**Estrutura de desenvolvimento do domus web**

1. **Servidor domus**

Hoje são disponibilizado dois dominios dentro da puc para o projeto Domus: <http://domus.pucpr.br> e <http://powerdomus.pucpr.br>. Esses dominios estão configuradas em maquinas que ficam alocadas no 7º andar do prédio administrativo da PUCPR, sobre os cuidados da equipe de rede 2º nível.

* 1. **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

* 1. **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/ –** Ficam todos os arquivos publicados 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 (\*) para especificar uma execução constante. Por exemplo no lugar de dias do mês conter o \*, o comando relacionado será executado todos os dias.

1. **CmapTools**
   1. **Instalação e configuração**
2. Criar na mesma pasta onde está o instalador um arquivo de texto chamado "InstallCmapServer.properties";
3. Modificar as permissões dos arquivos com o comando chmod 775 LinuxCmapServer\_vXX.bin e chmod 775 InstallCmapServer.properties. Obs: O instalador encontra-se no diretório /home/files/programas/LinuxCmapServer\_v5.04\_07-16-10.bin do servidor domus.pucpr.br;
4. 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 usuário **admin** e senha **admin.** O serviço utiliza a porta 81, e pode ser acessada pela a url <HTTP://domus.pucpr.br:81> .

1. **Versionador**
   1. **Instalação e Configuração do versionador no servidor**
2. yum install subversion
3. yum install mod\_dav\_svn
4. Criando o repositório:
   1. mkdir /home/www/svn/repos  
      svnadmin create /home/www/svn/repos/domus
5. Alterar a propriedade da pasta para o apache:
   1. chown -R apache.apache /home/www/svn
6. Criar arquivo /home/www/svn/repos/domus/svnauth com os usuários autorizados para acessar o repositório. Conteúdo do arquivo:
   1. 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

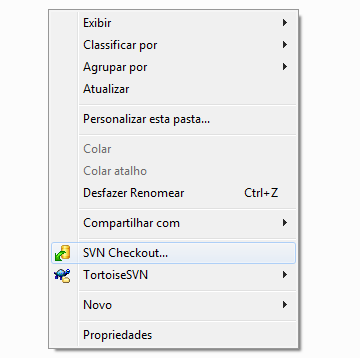
1. Criar o arquivo /home/www/svn/repos/domus/svnpass
   1. htpasswd -bcm /home/www/svn/repos/domus/svnpass user1 passwordUser1   
      htpasswd -bm /home/www/svn/repos/domus/svnpass user2 passwordUser2

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 restart do apache com o comando httpd restart.
   1. **Configurando no ambiente de desenvolvimento**

Configurando SVN Tortoise

Primeiro baixar o 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 “SVN Checkout...”

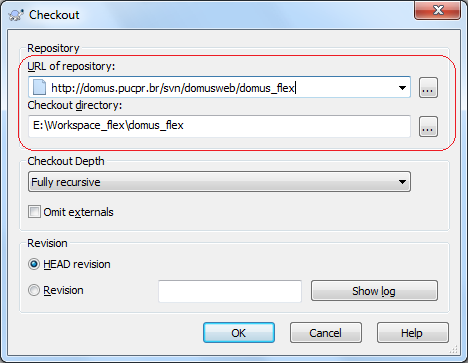


Digite a URL do Repositório aonde se encontra os arquivos.

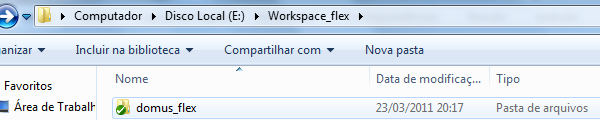
Fonte dos arquivos Flex: <http://domus.pucpr.br/svn/domusweb/domus_flex>

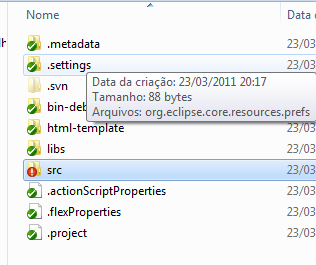
Fonte dos arquivos PHP: <http://domus.pucpr.br/svn/domusweb/domus_web>

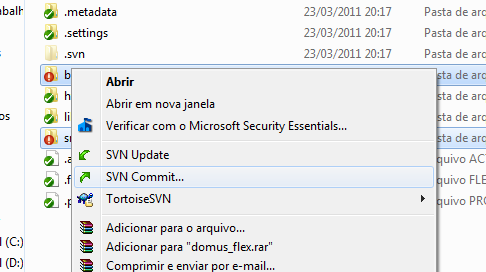
Fonte com documentações: <http://domus.pucpr.br/svn/domusweb/domus_docs>

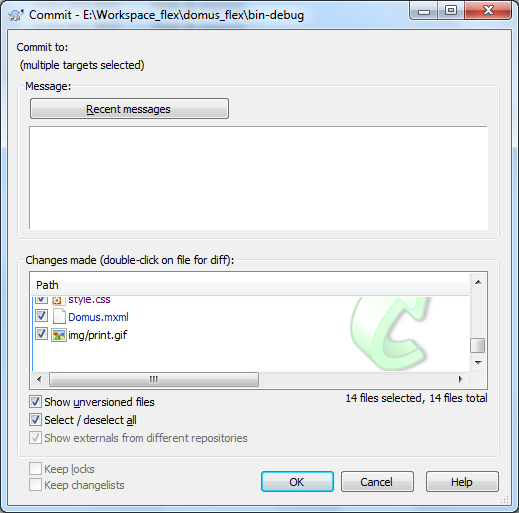


Após baixar os arquivos é só editá-los. Quando houver alguma alteração a imagem do diretório se altera ficando vermelho e com o símbolo “!”. Para subir está versão para o servidor é só selecionar o arquivo(s) ou diretório(s) clicar com o botão direito e dar “Commit”, conforme mostrado abaixo:



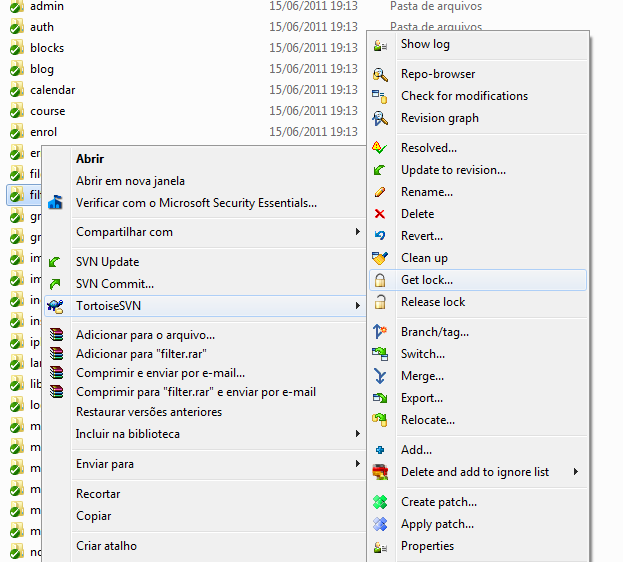






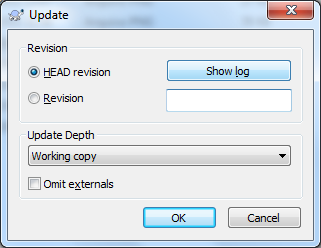
Existe outras opções que devem ser seguidas para que seja realizada a alteração no servidor também ao dar commit:

* Em alteração de nome do arquivo ou diretório utilizar a opção “Rename...”
* “Ao adicionar novo diretório ou arquivo o mesmo irá mostar um icone com o simbolo”?”(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”

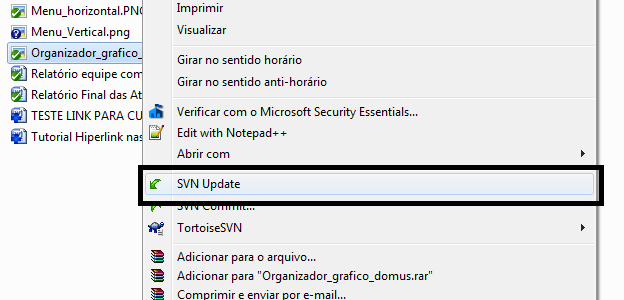


Para desfazer alterações que realizou em um arquivo ou diretório é só selecionar a opção “Revert” que ele atualizará os arquivos para a versão atual que está no servidor.

Para voltar uma versão anterior 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.



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



Outra observação importante é configurar o arquivo “config.php” no raiz do moodle para as configurações do seu computador, este arquivo não precisa fazer commit.

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

1. **Moodle**
   1. **Estruturas modificada**

**/blocks/menu\_curso/** – Diretório onde fica a programação do menu principal flayout ;

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

**/include/js/** – Encontra-se os arquivos JavaScripts usado em todo o projeto;

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

**/lib/** – Diretório de bibliotecas usada em todo o projeto;

**/lib/javascript.php** – Arquivo onde adiciona no header os javascripts 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;

* 1. **Módulos desenvolvidos/alterados**

**Estrutura de cursos:** Foi elaborada uma classe responsável em montar a estrutura dos cursos criados no domus. Nela é respeitada a hierarquia e as ordens definidas no admin pelo o usuário. A classe fica localizada no arquivo **/util/cursos\_domus.php** .

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 admin;

|  |  |  |
| --- | --- | --- |
| **$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  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 | |  |

**function** IsLink**(**$name**)**

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

**function** Concat**()**

* Método retorna duas arrays 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 Admin;
* 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 flayout principal. É criado uma variável e os métodos javascript(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 admin;
* Para iniciar o menu flay-out na biblioteca a variável deve ter a seguinte estrutura:

**$cores**=’#dedede','#c0c0c0','#dedede','#409db5','#409db5','#cdcdcd','','#c0c0c0','','','','','#409db5','#ffffff','#ffffff','#409db5','#409db5'

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','#','','','',0,0,

0,0,'G','G','#','','','',0,0,

0,1,'H','H','#','','','',0,0,

1,2,$cores,

0,2,'B','B','#','','','',0,0,

0,0,'C','C','#','','','',0,0,

2,3,$cores,

0,0,'D','D','#','','','',0,0,

0,0,'E','E','#','','','',0,0,

0,0,'F','F','#','','','',0,0,

3,1,$cores,

0,0,'I','I','#','','','',0,0];

Irá imprimir o menu na estrutura abaixo:

**I**

**H**

**G**

**F**

**E**

**A**

**B**

**D**

**C**

**Menu vertical principal:**  dentro do arquivo /blocks/menu\_curso/block\_menu\_curso.php no método **get\_content(),** é instanciado o objeto CursosDomus e chamado o método ImprimirMenu(). O arquivo cursos\_domus.php está inserido no arquivo de configuração do projeto **/config.php**;

**Menu horizontal:** está localizando dentro do arquivo **/theme/menu.php.** Ele é fixo e é montado via jQuery;