



ISSN 1677-8464

Tradução de Interfaces de Aplicações para Idiomas Distintos

Maria Angelica de Andrade Leite¹
Luiz Manoel Silva da Cunha²
Fábio César da Silva³

Quando se fala em utilização de uma aplicação em vários idiomas tem-se um problema maior denominado internacionalização (Diffuse, 2003). Internacionalização consiste do projeto de uma aplicação de forma que ela possa ser adaptada para várias regiões com línguas e padrões diferentes sem a necessidade de alteração direta da aplicação. O termo é, em geral, abreviado como i18n porque existem 18 letras entre a letra "i" e a letra "n" do termo traduzido *internationalization*. Um projeto que se preocupa com internacionalização possui as seguintes características gerais:

- as mensagens e os componentes da interface não são codificados no programa, sendo armazenados fora do código e recuperados dinamicamente;
- o suporte para novos idiomas não requer recompilação do código;
- dados dependentes de uma cultura como data, moeda e unidades de medidas aparecem em formatos que são compatíveis com o utilizado regionalmente.

A preocupação com aspectos de internacionalização surgiu no projeto do banco de dados de experimentos em manejo de fertilizantes para cana-de-açúcar (Leite et al., 2003) que está sendo desenvolvido no âmbito do convênio existente entre a Embrapa e o Instituto Internacional do Potássio - IPI (Embrapa Informática Agropecuária, 2003). O banco de dados possui

uma interface que pode ser acessada via World Wide Web, através de páginas HTML, com o objetivo de que possa ser utilizado por pessoas em diferentes lugares. Inicialmente, o banco de dados estava sendo desenvolvido para armazenar dados de experimentos realizados no Brasil. No decorrer do projeto, o IPI requisitou que a aplicação desenvolvida pudesse ser utilizada, também, por pesquisadores em outros países e que houvesse uma versão da interface da aplicação em inglês. A partir desta requisição foi elaborado um esquema de tradução automática das páginas HTML para uma língua específica.

No projeto do banco de dados o aspecto relativo à internacionalização abrange a parte referente à tradução das telas de entrada de dados do usuário. Na especificação do sistema decidiu-se adotar um formato de data único que obedece ao esquema dd/mm/aaaa onde "d" refere-se ao dia, "m" refere-se ao mês e "a" refere-se ao ano. Para as unidades de medidas foi adotado o padrão internacional de medidas (Unidades..., 2003).

Este comunicado apresenta o esquema de tradução proposto para as telas de entrada de dados do banco de dados de experimentos em manejo de fertilizantes para cana-de-açúcar. Inicialmente, será apresentada uma visão geral de duas propostas para internacionalização encontradas na literatura para a linguagem PHP (The PHP Group, 2003) que é

¹ M.Sc. em Ciência da Computação, Pesquisadora da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: angelica@cnptia.embrapa.br)

² M.Sc. em Ciência de Computação e Matemática Computacional, Técnico de Nivel Superior da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: luizm@cnptia.embrapa.br)

³ Doutor em Solos e Nutrição de Plantas, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: fcesar@cnptia.embrapa.br)

a utilizada no desenvolvimento da aplicação de manipulação do banco de dados. Em seguida será mostrado o esquema adotado no projeto e, ao final do documento, serão apresentadas algumas considerações sobre o esquema adotado e os pontos de melhoria para outras versões.

Propostas de internacionalização de aplicações

Este tópico apresenta uma visão geral de duas propostas, encontradas na literatura, para tradução de telas e mensagens utilizando a linguagem PHP: catálogo de mensagens e utilitários *Gettext*. Não se pretende esgotar o assunto mas prover as idéias gerais e referências para que os interessados tenham um ponto de partida para obter mais informações.

Catálogo de mensagens

A primeira proposta utiliza um catálogo de mensagens (Trachtenberg, 2003) especificado como o vetor *mensagens* que contém a sentença a ser traduzida associada a sua respectiva tradução de acordo com o idioma especificado. A função `msg()` tem como parâmetro de entrada a sentença a ser traduzida e retorna o rótulo do catálogo em função do idioma fornecido. Um exemplo desta proposta é o catálogo de frutas ilustrado a seguir:

```
< ?php
$mensagens = array (
  'pt_BR' => array(
    'Minhas frutas favoritas sao: ' => 'Minhas frutas favoritas são: ',
    'morango' => 'Morango',
    'pessego' => 'Pêssego',
    'abacaxi' => 'Abacaxi',
    'laranja' => 'Laranja'
  ),
  'en_US' => array(
    'Minhas frutas favoritas sao: ' => 'My favorite fruits are: ',
    'morango' => 'Strawberry',
    'pessego' => 'Peach',
    'abacaxi' => 'Pineapple',
    'laranja' => 'Orange'
  )
);

function msg($s) {
  global $LANG;
  global $mensagens;

  if (isset($mensagens[$LANG][$s])) {
    return $mensagens [$LANG][$s];
  }else {
    error_log("Erro:LANG:" .
      "$lang,message:'$s'");
  }
}
?>
```

O programa `frutas.php`, a seguir, utiliza o catálogo de mensagens para imprimir uma lista de frutas através de chamadas à função `msg()`. No caso o idioma escolhido é o inglês.

```
< ?php
$LANG = 'en_US';

print msg('Minhas frutas favoritas sao:')." \n";
print msg('morango')." \n";
print msg('abacaxi')." \n";
print msg('laranja')." \n";
?>
```

```
My favorite fruits are:
Strawberry
Pineapple
Orange
```

O catálogo de mensagens ilustrado oferece uma solução simples de tradução de sentenças para diversos idiomas. Pode ser implementado também utilizando a orientação a objetos.

Uso dos utilitários *Gettext*

O projeto GNU (Free Software Foundation, 2003b) possui um conjunto de ferramentas para produzir internacionalização de aplicações denominado *gettext* (Argerick, 2003; FutureQuest, 2003; Free Software Foundation, 2003a). Este conjunto foi desenvolvido para aplicações nas linguagens C e C++ mas ele pode ser utilizado para outras linguagens tal como o PHP. O conjunto de ferramentas *gettext* está disponível tanto para ambientes UNIX quanto WINDOWS. Para utilizar *gettext*, em scripts PHP, as mensagens a serem traduzidas devem ser modificadas colocando-se o símbolo *underscore* (`_`) ou *gettext*. O exemplo do catálogo de mensagens, o arquivo `frutas.php`, fica como mostrado a seguir:

```
< ?php
print(_("Minhas frutas favoritas sao:"))." \n";
print(_("morango"))." \n";
print(_("abacaxi"))." \n";
print(_("laranja"))." \n";
?>
```

Neste caso o símbolo *underscore* (`_`) funciona como um alias para a função *gettext*. Uma vez que as mensagens estão no formato requerido o utilitário *gettext* oferece uma série de comandos para tratar estas mensagens. O comando *xgettext* captura estas mensagens em um arquivo texto do tipo `.po` que é um arquivo de mensagens portátil. Este arquivo contém a mensagem original mais um espaço reservado para se colocar a tradução requerida. No exemplo das frutas o comando `"xgettext -d mensagens frutas.php"` vai gerar o arquivo *messages.po* dado por:

```
#frutas.php
domain "messages"
msgid "Minhas frutas favoritas sao:"
msgstr "My favorite fruits are:"
msgid "morango"
msgstr "Strawberry"
msgid "abacaxi"
msgstr "Pineapple"
msgid "laranja"
msgstr "Orange"
```

As entradas dadas por `"msgid"` vão conter as sentenças extraídas do código e as entradas `"msgstr"` são geradas pelo comando *xgettext* e devem ser utilizadas para armazenar a tradução das sentenças como ilustrado considerando o idioma inglês. Para cada idioma deve ser gerado um arquivo

específico. A partir do arquivo traduzido deve ser gerado um arquivo binário do tipo .mo através do comando "*msgfmt -o messages.mo messages.po*" do conjunto de utilitários. Após esta etapa, ambos os arquivos .po e .mo são considerados no processo de tradução e devem ser armazenados em diretórios denominados LC_MESSAGES associados ao seu idioma. Uma sugestão de organização de diretórios destes arquivos é por meio da criação de um diretório "/locale/< idioma >/LC_MESSAGES/< arquivos .po e .mo >" como ilustrado a seguir:

```
/fontes
/locale/en/LC_MESSAGES/messages.po
                messages.mo
                es/LC_MESSAGES/messages.po
                Messages.mo
```

Existe o padrão ISO 639.2 para descrever as iniciais de cada idioma (Library of Congress, 2003). Neste padrão as iniciais "en" significam inglês e as iniciais "es" significam espanhol. Para que ocorra a tradução, utilizando os arquivos gerados, o programa php original deve ser adaptado através da atribuição de valores a algumas variáveis de ambiente que vão indicar o idioma a ser considerado e o diretório contendo os respectivos arquivos traduzidos como ilustrado, com o arquivo frutas.php, a seguir:

```
< ?php
// Associa o idioma inglês
putenv ("LC_ALL=en");

// Especifica a localização dos arquivos de tradução
bindtextdomain ("messages", "../locale");

// Associa um domínio
textdomain ("messages");

print_("Minhas frutas favoritas sao:")."\n";
print_("morango")."\n";
print_("abacaxi")."\n";
print_("laranja")."\n";

?>
```

Esquema de tradução adotado no banco de dados

Dentro do projeto de banco de dados de manejo de fertilizantes adotou-se o catálogo de mensagens por ser a solução mais simples. O catálogo de mensagens foi organizado em função das telas de interface do sistema e do idioma escolhido. Em linhas gerais, em sistemas que envolvem inserção de dados em um banco de dados, as informações a serem inseridas são organizadas em tabelas e para cada tabela a aplicação deve prover quatro funções básicas: inserção, alteração, consulta e exclusão dos dados na tabela. No banco de dados de manejo de fertilizantes, para cada uma destas 4 funções, são construídas 2 telas de interface com o usuário: a primeira tela captura os dados e a segunda tela indica o sucesso ou não da operação. Dado que o banco possui 20 tabelas então vão existir, aproximadamente, um total de "20 tabelas x 4 funções x 2 = 160" telas de interface com o usuário. No esquema de tradução proposto foi criada a função "Traducao" que tem como parâmetros de entrada o nome da tela a ser traduzida e o idioma desejado. A função Traducao() retorna um vetor com os rótulos traduzidos referentes à tela e idioma

fornecidos. O exemplo abaixo mostra a função Traducao() considerando o programa da tela para inclusão de projeto FincProjeto.php, onde "F" significa formulário de captura de dados do usuário, "Inc" significa função de inclusão e "Projeto" denota a tabela projeto.

```
< ?php
function Traducao($idioma, $tela){

    if ($idioma == "pt") {
        switch ($tela) {
            case "FincProjeto":
                return (
                    $rotulos = array(
                        "tituloForm" => "Base de Dados de Experimentos de Cana-
                            de-açúcar: Inclusão de Projeto",
                        "tituloNavegacao" => "Configuração -> Projeto ->
                            Incluir",
                        "tituloTela" => "Identificação do Projeto",
                        "obrigatorio" => "Campo Obrigatório",
                        "titulo" => "Título",
                        "descricao" => "Descrição",
                        "responsavel" => "Responsável",
                        "incluir" => "Incluir",
                        "limpar" => "Limpar",
                        "mensagemInsucesso" => "Problema na inclusão do
                            projeto.",
                        "contactar" => "Contactar o administrador.",
                        "erro" => "ERRO:",
                        "voltar" => "Voltar",
                        "ajuda" => "Ajuda");
                    break;
                }
            }
        else if ($idioma == "en") {
            switch ($tela) {
                case "FincProjeto":
                    return (
                        $rotulos = array(
                            "tituloForm" => "Sugar Cane Trial Data Base: Project
                                Insertion",
                            "tituloNavegacao" => "Configuration -> Project ->
                                Insert",
                            "tituloTela" => "Project Identification",
                            "obrigatorio" => "Required Information",
                            "titulo" => "Title",
                            "descricao" => "Description",
                            "responsavel" => "Responsible",
                            "incluir" => "Insert",
                            "limpar" => "Clear",
                            "mensagemInsucesso" => "Problem in project insertion",
                            "contactar" => "Contact the administrator",
                            "erro" => "ERROR:",
                            "voltar" => "Back",
                            "ajuda" => "Help");
                        Break;
                    }
                }
            }
        }
    }
?>
```

Dentro do código da interface a chamada da função Traducao() é feita através do comando *include*. O conjunto de rótulos retornados pela função Traducao() para a tela fornecida como parâmetro de entrada é armazenado no vetor "rotulos".

```
< ?php
$idioma = $_SESSION['idioma'];
include ("../libCana/Traducao_inc.php");
$rotulos = Traducao($idioma, "FincProjeto");
?>
```

As mensagens da interface são impressas através de chamadas ao vetor de rótulos como ilustrado no código

a seguir que imprime a linha com o título do projeto na tabela com dados de projeto:

```
<tr>
  <td width="250"> <font class="fonteElemento"> <?echo
    $rotulos["titulo"];?>
    <span class="fonteObrigatorio"> * </span> </font> </td>
  <td width="430"> <textarea name="titprojeto" cols="75"
    rows="2"> </textarea> </td>
</tr>
```

O resultado da tradução para a tela de inserção de dados de projeto pode ser visto nas Fig. 1 e 2. A Fig. 1 ilustra a tela de inclusão de dados de projeto para o idioma português e a Fig. 2 ilustra a mesma tela para o idioma inglês.

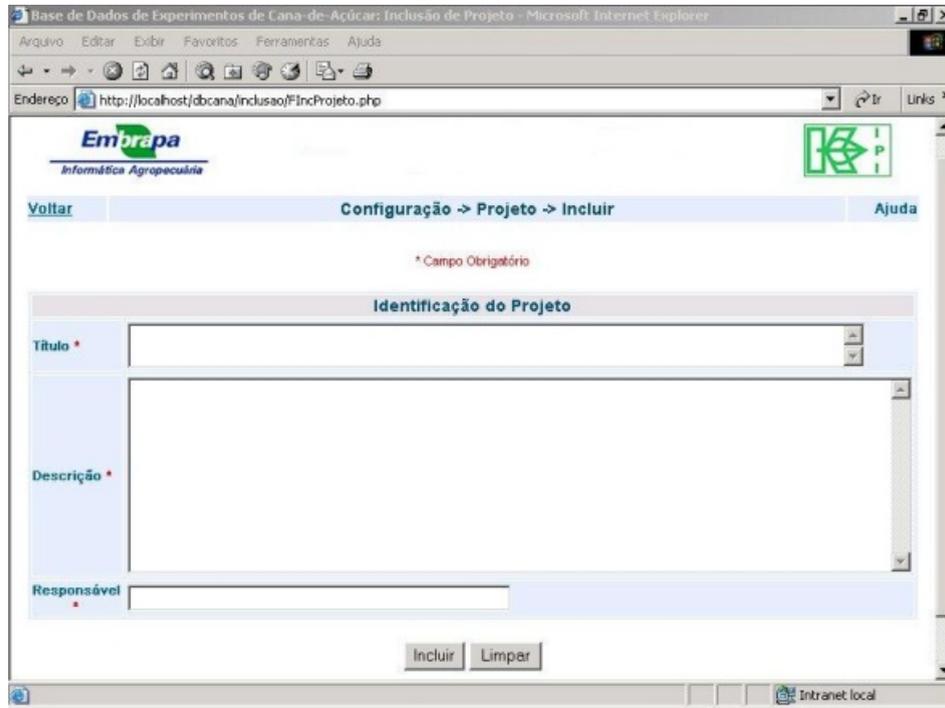


Fig. 1. Tela de inclusão de projeto em português.

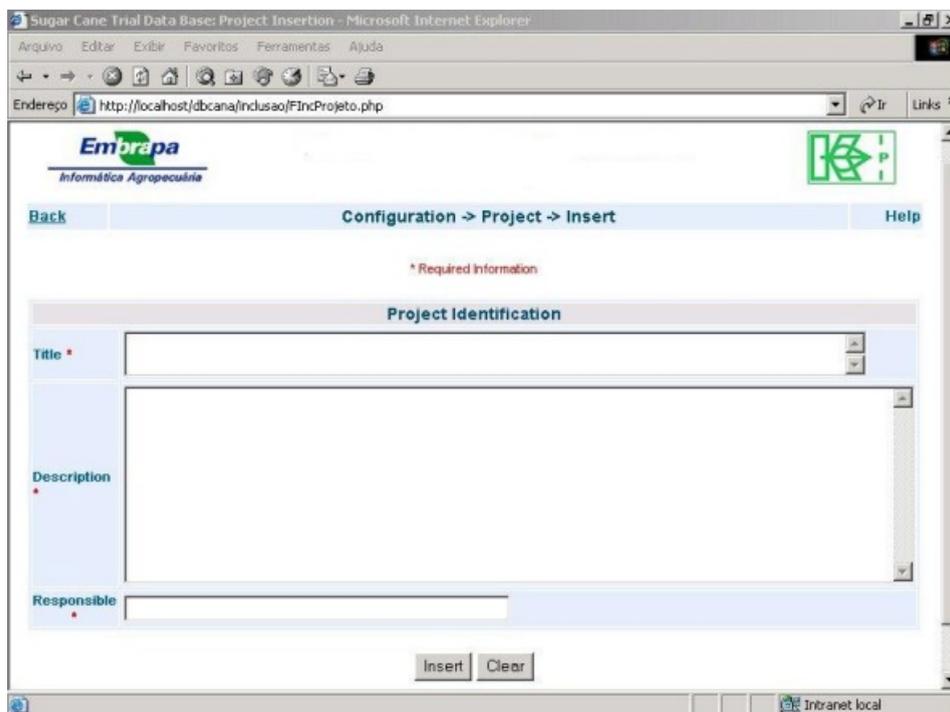


Fig. 2. Tela de inclusão de projeto em inglês.

Considerações finais

Este comunicado apresentou a experiência de tradução de interfaces adquirida no desenvolvimento da aplicação de manipulação do banco de dados de experimentos agrícolas em manejo de fertilizantes. Por se tratar de um sistema de pequeno porte adotou-se o esquema de catálogo de mensagens visando entender melhor os aspectos envolvidos no problema de tradução. O catálogo de mensagens foi construído através de um vetor de rótulos traduzidos para cada tela do sistema num total aproximado de 160 telas, conforme mostrado no tópico "Esquema de tradução adotado no banco de dados". Durante o desenvolvimento do sistema observou-se que, para uma mesma tabela, os rótulos que se referem aos dados armazenados na tabela se repetem nas telas de funções de inserção, alteração, consulta e exclusão. Assim, da forma como o catálogo está organizado, existe uma grande duplicação de rótulos uma vez que o rótulo de um mesmo dado vai ser considerado em cada uma das telas de funções do banco num total de 160 entradas no vetor da função Traducao(). Uma melhoria que pode ser feita é não tratar cada tela de função em separado mas tratar cada tabela como um componente do vetor tradução. Desta forma vai haver uma economia na tradução dos rótulos dos dados das tabelas que será realizada uma única vez. Uma outra melhoria é detectar quais os rótulos são comuns em todas, ou na maioria, das telas para que sejam considerados como uma única entrada no catálogo como por exemplo rótulos relativos às ações de voltar, ajuda, limpar, confirmar, etc. Considerando a proposta de melhoria, o vetor de rótulos da função Traducao() vai trabalhar com 21 entradas (20 tabelas + 1 entrada de rótulos comuns). Cada uma das entradas de tabelas do vetor irá conter os rótulos específicos para cada função de manipulação dos dados do banco (inserção, alteração, consulta e exclusão) mais os rótulos referentes aos dados manipulados na tabela. A entrada de rótulos comuns vai ser utilizada nas telas do sistema para todas as tabelas.

Uma solução que deve ser considerada na próxima versão do sistema, como também em novos sistemas a serem desenvolvidos, é o uso dos utilitários *gettext* ao invés do catálogo de mensagens. O trabalho inicial é marcar todos os rótulos do texto com o *underscore*, símbolo do *gettext*. A partir disto o utilitário possui funções que extraem automaticamente os rótulos do código, para montar o catálogo, tirando esta função da responsabilidade do desenvolvedor. Um ponto favorável do *gettext* é que um mesmo rótulo é considerado apenas uma vez evitando a duplicação de rótulos que apareçam em mais de uma tela. Para aplicações com um grande número de telas o esquema de tradução, utilizando o conjunto de utilitários *gettext*, é o mais recomendado na literatura. Em função das facilidades oferecidas acredita-se que os utilitários *gettext* possam ser úteis mesmo para aplicações de pequeno porte como banco de dados de experimentos em manejo de fertilizantes. Assim, recomenda-se que quando for necessário desenvolver uma aplicação com características de internacionalização seja considerada a utilização dos utilitários *gettext* como solução.

Atualmente existe o paradigma de programação orientada a aspectos - AOP (The Aspect-Oriented Software Development, 2004; Elrad et al., 2001) que pretende facilitar a separação, modularização e implementação de vários aspectos, intrínsecos aos sistemas de programação, que

ficam espalhados dentro do código das classes de objetos dos sistemas. Exemplos de aspectos desta natureza são: internacionalização, integridade de transações, autenticação, segurança e desempenho, entre outros. As atuais implementações de AOP estão voltadas para as linguagens Java, C e C++ . Acredita-se que, com a evolução deste paradigma, outras linguagens também serão englobadas e pode ser uma nova solução para o problema de internacionalização das aplicações.

Referências bibliográficas

ARGERICK, L. Internationalization using PHP and GetText. Disponível em: <<http://zez.org/article/articleview/42/0/>> . Acesso em: 03 dez. 2003.

THE ASPECT-ORIENTED SOFTWARE ASSOCIATION. Aspect-Oriented Software Development [home page]. Disponível em: <<http://aosd.net/>> . Acesso em: 12 fev. 2004.

DIFFUSE. **Guide to internationalization/localization**. Disponível em: <<http://www.diffuse.org/IntLocGuide.html>>. Acesso em: 03 dez. 2003.

ELRAD, T.; FILMAN, R. E.; BADER, A. Aspect-oriented programming: introduction. Communications of the ACM, v. 44, n. 10, p. 29-32, Oct. 2001.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. Projeto Brasil - Suporte para Aplicação Balanceada de Adubação Potássica. Disponível em: <<http://www.cnptia.embrapa.br/projetos/ipi/ipi/homepage/index.htm>> . Acesso em: 04 dez. 2003.

FREE SOFTWARE FOUNDATION. GNU gettext utilities. Disponível em: <http://www.gnu.org/software/gettext/manual/html_mono/gettext.html> . Acesso em: 03 dez. 2003a.

FREE SOFTWARE FOUNDATION. GNU's not Unix! - the GNU Project and the Free Software Foundation (FSF). Disponível em: <<http://www.gnu.org/>> . Acesso em: 04 dez. 2003b.

FUTUREQUEST INC. PHP + gettext, internationalization of your website, tutorial. Disponível em: <<http://www.aota.net/forums/printthread.php?threadid=10615>> . Acesso em: 03 dez. 2003.

LEITE, M. A. de A.; SILVA, F. C. da; CUNHA, L. M. S. Banco de dados de experimentos em manejo de fertilizantes para cana-de-açúcar. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA (SBI-AGRO), 4., 2003, Porto Seguro. Trabalhos apresentados... Lavras: SBRI-AGRO, 2003. 4 p. Sessão 7 - Aplicação na Internet.

LIBRARY OF CONGRESS (ESTADOS UNIDOS). ISO 639.2 - registration authority - codes for the representation of names of languages. Disponível em: <[Http://www.loc.gov/standards/iso639-2/langcodes.html](http://www.loc.gov/standards/iso639-2/langcodes.html)> . Acesso em: 03 dez. 2003.

THE PHP GROUP. PHP: Hypertext Preprocessor. Disponível em: < <http://www.php.net/> > . Acesso em: 04 dez. 2003.

TRACHTENBERG, A. ONLamp.com: internationalization and localization with PHP. In: O'REILLY MEDIA. ONLamp.com: PHP DevCenter. Disponível em: < http://www.onlamp.com/pub/a/php/2002/11/28/php_i18n.html > . Acesso em: 03 dez. 2003.

UNIDADES e símbolos SI. In: ANDRADE, J. C. de; CUSTODIO, R. Chemkeys - o seu site de química!!! Disponível em: < http://www.chemkeys.com/bra/ag/uec_7/sidu_4/uess_6/uess_6.htm > . Acesso em: 03 dez. 2003.

Comunicado Técnico, 54

Ministério da Agricultura, Pecuária e Abastecimento

Governo Federal

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-970 - Campinas, SP
Fone: (19) 3789-5743
Fax: (19) 3289-9594
e-mail: sac@cnptia.embrapa.com.br

1ª edição on-line - 2003

Ó Todos os direitos reservados.

Comitê de Publicações

Presidente: *Luciana Alvim Santos Romani*
Membros Efetivos: *Carla Geovana Macário, José Ruy Porto de Carvalho, Marcia Izabel Fugisawa Souza, Marcos Lordello Chaim, Suzilei Almeida Carneiro.*
Suplentes: *Carlos Alberto Alves Meira, Eduardo Delgado Assad, Maria Angelica Andrade Leite, Maria Fernanda Moura, Maria Goretti Gurgel Praxedis.*

Expediente

Supervisor editorial: *Ivanilde Dispatto*
Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*
Editoração eletrônica: *Área de Comunicação e Negócios*