

SiGE - Sistema de Gerência de Eventos

Equipe COMSOLiD

Instalação e Configuração

Programas necessários:

- PostgreSQL;
- Apache HTTP Server: `sudo apt-get install apache2`;
- php5: `sudo apt-get install php5 php5-pgsql libapache2-mod-php5`;
 - extensão intl: `sudo apt-get install php5-intl`;
 - extensão GD: `sudo apt-get install php5-gd`;
- Zend Framework;
- git (opcional);

Base de dados

Schema da Base de dados

A instalação da base de dados é feita pelo arquivo `ddl-schema.sql`. Abra o arquivo e defina alguns parâmetros:

Encoding do servidor

```
SET client_encoding = 'LATIN1';
```

ou

```
SET client_encoding = 'UTF8';
```

Permissão ao usuário do banco de dados

```
REVOKE ALL ON SCHEMA public FROM PUBLIC;  
REVOKE ALL ON SCHEMA public FROM postgres;  
GRANT ALL ON SCHEMA public TO postgres;  
GRANT ALL ON SCHEMA public TO PUBLIC;
```

Modifique `postgres` para seu usuário.

Note que o script possui `START TRANSACTION;` e `ROLLBACK;`. Faça um teste inicial e execute o script para se certificar que tudo irá correr bem. Por fim substitua o comando `ROLLBACK;` por `COMMIT;` e execute novamente o script.

Observação: Para PostgreSQL 9.2 para trás comentar a linha 12:

```
SET lock_timeout = 0;
```

Para isso basta colocar `--` no início, dessa forma:

```
--SET lock_timeout = 0;
```

Esse comando faz parte das versões 9.3 para frente, mas não prejudica o uso de versões mais antigas.

Dados iniciais do sistema

A inserção dos dados iniciais pode ser encontrada em `ddl-dados-iniciais.sql`. Modifique as tabelas `estado`, `instituicao`, `municipio` e `sala` de acordo com sua necessidade.

obs.: para base de dados que usam codificação LATIN1 utilize o script `ddl-dados-iniciais-latin1.sql`.

As outras tabelas já estão devidamente preparadas.

Teste a execução de script e remova `START TRANSACTION;` e `ROLLBACK;`.

Criando um novo encontro

O primeiro passo para criar um encontro, e adicionar um registro na tabela `encontro` da seguinte forma:

```
INSERT INTO encontro(nome_encontro, apelido_encontro, data_inicio, data_fim,
    periodo_submissao_inicio, periodo_submissao_fim)
VALUES ('I Encontro de Software Livre', 'I ESL', '2013-11-07', '2013-11-09',
    '2013-05-01', '2013-11-06');
```

obs.: a coluna `ativo` será removida em breve.

Depois verifique o `id_encontro` gerado e crie dois registros na tabela `mensagem_email`, um para cada mensagem de `tipo_mensagem_email`:

```
INSERT INTO mensagem_email(id_encontro, id_tipo_mensagem_email,
    mensagem, assunto, link)
VALUES (1, 1, 'Nome: {nome}, E-mail: {email}, Senha: {senha},
    <a href="{href_link}" target="_blank">Clique aqui</a>',
    'I ESL - Cadastro Encontro',
    'http://www.esl.org/login');
```

obs.: Note que a `mensagem` traz elementos dentro de {}. Eles são utilizados no PHP para substituir valores reais, tornando a mensagem dinâmica.

Vale lembrar que a mensagem pode ser escrita em HTML. Coloque apenas tags referentes ao `body`.

Outro ponto importante é configurar o `encoding` ao inserir um encontro e suas mensagens. Para isso adicione `SET client_encoding = 'LATIN1'`; no início do `insert`.

Da mesma forma crie a mensagem de recuperação de senha:

```
INSERT INTO mensagem_email(id_encontro, id_tipo_mensagem_email,
    mensagem, assunto, link)
VALUES (1, 2, 'Nome: {nome}, E-mail: {email}, Senha: {senha},
    <a href="{href_link}" target="_blank">Clique aqui</a>',
    'I ESL - Recuperar Senha',
    'http://www.esl.org/login');
```

Por ser um exemplo, as mensagens ficaram uma muito parecida com a outra. Você deve adaptar de acordo com seu encontro.

obs.: esse passo serve somente para o primeiro encontro. Os demais podem ser criados a partir do SiGE em `/adim/encontro/criar/`.

SiGE

Zend

A versão utilizada pelo SiGE é [Zend 1.12.9](#).

A instalação é bem simples. Basta copiarmos o Zend para um diretório de bibliotecas do sistema. Baixe o pacote Full, descompacte e siga as instruções em um terminal:

```
$ sudo su
# mv ZendFramework-1.12.9 /usr/local/lib
# cd /usr/local/lib
# ln -s ZendFramework-1.12.9 zend
```

Baixando SiGE do Github para desenvolvimento

Para realizar clone da última versão do SiGE:

```
$ git clone https://github.com/comsolid/sige.git
```

obs.: é necessário instalar o git. No Ubuntu podemos instalar através do comando:

```
$ sudo apt-get install git
```

obs.: instale a partir do repositório somente se você está interessado em contribuir, estudar o código ou apenas testando.

Baixando versão estável SiGE

Procure pela versão mais atual do SiGE em:

<https://github.com/comsolid/sige/releases>

Renomeie a pasta para **sige** caso necessário.

Configurando VirtualHost

Para simular um host no mundo real que utiliza Zend precisamos criar um VirtualHost no apache. Com o Apache devidamente instalado, crie um arquivo em `/etc/apache2/sites-enable/` chamado **sige**. Nele copie o seguinte conteúdo, modificando conforme necessidade:

```
<VirtualHost *:80>
    ServerName sige.local

    DocumentRoot /var/www/sige/public
    <Directory "/var/www/sige/public">
        AllowOverride All
    </Directory>
</VirtualHost>
```

Adicionaremos o `ServerName` ao `/etc/hosts`:

```
127.0.0.1      localhost
# adicione a linha abaixo
127.0.0.1      sige.local
```

Obs.: este trecho deve ser feito apenas em ambiente de desenvolvimento.

Habilite o mod `rewrite` do apache: `$ sudo a2enmod rewrite`.

Reinicie o Apache: `$ sudo service apache2 restart`.

Instalar o Zend no SiGE

A instalação é bem simples, apenas crie um link simbólico dentro do diretório do projeto (daqui para frente chamado de `${SiGE}`) em `${SiGE}/library`:

```
$ sudo su
# cd /var/www/sige/library
# ln -s /usr/local/lib/Zend/library/Zend
```

Observação: Caso você não tenha permissão para instalar bibliotecas no sistema, por exemplo se você contratou um serviço externo, você pode copiar o Zend diretamente na pasta `library` do SiGE. Faça:

```
$ cp -R /caminho/para/ZendFramework1.12.9/library/Zend ${SiGE}/library
```

Caso use FTP suba o diretório para o mesmo local.

Permitir escrita para HTMLPurifier e Captcha

É necessário dar permissão total a dois diretórios, faça:

```
$ cd ${SiGE}/library/HTMLPurifier/DefinitionCache/
$ mkdir Serializer
$ chmod 777 Serializer/
$ cd ${SiGE}/public/
$ mkdir captcha
$ chmod 777 captcha/
```

Configurar SiGE para ambiente de produção

```
$ cd ${SiGE}/public
$ nano .htaccess
```

Mude `SetEnv APPLICATION_ENV development` para `SetEnv APPLICATION_ENV production`.

Configurar conexão com base de dados

Com o projeto configurado vamos editar os parâmetros de conexão com o PostgreSQL. Dentro do diretório do projeto abra o arquivo `${SiGE}/application/configs/application.ini`, copie os dados para a seção `[production]` (cole abaixo de `autoloaderNamespaces[] = "Sige"`) e edite os parâmetros abaixo:

```
resources.db.params.host      = "localhost"
resources.db.params.dbname    = "database"
resources.db.params.username  = "postgres"
resources.db.params.password  = "**secret**"
```

Configurar SMTP para envio de e-mail

Temos também que configurar o envio de e-mail para validar participantes, recuperação de senhas, etc. Ainda no arquivo `${SiGE}/application/configs/application.ini`, copie o trecho a seguir e cole logo abaixo dos dados da conexão com o banco. Edite o trecho:

```
resources.mail.transport.type = "smtp"; não precisa editar
resources.mail.transport.host = "smtp.esl.com"; servidor gmail: smtp.gmail.com
resources.mail.transport.port = "587"; 465
resources.mail.transport.ssl  = "tls"
resources.mail.transport.auth = "login"; não precisa editar
resources.mail.transport.username = "esl@esl.org";
resources.mail.transport.password = "**secret**"
resources.mail.transport.register = true; True by default
resources.mail.defaultFrom.email  = "esl@esl.org"
resources.mail.defaultFrom.name    = "I ESL"
resources.mail.defaultReplyTo.email = "esl@esl.org"
resources.mail.defaultReplyTo.name = "I ESL"
```

obs.: para a linha `resources.mail.transport.port`:

- 587 is the Outgoing server (SMTP) port for IMAP. It uses a TLS encryption connection.
- 587 é a porta de Saída de serviço (SMTP) para IMAP. Ela usa conexão TLS criptografada.
- 465 is the Outgoing server (SMTP) port for POP. It uses an SSL encryption connection.
- 465 é a porta de Saída de serviço (SMTP) para POP. Ela usa conexão SSL criptografada.

Mais detalhes em [Zend_Mail](#).

Configurar Encontro

Após criar um encontro no banco de dados, temos um `id_encontro`. No arquivo `${SiGE}/application/configs/application.ini` edite a linha:

```
encontro.codigo = 1
```

Observação: Mude o valor a cada novo encontro.

Crie o primeiro usuário administrador

Abra o SiGE no navegador e crie um usuário. Se tudo der certo um e-mail com uma senha padrão foi enviado para você. Tente fazer um login.

No banco de dados, na tabela `pessoa`, modifique a coluna `administrador` para `true`.

Tradução (i18n)

Para traduzir as mensagens do SiGE você deve editar o arquivo `${SiGE}/application/Bootstrap.php`. Na função `_initTranslate` edite a linha:

```
$locale = "pt_BR";
```

Verifique as traduções disponíveis em `${SiGE}/application/langs`. Coloque em `$locale` o mesmo nome do diretório da sua língua padrão.

Para traduzir as mensagens que estão nos arquivos Javascript abra o arquivo `${SiGE}/application/layouts/scripts/twbs3.phtml`.

Procure a linha:

```
$this->headScript()->prependFile($this->baseUrl('js/jed/locale/pt_BR.js'));
```

e mude para a língua desejada. As opções estão em `${SiGE}/public/js/jed/locale`.

Certificados

Os arquivos relativos aos certificados, participante e palestrante, ficam localizados em `${SiGE}/public/img/certificados/`. Lá teremos um diretório `default/` que contém arquivos iniciais para que um certificado possa ser gerado sem nenhuma configuração.

Para criar certificados para um determinado encontro devemos criar um diretório em `/${SiGE}/public/img/certificados/` com o `id_encontro` do encontro. Por exemplo, se `id_encontro` for 1, criaremos o diretório `/${SiGE}/public/img/certificados/1/`.

Utilize os arquivos `modelo.svg` e `assinatura-modelo.svg` dentro de `/${SiGE}/public/img/certificados/default/` como modelos para a criação de seus certificados. Eles possuem as marcações e os tamanhos específicos. Os tamanhos são:

Arquivo	Tamanho
<code>modelo.svg</code>	1052x744 (mesmo tamanho de uma folha A4 paisagem)
<code>assinatura-modelo.svg</code>	250x140

obs.: o arquivo `modelo.svg` possui camadas para que ao trabalhar em cima do molde você não se atrapalhe com outros objetos. Usando Inkscape acesse as camadas com o comando *Shift + Ctrl + L*.

Após finalização do modelo, exporte o arquivo para **JPG**. Como o Inkscape não exporta diretamente para essa extensão, utilize o GIMP para essa tarefa.

O certificado do SiGE suporta até três assinaturas. Para isso você deve exportar o arquivo de assinatura da seguinte forma: **assinatura-1.png** - para que a assinatura apareça a esquerda do certificado, **assinatura-2.png** - para que a assinatura apareça no centro e **assinatura-3.png** - para que apareça a direita.

As assinaturas são opcionais e podem ser usadas da maneira que você deseje. Por exemplo, se você possui um certificado que tenha algum detalhes no centro você pode optar por criar apenas as assinaturas **assinatura-1.png** e **assinatura-3.png**.

obs.: Note que o arquivo de assinaturas possui a extensão **PNG**. Por ser uma imagem pequena e que necessita de *alpha*, optamos por usá-la.

Abaixo uma simulação da árvore de diretórios `${SiGE}/public/img/certificados/`:

```
${SiGE}/public/img/certificados/
|
+ -- 1/
|   |
|   + -- modelo.jpg
|       assinatura-1.png
|       assinatura-3.png
|
+ -- default/
|   |
|   + -- modelo.jpg
|       modelo.svg
|       assinatura-modelo.png
|       assinatura-1.png
```

Dados do evento

A página inicial contém os dados do evento. Para modificar os dados basta editar o arquivo `${SiGE}/public/js/index/index.js`.

Nesse arquivo temos um objeto chamado `conference`, no qual você vai descrever seu encontro.

Comece editando o nome curto do encontro, juntamente com o nome completo, por exemplo:

```
short_name: 'COMSOLiD',
full_name: 'Comunidade Maracanauense de Software Livre e Inclusão Digital',
```

Contagem Regressiva Para modificar a data e os dados referentes a contagem regressiva que aparece na página inicial do SiGE, modifique os atributos `starts_at` e `ends_at`, por exemplo:

```
starts_at: moment(new Date(2014, 11, 16, 8, 0)),
ends_at: moment(new Date(2014, 11, 19, 17, 0)),
```

obs.: lembrando que em Javascript os meses vão de 0 a 11.

`month`

Integer value representing the month, beginning with 0 for January to 11 for December.

Fonte: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date

Características do encontro Defina as características ou *features* do seu encontro, por exemplo:

```
features: {
  columns: 3,
  list: [
    {
      title: 'Feature 1',
      description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit...',
      icon: 'fa-graduation-cap'
    },
    {
      title: 'Feature 2',
      description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit...',
      icon: 'fa-gamepad'
    },
    {
      title: 'Feature 3',
      description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit...',
      icon: 'fa-desktop'
    }
  ]
},
```

O atributo `columns` pode ter valor 2 ou 3. Você pode criar quantas **features** quiser adicionando ao atributo `list`.

É recomendado que você crie no máximo 6 features, para que não polua muito a página.

Os ícones são do *toolkit* [FontAwesome](#). Acesse o link [Ícons](#) e veja as opções.

Mapa Modifique as coordenadas do local onde acontecerá seu encontro, por exemplo:

```
map: {
  latitude: -3.87259,
  longitude: -38.610976,
  zoom: 17,
  address: 'IFCE - Campus Maracanaú - Av. Parque Central...'
}
```

Obs.: Existem algumas políticas de uso referentes ao OpenStreetMap (api usada para mostrar o mapa). Para um bom uso da api:

- Use somente **User-Agent** válidos;
- Não envie cabeçalhos **no-cache** (“Cache-Control: no-cache”, “Pragma: no-cache” etc);
- Faça cache do **Tile** (**Tile** é a imagem do mapa obtida através da api, como o local não deve mudar, esse recurso deve ficar em cache);
- No máximo são usadas duas threads de download.

Browsers modernos com a configuração padrão geralmente passam por todas as especificações acima.

Referência:

- http://wiki.openstreetmap.org/wiki/Tile_usage_policy
- <https://leanpub.com/leaflet-tips-and-tricks/read>

Redes Sociais A última página contém os links para as redes sociais do seu encontro. Edite de acordo com o exemplo abaixo:

```
social_networks: [
  {
    url: 'https://twitter.com/comsolid',
    channel: 'twitter'
  },
  {
    url: 'https://facebook.com/comsolid',
    channel: 'facebook'
  },
  {
    url: 'https://github.com/comsolid',
    channel: 'github'
  },
],
```

Os botões são gerados usando o [Social Buttons 3](#), utilize os nomes oferecidos pela api.

Por que as configurações em JS e não no banco de dados? Você pode estar se perguntado isso e a resposta é:

Essas configurações são feitas apenas uma vez, e nós da COMSOLiD achamos que seria trabalhoso para o servidor ter que carregar essas informações do banco de dados cada vez que a página fosse acessada.

Twitter No arquivo `${SiGE}/application/configs/application.ini` altere as linhas:

```
twitter.username = "els"; sem "@"  
twitter.hashtags = "els1"; sem "#" e separadas por ", "
```

Layout do Sistema

Por usar o Twitter Bootstrap 3, fica mais fácil mudar o tema, ou até mesmo criar um.

O SiGE suporta por padrão 3 temas:

- [Padrão](#)
- [Lumen](#)
- [Darkly](#)

Outros temas podem ser encontrados em [Bootswatch](#).

Para editar o tema edite o arquivo `${SiGE}/application/layouts/twbs3.phtml`, procure pela linha:

```
$this->headLink()->prependStylesheet(  
    $this->baseUrl('lib/css/bootstrap/default/bootstrap.min.css'));
```

e mude `default` para `lumen` ou `darkly` ou ainda um tema que você tenha baixado.

Os temas devem ficar em `${SiGE}/lib/css/bootstrap/`.

Banner do Sistema

Para alterar o banner geral basta substituir o arquivo `${SiGE}/public/imagens/layout/topo_sige.png`. As dimensões são: **962x135**.

Versão Móvel

Edite o arquivo `application/layouts/scripts/mobile.phtml`:

Linha 44: `<h1>COMSOLiD <?=date('Y') ?></h1>`