
    } else {
        $status_op = 'down';
    }
}

$result = $DB->execPreparedSQL("SELECT id_med, oper_status, end_time FROM
medicoes_disponibilidade ms where fk_dispositivos = ? and fk_acordos = ? and type = ? and
oid_ip = ? and port_name_neighbour = ? order by end_time desc", array("iiiiis", $DB-
>escape($id_dis), $DB->escape($acordo), $DB->escape($tipo), $DB->escape($port), $DB-
>escape($port_name));
$last_status = (isset($result[0]['oper_status'])) ? ($result[0]['oper_status']) : "";
$last_time = (isset($result[0]['end_time'])) ? ($result[0]['end_time']) : "";
$id_med = (isset($result[0]['id_med'])) ? ($result[0]['id_med']) : "";

$arrayInfo['oper_status'] = "$status_op";

if (!$last_status || $last_status != $status_op){
    #INSERIR NOVA ENTRADA
    $arrayInfo['start_time'] = $time;

    $params[] = $DB->getTypeLabel($arrayInfo);
    foreach ($arrayInfo as $key => $valor) {
        $chaves_sql[] = $key;
        $posicoes[] = "?";
        $params[] = $DB->escape($valor);
    }
}

```



```

        $result = $DB->execPreparedSQL("INSERT INTO medicoes_disponibilidade
        (".implode("``,`",$chaves_sql).`) VALUES (".implode(",",$posicoes).")", $params, true);
    } else {

        #ATUALIZAR A ENTRADA ANTIGA
        $result = $DB->execPreparedSQL("UPDATE medicoes_disponibilidade SET end_time = ?
        WHERE id_med = ?", array("ii", $DB->escape($time), $DB->escape($id_med)), true);
    }

    ?>

<?php
ob_start();

?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title><?php echo $this->get_title(); ?></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>global.css" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>jquery-ui.custom.css" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>ui.jqgrid.css" />
<link rel="stylesheet" type="text/css" href="<?php echo URI_CSS; ?>chosen.css" />
<!--[if IE]>
<style type="text/css">
.menu_options {
    border-bottom: 1px solid #DBDBDB;
}
</style>
<![endif]-->
<?php
$this->import_css();
?>

<script type="text/javascript" src="<?php echo URI_JS; ?>functions.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery-1.8.3.min.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery-ui.custom.min.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery.form.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>jquery.jqGrid.min.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>grid.locale-pt-br.js"></script>
<script type="text/javascript" src="<?php echo URI_JS; ?>chosen.jquery.min.js"></script>

<?php
$this->import_js();
?>
</head>
<body>
<div id="main">
    <div id="header">
        <div id="header_content">
            <?php $this->show_header(); ?>
        </div>
    </div>
    <div id="body">

```

```

        <?php $this->show_body(); ?>
    </div>
</div>
<div id="footer">
    <div id="footer_content">
        <?php $this->show_footer(); ?>
    </div>
</div>
</body>
</html>
<?php ob_end_flush(); ?>

<?php
require_once(INCLUDE_PATH . "DB.php");
$DB = DB::getInstance();

$id_user = $_SESSION["id_usuario"];
$privilegios_modulo_sum = 0;
$privilegios_submodulo_sum = 0;

$result_priv = $DB->execPreparedSQL("SELECT fk_usuarios,SUM(res.modulo_privilegio) AS
priv_mod,SUM(res.soma_sub) AS priv_submod FROM
                                (SELECT DISTINCT
modulo_privilegio,SUM(submodulo_privilegio) AS soma_sub,fk_usuarios FROM
usuarios_privilegios
                                WHERE fk_usuarios=
? GROUP BY modulo_privilegio) res", array('i', $DB->escape($id_user)));

foreach ($result_priv as $res){
    $privilegios_modulo_sum = $res["priv_mod"];
    $privilegios_submodulo_sum = $res["priv_submod"];
}

$visibilidade = array();
$result_vis = $DB->execPreparedSQL("SELECT modulo_privilegio,visibilidade_privilegio FROM
usuarios_privilegios
                                WHERE fk_usuarios= ?",
array('i', $DB->escape($id_user)));

foreach ($result_vis as $res){
    $visibilidade[$res["modulo_privilegio"]] = $res["visibilidade_privilegio"];
}

$result_privilegios = $DB->execPreparedSQL("SELECT id_modulo, id_submodulo, nome_modulo,
nome_submodulo
                                FROM sistema_modulos m
                                JOIN sistema_submodulos sb
ON m.`id_modulo`=sb.`fk_sistema_modulos`
                                JOIN usuarios_privilegios
ON sb.`id_submodulo`= usuarios_privilegios.submodulo_privilegio
                                WHERE fk_usuarios = ?
ORDER BY posicao_privilegio,id_submodulo", array('i', $DB->escape($id_user)));

$modulos = array();
$privilegios_modulo = array();
$privilegios_submodulo = array();

foreach ($result_privilegios as $res){

```

```

$privilegio = false;
$privilegio_sub = false;

if ((int)$res["id_modulo"] & (int)$privilegios_modulo_sum) {
    $privilegio = true;
}

if ((int)$res["id_submodulo"] & (int)$privilegios_submodulo_sum) {
    $privilegio_sub = true;
}

$nomes_modulos[$res["id_modulo"]] = $res["nome_modulo"];
$nomes_submodulos[$res["id_submodulo"]] = $res["nome_submodulo"];

$privilegios_modulo[$res["id_modulo"]] = $privilegio;
$privilegios_submodulo[$res["id_submodulo"]] = $privilegio_sub;

$modulos_principais[$res["id_modulo"]][] = $res["id_submodulo"];
}

if ($this->get_module() == ID_MODULO_SISTEMA) {

    foreach ($modulos_principais as $id_modulo=>$modulo){

        if($privilegios_modulo[$id_modulo]){

            echo '<div class="menu_box" id="modulo_' . $id_modulo . '"><h4
class="menu_header">' . $nomes_modulos[$id_modulo];

            if ($visibilidade[$id_modulo]) {
                echo '<span class="submodulo_toggle" id="' . $id_modulo . '">-
</span></h4>';
                echo '<ul class="menu_options" id="submodulo_itens_' .
$id_modulo . '">';
            }
            else {
                echo '<span class="submodulo_toggle"
id="' . $id_modulo . '">+</span></h4>';
                echo '<ul class="menu_options" id="submodulo_itens_' .
$id_modulo . '" style="display: none">';
            }
            echo "</h4>";
            echo "<ul class='menu_options'>";

            foreach ($modulo as $id_submodulo){

                if($privilegios_submodulo[$id_submodulo]){

                    switch ($id_submodulo){

                        //Status
                        case ID_SUBMODULO_ADMIN_USUARIOS:
                            $active = ($this->get_sub_module_item()
== ID_SUBMODULO_ADMIN_USUARIOS) ? 'class="active"' : '';
                            echo "<li $active>" . '<a href="' .
URI_ADMIN_USUARIOS . '">' . $nomes_submodulos[$id_submodulo] . '</a></li>';
                            break;
                        case ID_SUBMODULO_ORGANIZACAO_INSTITUICOES:

```

```

                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_ORGANIZACAO_INSTITUICOES) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_ORGANIZACAO_INSTITUICOES . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_ORGANIZACAO_UNIDADES:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_ORGANIZACAO_UNIDADES) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_ORGANIZACAO_UNIDADES . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_GERENCIAVEIS_DISPOSITIVOS . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_GERENCIAVEIS_MARCAS:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_GERENCIAVEIS_MARCAS) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_GERENCIAVEIS_MARCAS . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_GERENCIAVEIS_MODELOS:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_GERENCIAVEIS_MODELOS) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_GERENCIAVEIS_MODELOS . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_ACORDOS_GERENCIAR:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_ACORDOS_GERENCIAR) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_ACORDOS_GERENCIAR . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_MONITORAMENTOS_GERENCIAR:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_MONITORAMENTOS_GERENCIAR) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_MONITORAMENTOS_GERENCIAR . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_MANUTENCOES_REGISTRAR:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_MANUTENCOES_REGISTRAR) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_MANUTENCOES_REGISTRAR . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_ACORDOS_RELATORIOS:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_ACORDOS_RELATORIOS) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_ACORDOS_RELATORIOS . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;
                case ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS:
                $active = ($this->get_sub_module_item()
== ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS) ? 'class="active"' : '';
                echo "<li $active>".<a href="" .
URI_GERENCIAVEIS_CIRCUITOS . "">'. $nomes_submodulos[$id_submodulo].</a></li>';
                break;

```



```

*****/
*
{
    margin: 0;
}

html,body {
    font-size: 0.9em;
    letter-spacing: 0.04em;
    line-height: 1.4em;
    font-family: Tahoma, "Lucida Sans Unicode", Verdana, Arial;
    background: #F4F4F4 url(../_img/bg_top.png) top left repeat-x;
    color: #333;
    margin: 0;
    padding: 0;
    height: 100%;
}

img {
    border: none;
}

/* Headers
*****/
h1,h2 {
    margin: 0;
    font-weight: bold;
}

h1 {
    font-size: 17px;
    line-height: 1.6em;
    font-weight: normal;
}

h2 {
    font-size: 1.4em;
    line-height: 1.5em;
    font-weight: normal;
}

/* Texts
*****/
p {
    margin: 0;
    padding: 0.4em 0;
}

em {
    font-style: italic;
    font-size: 1em;
    color: #808080;
}

a {
    color: #444;
    text-decoration: none;
}

```

```

a:hover {
    color: #000;
    text-decoration: underline;
}

/* Forms
*****/
label {
    font-weight: bold;
}

#fieldset {
    padding: 0 0.9em 1.0em 1.0em;
    margin: 0 0 .7em 0;
    border: 1px solid #ccc;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
}

#fieldset .rounded{
    margin-top: 5px;
}

legend {
    font-weight: bold;
    font-size: 1.1em;
    margin-bottom: 5px;
    margin-left: 10px;
}

input[type=text],input[type=password],input.text,input.title,textarea,select
{
    background-color: #fff;
    border: 1px solid #999;
    color: #444;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
    padding: 2px 3px;
    height: 19px;
}

input[type=text]:focus,input[type=password]:focus,input.text:focus,input.title:focus,textarea:focus,select:focus
{
    border-color: #222;
}

/*input[type=text],*/
input.text,input.title,textarea,select,label {
    margin: 0.20em 0.20;
}

form select {
    height: 22px;
}

input.text,select.text,input.title,textarea {

```

```

        width: 200px;
        padding: 5px;
    }

    input.title {
        font-size: 1em;
    }

    textarea {
        width: 232px;
        height: 100px;
    }

    select.text {
        width: 208px;
    }

    input[type=checkbox],input[type=radio],input.checkbox,input.radio {
        position: relative;
        top: .25em;
    }

    .cbox {
        top: 0 !important;
        margin: 2px;
    }

    input[type=button], input[type=submit], input[type=reset], button {
        background-color: #1F547F;
        color: white;
        padding: 2px 8px;
        margin-bottom: 4px;
        border-bottom: 1px solid #36471D;
        border-right: 1px solid #36471D;
        border-top: 1px solid #6C8F3D;
        border-left: 1px solid #6C8F3D;
        cursor: pointer;
        min-width: 40px;
        -moz-border-radius: 5px;
        -webkit-border-radius: 5px;
        border-radius: 5px;
    }

    input[type=button].disabled, input[type=submit].disabled,input[type=reset].disabled,
    button.disabled {
        color: #ddd;
        opacity: 0.6;
        filter:alpha(opacity=60);
        -khtml-opacity: 0.6;
        -moz-opacity: 0.6;
        cursor: default;
    }

    input[type=text].disabled {
        background-color: #eee;
    }

    form.inline p {
        float: left;
    }

```



```

        display: inline;
        margin-left: 9px;
    }

    .form_table td {
        padding: 2px;
    }

    .form_table td:first-child, .form_table td.Label {
        text-align: right;
        vertical-align: top;
        padding-top: 5px;
    }

    /* Containers
    *****/
    #header {
        width: 100%;
        height: 35px;
        background-color: #1F547F;
        position: absolute;
        top: 0;
        left: 0;
        /*background-color: #165189;*/
    }

    #header_content {
        width: 990px;
        margin: 0 auto;
        padding: 4px 0;
        color: #FFFFFF;
    }

    #header_content h1 {
        line-height: 24px;
        float: left;
    }

    #header_content a {
        color: #E5E5E5;
        text-decoration: none;
    }

    #header_content .modulos {
        float: right;
    }

    #Logout {
        background-color: #2470A5;
        border-radius: 0 0 10px 10px;
        float: right;
        font-size: 85%;
        margin-top: 7px;
        min-width: 175px;
        padding: 2px 10px;
    }

    #Logout input[type=submit] {
        padding: 1px 10px;
    }

```

```

        float: right;
        font-size: 100%;
        background-color: #1F547F;
        border-bottom: 1px solid #2470A5;
        border-right: 1px solid #2470A5;
        border-top: 1px solid #CCCCCC;
        border-left: 1px solid #CCCCCC;
        cursor: pointer;
        min-width: 40px;
        -moz-border-radius: 5px;
        -webkit-border-radius: 5px;
        border-radius: 5px;
    }

#main {
    width: 1090px;
    margin: 0 auto -90px;
    min-height: 100%;
    height: auto !important;
    height: 100%;
}

.submodulo_toggle {
    float: right;
    font-size: 18px;
    font-weight: bold;
    cursor: pointer;
}

#body {
    padding: 65px 0 80px 0;
}

#left_side {
    float: left;
    width: 160px;
}

#right_side {
    float: right;
    background-color: #FEFEFE;
    border: 1px solid #DBDBDB;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    width: 890px;
    /*min-height: 547px;*/
    padding: 10px;
}

#footer {
    background: url(../_img/bg_footer.png) bottom left repeat-x;
    border-bottom: 5px solid #1F547F;
    padding: 20px 0;
    width: 100%;
    height: 45px;
    clear: both;
}

```

```

#footer_content {
    width: 990px;
    margin: 0 auto;
}

#footer_left {
    float: left;
    width: 200px;
    font-size: 80%;
}

#footer_right {
    float: right;
    text-align: right;
}

/* Menu
*****
.menu_box {
    background-color: #EBEBEB;
    /*background-color: #CFDDE8;*/
    border: 1px solid #DBDBDB;
    margin-bottom: 20px;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
}

.menu_header {
    background: #1F547F url(../_img/bg_menu_header.png) top left repeat-x;
    /*background-color: #165189;*/
    color: #EDEDDE;
    margin: 0;
    padding: 6px 8px;
    border-radius-topleft: 10px;
    border-radius-topright: 10px;
    -moz-border-radius-topleft: 10px;
    -moz-border-radius-topright: 10px;
    -webkit-border-radius-topleft: 10px;
    -webkit-border-radius-topright: 10px;
}

.menu_options {
    margin: 0;
    padding: 0;
    list-style-type: none;
}

.menu_options li {
    padding: 4px 8px;
    border-top: 1px solid #E5E5E5;
}

.menu_options li:last-child {
    border-bottom-left-radius: 10px;
    border-bottom-right-radius: 10px;
}

.menu_options a {

```

```

        color: #666;
        text-decoration: none;
        display: block;
    }

    #left_side .active a {
        font-weight: bold;
        color: #333;
    }

    .menu_options li:hover,#left_side .active {
        background-color: #F7F7F7;
    }

    .clear {
        clear: both;
        padding: 0px;
        margin: 0px;
        font-size: 1px;
        height: 0px;
    }

    .left {
        float: left;
    }

    .right {
        float: right;
    }

    .center {
        text-align: center;
    }

    hr {
        border-top: 1px solid #E5E5E5;
        border-bottom: none;
        color: #fff;
        background-color: #fff;
        height: 1px;
    }

    .rounded {
        background-color: #FEFEFE;
        border: 1px solid #DBDBDB;
        border-radius: 10px;
        -moz-border-radius: 10px;
        -webkit-border-radius: 10px;
        padding: 5px;
        margin-bottom: 10px;
    }

    #export_xls {
        cursor: pointer;
    }

    .table_filter {
        font-family: Verdana, Arial, sans-serif !important;
        font-size: 0.9em !important;
    }

```

```

}

/*Novas Mensagens de Erro*/
.info, .success, .warning, .error {
    border: 1px solid;
    margin: 5px;
    padding: 8px 3px 8px 40px;
    background-repeat: no-repeat;
    background-position: 10px center;
    text-align: left;
    border-radius: 10px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
}

.info {
    color: #00529B;
    background-color: #BDE5F8;
    background-image: url('../_img/info.png');
}

.success {
    color: #4F8A10;
    background-color: #DFF2BF;
    background-image: url('../_img/ok.png');
}

.warning {
    color: #9F6000;
    background-color: #FEEFB3;
    background-image: url('../_img/warning.png');
}

.error {
    color: #D8000C;
    background-color: #FFBABA;
    background-image: url('../_img/error.png');
}

.ajax_loading {
    margin: 5px;
    padding: 8px 3px 8px 40px;
    background-repeat: no-repeat;
    background-position: 10px center;
    background-image: url('../_img/ajax-loader.gif');
}

.form_loading {
    min-width: 25px;
    min-height: 26px;
    text-align: left;
    background-repeat: no-repeat;
    background-position: 10px center;
    background-image: url('../_img/ajax-loader.gif');
}

.subgrid_info {
    padding: 8px 3px 8px 15px;
}

```

```

        border-radius: 10px;
        -moz-border-radius: 10px;
        -webkit-border-radius: 10px;
        background-color: #F3F3F3;
        border-bottom: 1px solid #cccccc
    }

    .edit {
        cursor: pointer;
    }

    input.input_error, select.input_error {
        background-color: #FFF4F4;
        color: #8a1f11;
        border: 1px solid #D95429 !important;
    }

    input.input_warning, select.input_warning {
        background-color: #FEEFB3;
        color: #9F6000;
        border: 2px solid !important;
    }

    /*CSS CHECKBOX STYLE*/
    .radio {
        height: 25px;
        width: 19px;
        clear: left;
        float: left;
        padding: 1px 2px 0 42px;
        background: url("../_img/radio.png");
        background-repeat: no-repeat;
        cursor: default;
    }

    .checkbox {
        height: 24px;
        width: 19px;
        clear: left;
        float: left;
        padding: 2px 3px 0 4px;
        background: url("../_img/checkbox.png") no-repeat;
        cursor: default;
        text-align: left;
        margin-left: 5px;
    }

    .checkbox input, .radio input {
        display: none;
    }

    .checkbox input.show, .radio input.show {
        display: inline;
    }

    .selected {
        background-position: 0 -52px;
    }

```

```

.checkbox_label {
    padding-left: 10px;
    float: left;
    text-align: left;
    width: 280px;
}

.checkbox_Label{
    width: 200px;
}

#right_side {
    border: none;
    background-color: transparent;
    width: 100%;
}

#login_wrapper {
    margin: 70px auto;
    width: 270px;
    padding: 15px;
    text-align: left;
}

/*
 * Função default para criação de Formulários usando AjaxForm /*
 * Recebe o nome do form e o div que será atualizado com o response
 * Lembrar que a resposta do php deve vir no formato:
 * exit(json_encode(array('status'=>'error','msg'=>'resposta')));
 */

function createForm(form, div_status) {
    var options = {
        type : 'POST',
        dataType : 'json',
        beforeSend : function() {
            $(form).find(".input_error").removeClass();
            $(form).find(".chzn-single").css("border-color", "#AAAAAA");
            $(form).find(".chzn-drop").css("border-color", "#AAAAAA");
            $(form).find("fieldset").css("border", "solid 1px #AAAAAA");
            $(form).find("#loading").addClass("form_loading");
            $(div_status).removeClass().html("");
        },
        success : function(response, status) {
            $(form).find("#loading").removeClass();
            if (response == null){
                showError(div_status);
            } else if (response.status == 'unauthorized') {
                location.reload(true);
            } else if (typeof response.errors == 'object'){
                var focus = true;
                var error_msg = '';
                $.each(response.errors, function(i, val) {
                    if (($('#' + i).get(0).tagName) != undefined){
                        var tipo = ($('#' + i).get(0).tagName);
                        if (tipo == "INPUT"){
                            $('#' + i).addClass('input_error');
                        } else if (tipo == "SELECT") {

```



```

function processLoadComplete(response) {
    if (response == null){
        showError($('#status'));
    } else if (response.status == 'unauthorized'){
        location.reload(true);
    } else if (response.records == 0 && response.filtro == undefined) {
        $('#status').addClass("warning").html('Nenhum registro encontrado !').show();
    }
}

function showError(div_status) {
    $(div_status).addClass('error').html("Ocorreu um erro inesperado.<br />Entre em
contato com o administrador do sistema.");
}

function voltar(){
    $('#formulario').html("").hide();
    $('#showtable, #buttons, #status').show();
    $('#status').removeClass().html("");
    $('#table').trigger('reloadGrid');
}

/**
 * Função para habilitar o checkbox Somente Leitura no cadastro de privilégio do
 * usuario
 */
function habilitaSomenteLeitura(form) {
    for ( var i = 0; i < form.elements.length; i++) {
        var x = form.elements[i];
        if (x.name == 'privilegios[]') {
            if (x.checked)
                form.elements[i + 1].disabled = false;
            else {
                form.elements[i + 1].disabled = true;
                form.elements[i + 1].checked = false;
            }
        }
    }
}

/**
 * Função para habilitar botão quando marcar checkbox
 */
function verificaCheckbox(table, botaoDel) {
    var id = table.attr('id');
    var botao = botaoDel;
    if ($('#' + id + ' :checkbox:checked').length > 0) {
        botao.removeClass('disabled').removeAttr('disabled');
    } else {
        botao.addClass('disabled').attr('disabled', 'disabled');
    }
}

function displayColorbox(url, width) {
    $.colorbox({
        href:url,
        innerWidth: width,
        onComplete:function() {

```

```

        $.colorbox.resize();
    });
}

/**
 * Função para formatar um CEP
 *
 * @param campo -
 *             Campo onde foi digitado o valor
 * @param teclapres -
 *             Evento da tecla pressionada
 */
function formatarCep(campo, teclapres) {
    var tecla = teclapres.keyCode ? teclapres.keyCode
        : teclapres.which ? teclapres.which : teclapres.charCode;
    campo.value = retiraNaoNumericos(campo.value);
    vr = campo.value;
    tam = vr.length + 1;
    if (tecla != 9 && tecla != 8) {
        if (tam > 5)
            campo.value = vr.substr(0, 5) + '-' + vr.substr(5, tam);
    }
}

/**
 * Função para retirar os caracteres não numéricos de um campo
 *
 * @param campo -
 *             Campo onde foi digitado o valor
 */
function retiraNaoNumericosCampo(campo) {
    campo.value = retiraNaoNumericos(campo.value);
}

/**
 * Função para retirar os caracteres não numéricos de uma String
 *
 * @param string -
 *             String que deseja retirar os caracteres
 * @return string - String com os caracteres retirados
 */
function retiraNaoNumericos(string) {
    var validChars = "0123456789";

    for (i = 0; i < string.length; i++) {
        if (validChars.indexOf(string.charAt(i)) == -1) {
            string = string.replace(string.charAt(i), "");
        }
    }
    return string;
}
}

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');

```

```

require_once(INCLUDE_PATH ."/DB.php");
require_once(INCLUDE_PATH ."/Auth.php");
require_once(CONTROLLER_PATH ."/ControleAcordo.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $nome_acordo = "";
    $fk_acordos_tipos = "";
    $fk_acordos_mon_tipos = "";
    $fk_acordos_tipo_periodicidade = "";
    $valor_quebra = "";
    $valor_critico = "";
    $descricao_acordo = "";
    $unidade = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleAcordo::getInstance();
        $result = $controle->getInfo();
        $nome_acordo = $result["nome_acordo"];
        $fk_acordos_tipos = $result["fk_acordos_tipos"];
        $unidade = $result["unidade"];
        $fk_acordos_tipo_periodicidade = $result["fk_acordos_tipo_periodicidade"];
        $fk_acordos_mon_tipos = $result["fk_acordos_mon_tipos"];
        $valor_quebra = sprintf("%.2f", $result["valor_quebra"]);
        $valor_critico = sprintf("%.2f", $result["valor_critico"]);
        $descricao_acordo = $result["descricao_acordo"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE ACORDOS" : "ALTERAÇÃO DE ACORDOS";
?>

<form name="cadastrar_acordo" id="cadastrar_acordo" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Nome*</label><br />
            <input name="nome_acordo" id="nome_acordo" type="text"
value="<?php echo $nome_acordo; ?>" style="width: 220px" />
        </p>

        <p>
            <label>Tipo de Acordo*</label><br />
            <select <?php if ($acao == "alterar") echo "disabled='disabled'";
?> data-placeholder="Selecione..." id="fk_acordos_tipos" name="fk_acordos_tipos"
style="width: 170px">

                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("SELECT id_tipo_acordo,
nome_tipo_acordo

```

from

acordos_tipos

```
ORDER BY id_tipo_acordo", array());
print "<option value=''></option>";
if (!empty($result)){
    foreach ($result as $res) {
        $id_tipo_acordo = $res["id_tipo_acordo"];
        $nome_tipo_acordo = $res["nome_tipo_acordo"];
        $checked = "";
        if ($fk_acordos_tipos == $id_tipo_acordo){
$checked = "selected='selected'"; }
        echo "<option $checked
value='$id_tipo_acordo'$nome_tipo_acordo</option>";
    }
}
?>
</select>
</p>
```

```
<p>
<label>Tipo de Monitoramento*</label><br />
<select <?php if ($acao == "alterar") echo
"disabled='disabled'"; ?> data-placeholder="Selecione o tipo de acordo..."
id="fk_acordos_mon_tipos" name="fk_acordos_mon_tipos" style="width: 220px">
<?php
$DB = DB::getInstance();
$result = $DB->execPreparedSQL("SELECT id_mon_tipo,
```

descricao_mon_tipo

```
FROM acordos_mon_tipos amt

ORDER BY descricao_mon_tipo", array());
print "<option value=''></option>";
if (!empty($result)){
    foreach ($result as $res) {
        $id_tipo = $res["id_mon_tipo"];
        $desc_tipo = "{$res["descricao_mon_tipo"]}";
        $checked = "";
        if ($fk_acordos_mon_tipos == $id_tipo){
$checked = "selected='selected'"; }
        echo "<option $checked
value='$id_tipo'$desc_tipo</option>";
    }
}
?>
</select>
</p>
```

```
<p>
<label>Período*</label><br />
<select data-placeholder="Selecione..."
id="fk_acordos_tipo_periodicidade" name="fk_acordos_tipo_periodicidade" style="width:
180px">
<?php
$DB = DB::getInstance();
$result = $DB->execPreparedSQL("SELECT id_periodicidade,
```

nome_tipo_periodo

acordos_tipo_periodicidade

```

ORDER BY id_periodicidade", array());
print "<option value=''></option>";
if (!empty($result)){
    foreach ($result as $res) {
        $id_periodicidade = $res["id_periodicidade"];
        $nome_tipo_periodo = $res["nome_tipo_periodo"];
        $checked = "";
        if ($fk_acordos_tipo_periodicidade ==
$id_periodicidade){ $checked = "selected='selected'"; }
        echo "<option $checked
value='$id_periodicidade'$nome_tipo_periodo</option>";
    }
}
?>
</select>
</p>

<div class="clear"></div>

<p style="width: 430px;">
    <label>Descrição</label><br />
    <input name="descricao_acordo" id="descricao_acordo"
type="text" value="<?php echo $descricao_acordo; ?>" style="width: 410px" />
</p>

<p style="width: 200px;">
    <label>Valor de Quebra</label><br />
    <input name="valor_quebra" id="valor_quebra" type="text"
value="<?php echo $valor_quebra; ?>" style="width: 120px; text-align:right;" />
    <label class='unidade'><?php echo $unidade;?></label>
</p>

<p>
    <label>Valor Crítico</label><br />
    <input name="valor_critico" id="valor_critico" type="text"
value="<?php echo $valor_critico; ?>" style="width: 120px; text-align:right;" />
    <label class='unidade'><?php echo $unidade;?></label>
</p>

<div class="clear"></div>

</div>

<div class="clear" style="margin-bottom: 10px;"></div>
<div id='botao' style="float: left;">
    <?php
    $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
    <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
    <input name="voltar" id="voltar" type="button" value="Voltar" />
    <div id="loading" style="float:right;"></div>
</div>
<div class="clear"></div>
<input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
<div id="status_cadastrar_acordo"></div>

```

```

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_acordo', '#status_cadastrar_acordo');
    $("#voltar").click(function(){ voltar(); });
    $("#fk_acordos_tipo_periodicidade, #fk_acordos_mon_tipos,
#fk_acordos_tipos").chosen({search_contains: true});

    //$("#valor_quebra, #valor_critico").priceFormat({
    //    prefix: '',
    //    centsSeparator: ',',
    //    thousandsSeparator: ''
    //});

    //$("#valor_quebra, #valor_critico").keyup(function(){
    //    if (parseFloat($(this).val().replace(/[.]/gi, '')) > 10000){
    //        $(this).val('100,00');
    //    }
    //});

    $("#fk_acordos_tipos").change(function() {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getTiposMon', 'select': $('#fk_acordos_tipos').val() },
            success: function(response){
                $('#fk_acordos_mon_tipos').html("").html(response.options);
                $('#fk_acordos_mon_tipos').trigger("liszt:updated");
                $('#.unidade').html("").html(response.unidade);
            },
            error: function(){
                $('#fk_acordos_mon_tipos').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
                $('#fk_acordos_mon_tipos').next().children().css("border-color",
"#9F6000");
                $('#status_cadastrar_acordo').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />Tipos de monitoramentos não
disponíveis.</div>');
            }
        });
    });
});
</script>

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_ACORDOS_GERENCIAR)) {
    unauthorized();
}

global $somente_leitura;

?>

```

```

<fieldset class="rounded">
  <legend>Acordos</legend>

  <div id="showtable">
    <table id="table"></table>
    <div id="pager"></div>
  </div>

  <div id='buttons'>
    <?php if (!$somete_leitura) { ?>
      <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Acordo' />
      <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
    <?php } ?>
  </div>

  <div id='status'></div>
  <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
  var grid = $('#table');

  var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

  grid.jqGrid({
    url: '?acao=getTableData',
    datatype: 'json',
    autowidth: true,
    height: 270,
    colNames: ['Nome', 'Período', 'Tipo de Acordo', 'Monitoramento', 'Valor de Quebra',
'Valor Crítico' <?php echo (!$somete_leitura ? ",'" : "" );?>],
    colModel: [
      {name:'nome_acordo', index:'nome_acordo', resizable:false,
width:'20%',align:'center'},
      {name:'nome_tipo_periodo', index:'nome_tipo_periodo', resizable:false,
width:'10%',align:'center'},
      {name:'nome_tipo_acordo', index:'nome_tipo_acordo', resizable:false,
width:'20%',align:'center'},
      {name:'descricao_mon_tipo', index:'descricao_mon_tipo', resizable:false,
width:'15%',align:'center'},
      {name:'valor_quebra', index:'valor_quebra', resizable:false,
width:'13%',align:'center'},
      {name:'valor_critico', index:'valor_critico', resizable:false,
width:'13%',align:'center'}
      <?php echo (!$somete_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum:10,
    rowList:[10,30,100],
    scrollOffset: 17,
    gridview: true,
    pager: '#pager',
    sortname: 'nome_acordo',
    sortorder: "asc",

```

```

toolbar: [false, 'top'],
viewrecords: true,
shrinkToFit: true,
hoverrows: false,
beforeSelectRow: function(rowId, e){
    if(e.target.type == "checkbox"){
        verificaCheckbox($('#table'), $('#delete'));
        return true;
    } else if(e.target.className == "edit") {
        $('#showtable, #buttons, #status').hide();
    }
}

$('#formulario').load("formGerenciarAcordo.php?acao=alterar&id="+rowId).show();
return false;
} else {
    grid.jqGrid('toggleSubGridRow', rowId);
}
},
subGrid: true,
subGridOptions: {
    plusicon : "ui-icon-triangle-1-e",
    minusicon : "ui-icon-triangle-1-s",
    openicon : "ui-icon-arrowreturn-1-e",
    expandOnLoad : false
},
subGridRowExpanded: function(subgrid_id, row_id) {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'showInfo', id: row_id },
        beforeSend: function(){
            $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
        },
        success: function(response){
            if (response.status == 'success'){
                $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>" + response.info + "</div>");
            } else if (response.status == 'error'){
                showError($("div[id='"+subgrid_id+"'"));
            } else if (response.status == 'unauthorized'){
                location.reload(true);
            }
        },
        error: function(){
            showError($("div[id='"+subgrid_id+"'"));
        }
    });
},
onSelectAll: function(){
    verificaCheckbox($('#table'), $('#delete'));
},
loadComplete: function(response){
    processLoadComplete(response);
    verificaCheckbox($('#table'), $('#delete'));
},
<?php if (!$somete_leitura) { echo 'multiselect: true, ' ;} ?>
search : true,
search : {
    caption: "Procurar..."
},

```



```

        caption: campoProcurar,
        loadError : function(xhr,st,err) {
            showError($('#status'));
        }
    });

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formGerenciarAcordo.php?acao=cadastrar").show();
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

```

```

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleAcordo.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleAcordo::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleAcordo::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleAcordo::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleAcordo::getInstance();
        exit($controller->showInfo());

    case 'getTiposMon':
        $controller = ControleAcordo::getInstance();
        exit($controller->getTiposMon());
}

```

```

default:
    require_once(INCLUDE_PATH . 'TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
    ));

    $template->set_js_files(array(
        "jquery.price_format.1.7.min.js"
    ));
    $template->set_content_file('acordos/gerenciar/gerenciar.php');
    $template->set_authenticated_only(true);
    $template->set_module(ID_MODULO_SISTEMA);
    $template->set_sub_module(ID_SUBMODULO_ACORDOS);
    $template->set_sub_module_item(ID_SUBMODULO_ACORDOS_GERENCIAR);
    $template->show();
}

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php
$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleUsuario.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $login_usuario = "";
    $senha_usuario = "";
    $nome_usuario = "";
    $email_usuario = "";
    $telefone_usuario = "";
    $fk_organizacao_instituicoes = "";
    $fk_organizacao_unidades = "";
    $privilegios = array();

    $selected = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleUsuario::getInstance();
        $result = $controle->getInfo();
        $privilegios = $controle->getPrivilegios();
        $nome_usuario = $result["nome_usuario"];
        $login_usuario = $result["login_usuario"];
        $senha_usuario = "xxxxxx";
        $email_usuario = $result["email_usuario"];

```

```

        $telefone_usuario = isset($result["telefone_usuario"]) ?
$result["telefone_usuario"] : "";
        $fk_organizacao_instituicoes = $result["fk_organizacao_instituicoes"];
        $fk_organizacao_unidades = $result["fk_organizacao_unidades"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE USUÁRIOS" : "ALTERAÇÃO DE USUÁRIOS";
?>
<form name="cadastrar_usuario" id="cadastrar_usuario" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Login*</label><br />
            <input name="Login_usuario" id="Login_usuario" type="text"
value="<?php echo $login_usuario; ?>" style="width: 120px" title="Insira o usuário(Login)
que será utilizado no Login do Administrador." /></span>
        </p>
        <p>
            <label>Senha*</label><br />
            <input autocomplete='off' name="senha_usuario"
id="senha_usuario" type="password" value="<?php echo $senha_usuario; ?>" style="width:
100px" /></span>
        </p>
        <p>
            <label>Nome*</label><br />
            <input name="nome_usuario" id="nome_usuario" type="text"
value="<?php echo $nome_usuario; ?>" style="width: 230px" /></span>
        </p>
        <p>
            <label>Email*</label><br />
            <input name="email_usuario" id="email_usuario" type="text"
value="<?php echo $email_usuario; ?>" style="width: 180px" /></span>
        </p>
        <p>
            <label>Telefone</label><br />
            <input name="telefone_usuario" id="telefone_usuario"
type="text" value="<?php echo $telefone_usuario; ?>" style="width: 120px" /></span>
        </p>
        <div class="clear"></div>
        <p>
            <label>Instituição</label><br />
            <select data-placeholder="Selecione..."
id="fk_organizacao_instituicoes" name="fk_organizacao_instituicoes" style="width: 160px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("select
id_instituicao,sigla_instituicao from organizacao_instituicoes order by sigla_instituicao",
array());
                    echo "<option value='0'>Todas</option>";
                    if (!empty($result)){

```

```

        foreach ($result as $res) {
            $id_inst = $res["id_instituicao"];
            $sigla = $res["sigla_instituicao"];
            $checked = "";
            if ($fk_organizacao_instituicoes ==
$id_inst){ $checked = "selected='selected'"; $selected = $id_inst; }
            echo "<option $checked
value='$id_inst'$sigla</option>";
        }
    }
    ?>
</select>
</p>

<p>
<label>Unidade</label><br />
<select data-placeholder="Selecione..."
id="fk_organizacao_unidades" name="fk_organizacao_unidades" style="width: 160px">
    <?php
        $DB = DB::getInstance();
        $result = $DB->execPreparedSQL("select
id_unidade,sigla_unidade,nome_unidade from organizacao_unidades WHERE
fk_organizacao_instituicoes = ? order by sigla_unidade", array('i', $DB-
>escape($selected)));
        echo "<option value='0'>Todas</option>";
        if (!empty($result)){
            foreach ($result as $res) {
                $id_uni = $res["id_unidade"];
                $sigla = $res["sigla_unidade"];
                $checked = "";
                $desc_unidade = "{$res["sigla_unidade"]}
[{$res["nome_unidade"]}];
                if ($fk_organizacao_unidades == $id_uni){
                    $checked = "selected='selected'"; }
                    echo "<option $checked
value='$id_uni'$desc_unidade</option>";
            }
        }
    }
    ?>
</select>
</p>

<div class="clear" style="margin-bottom: 10px;"></div>

<fieldset id="privilegio" class="rounded">
<legend name="privilegio">Privilégios</legend>

    <div id="accordion">
    <?php

        $primeiro = true;
        $temp_nome_modulo = "";
        $result = $DB->execPreparedSQL("SELECT
id_modulo,nome_modulo,id_submodulo,nome_submodulo FROM sistema_modulos
                                JOIN
sistema_submodulos ON id_modulo = fk_sistema_modulos ORDER BY id_modulo,id_submodulo",
array());
        foreach ($result as $inf){
            $id_modulo = $inf["id_modulo"];

```

```

        $id_submodulo = $inf["id_submodulo"];
        $nome_modulo = $inf["nome_modulo"];
        $nome_submodulo = $inf["nome_submodulo"];

        if($temp_nome_modulo != $nome_modulo){
            if (!$primeiro){
                echo "</div>";
            }
            echo "<h2>Módulo: $nome_modulo</h2>";
            echo "<div>";
            echo "<input class='select_all'
type='checkbox'/> Selecionar Todos";
            echo "</br>";
        }

        $checked =
array_key_exists($id_modulo."_".$id_submodulo, $privilegios) ? "checked=checked" : "";
        $somleitura_checked = ($checked &&
$privilegios[$id_modulo."_".$id_submodulo]) ? "checked='checked'" : "";
        $somleitura_disabled = ($checked) ? "" :
"disabled='disabled'";
        echo "</br><input style='float:left;
name='privilegios[]' $checked id='{ $id_modulo }_{ $id_submodulo }'
onclick='habilitaSomenteLeitura(this.form)' type='checkbox'
value='{ $id_modulo }_{ $id_submodulo }' /> <span style='margin-left:5px; min-width:130px;
float:left;'> $nome_submodulo</span>";
        echo "<input name='somenteLeitura[]'
id='somenteLeitura' $somleitura_checked $somleitura_disabled type='checkbox'
value='{ $id_modulo }_{ $id_submodulo }' /> Somente Leitura";

        $primeiro = false;
        $temp_nome_modulo = $nome_modulo;
    }
    ?>
</div>
</fieldset>

<div class="clear" style="margin-bottom: 10px;"></div>
<div id='botao' style="float: left;">
    <?php
    $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
    <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
    <input name="voltar" id="voltar" type="button" value="Voltar" />
    <div id="loading" style="float:right;"></div>
</div>
<div class="clear"></div>
<input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
<div id="status_cadastrar_usuario"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {

```

```

$("#accordion").accordion({
    collapsible: false,
    heightStyle: "content"
});

createForm('#cadastrar_usuario', '#status_cadastrar_usuario');
$("#voltar").click(function(){ voltar(); });
$("#fk_organizacao_instituicoes, #fk_organizacao_unidades").chosen({search_contains:
true});
$("#telefone_usuario").mask('(00) 0000-0000',
    {onKeyPress:
        function(phone, e, currentField, options){
            var new_phone = phone.match(/^(\\(11\\) 9(5[0-9]|6[0-9]|7[01234569]|8[0-9]|9[0-9])][0-9]{1})/g);
            new_phone ? $(currentField).mask('(00) 00000-0000', options) :
$(currentField).mask('(00) 0000-0000', options)
        }
    });

$("#accordion").accordion({heightStyle: "content"});
$("#fk_organizacao_instituicoes").change(function() {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'getUnidades', 'select':
$("#fk_organizacao_instituicoes").val() },
        success:function(response){
            $('#fk_organizacao_unidades').html("").html(response.options);
            $("#fk_organizacao_unidades").trigger("liszt:updated");
        },
        error:function(){
            $('#fk_organizacao_unidades').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
            $('#fk_organizacao_unidades').next().children().css("border-color",
"#9F6000");
            $('#status_cadastrar_usuario').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />Opções de unidades não
disponíveis.</div>');
        }
    });
});

$('.select_all').change(function(){
    checked = $(this).attr('checked');
    $(this).parent().find('input[name="privilegios[]"]').each(function(){
        sl = ($(this).next().next());
        if(checked=='checked'){
            $(this).attr("checked", true);
            sl.attr('disabled', false);
        } else {
            $(this).attr("checked", false);
            sl.attr('disabled', true);
        }
    });
});
});
</script>

```

<?php

```

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleUsuario.php");
require_once(CONTROLLER_PATH . "ControleUnidade.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleUsuario::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleUsuario::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleUsuario::getInstance();
        exit($controller->getTableData());

    case 'getUnidades':
        $controller = ControleUnidade::getInstance();
        exit($controller->getUnidades());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
            "jquery.mask.min.js"
        ));
        $template->set_content_file('admin/usuarios/usuarios.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_ADMIN);
        $template->set_sub_module_item(ID_SUBMODULO_ADMIN_USUARIOS);
        $template->show();
    }
}

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_ADMIN_USUARIOS)) {
    unauthorized();
}

global $somenteleitura;

?>
<fieldset class="rounded">
    <legend>Usuários</legend>

    <div id="showtable">
        <table id="table"></table>

```

```

        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somete_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Usuário' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
        <?php } ?>
    </div>

    <div id='status'></div>
    <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames: ['Nome', 'Login', 'Email', 'Instituição', 'Unidade', 'Telefone' <?php echo
(!$somete_leitura ? ",'" : "'");?>],
        colModel:[
            {name:'nome_usuario', index:'nome_usuario', resizable:false,
width:'20%',align:'center'},
            {name:'login_usuario', index:'login_usuario', resizable:false,
width:'10%',align:'center'},
            {name:'email_usuario', index:'email_usuario', resizable:false,
width:'25%',align:'center'},
            {name:'sigla_instituicao', index:'sigla_instituicao', resizable:false,
width:'15%',align:'center'},
            {name:'sigla_unidade', index:'sigla_unidade', resizable:false,
width:'15%',align:'center'},
            {name:'telefone_usuario', index:'telefone_usuario', resizable:false,
width:'15%',align:'center'}
            <?php echo (!$somete_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "'");?>
        ],
        rowNum:10,
        rowList:[10,30,100],
        scrollOffset: 17,
        gridview: true,
        pager: '#pager',
        sortname: 'nome_usuario',
        sortOrder: "asc",
        toolbar: [false,'top'],
        viewrecords: true,
        shrinkToFit: true,
        hoverrows: false,
        beforeSelectRow: function(rowId, e){

```



```

        if(e.target.type == "checkbox"){
            verificaCheckbox($('#table'),$('#delete'));
            return true;
        } else if(e.target.className == "edit") {
            $('#showtable, #buttons, #status').hide();
            $('#formulario').load("formUsuario.php?acao=alterar&id="+rowId).show();
            return false;
        } else {
            grid.jqGrid('toggleSubGridRow', rowId);
        }
    },
    onSelectAll: function(){
        verificaCheckbox($('#table'),$('#delete'));
    },
    loadComplete: function(response){
        processLoadComplete(response);
        verificaCheckbox($('#table'),$('#delete'));
    },
    <?php if (!$somete_leitura) { echo 'multiselect: true, ' ;} ?>
    search : true,
    search : {
        caption: "Procurar..."
    },
    caption: campoProcurar,
    loadError : function(xhr,st,err) {
        showError($('#status'));
    }
});

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formUsuario.php?acao=cadastrar").show();
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS)) {
    unauthorized();
}

global $somete_leitura;

```

```

?>
<fieldset class="rounded">
  <legend>Circuitos</legend>

  <div id="showtable">
    <table id="table"></table>
    <div id="pager"></div>
  </div>

  <div id='buttons'>
    <?php if (!$somete_leitura) { ?>
      <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Circuito' />
      <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
    <?php } ?>
  </div>

  <div id='status'></div>
  <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
  var grid = $('#table');

  var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

  grid.jqGrid({
    url: '?acao=getTableData',
    datatype: 'json',
    autowidth: true,
    height: 270,
    colNames:['Designação', 'Dispositivo', 'Operadora', 'Porta/IP do Neighbour' <?php
echo (!$somete_leitura ? ",'" : "" );?>],
    colModel:[
      {name:'designacao_circuito', index:'designacao_circuito', resizable:false,
width:'25%',align:'center'},
      {name:'nome_dispositivo', index:'nome_dispositivo', resizable:false,
width:'22%',align:'center'},
      {name:'nome_operadora', index:'nome_operadora', resizable:false,
width:'23%',align:'center'},
      {name:'port_neighbour', index:'port_neighbour', resizable:false,
width:'28%',align:'center'}
      <?php echo (!$somete_leitura ?",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum:10,
    rowList:[10,30,100],
    scrollOffset: 17,
    gridview: true,
    pager: '#pager',
    sortname: 'designacao_circuito',
    sortorder: "asc",
    toolbar: [false,'top'],
    viewrecords: true,

```

```

shrinkToFit: true,
hoverrows: false,
beforeSelectRow: function(rowId, e){
    if(e.target.type == "checkbox"){
        verificaCheckbox($('#table'),$('#delete'));
        return true;
    } else if(e.target.className == "edit") {
        $('#showtable, #buttons, #status').hide();
        $('#formulario').load("formCircuito.php?acao=alterar&id="+rowId).show();
        return false;
    } else {
        grid.jqGrid('toggleSubGridRow', rowId);
    }
},
subGrid: true,
subGridOptions: {
    plusicon : "ui-icon-triangle-1-e",
    minusicon : "ui-icon-triangle-1-s",
    openicon : "ui-icon-arrowreturn-1-e",
    expandOnLoad : false
},
subGridRowExpanded: function(subgrid_id, row_id) {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'showInfo', id: row_id },
        beforeSend:function(){
            $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
        },
        success:function(response){
            if (response.status == 'success'){
                $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>"+response.info+"</div>");
            } else if (response.status == 'error'){
                showError($("div[id='"+subgrid_id+"'"));
            } else if (response.status == 'unauthorized'){
                location.reload(true);
            }
        },
        error:function(){
            showError($("div[id='"+subgrid_id+"'"));
        }
    });
},
onSelectAll: function(){
    verificaCheckbox($('#table'),$('#delete'));
},
loadComplete: function(response){
    processLoadComplete(response);
    verificaCheckbox($('#table'),$('#delete'));
},
<?php if (!$somete_leitura) { echo 'multiselect: true,' ;} ?>
search : true,
search : {
    caption: "Procurar..."
},
caption:campoProcurar,
loadError : function(xhr,st,err) {
    showError($('#status'));
}

```

```

    }
  });

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid', '#pager',
  {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
  grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
  grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
  $('#showtable, #buttons, #status').hide();
  $('#formulario').load("formCircuito.php?acao=cadastrar").show();
});

$("#delete").click(function(){
  deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleCircuito.php");

global $somente_leitura;
if (!$somente_leitura) {

  $id = "";
  $fk_dispositivos = "";
  $fk_operadoras = "";
  $designacao_circuito = "";
  $ip_disp_alvo = "";
  $port_neighbour = "";
  $tipo_circuito = "";

  $selected = "";

  if ($acao == "alterar") {
    if (isset($_GET['id'])){
      $id = anti_injection($_GET['id']);
    } else {
      exit('<p class="error">ID não especificado.</p>');
    }
  }
}

```

```

        $DB = DB::getInstance();
        $controle = ControleCircuito::getInstance();
        $result = $controle->getInfo();
        $fk_dispositivos = $result["fk_dispositivos"];
        $fk_operadoras = $result["fk_operadoras"];
        $designacao_circuito = $result["designacao_circuito"];
        $ip_disp_alvo = $result["ip_disp_alvo"];
        $tipo_circuito = $result["tipo_circuito"];
        $port_neighbour = $result["port_neighbour"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE CIRCUITOS" : "ALTERAÇÃO DE
CIRCUITOS";
?>

<form name="cadastrar_circuito" id="cadastrar_circuito" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Designação do Circuito*</label><br />
            <input name="designacao_circuito" id="designacao_circuito"
type="text" value="<?php echo $designacao_circuito; ?>" style="width: 335px" />
        </p>
        <p>
            <label>Tipo*</label><br />
            <select data-placeholder="Selecione..." id="tipo_circuito"
name="tipo_circuito" style="width: 230px">
                <option value=''></option>
                <option value='1' <?php if($tipo_circuito==1)echo
"selected='selected'";?>>Por Porta/Interface</option>
                <option value='2' <?php if($tipo_circuito==2)echo
"selected='selected'";?>>Por IP do Neighbour</option>
            </select>
        </p>
        <p>
            <label>Operadora*</label><br />
            <select data-placeholder="Selecione..." id="fk_operadoras"
name="fk_operadoras" style="width: 230px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("SELECT id_operadora,
nome_operadora
FROM operadoras o
ORDER BY nome_operadora", array());
                    print "<option value=''></option>";
                    if (!empty($result)){
                        foreach ($result as $res) {
                            $id_operadora = $res["id_operadora"];
                            $nome_operadora = "{$res["nome_operadora"]}";
                            $checked = "";
                            if ($fk_operadoras == $id_operadora){
                                $checked = "selected='selected'";}
                </select>
            </p>
    </div>
</form>

```

```

        echo "<option $checked
value='$id_operadora'$nome_operadora</option>";
    }
}
?>
</select>
</p>

<div class="clear"></div>

<p>
<label>Dispositivo Alvo*</label><br />
<select data-placeholder="Selecione..." id="fk_dispositivos"
name="fk_dispositivos" style="width: 180px">
<?php
    $DB = DB::getInstance();
    $result = $DB->execPreparedSQL("SELECT id_dispositivo,
nome_dispositivo
FROM dispositivos d
ORDER BY nome_dispositivo", array());
    print "<option value=''></option>";
    if (!empty($result)){
        foreach ($result as $res) {
            $id_dispositivo = $res["id_dispositivo"];
            $nome_dispositivo = "{$res["nome_dispositivo"]}";
            $checked = "";
            if ($fk_dispositivos == $id_dispositivo){
                $checked = "selected='selected'"; $selected = $id_dispositivo; }
            echo "<option $checked
value='$id_dispositivo'$nome_dispositivo</option>";
        }
    }
?>
</select>
</p>

<div id='div_ip_alvo'>
<p>
<label>IP Dispositivo Alvo*</label><br />
<select data-placeholder="Selecione..." id="ip_disp_alvo"
name="ip_disp_alvo" style="width: 280px">
<?php
    $DB = DB::getInstance();
    $result = $DB->execPreparedSQL("select ipv4_dispositivo,
ipv6_dispositivo from dispositivos WHERE id_dispositivo = ? order by ipv4_dispositivo",
array('i', $DB->escape($selected)));
    print "<option value=''></option>";
    if (!empty($result)){
        foreach ($result as $res) {
            $ips4 = explode("|",$res["ipv4_dispositivo"]);
            $ips6 = explode("|",$res["ipv6_dispositivo"]);
            $ips = array_merge($ips4, $ips6);
            foreach ($ips as $ip){
                $checked = "";
                if ($ip_disp_alvo == $ip){ $checked =
"selected='selected'"; }

```

```

        echo "<option $checked
value='$ip'>$ip</option>";
    }
}
?>
</select>
</p>
</div>

<div class="clear"></div>

<div id='fieldset_ep' style='margin-top: 10px; display:none;'>
    <div id='snmpi' style="display: none;">
        <fieldset class='rounded' id='portas_field'>
            <legend>Portas:</legend>
            <input name="portas" id="portas" type="button"
value="Buscar Portas" />

            <div id='portas_indisponiveis'></div>
            <div class="clear"></div>
            <fieldset id='field_snmpi' class='rounded'

                <div id='snmpi_portas' class='rounded'>

                    </div>
                </fieldset>
            </fieldset>
            <div class="clear"></div>
        </div>

        <div id='snmpo'>
            <p>
                <label>IP do Neighbour</label><br />
                <input name="ip_neighbour" id="ip_neighbour"
type="text" value="<?php echo $port_neighbour; ?>" style="width: 300px" />
            </p>

            <div class="clear"></div>
        </div>

        <div class="clear"></div>
    </div>

</div>

<div class="clear" style="margin-bottom: 10px;"></div>
<div id='botao' style="float: left;">
    <?php
$button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
    <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
    <input name="voltar" id="voltar" type="button" value="Voltar" />
    <div id="loading" style="float:right;"></div>
</div>
<div class="clear"></div>
<input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
<div id="status_cadastrar_circuito"></div>

</form>

```

```

<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_circuito', '#status_cadastrar_circuito');
    $("#voltar").click(function(){ voltar(); });
    $("#fk_dispositivos, #tipo_circuito, #ip_disp_alvo,
#fk_operadoras").chosen({allow_single_deselect: false, search_contains: true});

    $("#tipo_circuito").change(function() {
        $('#status_cadastrar_circuito').removeClass().html("");
        if ($(this).val() == 1) {
            $('#snmpi').show();
            $('#snmpo').hide();
            $('#div_ip_alvo').show();
            $('#fieldset_ep').show();
            $('#input[name="ports[]"]').attr('disabled', false);
            $('#ip_neighbour').attr('disabled', true);
        } else if ($(this).val() == 2) {
            $('#snmpi').hide();
            $('#snmpo').show();
            $('#div_ip_alvo').hide();
            $('#fieldset_ep').show();
            $('#input[name="ports[]"]').attr('disabled', true);
            $('#ip_neighbour').attr('disabled', false);
        }
    });

    $("#fk_dispositivos").change(function() {
        $('#field_snmpi').hide();
        $('#snmpi_portas').removeClass().html("");
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getIps', 'select': $('#fk_dispositivos').val() },
            success: function(response){
                $('#ip_disp_alvo').html("").html(response.options);
                $("#ip_disp_alvo").trigger("liszt:updated");
            },
            error: function(){
                $('#ip_disp_alvo').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
                $('#ip_disp_alvo').next().children().css("border-color", "#9F6000");
                $('#status_cadastrar_circuito').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />IPs não disponíveis.</div>');
            }
        });
    });

    $("#ip_disp_alvo").change(function() {
        $('#field_snmpi').hide();
        $('#snmpi_portas').removeClass().html("");
    });

    $('#portas').bind('click', function() {
        $.ajax({
            type: 'POST',

```



```

        dataType: 'json',
        data: { acao: 'getPortas', 'dispositivo': $('#fk_dispositivos').val(),
'ip': $('#ip_disp_alvo').val(), 'id': '<?=$id?>', 'select_all': 0, 'tipo': 'circuitos' },
        beforeSend: function(){
            $('#field_snmpi').show();
            $('#snmpi_portas').removeClass().html('<div
class="ajax_loading">Verificando portas...</div>');
        },
        success: function(response){
            if (response == null){
                showError($('#snmpi_portas'));
            } else {
                if (response.ports_nd != undefined && response.ports_nd !=
""){
                    $('#portas_indisponiveis').addClass("warning").html("As
seguintes portas não estão mais disponíveis para monitoramento:<br/>" + response.ports_nd);
                }
                $('#field_snmpi').show();

                $('#snmpi_portas').removeClass().addClass(response.status).html(response.msg);
                $(".checkbox").dgStyle();
                $("#check_all_div").click(function() {
                    var checked = $(this).children().is(':checked');

                    $(this).parent().parent().find('.checkbox').each(function() {
                        if (checked){
                            $(this).dgCheck($(this));

                            $(this).find('input[type="checkbox"]').attr('checked', true);
                        } else {
                            $(this).dgUncheck($(this));

                            $(this).find('input[type="checkbox"]').attr('checked', false);
                        }
                    });
                });
            }
        },
        error: function(){
            showError($('#snmpi_portas'));
        }
    });
});

$("#tipo_circuito").trigger("change");
<?php if ($acao == 'alterar' && $tipo_circuito == 1) echo
'$("#portas").trigger("click");'; ?>

});
</script>

<?php
require_once(dirname(__FILE__) . '/../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleCircuito.php");
require_once(CONTROLLER_PATH . "ControleDispositivo.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';

```

```

switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleCircuito::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleCircuito::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleCircuito::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleCircuito::getInstance();
        exit($controller->showInfo());

    case 'getPortas':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getPortas());

    case 'getIps':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getIps());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
            "jquery.custom_radio_checkbox.js"
        ));
        $template->set_content_file('gerenciaveis/circuitos/circuitos.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_GERENCIAVEIS);
        $template->set_sub_module_item(ID_SUBMODULO_GERENCIAVEIS_CIRCUITOS);
        $template->show();
    }

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS)) {
    unauthorized();
}

global $somenteleitura;

?>
<fieldset class="rounded">
    <legend>Dispositivos</legend>

    <div id="showtable">

```

```

        <table id="table"></table>
        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somento_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Dispositivo' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
        <?php } ?>
    </div>

    <div id='status'></div>
    <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames: ['Nome', 'IPv4', 'IPv6', 'Unidade' <?php echo (!$somento_leitura ? ",'" :
"" );?>],
        colModel: [
            {name:'nome_dispositivo', index:'nome_dispositivo', resizable:false,
width:'20%',align:'center'},
            {name:'ipv4_dispositivo', index:'ipv4_dispositivo', resizable:false,
width:'15%',align:'center'},
            {name:'ipv6_dispositivo', index:'ipv6_dispositivo', resizable:false,
width:'35%',align:'center'},
            {name:'concat_inst_unidade_dispositivo',
index:'concat_inst_unidade_dispositivo', resizable:false, width:'15%',align:'center'}
            <?php echo (!$somento_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
        ],
        rowNum:10,
        rowList:[10,30,100],
        scrollOffset: 17,
        gridView: true,
        pager: '#pager',
        sortname: 'nome_dispositivo',
        sortOrder: "asc",
        toolbar: [false,'top'],
        viewrecords: true,
        shrinkToFit: true,
        hoverrows: false,
        beforeSelectRow: function(rowId, e){
            if(e.target.type == "checkbox"){
                verificaCheckbox($('#table'),$('#delete'));
                return true;
            }
        }
    });
}

```

```

    } else if(e.target.className == "edit") {
        $('#showtable, #buttons, #status').hide();
        $('#formulario').load("formDispositivo.php?acao=alterar&id="+rowId).show();
        //displayColorbox("formDispositivo.php?acao=alterar&id="+rowId,
"515px");
        return false;
    } else {
        grid.jqGrid('toggleSubGridRow', rowId);
    }
},
subGrid: true,
subGridOptions: {
    plusicon : "ui-icon-triangle-1-e",
    minusicon : "ui-icon-triangle-1-s",
    openicon : "ui-icon-arrowreturn-1-e",
    expandOnLoad : false
},
subGridRowExpanded: function(subgrid_id, row_id) {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'showInfo', id: row_id },
        beforeSend: function(){
            $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
        },
        success: function(response){
            if (response.status == 'success'){
                $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>"+response.info+"</div>");
            } else if (response.status == 'error'){
                showError($("div[id='"+subgrid_id+"'"));
            } else if (response.status == 'unauthorized'){
                location.reload(true);
            }
        },
        error: function(){
            showError($("div[id='"+subgrid_id+"'"));
        }
    });
},
onSelectAll: function(){
    verificaCheckbox($('#table'), $('#delete'));
},
loadComplete: function(response){
    processLoadComplete(response);
    verificaCheckbox($('#table'), $('#delete'));
},
<?php if (!$somete_leitura) { echo 'multiselect: true, ' ; } ?>
search : true,
search : {
    caption: "Procurar..."
},
caption: campoProcurar,
loadError : function(xhr, st, err) {
    showError($('#status'));
}
});

$('.ui-jqgrid-titlebar-close').remove();

```

```

grid.jqGrid('navGrid','#pager',
            {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formDispositivo.php?acao=cadastrar").show();
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleDispositivo.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $nome_dispositivo = "";
    $local_dispositivo = "";
    $ipsv4 = array();
    $ipsv6 = array();
    $descricao_dispositivo = "";
    $serie_dispositivo = "";
    $fk_dispositivos_marcas = "";
    $fk_dispositivos_modelos = "";
    $fk_organizacao_unidades = "";
    $comunidade_snmp_dispositivo = "";

    $selected = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }
    }
}

```

```

        $DB = DB::getInstance();
        $controle = ControleDispositivo::getInstance();
        $result = $controle->getInfo();
        $nome_dispositivo = $result["nome_dispositivo"];
        $ipsv4 = isset($result["ipv4_dispositivo"]) ? explode("|",
$result["ipv4_dispositivo"]) : array();
        $ipsv6 = isset($result["ipv6_dispositivo"]) ? explode("|",
$result["ipv6_dispositivo"]) : array();
        $local_dispositivo = $result["local_dispositivo"];
        $descricao_dispositivo = $result["descricao_dispositivo"];
        $nserie_dispositivo = $result["nserie_dispositivo"];
        $fk_dispositivos_marcas = $result["fk_dispositivos_marcas"];
        $fk_dispositivos_modelos = $result["fk_dispositivos_modelos"];
        $fk_organizacao_unidades = $result["fk_organizacao_unidades"];
        $comunidade_snmp_dispositivo = $result["comunidade_snmp_dispositivo"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE DISPOSITIVOS" : "ALTERAÇÃO DE
DISPOSITIVOS";
?>

<form name="cadastrar_dispositivo" id="cadastrar_dispositivo" action="?acao=<?echo
$acao;?>" class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Nome*</label><br />
            <input name="nome_dispositivo" id="nome_dispositivo"
type="text" value="<?php echo $nome_dispositivo; ?>" style="width: 160px" />
        </p>
        <p>
            <label>Unidade*</label><br />
            <select data-placeholder="Selecione..."
id="fk_organizacao_unidades" name="fk_organizacao_unidades" style="width: 220px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("SELECT id_unidade,
sigla_instituicao, sigla_unidade, nome_unidade
FROM organizacao_unidades ou
JOIN organizacao_instituicoes oi
ON ou.`fk_organizacao_instituicoes` = oi.`id_instituicao`
ORDER BY sigla_instituicao, sigla_unidade", array());
                    print "<option value=''></option>";
                    if (!empty($result)){
                        foreach ($result as $res) {
                            $id_unidade = $res["id_unidade"];
                            $desc_unidade = "{$res["sigla_instituicao"]} -
{$res["sigla_unidade"]} [{$res["nome_unidade"]}];
                            $checked = "";
                            if ($fk_organizacao_unidades ==
$id_unidade){ $checked = "selected='selected'"; }
                            echo "<option $checked
value='$id_unidade'>$desc_unidade</option>";
                        }
                    }
                }
            }
        </div>
    </div>
</form>

```

```

    }
    ?>
</select>
</p>

<p>
    <label>Local</label><br />
    <input name="Local_dispositivo" id="Local_dispositivo"
type="text" value="<?php echo $local_dispositivo; ?>" style="width: 200px" />
</p>

<p>
    <label>Descrição</label><br />
    <input name="descricao_dispositivo" id="descricao_dispositivo"
type="text" value="<?php echo $descricao_dispositivo; ?>" style="width: 200px" />
</p>

<div class="clear"></div>

<p>
<fieldset style='margin-top: 5px; float: left; '>
    <label>Endereços IPv4</label><br />
    <div style='margin: 5px;' id='ipsv4'>
        <?php
        if (!empty($ipsv4)) {
            foreach ($ipsv4 as $ipv4) { ?>
                <div>
                    <input type='button' value='+'
class='addipv4' />
                    <input type='button' value='- '
class='removeipv4' />
                    <input type='text' style='width:
200px;' id='enderecosipv4_0' name='enderecos_ipv4[]' value='<?php echo $ipv4; ?>' />
                </div>
                <?php } } else { ?>
                <div>
                    <input type='button' value='+'
class='addipv4' />
                    <input type='button' value='- '
class='removeipv4' />
                    <input type='text' style='width:
200px;' id='enderecosipv4_0' name='enderecos_ipv4[]' />
                </div>
                <?php } ?>
            </div>
        </fieldset>
</p>

<p>
<fieldset style='margin-top: 5px; float: left; '>
    <label>Endereços IPv6</label><br />
    <div style='margin: 5px;' id='ipsv6'>
        <?php
        if (!empty($ipsv6)) {
            foreach ($ipsv6 as $ipv6) { ?>
                <div>
                    <input type='button' value='+'
class='addipv6' />

```



```

        $result = $DB->execPreparedSQL("select id_modelo_dispositivo,
nome_modelo_dispositivo from dispositivos_modelos WHERE fk_dispositivos_marcas = ? order by
nome_modelo_dispositivo", array('i', $DB->escape($selected)));
        print "<option value='0'></option>";
        if (!empty($result)){
            foreach ($result as $res) {
                $id_modelo = $res["id_modelo_dispositivo"];
                $modelo = $res["nome_modelo_dispositivo"];
                $checked = "";
                if ($fk_dispositivos_modelos ==
$id_modelo){ $checked = "selected='selected'"; }
                echo "<option $checked
value='$id_modelo'$modelo</option>";
            }
        }
    ?>
</select>
</p>

<p>
    <label>Nº de Série</label><br />
    <input name="nserie_dispositivo" id="nserie_dispositivo"
type="text" value="<?php echo $nserie_dispositivo; ?>" style="width: 210px" />
</p>

    <div class="clear"></div>
</div>

<div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
        $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
        <input name="voltar" id="voltar" type="button" value="Voltar" />
    <div id="loading" style="float:right;"></div>
</div>
<div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
    <div id="status_cadastrar_dispositivo"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_dispositivo', '#status_cadastrar_dispositivo');
    $("#voltar").click(function(){ voltar(); });
    $("#fk_dispositivos_marcas, #fk_dispositivos_modelos,
#fk_organizacao_unidades").chosen({allow_single_deselect: true, search_contains: true});
    $("#fk_dispositivos_marcas").change(function() {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getModelos', 'select': $('#fk_dispositivos_marcas').val() },
            success:function(response){
                $('#fk_dispositivos_modelos').html("").html(response.options);
            }
        });
    });
});

```

```

        $("#fk_dispositivos_modelos").trigger("liszt:updated");
    },
    error: function() {
        $('#fk_dispositivos_modelos').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
        $('#fk_dispositivos_modelos').next().children().css("border-color",
"#9F6000");
        $('#status_cadastrar_dispositivo').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />Opções de Modelos não
disponíveis.</div>');
    }
});
});

function criaCamposIP(objeto, tipo){
    div = objeto.parent();
    divpai = objeto.parent().parent();
    clone = div.clone(true);
    clone.children().each(function(){
        if($(this).attr("type")==="text"){ $(this).removeClass().val(''); }
    });
    clone.insertAfter(div);
    if ($.add+tipo).length == '5'){
        $('#add'+tipo).addClass("disabled").attr("disabled", true); }
    var cont = 0;
    divpai.find($('input[type=text]')).each(function(){
        $(this).attr("id", "enderecos"+tipo+"_"+cont);
        cont = cont + 1;
    });
}

function removeCamposIP(objeto, tipo){
    if ($.remove+tipo).length == '1'){
        objeto.next().val("");
    } else {
        div = objeto.parent();
        divpai = objeto.parent().parent();
        div.remove();
        if ($.add+tipo).length < '5'){
            $('#add'+tipo).removeClass("disabled").attr("disabled", false); }
        var cont = 0;
        divpai.find($('input[type=text]')).each(function(){
            $(this).attr("id", "enderecos"+tipo+"_"+cont);
            cont = cont + 1;
        });
    }
}

$('#addipv4').click(function(){
    criaCamposIP($(this), 'ipv4');
});

$('#addipv6').click(function(){
    criaCamposIP($(this), 'ipv6');
});

$('#removeipv4').click(function(){
    removeCamposIP($(this), 'ipv4');
});

```

```

        $('removeipv6').click(function(){
            removeCamposIP($(this), 'ipv6');
        });
    });
</script>

<?php
require_once(dirname(__FILE__) . '/../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleDispositivo.php");
require_once(CONTROLLER_PATH . "ControleModelo.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleDispositivo::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleDispositivo::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleDispositivo::getInstance();
        exit($controller->showInfo());

    case 'getModelos':
        $controller = ControleModelo::getInstance();
        exit($controller->getModelos());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
        ));
        $template->set_content_file('gerenciaveis/dispositivos/dispositivos.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_GERENCIAVEIS);
        $template->set_sub_module_item(ID_SUBMODULO_GERENCIAVEIS_DISPOSITIVOS);
        $template->show();
    }
}

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

```

```

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleMarca.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $nome_marca_dispositivo = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleMarca::getInstance();
        $result = $controle->getInfo();
        $nome_marca_dispositivo = $result["nome_marca_dispositivo"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE MARCAS" : "ALTERAÇÃO DE MARCAS";
?>
<form name="cadastrar_marca" id="cadastrar_marca" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Nome da Marca*</label><br />
            <input name="nome_marca_dispositivo"
id="nome_marca_dispositivo" type="text" value="<?php echo $nome_marca_dispositivo; ?>"
style="width: 290px" />
        </p>
        <div class="clear"></div>
    </div>

    <div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
        $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
        <input name="voltar" id="voltar" type="button" value="Voltar" />
        <div id="loading" style="float:right;"></div>
    </div>
    <div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
    <div id="status_cadastrar_marca"></div>

</form>
<?php } else { ?>

```

```

        <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">

$(document).ready(function() {
    createForm('#cadastrar_marca', '#status_cadastrar_marca');
    $("#voltar").click(function(){ voltar(); });
});
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleMarca.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';

switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleMarca::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleMarca::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleMarca::getInstance();
        exit($controller->getTableData());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
        ));
        $template->set_content_file('gerenciaveis/marcas/marcas.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_GERENCIAVEIS);
        $template->set_sub_module_item(ID_SUBMODULO_GERENCIAVEIS_MARCAS);
        $template->show();
    }

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_GERENCIAVEIS_MARCAS)) {
    unauthorized();
}

global $somente_leitura;

```

```

?>
<fieldset class="rounded">
  <legend>Marcas</legend>

  <div id="showtable">
    <table id="table"></table>
    <div id="pager"></div>
  </div>

  <div id='buttons'>
    <?php if (!$somete_leitura) { ?>
      <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Marca' />
      <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
      <?php } ?>
    </div>

    <div id='status'></div>
    <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
  var grid = $('#table');

  var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

  grid.jqGrid({
    url: '?acao=getTableData',
    datatype: 'json',
    autowidth: true,
    height: 270,
    colNames:['Marca' <?php echo (!$somete_leitura ? "','" : "" );?>],
    colModel:[
      {name:'nome_marca_dispositivo', index:'marca_dispositivo', resizable:false,
width:'100%',align:'center'}
      <?php echo (!$somete_leitura ?",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum:10,
    rowList:[10,30,100],
    scrollOffset: 17,
    gridview: true,
    pager: '#pager',
    sortname: 'nome_marca_dispositivo',
    sortOrder: "asc",
    toolbar: [false,'top'],
    viewrecords: true,
    shrinkToFit: true,
    hoverrows: false,
    beforeSelectRow: function(rowId, e){
      if(e.target.type == "checkbox"){
        verificaCheckbox($('#table'),$('#delete'));
        return true;
      } else if(e.target.className == "edit") {

```

```

        $('#showtable, #buttons, #status').hide();
        $('#formulario').load("formMarca.php?acao=alterar&id="+rowId).show();
        //displayColorbox("formMarca.php?acao=alterar&id="+rowId, "335px");
        return false;
    }
},
onSelectAll: function(){
    verificaCheckbox($('#table'), $('#delete'));
},
loadComplete: function(response){
    processLoadComplete(response);
    verificaCheckbox($('#table'), $('#delete'));
},
<?php if (!$somete_leitura) { echo 'multiselect: true, ' ;} ?>
search : true,
search : {
    caption: "Procurar..."
},
caption: campoProcurar,
loadError : function(xhr, st, err) {
    showError($('#status'));
}
});

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid', '#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formMarca.php?acao=cadastrar").show();
    //displayColorbox("formMarca.php?acao=cadastrar", "335px");
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");

```

```

require_once(INCLUDE_PATH ."/Auth.php");
require_once(CONTROLLER_PATH ."/ControleModelo.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $id_marca_dispositivo = "";
    $nome_modelo_dispositivo = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleModelo::getInstance();
        $result = $controle->getInfo();
        $id_marca_dispositivo = $result["fk_dispositivos_marcas"];
        $nome_modelo_dispositivo = $result["nome_modelo_dispositivo"];
    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE MODELOS" : "ALTERAÇÃO DE MODELOS";
?>
<form name="cadastrar_modelo" id="cadastrar_modelo" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Marca</label><br />
            <select data-placeholder="Selecione..."
id="fk_dispositivos_marcas" name="fk_dispositivos_marcas" style="width: 170px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("select id_marca_dispositivo,
nome_marca_dispositivo from dispositivos_marcas order by nome_marca_dispositivo", array());
                    print "<option value=''></option>";
                    if (!empty($result)){
                        foreach ($result as $res) {
                            $id_marca = $res["id_marca_dispositivo"];
                            $nome = $res["nome_marca_dispositivo"];
                            $checked = "";
                            if ($id_marca_dispositivo == $id_marca){
                                $checked = "selected='selected'"; }
                            echo "<option $checked
value='$id_marca'>$nome</option>";
                        }
                    }
                ?>
            </select>
        </p>
        <p>
            <label>Nome*</label><br />
            <input name="nome_modelo_dispositivo"
id="nome_modelo_dispositivo" type="text" value="<?php echo $nome_modelo_dispositivo; ?>"
style="width: 240px" />

```



```

        </p>
        <div class="clear"></div>
    </div>

    <div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
            $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
            <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
            <input name="voltar" id="voltar" type="button" value="Voltar" />
            <div id="loading" style="float:right;"></div>
        </div>
        <div class="clear"></div>
        <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
        <div id="status_cadastrar_modelo"></div>
</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_modelo', '#status_cadastrar_modelo');
    $("#voltar").click(function(){ voltar(); });
    $("#fk_dispositivos_marcas").chosen({search_contains: true});
});
</script>

<?php
require_once(dirname(__FILE__) . '/../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleModelo.php");
require_once(CONTROLLER_PATH . "ControleMarca.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';

switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleModelo::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleModelo::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleModelo::getInstance();
        exit($controller->getTableData());

    case 'getMarcas':
        $controller = ControleMarca::getInstance();
        exit($controller->getMarcas());

    default:

```

```

        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
        ));
        $template->set_content_file('gerenciaveis/modelos/modelos.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_GERENCIAVEIS);
        $template->set_sub_module_item(ID_SUBMODULO_GERENCIAVEIS_MODELOS);
        $template->show();
    }

    <?php
    require_once(INCLUDE_PATH . "Auth.php");
    $auth = new Auth();
    if (!$auth->check(ID_SUBMODULO_GERENCIAVEIS_MODELOS)) {
        unauthorized();
    }

    global $somente_leitura;

    ?>
    <fieldset class="rounded">
        <legend>Modelos</legend>

        <div id="showtable">
            <table id="table"></table>
            <div id="pager"></div>
        </div>

        <div id='buttons'>
            <?php if (!$somente_leitura) { ?>
                <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
                0 8px 8px;' value='Cadastrar Modelo' />
                <input id='delete' disabled='disabled' class='disabled' type='button'
                alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
            <?php } ?>
        </div>

        <div id='status'></div>
        <div id='formulario'></div>
    </fieldset>

    <script type="text/javascript">
    $(function() {
        var grid = $('#table');

        var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
        id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

        grid.jqGrid({
            url: '?acao=getTableData',
            datatype: 'json',
            autowidth: true,
            height: 270,

```

```

        colNames:['Marca', 'Modelo' <?php echo (!$somete_leitura ? "','" : "" );?>],
        colModel:[
            {name:'nome_marca_dispositivo', index:'nome_marca_dispositivo',
resizable:false, width:'50%',align:'center'},
            {name:'nome_modelo_dispositivo', index:'nome_modelo_dispositivo',
resizable:false, width:'50%',align:'center'}
            <?php echo (!$somete_leitura ?",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
        ],
        rowNum:10,
        rowList:[10,30,100],
        scrollOffset: 17,
        gridView: true,
        pager: '#pager',
        sortname: 'nome_marca_dispositivo',
        sortOrder: "asc",
        toolbar: [false,'top'],
        viewrecords: true,
        shrinkToFit: true,
        hoverrows: false,
        beforeSelectRow: function(rowId, e){
            if(e.target.type == "checkbox"){
                verificaCheckbox($('#table'),$('#delete'));
                return true;
            } else if(e.target.className == "edit") {
                $('#showtable, #buttons, #status').hide();
                $('#formulario').load("formModelo.php?acao=alterar&id="+rowId).show();
                //displayColorbox("formModelo.php?acao=alterar&id="+rowId, "475px");
                return false;
            }
        },
        onSelectAll: function(){
            verificaCheckbox($('#table'),$('#delete'));
        },
        loadComplete: function(response){
            processLoadComplete(response);
            verificaCheckbox($('#table'),$('#delete'));
        },
        <?php if (!$somete_leitura) { echo 'multiselect: true,' ;} ?>
        search : true,
        search : {
            caption: "Procurar..."
        },
        caption:campoProcurar,
        loadError : function(xhr,st,err) {
            showError($('#status'));
        }
    });

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

```

```

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formModelo.php?acao=cadastrar").show();
    //displayColorbox("formModelo.php?acao=cadastrar", "475px");
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleOperadora.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $nome_operadora = "";
    $resp_abertura_chamado = "";
    $tel_abertura_chamado = "";
    $resp_escalonamento_nivel_1 = "";
    $tel_escalonamento_nivel_1 = "";
    $resp_escalonamento_nivel_2 = "";
    $tel_escalonamento_nivel_2 = "";

    $selected = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleOperadora::getInstance();
        $result = $controle->getInfo();

        $nome_operadora = $result["nome_operadora"] ? $result["nome_operadora"] : "";
        $resp_abertura_chamado = $result["resp_abertura_chamado"] ?
$result["resp_abertura_chamado"] : "";
        $tel_abertura_chamado = $result["tel_abertura_chamado"] ?
$result["tel_abertura_chamado"] : "";
        $resp_escalonamento_nivel_1 = $result["resp_escalonamento_nivel_1"] ?
$result["resp_escalonamento_nivel_1"] : "";

```

```

        $tel_escalamento_nivel_1 = $result["tel_escalamento_nivel_1"] ?
$result["tel_escalamento_nivel_1"] : "";
        $resp_escalamento_nivel_2 = $result["resp_escalamento_nivel_2"] ?
$result["resp_escalamento_nivel_2"] : "";
        $tel_escalamento_nivel_2 = $result["tel_escalamento_nivel_2"] ?
$result["tel_escalamento_nivel_2"] : "";

    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE OPERADORAS" : "ALTERAÇÃO DE
OPERADORAS";
    ?>

<form name="cadastrar_operadora" id="cadastrar_operadora"
    action="?acao=<?echo $acao;?>" class="inline">
    <div style='text-align: center; '>
        <strong><?=$titulo?> </strong>
    </div>
    <div id='fieldset'>
        <p>
            <label>Nome* </label><br /> <input name="nome_operadora"
id="nome_operadora" type="text" value="<?php echo $nome_operadora; ?>" style="width: 265px"
/>
        </p>

        <div class="clear"></div>

        <p>
            <label>Responsável Abertura de Chamado </label><br /> <input
name="resp_abertura_chamado" id="resp_abertura_chamado" type="text" value="<?php echo
$resp_abertura_chamado; ?>" style="width: 265px" />
        </p>

        <p>
            <label>Telefone </label><br /> <input name="tel_abertura_chamado"
id="tel_abertura_chamado" type="text" value="<?php echo $tel_abertura_chamado; ?>"
style="width: 95px" />
        </p>

        <div class="clear"></div>

        <p>
            <label>Responsável Escalonamento Nível 1 </label><br /> <input
name="resp_escalamento_nivel_1" id="resp_escalamento_nivel_1" type="text" value="<?php
echo $resp_escalamento_nivel_1; ?>" style="width: 265px" />
        </p>

        <p>
            <label>Telefone </label><br /> <input name="tel_escalamento_nivel_1"
id="tel_escalamento_nivel_1" type="text" value="<?php echo $tel_escalamento_nivel_1;
?>" style="width: 95px" />
        </p>

        <div class="clear"></div>

        <p>
            <label>Responsável Escalonamento Nível 2 </label><br /> <input
name="resp_escalamento_nivel_2" id="resp_escalamento_nivel_2" type="text" value="<?php
echo $resp_escalamento_nivel_2; ?>" style="width: 265px" />

```

```

        </p>

        <p>
            <label>Telefone</label><br /> <input name="tel_escalonamento_nivel_2"
            id="tel_escalonamento_nivel_2" type="text" value="<?php echo $tel_escalonamento_nivel_2;
            ?>" style="width: 95px" />
        </p>

        <div class="clear"></div>

    </div>

    <div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
        $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php echo
        $button_value;?>" /> <input name="voltar" id="voltar" type="button" value="Voltar" />
        <div id="loading" style="float: right;"></div>
    </div>
    <div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
    <div id="status_cadastrar_operadora"></div>

</form>
<?php } else { ?>
<div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_operadora', '#status_cadastrar_operadora');
    $('#voltar').click(function(){ voltar(); });

    $('#tel_abertura_chamado, #tel_escalonamento_nivel_1,
    #tel_escalonamento_nivel_2').mask('(00) 0000-0000',
        {onKeyPress:
            function(phone, e, currentField, options){
                var new_phone = phone.match(/^(\(11\) 9(5[0-9]|6[0-9]|7[01234569]|8[0-
                9]|9[0-9])[0-9]{1})/g);
                new_phone ? $(currentField).mask('(00) 00000-0000', options) :
                $(currentField).mask('(00) 0000-0000', options)
            }
        });
});
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleOperadora.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

```

```

        $controller = ControleOperadora::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleOperadora::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleOperadora::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleOperadora::getInstance();
        exit($controller->showInfo());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
            "jquery.mask.min.js"
        ));
        $template->
>set_content_file('gerenciaveis/operadoras/operadoras.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_GERENCIAVEIS);
        $template->set_sub_module_item(ID_SUBMODULO_GERENCIAVEIS_OPERADORAS);
        $template->show();
    }

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_GERENCIAVEIS_OPERADORAS)) {
    unauthorized();
}

global $somete_leitura;

?>
<fieldset class="rounded">
    <legend>Operadoras</legend>

    <div id="showtable">
        <table id="table"></table>
        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somete_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Operadora' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
        <?php } ?>

```

```

</div>

<div id='status'></div>
<div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames:['Nome', 'Responsável (Abertura de Chamado)', 'Telefone (Abertura de
Chamado)' <?php echo (!$samente_leitura ? ",'" : "" );?>],
        colModel:[
            {name:'nome_operadora', index:'nome_operadora', resizable:false,
width:'30%',align:'center'},
            {name:'resp_abertura_chamado', index:'resp_abertura_chamado', resizable:false,
width:'35%',align:'center'},
            {name:'tel_abertura_chamado', index:'tel_abertura_chamado', resizable:false,
width:'35%',align:'center'}
            <?php echo (!$samente_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
        ],
        rowNum:10,
        rowList:[10,30,100],
        scrollOffset: 17,
        gridview: true,
        pager: '#pager',
        sortname: 'nome_operadora',
        sortOrder: "asc",
        toolbar: [false,'top'],
        viewrecords: true,
        shrinkToFit: true,
        hoverrows: false,
        beforeSelectRow: function(rowId, e){
            if(e.target.type == "checkbox"){
                verificaCheckbox($('#table'),$('#delete'));
                return true;
            } else if(e.target.className == "edit") {
                $('#showtable, #buttons, #status').hide();
                $('#formulario').load("formOperadora.php?acao=alterar&id="+rowId).show();
                return false;
            } else {
                grid.jqGrid('toggleSubGridRow', rowId);
            }
        },
        subGrid: true,
        subGridOptions: {
            plusicon : "ui-icon-triangle-1-e",
            minusicon : "ui-icon-triangle-1-s",
            openicon : "ui-icon-arrowreturn-1-e",

```



```

        expandOnLoad : false
    },
    subGridRowExpanded: function(subgrid_id, row_id) {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'showInfo', id: row_id },
            beforeSend: function(){
                $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
            },
            success: function(response){
                if (response.status == 'success'){
                    $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>"+response.info+"</div>");
                } else if (response.status == 'error'){
                    showError($("div[id='"+subgrid_id+"'"));
                } else if (response.status == 'unauthorized'){
                    location.reload(true);
                }
            },
            error: function(){
                showError($("div[id='"+subgrid_id+"'"));
            }
        });
    },
    onSelectAll: function(){
        verificaCheckbox($('#table'), $('#delete'));
    },
    loadComplete: function(response){
        processLoadComplete(response);
        verificaCheckbox($('#table'), $('#delete'));
    },
    <?php if (!$somete_leitura) { echo 'multiselect: true, ' ; } ?>
    search : true,
    search : {
        caption: "Procurar..."
    },
    caption: campoProcurar,
    loadError : function(xhr, st, err) {
        showError($('#status'));
    }
});
$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid', '#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formOperadora.php?acao=cadastrar").show();
});

```

```

        $("#delete").click(function(){
            deleteFromGrid();
        });
    });
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleManutencao.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $fk_dispositivos = "";
    $port = "";
    $start_time = "";
    $end_time = "";
    $descricao_manutencao = "";
    $inicio_manutencao = "";
    $fim_manutencao = "";

    $selected = "";

    if ($acao == "alterar") {
        if (isset($_GET['id'])){
            $id = anti_injection($_GET['id']);
        } else {
            exit('<p class="error">ID não especificado.</p>');
        }

        $DB = DB::getInstance();
        $controle = ControleManutencao::getInstance();
        $result = $controle->getInfo();
        $fk_dispositivos = $result["fk_dispositivos"];
        $ip_dispositivo = $result["ip_dispositivo"];
        $descricao_manutencao = $result["descricao_manutencao"];
        $start_time = date("d-m-Y H:i", $res['start_time']);
        $end_time = date("d-m-Y H:i", $res['end_time']);

    }

    $titulo = ($acao == "cadastrar") ? "CADASTRO DE MANUTENÇÕES" : "ALTERAÇÃO DE MANUTENÇÕES";
?>

<form name="cadastrar_manutencao" id="cadastrar_manutencao" action="?acao=<?echo $acao;?>" class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>

```

```

<div id='fieldset'>
    <p>
        <label>Dispositivo Alvo*</label><br />
        <select data-placeholder="Selecione..." id="fk_dispositivos"
name="fk_dispositivos" style="width: 230px">
            <?php
                $DB = DB::getInstance();
                $result = $DB->execPreparedSQL("SELECT id_dispositivo,
nome_dispositivo
FROM dispositivos d
ORDER BY nome_dispositivo", array());
                print "<option value=''></option>";
                if (!empty($result)){
                    foreach ($result as $res) {
                        $id_dispositivo = $res["id_dispositivo"];
                        $nome_dispositivo = "{$res["nome_dispositivo"]}";
                        $checked = "";
                        if ($fk_dispositivos == $id_dispositivo){
                            $checked = "selected='selected'"; $selected = $id_dispositivo; }
                        echo "<option $checked
value='{$id_dispositivo}'>$nome_dispositivo</option>";
                    }
                }
            ?>
        </select>
    </p>
    <div id='div_ip_alvo'>
        <p>
            <label>IP Dispositivo Alvo*</label><br />
            <select data-placeholder="Selecione..." id="ip_dispositivo"
name="ip_dispositivo" style="width: 280px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("select ipv4_dispositivo,
ipv6_dispositivo from dispositivos WHERE id_dispositivo = ? order by ipv4_dispositivo",
array('i', $DB->escape($selected)));
                    print "<option value=''></option>";
                    if (!empty($result)){
                        foreach ($result as $res) {
                            $ips4 = explode("|",$res["ipv4_dispositivo"]);
                            $ips6 = explode("|",$res["ipv6_dispositivo"]);
                            $ips = array_merge($ips4, $ips6);
                            foreach ($ips as $ip){
                                $checked = "";
                                if ($ip_dispositivo == $ip){ $checked =
"selected='selected'"; }
                                echo "<option $checked
value='{$ip}'>$ip</option>";
                            }
                        }
                    }
                ?>
            </select>
        </p>
    </div>

```

```

        <p>
            <label>Descrição*</label><br />
            <input name="descricao_manutencao" id="descricao_manutencao"
type="text" value="<?php echo $descricao_manutencao; ?>" style="width: 300px" />
        </p>

        <p>
            <label>Data de Início*</label><br />
            <input name="inicio_data_manutencao"
id="inicio_data_manutencao" type="text" value="<?php echo $inicio_manutencao; ?>"
style="width: 105px" />
        </p>

        <p>
            <label>Hora de Início*</label><br />
            <input name="inicio_hora_manutencao"
id="inicio_hora_manutencao" type="text" value="<?php echo $inicio_manutencao; ?>"
style="width: 105px" />
        </p>

        <p>
            <label>Data de Fim*</label><br />
            <input name="fim_data_manutencao" id="fim_data_manutencao"
type="text" value="<?php echo $inicio_manutencao; ?>" style="width: 105px" />
        </p>

        <p>
            <label>Hora de Fim*</label><br />
            <input name="fim_hora_manutencao" id="fim_hora_manutencao"
type="text" value="<?php echo $inicio_manutencao; ?>" style="width: 105px" />
        </p>

        <div class="clear"></div>

        <div id='fieldset_ep' style='margin-top: 10px; display:none;'>
            <div id='snmpi' style="display: none;">
                <fieldset class='rounded' id='portas_field'>
                    <legend>Portas:</legend>
                    <input name="portas" id="portas" type="button"
value="Buscar Portas" />

                    <div id='portas_indisponiveis'></div>
                    <div class="clear"></div>
                    <fieldset id='field_snmpi' class='rounded'

                    <div id='snmpi_portas' class='rounded'>

                    </div>
                    </fieldset>
                </fieldset>
                <div class="clear"></div>
            </div>
            <div class="clear"></div>
        </div>

        <div class="clear" style="margin-bottom: 10px;"></div>
        <div id='botao' style="float: left;">

```

```

        <?php
        $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
        <input name="voltar" id="voltar" type="button" value="Voltar" />
        <div id="loading" style="float:right;"></div>
    </div>
    <div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
    <div id="status_cadastrar_manutencao"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_manutencao', '#status_cadastrar_manutencao');
    $('#voltar').click(function(){ voltar(); });
    $('#fk_dispositivos, #ip_dispositivo').chosen({allow_single_deselect: false,
search_contains: true});

    $('#fk_dispositivos').change(function() {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getIps', 'select': $('#fk_dispositivos').val() },
            success: function(response){
                $('#ip_dispositivo').html("").html(response.options);
                $('#ip_dispositivo').trigger("liszt:updated");
                $('#fieldset_ep').show();
                $('#snmpi').show();
            },
            error: function(){
                $('#ip_dispositivo').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
                $('#ip_dispositivo').next().children().css("border-color",
"#9F6000");
                $('#status_cadastrar_manutencao').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />IPs não disponíveis.</div>');
            }
        });
    });

    $('#portas').bind('click', function() {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getPortas', 'dispositivo': $('#fk_dispositivos').val(),
'ip': $('#ip_dispositivo').val(), 'id': '<?=$id?' },
            beforeSend: function(){
                $('#field_snmpi').show();
                $('#snmpi_portas').removeClass().html('<div
class="ajax_loading">Verificando portas...</div>');
            },
            success: function(response){
                if (response == null){
                    showError($('#snmpi_portas'));
                }
            }
        });
    });
}

```

```

        } else {
            if (response.ports_nd != undefined && response.ports_nd !=
""){
                $('#portas_indisponiveis').addClass("warning").html("As
seguintes portas não estão mais disponíveis para monitoramento:<br/>" + response.ports_nd);
            }
            $('#field_snmpi').show();

$('#snmpi_portas').removeClass().addClass(response.status).html(response.msg);
$('.checkbox').dgStyle();
$('#check_all_div').click(function() {
    var checked = $(this).children().is(':checked');

$(this).parent().parent().find('.checkbox').each(function() {
    if (checked){
        $(this).dgCheck($(this));

$(this).find('input[type="checkbox"]').attr('checked', true);
    } else {
        $(this).dgUncheck($(this));

$(this).find('input[type="checkbox"]').attr('checked', false);
    }
});
});
    },
    error:function(){
        showError($('#snmpi_portas'));
    }
});
});

$('#inicio_data_manutencao').datepicker({
    dateFormat: "dd/mm/yy",
    monthNames: ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho',
'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro'],
    onSelect: function (dateValue, inst) {
        $('#fim_data_manutencao').datepicker("option", "minDate", dateValue);
    }
});

$('#fim_data_manutencao').datepicker({
    dateFormat: "dd/mm/yy",
    monthNames: ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho',
'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro']
});

$('#inicio_data_manutencao, #fim_data_manutencao').mask("99/99/9999", { placeholder:
" " });

$('#inicio_hora_manutencao, #fim_hora_manutencao').timepicker({
    showPeriodLabels: false,
    minutes: { interval: 5 },
    showNowButton: false,
    showDeselectButton: true,
    defaultTime: '', // removes the highlighted time for when the input is empty.
    showCloseButton: true
});

```

```

});
    $('#inicio_hora_manutencao, #fim_hora_manutencao').mask("99:99", { placeholder: " "
});

    <?php if ($acao == 'alterar') echo '$("#portas").trigger("click");'; ?>

});
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleManutencao.php");
require_once(CONTROLLER_PATH . "ControleDispositivo.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleManutencao::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleManutencao::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleManutencao::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleManutencao::getInstance();
        exit($controller->showInfo());

    case 'getPortas':
        $controller = ControleManutencao::getInstance();
        exit($controller->getPortas());

    case 'getIps':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getIps());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
            "jquery.ui.timepicker.css"
        ));

        $template->set_js_files(array(
            "jquery.custom_radio_checkbox.js",
            "jquery.ui.timepicker.js",
            "jquery.ui.timepicker-pt-BR.js",
            "jquery.mask.min.js"
        ));

```

```

    ));
    $template->set_content_file('manutencoes/registrar/registrar.php');
    $template->set_authenticated_only(true);
    $template->set_module(ID_MODULO_SISTEMA);
    $template->set_sub_module(ID_SUBMODULO_MANUTENCOES);
    $template->set_sub_module_item(ID_SUBMODULO_MANUTENCOES_REGISTRAR);
    $template->show();
}

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_MANUTENCOES_REGISTRAR)) {
    unauthorized();
}

global $somente_leitura;

?>
<fieldset class="rounded">
    <legend>Manutenções</legend>

    <div id="showtable">
        <table id="table"></table>
        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somente_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Manutenção' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Selecionados' />
            <?php } ?>
        </div>

        <div id='status'></div>
        <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames:['Dispositivo', 'Início', 'Fim' <?php echo (!$somente_leitura ? ",'" : "'";
    );?>],
        colModel:[
            {name:'nome_dispositivo', index:'nome_dispositivo', resizable:false,
width:'20%',align:'center'},

```



```

        {name:'start_time', index:'start_time', resizable:false,
width:'20%',align:'center'},
        {name:'end_time', index:'end_time', resizable:false,
width:'20%',align:'center'}
        <?php echo (!$somete_leitura ?",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum:10,
    rowList:[10,30,100],
    scrolloffset: 17,
    gridview: true,
    pager: '#pager',
    sortname: 'nome_dispositivo',
    sortorder: "asc",
    toolbar: [false,'top'],
    viewrecords: true,
    shrinkToFit: true,
    hoverrows: false,
    beforeSelectRow: function(rowId, e){
        if(e.target.type == "checkbox"){
            verificaCheckbox($('#table'),$('#delete'));
            return true;
        } else if(e.target.className == "edit") {
            $('#showtable, #buttons, #status').hide();
            $('#formulario').load("formManutencao.php?acao=alterar&id="+rowId).show();
            return false;
        } else {
            grid.jqGrid('toggleSubGridRow', rowId);
        }
    },
    subGrid: true,
    subGridOptions: {
        plusicon : "ui-icon-triangle-1-e",
        minusicon : "ui-icon-triangle-1-s",
        openicon : "ui-icon-arrowreturn-1-e",
        expandOnLoad : false
    },
    subGridRowExpanded: function(subgrid_id, row_id) {
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'showInfo', id: row_id },
            beforeSend:function(){
                $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
            },
            success:function(response){
                if (response.status == 'success'){
                    $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>"+response.info+"</div>");
                } else if (response.status == 'error'){
                    showError($("#div[id='"+subgrid_id+"'"));
                } else if (response.status == 'unauthorized'){
                    location.reload(true);
                }
            },
            error:function(){
                showError($("#div[id='"+subgrid_id+"'"));
            }
        });
    });

```

```

    },
    onSelectAll: function(){
        verificaCheckbox($('#table'),$('#delete'));
    },
    loadComplete: function(response){
        processLoadComplete(response);
        verificaCheckbox($('#table'),$('#delete'));
    },
    <?php if (!$somento_leitura) { echo 'multiselect: true,' ;} ?>
    search : true,
    search : {
        caption: "Procurar..."
    },
    caption: campoProcurar,
    loadError : function(xhr,st,err) {
        showError($('#status'));
    }
});

$('#.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$('#gridsearch').val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formManutencao.php?acao=cadastrar").show();
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleMonitoramento.php");

global $somento_leitura;
if (!$somento_leitura) {

    $id = "";

```

```

$fk_acordos = "";
$fk_dispositivos = "";
$fk_dispositivos_verificador = "";
$descricao_mon = "";
$ping_count = 10;
$ping_size = 100;
$ip_disp_alvo = "";
$ip_neighbour = "";
$fk_acordos_mon_tp = "";

$selected = "";

if ($acao == "alterar") {
    if (isset($_GET['id'])) {
        $id = anti_injection($_GET['id']);
    } else {
        exit('<p class="error">ID não especificado.</p>');
    }

    $DB = DB::getInstance();
    $controle = ControleMonitoramento::getInstance();
    $result = $controle->getInfo();
    $fk_acordos = $result["fk_acordos"];
    $fk_dispositivos = $result["fk_dispositivos"];
    $fk_dispositivos_verificador = $result["fk_dispositivos_verificador"];
    $fk_acordos_mon_tipos = $result["fk_acordos_mon_tipos"];
    $descricao_mon = $result["descricao_mon"];
    $ip_disp_alvo = $result["ip_disp_alvo"];

    $fk_acordos_mon_tp = "$fk_acordos###$fk_acordos_mon_tipos";

    $ping_count = $result["ping_count"] ? $result["ping_count"] : "";
    $ping_size = $result["ping_size"] ? $result["ping_size"] : "";

    $ip_neighbour = $result["ip_neighbour"] ? $result["ip_neighbour"] : "";
}

$titulo = ($acao == "cadastrar") ? "CADASTRO DE MONITORAMENTOS" : "ALTERAÇÃO DE
MONITORAMENTOS";
?>

<form name="cadastrar_monitoramento" id="cadastrar_monitoramento" action="?acao=<?echo
$acao;?>" class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Acordo*</label><br />
            <select <?php if ($acao == "alterar") echo "disabled='disabled'";
?> data-placeholder="Selecione..." id="fk_acordos" name="fk_acordos" style="width: 420px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("SELECT id_acordo, nome_acordo,
id_mon_tipo, descricao_mon_tipo, nome_tipo_acordo

FROM acordos a

LEFT JOIN acordos_mon_tipos amt ON a.`fk_acordos_mon_tipos`=amt.`id_mon_tipo`

```

```

LEFT JOIN `acordos_tipos` at ON a.`fk_acordos_tipos` = at.`id_tipo_acordo`

ORDER BY nome_acordo", array());
print "<option value=''></option>";
if (!empty($result)){
    foreach ($result as $res) {
        $id_acordo = $res["id_acordo"];
        $id_mon_tipo = $res["id_mon_tipo"];
        $nome_acordo = "{$res["nome_acordo"]}";
        $nome_tipo_acordo = "{$res["nome_tipo_acordo"]}";
        $tipo_mon = "{$res["descricao_mon_tipo"]}";
        $checked = "";
        if ($fk_acordos == $id_acordo){ $checked =
"selected='selected'"; }
        echo "<option $checked
value='$id_acordo###$id_mon_tipo'$>$nome_acordo [$nome_tipo_acordo] [$tipo_mon]</option>";
    }
}
?>
</select>
</p>

<p>
    <label>Descrição</label><br />
    <input name="descricao_mon" id="descricao_mon" type="text"
value="<?php echo $descricao_mon; ?>" style="width: 395px" />
</p>

<div class="clear"></div>

<p>
<label>Dispositivo Alvo*</label><br />
<select data-placeholder="Selecione..." id="fk_dispositivos"
name="fk_dispositivos" style="width: 230px">
    <?php
        $DB = DB::getInstance();
        $result = $DB->execPreparedSQL("SELECT id_dispositivo,
nome_dispositivo
FROM dispositivos d
ORDER BY nome_dispositivo", array());
        print "<option value=''></option>";
        if (!empty($result)){
            foreach ($result as $res) {
                $id_dispositivo = $res["id_dispositivo"];
                $nome_dispositivo = "{$res["nome_dispositivo"]}";
                $checked = "";
                if ($fk_dispositivos == $id_dispositivo){
$checked = "selected='selected'"; $selected = $id_dispositivo; }
                echo "<option $checked
value='$id_dispositivo'$>$nome_dispositivo</option>";
            }
        }
    }
?>
</select>
</p>

```

```

<p>
<label>IP Dispositivo Alvo*</label><br />
<select data-placeholder="Selecione..." id="ip_disp_alvo"
name="ip_disp_alvo" style="width: 350px">
<?php
    $DB = DB::getInstance();
    $result = $DB->execPreparedSQL("select ipv4_dispositivo,
ipv6_dispositivo from dispositivos WHERE id_dispositivo = ? order by ipv4_dispositivo",
array('i', $DB->escape($selected)));
    print "<option value=''></option>";
    if (!empty($result)){
        foreach ($result as $res) {
            $ips4 = explode("|",$res["ipv4_dispositivo"]);
            $ips6 = explode("|",$res["ipv6_dispositivo"]);
            $ips = array_merge($ips4, $ips6);
            foreach ($ips as $ip){
                $checked = "";
                if ($ip_disp_alvo == $ip){ $checked =
"selected='selected'"; }
                echo "<option $checked
value='$ip'>$ip</option>";
            }
        }
    }
?>
</select>
</p>

<p>
<label>Dispositivo Verificador*</label><br />
<select data-placeholder="Selecione..."
id="fk_dispositivos_verificador" name="fk_dispositivos_verificador" style="width: 235px">
<?php
    $DB = DB::getInstance();
    $result = $DB->execPreparedSQL("SELECT id_dispositivo,
nome_dispositivo
FROM dispositivos d
ORDER BY nome_dispositivo", array());
    print "<option value=''></option>";
    if (!empty($result)){
        foreach ($result as $res) {
            $id_dispositivo = $res["id_dispositivo"];
            $nome_dispositivo = "{$res["nome_dispositivo"]}";
            $checked = "";
            if ($fk_dispositivos_verificador ==
$id_dispositivo){ $checked = "selected='selected'"; }
            echo "<option $checked
value='$id_dispositivo'>$nome_dispositivo</option>";
        }
    }
?>
</select>
</p>

<div class="clear"></div>

<div id='fieldset_ep' style='margin-top: 10px; display:none;'>

```

```

        <div id='ping'>
            <p>
                <label>Qtd. Pacotes (Count)</label><br />
                <input name="ping_count" id="ping_count"
type="text" value="<?php echo $ping_count; ?>" style="width: 140px" />
            </p>

            <p>
                <label>Tamanho Bytes (Size)</label><br />
                <input name="ping_size" id="ping_size"
type="text" value="<?php echo $ping_size; ?>" style="width: 150px" />
            </p>

            <div class="clear"></div>
        </div>

        <div id='snmpi' style="display: none;">
            <fieldset class='rounded' id='portas_field'>
                <legend>Portas:</legend>
                <input name="portas" id="portas" type="button"
value="Buscar Portas" />

                <div id='portas_indisponiveis'></div>
                <div class="clear"></div>
                <fieldset id='field_snmpi' class='rounded'

                <div id='snmpi_portas' class='rounded'>

                    </div>
                </fieldset>
            </fieldset>
            <div class="clear"></div>
        </div>

        <div id='snmpo'>
            <p>
                <label>IP do Neighbour</label><br />
                <input name="ip_neighbour" id="ip_neighbour"
type="text" value="<?php echo $ip_neighbour; ?>" style="width: 300px" />
            </p>

            <div class="clear"></div>
        </div>

        <div class="clear"></div>
    </div>

    <div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
        $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
        <input name="voltar" id="voltar" type="button" value="Voltar" />
        <div id="loading" style="float:right;"></div>
    </div>
    <div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />

```

```

        <input name="fk_acordos_altera" id="fk_acordos_altera" type="hidden"
value="<?php echo $fk_acordos_mon_tp;?>" />
        <div id="status_cadastrar_mon"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_monitoramento', '#status_cadastrar_mon');
    $('#voltar').click(function(){ voltar(); });
    $('#fk_acordos, #fk_dispositivos, #fk_dispositivos_verificador,
#ip_disp_alvo').chosen({allow_single_deselect: false, search_contains: true});

    $('#fk_acordos').change(function() {
        $('#status_cadastrar_mon').removeClass().html("");
        if ($(this).val().split("###")[1] == 1){
            $('#ping').show();
            $('#snmpi').hide();
            $('#snmpo').hide();
            $('#fieldset_ep').show();
            $('#ping_count, #ping_size').attr('disabled',false);
            $('#input[name="ports[]"]').attr('disabled',true);
            $('#ip_neighbour').attr('disabled',true);
        } else if ($(this).val().split("###")[1] == 2) {
            $('#snmpi').show();
            $('#ping').hide();
            $('#snmpo').hide();
            $('#fieldset_ep').show();
            $('#ping_count, #ping_size').attr('disabled',true);
            $('#input[name="ports[]"]').attr('disabled',false);
            $('#ip_neighbour').attr('disabled',true);
        } else if ($(this).val().split("###")[1] == 3) {
            $('#snmpi').hide();
            $('#ping').hide();
            $('#snmpo').show();
            $('#fieldset_ep').show();
            $('#ping_count, #ping_size').attr('disabled',true);
            $('#input[name="ports[]"]').attr('disabled',true);
            $('#ip_neighbour').attr('disabled',false);
        }
    });

    $('#fk_acordos').trigger("change");

    $('#fk_dispositivos').change(function() {
        $('#field_snmpi').hide();
        $('#snmpi_portas').removeClass().html("");
        $.ajax({
            type: 'POST',
            dataType: 'json',
            data: { acao: 'getIps', 'select': $('#fk_dispositivos').val() },
            success: function(response){
                $('#ip_disp_alvo').html("").html(response.options);
                $("#ip_disp_alvo").trigger("liszt:updated");
            },
            error: function(){

```

```

        $('#ip_disp_alvo').html("<option
value=''>Indisponível</option>").trigger("liszt:updated");
        $('#ip_disp_alvo').next().children().css("border-color", "#9F6000");
        $('#status_cadastrar_mon').removeClass().html('<div
class="warning">Ocorreu um erro inesperado.<br />IPs não disponíveis.</div>');
    }
});
});

$("#ip_disp_alvo").change(function() {
    $('#field_snmpi').hide();
    $('#snmpi_portas').removeClass().html("");
});

$('#portas').bind('click', function() {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'getPortas', 'dispositivo': $('#fk_dispositivos').val(),
'ip': $('#ip_disp_alvo').val(), 'id': '<?=$id?>', 'tipo': 'monitoramentos' },
        beforeSend: function(){
            $('#field_snmpi').show();
            $('#snmpi_portas').removeClass().html('<div
class="ajax_loading">Verificando portas...</div>');
        },
        success: function(response){
            if (response == null){
                showError($('#snmpi_portas'));
            } else {
                if (response.ports_nd != undefined && response.ports_nd !=
""){
                    $('#portas_indisponiveis').addClass("warning").html("As
seguintes portas não estão mais disponíveis para monitoramento:<br/>" + response.ports_nd);
                }
                $('#field_snmpi').show();

                $('#snmpi_portas').removeClass().addClass(response.status).html(response.msg);
                $(".checkbox").dgStyle();
                $("#check_all_div").click(function() {
                    var checked = $(this).children().is(':checked');

                    $(this).parent().parent().find('.checkbox').each(function() {
                        if (checked){
                            $(this).dgCheck($(this));

                            $(this).find('input[type="checkbox"]').attr('checked', true);
                        } else {
                            $(this).dgUncheck($(this));

                            $(this).find('input[type="checkbox"]').attr('checked', false);
                        }
                    });
                });
            }
        },
        error: function(){
            showError($('#snmpi_portas'));
        }
    });
});

```



```

    });

    <?php if ($acao == 'alterar' && $fk_acordos_mon_tipos == 2) echo
'$("#portas").trigger("click");'; ?>

});
</script>

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR)) {
    unauthorized();
}

global $somente_leitura;

?>
<fieldset class="rounded">
    <legend>Monitoramentos</legend>

    <div id="showtable">
        <table id="table"></table>
        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somente_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Monitoramento' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
            <?php } ?>
        </div>

        <div id='status'></div>
        <div id='formulario'></div>
    </fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames:['Acordo', 'Tipo de Acordo', 'Tipo de Monit.', 'Dispositivo Alvo', 'IP do
Alvo' <?php echo (!$somente_leitura ? ",'" : "" );?>],
        colModel:[
            {name:'nome_acordo', index:'nome_acordo', resizable:false,
width:'25%',align:'center'},
            {name:'nome_tipo_acordo', index:'nome_tipo_acordo', resizable:false,
width:'22%',align:'center'},

```

```

        {name:'descricao_mon_tipo', index:'descricao_mon_tipo', resizable:false,
width:'23%',align:'center'},
        {name:'nome_dispositivo', index:'nome_dispositivo', resizable:false,
width:'25%',align:'center'},
        {name:'ip_disp_alvo', index:'ip_disp_alvo', resizable:false,
width:'30%',align:'center'}
        <?php echo (!$somenteleitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum:10,
    rowList:[10,30,100],
    scrollOffset: 17,
    gridView: true,
    pager: '#pager',
    sortname: 'nome_acordo',
    sortOrder: "asc",
    toolbar: [false,'top'],
    viewrecords: true,
    shrinkToFit: true,
    hoverrows: false,
    beforeSelectRow: function(rowId, e){
        if(e.target.type == "checkbox"){
            verificaCheckbox($('#table'),$('#delete'));
            return true;
        } else if(e.target.className == "edit") {
            $('#showtable, #buttons, #status').hide();
        }
    },
    $('#formulario').load("formGerenciarMonitoramento.php?acao=alterar&id="+rowId).show();
        return false;
    } else {
        grid.jqGrid('toggleSubGridRow', rowId);
    }
},
subGrid: true,
subGridOptions: {
    plusicon : "ui-icon-triangle-1-e",
    minusicon : "ui-icon-triangle-1-s",
    openicon : "ui-icon-arrowreturn-1-e",
    expandOnLoad : false
},
subGridRowExpanded: function(subgrid_id, row_id) {
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: { acao: 'showInfo', id: row_id },
        beforeSend:function(){
            $("div[id='"+subgrid_id+"'").html('<div class="ajax_loading"></div>');
        },
        success:function(response){
            if (response.status == 'success'){
                $("div[id='"+subgrid_id+"'").empty().html("<div
class='subgrid_info'>"+response.info+"</div>");
            } else if (response.status == 'error'){
                showError($("div[id='"+subgrid_id+"'"));
            } else if (response.status == 'unauthorized'){
                location.reload(true);
            }
        }
    },
    error:function(){

```

```

        showError($("#div[id='"+subgrid_id+"'"]);
    });
    },
    onSelectAll: function(){
        verificaCheckbox($('#table'),$('#delete'));
    },
    loadComplete: function(response){
        processLoadComplete(response);
        verificaCheckbox($('#table'),$('#delete'));
    },
    <?php if (!$somete_leitura) { echo 'multiselect: true,' ;} ?>
    search : true,
    search : {
        caption: "Procurar..."
    },
    caption: campoProcurar,
    loadError : function(xhr,st,err) {
        showError($('#status'));
    }
});

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formGerenciarMonitoramento.php?acao=cadastrar").show();
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleMonitoramento.php");
require_once(CONTROLLER_PATH . "ControleDispositivo.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';
switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleMonitoramento::getInstance();
        exit($controller->add_update());

```

```

    case 'deletar':
        $controller = ControleMonitoramento::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleMonitoramento::getInstance();
        exit($controller->getTableData());

    case 'showInfo':
        $controller = ControleMonitoramento::getInstance();
        exit($controller->showInfo());

    case 'getPortas':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getPortas());

    case 'getIps':
        $controller = ControleDispositivo::getInstance();
        exit($controller->getIps());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
            "jquery.custom_radio_checkbox.js"
        ));
        $template->
>set_content_file('monitoramentos/gerenciar/gerenciar.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_MONITORAMENTOS);
        $template->
>set_sub_module_item(ID_SUBMODULO_MONITORAMENTOS_GERENCIAR);
        $template->show();
    }

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php

$acao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../application/config/constantes.php');
require_once(INCLUDE_PATH . "/DB.php");
require_once(INCLUDE_PATH . "/Auth.php");
require_once(CONTROLLER_PATH . "/ControleInstituicao.php");

global $somente_leitura;
if (!$somente_leitura) {

    $id = "";
    $sigla_instituicao = "";

```

```

$nome_instituicao = "";

if ($acao == "alterar") {
    if (isset($_GET['id'])){
        $id = anti_injection($_GET['id']);
    } else {
        exit('<p class="error">ID não especificado.</p>');
    }

    $DB = DB::getInstance();
    $controle = ControleInstituicao::getInstance();
    $result = $controle->getInfo();
    $nome_instituicao = $result["nome_instituicao"];
    $sigla_instituicao = $result["sigla_instituicao"];
}

$titulo = ($acao == "cadastrar") ? "CADASTRO DE INSTITUIÇÕES" : "ALTERAÇÃO DE
INSTITUIÇÕES";

?>
<form name="cadastrar_instituicao" id="cadastrar_instituicao" action="?acao=<?echo
$acao;?>" class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Sigla*</label><br />
            <input name="sigla_instituicao" id="sigla_instituicao"
type="text" value="<?php echo $sigla_instituicao; ?>" style="width: 80px" />
        </p>
        <p>
            <label>Nome da Instituição*</label><br />
            <input name="nome_instituicao" id="nome_instituicao"
type="text" value="<?php echo $nome_instituicao; ?>" style="width: 290px" />
        </p>
        <div class="clear"></div>
    </div>

    <div class="clear" style="margin-bottom: 10px;"></div>
    <div id='botao' style="float: left;">
        <?php
$button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar";
?>
        <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value;?>" />
        <input name="voltar" id="voltar" type="button" value="Voltar" />
        <div id="loading" style="float:right;"></div>
    </div>
    <div class="clear"></div>
    <input name="id" id="id" type="hidden" value="<?php echo $id;?>" />
    <div id="status_cadastrar_instituicao"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {

```

```

        createForm('#cadastrar_instituicao', '#status_cadastrar_instituicao');
        $("##voltar").click(function(){ voltar(); });
    });
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleInstituicao.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';

switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleInstituicao::getInstance();
        exit($controller->add_update());

    case 'deletar':
        $controller = ControleInstituicao::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleInstituicao::getInstance();
        exit($controller->getTableData());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
        ));
        $template->set_content_file('organizacao/instituicoes/instituicoes.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_ORGANIZACAO);
        $template->set_sub_module_item(ID_SUBMODULO_ORGANIZACAO_INSTITUICOES);
        $template->show();
    }

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_ORGANIZACAO_INSTITUICOES)) {
    unauthorized();
}

global $somente_leitura;

?>
<fieldset class="rounded">
    <legend>Instituições</legend>

    <div id="showtable">

```

```

        <table id="table"></table>
        <div id="pager"></div>
    </div>

    <div id='buttons'>
        <?php if (!$somete_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Instituição' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Seleccionados' />
        <?php } ?>
    </div>

    <div id='status'></div>
    <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

    var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

    grid.jqGrid({
        url: '?acao=getTableData',
        datatype: 'json',
        autowidth: true,
        height: 270,
        colNames: ['Sigla', 'Nome' <?php echo (!$somete_leitura ? ",'" : "" );?>],
        colModel: [
            {name: 'sigla_instituicao', index: 'sigla_instituicao', resizable: false,
width: '20%', align: 'center'},
            {name: 'nome_instituicao', index: 'nome_instituicao', resizable: false,
width: '80%', align: 'center'}
            <?php echo (!$somete_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
        ],
        rowNum: 10,
        rowList: [10, 30, 100],
        scrollOffset: 17,
        gridview: true,
        pager: '#pager',
        sortname: 'sigla_instituicao',
        sortOrder: "asc",
        toolbar: [false, 'top'],
        viewrecords: true,
        shrinkToFit: true,
        hoverrows: false,
        beforeSelectRow: function(rowId, e){
            if(e.target.type == "checkbox"){
                verificaCheckbox($('#table'), $('#delete'));
                return true;
            } else if(e.target.className == "edit") {
                $('#showtable, #buttons, #status').hide();
                $('#formulario').load("formInstituicao.php?acao=alterar&id="+rowId).show();
                //displayColorbox("formInstituicao.php?acao=alterar&id="+rowId,
"435px");
            }
        }
    });
}

```

```

        return false;
    }
},
onSelectAll: function(){
    verificaCheckbox($('#table'),$('#delete'));
},
loadComplete: function(response){
    processLoadComplete(response);
    verificaCheckbox($('#table'),$('#delete'));
},
<?php if (!$somento_leitura) { echo 'multiselect: true,' ;} ?>
search : true,
search : {
    caption: "Procurar..."
},
caption: campoProcurar,
loadError : function(xhr,st,err) {
    showError($('#status'));
}
});

$('#.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url: '?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formInstituicao.php?acao=cadastrar").show();
    //displayColorbox("formInstituicao.php?acao=cadastrar", "435px");
});

$("#delete").click(function(){
    deleteFromGrid();
});
});
</script>

<script type="text/javascript">
if (typeof jQuery == 'undefined'){ window.location = 'about:blank'; }
</script>

<?php
$sacao = isset($_GET['acao']) ? $_GET['acao'] : exit('<p class="error">Ocorreu um erro inesperado. Entre em contato com o administrador do sistema.</p>');

require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH ."/DB.php");
require_once(INCLUDE_PATH ."/Auth.php");
require_once(CONTROLLER_PATH ."/ControleUnidade.php");

global $somento_leitura;
if (!$somento_leitura) {

```



```

$sid = "";
$fk_organizacao_instituicoes = "";
$sigla_unidade = "";
$nome_unidade = "";
$descricao_unidade = "";

if ($acao == "alterar") {
    if (isset($_GET['id'])){
        $sid = anti_injection($_GET['id']);
    } else {
        exit('<p class="error">ID não especificado.</p>');
    }

    $DB = DB::getInstance();
    $controle = ControleUnidade::getInstance();
    $result = $controle->getInfo();
    $fk_organizacao_instituicoes = $result["fk_organizacao_instituicoes"];
    $nome_unidade = $result["nome_unidade"];
    $sigla_unidade = $result["sigla_unidade"];
    $descricao_unidade = $result["descricao_unidade"];
}

$titulo = ($acao == "cadastrar") ? "CADASTRO DE UNIDADES" : "ALTERAÇÃO DE UNIDADES";
?>
<form name="cadastrar_unidade" id="cadastrar_unidade" action="?acao=<?echo $acao;?>"
class="inline">
    <div style='text-align:center;'><strong><?=$titulo?></strong></div>
    <div id='fieldset'>
        <p>
            <label>Sigla*</label><br />
            <input name="sigla_unidade" id="sigla_unidade" type="text"
value="<?php echo $sigla_unidade; ?>" style="width: 80px" />
        </p>

        <p>
            <label>Nome*</label><br />
            <input name="nome_unidade" id="nome_unidade" type="text"
value="<?php echo $nome_unidade; ?>" style="width: 270px" />
        </p>

        <p>
            <label>Instituição</label><br />
            <select data-placeholder="Selecione..."
id="fk_organizacao_instituicoes" name="fk_organizacao_instituicoes" style="width: 180px">
                <?php
                    $DB = DB::getInstance();
                    $result = $DB->execPreparedSQL("select
id_instituicao,sigla_instituicao from organizacao_instituicoes order by nome_instituicao",
array());

                    print "<option value=''></option>";
                    if (!empty($result)){
                        foreach ($result as $res) {
                            $sid_inst = $res["id_instituicao"];
                            $sigla = $res["sigla_instituicao"];
                            $checked = "";
                            if ($fk_organizacao_instituicoes ==
$sid_inst){ $checked = "selected='selected'"; }

```

```

value='{$id_inst'}$sigla</option>";
        }
    }
    ?>
</select>
</p>

<p>
    <label>Descrição</label><br />
    <input name="descricao_unidade" id="descricao_unidade"
type="text" value="<?php echo $descricao_unidade; ?>" style="width: 250px" />
</p>
<div class="clear"></div>
</div>

<div class="clear" style="margin-bottom: 10px;"></div>
<div id='botao' style="float: left;">
    <?php
    $button_value = ($acao == "cadastrar") ? "Cadastrar" : "Alterar"; ?>
    <input name="cadastrar" id="cadastrar" type="submit" value="<?php
echo $button_value; ?>" />
    <input name="voltar" id="voltar" type="button" value="Voltar" />
    <div id="loading" style="float:right;"></div>
</div>
<div class="clear"></div>
<input name="id" id="id" type="hidden" value="<?php echo $id; ?>" />
<div id="status_cadastrar_unidade"></div>

</form>
<?php } else { ?>
    <div class='error'>Você não tem privilégios para executar essa operação.</div>
<?php } ?>

<script type="text/javascript">
$(document).ready(function() {
    createForm('#cadastrar_unidade', '#status_cadastrar_unidade');
    $("#voltar").click(function(){ voltar(); });
    $("#fk_organizacao_instituicoes").chosen({search_contains: true});
});
</script>

<?php
require_once(dirname(__FILE__) . '/../../../application/config/constantes.php');
require_once(INCLUDE_PATH . "Auth.php");
require_once(INCLUDE_PATH . "DB.php");
require_once(CONTROLLER_PATH . "ControleUnidade.php");

$action = isset($_REQUEST['acao']) ? anti_injection($_REQUEST['acao']) : '';

switch ($action) {

    case 'cadastrar':
    case 'alterar':

        $controller = ControleUnidade::getInstance();
        exit($controller->add_update());

    case 'deletar':

```

```

        $controller = ControleUnidade::getInstance();
        exit($controller->delete());

    case 'getTableData':
        $controller = ControleUnidade::getInstance();
        exit($controller->getTableData());

    default:
        require_once(INCLUDE_PATH . 'TemplateHandler.php');
        $template = new TemplateHandler();
        $template->set_css_files(array(
        ));

        $template->set_js_files(array(
        ));
        $template->set_content_file('organizacao/unidades/unidades.php');
        $template->set_authenticated_only(true);
        $template->set_module(ID_MODULO_SISTEMA);
        $template->set_sub_module(ID_SUBMODULO_ORGANIZACAO);
        $template->set_sub_module_item(ID_SUBMODULO_ORGANIZACAO_UNIDADES);
        $template->show();
    }

<?php
require_once(INCLUDE_PATH . "Auth.php");
$auth = new Auth();
if (!$auth->check(ID_SUBMODULO_ORGANIZACAO_UNIDADES)) {
    unauthorized();
}

global $somente_leitura;

?>
<fieldset class="rounded">
    <legend>Unidades</legend>

    <div id="showtable">
        <table id="table"></table>
        <div id="pager"></div>
    </div>
    <div id="empty"></div>

    <div id='buttons'>
        <?php if (!$somente_leitura) { ?>
            <input id='cadastrar' type='button' alt='cadastrar' style='margin: 8px
0 8px 8px;' value='Cadastrar Unidade' />
            <input id='delete' disabled='disabled' class='disabled' type='button'
alt='excluir' style='margin: 8px 0 8px 8px;' value='Excluir Selecionados' />
            <?php } ?>
        </div>

    <div id='status'></div>
    <div id='formulario'></div>

</fieldset>

<script type="text/javascript">
$(function() {
    var grid = $('#table');

```

```

var campoProcurar = '<span style="float:right;">Procurar: <input type="search"
id="gridsearch" placeholder="Digite o texto..." results="0" class="gridsearch"/></span>';

grid.jqGrid({
    url: '?acao=getTableData',
    datatype: 'json',
    autowidth: true,
    height: 270,
    colNames: ['Sigla', 'Nome', 'Instituição', 'Descrição' <?php echo (!$samente_leitura
? ",'" : "" );?>],
    colModel: [
        {name: 'sigla_unidade', index: 'sigla_unidade', resizable: false,
width: '15%', align: 'center'},
        {name: 'nome_unidade', index: 'nome_unidade', resizable: false,
width: '30%', align: 'center'},
        {name: 'sigla_instituicao', index: 'sigla_instituicao', resizable: false,
width: '20%', align: 'center'},
        {name: 'descricao_unidade', index: 'descricao_unidade', resizable: false,
width: '35%', align: 'center'}
        <?php echo (!$samente_leitura ? ",{name:'edit', index:'edit', resizable:false
,sortable:false,width:'3%',align:'center'}" : "");?>
    ],
    rowNum: 10,
    rowList: [10,30,100],
    scrollOffset: 17,
    gridview: true,
    pager: '#pager',
    sortname: 'sigla_unidade',
    sortOrder: "asc",
    toolbar: [false,'top'],
    viewrecords: true,
    shrinkToFit: true,
    hoverrows: false,
    beforeSelectRow: function(rowId, e){
        if(e.target.type == "checkbox"){
            verificaCheckbox($('#table'),$('#delete'));
            return true;
        } else if(e.target.className == "edit") {
            $('#showtable, #buttons, #status').hide();
            $('#formulario').load("formUnidade.php?acao=alterar&id="+rowId).show();
            //displayColorbox("formUnidade.php?acao=alterar&id="+rowId, "475px");
            return false;
        }
    },
    onSelectAll: function(){
        verificaCheckbox($('#table'),$('#delete'));
    },
    loadComplete: function(response){
        processLoadComplete(response);
        verificaCheckbox($('#table'),$('#delete'));
    },
    <?php if (!$samente_leitura) { echo 'multiselect: true,' ;} ?>
    search : true,
    search : {
        caption: "Procurar..."
    },
    caption: campoProcurar,
    loadError : function(xhr,st,err) {

```

```

        showError($('#status'));
    }
});

$('.ui-jqgrid-titlebar-close').remove();

grid.jqGrid('navGrid','#pager',
    {add:false, edit:false, view:false, del:false, search: false, refresh: true}
);

$('#gridsearch').keyup(function() {
    grid.setGridParam({url:'?acao=getTableData&search='+$("#gridsearch").val()});
    grid.trigger('reloadGrid');
});

$("#cadastrar").click(function(){
    $('#showtable, #buttons, #status').hide();
    $('#formulario').load("formUnidade.php?acao=cadastrar").show();
    //displayColorbox("formUnidade.php?acao=cadastrar", "475px");
});

$("#delete").click(function(){
    deleteFromGrid();
});

});
</script>

<?php require_once(dirname(__FILE__) . '/../application/config/constantes.php'); ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
<div style="text-align: center;">
    
    <h1>Ocorreu um erro fatal.</h1>
    <em>Favor entrar em contato com o administrador do sistema.</em>
</div>
</body>
</html>

<?php
/* Verifica se o usuário está logado */
require_once(dirname(__FILE__) . '/../application/config/constantes.php');
require_once(APPLICATION_PATH . "/include/Auth.php");

$auth = new Auth();
$DB = DB::getInstance();

if (!$auth->check()) {
    require_once(APPLICATION_PATH . '/include/TemplateHandler.php');
    $template = new TemplateHandler();
    $template->set_css_files(array(
        "public.css"
    ));
}

```

```

        $template->set_js_files(array(
        ));
        $template->set_content_file("login.php");
        $template->set_authenticated_only(false);
        $template->set_title(TITULO_PADRAO . " - Login");
        $template->set_body_file("body_simple.php");
        $template->show();
    }
    else {
        /* Redireciona para o módulo que o usuário possui acesso */
        $id = $_SESSION["id_usuario"];
        $result = $DB->execPreparedSQL("SELECT pasta_modulo, pasta_submodulo
FROM
usuarios_privilegios
sistema_modulos ON modulo_privilegio = id_modulo
sistema_submodulos ON submodulo_privilegio = id_submodulo
WHERE
fk_usuarios = ? ORDER BY submodulo_privilegio", array("i", $DB->escape($id)));
        if (!empty($result)){
            @header("Location:
".$result[0]['pasta_modulo'].$result[0]['pasta_submodulo']);
        } else {
            $auth->logout();
            @header("Location: index.php");
        }
    }
}
?>

<?php
/**
 * Arquivo responsável pelo controle de sessões: Login e Logout do sistema
 */

/* Autenticação */
require_once(dirname(__FILE__) . '/../application/config/constantes.php');
require_once(INCLUDE_PATH . "/Auth.php");

$acao = isset($_GET["acao"]) ? $_GET["acao"] : "";

switch ($acao) {

    case "login":

        /* Evitando warning */
        ob_start();

        /* Não gravar em cache */
        $gmtDate = gmdate("D, d M Y H:i:s");
        header("Expires: {$gmtDate} GMT");
        header("Last-Modified: {$gmtDate} GMT");
        header("Cache-Control: no-cache, must-revalidate");
        header("Pragma: no-cache");
        header("Content-Type: text/html; charset=utf-8");

        $auth = new Auth();

```

```

        $_REQUEST['vImageCodP'] = isset($_REQUEST['vImageCodP']) ? $_REQUEST['vImageCodP']
: "";
        echo $auth->login($_REQUEST["login_usuario"], $_REQUEST["login_senha"],
$_REQUEST["chk"], $_REQUEST['vImageCodP']);

        /* Limpa o buffer */
        ob_end_flush();
        exit();

        break;

    case "logout":

        $auth = new Auth();
        $auth->logout();

        header("Location: " . URI_RAIZ);
        exit();

        break;

    case "imgVerificacao":

        require_once(INCLUDE_PATH . "/vImage.php");
        $vImage = new vImage();

        $random_num = rand();

        echo "Copie o texto abaixo:<br />";
        echo "<img
src='../application/include/vImageShow.php?size=6&rand=$random_num'>";
        echo "<br />";

        $vImage->showCodBox(1,"class='inputbox' style='width: 88px; margin-top:
5px'");

        exit();
}

require_once(dirname(__FILE__) . "/login_form.php");
?>

<div id="Login_wrapper">
<!-- <h3>Preencha os campos abaixo para entrar</h3-->
<br />
<?php
/* Código de verificação para dificultar roubo de sessão */
$codigo_verificacao = md5($_SERVER['HTTP_USER_AGENT'] . "%cH3Ck#C0d3%");
?>
<form id="Login" name="Login" action="<?php echo URI_RAIZ .
"login.php?acao=login&chk=$codigo_verificacao"; ?>">
<p><label>Usuário</label><br/>
<input type="text" id="Login_usuario" style='width:270px' name="Login_usuario"
class="title">
</p>
<p>
<label>Senha</label><br/>
<input type="password" id="Login_senha" style='width:270px' name="Login_senha"
class="title">

```

```

    </p>
    <div id="excedeu_tentativas" class="formfield"></div>
    <input type="submit" id="Login_submit" style='float:right;' name="Login_submit"
class="buttonbox" value="Login">
</form>
    <div class='clear'></div>
    <div id="status_login" style='width: 230px;'></div>
</div>

<script type="text/javascript">
$(document).ready(function() {
    var options = {
        dataType: 'json',
        beforeSend: function() {
            $('#status_login').removeClass();
            $('#status_login').html("");
            $('#status_login').html('');
        },
        success: function(response, status) {
            if (response.error == "tentativas"){
                $('#excedeu_tentativas').load('login.php?acao=imgVerificacao');
                $('#status_login').addClass(response.status).html(response.msg);
            } else if (response.status == "success"){
                $('#status_login').addClass(response.status).html(response.msg);
                window.location = "index.php";
            } else {
                $('#status_login').addClass(response.status).html(response.msg);
            }
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            $('#status_login').addClass('error').html("Ocorreu um erro inesperado. Entre
em contato com o administrador do sistema.<br/>");
        },
        dataType: 'json',
        type: 'POST'
    };

    $('#login').ajaxForm(options); //Bind the form to the AJAX Form plugin
    $('#login_usuario').focus();
});

</script>

<?php
require_once(dirname(__FILE__) . '/../application/include/DB.php');

$ordem = isset($_REQUEST['modulo']) ? $_REQUEST['modulo'] : array();

if (!$ordem) {
    trigger_error('Módulos não especificados');
}

if (!isset($_SESSION)) {
    session_start();
}

if (!isset($_SESSION['id_usuario'])) {
    trigger_error('Você não está logado');
}

```



```

}

$DB = DB::getInstance();
foreach ($ordem as $posicao=>$ordem_submodulo) {
    $DB->execPreparedSQL("UPDATE usuarios_privilegios SET posicao_privilegio = ? WHERE
modulo_privilegio = ? AND fk_usuarios = ?",
                        array('iii', $DB->escape($posicao+1),
$DB->escape($ordem_submodulo), $DB->escape($_SESSION['id_usuario'])), true);
}

<?php

require_once(dirname(__FILE__) . '/../application/include/DB.php');

$submodulo = isset($_REQUEST['submodulo']) ? $_REQUEST['submodulo'] : null;
$status = isset($_REQUEST['status']) ? $_REQUEST['status'] : null;

if ($status != 0 && $status != 1) {
    exit('Status de visibilidade inválido');
}

if (!isset($_SESSION)) {
    session_start();
}

if (!isset($_SESSION['id_usuario'])) {
    exit('Você não está logado');
}

$DB = DB::getInstance();
$DB->execPreparedSQL("UPDATE usuarios_privilegios SET visibilidade_privilegio = ? WHERE
modulo_privilegio = ? AND fk_usuarios = ?",
                    array('iii', $DB->escape($status), $DB-
>escape($submodulo), $DB->escape($_SESSION['id_usuario'])), true);

#####FIM DO CÓDIGO#####

```

Desenvolvimento e estudo de uma ferramenta para gerência de SLA aplicado na Rede Nacional de Ensino e Pesquisa

Rodrigo Pescador, Paulo Brandtner

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

Florianópolis - SC - Brasil

{pescador,paulo.brandtner}@inf.ufsc.br

***Abstract.** Service Level Agreement is a contract between two parts: the service supplier and the customer. This contract establishes terms and ways of measuring and its quality observed. A SLA may be about performance, tariffing and billing, service delivery and compensations. Once many companies outsource their services, SLA is an important instrument when hiring a service to keep high levels of service quality. This paper presents a proposal of different ways to measure and collect measurement data to create SLA reports. Additionally, a SLA software implementation is designed to support the operational needs of the National Research and Education Network in Brazil.*

***Resumo.** O Acordo de Nível de Serviço é um contrato entre duas partes: o fornecedor do serviço e o cliente. Ele estabelece termos e formas de medição da qualidade observada do produto. Um SLA pode ser sobre o desempenho, forma de entrega do serviço e suas compensações. Uma vez que muitas empresas terceirizam seus serviços, a utilização de um instrumento de SLA é importante ao contratar um serviço a fim de manter altos níveis de qualidade do que é oferecido. Este trabalho apresenta uma proposta de diferentes formas de medição e coleta de dados para criar relatórios de SLA. Adicionalmente, a implementação de software SLA é projetado para suportar as necessidades operacionais da Rede Nacional de Ensino e Pesquisa.*

1. Introdução

O aumento da utilização das redes de dados como base fundamental para o funcionamento e operação dos negócios das empresas traz a necessidade de que este sistema seja tolerante a falhas, ou que pelo menos estas falhas sejam resolvidas em um curto período de tempo de modo a evitar paralisações que possam causar algum transtorno para a empresa, seja ele qual for.

A grande preocupação das empresas prestadoras de serviço de telecomunicações hoje em dia é a "qualidade percebida pelo cliente". E a estratégia adotada pelos provedores de telecomunicações para atuar com sucesso sobre este tema é a obtenção do acordo de Nível de Serviço (SLA - Service Level Agreement) entre clientes e provedores. (PIMENTEL, 2008). Segundo recomendação E.860 da ITU-T, o SLA é o resultado de uma negociação entre duas ou mais partes, no caso o cliente e o provedor do serviço, que tem como objetivo alcançar um

entendimento comum em relação a entrega do serviço: seus níveis de qualidade, responsabilidades, prioridades, etc. O principal objetivo do SLA é prover informações do serviço prestado de forma objetiva e quantificável.

Em uma primeira etapa, o artigo apresenta o estudo de formas de medição e coleta de dados para confecção de relatórios de SLA voltados à telecomunicações, especialmente focados nas necessidades da Rede Nacional de Ensino e Pesquisa. Por fim, é apresentada uma ferramenta desenvolvida para o gerenciamento e controle dos acordos SLA.

2. Gerenciamento de redes

As redes de computadores e seus recursos associados possuem tamanha importância para uma organização que estes componentes praticamente não podem parar. Em outras palavras, isso significa que as falhas e degradação dos serviços devem ser cada vez menores. STALLINGS (1998 apud LOPES, 2003, p. 17) diz que o objetivo da gerência de redes é monitorar e controlar os elementos da rede (físicos e lógicos), de forma a garantir determinado nível de qualidade do serviço. Para isto, os gerentes e administradores de rede são auxiliados por sistemas de gerenciamento de redes.

Um sistema de gerenciamento de rede pode ser definido como um conjunto de ferramentas de monitoração e controle dos dispositivos da rede, onde existe uma interface com informações sobre o estado da rede.

2.1 SNMP

O protocolo SNMP é um protocolo de gerência definido a nível de aplicação e é utilizado para obter informações de servidores SNMP - agentes espalhados em uma rede. Os dados são obtidos através de requisições de um gerente a um ou mais agentes utilizando os serviços do protocolo de transporte UDP para enviar e receber suas mensagens através da rede. Dentre as variáveis que podem ser requisitadas estão a MIB II ou MIB privada de cada fabricante.

O gerenciamento da rede através do SNMP permite o acompanhamento simples e fácil do estado, em tempo real, da rede, podendo ser utilizado para gerenciar diferentes tipos de sistemas. (DIAS & JUNIOR). Devido a sua ampla implementação por diversos fabricantes, o gerenciamento da rede através deste protocolo se torna bastante interessante e escalável à medida que novos componenetes surgem na rede. Através dele também é possível coletar diversos dados e medições para criação de relatórios de qualidade da rede, servindo como base para o documento de SLA.

É importante destacar que o protocolo SNMP possui dois componentes básicos: o **agente** (processo responsável pela resposta das informações quando solicitadas pelo gerente - mantém uma base de informações local) e o **gerente** (software executado em uma estação que realiza as solicitações de informações aos agentes, geração de relatórios e alertas).

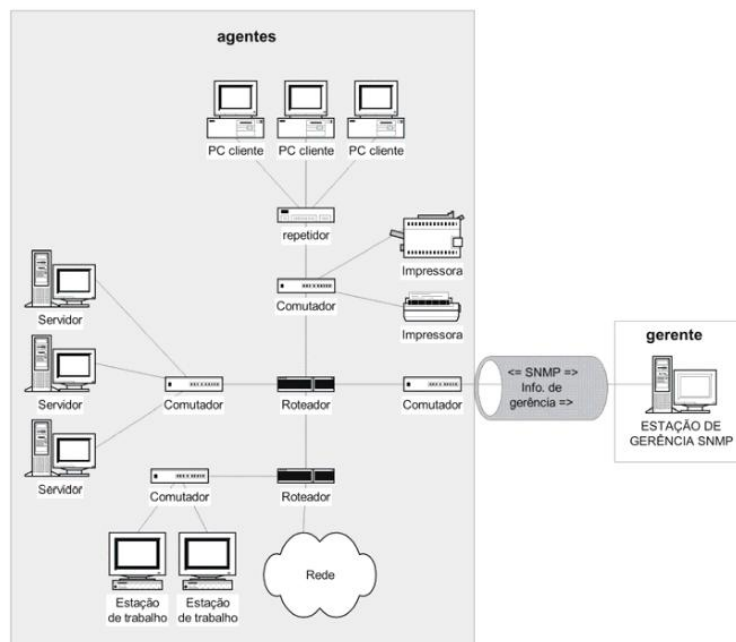


Figura 18 - Arquitetura geral de uma solução de gerência (LOPES 2003)

3. Gerenciamento de acordos de nível de serviço

A crescente terceirização no mercado de telecomunicações vem alterando a relação entre seus compradores e fornecedores de serviços. Um dos pontos importantes nesta mudança é a aplicação de SLAs em contratos. Desta forma, as empresas possuem garantias, que devem ser cumpridas por seus fornecedores, de níveis de qualidade e prazos.

Segundo a recomendação E.860 da ITU-T (2002), o SLA é um acordo estabelecido entre o provedor e seu cliente para definir formalmente o nível de qualidade que deve ser atingido na prestação de um serviço. O objetivo é tornar a medição da qualidade objetiva e quantificável, introduzindo meios para mensurar a expectativa do cliente e, portanto, para gerir a qualidade percebida por ele.

A aplicação do SLA ainda precisa se desenvolver, uma vez que a maioria das companhias que adotam algum tipo de SLA o fazem de forma reativa, apresentando métricas somente de disponibilidade e desempenho. Isto pode ser explicado pela falta de ferramentas automatizadas que permitam fazer uma gestão efetiva dos acordos de nível de serviço.

3.1 SLM

O SLM (*Service Level Management*) é o conjunto de processos e procedimentos que visam garantir o cumprimento dos SLAs. É responsável pelo planejamento e coordenação dos acordos. Além disso, deve garantir a qualidade requerida para os serviços através do acompanhamento dos relatórios SLA e cruzamento dos parâmetros estabelecidos em contrato.

O instrumento de controle do SLM é o Acordo de Nível de Serviço (SLA), um contrato celebrado entre a empresa ou setor de TI e seus clientes, que define parâmetros de qualidade do serviço, como por exemplo capacidade do sistema, o desempenho da rede,

tempo de resposta global necessário para atender aos objetivos empresariais. Além disso, descreve a compensação devida ao cliente se os objetivos e regras estabelecidas não forem cumpridos. (LEÃO, 2002).

A partir da detecção de uma violação, descobre-se qual foi o elemento que causou o problema, como uma linha de comunicação, uma aplicação ou um banco de dados muito carregado. A análise fim-a-fim leva a um isolamento rápido das causas dos rompimentos de SLA para que o ambiente possa ser avaliado e dimensionado para atender as demandas de seus usuários em relação aos serviços oferecidos. (PEREIRA, 2004).

O gerenciamento de nível de serviço é uma questão de sobrevivência. Segundo STURM & MORRIS & JANDER (2001), é raro encontrar uma empresa bem sucedida que venda seus produtos na forma do "pegar ou largar". Em vez disso, as empresas se esforçam e investem para definir claramente as capacidades e limitações do produto que estão vendendo.

Em suma, o SLM serve principalmente para assegurar o cumprimento do SLA de forma a prevenir e reduzir os impactos negativos na qualidade dos serviços, manter e melhorar a qualidade dos serviços contratados/oferecidos e avaliar as mudanças de estratégias.

3.2 SLA

Quando uma organização contrata serviços de um provedor, é necessário estabelecer algumas garantias em relação a forma de entrega do serviço e sua qualidade associada, tanto para o contratante como para o contratado. Estas regras são fornecidas através de um contrato chamado de Contrato de Nível de Serviço - SLA (*Service Level Agreement*). O SLA é o ponto de referência para o controle da qualidade a ser fornecida ou recebida por uma empresa (SILVA, 2005).

Ele é descrito como uma declaração de expectativas e obrigações que existem no relacionamento de negócio entre duas organizações e seu consumidor. Esta declaração, muitas vezes chamada de contrato, especifica os níveis de qualidade de serviço que o fornecedor se compromete em disponibilizar, além de cláusulas legais e quais as consequências para cada parte se houver descumprimento de deveres. (RIGHI et al., 2006).

O SLA fornece uma linguagem comum de entendimento entre as duas partes envolvidas: o prestador de serviço e o cliente. A documentação de um entendimento mútuo resultante da negociação do acordo de nível de serviço deve trazer clareza aos negócios acordados. Para monitorar o cumprimento desse negócio é necessário que as informações sobre um determinado serviço sejam coletadas e tratadas, gerando-se relatórios dos níveis de serviços, os quais possuem valores reais dos parâmetros em um determinado período (hora, dia, mês, ano, etc.). (PEREIRA, 2004).

4. Proposta do sistema

Apresentaremos o ambiente para gerência de SLA que foi proposto e desenvolvido baseado nos estudos realizados ao longo do trabalho. A **Figura 19** mostra a estrutura de coleta de medições da rede nos equipamentos remotos.

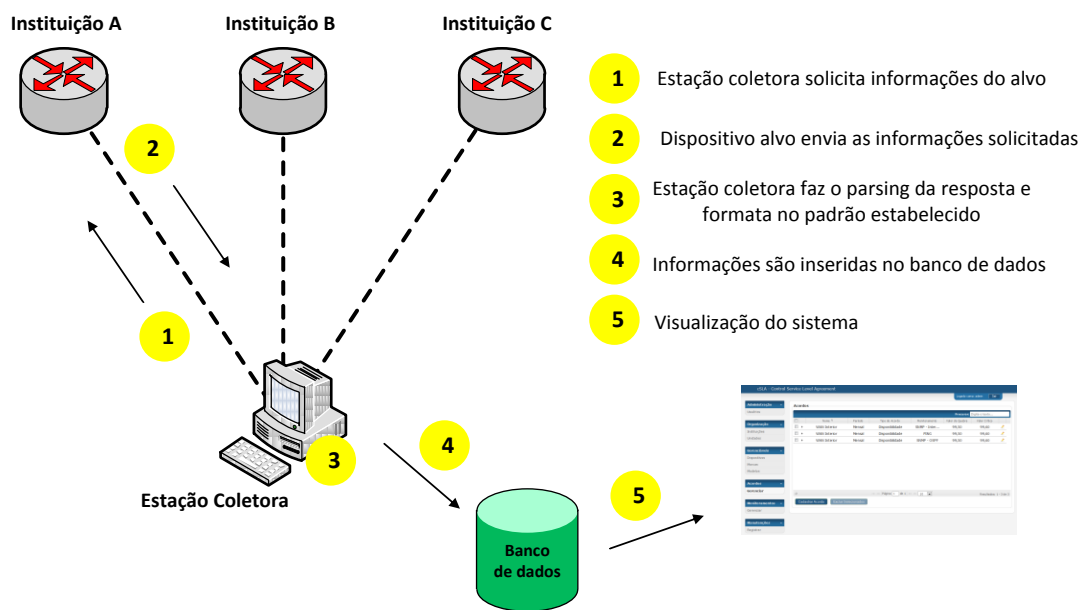


Figura 19 - Coleta das medições do sistema de controle de SLA

No ambiente desenvolvido existe a máquina gerente (estação central) responsável pela coleta/obtenção dos dados das medições realizadas. Tudo é processado por esta máquina, sendo ela o núcleo do sistema. O funcionamento do ambiente se dá basicamente da seguinte forma:

1. A estação gerente dispara a coleta das informações contra um determinado alvo;
2. O dispositivo alvo responde, ou não, a esta requisição com as informações solicitadas;
3. A informação recebida é tratada e é feito o *parsing* da mensagem para se adequar ao modelo de dados do sistema;
4. Uma vez que os dados estão padronizados, eles são inseridos no banco de dados para posterior recuperação e geração de relatórios;

Os dados são obtidos em intervalos regulares de um minuto por um agente que percorre todos os dispositivos alvo cadastrados nos acordos requisitando seus estados e demais informações necessárias para o cálculo das métricas. Vale destacar que as medições são feitas de forma paralela e independente, evitando, assim, a interferência de um processo em outro. Caso qualquer processo falhe ou trave, os demais não serão afetados e o programa continua a ser executando seu ciclo normal. Além disso, é possível cadastrar os dispositivos alvo para que sejam alcançáveis através de IPv6.

4.1 Métricas suportadas

Em um SLA voltado a rede de dados é possível estabelecer diversas métricas para verificação da qualidade do serviço prestado. O dia a dia na operação e administração de redes nos mostra alguns pontos importantes que devem ser observados prioritariamente neste processo. Desta forma, as métricas implementadas para utilização no ambiente são detalhadas abaixo.

Disponibilidade

Este indicador descreve o percentual de tempo que para um dado período de um acordo, o enlace, incluindo o equipamento alvo, ficou disponível e em operação.

Perda de pacotes

Esta métrica indica a relação entre pacotes enviados e recebidos para um determinado dispositivo alvo e é expressada em porcentagem.

Latência

A latência, também chamada de RTT (*round trip time*), é o somatório do tempo de ida e volta de um pacote de um ponto A para um ponto B. Caso um ponto A não envie um pacote ou o ponto B não devolva uma resposta, o valor da medição é considerado desconhecido.

Jitter

O jitter é a variação do atraso da entrega de pacotes sucessivos na rede entre um ponto A e um ponto B. Para medir o jitter, basta obter a soma da diferença de tempo entre pacotes seguidos em uma rede e dividir pelo total de amostras - 1.

Banda utilizada

Representa a quantidade de dados passantes por um enlace em um determinado tempo. Caso um valor definido seja atingido e se mantenha por um certo período, existe a quebra do acordo. A unidade de medida típica é bits.

Erros de interface

Representa o número de pacotes recebidos com erros em uma interface e que não poderão ser repassados às camadas superiores.

4.2 Formas de medição e coleta

Para a obtenção dos dados, seja qual for a métrica, utilizamos basicamente a ferramenta *fping* e o protocolo SNMP. DZUBIN (2007), mantenedor atual do aplicativo, explica que o *fping* é um programa que utiliza Internet Control Message Protocol para determinar se um dispositivo alvo está disponível, trazendo algumas facilidades quando utilizado com scripts, pois sua saída é facilmente parseada. As funcionalidade conhecidas na implementação tradicional no *ping* estão disponíveis também no *fping*.

A Tabela 2 apresenta as métricas disponíveis e suas formas de coleta dos dados.

| Métrica | Formas de coleta dos dados |
|--------------------|---|
| Disponibilidade | Ping (fping), snmp interface, snmp ospf |
| Perda de pacotes | ICMP (fping) |
| Latência | ICMP (fping) |
| Jitter | ICMP (fping) |
| Banda utilizada | snmp interface |
| Erros de interface | snmp interface |

Tabela 4 - Métricas vs. formas de coleta dos dados

Para a métrica de disponibilidade, conforme a Tabela 2 mostra, é possível realizar as consultas dos dados de três maneiras diferentes: ping, snmp interface e snmp ospf. Já para métricas de perda de pacotes, latência e jitter está disponível somente a coleta através de ICMP. Por último, para coleta de banda utilizada e erros de interface somente o SNMP está disponível.

No caso das métricas que utilizam SNMP, os OID's necessários estão relacionados na Tabela 3.

| | |
|----------------------------------|-------------------------|
| SNMP Interface (ifOperStatus) | 1.3.6.1.2.1.2.2.1.8 |
| SNMP OSPF (ospfNbrState) | 1.3.6.1.2.1.14.10.1.6 |
| Banda utilizada (ifHCInOctets) | 1.3.6.1.2.1.31.1.1.1.6 |
| Banda utilizada (ifHCOutOctets) | 1.3.6.1.2.1.31.1.1.1.10 |
| Erros de interface (ifInErrors) | 1.3.6.1.2.1.2.2.1.14 |
| Erros de interface (ifOutErrors) | 1.3.6.1.2.1.2.2.1.20 |

Tabela 5 - OID's utilizados pelo sistema

4.3 Algumas funcionalidades do sistema

Geração de relatórios

Este módulo permite definir parâmetros como tipo de acordo, tipo de monitoramento, unidade, métrica ou um conjunto deles.

O usuário poderá selecionar o intervalo de tempo para o qual deseja gerar o documento e esse contemplará as informações referentes aos eventos ocorridos no período, como quedas, abertura de chamados e outras anotações relevantes.

Uma das ideias propostas pelo módulo é gerar um documento basicamente pronto no formato DOC, permitindo que o usuário utilize ferramentas de edição de arquivos de texto e faça as alterações necessárias e observações pertinentes.

Para isso, serão utilizados templates que auxiliarão na automatização e confecção do relatório dos acordos definidos, que muitas vezes são periodicamente solicitados pelo cliente.

Manutenções programadas

Observa-se geralmente nas unidades, a necessidade da realização de manutenções preventivas em alguns de seus dispositivos, sendo que esses podem fazer parte de acordos que estão sendo monitorados pelo sistema.

Essas paradas, chamadas de manutenções programadas, não podem e não devem afetar os valores coletados pelo software para definir uma possível quebra de acordo, portanto devem ser desconsideradas no momento da geração do relatório.

Tendo em vista essa necessidade, o aplicativo permite cadastrar essas paradas, informando o motivo e período no qual o dispositivo pode ficar indisponível e esse período será descontado do cálculo total do SLA.

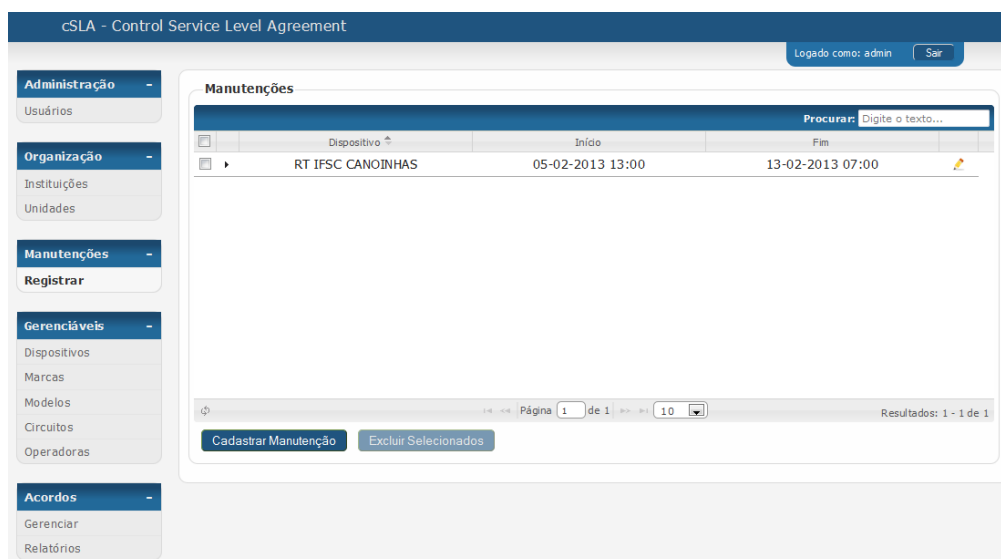


Figura 20 - Tela de cadastro de manutenções programadas

5. Medições

Apesar de a interface de visualização dos resultados das medições necessitar alguns ajustes de *layout*, a parte funcional, os resultados propriamente ditos, podem ser verificados.

Nas **Figura 21** e **Figura 22** podemos observar um esboço do resultado obtido por dois exemplos de monitoramentos, um com uma queda registrada e outro com uma interface apresentando tempo de disponibilidade de 100%, respectivamente.

| |
|---|
| Período: 15-01-2013 00:00:00 a 20-02-2013 13:00:00 |
| DISPOSITIVO: M10i |
| PORTA: lsq-1/3/0.5 |
| UPTIME: 876:07:00 ==> 99.79% |
| DOWNTIME: 01:49:00 |
| INDEFINIDO: 00:04:00 |

| |
|--|
| QUEDAS |
| 16-02-2013 19:09:02 -> 16-02-2013 20:58:02 |

Figura 21 - Exemplo de resultado obtido por um monitoramento com evento de queda.

| |
|---|
| Período: 15-01-2013 00:00:00 a 20-02-2013 13:00:00 |
| DISPOSITIVO: M10i |
| PORTA: ge-0/0/1 |
| UPTIME: 877:58:00 ==> 100.00% |
| INDEFINIDO: 00:02:00 |

Figura 22 - Exemplo de resultado obtido por um monitoramento com disponibilidade de 100%.

Uma vez que a coleta de dados é feita de minuto em minuto, existe o que chamamos de tempo de incerteza, período este que não é possível avaliar com precisão qual o estado real do equipamento/interface. Se houve mudança de estado entre uma medição e outra, é contabilizado um minuto de tempo de incerteza, que para efeitos de cálculos é considerado como disponível.

Esses resultados, juntamente com as informações registradas nos sistema de eventos, somadas aos gráficos, também em fase de finalização, compõem os relatórios que são gerados pelo software para verificação de quebra de acordos e a manutenção da qualidade do nível de serviço pela operadora, podendo servir para o cliente acompanhar de forma mais organizada se o acordo está sendo cumprido e se há penalidades ou multas a serem pagas por parte da prestadora do serviço.

6. Conclusão e trabalhos futuros

Visando garantir a qualidade dos serviços prestados através de um controle eficiente dos acordos em um SLA, é necessária a utilização de uma ferramenta integrada que meça e gere

os respectivos indicadores em forma de relatórios. A ferramenta desenvolvida apresenta ao administrador ou gestor as informações que permitem a aplicação do SLA, porém cabe a organização definir de que forma estes dados serão usados para a manutenção de um bom nível de prestação dos serviços (seja por multas ou por descontos em serviços).

O sistema é de grande auxílio para o acompanhamento dos acordos estabelecidos, mas não basta inserir um valor fictício de nível de serviço e esperar que ele seja cumprido. O SLA é a negociação entre duas ou mais partes que devem chegar a um entendimento comum em relação a qualidade da entrega do serviço, responsabilidades, etc. Sendo assim, deve haver um esforço de ambos os lados para que os objetivos sejam alcançados.

É possível dizer que a ferramenta desenvolvida irá trazer, principalmente, uma grande economia de tempo na confecção dos relatórios, uma vez que seu processo é praticamente todo automatizado através da geração de templates editáveis. A principal arma de um SLA é um relatório bem elaborado e detalhado sobre os eventos e acontecimentos.

O software possui módulos a serem melhorados e outros que precisam ser desenvolvidos. Uma funcionalidade interessante que será implementada posteriormente é a possibilidade de realizar um cálculo automático do valor da multa (R\$) a ser paga no caso da quebra de um acordo. Também está na lista de próximas funcionalidades a geração de alertas automáticos para informar aos operadores se determinado acordo está prestes a ser rompido. Além disso, outras formas de coleta de dados e novas métricas serão estudadas para possível implementação.

7. Referências

- PIMENTEL, Adda Regina de Oliveira; QUINTELLA, Heitor Luiz Murat de Meirelles. A importância do SLA no setor de telecomunicações. *Rev. Eletrônica Produção & Engenharia*. v. 1, n. 1, p. 1-12, set./dez. 2008.
- ITU-T – Telecommunication standardization sector of ITU. Recomendação E.860. [S.l.]: Framework of a Service Level Agreement, 2002.
- LOPES, Raquel V. et al. *Melhores Práticas para Gerência de Redes de Computadores*. Rio de Janeiro: Campus, 2003.
- DIAS, Beethovem Z.; JUNIOR, Nilton A. Protocolo de Gerenciamento SNMP. Nota técnica CBPF-NT-006/01.
- LEÃO, Maximilliam M. Um estudo sobre a concepção e desenvolvimento do acordo de nível de serviço para ambientes de redes de computadores. Florianópolis, 2002. 127 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.
- PEREIRA, Mateus C. Análise de desempenho em redes Wireless ad-hoc e estabelecimento de um acordo de nível de serviço pró-ativo. 2004. 130 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.
- STURM, Rick; MORRIS, Wayne; JANDER, Mary. *Service Level Management*. Campus, 2001. 272p.

- SILVA, Marco Antônio da. Um SLA para VoIP e seu mapeamento em uma rede DiffServ/MPLS. Florianópolis, 2005. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.
- RIGHI, Rafael R.;PELLISSARI, Felipe R.; WESTPHALL, Carlos B. Especificação e validação de métricas para acordos de níveis de serviço orientados à segurança. Artigo – Universidade Federal de Santa Catarina. Florianópolis, SC, 2004.
- DZUBIN, Thomas. 2007. Disponível em <<http://fping.sourceforge.net>>. Acessado em 10/12/2012.