

**Vitor Rozsa**

***Envio Automático de SMS - Módulo de Extensão  
para o Moodle***

São José – SC

Julho / 2012

**Vitor Rozsa**

***Envio Automático de SMS - Módulo de Extensão  
para o Moodle***

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientadora:

Prof<sup>a</sup>. Claudinice Carla Bertotti

Co-orientador:

Prof. Ederson Torresini

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES  
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC

Julho / 2012

Monografia sob o título “*Envio Automático de SMS - Módulo de Extensão para o Moodle*”, defendida por Vitor Rozsa e aprovada em 12 de julho de 2012, em São José, Santa Catarina, pela banca examinadora assim constituída:

---

Prof<sup>a</sup>. Claudinice Carla Bertotti, Tecn. Telecom  
Orientadora

---

Prof. Ederson Torresini, M.Sc.  
Co-orientador

---

Prof. Marcos Moeck, Dr.  
IFSC

---

Prof. Rogério Santos Pedroso, M.Sc.  
Unifebe

A maior ilusão é viver apenas de ideias enquanto não temos apenas a mente a tratar, mas também o mundo físico, as pessoas e as situações para viver. Para ser.

*Vitor Rozsa.*

# *Agradecimentos*

Primeiro, gostaria de agradecer aos professores do IF-SC pela dedicação que tem com as aulas e para com os alunos. Sem eles eu não teria chegado até aqui.

Em segundo, devo agradecer ao meu irmão Rubens Rozsa Neto, que prestou grande auxílio nas etapas finais deste trabalho.

E por último e mais importante, quero agradecer aos meus orientadores, que desempenharam maravilhosamente seus papéis. Seus esforços praticamente ilimitados permitiram existir as coisas boas neste trabalho.

# *Resumo*

Apesar do caráter quase ubíquo da Internet, há casos em que sua utilização não é possível. Seja por uma queda na rede local de dados, seja por má qualidade do sinal de uma Internet sem fio. Mediante essa possibilidade de privação, este trabalho pretende permitir que a utilização de serviços em fóruns esteja disponível através de uma solução em forma de módulo de extensão (*plug-in*), utilizando o Serviço de Mensagens Curtas como forma alternativa para comunicação.

A implementação do *plug-in*, de forma extensível, é direcionada ao Sistema de Gestão da Aprendizagem (SGA) Moodle, podendo ser utilizado em outros fóruns e aplicações mediante adaptações necessárias às funcionalidades de cada sistema.

Além desta disponibilidade, outros serviços relacionados ao uso do Moodle foram implementados, como “Agendamento do Envio de SMS” e “Envio de SMS para Múltiplos Destinos”.

**Palavras chave:** Módulo de extensão, sistema de gestão da aprendizagem, SMS, controle de serviços, disponibilidade.

# *Abstract*

Despite of the almost ubiquitous feature of the Internet, there is some cases that his uses isn't possible. Whether by a drop in the local data network or by poor signal quality of a wireless Internet. Through this possibility of privation, this work intends to allow that the use of services on forums are available through a solution in form of a plug-in, using the *Short Message Service* (SMS) as an alternative way for communication.

The implementation of the plug-in, so extensible, is directed to the *Learning Environment System* (LMS) Moodle, can be used in other applications and forums through adjustments required by the features of each system.

In addition to this availability, other services related to the use of Moodle have been implemented, such as "Schedule SMS" and "SMS to multiple destinations".

**Keywords:** Plug-in, Learning Environment System, SMS, services, availability.

# *Sumário*

## **Lista de Figuras**

## **Lista de Tabelas**

<b>1</b>	<b>Introdução</b>	p. 12
1.1	Objetivos	p. 13
1.1.1	Objetivos Gerais	p. 13
1.1.2	Objetivos Específicos	p. 13
1.2	Organização do texto	p. 14
<b>2</b>	<b>Ferramentas e Tecnologias</b>	p. 15
2.1	Módulo de Extensão	p. 15
2.1.1	Arquitetura	p. 16
2.1.2	Funcionalidades	p. 16
2.2	Blocos para Moodle	p. 17
2.3	Daemon	p. 18
2.3.1	Criação de Daemons	p. 19
2.4	Serviço de Mensagens Curtas	p. 19
2.4.1	Comunicação com a rede GSM	p. 20
2.4.2	Funcionamento	p. 20
2.4.3	Modems GSM	p. 20
2.5	Remote Procedure Call	p. 21
2.5.1	XML-RPC	p. 21



2.6	Sistema Gerenciador de Banco de Dados . . . . .	p. 21
2.6.1	PostgreSQL . . . . .	p. 22
2.7	Linguagens de desenvolvimento . . . . .	p. 22
<b>3</b>	<b>Implementação, Funcionamento e Testes</b>	<b>p. 23</b>
3.1	Funcionamento dos Serviços . . . . .	p. 23
3.1.1	Formato das solicitações . . . . .	p. 23
3.1.2	Solicitação e Execução do Serviço . . . . .	p. 25
3.2	Bloco de controle . . . . .	p. 27
3.2.1	Serviços . . . . .	p. 27
3.2.2	Regras . . . . .	p. 28
3.3	Daemon de serviços . . . . .	p. 28
3.3.1	Banco de Dados . . . . .	p. 29
3.3.2	Serviço de Mensagens Curtas . . . . .	p. 31
3.3.3	Thread-Filha . . . . .	p. 31
3.4	Comunicação entre bloco de controle e daemon de serviços . . . . .	p. 34
3.5	Ferramenta de registros e documentação . . . . .	p. 35
3.6	Características do Plug-in . . . . .	p. 36
3.6.1	Desempenho . . . . .	p. 36
3.6.2	Segurança . . . . .	p. 36
3.6.3	Modularização . . . . .	p. 37
<b>4</b>	<b>Conclusões</b>	<b>p. 39</b>
4.1	Disponibilização do Plug-in . . . . .	p. 39
4.2	Trabalhos Futuros . . . . .	p. 40
	<b>Apêndice A – Manual de Uso do Plug-in</b>	<b>p. 41</b>
A.1	Instalação . . . . .	p. 41

A.1.1	Python e módulos necessários . . . . .	p. 41
A.2	Configuração . . . . .	p. 43
A.2.1	Acrescentar ao espaço de trabalho . . . . .	p. 43
A.2.2	Configuração e inicialização do daemon de serviços . . . . .	p. 45
A.2.3	Banco de dados do daemon . . . . .	p. 45
A.3	Configuração do Serviço de Mensagens Curtas . . . . .	p. 46
A.3.1	Telefone celular oferecendo o Serviço de Mensagens Curtas . . . . .	p. 47
A.4	Uso . . . . .	p. 48
A.4.1	Permissões . . . . .	p. 48
A.4.2	Acesso pelo fórum virtual . . . . .	p. 48
A.5	Requisições por SMS . . . . .	p. 51
A.5.1	Requisição simples . . . . .	p. 52
A.5.2	Requisição com agendamento . . . . .	p. 53
A.5.3	Habilitar/Desabilitar o uso . . . . .	p. 54
A.6	Mensagens do Plug-in . . . . .	p. 54
<b>Lista de Abreviaturas</b>		p. 57
<b>Referências Bibliográficas</b>		p. 58

## *Lista de Figuras*

2.1	Representação em quebra cabeça da arquitetura de <i>plug-ins</i> . . . . .	p. 16
2.2	Tabela comparativa dos blocos para Moodle que possuem funcionalidades relacionadas ao uso de SMS. . . . .	p. 17
3.1	Diagrama de Entidades do <i>plug-in</i> . . . . .	p. 24
3.2	Formato genérico da solicitação de serviços através de SMS. . . . .	p. 24
3.3	Solicitação do serviço através de SMS. . . . .	p. 25
3.4	Validação da origem e busca dos destinos. . . . .	p. 26
3.5	Recuperação e execução do serviço. . . . .	p. 26
3.6	Estrutura lógica do <i>daemon</i> de serviços . . . . .	p. 29
3.7	Execução da <i>thread</i> -filha. . . . .	p. 32
3.8	Pontos de chamadas de <i>scripts</i> externos pelo <i>daemon</i> de serviços. . . . .	p. 34
A.1	Interface de instalação de novos <i>plug-ins</i> . . . . .	p. 42
A.2	Confirmação de instalação bem sucedida. . . . .	p. 42
A.3	Minha página inicial. . . . .	p. 43
A.4	Interface de edição de páginas. . . . .	p. 44
A.5	Interface do <i>plug-in</i> no Moodle. . . . .	p. 44
A.6	Inteface de agendamento/envio de SMS pelo fórum virtual. . . . .	p. 49
A.7	Inteface de relatórios. . . . .	p. 51
A.8	Inteface de registros. . . . .	p. 52
A.9	Acessar interface “Visão geral dos Plugins”. . . . .	p. 55
A.10	Acessar configurações do <i>plug-in</i> . . . . .	p. 55
A.11	Configurações do <i>plug-in</i> no Moodle. . . . .	p. 56

# *Lista de Tabelas*

3.1 Colunas da tabela utilizada pelo *daemon* de serviços. . . . . p. 30

# 1 *Introdução*

A Internet, a partir de sua forma mais primitiva, quando ainda era denominada “ARPA-NET” e “DARPANET”, vem sendo utilizada como meio de comunicação através de *e-mails*, páginas estáticas de texto, listas de discussão, entre outros. Com a popularização do acesso, a Internet possibilitou a comunicação de uma forma rápida, direta e cobrindo grandes distâncias entre os interlocutores; algo antes possível somente através de cartas e da telefonia, que tinham limitações em diversos aspectos. Essa troca constante de informações abriu espaço, em listas de discussão, para que diversos interlocutores participassem ao mesmo tempo, tendo destaque a sintetização do conhecimento, no que atualmente são conhecidos como *fóruns virtuais*.

Os fóruns virtuais são locais de discussões nos quais membros podem publicar tópicos sobre assuntos diversos e conversar com outros membros a partir destes tópicos. O que difere os fóruns entre si são os assuntos aos quais estão relacionados e os serviços que oferecem para facilitar e melhorar a experiência dos usuários. Os temas de discussão são os mais variados, voltados para esportes, religião, tecnologia, dentre outros (vBulletin, 2012). E na educação, os fóruns virtuais estão bastante presentes. São utilizados pelas instituições de ensino para auxiliar na comunicação entre professores, alunos, servidores administrativos e acadêmicos.

Para utilização de fóruns virtuais existem diversos tipos de sistemas digitais, com ferramentas e finalidades diferenciadas. Um sistema que atende às necessidades voltadas ao contexto educacional é o Moodle, conhecido como Sistema de Gestão de Cursos (SGC). Destina-se, principalmente, à criação de ambientes virtuais de aprendizagem voltados às instituições de ensino, oferecendo diversas ferramentas para este fim. Sua atualização é constante, devido à grande comunidade *on-line* que colabora para o desenvolvimento deste em código aberto e livre.

O sistema Moodle fornece uma ferramenta em forma de fórum que tem por objetivo básico a mediação virtual para discussão de assuntos da área acadêmica, mas também possibilita o gerenciamento dos participantes, através do cadastro destes em cursos, grupos e disciplinas. Permite ainda gerenciar e utilizar os tópicos de discussão, calendário, lembretes, mensagens instantâneas, alertas de mensagens por *e-mail*, dentre outros serviços.

Quando uma necessidade específica se apresenta, é possível adicionar funcionalidades ao sistema via módulo de extensão (*plug-in*). Essa possibilidade facilita o desenvolvimento de aplicações, uma vez que o desenvolvedor poderá se dedicar apenas ao estudo das especificações destinadas à integração do seu *plug-in* ao sistema, não necessariamente todos os módulos pré-existent.

Apesar do caráter quase ubíquo da Internet, há casos em que sua utilização não é possível. Seja por uma queda na rede local de dados, seja por má qualidade do sinal de uma Internet sem fio, entre outras situações. Mediante essa possibilidade de privação do uso imediato da Internet, esse trabalho pretende permitir que a utilização dos serviços em fóruns virtuais estejam disponíveis através de outra ferramenta. Para que esta *disponibilidade* de serviços seja real, é apresentada uma solução em forma de *plug-in* para o **Moodle 2.2.1+**, utilizando o Serviço de Mensagens Curtas como forma alternativa no controle dos serviços do sistema.

Da forma como se propõe a implementação do *plug-in*, este se torna extensível, podendo ser utilizado em outros fóruns virtuais e aplicações, mediante adaptações necessárias às funcionalidades de cada sistema. Além da disponibilidade, outros serviços relacionados ao uso do Moodle (dos quais o sistema não dispõe na forma proposta) foram implementados, são esses: “agendamento do envio de SMS” e “Envio de SMS para múltiplos destinos”. Estes serviços também compõe a parte extensível do *plug-in*.

## 1.1 Objetivos

### 1.1.1 Objetivos Gerais

- Desenvolver um *plug-in* que permita a utilização de serviços em fóruns virtuais e outras aplicações através do Serviço de Mensagens Curtas, utilizando como exemplo o Sistema Moodle;

### 1.1.2 Objetivos Específicos

- Implementar o serviço “Agendamento do envio de SMS”, proporcionando uma forma de postergar o envio de informações;
- Implementar o serviço “Envio de SMS para múltiplos destinos”, permitindo a difusão de informações para diversos destinos.

## 1.2 Organização do texto

No Capítulo 2 são descritas tecnologias, técnicas e conceitos necessários para o entedimento da estrutura do *plug-in*. No Capítulo 3, o *plug-in* é explorado e seu funcionamento interno é detalhado, bem como de que forma as entidades que o compõe - bloco de controle e daemon de serviços - foram implementadas, como se comportam e como se relacionam, e o desempenho apresentado. E por fim, no Capítulo 4, são apresentadas observações sobre o funcionamento do projeto.

## 2 *Ferramentas e Tecnologias*

Nas primeiras seções é introduzido o conceito de “blocos” para o Moodle e justificada as escolhas de quais serviços o *plug-in* pode oferecer. Nas seções seguintes são realizadas descrições e observações referentes as principais ferramentas e tecnologias presentes no trabalho, de forma a fundamentar teoricamente o leitor para o capítulo seguinte no qual, de fato, será descrito o *plug-in*.

### 2.1 Módulo de Extensão

Módulo de extensão, mais conhecido como *plug-in*, é um tipo de programa que integra-se a aplicações para adicionar uma ou mais funcionalidades. As aplicações devem ter sido projetadas para receberem os *plug-ins* e, geralmente, são publicadas especificações para permitir que terceiros desenvolvam novas funcionalidades para estas aplicações. (CNET GLOSSARY, Termo “plug-in”, 2012)

Algumas das vantagens que o uso de *plug-ins* pode vir a oferecer são listadas a seguir:

- Permite a expansão de aplicações sem que sua execução seja interrompida;
- A complexidade do sistema é minimizada através da modularização;
- Permite ampliação de sistemas através do desenvolvimento de componentes sem interferir em outros módulos e sem interferir em todo o sistema;
- Permite a incorporação de funcionalidades desenvolvidas por terceiros.

Uma diferença importante entre aplicações baseadas em *plug-ins* e aplicações baseadas em outros componentes é que os primeiros são opcionais ao invés de componentes obrigatórios. O sistema deve executar corretamente com ou sem o uso de *plug-ins*, oferecendo suas funcionalidades básicas.



### 2.1.1 Arquitetura

A arquitetura de um *plug-in* pode ser vista como um quebra-cabeça, no qual há “lacunas” e “peças” que completam as lacunas. Uma lacuna é uma interface conhecida pela aplicação com um formato específico, e as peças tratam-se de classes no *plug-in* que possuem este formato. (EISENBACH, S. et al., 2003)

Nesta arquitetura, a interface define um conjunto de regras (o formato). Estas são, por exemplo, linguagem de programação, estrutura de diretórios do código, a presença de determinada classe no projeto, entre outros. Se o *plug-in* satisfizer as especificações definidas para a interface, uma conexão entre ambos é possível. A partir disto o sistema que estará recebendo o novo *plug-in* terá condições de executá-lo corretamente.

A Figura 2.1 mostra como os *plug-ins* 1 e 2 preenchem duas lacunas, cada um com suas próprias especificações. Em uma analogia com um *player* de vídeo, a aplicação seria a unidade principal, com menus de opções e a interface de execução, e os *plug-ins* poderiam oferecer novos formatos de áudio e vídeo.

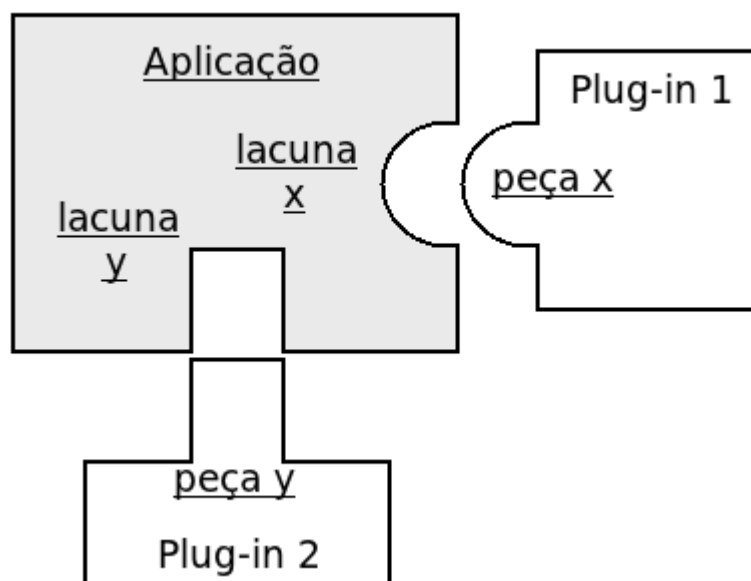


Figura 2.1: Representação em quebra cabeça da arquitetura de *plug-ins*.

### 2.1.2 Funcionalidades

As aplicações que permitem a adição de novas funcionalidades através dos *plug-ins* definem um conjunto de ações e permissões para estes. A partir deste conjunto de ações que serão criadas as novas funcionalidades e serviços. Os *plug-ins* podem ser limitados à um apenas

este conjunto de ações ou podem ter permissão para acessar livremente o sistema operacional. Para estes últimos a variedade das funcionalidades que pode oferecer é muito alta, sendo capaz de comunicar-se com outras aplicações para disparar eventos, como iniciar a execução de um programa, ou buscar resultados, ao final da execução de um programa.

## 2.2 Blocos para Moodle

Do ponto de vista do Moodle, *plug-ins* são “Blocos” a serem anexados ao sistema e que trazem novas funcionalidades, sendo identificados por “blocos para Moodle”. A comunidade *on-line* oficial do Moodle - encontrada no endereço <http://moodle.org/> - possui uma vasta lista de blocos personalizados para Moodle, produzidos pelos membros e com variadas funcionalidades. A partir desta lista foram pesquisados os blocos para Moodle que ofereçam serviços relacionados ao uso do SMS e a tabela 2.2 foi montada com os dados da pesquisa (*Dados coletados em 21/02/2012*).

Plug-in	SMS como parte fixa do plug-in	Uso a partir do Moodle				Uso a partir de um celular			Faz uso de um provedor
		Enviar SMS	Receber SMS	SMS automático	Múltiplos destinos	Enviar SMS	Receber SMS	Acessar serviços	
Notifications	X	X	X	O	X	X	O	X	O
Appointments	X	X	X	O	X	X	O	X	O
Moodletxt*	O	O	O	X	O	O	O	X	O
Sending SMS block	O	O	X	X	O	X	O	X	O
Ozek, NG SMS for Moodle*	O	O	X	X	O	X	O	X	O

X - Indica a inexistência do serviço; O - Indica a existência ou obrigatoriedade do item/serviço.

\* Tanto o "Moodletxt" quando o "Ozek, NG SMS for Moodle" exigem o uso de um provedor de serviços específico, que é o mesmo que desenvolveu o plug-in.

Figura 2.2: Tabela comparativa dos blocos para Moodle que possuem funcionalidades relacionadas ao uso de SMS.

As colunas na tabela representada na Figura 2.2 são descritas a seguir:

- **SMS como parte fixa do *plug-in*** - Refere-se a necessidade de modificações no bloco para que este venha a fazer o uso do serviço de SMS.
- **Uso a partir do Moodle** - As funcionalidades do serviço de SMS acessíveis a partir do Moodle, que são:
  - Enviar SMS - Oferece uma interface para o envio de SMS;

- Receber SMS - Oferece uma interface para a leitura de SMS recebidos;
  - SMS automático - O SMS se apresenta como funcionalidade extra de outro serviço;
  - Múltiplos destinos - Permite o envio de SMS para múltiplos destinos.
- **Uso a partir de um celular** - As funcionalidades de disponibilidade oferecidas pelo bloco:
    - Enviar SMS - Permite que uma mensagem seja enviada de um celular e seja acessada através do fórum;
    - Receber SMS - Permite que uma mensagem seja enviada do fórum para um celular;
    - Acessar serviços - Permite que serviços dentro do fórum sejam acessados a partir de um celular.
  - **Faz uso de um provedor** - Indica se as funcionalidades do serviço de SMS oferecidas pelo bloco ficam dependentes da contratação de um provedor, na Internet, para o envio de SMS.

Através da análise da tabela pode-se concluir principalmente que os blocos para Moodle não permitem a execução de um serviço no sistema Moodle, através de um bloco, utilizando um telefone celular. Ainda, obrigam o uso de provedores de Serviço de Mensagens Curtas na Internet, impedindo a utilização de outros meios para prover este serviço.

Finda esta análise, optou-se por implementar um *plug-in* para atender estes pontos, permitindo a escolha de como o serviço de SMS será fornecido, seja por um provedor na Internet, seja por um IPBX local, e por permitir que serviços no *plug-in* sejam acessados a partir de um telefone celular, sendo estes serviços o Agendamento de envio de SMS, e o Envio para múltiplos destinos, podendo ser combinados.

## 2.3 Daemon

*Daemon* é o termo utilizado para designar processos que rodam em *background* e que independem do controle de um usuário. Estes processos podem ser utilizados para responder automaticamente a determinados eventos, executar operações em determinados intervalos de tempo ou até mesmo realizar a aquisição e tratamento de dados. (LOVE, 2007)

Algumas das características que um *daemon* pode apresentar são:

- Ser inicializado durante o *boot* do sistema;

- Ser filho direto do processo *init* (em ambientes Linux);
- Não requerer interação com um usuário;
- Executar de forma independente aos outros processos do sistema.

### 2.3.1 Criação de Daemons

Basicamente, o primeiro passo para um processo tornar-se um *daemon* é fazendo um *fork*. *Forking*, ou bifurcação, é o método pelo qual o processo é clonado, criando um processo-filho idêntico. Em seguida, o processo chama a função *exit* e finaliza sua execução, fazendo com que o processo-filho seja adotado pelo processo *init*<sup>1</sup>. Durante todo este procedimento o processo-filho já deverá estar executando seu programa. (SUSE DOC., O processo *init*, 2012)

## 2.4 Serviço de Mensagens Curtas

O Serviço de Mensagens Curtas (*Short Message Service* (SMS)) é um serviço que permite a transmissão e recepção de mensagens de texto, conhecidas por “SMS”, através de telefones celulares ou quaisquer outros dispositivos capazes de gerar SMS. As mensagens podem possuir no máximo 160 caracteres, com codificação *Global System for Mobile* (GSM) (codificados com 7 bits), ou 140 caracteres (codificados com 8 bits).

Originalmente o SMS foi projetado como parte da tecnologia de redes *Global System for Mobile* (GSM), que é utilizada neste trabalho e sua relação com o SMS será melhor explicada nas sub-seções seguintes, mas atualmente está disponível em uma grande quantidade de outras redes. Também se discute e planeja-se a evolução de mensagens de texto para mensagens com multimídia (*Multimedia Messaging Service* (MMS)), com o usuário podendo enviar, por exemplo, imagens e áudio dentro do SMS.

---

<sup>1</sup>O processo *init* é o primeiro programa executado em um ambiente Linux. Quando este processo é iniciado, ele completa o procedimento de inicialização, executando diversos programas responsáveis pelas funções básicas do sistema. Todos os programas são iniciados diretamente pelo processo *init* ou por um de seus processos-filho.

### 2.4.1 Comunicação com a rede GSM

O protocolo de transmissão para o SMS é especificado pelo Instituto Europeu de Padrões de Telecomunicações (*European Telecommunications Standards Institute (ETSI)*), e dois serviços diferentes são definidos: o *SMS-SUMIT*, quando o celular é o destino; e o *SMS-DELIVER*, quando o celular é a origem. (MEDEIROS, D. et al., 2008)

### 2.4.2 Funcionamento

As mensagens enviadas partem do aparelho celular para um centro de serviço de mensagens (*Short Message Service Center (SMSC)*). No SMSC as mensagens são processadas utilizando-se um mecanismo de “store and forward” - armazenamento e encaminhamento - sendo enviadas ao destino assim que eles estiverem disponíveis. A entrega das mensagens utilizam a ideia do “melhor esforço”, na qual não há garantias de que a mensagem será realmente entregue. E quanto a velocidade de envio, a Agência Nacional de Telecomunicações (ANATEL) impõe que que os SMS não podem demorar mais que um minuto para serem entregues em 95% dos casos.

### 2.4.3 Modems GSM

Um modem GSM é um tipo especializado de modem que utiliza um “cartão SIM”<sup>2</sup> e opera registrado em uma operadora de telefonia móvel, como um se fosse um telefone celular, podendo inclusive utilizar o Serviço de Mensagens Curtas. De fato, telefones celulares podem funcionar como modems GSM se assim forem especificados pelo fabricante.

Os modems GSM são conectados serialmente e comunicam-se com o host através do protocolo “Attention”. O protocolo Attention é composto por um conjunto de comandos - conhecido por comandos AT - que deverão ser enviados pelo *host* para o modem GSM. O modem, por sua vez, executa o comando solicitado e retorna uma frase curta indicando o estado da operação solicitada pelo comando. Para diferentes fornecedores de modems GSM podem haver conjuntos de comandos especiais e únicos. Mas, por regra, eles deverão possuir pelo menos o conjunto básico - e igual para todos - de comandos AT. (DEVELOPER’S HOME, SMS Tutorial, 2012)

---

<sup>2</sup>Cartão SIM, ou “SIM Card”, é um circuito impresso, de uso obrigatório, utilizado para armazenar dados em telefones celulares da tecnologia GSM.

## 2.5 Remote Procedure Call

O *Remote Procedure Call* (RPC) é uma técnica utilizada para construção de aplicações distribuídas baseadas no modelo cliente-servidor. Esta técnica utiliza a noção de chamadas de procedimentos que executam localmente, porém, no RPC o procedimento chamado não precisa necessariamente ser local. Isto implica na capacidade de processos presentes em sistemas diferentes comunicarem-se através de uma rede, executando procedimentos de forma remota. Ainda, os procedimentos executados através de RPC também podem ser chamados de *callbacks*, que nada mais são do que referências para a execução dos procedimentos. (CARDIFF UNIVERSITY, Remote Procedure Calls (RPC), 2012)

### 2.5.1 XML-RPC

*Remote Procedure Call with XML* (XML-RPC) é um protocolo de chamada de procedimento remoto que utiliza o *Extensible Markup Language* (XML) para descrever os procedimentos de chamadas e o *Hyper Text Transfer Protocol* (HTTP) como mecanismo de transporte (LAURENT, S. et al., 2001).

Uma mensagem XML-RPC é uma requisição “HTTP-POST”, sendo que o corpo da mensagem é escrito em XML. Um procedimento é executado no servidor e um valor é retornado também em XML. Pelo fato de usar o HTTP como mecanismo de transporte, o XML-RPC atinge uma grande quantidade de aplicações, mas também apresenta algumas limitações. Para o uso em larga-escala e aplicações que necessitam de alta velocidade no tempo de resposta, o XML-RPC apresenta ineficiências. Porém, sua simplicidade e eficiência são valores altamente considerados para aplicações com menor processamento de requisições, e nas quais o tempo de resposta não é tão importante.

## 2.6 Sistema Gerenciador de Banco de Dados

Um Sistema Gerenciador de Banco de Dados (SGBD) consiste em uma coleção de dados e uma coleção de programas que permitem acesso á estes dados. O objetivo principal de um SGBD é prover um ambiente que seja adequado e eficiente para uso na recuperação e armazenamento de informações (UNICAMP, 2012). Para este trabalho foi escolhido o PostgreSQL como SGBD, pela sua fácil utilização e vasta documentação.

### 2.6.1 PostgreSQL

O PostgreSQL é um projeto *open source* coordenado pelo PostgreSQL Global Development Group. É um SGBD que se encaixa dentro do conceito “objeto-relacional” e possui mais de 15 anos de desenvolvimento. É considerado extremamente robusto e confiável, além de ser flexível e contar com muitos recursos. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

## 2.7 Linguagens de desenvolvimento

Para este trabalho foram utilizadas como ferramenta as seguintes linguagens: “Python 2.7.2+” para o desenvolvimento do *daemon*, *PHP: Hypertext Preprocessor* (PHP) 5.4.0, *Hyper Text Markup Language* (HTML), JavaScript 1.5 e *Cascading Style Sheets* (CSS) para o desenvolvimento do código do bloco e *scripts* auxiliares.

## 3 *Implementação, Funcionamento e Testes*

O *plug-in* proposto neste trabalho é composto por duas entidades: o “bloco de controle” e o “*daemon* de serviços”. Define-se por bloco de controle a entidade que representa diretamente o *plug-in*, que está anexada ao Moodle, servindo como interface ao usuário, enquanto o *daemon* de serviços permite que os serviços do *plug-in* sejam fornecidos ao sistema. A Figura 3.1 ilustra o conjunto das entidades e os serviços<sup>1</sup> auxiliares anexados a elas.

Neste capítulo são ilustrados os serviços que o *plug-in* é responsável por oferecer e também o funcionamento interno do mesmo. Nas primeiras seções o serviço de Agendamento de SMS e Envio para Múltiplos destinos são explicados em forma de etapas. Nas seções seguintes é explicado de que forma foram implementados o bloco de controle e o *daemon* de serviços, seu relacionamento e como se comportam. Nas seções finais são feitas considerações sobre a implementação e o desempenho apresentado pelo *plug-in*.

### 3.1 Funcionamento dos Serviços

O *plug-in* permite o acesso aos serviços oferecidos através de duas interfaces: pelo bloco de controle, através do Moodle na internet, e pelo SMS. O uso dos serviços é ilustrado no Apêndice A e ambas as interfaces possuem o funcionamento análogo uma a outra. Por isto, apenas o processo de utilização de serviços através do SMS será aqui representada.

#### 3.1.1 Formato das solicitações

Em geral, a utilização de um serviço é feita enviando-se um SMS para o *plug-in*, que deverá processar a informação contida no SMS, e executar o serviço solicitado.

---

<sup>1</sup>O termo serviços será utilizado para denominar tanto os serviços que o *plug-in* oferece ao usuário quanto o Serviço de Banco de Dados e Serviço de Mensagens Curtas, necessários ao funcionamento do *plug-in*.



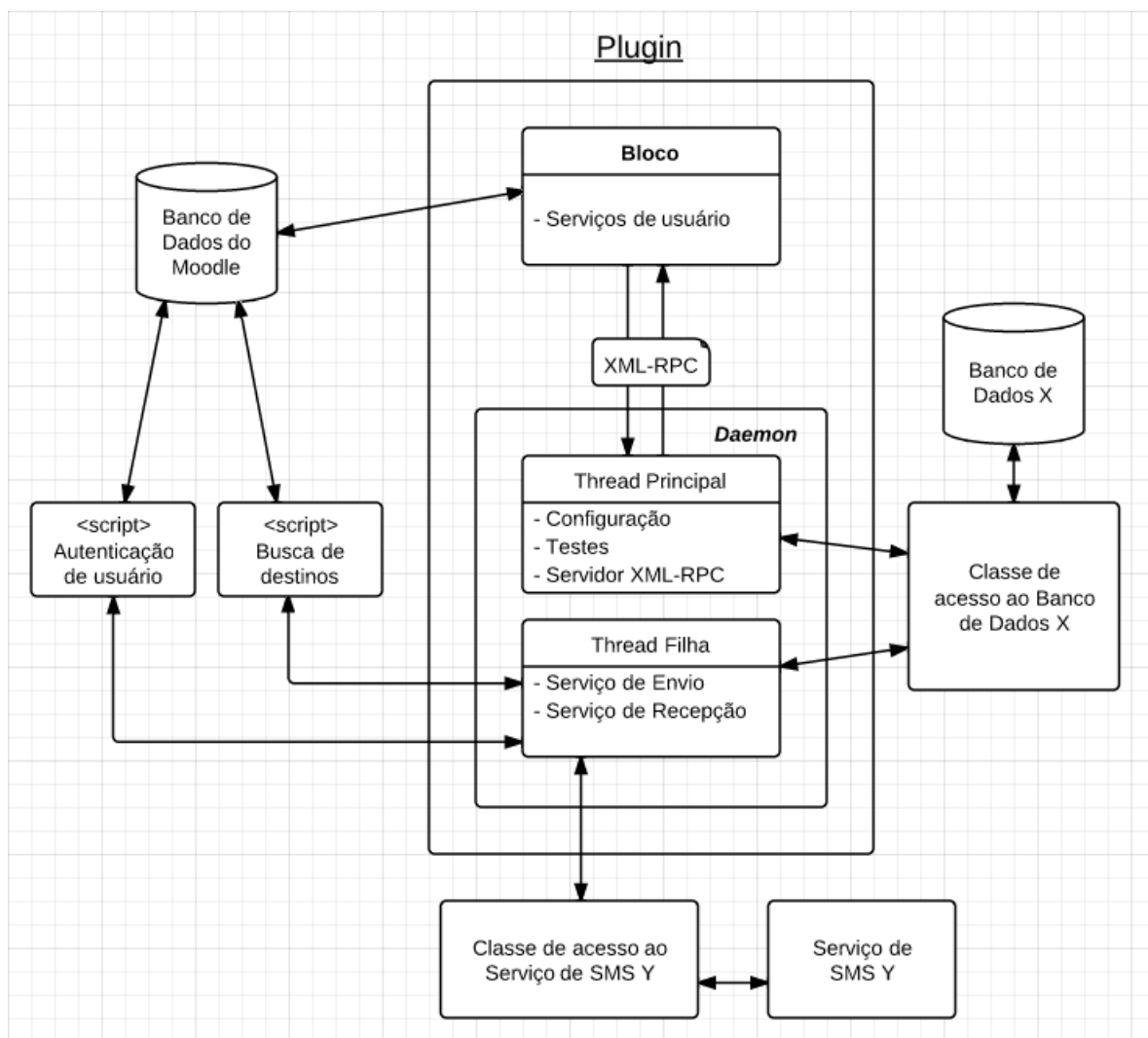


Figura 3.1: Diagrama de Entidades do *plug-in*.

A informação contida no SMS, que é uma mensagem de texto, deve possuir um formato específico, contendo campos de dados. Estes campos são utilizados no *plug-in*, pelo *daemon* de serviços para identificar: o tipo de serviço solicitado, o solicitante do serviço, os destinos da mensagem, a data e hora em que o serviço deve ser executado - caso haja agendamento -, e a mensagem em si. A Figura 3.2 ilustra o formato do SMS para a solicitação de serviços.

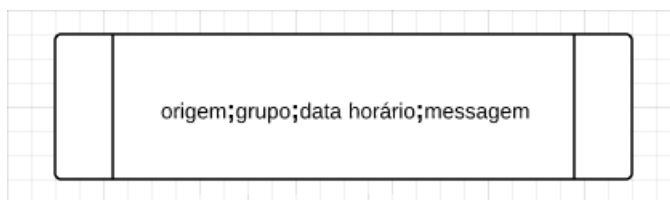


Figura 3.2: Formato genérico da solicitação de serviços através de SMS.

### 3.1.2 Solicitação e Execução do Serviço

Genericamente, as etapas para a utilização de um serviço são as seguintes:

1. Um usuário com permissão - professores e gerentes do sistema Moodle possuem permissão de uso dos serviços - envia, através de seu celular, uma mensagem formatada para o *plug-in*;
2. O *plug-in*, que é composto por um provedor de SMS<sup>2</sup>, com um número de celular, recebe o SMS neste número (ver Figura 3.3);

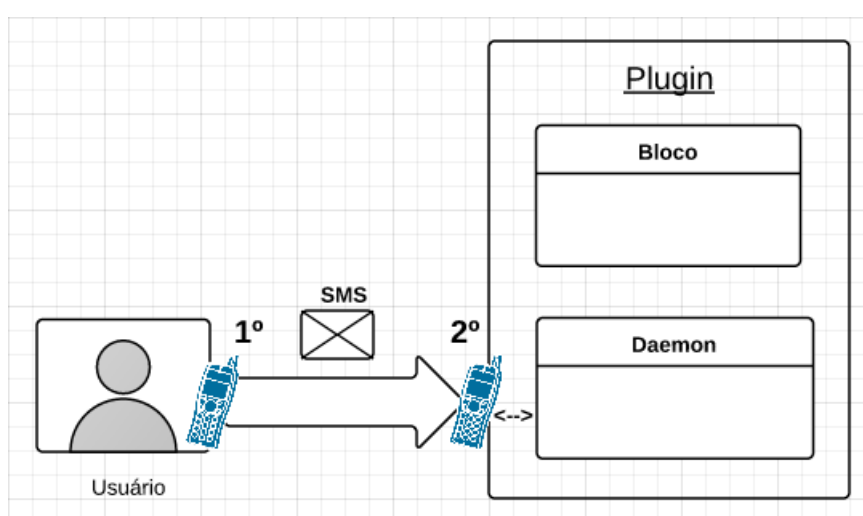


Figura 3.3: Solicitação do serviço através de SMS.

3. O *plug-in* utiliza o número de origem do SMS para validar a origem, realizando uma busca no banco de dados do Moodle pelo usuário através do número de origem do SMS. Se encontrar um usuário que possua o número cadastrado, e este usuário possuir permissão, o *plug-in* continua o processamento da solicitação, caso contrário, descarta a mensagem (ver Figura 3.4);
4. Ainda durante o processamento da mensagem, o *plug-in* usa o campo de destinos da mensagem para, de fato, identificar quem deverá receber a mensagem. Este campo contém uma identificação para os destinos, e esta identificação pode ser um nome ou número que remeta a um grupo pré-cadastrado no sistema Moodle (ver Figura 3.4);
5. Após adquirir os destinos da mensagem e tendo separado os dados de todos os campos, o *plug-in* deverá salvar todos estes dados em um banco de dados. O campo de data e hora

<sup>2</sup>O provedor de SMS é qualquer sistema ou entidade que permita o acesso ao serviço de mensagens curtas na rede GSM. Neste caso, um celular é utilizado para acessar o Serviço de Mensagens Curtas.

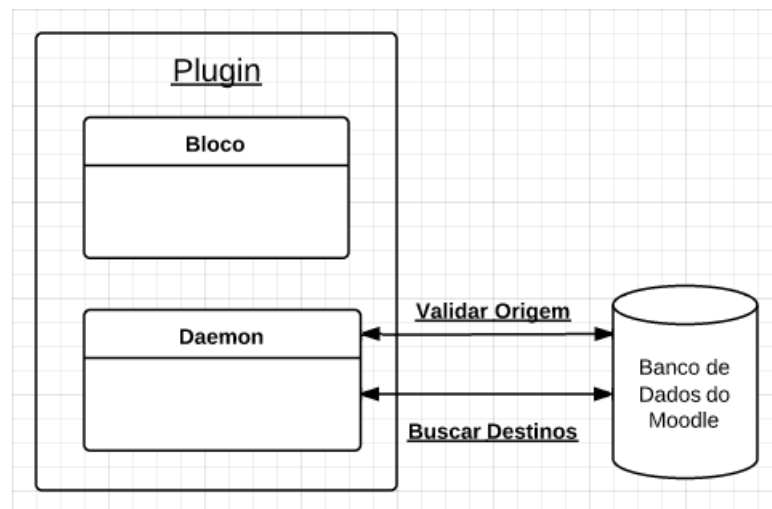


Figura 3.4: Validação da origem e busca dos destinos.

caracteriza o serviço solicitado. Caso não seja preenchido pela origem da mensagem, o serviço será definido não como agendamento e sim como envio imediato, mas mesmo assim o *plug-in* armazenará os dados e o serviço é executado assim que possível.

6. Caso na solicitação houver sido especificado o agendamento, as mensagens serão enviadas apenas na data e hora marcadas e para os destinos salvos no banco de dados (ver Figura 3.5). Após a execução do serviço, a entrada da solicitação no banco de dados é marcada com o estado da execução, permitindo ao usuário visualizar, através do bloco de controle, se o serviço foi executado corretamente.

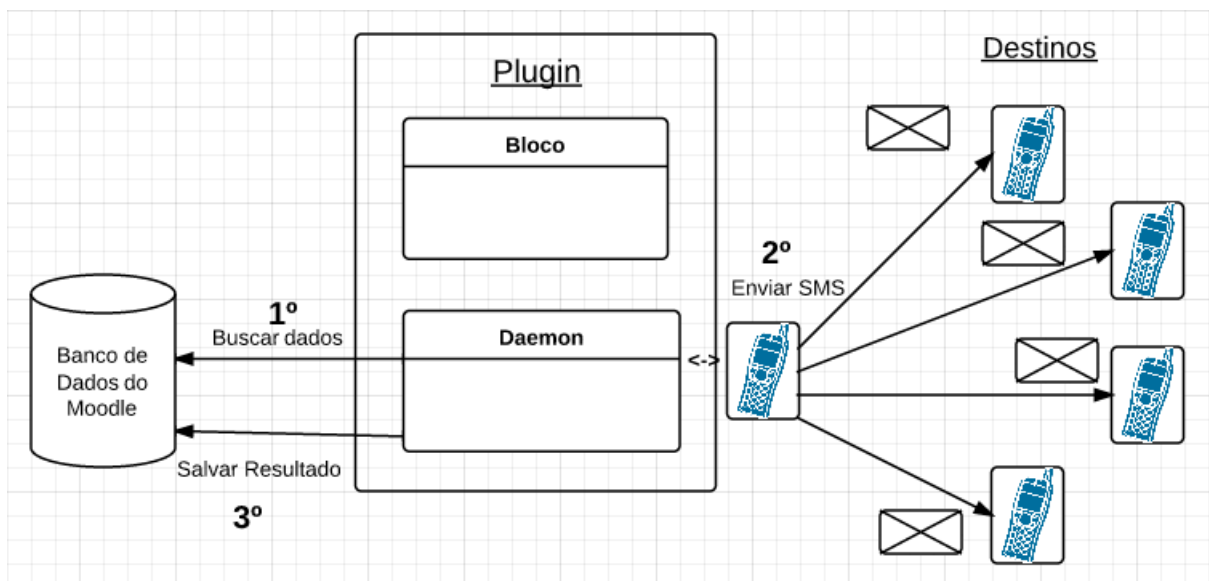


Figura 3.5: Recuperação e execução do serviço.

## 3.2 Bloco de controle

O bloco de controle oferece uma forma de o usuário configurar e monitorar os serviços oferecidos pelo sistema. Ele é a parte visível do *plug-in* e deve ser instalado no Moodle pelo administrador do sistema. Também fornece um menu de opções onde são abertas janelas com os serviços do sistema. As opções da janela são: *Agendar SMS*, *Registros* e *Relatórios*.

Dentro do Moodle os usuários são vistos através de níveis de acesso, podendo ser classificados como “administradores”, “professores”, “alunos”, “visitantes”, entre outros. A partir desta classificação o bloco de controle implementa um controle de acesso aos serviços. Os serviços oferecidos pelo bloco de controle são acessíveis apenas pelo administrador do sistema e pelos usuários com nível de acesso de professor.

Além da limitação por permissão de acesso, o administrador possui a opção de “ligar” e “desligar” o *plug-in*, tornando-o indisponível pela interface do bloco de controle para todos os usuários. Tais opções e atribuições no fórum virtual são descritas no manual de instalação e operação do *plug-in*, que consta no Apêndice A.

### 3.2.1 Serviços

O funcionamento e objetivos dos serviços presentes no menu do bloco de controle são descritos a seguir:

- **Agendar SMS:** Este serviço, acessível também por SMS, oferece opções de agendamento e envio instantâneo de mensagens (para um ou múltiplos destinos). O envio instantâneo de mensagens não deve ser entendido com uma troca rápida de mensagens, mas sim que a requisição será executada o mais rápido possível. Neste serviço devem ser inseridos dados como destinos, mensagem e data/hora de agendamento, e estes dados são submetidos ao *daemon* de serviços, registrando uma nova requisição<sup>3</sup>;
- **Relatórios:** Este serviço completa o serviço de agendamento de SMS pois permite o monitoramento dos estados das requisições registradas. Ao acessar a interface deste serviço pode-se selecionar qual o estado de requisição a ser monitorado. Os possíveis estados são “ativos”, “falhos” ou “enviados”. Ao selecionar um dos estados, o bloco de controle comunica-se com o *daemon* de serviços e recupera todas as requisições que estiverem no

---

<sup>3</sup>Por requisição deve ser entendido como “o conjunto de dados registrados para solicitar a execução de um serviço”. Por exemplo, o conjunto de “uma mensagem”, “destinos” e “data/hora de envio” formam uma requisição de envio.

estado escolhido<sup>4</sup>;

- **Registros:** Este serviço serve como uma forma de monitorar o funcionamento do *daemon* de serviços. Ele trata de recuperar todas as mensagens de registros do *daemon* de serviços e apresentar no bloco de controle. Também oferece a opção de apagar todo o arquivo de registros.

### 3.2.2 Regras

Para o desenvolvimento do bloco de controle foi necessário seguir as regras de compatibilidade para *plug-ins* do Moodle, para correta execução do *plug-in*. Entre estas regras estão:

1. Utilizar a linguagem PHP para escrever o código do bloco de controle;
2. A presença de uma classe base<sup>5</sup> - identificada neste projeto pelo nome “`block_control_by_sms.php`” na qual funções de configuração e inicialização do bloco de controle são escritas;
3. Um arquivo contendo a versão do bloco de controle, para fins de compatibilidade entre o *plug-in* e versões de Moodle diferentes a qual o *plug-in* foi desenvolvido;
4. A presença de um diretório de linguagens, do qual são extraídos os textos e as mensagens a serem exibidas no bloco de controle.

A regra 4 apresenta um detalhe, a linguagem utilizada para o desenvolvimento da parte visual do *plug-in* é o português, mas no diretório de linguagens está sendo identificado como “en” de *english* (inglês). Não foi possível encontrar na documentação dos blocos para Moodle como alterar a tipo de linguagem a ser recuperado do diretório de linguagens, então o nome padrão “en” foi mantido, contendo textos em português.

## 3.3 Daemon de serviços

O *daemon* de serviços é responsável por receber e tratar requisições. Ele é composto por duas *threads*<sup>6</sup>, uma principal e outra secundária, que será referida por “*thread-filha*”.

<sup>4</sup>Os relatórios apresentam o estado do processamento das requisições, não da entrega do SMS.

<sup>5</sup>“Classe base” está relacionado ao conceito de “heranças”, da programação orientada a objetos. O conceito de heranças permite que uma classe tenha o comportamento de outra classe - a classe base - e estenda ou modifique este comportamento para situações específicas. (IBM, 2012)

<sup>6</sup>Processos são vistos como programas em execução com uma única linha de controle. Em alguns sistemas operacionais, como o Linux, é possível executar múltiplas linhas de controle dentro de um processo. Estas linhas

A *thread* principal, ao ser inicializada, faz chamadas de funções de configuração e teste de serviços, lança a *thread*-filha e em seguida estabelece um servidor XML-RPC. Enquanto isto, a *thread*-filha fica responsável por tratar requisições pelo recebidas via SMS.

A execução de uma requisição ocorre na data e horário definidos durante o registro da requisição. Caso o usuário opte pelo envio imediato das mensagens, a requisição será registrada com uma marcação para envio imediato. A estrutura de inicialização do *daemon* de serviços é representada na Figura 3.6.

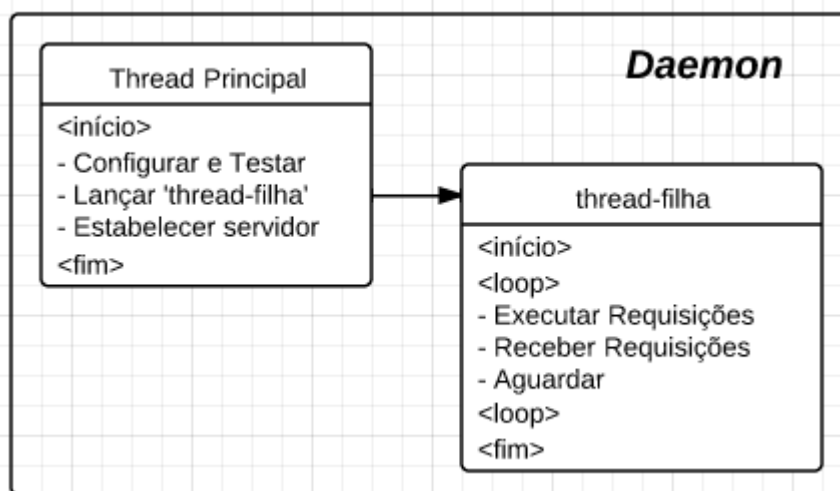


Figura 3.6: Estrutura lógica do *daemon* de serviços

### 3.3.1 Banco de Dados

O *daemon* de serviços foi implementado de forma que o Serviço de Banco de Dados seja transparente. Para isto acontecer foi criada uma classe base que possui funções de acesso bem definidas. Esta classe base permite à um desenvolvedor escolher seu banco de dados de preferência e implementar uma nova classe com funções específicas para este banco de dados. Esta abstração é interessante porque modulariza os serviços, tornando possível a escolha da melhor opção para cada cenário.

O banco de dados utilizado neste trabalho é PostgreSQL. Sendo assim, foi criada uma classe - a classe "Pgcom" - que implementa de forma específica as funções exigidas pela classe base, chamada "DatabaseTemplate". Ambas as classes podem ser encontradas no diretório:

```
"diretorio_raiz_do_plugin/daemon_src/libs/dbcom".
```

Entre as principais funções de acesso ao banco de dados estão:

de controle normalmente são chamadas de "threads", ou em certas ocasiões, "processos leves" (TANENBAUM, A. et al., 2000).

- Configuração do banco de dados. Definir informações de comunicação e acesso (endereço e porta; usuário e senha; nome do banco de dados registrado);
- Registro de requisições;
- Recuperação de requisições;
- Mudança de estado de requisições.

Durante a configuração do Serviço de Banco de Dados também é realizado um teste para verificar a presença ou não da tabela que o *daemon* de serviços utiliza para armazenar seus dados<sup>7</sup>. Quando a tabela não é encontrada - como no caso do *daemon* de serviços estar rodando pela primeira vez - ela é criada no banco de dados.

A Tabela 3.1 representa as colunas da tabela criada no banco de dados durante a primeira inicialização do *daemon* de serviços.

orig	dest	msg	oper	send	blow	stat	src	ext	id
------	------	-----	------	------	------	------	-----	-----	----

Tabela 3.1: Colunas da tabela utilizada pelo *daemon* de serviços.

A seguir é apresentada uma breve descrição de cada coluna da tabela representada na Figura 3.1, o que ajudará a compreender melhor o processamento dos serviços pelo *daemon* de serviços:

- **orig** - Um nome curto que constituirá o SMS enviado, identificando a origem;
- **dest** - Os números de destino das mensagens;
- **msg** - O conteúdo do SMS;
- **oper** - A operadora preferencial para o envio do SMS - não é utilizado<sup>8</sup>;
- **send** - Informa se a requisição deve ser processada imediatamente;
- **blow** - A data e o horário em que a requisição deve ser processada;
- **stat** - O estado da requisição (ativa, enviada ou falhou);

<sup>7</sup>Deve-se salientar que o *daemon* de serviços conhece apenas o nome da função a ser chamada, a verificação que ela executa faz parte da implementação definida pelo desenvolvedor. Neste caso existe a verificação da tabela no banco de dados, mas para outros casos isto poderia não acontecer.

<sup>8</sup>O campo não utilizado continua presente na tabela como sugestão para aperfeiçoamento e implementações futuras do *plug-in*.

- **src** - Como a consulta foi registrada (se foi pelo bloco de controle ou pelo Serviço de Mensagens Curtas);
- **ext** - O número de celular do usuário solicitante do serviço;
- **id** - O identificador único do registro da requisição (utilizado para consultas).

### 3.3.2 Serviço de Mensagens Curtas

A comunicação com o Serviço de mensagens curtas foi implementada de maneira semelhante à do Serviço de Banco de Dados. Sendo assim, o usuário terá total liberdade para escolher como o Serviço de Mensagens Curtas será fornecido. Basta que implemente as funções definidas na classe base, neste caso, chamada de “GsmTemplate” e contida no diretório:

```
“diretorio_raiz_do_plugin/daemon_src/libs/gsmcom/GsmTemplate.py”.
```

Caso o usuário deseje utilizar, por exemplo, um servidor PBX para oferecer o Serviço de Mensagens Curtas, ele deve implementar as funções de acesso com o protocolo de comunicação específico para aquele PBX. O mesmo se repetirá para outras formas de acessar um Serviço de Mensagens Curtas, apenas adaptando as funções de acesso com as especificidades do provedor.

### 3.3.3 Thread-Filha

A *thread*-filha complementa a função que a *thread* principal desempenha, fornecendo a disponibilidade e tratando dos envios de SMS. O código da *thread*-filha pode ser encontrada na classe “Manager”, contida no diretório de bibliotecas do *daemon* de serviços:

```
“diretorio_raiz_do_plugin/daemon_src/libs/manager/Manager.py”.
```

Para desempenhar seu papel, a *thread*-filha possui um laço de repetição que executa indefinidamente (i.e., nunca para sua execução). Dentro deste laço há funções de recepção, execução e controle de novas requisições. A cada ciclo de chamadas no laço a *thread*-filha aguarda um tempo para executar novamente. A figura 3.7 ilustra este comportamento.

### Execução e Controle

Quando a função de execução e controle é chamada, são realizados uma série de passos, desde recuperar os dados das requisições já registradas até realizar ou não o processamento da requisição. Os passos executados são apresentados com detalhes na lista a seguir:



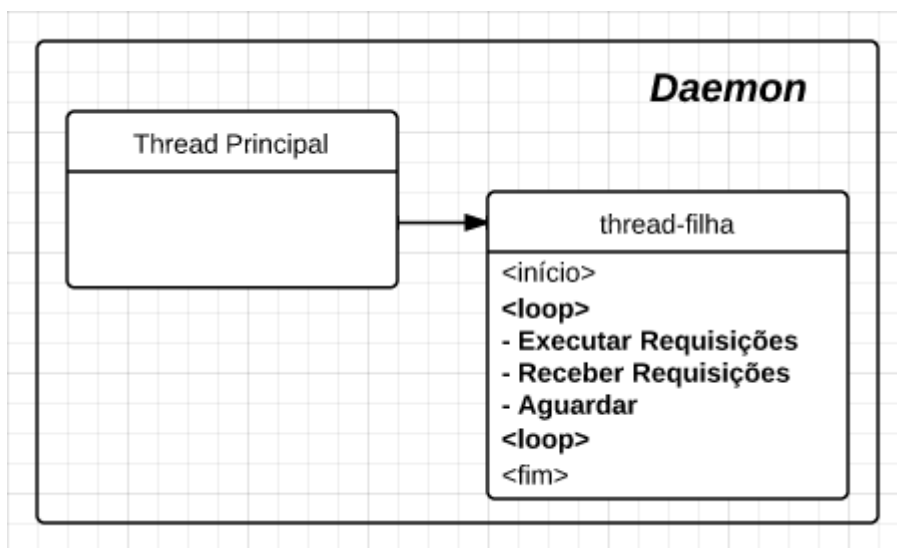


Figura 3.7: Execução da *thread-filha*.

1. São recuperados todos os registros no banco de dados que estiverem no estado “ativo”. Esta busca é feita utilizando-se a coluna “stat”, descrita na Tabela 3.1;
2. Para cada registro ativo:
  - (a) É realizado o cálculo da diferença de tempo entre a data e horário correntes e a data e horário definidos para a execução da requisição (se a requisição for marcada para envio imediato o sistema registra um valor arbitrário);
  - (b) Caso a requisição esteja fora do limite de atraso (muito atrasada) ela será cancelada, tendo seu estado configurado para “falhou”;
  - (c) Caso a requisição esteja marcada para envio imediato (implicando que ela foi aprovada no passo anterior), ela será executada;
  - (d) Caso a requisição esteja muito adiantada, é ignorada e será avaliada na próxima volta do laço;
  - (e) Caso a requisição esteja dentro do intervalo de adianto e atraso<sup>9</sup>, ela será executada.
3. A execução da requisição ocorre através da chamada de uma função (fornecida pelo Serviço de Mensagens Curtas), para cada destino registrado, que recebe como argumentos o destino e a mensagem a ser enviada, e por sua vez, retorna o estado da execução do serviço;
4. Quando a mensagem for enviada a todos os destinos, o estado da requisição é alterado. Se todas as mensagens forem enviadas com sucesso, a requisição é configurada para o

<sup>9</sup>O intervalo de adianto e atraso é um intervalo de tempo calculado a partir do horário registrado para a requisição

estado “enviada”. Se um ou mais envios não tiverem sido completados, a requisição é configurada para “falhou”.

## Recepção

Em seguida ao serviço de execução e controle, é executado o serviço de recepção de novas requisições. Quando é solicitado o envio de uma mensagem, a função leva algum tempo para ser executada e retornar o resultado. Este atraso é dependente da maneira como o Serviço de Mensagens Curtas é fornecido ao sistema.

A disposição da chamada da função de execução e controle e da função de recepção, assim como a presença de um intervalo de adianto e atraso (comentado na subseção 3.3.3), são justificados pelo atraso inserido através do SMS. A recepção de novas requisições executa os seguintes passos:

1. É adquirida uma contagem das mensagens recebidas, utilizando para isto uma função específica do Serviço de Mensagens Curtas;
2. A partir da contagem, uma a uma as mensagens são recuperadas e testadas:
  - (a) O número de origem do SMS é utilizado para testar a permissão de acesso aos serviços;
  - (b) O conteúdo do SMS é tratado, separando os campos de dados contidos na mensagem;
  - (c) Os destinos para a requisição são recuperados através de um dos campos separados no passo anterior.
3. Já com o conteúdo do SMS recuperado, ele não é mais necessário e uma função é chamada para removê-lo;
4. Por fim, caso a o SMS seja válido e tenha passado por todos os passos anteriores com sucesso, ele é registrado no banco de dados em forma de uma nova requisição. Caso tenha ocorrido alguma falha ou alguma incoerência seja encontrada, a requisição é descartada.

Quando o sistema adquire as informações do SMS recebido, utiliza o número de origem da mensagem para validar as permissões de acesso aos serviços. Na verdade, o *daemon* de serviços executa um *script* externo e independente, passando-lhe como argumento o número a ser validado, e este retorna positivo, se o número pertence à um usuário com acesso aos serviços, e negativo, caso contrário. O *script* de validação de usuário está contido no caminho:

“diretorio\_raiz\_do\_plugin/daemon\_modules/validation.php”.

Para o item 2c, com a recuperação de destinos, é realizado o mesmo procedimento. O *daemon* de serviços executa um *script* passando-lhe um argumento e, caso o argumento seja válido, recebe uma lista de destinos para a requisição. Para as requisições que são registradas através de SMS, os destinos são especificados através do nome dos grupos do Moodle. Sendo assim, todos os membros associados ao grupo especificado receberão a mensagem. O *script* que recupera os contatos está contido no caminho:

“diretorio\_raiz\_do\_plugin/daemon\_modules/get\_contacts.php”.

Ambos os *scripts* executam chamadas ao banco de dados do Moodle, com regras pré-definidas para validação da origem e busca pelos destinos. A chamada de *scripts* externos pelo *daemon* de serviços é representada na Figura 3.8.

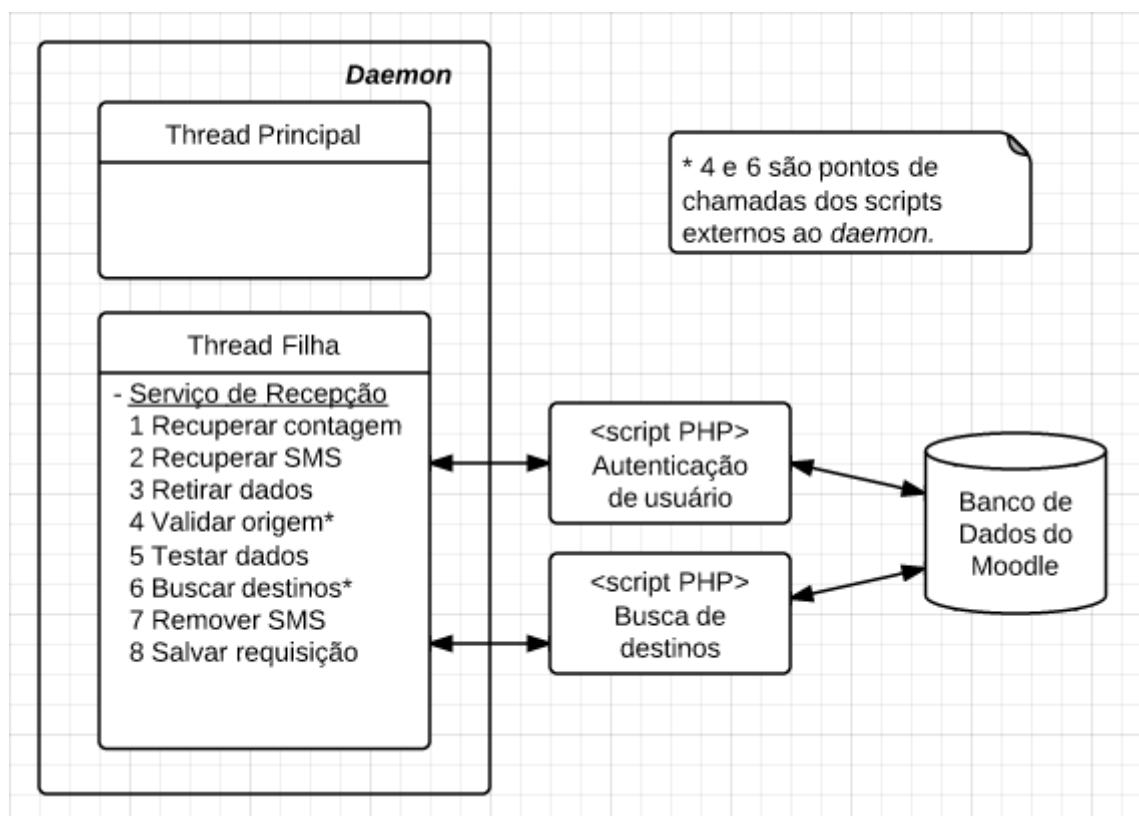


Figura 3.8: Pontos de chamadas de *scripts* externos pelo *daemon* de serviços.

### 3.4 Comunicação entre bloco de controle e daemon de serviços

A comunicação entre bloco de controle e *daemon* de serviços é realizada através do servidor XML-RPC, estabelecido na *thread* principal. Sendo possível para o bloco de controle acessar

as funções no *daemon* de serviços através das *callbacks*. Para o estabelecimento do servidor foi utilizada uma classe base do *Python*, o “SimpleXMLRPCServer”. A configuração do servidor consiste em dois passos básicos: definir endereço e porta de acesso e registrar as funções de *callbacks* que serão oferecidas pelo servidor.

A configuração do endereço e porta devem ser definidas manualmente pelo usuário. Os valores padrão para esta configuração são: endereço “127.0.0.1”; porta “3435”. Estes valores podem ser alterados antes da inicialização do *daemon* de serviços, direto no código. O Apêndice A possui uma explicação de como configurar estes valores.

O bloco de controle é a única entidade capaz de iniciar a comunicação, sendo atribuído ao *daemon* de serviços o papel de apenas receber as chamadas e nunca iniciar uma. As *callbacks* que o bloco de controle pode realizar são:

- **newRequisition:** Realiza o registro de uma nova requisição no sistema;
- **getRequisitions:** Recupera requisições previamente registradas no sistema;
- **getLogs:** Recuperar os registros de funcionamento do *daemon* de serviços;
- **cleanLogs:** Apaga os registros de funcionamento do *daemon* de serviços.

## 3.5 Ferramenta de registros e documentação

A documentação do código foi feita principalmente no estilo *inline*<sup>10</sup>, sendo documentadas quase todas as funções e os trechos julgados mais importantes.

Para os códigos do bloco de controle, abrangendo os códigos escritos em PHP, JavaScript, HTML e CSS, foi realizada apenas a documentação *inline*. Para o código do *daemon* de serviços, essencialmente escrito em Python, além da documentação *inline* para trechos de código, foi utilizada a ferramenta de documentação da própria linguagem, as “DocStrings”<sup>11</sup>.

O *daemon* de serviços utiliza uma ferramenta de registros chamada “slog”, criada neste trabalho para auxiliar no desenvolvimento e controle da execução do mesmo. Praticamente todas as ações executadas pelo *daemon* de serviços são registradas, facilitando a depuração de quaisquer erros registrados pelo sistema. A interface do bloco de controle permite o acesso, apenas pelo administrador do sistema, à estes registros, como comentado no Item 3.2.1.

<sup>10</sup>*Inline Documentation* - Documentação em linha, ou informações que são inseridas dentro do próprio código em questão, servindo para auxiliar em posteriores modificações e manutenção do código.

<sup>11</sup>“DocStrings” é uma ferramenta que permitem que toda a documentação *inline* (registrada de uma forma determinada) seja recuperada em um formato especial para leitura.

## 3.6 Características do Plug-in

### 3.6.1 Desempenho

A análise de desempenho do *plug-in* foi baseada em observações sobre a sua capacidade de processar requisições por intervalo de tempo. O processamento pode ser dividido em duas partes: o tratamento dos dados e o envio de SMS. Este último é totalmente dependente do provedor do Serviço de Mensagens Curtas.

O Serviço de Mensagens Curtas para este trabalho foi fornecido por um modem GSM de um telefone celular “Nokia E71”. A classe criada - a partir da classe base - para suportá-lo é específica para o tipo de modem que ele utiliza, devido aos comandos AT utilizados.

Foram observadas situações em que requisições agendadas para a mesma data e horário entravam em conflito. Desta forma apenas uma delas era executada, e a outra falhava por estar muito atrasada.

O tempo médio de envio de um SMS - apenas do telefone celular para o SMSC -, para este sistema, é de aproximadamente 10 segundos. Um tempo consideravelmente longo, tendo em vista que uma requisição poderá possuir uma grande quantidade de destinos. Se houverem dois ou mais agendamento para horários muito próximos, há uma chance muito grande de apenas um deles ser processado com sucesso.

O intervalo de adianto e atraso foi introduzido no *plug-in* para amenizar o impacto do processamento das requisições, e deve ser calibrado de acordo com o provedor do Serviço de Mensagens Curtas utilizado no sistema.

Sugere-se que para a utilização do *plug-in* apresentado neste trabalho, seja utilizado um provedor de Serviço de Mensagens Curtas com tempo médio de envio de SMS coerente com a demanda pelos serviços do *plug-in*. Isto para que sejam evitadas ao máximo a perda de requisições por atraso no processamento.

### 3.6.2 Segurança

A implementação deste *plug-in* prestou pouca atenção a questão de segurança, oferecendo basicamente limitações de acesso aos serviços e monitoramento de utilização dos mesmos (registros de requisições).

### Bloco de controle

O acesso aos serviços pela interface do bloco apresenta uma falha. No menu básico oferecido pelo bloco são lançadas novas janelas nas quais são transmitidos alguns parâmetros. Entre estes parâmetros está o “id” do usuário, que serve para limitar o acesso de usuários a aquela interface. Como os parâmetros e os valores são passados através do *Uniform Resource Locator* (URL), é possível que um usuário copie o endereço e acesse a interface fora do ambiente Moodle, sem reais restrições de acesso.

Para corrigir esta falha, deve-se investigar um mecanismo que lance as janelas do bloco ocultando do usuário a URL, que contém os parâmetros de controle de acesso.

### Daemon de serviços

Na comunicação com o *daemon* de serviços, através do RPC, não há qualquer tipo de limitação de acesso, bastando que uma entidade possua o endereço e porta nos quais o *daemon* de serviços estabeleceu seu servidor RPC para acessar os procedimentos e, por consequência, os serviços oferecidos pelo *daemon* de serviços.

Uma solução para a falha no *daemon* de serviços é a criação de regras de *firewall*, permitindo o acesso ao endereço e porta do *daemon* de serviços apenas pelo servidor Moodle.

### 3.6.3 Modularização

Tendo em vista a modularização como sendo “a subdivisão de um sistema em partes menores e com funções específicas”, o *plug-in* apresenta dois contextos de modularização, a “Modularização de Entidades” e a “Modularização de Serviços”.

#### Modularização de Entidades

A Modularização de Entidades refere-se à maneira como o *plug-in* está organizado - em duas entidades distintas e que interagem, o bloco de controle e o *daemon* de serviços - e como é possível reaproveitar uma parte ou outra.

Pela forma como o bloco de controle e o *daemon* de serviços interagem - pelo XML-RPC-, o *daemon* de serviços pode ser “retirado” do conjunto e anexado à qualquer outra aplicação que tenha a capacidade de comunicar-se através de XML-RPC. Esta capacidade é uma grande vantagem do *daemon* de serviços, que por si só consegue oferecer a disponibilidade e determi-

nados serviços, mas também pode ser aproveitado em diversas outras situações que não sejam em fóruns escolares.

Um detalhe pensado para garantir esta capacidade pelo *daemon* de serviços, foi criar a validação do acesso e a busca de destinos para as requisições por *scripts* externos ao *daemon* de serviços, como citado na subseção 3.3.3. Desta forma, para anexar o *daemon* de serviços em uma aplicação diferente, além da comunicação é necessário preocupar-se em criar estes dois *scripts*.

### **Modularização de Serviços**

A Modularização de Serviços refere-se a capacidade de montar diferentes configurações para o *plug-in*. Isto reflete na implementação apresentada neste trabalho, de forma que utilizou o PostgreSQL como SGBD, e um celular com acesso ao modem GSM como provedor do Serviço de Mensagens Curtas.

Com a modularização dos serviços, presente por consequência da existência de uma classe-base (vista nas subseções 3.3.1 e 3.3.2), o usuário é livre para escolher o tipo e provedor do SGBD e Serviço de Mensagens Curtas que melhor satisfazer suas necessidades.

## 4 Conclusões

Este trabalho apresentou a proposta de um *plug-in* para Moodle para oferecer a disponibilidade. Para melhor ilustrar a ideia, foi realizada uma implementação aplicando técnicas e ferramentas específicas, que algumas vezes não demonstraram ser a melhor solução.

Durante o desenvolvimento da solução proposta, percebeu-se que sozinho o bloco para Moodle não possui recursos para alcançar os objetivos do trabalho, por isto optou-se pela utilização de um *daemon* como processo auxiliar, tornando o *plug-in* um conjunto composto por bloco de controle e *daemon* de serviços.

Através do *daemon* de serviços são oferecidos tanto a disponibilidade quanto os serviços do *plug-in*. Deve ser esclarecido que a disponibilidade pode ser utilizada independente dos serviços que o *daemon* de serviços fornece. E, novamente, que pela carência encontrada nos *plug-ins* para Moodle em relação ao uso do SMS, este trabalho optou por oferecer serviços deste tipo.

Por possuir o caráter de uma proposta em forma de implementação, o *plug-in* deverá servir apenas como base para a criação de outros similares. Deve-se considerar que a implementação não foi feita focando-se em questões de desempenho, estabilidade e escalabilidade, apesar de as considerar. Tanto o bloco de controle como o *daemon* de serviços apresentam alguns pontos em que não satisfazem os requisitos para que o *plug-in* seja utilizado em situações reais de uso, como visto anteriormente.

### 4.1 Disponibilização do Plug-in

Todo o desenvolvimento do *plug-in* contou com uma ferramenta de versionamento, chamada “Git-hub”, na qual o código está atualmente disponível para acesso. O repositório no qual encontra-se o código está no seguinte endereço:

“<https://github.com/dendriel/Trigger/tree/stable>”.



## 4.2 Trabalhos Futuros

Para trabalhos futuros sugere-se implementar melhorias no *plug-in* para que torne seu funcionamento mais eficiente. A seguir são listadas sugestões de características do *plug-in* a serem melhoradas.

- Bloco de controle
  - Reestruturar a interface das janelas do bloco de controle, tornando seu estilo totalmente dependente do estilo geral do Moodle;
- *Daemon* de serviços
  - Implementar classes específicas para novos provedores do Serviço de Mensagens Curtas - por exemplo: PBX, HTTP;
  - Adaptar o *daemon* de serviços para que rode em sistemas operacionais diferentes do GNU/LINUX;
  - Reescrever o *daemon* de serviços na linguagem C, visando aumentar o seu desempenho.
- Geral
  - Desenvolver uma estrutura de processamento distribuído de requisições, tendo em vista expansão da capacidade do *plug-in*;
  - Informar à origem o estado de uma requisição, assim que ele for alterado;
  - Analisar o desempenho do *plug-in* (tempo de envio de mensagens) utilizando diferentes provedores do Serviço de Mensagens Curtas;
  - Identificar e satisfazer as necessidades de segurança do *plug-in*;
  - Estudar e, se possível, oferecer a opção de enviar MMS através do *plug-in*.

## APÊNDICE A – Manual de Uso do Plug-in

### A.1 Instalação

Para instalar um bloco no Moodle<sup>1</sup>, deve-se primeiro adquirir os arquivos do *plug-in*, e em seguida copiá-los para o diretório “/moodle/blocks/”, onde o Moodle se encontra instalado. Feito isto, o próprio sistema identificará o novo *plug-in* e requisitará ao administrador a instalação do mesmo.

O administrador deve entrar em sua conta, no Moodle, e acessar a página inicial do fórum virtual. Ao acessar esta página será lançada a interface de instalação de novos *plug-ins*. Nesta interface existe uma lista de *plug-ins* com seus estados. O novo *plug-in* deve constar nela, com seu estado (na coluna *status*) marcado com “To be installed”. E também há uma opção para atualização (ver figura A.1). Ao selecionar a opção de atualização, será carregada uma nova tela e os *plug-ins* pendentes na lista anterior aparecerão como instalados (ver figura A.2).

#### A.1.1 Python e módulos necessários

Pelo fato de o *daemon* de serviços ser escrito em Python, um requisito básico para o funcionamento deste é que exista um interpretador Python instalado na máquina servidora (nos sistemas Linux geralmente há uma versão do Python instalada por padrão).

Um outro ponto a ser visto são os módulos extras utilizados pelo *daemon* de serviços. Módulos extras são conjuntos de funcionalidades utilizadas que não compõe a parte básica do Python. Sendo assim, é necessário instalar estes módulos para que o *daemon* de serviços execute corretamente.

---

<sup>1</sup>Tanto o *plug-in* quanto o manual foram criados baseando-se para versão 2.2.1+ do Moodle. É possível que o *plug-in* funcione para versões diferentes, mas os passos para instalação, configuração e uso podem diferir dos passos apresentados neste manual.

Verificação de plugins Você acessou como admin IFSC (Sair)

This page displays plugins that may require your attention during the upgrade. Highlighted items include new plugins that are about to be installed, updated plugins that are about to be upgraded and any missing plugins. Contributed plugins are also highlighted. It is recommended that you check whether there are more recent versions of contributed plugins available and update their source code before continuing with this Moodle upgrade.

**Number of plugins requiring attention during this upgrade: 7**

[Display the full list of installed plugins](#)

Plugin name	Directory	Source	Current version	New version	Requires	Status
<b>Blocos</b>						
Control by SMS	/blocks/control_by_sms	Extension		2011062800	Moodle 2010112400	To be installed
Notifications	/blocks/moodle_notifications	Extension	2011111701	2011111701	Moodle 2011033003	Up-to-date
PaperCut Print Quotas	/blocks/papercut	Extension	2012010300	2012010300	Moodle 2010112400	Up-to-date
Quick Course List	/blocks/quickcoursest	Extension	2011122300	2011122300	Moodle 2010112400	Up-to-date
Simple Clock	/blocks/simple_clock	Extension	2011111701	2011111701		Up-to-date
Simple html block	/blocks/simplehtml	Extension	2011111701	2011111701	Moodle 2011033003	Up-to-date
<b>Local plugins</b>						
Moodle Adminer	/local/adminer	Extension	2011040401	2011040401	Moodle 2010112400	Up-to-date




Figura A.1: Interface de instalação de novos *plug-ins*.

Atualizando para nova versão Você acessou como admin IFSC (Sair)

**block\_control\_by\_sms**

Sucesso

---

Figura A.2: Confirmação de instalação bem sucedida.

É utilizado apenas um módulo extra que chama-se “python-psycopg2”, e auxilia no controle do SGBD. Pode ser instalado com o seguinte comando<sup>2</sup>: “sudo apt-get install python-psycopg2”.

<sup>2</sup>Os comandos podem variar de distribuição para distribuição do sistema operacional.

## A.2 Configuração

### A.2.1 Acrescentar ao espaço de trabalho

Após a instalação do *plug-in* é necessário adicioná-lo ao espaço de trabalho do usuário. Cada usuário possui seu espaço de trabalho e por isto cada um deverá adicionar o *plug-in* individualmente. Ao entrar com uma conta e senha no fórum virtual, é carregado a página inicial do Moodle. A partir desta página deve-se selecionar a opção “Minha página Inicial” para acessar o espaço de trabalho de usuário (ver figura A.3).

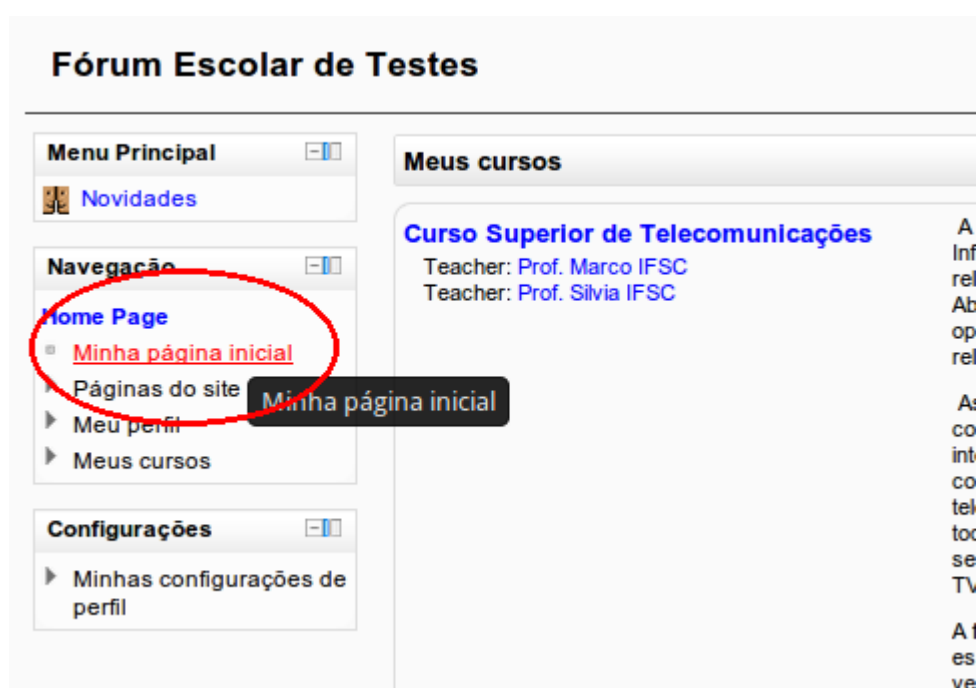
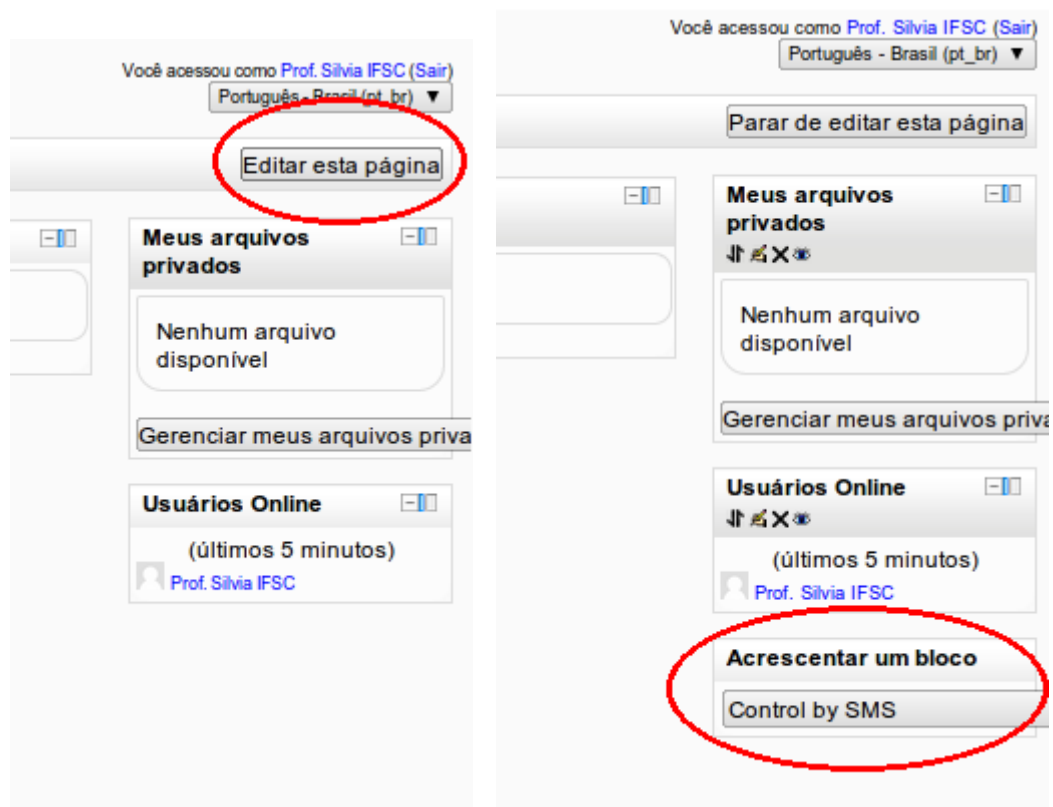


Figura A.3: Minha página inicial.

Na página seguinte - “Espaço de trabalho de usuário” - há a opção “Editar esta página”, que deve ser selecionada para habilitar a edição de páginas (ver figura A.4(a)). Com a opção de edição de páginas habilitada surgirá uma sub-aba chamada “Acrescentar um bloco” no espaço de trabalho do usuário. Esta sub-aba oferece todas opções de *plug-ins* instalados no Moodle. Nela deve ser selecionado o *plug-in* a ser instalado (ver figura A.4(b)).

Assim que o *plug-in* for selecionado na sub-aba de edição, será criada uma nova aba do *plug-in* no espaço de trabalho do usuário, e a partir dela é possível utilizar as funcionalidades deste (ver figura A.5).



(a) Botão “Editar esta página”.

(b) Aba “Acrescentar um bloco”.

Figura A.4: Interface de edição de páginas.

Figura A.5: Interface do *plug-in* no Moodle.

## A.2.2 Configuração e inicialização do daemon de serviços

Após ter instalado o *plug-in* no Moodle é necessário configurar e iniciar o *daemon* de serviços no servidor. Por padrão, o *daemon* de serviços está configurado para executar dentro do próprio servidor em que esta contido o Moodle. Se esta for a configuração atual do sistema, não há necessidade de configurar o *daemon* de serviços. Caso não seja, é necessário configurar o endereço IP (*Internet Protocol* (IP)) que o bloco de controle utilizará para comunicação com o *daemon* de serviços, e também o endereço IP no qual o *daemon* de serviços receberá requisições (estes endereços devem ser iguais). Os arquivos que devem ser configurados são:

- Para o *daemon* de serviços, o arquivo “diretorio\_raiz\_do\_plugin/daemon\_src/main.py”. Deve-se alterar a linha 237, que contém “bind\_address = “127.0.0.1” ”, e substituir o endereço IP “127.0.0.1” pelo endereço IP ao qual deseja-se associar o *daemon* de serviços.
- Para o bloco de controle, o arquivo “diretorio\_raiz\_do\_plugin/php/defines.php”. Deve-se alterar a segunda linha, que contém “\$DAEMON\_ADDRESS = “127.0.0.1” ” e substituir o endereço IP “127.0.0.1” pelo IP em que o *daemon* de serviços estiver configurado.

Feita a configuração das entidades, é necessário iniciar o *daemon* de serviços para que seja possível receber e tratar as requisições do sistema. Este é escrito em python e é inicializado executando, por linha de comando, a entidade principal “main.py”, contida no diretório:

```
“diretorio_raiz_do_plugin/daemon_src/main.py”.
```

A sintaxe do comando é a seguinte “python main.py &”. O “e comercial” contido no comando realiza a execução do *daemon* de serviços em background, evitando o travamento do terminal de trabalho. Em seguida ao início do *daemon* de serviços, pode-se executar no terminal a função “exit”, para que o terminal seja finalizado e o *daemon* de serviços fique sob responsabilidade do sistema. Caso o terminal seja finalizado de forma inesperada - por algum erro - ou fechando-se a janela, o *daemon* de serviços será finalizado junto com ele.

## A.2.3 Banco de dados do daemon

Todas as requisições válidas que chegam até o *daemon* de serviços são armazenadas em um banco de dados. Este banco de dados deve ser diferente do banco de dados do Moodle (soluções em que o *daemon* de serviços utiliza o banco de dados do Moodle podem ser estudadas). O banco de dados controlado pelo *daemon* de serviços é visto de forma abstrata,

utilizando-se de funções genéricas implementadas em uma classe base para controlá-lo. Para este sistema, a classe de acesso ao banco de dados foi implementada baseando-se no uso do SGBD PostgreSQL.

Deve-se disponibilizar - criar ou reutilizar - uma conta, senha e banco de dados para acesso e uso pelo *daemon* de serviços. As configurações no *daemon* de serviços para acessar corretamente o banco de dados devem ser feitas no arquivo contido em “nome\_do\_plugin/libs/defines/defines.py” nas linhas a seguir:

- Linha 65 - contendo “DB\_HOST = “localhost” ” - deve-se substituir a palavra “localhost” pelo endereço IP ao qual o banco de dados está associado;
- Linha 66 - contendo “DB\_PORT = 5432” - deve-se substituir o número “5432” pela porta a qual o banco de dados recebe conexões;
- Linha 67 - contendo “DB\_USER = “trigger” ” - deve-se substituir a palavra “trigger” pelo nome de usuário configurado para uso do *daemon* de serviços;
- Linha 68 - contendo “DB\_PASS = “trigger” ” - deve-se substituir a palavra “trigger” pela senha configurada para uso do *daemon* de serviços;
- Linha 69 - contendo “DB\_NAME = “trigger” ” - deve-se substituir a palavra “trigger” pelo nome do banco de dados criado para uso do *daemon* de serviços.

Caso exista o interesse ou a necessidade do uso de um SGBD diferente, pode-se criar uma nova classe utilizando-se a classe base contida em:

“nome\_do\_plugin/daemon\_src/libs/dbcom/DatabaseTemplate.py”.

## A.3 Configuração do Serviço de Mensagens Curtas

O Serviço de Mensagens Curtas é acessado pelo *daemon* de serviços através de funções padrões contidas na classe base. Qualquer entidade que ofereça o Serviço de Mensagens Curtas pode ser associada ao *plug-in* implementando-se a classe base. Entidades como módulos GSM, telefones celulares, *Private Automatic Branch* (PBX), PBX por software (como Asterisk), e serviços Web de SMS. Para utilizar algum destes outros provedores, basta criar uma classe baseando-se na classe base de modularização do Serviço de Mensagens Curtas. Esta classe base encontra-se no caminho:

`“diretorio_raiz_do_plugin/daemon_src/libs/gsmcom/GsmTemplate.py”`.

A exemplo, o módulo que oferece o Serviço de Mensagens Curtas foi implementado baseando-se na utilização de um telefone celular. A classe desenvolvida encontra-se no diretório:

`“diretorio_raiz_do_plugin/daemon_src/libs/gsmcom/Atcom.py”`

e pode ser utilizada como exemplo para a criação de novas classes que utilizem outros provedores.

### A.3.1 Telefone celular oferecendo o Serviço de Mensagens Curtas

Como citado anteriormente, foi implementado uma classe que oferece o Serviço de Mensagens Curtas baseando-se na utilização de um celular. Para que o sistema funcione neste formato deve-se configurá-lo como se segue.

1. O telefone celular deve possuir um cartão SIM com saldo, para o envio de SMS;
2. Ligar o telefone celular no servidor do *daemon* de serviços através de uma porta serial (RS232; *Universal Serial Bus* (USB));
3. Ao conectar o telefone celular no equipamento, poderá ser questionado (no celular) sobre o modo de operação, então selecionar a opção “Modem”<sup>3</sup>;
4. Identificar a qual porta o telefone celular foi associado no equipamento. Em linux, surgirá um novo “descriptor” no diretório `“/dev/”` do sistema. Geralmente sob o nome `“ttyACM0”` ou `“ttyACM1”`. Em windows, o telefone celular será identificado sob o nome `“COMx”`. Sendo `“x”` variável;
5. É possível testar a funcionalidade do telefone celular como modem. Para isto utiliza-se o programa `“Minicom”`. Dentro de um terminal basta digitar `“minicom -D /dev/-nome_da_porta”` (sem aspas). Quando o programa abrir o terminal para comunicação, digitar `“AT”` e então pressionar ENTER. Deve-se obter como resposta um `“OK”`<sup>4</sup>;
6. Configurar o nome e caminho no qual o telefone celular foi associado. Este dado está contido no arquivo do diretório; `“diretorio_raiz_do_plugin/daemon_src/libs/defines/defines.py”`, na linha 100, que contém a seguinte linha `“MODULE_PORT = “/dev/ttyS0” ”`;

<sup>3</sup>Alguns telefones celulares não possuem esta funcionalidade e não podem ser utilizados

<sup>4</sup>Os comandos descritos são genéricos. Para diferentes telefones celulares pode-se necessitar de comandos diferentes e receber respostas mais complexas



7. Ao iniciar o sistema será registrada uma mensagem de sucesso - dentro do arquivo de registros chamado "system.out" - durante a tentativa de comunicação com a interface GSM. Caso o sistema não obtenha sucesso nesta tentativa, será registrado um erro.

## A.4 Uso

### A.4.1 Permissões

Este *plug-in* possui dois tipos de permissões: permissão de administrador e permissão de professor. Em conjunto com as permissões é utilizado o número de celular do usuário, que serve para validar as requisições recebidas por SMS.

- Permissão de administrador: todos os administradores do fórum virtual possuem esta permissão e tem total controle sobre o *plug-in*. Permite utilizar os serviços, visualizar/apagar registros e habilitar ou desabilitar o uso do *plug-in*.
- Permissão de professor: todos os professores possuem esta permissão. Permite o uso dos serviços oferecidos pelo *plug-in*, quando este estiver habilitado.

Além do acesso aos serviços através do fórum virtual, é possível acessá-los através do Serviço de Mensagens Curtas. Para poder acessar os serviços por SMS, primeiro, é necessário ter permissão de administrador ou permissão de professor, segundo, é preciso cadastrar o número de celular que deseja-se utilizar no perfil do usuário. Apenas os números cadastrados (no campo de número de celular), e com uma das permissões citadas acima, poderão utilizar os serviços do *plug-in*.

### A.4.2 Acesso pelo fórum virtual

A aba criada durante a configuração do *plug-in* possui algumas opções para acesso aos serviços. Cada opção lança uma nova janela com uma interface. O uso dessas interfaces são discutidas a seguir.

#### Agendar/Enviar SMS

A interface de agendamento/envio de SMS conta com alguns campos e opções que, quando selecionados, podem tornar obrigatório o preenchimento de outros campos. Com os dados

dos campos é gerada uma “requisição”. Uma requisição é um pacote que contém informações como, por exemplo, origem, destino, conteúdo da mensagem, e servirá para orientar o sistema para a correta realização do serviço desejado.

Os campos básicos para o criar uma requisição nesta interface são “Destinos”, “Origem” e “Mensagem”. Existem duas opções de serviço nesta interface, que são “Enviar agora” e “Agendar SMS”. O serviço selecionado por padrão é o de Enviar Agora, e exige apenas o preenchimento dos campos básicos. Quando o segundo serviço for selecionado “Agendar SMS”, dois outros campos tornam-se obrigatórios, que são “Data” e “Hora” de envio do SMS para os destinos (ver figura A.6).

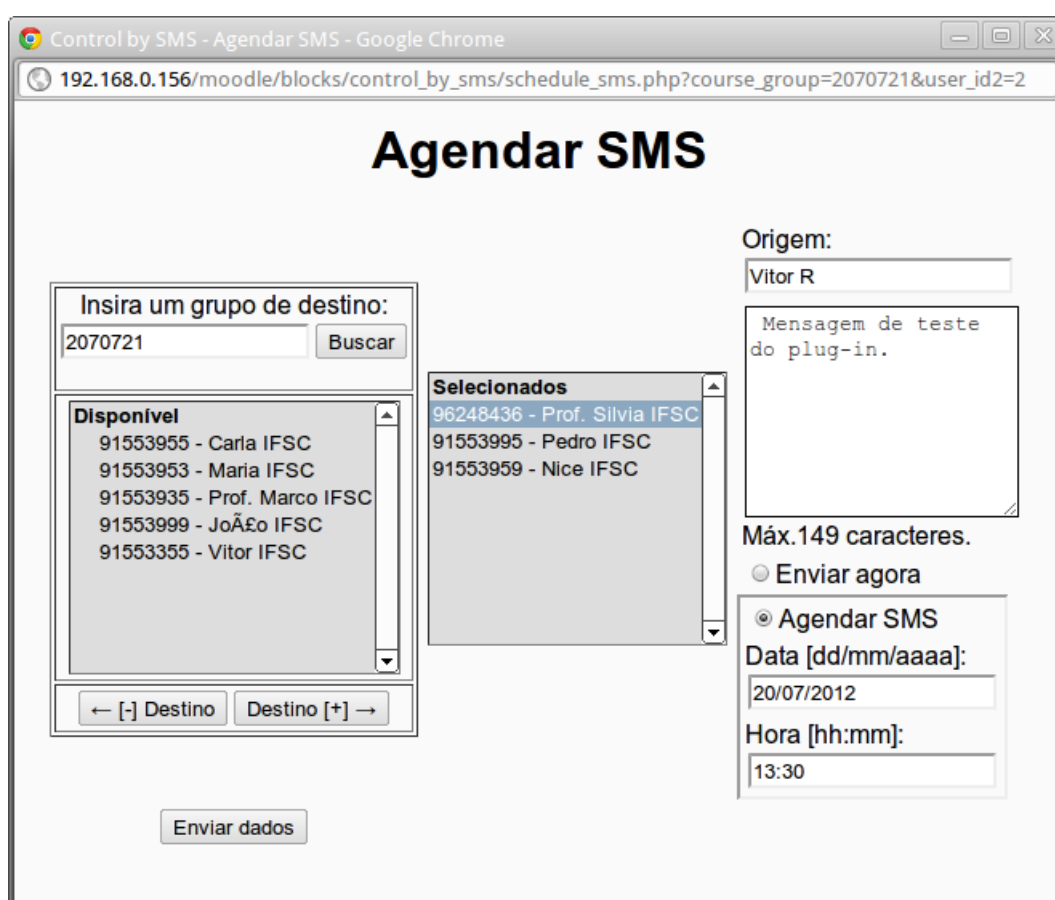


Figura A.6: Interface de agendamento/envio de SMS pelo fórum virtual.

Além dos campos utilizados para adicionar dados à requisição, existe ainda um campo de busca. Este campo é utilizado para localizar destinos para a requisição a ser feita. Dos campos:

- **Campo de Busca** - O *plug-in* permite o envio de SMS apenas para usuários que estiverem cadastrados em grupos. Com o nome/código destes grupos pode-se recuperar uma lista com todos os seus membros. Este campo deve ser preenchido com o nome de um grupo já existente. Quando a opção de busca for selecionada, e o campo estiver preenchido, será

gerada uma lista contendo nome e número de celular de todos os integrantes do grupo especificado;

- **Destinos** - O campo destinos é composto por duas listas, “Disponível” e “Selecionados”, e por opções de adição e deleção entre as listas. Os destinos gerados através do Campo de Busca deverão constar na lista Disponível. Os destinos que realmente serão passados na requisição serão os que estiverem contidos na lista Selecionados. Para trocar os usuários de uma lista para outra, deve-se selecionar um ou mais usuários dentro da lista (utilizando o ponteiro do mouse), e em seguida selecionar a opção de adição ou deleção entre as listas;
- **Origem** - Este campo permite ao usuário inserir uma curta identificação de sua pessoa. Esta identificação será anexada ao conteúdo do SMS quando for enviado;
- **Data** - Necessário somente se a opção Agendar SMS for selecionada - A data em que o SMS deve ser enviado;
- **Hora** - Necessário somente se a opção Agendar SMS for selecionada - A hora em que o SMS deve ser enviado.

Após o correto preenchimento dos campos deve-se selecionar a opção “Enviar dados” para finalizar o preenchimento da requisição e submetê-la ao sistema.

## Relatórios

A interface de relatórios existe para oferecer uma maneira de monitorar as requisições feitas e oferecer dados para monitoramento do uso e performance do sistema (ver figura A.7). É possível ver registros de requisições com o estado “Ativo”, “Falho” e “Enviado”. Todos os estados de requisições possuem os seguintes campos de dados:

- **Fonte** - De qual interface, via Moodle (Web) ou via SMS, a requisição foi enviada para o sistema;
- **Envio** - Data e hora escolhidas para o envio. Caso nenhuma data/hora tenha sido especificada, o campo assume o valor padrão “23:59\_30/12/2030”;
- **Extensão** - Identificação da origem. O número de celular cadastrado para o usuário que criou a requisição;
- **Origem** - A identificação utilizada pelo usuário que foi anexada ao SMS enviado;

- **Mensagem** - O conteúdo do SMS;
- **Destinos** - A lista de destinos selecionados para aquela requisição.



**Relatórios**

[Ativos](#) [Falhos](#) [Enviados](#)

**Ativos**

Fonte	Envio	Extensão	Origem	Mensagem	Destinos
WEB	13:30_20/07/2012	99179331	Vitor R	Mensagem de teste do plug-in.	96248436,91553935,91553953,99179331

Figura A.7: Interface de relatórios.

### Registros (logs)

Os logs são uma maneira de monitorar o estado do *daemon* de serviços. Esta interface oferece acesso direto às mensagens de registros do *daemon* de serviços. Ainda, esta interface oferece a opção de remover os logs antigos (ver figura A.8). O significado das mensagens de logs do *daemon* de serviços não estão diretamente relacionadas ao uso e não serão discutidas neste manual.

## A.5 Requisições por SMS

O Serviço de Mensagens Curtas consiste em oferecer a disponibilidade para o sistema. Através do SMS é possível enviar requisições e utilizar os serviços. O SMS deve ser enviado



Figura A.8: Interface de registros.

para o número do cartão SIM do sistema. Também, o conteúdo do SMS deve estar em um formato específico para que possa ser interpretado pelo sistema.

Os dados transmitidos, tanto nas requisições simples como nas requisições com agendamento, fazem uso do caractere separador “;”. Este caractere é necessário durante o processamento da requisição e divide o espaço das informações contidas na mensagem.

### A.5.1 Requisição simples

Uma requisição simples corresponde ao “Enviar Agora” da interface do bloco de controle no fórum virtual. O formato da mensagem deve ser o seguinte:

“origem;grupo;;mensagem”.

- *origem* - Corresponde ao campo Origem no bloco de controle. Limitado em no máximo 7 (sete) caracteres;
- *grupo* - Corresponde à lista de destinos selecionados no bloco de controle. Ao especi-

ficar um grupo em uma requisição por SMS, todos os membros do grupo receberão a mensagem;

- *mensagem* - A informação que se deseja transmitir. Se forem encontradas ocorrências do caractere separador dentro deste campo, ele será substituído por um espaço em branco.

Exemplo de uma requisição simples:

```
“Pfr Jô;turma205;;A aula será atrasada em meia hora devido ao trânsito na ponte. Realizem a tarefa até eu chegar!!”.
```

Todos os membros do grupo ”turma205” receberiam um SMS contendo o seguinte texto:

```
“Pfr Jô: A aula será atrasada em meia hora devido ao trânsito na ponte. Realizem a tarefa até eu chegar!!”.
```

## A.5.2 Requisição com agendamento

O que difere a requisição simples da requisição com agendamento é a presença do campo data/horário. Este campo pode ser deixado em branco, caracterizando o envio imediato da mensagem (requisição simples), ou pode ser preenchido com a hora do envio, ou a data e a hora do envio, caracterizando o agendamento.

Para apenas agendar o SMS com o horário de envio, sendo a data de envio a atual, segue o formato:

```
“origem;grupo;horário;mensagem”.
```

O formato do campo horário deve ser preenchido com hh:mm (hora:minuto). Um exemplo de requisição com este formato:

```
“Paulo;2070721;12:30;Turma, não se esqueçam de levar a resolução do trabalho para a aula.”.
```

Para agendar uma requisição com data e horário marcado, deve-se seguir o seguinte formato:

```
“origem;grupo;data horário;mensagem”.
```

O formato do campo data deve ser preenchido com dd/mm (dia/mês), sendo a parte do ano preenchida com o ano atual. Um exemplo de requisição com este formato:

```
“Paulo;2070721;15/08 14:00;Não se esqueçam de fazer as anotações para a prova de amanhã.”.
```

A confirmação do correto agendamento da requisição poderá apenas ser vista através do *plug-in* no Moodle.

### A.5.3 Habilitar/Desabilitar o uso

O *plug-in* pode ser habilitado ou desabilitado pelo administrador do fórum virtual. Quando habilitado, suas funcionalidades estarão disponíveis para uso. Quando estiver desabilitado, o *plug-in* apresentará a seguinte mensagem “**Serviço interrompido. Entre em contato com o administrador para mais informações.**”, impossibilitando o uso dos serviços através da interface Web. Porém, ainda serão aceitas as requisições recebidas por SMS.

Para desabilitar o uso do *plug-in* pelo Moodle deve-se primeiro acessar a interface “Visão geral dos Plugins”, no menu de configurações, pela conta de administrador (ver figura A.9). Uma lista deverá aparecer, buscar na lista o *plug-in* a ser configurado. Selecionar a opção “Configurações” nas opções do *plug-in* (ver figura A.10). Uma nova interface será carregada e apresentará uma caixa com a seguinte descrição “Use esta opção para permitir ou negar o uso do serviço de SMS pelo fórum.”. Desmarque esta caixa e salve as alterações para desabilitar os serviços. De forma contrária, marque a caixa e salve as alterações para habilitar os serviços (ver figura A.11).

É possível interromper o serviço completamente desabilitando o *daemon* de serviços. Desta forma, o acesso aos serviços através do fórum virtual ficará impossibilitado, e as requisições recebidas por SMS não serão tratadas.

## A.6 Mensagens do Plug-in

A seguir são listadas as possíveis mensagens de aviso que o *plug-in* pode exibir durante a utilização de algum serviço.

- “A requisição foi registrada com sucesso!”;
- “Falha no processamento da requisição! Se o problema persistir, entre em contato com o administrador do sistema.”;
- “Você não tem permissão para utilizar esta aplicação!”;
- “Conexão com o banco de dados falhou! Entre em contato com o administrador do sistema.”.

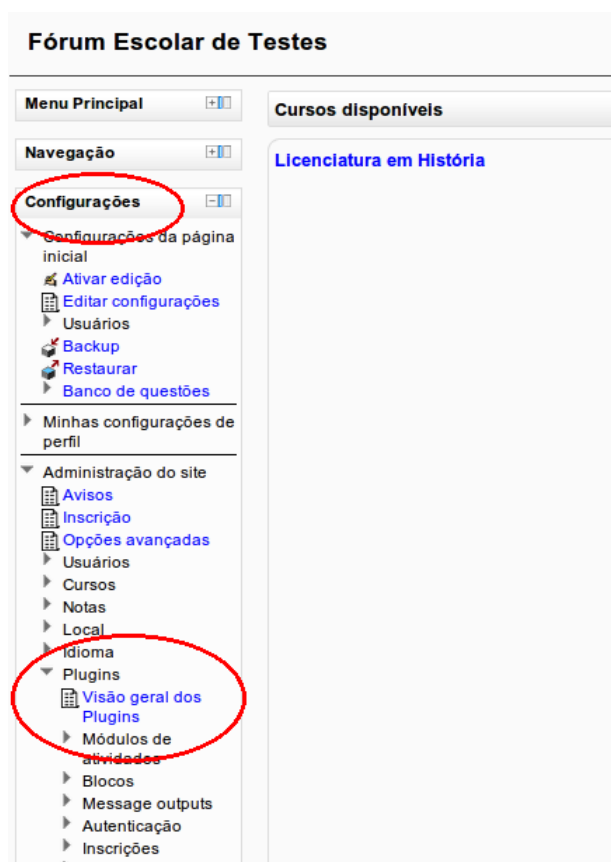


Figura A.9: Acessar interface “Visão geral dos Plugins”.

Nome	Identificador	Tipo	Id	Status	Ações
Comentários	block_comments	Standard	2011112900	Enabled	Desinstal
Localizador da comunidade	block_community	Standard	2011112900	Enabled	Desinstal
Estado de andamento do curso	block_completionstatus	Standard	2011112900	Enabled	Desinstal
Control by SMS	block_control_by_sms	Extension	2011062800	Enabled	Configurações Desinstal
Lista de cursos	block_course_list	Standard	2011112900	Enabled	Configurações Desinstal
Resumo do curso	block_course_overview	Standard	2011112900	Enabled	Desinstal
Descrição do curso/site	block_course_summary	Standard	2011112900	Enabled	Desinstal
Feedback	block_feedback	Standard	2011112900	Disabled	Desinstal
Entrada aleatória do glossário	block_glossary_random	Standard	2011112900	Enabled	Desinstal
HTML	block_html	Standard	2011112900	Enabled	Desinstal

Figura A.10: Acessar configurações do *plug-in*.



The screenshot shows the Moodle administrator interface for the 'Control by SMS' plugin configuration. The breadcrumb trail at the top reads: Home Page > Administração do site > Plugins > Blocos > Control by SMS. A button 'Habilitar Edição de Bloco' is visible in the top right. On the left, there is a navigation sidebar with sections: 'Navegação', 'Favoritos do administrador' (with 'Marcar esta página' link), and 'Configurações' (with 'Minhas configurações de perfil' link). Under 'Configurações', a tree view shows: 'Administração do site' (with sub-items: Avisos, Inscrição, Opções avançadas, Usuários, Cursos, Notas, Local, Idioma), 'Plugins' (with sub-item: Visão geral dos Plugins), 'Módulos de atividades', and 'Blocos' (with sub-items: Gerenciar Blocos, **Control by SMS**, Lista de cursos, Notifications, Usuários Online, PaperCut Print). The main content area is titled 'Control by SMS' and contains a configuration box with the following text: 'Permitir aos professores utilizarem o serviço de SMS?' with a checked checkbox and 'Padrão: Sim'. Below this is the instruction: 'Use esta opção para permitir ou negar o uso do serviço de SMS pelo fórum.' and the parameter 'control\_by\_sms | Open\_to\_use'. At the bottom of this box is a button labeled 'Salvar mudanças', which is circled in red. A red arrow points down to the checkbox.

Figura A.11: Configurações do *plug-in* no Moodle.

## *Lista de Abreviaturas*

**ANATEL** Agência Nacional de Telecomunicações

**CSS** *Cascading Style Sheets*

**ETSI** *European Telecommunications Standards Institute*

**GSM** *Global System for Mobile*

**HTML** *Hyper Text Markup Language*

**HTTP** *Hyper Text Transfer Protocol*

**IP** *Internet Protocol*

**LMS** *Learning Environment System*

**MMS** *Multimedia Messaging Service*

**PBX** *Private Automatic Branch*

**PHP** *PHP: Hypertext Preprocessor*

**RPC** *Remote Procedure Call*

**SGA** Sistema de Gestão da Aprendizagem

**SGC** Sistema de Gestão de Cursos

**SGBD** Sistema Gerenciador de Banco de Dados

**SMS** *Short Message Service*

**SMSC** *Short Message Service Center*

**URL** *Uniform Resource Locator*

**USB** *Universal Serial Bus*

**XML** *Extensible Markup Language*

**XML-RPC** *Remote Procedure Call with XML*

## *Referências Bibliográficas*

- CARDIFF UNIVERSITY, Remote Procedure Calls (RPC). In: \_\_\_\_\_. 2012. Disponível em: <<http://www.cs.cf.ac.uk/Dave/C/node33.html>>. Acesso em: 04 junho. 2012.
- CHATLEY, R. *Predictable Dynamic Plugin Architectures*. [S.l.]: Computing of the University of London, 2005.
- CNET GLOSSARY, Termo "plug-in". In: \_\_\_\_\_. 2012. Disponível em: <<http://www.cnet.com/Resources/Info/Glossary/Terms/plugin.html>>. Acesso em: 07 junho. 2012.
- DEVELOPER'S HOME, SMS Tutorial. In: \_\_\_\_\_. 2012. Disponível em: <<http://www.developershomes.com/sms/atCommandsIntro.asp>>. Acesso em: 11 de junho. 2012.
- EISENBACH, S. et al. *Painless Plugins*. [S.l.]: Department of Computing, Imperial College London, 2003.
- HEUSER, C. *Projeto de Banco de Dados*. [S.l.]: Carlos A. Heuser, 1998.
- IBM. In: \_\_\_\_\_. North Carolina State University, 2012. Disponível em: <<http://www.inf.ufsc.br/poo/smalltalk/ibm/tutorial/oop.html>>. Acesso em: 07 junho. 2012.
- L. MARK. *Programming Python*. [S.l.]: O'Reilly, 2006.
- LAURENT, S. et al. *Programming Web Services with XML-RPC*. [S.l.]: O'Reilly, 2001.
- LOVE, R. *Linux System Programming*. [S.l.]: O'Reilly, 2007.
- MACHADO, F. et al. *Projeto de Banco de Dados - Uma Visão Prática*. [S.l.]: Editora Érica Ltda., 1996.
- MAYER, J. et al. *Lightweight Plug-In-Based Application Development*. [S.l.]: Department of Applied Information Processing, University of Ulm, Helmholtzstr, 2002.
- MEDEIROS, D. et al. *Transmissão de Sequências de Imagens Através da Rede Celular GSM/EDGE*. [S.l.]: IFSC, 2008.
- Python Software Foundation. In: \_\_\_\_\_. 2012. Disponível em: <<http://www.python.org/>>. Acesso em: 25 de junho. 2012.
- SUSE DOC., O processo init. In: \_\_\_\_\_. 2012. Disponível em: <[http://guidalinux.altervista.org/suselinux-manual\\_pt\\_BR-10.1-10/sec.boot.init.html](http://guidalinux.altervista.org/suselinux-manual_pt_BR-10.1-10/sec.boot.init.html)>. Acesso em: 05 junho. 2012.

TANENBAUM, A. et al. *Sistemas Operacionais: Projeto e Implementação*. [S.l.]: Bookman, 2000.

UNICAMP. In: \_\_\_\_\_. Cidade Universitária Zeferino Vaz, Barão Geraldo, Campinas, 2012. Disponível em: <<ftp://ftp.unicamp.br/pub/apoio/treinamentos/bancodados/cursodb.pdf>>. Acesso em: 07 junho. 2012.

vBulletin. In: \_\_\_\_\_. 2012. Disponível em: <[https://www.vbulletin.com/forum/faq.php?faq=vb3\\_board\\_usage#faq\\_vb3\\_forums\\_threads\\_posts](https://www.vbulletin.com/forum/faq.php?faq=vb3_board_usage#faq_vb3_forums_threads_posts)>. Acesso em: 30 maio. 2012.

W3Schools. In: \_\_\_\_\_. 2012. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 25 de junho. 2012.

WILDE, E. et al. *RFC5724 - URI Scheme for Global System for Mobile Communications (GSM) - Short Message Service (SMS)*. [S.l.]: Internet Engineering Task Force (IETF), 2010.

XML-RPC.Com. In: \_\_\_\_\_. 2012. Disponível em: <<http://xmlrpc.scripting.com/spec.html>>. Acesso em: 25 de junho. 2012.