SUMÁRIO

[1 Servidor domus 3](#_Toc310533892)

[1.1 Configuração/Manutenção do servidor 4](#_Toc310533893)

[1.2 Estrutura de arquivos no servidor 4](#_Toc310533894)

[1.3 script de backup do servidor 5](#_Toc310533895)

[2 CmapTools 6](#_Toc310533896)

[2.1 Instalação e configuração DO CMAPTOOLS SERVER 7](#_Toc310533897)

[3 Subversion 8](#_Toc310533898)

[3.1 Instalação e Configuração do Subversion 8](#_Toc310533899)

[3.2 SVN Tortoise 9](#_Toc310533900)

[3.2.1 Configurando SVN Tortoise 9](#_Toc310533901)

[4 Moodle 13](#_Toc310533902)

[4.1 instalação do moodle 15](#_Toc310533903)

[4.2 Estruturas modificadas 16](#_Toc310533904)

[4.3 Módulos desenvolvidos/alterados 16](#_Toc310533905)

[4.3.1 Alteração template do Moodle 16](#_Toc310533906)

[4.3.2 Estrutura de Cursos 18](#_Toc310533907)

[4.3.3 Menu Vertical Principal 20](#_Toc310533908)

[4.3.4 Menu Horizontal 23](#_Toc310533909)

[4.3.5 Novo Gerenciador de Arquivos 23](#_Toc310533910)

[4.3.6 Pop-up de Imagens Automatizado 25](#_Toc310533911)

[4.3.7 Instalação do plugin DragMath 26](#_Toc310533912)

[4.3.8 Histórico 28](#_Toc310533913)

[4.3.9 Novo Fórum 28](#_Toc310533914)

[5 Organizador gráfico do domus 28](#_Toc310533915)

[5.1 Estrutura 28](#_Toc310533916)

[5.2 Configuração do Mapa no Adobe Flex Builder 3 30](#_Toc310533917)

[5.2.1 Inicialização do projeto/básico no Flex 30](#_Toc310533918)

[5.2.1.1 Arquivos/Scripts 30](#_Toc310533919)

[5.2.2 Configuração dos componentes 31](#_Toc310533920)

[5.2.3 Botões de links de texto no mapa 33](#_Toc310533921)

[5.2.4 Botões de links de vídeo ou slide no mapa 33](#_Toc310533922)

[5.2.4.1 Carregar o arquivo no servidor 33](#_Toc310533923)

[5.2.4.2 Criar arquivo “php” para os vídeos, no servidor. 34](#_Toc310533924)

[5.2.4.3 Criar a miniatura do vídeo 34](#_Toc310533925)

[5.2.4.4 Criar recurso na plataforma para nivelar usuário 35](#_Toc310533926)

[5.2.4.5 Configure o botão de link no Flex: 36](#_Toc310533927)

[5.2.5 Compilação e SWF 37](#_Toc310533928)

[5.3 Módulos desenvolvidos/alterados 37](#_Toc310533929)

# Servidor domus

Hoje são disponibilizados dois dominios para o projeto Domus: <http://domus.pucpr.br> e <http://powerdomus.pucpr.br>. Esses dominios estão configurados em maquinas que ficam alocadas no 7º andar do prédio administrativo da Pontifícia Universidade Católica do Paraná, sobre os cuidados da equipe de rede 2º nível.

O projeto contava com um servidor ultrapassado e com um sistema operacional já absoleto pela comunidade open source, que já não disponibilizava mais pacotes para ser atualizado. Para instalar um SO mais recente, ter um ganho no desempenho no Moodle e suportar vários acessos simultâneos houve a necessidade de adquirir um novo servidor com maior capacidade.

O novo servidor que hospeda o domínio HTTP://domus.pucpr.br, contém as seguintes configurações:

|  |  |
| --- | --- |
| **Hardware** | **Descrição** |
| Processador | Intel(R) Xeon(R) CPU X3450 2.67GHz 8mb de cachê |
| Memória | 8Gb 1333 MHz |
| Placa de vídeo | GeForce 8400 GS pci express 512mb ddr3 |
| Rede | RTL8111/8168B PCI Express Gigabit Ethernet controller 10/100/1000 |
| Disco rígido | Duas unidades SAMSUNG HD502HI de 500gb cada, ligados em RAID |

Além das configurações básicas de instalação do servidor, foram adicionados os seguintes programas para uso do projeto:

* **Fedora v.13** – Sistema Operacional que é uma distribuição Linux baseada em pacotes RPM, criada pela Red Hat;
* **Mysql v.5.1.51** – É o sistema gerenciador de banco de dados;
* **Php v.5.3.3** – Linguagem de programação;
* **Apache v.2.2** – É um servidor web livre;
* **PhpMyADM v.3.3.7** – Administrador do Mysql em Php, onde o banco pode ser gerenciado via browser;
* **ProFTPd**– Servidor FTP;
* **Webalizer**– Aplicativo para gerar páginas de análise;
* **SNMP / Mrtg** – Ferramenta de monitoração de carga de tráfego coletados pelo SNMP;
* **CmapServer v.5.04** – Permite que usuários se conectem para a construção dos MCs e atua como um repositório compartilhado;
* **SVN** – Subversion é um sistema de controle de versão;

## Configuração/Manutenção do servidor

A manutenção é feita por acesso remoto via SSH pela porta 22 em ambos os servidores. Para acessá-los são usadas as seguintes informações:

Servidor <http://domus.pucpr.br>:

**Host:** domus.pucpr.br

**Usuário:** root

**Senha:** $RFV%tgb2011

Servidor <http://powerdomus.pucpr.br>:

**Host:** powerdomus.pucpr.br

**Usuário:** root

**Senha:** $RFV%tgb2011

## Estrutura de arquivos no servidor

1. /home/files/backup/files – Fica armazenado o backup dos últimos sete dias do diretório /home/www/moodledata;
2. /home/files/backup/bd – Fica armazenado o backup dos últimos sete dias de toda a base de dados do mysql
3. /home/files/backup/www – Fica armazenado o backup dos últimos sete dias do diretório /home/www/html/
4. /home/ihmc – Repositório do CmapTools
5. /home/www/html/ – Diretório de publicação, onde ficam todos os arquivos que serão usados no site
6. /home/www/moodledata/ – Diretório que fica os arquivos do conteúdo do site
7. /home/www/html/svn/– Repositório do versionador

O backup está programado no *crontab* do Linux para rodar o arquivo de script /bin/backup.sh todos os dias à 05:00 horas da manhã. Para programar o *crontab* é só dar o comando crontab –e e tem que seguir o seguinte padrão abaixo:

[minutos] [horas] [dias do mês] [mês] [dias da semana] [usuário] [comando]

Pode ser usado o ‘\*’ (asterisco) para especificar uma execução constante. Por exemplo, no lugar de dias do mês conter o ‘**\*’**, o comando relacionado será executado todos os dias.

## script de backup do servidor

Script usado para fazer backup do servidor dos diretórios \home\www\html\ e \home\www\moodledatae de toda a base de dados. O script está salvo em \bin\backup.sh no novo servidor e está configurado no *Crontab* do linux para rodar todos os dias as 5 horas da manhã. O backup mantêm guardados os dados dos últimos sete dias, para não comprometer o espaço físico do servidor. Os dados ficam armazenados da seguinte maneira:

1. \home\files\backup\arquivos\: Os arquivos utilizados pelo o Moodle que ficam no diretório moodledata;
2. \home\files\backup\www\: Os arquivos fontes do site Domus;
3. \home\files\backup\bd\: Os arquivos do banco de dados

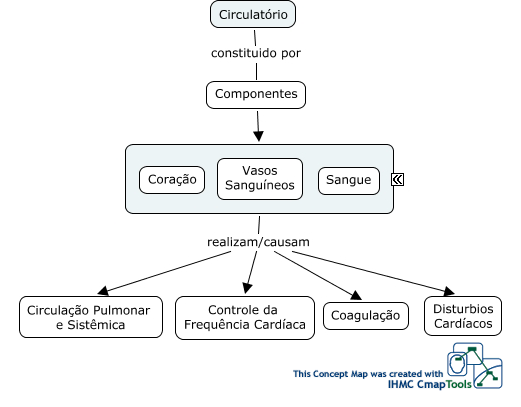
A versão 1.0 do script está escrita da seguinte forma:

|  |
| --- |
| # #Script de backup de arquivos do domus #Criado por: Hersilio B de Oliveira #Data:05/05/2011  #Versão: 1.0 echo "" echo "iniciando backup dos arquivos..."  #variaveis --------------------------------------------------------- DIR\_ORIG="/home/www/" DIR\_DEST="/home/files/backup/" EXT1="\_bkp\_"`date +%Y%m%d`".tar.gz" EXT2="\_bkp\_"`date '+%Y%m%d' --date='7 days ago'`".tar.gz" EXT3="\_bkp\_"`date +%Y%m%d`".sql"  #parar o servico do apache ------------------------------------------- echo "" echo "parando servico httpd..." /etc/init.d/httpd stop  #bkp do diretorio moodledata ---------------------------------------- echo "" CURRENT="moodledata" echo "iniciando backup diretorio: ${DIR\_ORIG}${CURRENT}" tar -cvzf ${DIR\_DEST}arquivos/${CURRENT}${EXT1} ${DIR\_ORIG}${CURRENT} rm -r -f ${DIR\_DEST}arquivos/${CURRENT}${EXT2} echo "fim backup diretorio: ${CURRENT}"  #bkp diretorio html ------------------------------------------------- echo "" CURRENT="html" echo "iniciando backup diretorio: ${DIR\_ORIG}${CURRENT}" tar -cvzf ${DIR\_DEST}www/${CURRENT}${EXT1} ${DIR\_ORIG}${CURRENT} rm -r -f ${DIR\_DEST}www/${CURRENT}${EXT2} echo "fim backup diretorio: ${CURRENT}"  #bkp banco de dados ----------------------------------------------- echo "" echo "iniciando backup do banco de dados..." mysqldump --host localhost --user root --all-databases > ${DIR\_DEST}bd/bd${EXT3} rm -r -f ${DIR\_DEST}/bd/bd\_bkp\_"`date '+%Y%m%d' --date='7 days ago'`".sql echo "fim backup do banco de dados!"  #iniciar o servico do apache ------------------------------------------ echo "" echo "iniciando servico httpd..." /etc/init.d/httpd start  echo "" echo "Backup realizado com sucesso!"  echo "" exit 0 |

# CmapTools

Ferramenta com finalidade de elaborar esquemas conceituais e representa-los graficamente, ou seja, um programa que auxilia em criar mapas conceituais.

O pesquisador define mapa conceitual como uma ferramenta para organizar e representar o conhecimento. Os conceitos aparecem dentro de caixas e as relações entre eles são especificados por meios de frases de ligação, que unem cada um dos conceitos. Exemplo de mapa desenvolvido no *CmapTools*:



1. Mapa conceitual criado no CmapTools

Com acesso via internet pelo o *CmapTools Server*, configurado no servidor do Domus, há uma coleção de trabalhos que podem ser utilizados como referência ou consulta. Todos os trabalhos desenvolvidos pela a ferramenta podem ser convertidos em formatos para apresentação na Web, para publicação e difusão do projeto em um ambiente virtual.

## Instalação e configuração DO CMAPTOOLS SERVER

1. Criar na mesma pasta onde está o instalador um arquivo de texto chamado InstallCmapServer.properties;
2. Modificar as permissões dos arquivos com o comando chmod 775 LinuxCmapServer\_vXX.bin e chmod 775InstallCmapServer.properties. Obs: O instalador encontra-se no diretório /home/files/programas/LinuxCmapServer\_v5.04\_07-16-10.bin do servidor domus.pucpr.br;
3. Instalar o cmaps com o comando ./LinuxCmapServer\_vXX.bin -i console -f InstallCmapServer.properties.

No servidor foi instalado o repositório do *Cmap Server* no diretório /home/ihmc e criado o login padrão: **usuário:** admin; e senha admin. O serviço utiliza a porta 81, e pode ser acessada pela a url <HTTP://domus.pucpr.br:81>.

# Subversion

Também conhecido como SVN, é um sistema de controle de versão. Foi implantado no ambiente de desenvolvimento com a finalidade de manter o histórico e o controle de alterações feitas nos códigos fontes. Assim podemos facilmente voltar em uma versão especifica, quando há a necessidade de rever o que foi feito ou algum problema gerado em novas versões.

## Instalação e Configuração do Subversion

1. Comando yum install subversion
2. Comando yum install mod\_dav\_svn
3. Criando o repositório:  
   mkdir /home/www/svn/repos  
   svnadmin create /home/www/svn/repos/domus
4. Alterar a propriedade da pasta para o apache:  
   chown -R apache.apache /home/www/svn
5. Criar arquivo /home/www/svn/repos/domus/svnauth com os usuários autorizados para acessar o repositório. Conteúdo do arquivo:

O exemplo abaixo do conteúdo do arquivo, o user1 terá acesso leitura-gravação enquanto o usuário user2 terá acesso somente leitura para o repositório inteiro.

[/]

user1 = rw

user2 = r

1. Criar o arquivo /home/www/svn/repos/domus/svnpass

htpasswd -bcm /home/www/svn/repos/domus/svnpass user1 passwordUser1

htpasswd -bm /home/www/svn/repos/domus/svnpass user2 passwordUser2

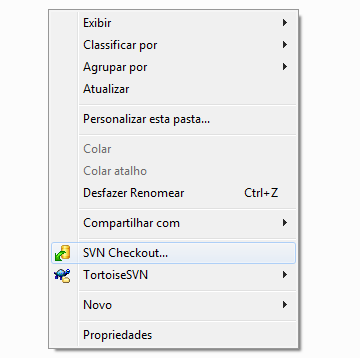
Obs. Para o primeiro usuário é usado o comando –bcm para o restante é –bm

1. Adicionar o seguinte código no arquivo de configuração do Apache que se encontra em /etc/httpd/conf/httpd.conf. Em seguida deve-se fazer o reinicio do apache com o comando httpd restart.

## SVN Tortoise

### Configurando SVN Tortoise

Primeiro baixar o software *tortoise* na url: <http://tortoisesvn.net/downloads.html>. Após instalar clique com o botão direito no diretório aonde será criado o repositório e clique em clique com o botão direito no diretório aonde será criado o repositório e clique em “*SVN Checkout...*”.



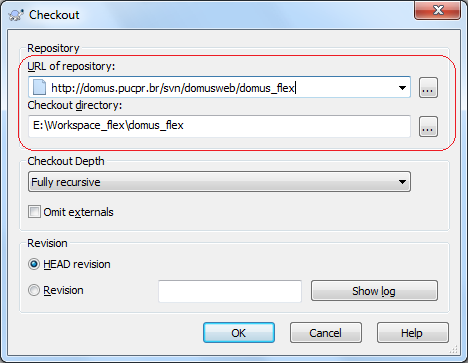
1. *“Checkout”* do repositório

Digite a URL do Repositório aonde se encontra os arquivos.

Fonte dos arquivos Flex: <http://domus.pucpr.br/svn/domusweb/domus_flex>

Fonte dos arquivos PHP: <http://domus.pucpr.br/svn/domusweb/domus_web>

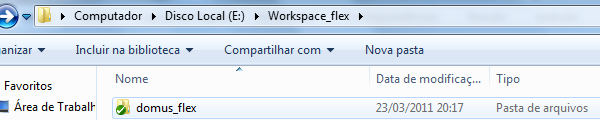
Fonte com documentações: <http://domus.pucpr.br/svn/domusweb/domus_docs>

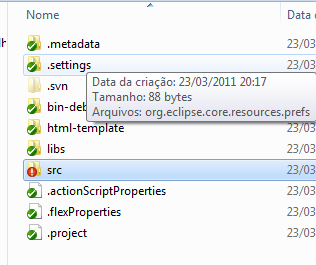


1. Url do repositório para “*Checkout”*

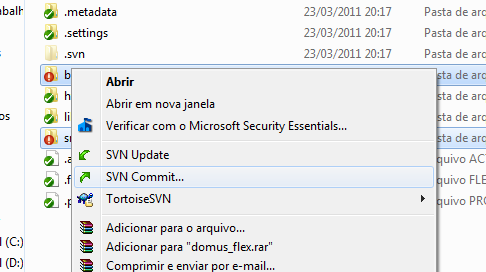
Após baixar os arquivos é só editá-los. Quando houver alguma alteração a imagem do diretório se altera ficando em vermelho e com o símbolo ‘**!’**. Para subir está versão para o servidor é só selecionar o arquivo(s) ou diretório(s) clicar com o botão direito e dar “*Commit*”, conforme mostrado abaixo:

1. Selecione o arquivo/diretório que há alterações

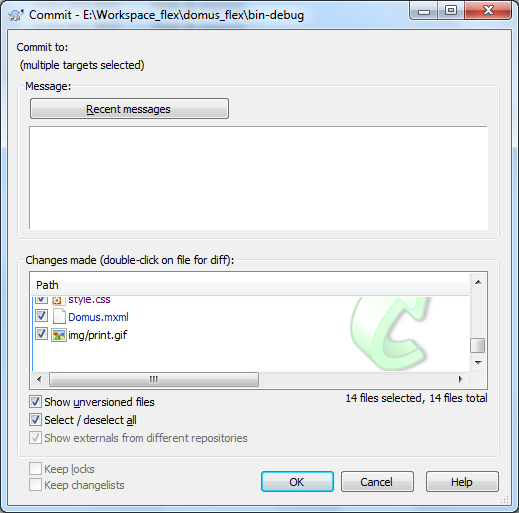




1. Clicando com o botão direito selecione a opção “*SVN Commit...*”

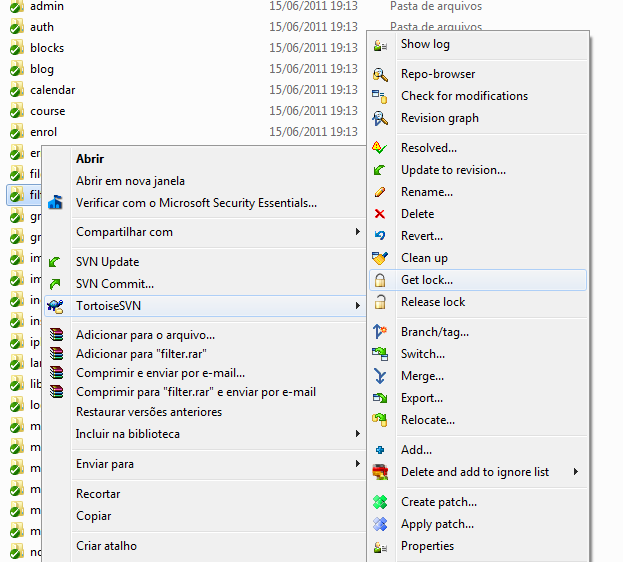


1. Os arquivos alterados serão mostrados. Selecione os que subirão na versão e clique em OK



Existem outras funcionalidades que devem ser seguidas para que seja realizada a alteração no servidor também ao dar *commit*:

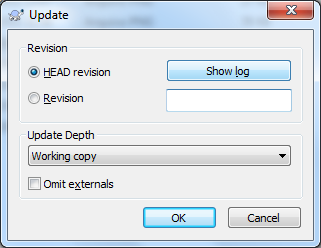
* Em alteração de nome do arquivo ou diretório utilizar a opção ‘*Rename...*’
* Ao adicionar novo diretório ou arquivo o mesmo irá mostrar um ícone com o símbolo ‘?’ (ex:), clique com o botão direito e selecione a opção ‘*Add..*’, o ícone irá alterar para o símbolo ‘+’ (ex: )
* Para excluir um arquivo ou diretório deve-se usar a opção ‘*Delete*’



1. Menu de opções do SVN

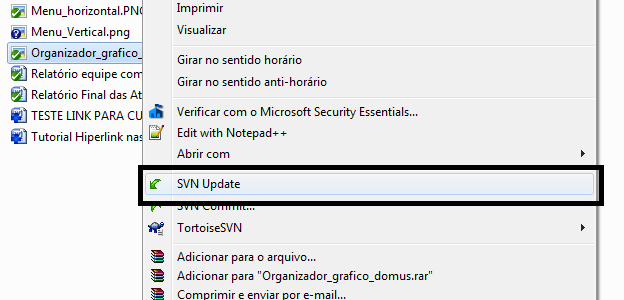
Para desfazer alterações que realizou em um arquivo ou diretório é só selecionar a opção ‘*Revert*’ que ele atualizará os arquivos para a versão atual que está no servidor.

Para voltar uma versão especifica é só selecionar o arquivo ou diretório e na opção ‘*Update to revision’* e irá abrir a tela abaixo. Em seguida coloca o número da versão que deseja baixar, caso não saiba qual o número, é só clicar na opção ‘show log’ que mostrará todas as versões que foram para o servidor e quem as fez.



1. Tela para voltar versões do SVN

Obs. Importante: TODA vez que for editar um arquivo ou diretório fazer *Update* de versão na opção “*SVN Update*” para não gerar conflitos de códigos, e ter que fazer um merge do código fonte. Toda vez que terminar de editar fazer o *Commit* para preservar o servidor sempre atualizado.



1. Opção de Update do SVN

Outra observação importante é configurar o arquivo /config.php no raiz do Moodle para as configurações do computador de desenvolvimento, este arquivo não precisa fazer *Commit*.

Existem outras opções interessantes no SVN, mas por hora essas são as básicas e as mais importantes para um bom trabalho de versionamento de código.

# Moodle

O Moodle (Modular Object-Oriented Dynamic Learning Environment, em português, Ambiente de Aprendizagem Dinâmica Modular Orientado a Objeto) é um software de apoio a aprendizagem, executado num ambiente virtual.

Criado em 2001 pelo educador Martin Dougiamas, o Moodle tem por objetivo facilitar a administração de ambientes educacionais voltados à aprendizagem colaborativa. O software permite uma fácil integração entre professores e acadêmicos por meio de cursos on-line. Diversas instituições de ensino estão adaptando a plataforma aos próprios conteúdos, graças ao fato do programa ser disponibilizado gratuitamente sobre a licença GNU-GLP e ser facilmente instalado em ambientes virtuais (servidores) que executem a linguagem PHP e uma base de dados, como Mysql. A licença GNU-GLP permite que qualquer pessoa possa utilizar um software sob esta licença, fazendo modificações no mesmo e redistribuindo o programa e seu código-fonte, desde que garantindo para todos os mesmos direitos.

As principais funcionalidades do Moodle são:

* **Fóruns de discussão:** são ferramentas para páginas de Internet destinadas a promover debates através de trocas de mensagens publicadas sobre um mesmo tema;
* **Gestão de conteúdos:** são meio de apoiar ao gestor de um curso organizar e distribuir conteúdos originados de várias fontes;
* **Questionários e pesquisas com diversos formatos;**
* **Blogs:** é um site cuja estrutura permite a atualização rápida a partir de acréscimos de artigos, também denominados ‘*posts*’;
* **Wikis:** sãosoftwares colaborativos que permitem a edição coletiva de documentos utilizando um sistema não necessitando que o conteúdo tenha que ser revisto antes da sua publicação;
* **Geração e gestão de bases de dados;**
* **Chat;**
* **Glossários;**
* ***Peer assessment*:** é um sistema de avaliação pelos pares, nos quais os alunos comentam e julgam os trabalhos de seus colegas**;**
* **Suporte multi-idioma.**

O Moodle permite que o desenvolvimento das atividades seja feito empregando diferentes recursos, como:

* Materiais;
* Avaliação do Curso;
* Chat;
* Diálogo;
* Diário;
* Fórum;
* Glossário;
* Lição;
* Pesquisa de Opinião;
* Questionário;
* SCORM;
* Tarefa;
* Trabalho com Revisão;
* Wiki.

Na próxima seção serão abordados os procedimentos para instalação do Moodle.

## instalação do moodle

O Moodle utiliza alguns softwares durante sua execução, para isso eles deverão ser instalados antes de instalar o Moodle. São eles:

* **Servidor web**: programa de computador responsável por aceitar requisições de páginas HTTP de navegadores (clientes) e servi-los com respostas também no formato HTTP. O servidor web mais conhecido e utilizado é o Apache;
* **Linguagem PHP**;
* **Servidor de bases de dados**: semelhante ao servidor web, com a diferença que provê dados ao invés de informações HTTP. O mais utilizado é o Mysql.

O primeiro passo é a obtenção da última versão estável do Moodle (v1.9) no endereço <http://download.moodle.org/download.php/stable19/moodle-weekly-19.zip>. Depois de efetuado o download, o próximo passo é descomprimir o arquivo na pasta de publicação do servidor web. A instalação é feita por meio de um script localizado no arquivo install.php que é responsável por configurar o arquivo config.php do Moodle. Para executar o script deve-se simplesmente acessar a página principal do Moodle após ter descompactado os arquivos na pasta de publicação. O Moodle detectará que é preciso fazer as configurações do sistema antes de iniciar a utilização e irá conduzir a instalação por meio de páginas “html” que servirão para configurar o arquivo config.php e também criar o banco de dados no servidor de banco de dados. Durante este processo o Moodle informa ao usuário o sucesso ou não dos procedimentos de instalação. Maiores informações sobre a instalação do Moodle podem ser obtidas em: <http://docs.moodle.org/pt/Instalação_do_Moodle>.

A diferença entre instalar localmente o Moodle e instalar no servidor é apenas onde serão descompactados os arquivos. No caso da instalação local, os arquivos serão descompactados no disco rígido e o caminho padrão da aplicação será <http://localhost/>. Na instalação do Moodle num servidor não local os arquivos terão que ser descompactados localmente e depois copiados para a pasta ‘/srv/www/htdocs’ do servidor. Uma pasta chamada de ‘Moodle’ foi criada no servidor do PowerDomus para separar a estrutura de arquivos utilizada pelo Moodle dos arquivos já existentes da plataforma anterior. Assim, o endereço de acesso do software Moodle no servidor antigo do PowerDomus é <http://powerdomus.pucpr.br/moodle/> e o endereço de acesso da plataforma antiga continua o mesmo ([http://powerdomus.pucpr.br](http://powerdomus.pucpr.br/)). Atualmente o endereço com a nova plataforma é o <http://domus.pucpr.br>, ficando a antiga para testes.

## Estruturas modificadas

/blocks/menu\_curso/ – Diretório onde fica a programação do menu principal fly-out;

/files/ – Diretório do sistema de gerenciador de arquivos;

/include/js/ – Encontram-se os arquivos Java Scripts usado em todo o projeto;

/include/css/ – Encontram-se os arquivos de estilo usado em todo o projeto;

/lib/ – Diretório de bibliotecas usado em todo o projeto;

/lib/javascript.php – Arquivo onde adiciona no header os Java scripts adicionados no diretório /include/js/;

/mapa\_r/ – Diretório do projeto em flex do Organizador Gráfico do Domus;

/phpMyAd/ – Sistema gerenciador de banco de dados;

/config.php – Arquivo onde é configurado o ambiente que será rodado a aplicação;

/index.php – Arquivo raiz do projeto onde é montado a tela principal;

/theme/powerdomus/ – Diretório onde se encontra o tema modificado para se enquadrar no projeto Domus;

/moodledata/ – Diretório onde ficam todos os arquivos inseridos no gerenciador de arquivos do Domus;

## Módulos desenvolvidos/alterados

### Alteração template do Moodle

Um template, ou tema, é a forma como os elementos como figuras ou textos do site são mostrados ao usuário da aplicação. O Moodle organiza os temas em pastas, sendo possível ao administrador do site escolher qual tema irá utilizar. Para isso ele deve acessar a opção “Aparência” do menu de administração do sistema, em seguida clicar na opção “Temas” e escolher a opção “Seletor de temas”. Todos os temas instalados serão mostrados.

Cada tema contém uma pasta com os seguintes arquivos (maiores informações podem ser obtidas em http://docs.moodle.org/en/Theme\_basics):

pix/

config.php

favicon.ico

footer.html

header.html

styles.php

styles\_color.css

styles\_fonts.css

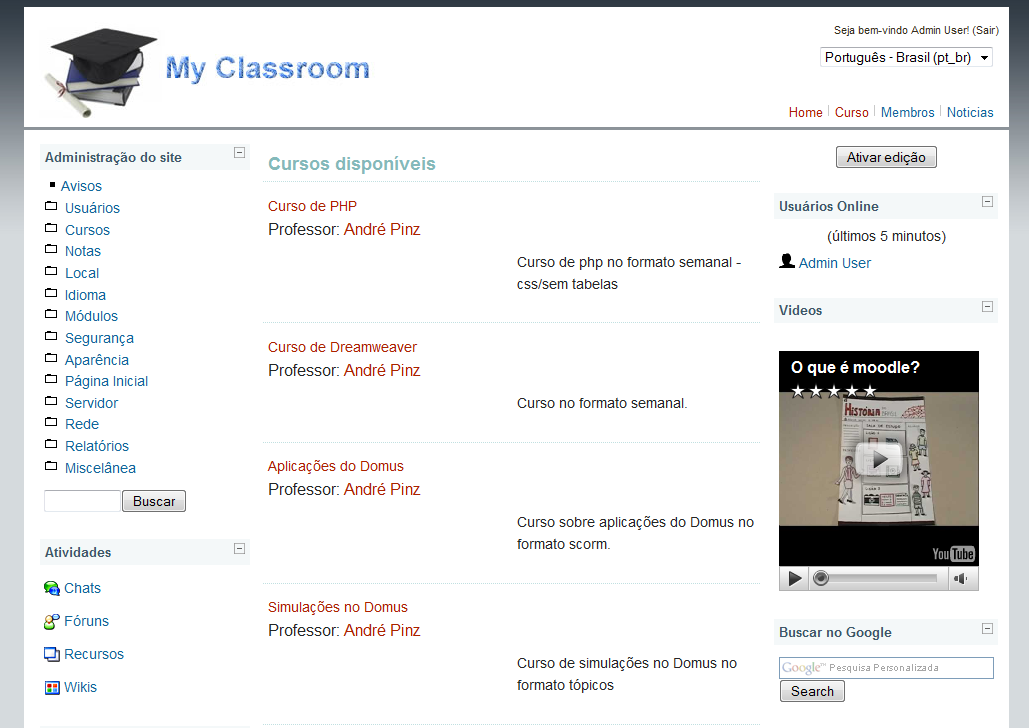
styles\_layout.css

styles\_moz.css

Para instalar um tema, basta copiar a pasta com os arquivos de configuração do tema para a pasta *theme* do Moodle.

Para criar um tema novo, é recomendado que seja copiada uma pasta de um tema já existente e renomeada com o nome desejado. Renomear a pasta evita que ao atualizar o Moodle as configurações personalizadas sejam perdidas.

O tema utilizado pelo Domus está configurado na pasta /theme/powerdomus. As alterações feitas até agora foram principalmente no arquivo my.css, que contém as configurações de cores e disposições de cada item na tela. O tema base utilizado foi o *bright\_mood*, mostrado na figura abaixo:



Tema base do Moodle

Após longas alterações no template, o resultado final foi um tema voltado para o projeto Domus, pensando na fácil usabilidade para o usuário final.



Moodle alterado para o layout do domus

### Estrutura de Cursos

Foi elaborada uma classe responsável em montar dinamicamente a estrutura dos cursos criados no Domus. Nela é respeitada a hierarquia e as ordens definidas pelo o usuário no administrador de cursos. A classe fica localizada no arquivo /util/cursos\_domus.php . Ele é incluído no arquivo principal de configuração do Moodle (/config.php), assim pode ser instanciada o objeto em qualquer lugar do projeto.

Métodos da classe:

function CursosDomus()

* Inicializa a classe;

function AlteraidPais()

* Método atualiza o atributo array\_cursos da classe, colocando em sequência ordena o id dos Pais para ser utilizado na biblioteca fly-out;

function ArrayPais()

* Método retorna um novo array só com os id’s que contêm filhos do atributo array\_cursos da classe;

function MenuPai($array)

* Método recebe como parâmetro o array montado na função “Sequence” e retorna preenchido o campo Pai, Qtd\_Filhos e Pai\_Id;

function Sequence($id\_course, $nome)

* Método recebe como parâmetro o id do curso e o nome. Retorna estrutura abaixo com todos os seus filhos e a endentação à direita na sequência correta definida no gerenciador de cursos;

|  |  |  |
| --- | --- | --- |
| $this ->array\_cursos (array) | |  |
| 0 | |  |  | | --- | --- | | array | | | id | 13 | | nome | Arquitetura Bioclimática | | indent | -1 | | pai | 13 | | qtd\_filhos | 6 | | pai\_id | 0 | | Id do curso ou atividade  Nome do curso ou atividade  Endentação do menu  Se possuir filhos é preenchido com o seu id  Número de filhos que possui  Id do seu pai se possuir |
| 1 | |  |  | | --- | --- | | Array | | | id | 403 | | nome | Introdução | | indent | 0 | | pai | 0 | | qtd\_filhos | 0 | | Pai\_id | 13 | |  |

1. Exemplo de Estrutura do Array de Cursos

function IsLink($name)

* Método retorna um valor boolean (true/false) se a variável $name é um link;

function Concat()

* Método retorna duas array’s concatenadas recebidas como parâmetro;

function CoresMenu()

* Método retorna a estrutura utilizada para definir as cores do menu fly-out principal;

function SetMenuCache($menu)

* Método atualiza o cache na tabela **mdl\_cache\_menu\_curso** recebendo com o parâmetro $menu;

function GetMenuCache()

* Método verifica se há cache e retorna o menu já montado da tabela **mdl\_cache\_menu\_curso**;

function ImprimirTree()

* Método retorna estrutura para montar a tree view do novo gerenciador de arquivos do domus (Em desenvolvimento);

function ImprimirMenu()

* Método retorna o menu **fly-out** principal com todos os cursos definidas no gerenciador do curso;

### Menu Vertical Principal

Foi criado a classe **MenuJS** dentro do arquivo \util\menu\_js.php, onde é criado dinamicamente o arquivo \blocks\menu\_curso\js\menu.js, que é a biblioteca responsável por montar o menu fly-out. É criado uma variável e os métodos Java script (no padrão XXIEHV[N], onde [N] é um numero sequencial atribuído a categoria. Zero sempre será atribuído ao bloco de cooperação que é outro SQL de busca), para cada bloco de categoria criada no gerenciador de cursos;

Para iniciar o Menu **fly-out** na biblioteca a variável deve ter a seguinte estrutura:

|  |
| --- |
| $cores=’#dedede','#c0c0c0','#dedede','#409db5','#409db5','#cdcdcd','','#c0c0c0','','','','','#409db5','#ffffff','#ffffff','#409db5','#409db5'  //Variável javascript para montagem do menu  var strm = [$id\_sequencial\_pai,$qtd\_filhos ,$cores, //inicio de nível  0,$id\_sequencial\_pai,'$nome\_curso\_ou\_atividade','$nome\_curso\_ou\_atividade','$url','','','',0,0 //filhos do nível atual  ...]; |

Se o nível não tem filhos substitui o $id\_sequencial\_pai por zero. Exemplo:

|  |
| --- |
| var strm1 = [0,3,$cores,  0,1,'A','A','javascript:void(0)','','','',0,0,  0,0,'G','G','javascript:void(0)','','','',0,0,  0,1,'H','H','javascript:void(0)','','','',0,0,  1,2,$cores,  0,2,'B','B','javascript:void(0)','','','',0,0,  0,0,'C','C','javascript:void(0)','','','',0,0,  2,3,$cores,  0,0,'D','D','javascript:void(0)','','','',0,0,  0,0,'E','E','javascript:void(0)','','','',0,0,  0,0,'F','F','javascript:void(0)','','','',0,0,  3,1,$cores,  0,0,'I','I','javascript:void(0)','','','',0,0]; |

Este código irá imprimir um menu na seguinte estrutura:

**A**

**B**

**D**

**G**

**E**

**F**

**C**

**H**

**I**

**C**

Dentro do arquivo /blocks/menu\_curso/block\_menu\_curso.php no método get\_content(), é instanciado o objeto CursosDomus e chamado o método ImprimirMenu().

Desta maneira o novo menu de cursos no Moodle, foi alterado para tratar essa hierarquia para que seja aceita ‘N’ níveis de cada categoria criada no gerenciador de cursos. Teve uma melhora significativa na usabilidade, pois ele se adapta a resolução do monitor não precisando dar o ‘scroll’ na tela para ir até aos níveis mais baixos. Exemplo do funcionamento do plugin em javascript junto com o algoritmo desenvolvido:



1. Funcionamento do Menu Fly-Out

### Menu Horizontal

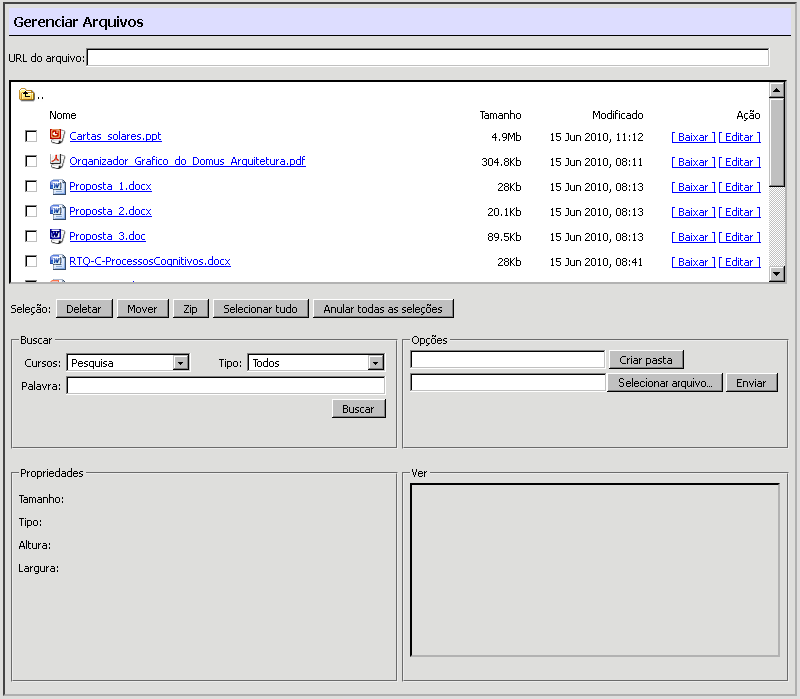
Está localizando dentro do arquivo /theme/powerdomus/menu.php. Ele é fixo e é montado basicamente via jQuery pelo o seguinte código:

|  |
| --- |
| $(document).ready(**function()**{  $("#nav-one li").hover(  **function()**{ $("ul", this).fadeIn("fast"); },  **function()** { }  );  **if** (document.all) {  $("#nav-one li").hoverClass ("sfHover");  }  });    $.fn.hoverClass = **function(**c**)** {  **return** this.each(**function()**{  $(this).hover(  **function()** { $(this).addClass(c); },  **function()** { $(this).removeClass(c); }  );  });  }; |

O Menu só tem 1 nível e só pode ser alterado no código fonte, por não ter muitas alterações foi adotada esta forma sem Admim.

### Novo Gerenciador de Arquivos

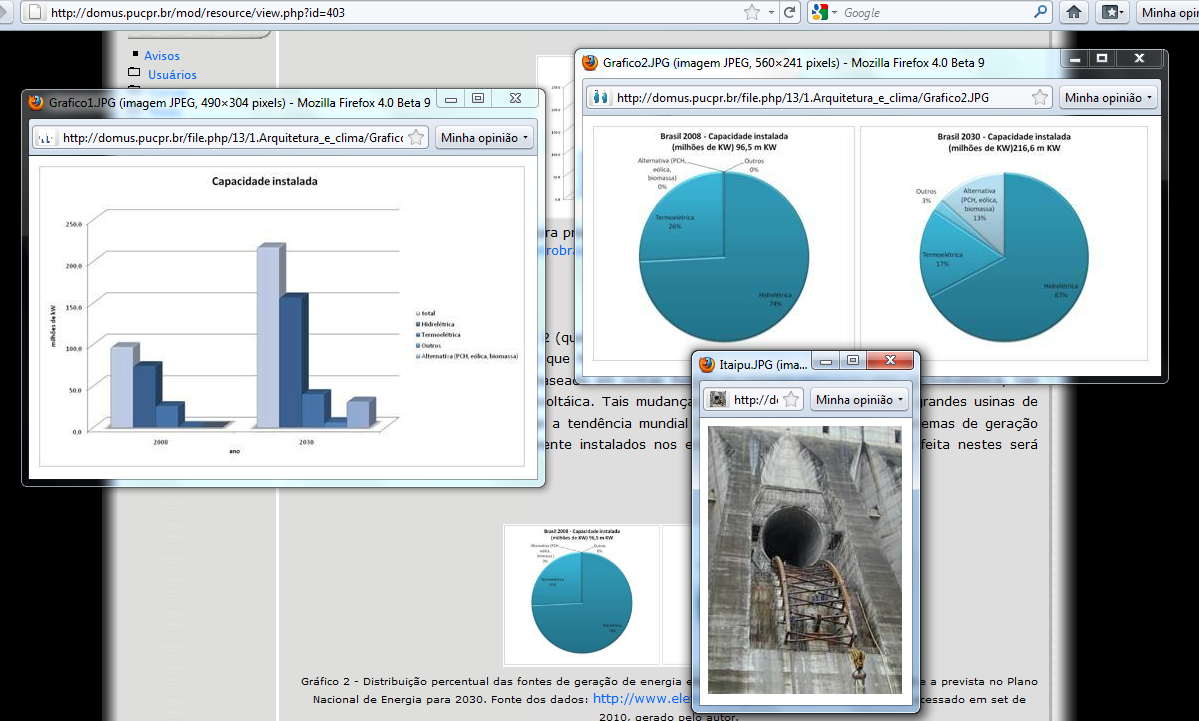
O gerenciamento de arquivos dos cursos também precisava de grandes alterações, tanto em layout como usabilidade, para ficar mais amigável a sua utilização. Na ferramenta que o Moodle disponibiliza, as suas funções fazem com que o usuário navegue em várias telas sem necessidade, além de faltar opções em que o projeto precisaria. Foram colocadas novas funcionalidades como: baixar arquivos busca dentro de outros cursos sem precisar mudar de página, pré-visualização de imagens, entre outros. As edições foram colocadas dentro de Iframes, assim as funcionalidades são feitas sempre na mesma página. O código fonte foi centralizado dentro do diretório /files/ do projeto.



1. Novo gerenciador de arquivos do Domus
2. URL do arquivo no servidor
3. Listagem dos arquivos e diretórios
4. Fazer o download do arquivo
5. Editar o nome do arquivo
6. Apagar o(s) arquivo(s) e diretório(s) selecionado(s)
7. Mover o(s) arquivo(s) e diretório(s) selecionado(s) para outro diretório
8. Zipar o(s) arquivo(s) e diretório(s) selecionado(s)
9. Selecionar todo(s) o(s) arquivo(s) e diretório(s)
10. Tirar as seleções de do(s) arquivo(s) e/ou diretório(s) selecionado(s)
11. Busca de arquivos utilizando os filtros selecionados
12. Ao clicar em um arquivo, mostra informações de tamanho, tipo e altura/largura para imagens.
13. Enviar um arquivo ao servidor ou criar novo diretório
14. Pré-visualização de arquivos imagem

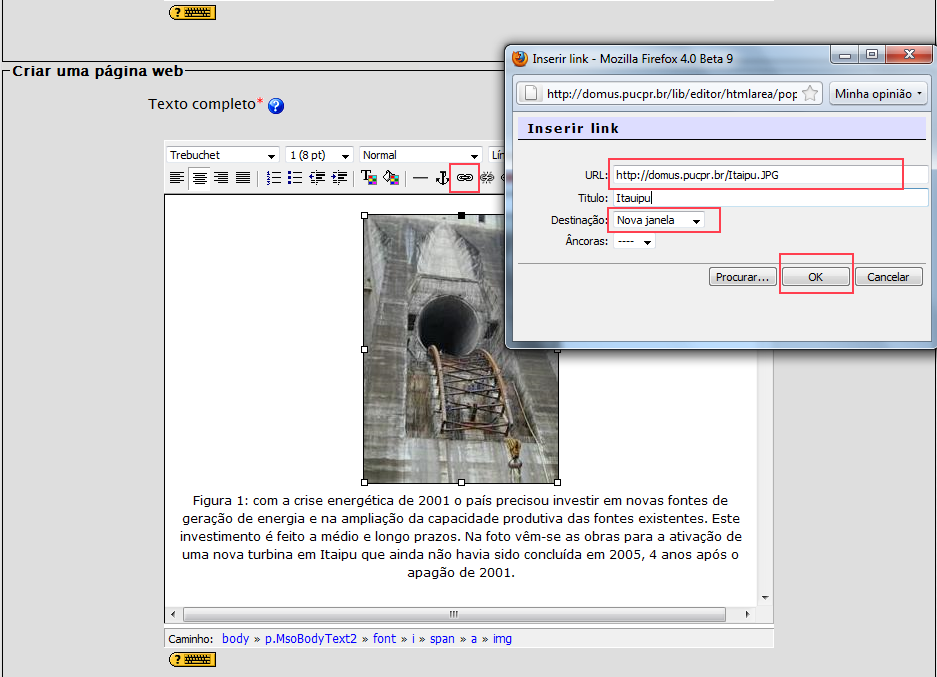
### Pop-up de Imagens Automatizado

Utilizando o framework jQuery, foi criada uma opção dentro do editor HTML que possibilita a abertura de imagens no tamanho natural em pop-up’s, com a finalidade de o usuário poder fazer comparativos entre várias imagens ao mesmo tempo, como demonstrado na imagem.



1. Demonstrativo de vários pop-up’s de imagens abertas

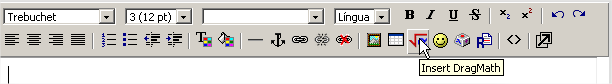
Para usar essa função no editor de textos ficou bastante simples, basta selecionar a imagem e com a opção de inserir hyperlink, coloca a URL de origem e destinação deste link em “Nova janela” como demonstrado logo abaixo na imagem. Após salvar o framework se encarrega de procurar as imagens com a opção de mostrar em pop-up.



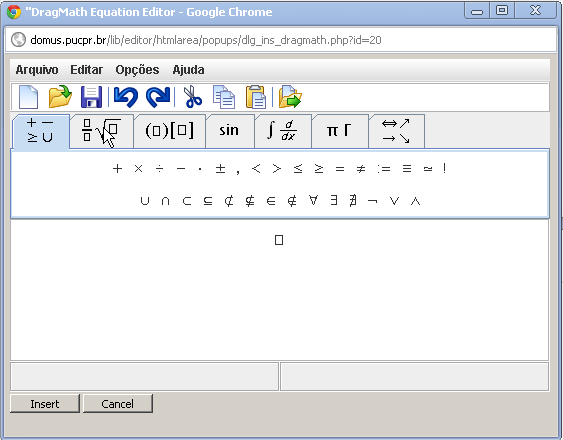
1. Demonstração de como criar um pop-up automatizado

### Instalação do plugin DragMath

Este plugin visa criar uma facilidade de inserir símbolos matemáticos no momento de edição de textos para a plataforma. Ele dispõe de uma gama muito grande de símbolos conforme mostrado na figura abaixo:



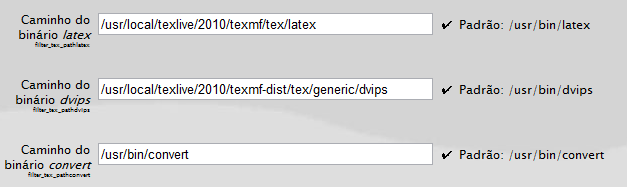
Opção no editor de textos para chamar o DragMath



Tela do DragMath com todas as opções de Fórmulas

Os passos para a instalação do plugin segue na url <http://docs.moodle.org/20/en/DragMath_equation_editor#Installing_DragMath_from_the_non-CVS_Zip_-_Recommended.21> publicado no fórum geral do Moodle. A instalação só será completa após a ativação da notação Tex no Moodle. Segue os passos para ativação:

1. No Menu Administração do site entrar em /Módulos/Filtros/Gerenciar Filtros
2. Ativar a Anotação Tex
3. Configurar o diretório dos binários do Latex como mostrado na imagem abaixo:



Após esses passo é necessário a instalação do “TexLive” no servidor. Para instalar segue os passos:

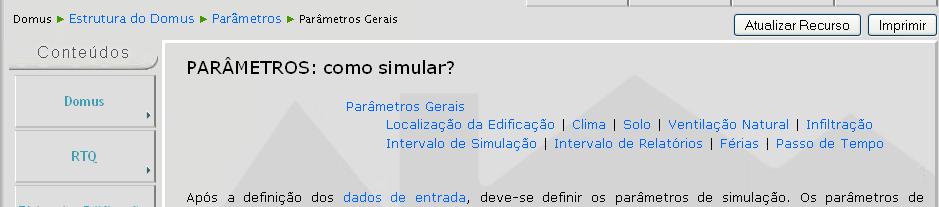
1. Baixado o arquivo zip install-tl-unx.tar.gz da URL <http://ctan.tche.br/systems/texlive/tlnet/>;
2. Descompactar o arquivo no servidor;
3. Navegar dentro do diretório aonde se encontra o arquivo install-tl;
4. Dar o comando "perl install-tl";

Por fim é preciso a instalação **ld-linux.so.2.** O Tex Filter precisa deste binário para funcionar. Para instalar apenas dar o comando: yum install ld-linux.so.2, que será instalado no diretório /lib

### Histórico

O sistema de histórico foi desenvolvido para que o usuário pudesse ver o caminho que percorreu para chegar ao assunto atual. Neste sistema é possível voltar para assunto anterior apenas clicando no link.

O código que controla a estrutura de navegação está fixo e se encontra no caminho \lib\weblib.php na função build\_navigation().



### Novo Fórum

O novo forum foi desenvolvido com as funções básicas do fórum já disponibilizado pelo moodle, com a estrutura de layout modificada, e com mais algumas funções como conceitos, bibliografia e anotações.

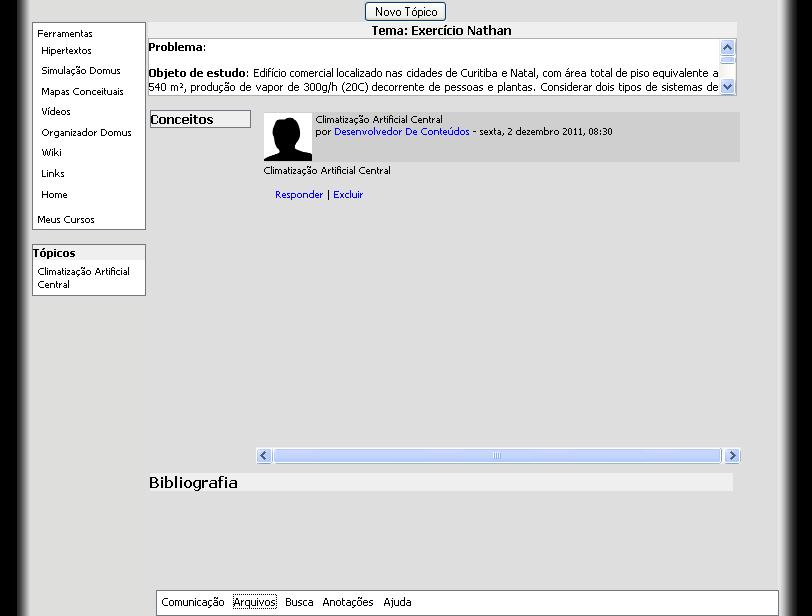
- Conceitos: o professor poderá lançar conceitos referentes a cada tópico de discussão.

- Bibliografia: os alunos terão acesso a outras bases de pesquisa, como sites e livros que foram usados para embasar a discussão no fórum.

- Anotações: o aluno poderá fazer suas anotações em diferentes fóruns, sendo que as mesmas só poderão ser visualizadas por ele.

Neste novo módulo o usuário poderá interagir com suas respostas através de pop-ups, sem precisar sair da mesma tela, e a cada alteração feita as respostas são atualizadas automaticamente sem que para isso seja recarregada toda a tela.

Todos os arquivos deste modulo estão em \mod\myforum e todas as tabelas relacionadas a ele possuem o prefixo mdl\_myforum.



# Organizador gráfico do domus

## Estrutura

O organizador é um gráfico no formato de mapa conceitual, com boxes para os títulos e subtítulos, com tamanhos e cores para diferenciá-los. Deve possibilitar a adição de links para outros recursos tipo textos, vídeos e slides.

Inicialmente, o organizador gráfico era uma aplicação em SVG, possuía suas tags próprias, portanto não aceita tags html dentro de sua estrutura, apenas scripts em Java script. Foi elaborado um modelo de testes com *PopUp* externa, para abrir os links do organizador gráfico. Criou-se um retângulo que fica invisível por natureza em cima dos itens a serem destacados, então, é criada uma área invisível em cima dos itens do RTQ, para detectar quando o mouse passar por cima dela. Quando detectar o mouse com o evento “*onmouseover*”, ele chama uma função, mostrada abaixo na imagem 7 – Função em Java script para destacar itens no mapa, tornando visível a sombra.

Foram feitos testes simples, criando formas diretamente no SVG e encontrado um problema grave: o SVG possui um problema grande de incompatibilidade com o browser Internet Explorer e outros navegadores, o problema envolve o posicionamento do SVG em relação à tela. Não foi encontrada uma solução eficaz para se criar o novo mapa totalmente em SVG, sem utilizar o mapeamento de imagem, como é usado no atual, sem que haja o problema de incompatibilidade.

Utilizou-se, então o framework que seria uma junção de várias funcionalidades de uma linguagem de programação, em um único recurso, o qual pode ser utilizado de modo mais eficaz pelo programador. O Adobe Flex 3 possui diversos componentes gráficos, que puderam ser utilizados de diversas formas na criação de um novo mapa, mais dinâmico e prático.

O Flex possibilita a edição pelo seu modo “Design” que é a parte visual da aplicação, e também a parte de código, que é onde ficam armazenados todos os objetos da aplicação. É semelhante ao XML do SVG.

Foram desenvolvidos os links, utilizando funções da linguagem de scripts do Flash, Action Script 3.

O menu RTQ tem uma rotina no Flex para quando o usuário clicar nas opções de expandir, em forma de arvore, o menu RTQ, ele mostrar novos itens que anteriormente estavam escondidos. O evento para expandir o menu foi feito manualmente pela linguagem de script do Flex, que é o Action Script 3.

Para os links de vídeo, utiliza-se o software ‘*VídeoBox Light’*, que é um aplicativo que gera um arquivo html, com os vídeos (no formato ‘flv’) em miniatura, quando clicado, expande uma janela com o player. As miniaturas são exibidas em uma página com o mesmo padrão do site. As páginas de redirecionamento dos links mandam o usuário para a pagina destino na qual possui uma rotina em PHP que tem como argumento, o diretório a ser varrido pela mesma. Ela varre o diretório, e imprime na tela, a galeria de vídeos, com todos os arquivos “flv” que tiverem na pasta.

Para controlar o acesso aos vídeos e slides pelos usuários, foram criados recursos no Curso do tipo links a página web, que ficam invisíveis no menu, configurados com o grupo que pode acessá-lo.

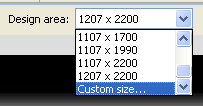
Também os slides são listados da mesma maneira que os vídeos. Em vez de uma rotina em PHP que imprime a galeria de vídeos, ela imprime links com os arquivos PPT achados na pasta. A pasta raiz dos slides está em /slides/. Os arquivos do PowerPoint ficam em uma pasta no servidor, localizados na raiz do sistema, e o script em php procura os slides relacionados à categoria (ex.: climatização, edificação, etc.). O controle do acesso pelos usuários também funciona como o dos vídeos.

## Configuração do Mapa no Adobe Flex Builder 3

### Inicialização do projeto/básico no Flex

Ao iniciar o Flex Builder 3, vá em File > Import > Flex Project, selecione Project Folder e procure a pasta em que se encontra o código fonte do projeto.

O organizador possui dimensões diferentes do padrão na área de desenvolvimento, procure no menu superior a opção Design Área, entre nela e vá em Custom Size e defina 1207 x 2200, para conseguir trabalhar com o mapa.



Clicando na aba **Outline**, no menu esquerdo inferior, você verá a hierarquia dos componentes usados no mapa.

É possível, por ali, selecionar os objetos no mapa, mudar suas configurações etc.

O layout da página estilo Domus no fundo é uma imagem. Sobre ela foi colocado um **Panel** (componente do Flex) chamado **panelMapa** e, dentro dele, estão todos componentes do mapa.

#### Arquivos/Scripts

O arquivo principal que modela o projeto, é o Domus.mxml.

Você pode alternar entre visualizar o design do mesmo ou o código-fonte apenas clicando no menu superior, que possui as duas opções. (Source/Design)



No inicio do código-fonte do arquivo “mxml”, é carregado um arquivo chamado Functions.as. É nele que fica grande parte das funções mais complexas que o aplicativo faz, como a parte do menu RTQ, que esconde/mostra itens. Para abrir este arquivo, basta dar dois cliques em cima dele, no menu esquerdo superior.



### Configuração dos componentes

Cada componente, quando selecionado, habilita no menu direito, diversas opções para manusear o mesmo, tanto opções visuais, como eventos, entre outras.

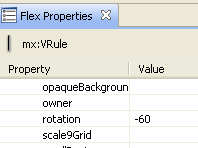
É possível ajustar todas as configurações sem problema algum, apenas selecionando o objeto a ser modificado e alterando.

O Flex, por padrão, exibirá um menu das configurações possíveis para serem ajustadas de maneira resumida, mas há também a possibilidade de visualização de todas as propriedades possíveis dos componentes, basta alterar nas opções a seguir:



As caixinhas dos itens do mapa são toda componente Button.

Obs.: As linhas diagonais, que estão no topo do mapa, têm seu ângulo de inclinação definido na propriedade rotação.



Os botões estão padronizados da seguinte forma:

- Tamanho dos botões através das propriedades de Layout, parâmetros Width e Height:

-- as caixas “Dados de Entrada”, “Parâmetros” e “Relatórios de Saída” têm Width 245 e Height 40.

-- as caixas azuis (Edificação, Horários de Uso, Ganhos Internos etc.) têm Width 213.75 e Height 45

-- as caixas brancas (Visualização, Edição, Equipamentos etc.) têm Width 180 e Height 22.45.

-- os ícones de links (T, V ou S) têm Width 12 e Height 13.

-- as linhas horizontais de subtítulos têm Width 19.8

- Alinhamento dos botões através das propriedades de Layout, parâmetros X para alinhamento vertical e Y para alinhamento horizontal.

-- as caixas “Dados de Entrada“, “Parâmetros” e “Relatórios de Saída” têm alinhamento horizontal Y 366.85

-- as caixas primeiras caixas azuis (Edificação na 1ª coluna, Parâmetros Gerais na 2ª coluna e Conforto Térmico na 3ª coluna) têm alinhamento horizontal Y 443.55.

-- as caixas brancas (Visualização, Edição, Equipamentos etc.) têm alinhamento vertical X 74 na 1ª coluna, X 454 na 2ª coluna e X 818 na 3ª coluna.

-- os ícones de links (T, V ou S) possuem alinhamento vertical. Desse modo, na primeira coluna, X 260 nas caixas azuis e X 255 nas brancas/cinzas; na segunda coluna, X 640 nas caixas azuis e X 637 nas brancas/cinzas; da 2ª coluna; e, por fim, na terceira coluna, X 1004 nas caixas azuis e X 1001 nas brancas/cinzas. Em locais onde o título não está em uma caixa, o ícone é colocado onde for mais apropriado.

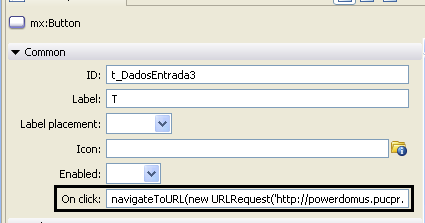
- as linhas verticais de subtítulo possuem alinhamento vertical X 50 na 1ª coluna, X 427 na 2ª coluna e X 793 na 3ª coluna.

- as linhas horizontais de subtítulo têm alinhamento da 1ª coluna de X 23.65 para caixas azuis e X 56 para as brancas/cinzas. Na segunda coluna de X 05.05 para caixas azuis e X 434.25 para brancas/cinzas e, por fim, na terceira coluna de X 767 para caixas azuis e X 799 para brancas/cinzas.

### Botões de links de texto no mapa

Para adicionar um link, basta selecionar uma caixinha de link no mapa, copiar, colar, ajustar a posição e alterar a URL destino, na propriedade **OnClick**. Lembre-se de renomear o botão recém-criado na propriedade ID para **t\_*TítuloDoRecurso***; e **Label** T, S ou V (texto, vídeo ou slide);

Na propriedade ***onClick*** (categoria eventos), a função **navigateToUrl** envia o usuário para a URL informada no primeiro parâmetro. O segundo parâmetro define como a página deve ser aberta: \_self, \_blank etc... Atualmente, todos os botões abrem na mesma janela (‘\_self’). Por exemplo, o link do texto de Edificação terá na propriedade **onClick**: *navigateToURL(new URLRequest('http://domus.pucpr.br/mod/resource/view.php?id=210'), '\_self');*



### Botões de links de vídeo ou slide no mapa

Os links de vídeo ou slide do mapa direcionam o usuário para um recurso da plataforma que permite que o aluno prossiga caso esteja inscrito no nível correto e o encaminhe para um aplicativo que exiba os vídeos ou slides.

Após criar um novo botão de link no mapa, siga os passos a seguir.

#### Carregar o arquivo no servidor

Localize as pastas que contenham arquivos de vídeo e de slides: /home/www/html/mapa\_r/slides/arquivos\_slides e /home/www/html/mapa\_r/videos/data/arquivos\_video.

Essas pastas contêm outras subpastas, cada uma representando um item do mapa. Quando for necessário criar novas pastas, mantenha o padrão de nome: sem espaços, com inicial minúscula e demais palavras com inicial maiúscula (por exemplo, “/horariosDeUso”)

Para adicionar slides, basta colocar arquivos com a extensão “ppt” ou ”pps” na respectiva pasta; e extensão “flv” para vídeos. O nome do arquivo aparecerá na página como legenda. Atualmente, não são aceitos nomes com acentuações.

#### Criar arquivo “php” para os vídeos, no servidor.

Para os slides, somente a criação da subpasta em home/www/html/mapa\_r/slides/arquivos\_slide contendo os arquivos “ppt” ou “pps” é suficiente, mas no caso de vídeos, além da subpasta em /home/www/html/mapa\_r/videos/data/arquivos\_video, um arquivo “php” com o mesmo nome do recurso, deve ser criado em /home/www/html/mapa\_r/vídeos. Utilize um dos arquivos existentes nesta pasta como modelo, salve como *nomeDoRecurso*.php e altere no código a seguinte parte:

|  |
| --- |
| <?php  include 'Util.php';  Varre("../../mapa\_r/vídeos/data/arquivos\_video/*nomeDoRecurso*/", ".flv");  ?> |

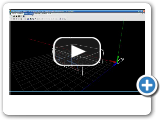
Note que nomeDoRecurso será o mesmo nome da pasta que contém os vídeos sobre aquele item.

Por exemplo, horariosDeUso.php tem no seu código:

|  |
| --- |
| Varre("../../videos/horariosDeUso/", ".flv"); |

#### Criar a miniatura do vídeo

Cada vídeo aparecerá na galeria de vídeos, com uma miniatura e um título. O título é o próprio nome do arquivo “flv” que está na pasta /home/www/html/mapa\_r/videos/data/arquivos\_video/*subpasta* do servidor. Já a miniatura é uma imagem “png”, também com o mesmo nome do arquivo “flv” (exatamente), de tamanho 160x120 pixels, que ficará na pasta home/www/html/mapa\_r/vídeos/data/thumbnails e segue o seguinte modelo:



Fundo branco sombreado, imagem do vídeo, símbolo de “play” e de “+”.

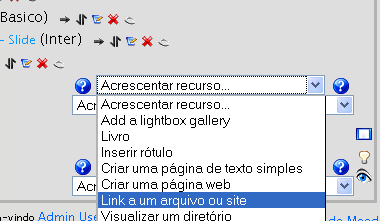
#### Criar recurso na plataforma para nivelar usuário

No momento, o sistema de links funciona de tal modo que para cada link do mapa, é necessário criar na plataforma um recurso que nivele o usuário (verifica se este pode ver tal recurso) e, assim, possa direcioná-lo para a página correspondente. Essa URL gerada pela plataforma é a que deve ir ao mapa em flex, dentro do evento **onClick**.

Para criar recurso na plataforma, entre como Administrador no **ambiente > ativar a edição > ir a Cursos > Acrescentar/ Modificar Cursos**, selecione Domus.

Na próxima lista selecione Domus novamente e será exibida toda a hierarquia dos itens presentes no mapa. Os links que já foram criados estão no final da lista.

Para acrescentar um novo recurso, vá em Acrescentar Recursos:



Feito isso, defina as configurações padrões:

Na seção Geral, configure o nome:

Nome: [-LINK] *Nome do Recurso* – *Tipo de arquivo* (por exemplo: “[-LINK] Edificação – Vídeo”)

Obs.: essa tag *[-LINK]* é utilizada pelo sistema para não mostrar tais recursos no menu da página inicial (sem ela, todos recursos criados aparecerão no menu da página inicial).

Na seção *Link a um arquivo ou site*, configure a localização do arquivo:

**Localização:**

Para vídeos, o link do arquivo php criado na etapa anterior: [http://domus.pucpr.br/mapa\_r/videos/***nomeDoRecurso***.php](http://domus.pucpr.br/mapa_r/videos/nomeDoRecurso.php)

(exemplo: http://domus.pucpr.br/mapa\_r/videos/edificacao.php)

**Para slides:** http://domus.pucpr.br/mapa\_r/slides/slides.php?slide=*nomeDoRecurso*&iN=***nomeDoRecurso***

Na seção de Configuração de módulos comuns:

Marque a opção “Disponível apenas para membros do grupo” e em seguida escolha em agrupamento qual o nível de usuário que pode acessar esse arquivo (o mesmo nível do recurso de texto)

Em Visível, escolha “Mostrar”.



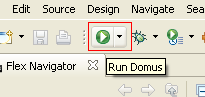
Agora basta Salvar e voltar ao Curso.

#### Configure o botão de link no Flex:

Do mesmo modo da configuração de link para recurso de texto, o botão de link para vídeo ou slide será configurado na propriedade **OnClick**, direcionando o usuário para o respectivo recurso [-LINK] criado no curso Domus, na plataforma.

### Compilação e SWF

Para compilar o projeto, basta clicar em Run:



Uma página HTML será aberta, com o projeto.

A compilação do projeto gera um arquivo “swf“ na pasta dos códigos fontes, dentro de uma subpasta chamada Bin-debug.

Ao enviar o mapa ao servidor, todos os arquivos da pasta Bin-debug devem ser transferidos.

## Módulos desenvolvidos/alterados

**Menu RTQ**

As opções do RTQ aparecem e desaparecem conforme o link clicado. Para isso, os elementos tornam-se visíveis/invisíveis e movem-se quando necessário. A configuração foi feita alterando-se o código da função **MoveDown**, em Function.as.

|  |
| --- |
| **public** **function** MoveDown(button: Button): **void**{  **var** box\_x : Number = 845;  **if**(button.id == **'btn\_comPresc'**){  **if**(button.label == **'+'**) {  button.label = **'-'**;  **var** box\_Envoltorio\_1 : Number = 1603.4;  *//Componentes Gráficos*  com\_Simul.move(com\_Simul.x, com\_Simul.y + 108);  residencial.move(residencial.x, residencial.y + 108);  res\_Presc.move(res\_Presc.x, res\_Presc.y + 108);  res\_Simul.move(res\_Simul.x, res\_Simul.y + 108);  *//linha\_Res.move(linha\_Res.x, linha\_Res.y + 108);*  *//Botão Simulacao Comercial*  btn\_comSimul.move(btn\_comSimul.x, btn\_comSimul.y + 108);  btn\_comSimul.enabled = **false**;  *//Botao Prescritivo Residencial*  btn\_resPresc.move(btn\_resPresc.x, btn\_resPresc.y + 108);  btn\_resPresc.enabled = **false**;  *//Botao Simulacao Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y + 108);  btn\_resSimul.enabled = **false**;  *//Componentes do Box*  box\_1.move(box\_x, box\_Envoltorio\_1); *//Envoltorio*  ComPrescEnv.move(box\_x + 100, box\_Envoltorio\_1); *// texto*  ComPrescEnvCmap.move(box\_x + 115, box\_Envoltorio\_1);*//cmap*  box\_2.move(box\_x, box\_Envoltorio\_1 + 16.85); *//Iluminacao*  ComPrescIlum.move(box\_x + 100, box\_Envoltorio\_1 + 16.85); *//texto*  ComPrescIlumCmap.move(box\_x + 115, box\_Envoltorio\_1 + 16.85);*//cmap*  box\_3.move(box\_x, box\_Envoltorio\_1 + 33.7); *//Condicionamento*  ComPrescCondic.move(box\_x + 100, box\_Envoltorio\_1 + 33.7);*//texto*  ComPrescCondicCmap.move(box\_x + 115, box\_Envoltorio\_1 +33.7);*//cmap*  box\_4.move(box\_x, box\_Envoltorio\_1 + 50.55); *//Bonificacao*  ComPrescBonifCmap.move(box\_x + 115, box\_Envoltorio\_1 +50.55);*//cmap*  box\_5.move(box\_x, box\_Envoltorio\_1 + 67.4);  box\_1.visible = **true**;  ComPrescEnv.visible = **true**;  ComPrescEnvCmap.visible = **true**;  box\_2.visible = **true**;  ComPrescIlum.visible = **true**;  ComPrescIlumCmap.visible = **true**;  box\_3.visible = **true**;  ComPrescCondic.visible = **true**;  ComPrescCondicCmap.visible = **true**;  box\_4.visible = **true**;  ComPrescBonifCmap.visible = **true**;  box\_5.visible = **true**;  *//Move Linhas*  vRuleVertical3.height = vRuleVertical3.height + 107;  vRuleVertical1.height = vRuleVertical1.height + 107  hRuleHorizontal1.move(hRuleHorizontal1.x,hRuleHorizontal1.y + 108);  hRuleHorizontal2.move(hRuleHorizontal2.x,hRuleHorizontal2.y + 108);  hRuleHorizontal3.move(hRuleHorizontal3.x,hRuleHorizontal3.y + 108);  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y + 108);  vRuleVertical2.move(vRuleVertical2.x,vRuleVertical2.y + 108);  }  **else** {  button.label = **'+'**;  com\_Simul.move(com\_Simul.x, com\_Simul.y - 108);  residencial.move(residencial.x, residencial.y - 108);  res\_Presc.move(res\_Presc.x, res\_Presc.y - 108);  res\_Simul.move(res\_Simul.x, res\_Simul.y - 108);  *//linha\_Res.move(linha\_Res.x, linha\_Res.y - 108);*  *//Botão Simulacao Comercial*  btn\_comSimul.move(btn\_comSimul.x, btn\_comSimul.y - 108);  btn\_comSimul.enabled = **true**;  *//Botao Prescritivo Residencial*  btn\_resPresc.move(btn\_resPresc.x, btn\_resPresc.y - 108);  btn\_resPresc.enabled = **true**;  *//Botao Simul Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y - 108);  btn\_resSimul.enabled = **true**;  *//Componentes do Box*  box\_1.visible = **false**;  box\_2.visible = **false**;  box\_3.visible = **false**;  box\_4.visible = **false**;  box\_5.visible = **false**;  ComPrescEnv.visible = **false**;  ComPrescEnvCmap.visible = **false**;  ComPrescIlum.visible = **false**;  ComPrescIlumCmap.visible = **false**;  ComPrescCondic.visible = **false**;  ComPrescCondicCmap.visible = **false**;  ComPrescBonifCmap.visible = **false**;  *//Move Linhas*  vRuleVertical3.height = vRuleVertical3.height - 107;  vRuleVertical1.height = vRuleVertical1.height - 107  hRuleHorizontal1.move(hRuleHorizontal1.x,hRuleHorizontal1.y - 108);  hRuleHorizontal2.move(hRuleHorizontal2.x,hRuleHorizontal2.y - 108);  hRuleHorizontal3.move(hRuleHorizontal3.x,hRuleHorizontal3.y - 108);  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y - 108);  vRuleVertical2.move(vRuleVertical2.x,vRuleVertical2.y - 108);  }  }  *//SEGUNDO BOTÃO*  **else** **if**(button.id == **"btn\_comSimul"**){  **if**(button.label == **"+"**){  **var** box\_Envoltorio\_1 : Number = 1634.2;  button.label = **"-"**;  *//Componentes Gráficos*  residencial.move(residencial.x, residencial.y + 108);  res\_Presc.move(res\_Presc.x, res\_Presc.y + 108);  res\_Simul.move(res\_Simul.x, res\_Simul.y + 108);  *//linha\_Res.move(linha\_Res.x, linha\_Res.y + 108);*  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **false**;  *//Botao Pres Residencial*  btn\_resPresc.move(btn\_resPresc.x, btn\_resPresc.y + 108);  btn\_resPresc.enabled = **false**;  *//Botao Simul Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y + 108);  btn\_resSimul.enabled = **false**;  *//Componentes do Box*  box\_1.move(box\_x, box\_Envoltorio\_1);  box\_2.move(box\_x, box\_Envoltorio\_1 + 16.85);  box\_3.move(box\_x, box\_Envoltorio\_1 + 33.7);  box\_4.move(box\_x, box\_Envoltorio\_1 + 50.55);  box\_5.move(box\_x, box\_Envoltorio\_1 + 67.4);  box\_1.visible = **true**;  box\_2.visible = **true**;  box\_3.visible = **true**;  box\_4.visible = **true**;  box\_5.visible = **true**;  *//Move Linhas*  vRuleVertical1.height = vRuleVertical1.height + 107;  hRuleHorizontal2.move(hRuleHorizontal2.x,hRuleHorizontal2.y + 108);  hRuleHorizontal3.move(hRuleHorizontal3.x,hRuleHorizontal3.y + 108);  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y + 108);  vRuleVertical2.move(vRuleVertical2.x,vRuleVertical2.y + 108);  }  **else**{  button.label = **"+"**;  residencial.move(residencial.x, residencial.y - 108);  res\_Presc.move(res\_Presc.x, res\_Presc.y - 108);  res\_Simul.move(res\_Simul.x, res\_Simul.y - 108);  *//linha\_Res.move(linha\_Res.x, linha\_Res.y - 108);*  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **true**;  *//Botao Pres Residencial*  btn\_resPresc.move(btn\_resPresc.x, btn\_resPresc.y - 108);  btn\_resPresc.enabled = **true**;  *//Botao Simul Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y - 108);  btn\_resSimul.enabled = **true**;  *//Componentes do Box*  box\_1.visible = **false**;  box\_2.visible = **false**;  box\_3.visible = **false**;  box\_4.visible = **false**;  box\_5.visible = **false**;  *//Move Linhas*  vRuleVertical1.height = vRuleVertical1.height - 107;  hRuleHorizontal2.move(hRuleHorizontal2.x,hRuleHorizontal2.y - 108);  hRuleHorizontal3.move(hRuleHorizontal3.x,hRuleHorizontal3.y - 108);  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y - 108);  vRuleVertical2.move(vRuleVertical2.x,vRuleVertical2.y - 108);  }  }  **else** **if**(button.id == **"btn\_resPresc"**){  **if**(button.label == **"+"**){  **var** box\_Envoltorio\_1 : Number = 1706.3;  button.label = **"-"**;  *//Componentes Gráficos*  res\_Simul.move(res\_Simul.x, res\_Simul.y + 108);  *//Botao Simul Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y + 108);  btn\_resSimul.enabled = **false**;  *//Botão Simulacao Comercial*  btn\_comSimul.enabled = **false**;  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **false**;  *//Componentes do Box*  box\_1.move(box\_x, box\_Envoltorio\_1);  box\_2.move(box\_x, box\_Envoltorio\_1 + 16.85);  box\_3.move(box\_x, box\_Envoltorio\_1 + 33.7);  box\_4.move(box\_x, box\_Envoltorio\_1 + 50.55);  box\_5.move(box\_x, box\_Envoltorio\_1 + 67.4);  box\_1.visible = **true**;  box\_2.visible = **true**;  box\_3.visible = **true**;  box\_4.visible = **true**;  box\_5.visible = **true**;  *//Move Linhas*  vRuleVertical2.height = vRuleVertical2.height + 107;  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y + 108);  }  **else** {  button.label = **"+"**;  *//Componentes Gráficos*  res\_Simul.move(res\_Simul.x, res\_Simul.y - 108);  *//Botao Simul Residencial*  btn\_resSimul.move(btn\_resSimul.x, btn\_resSimul.y - 108);  btn\_resSimul.enabled = **true**;  *//Botão Simulacao Comercial*  btn\_comSimul.enabled = **true**;  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **true**;  box\_1.visible = **false**;  box\_2.visible = **false**;  box\_3.visible = **false**;  box\_4.visible = **false**;  box\_5.visible = **false**;  *//Move Linhas*  vRuleVertical2.height = vRuleVertical2.height - 107;  hRuleHorizontal5.move(hRuleHorizontal5.x,hRuleHorizontal5.y - 108);  }  }  **else** **if**(button.id == **"btn\_resSimul"**){  **if**(button.label == **"+"**){  button.label = **"-"**;  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **false**;  *//Botao Simul Comercial*  btn\_comSimul.enabled = **false**;  *//Botao Prescritivo Residencial*  btn\_resPresc.enabled = **false**;  **var** box\_Envoltorio\_1 : Number = 1739.55;  *//Componentes do Box*  box\_1.move(box\_x, box\_Envoltorio\_1);  box\_2.move(box\_x, box\_Envoltorio\_1 + 16.85);  box\_3.move(box\_x, box\_Envoltorio\_1 + 33.7);  box\_4.move(box\_x, box\_Envoltorio\_1 + 50.55);  box\_5.move(box\_x, box\_Envoltorio\_1 + 67.4);  box\_1.visible = **true**;  box\_2.visible = **true**;  box\_3.visible = **true**;  box\_4.visible = **true**;  box\_5.visible = **true**;  }  **else**{  button.label = **"+"**;  *//Botao Prescritivo Comercial*  btn\_comPresc.enabled = **true**;  *//Botao Simul Comercial*  btn\_comSimul.enabled = **true**;  *//Botao Prescritivo Residencial*  btn\_resPresc.enabled = **true**;  box\_1.visible = **false**;  box\_2.visible = **false**;  box\_3.visible = **false**;  box\_4.visible = **false**;  box\_5.visible = **false**;  }  }  } |