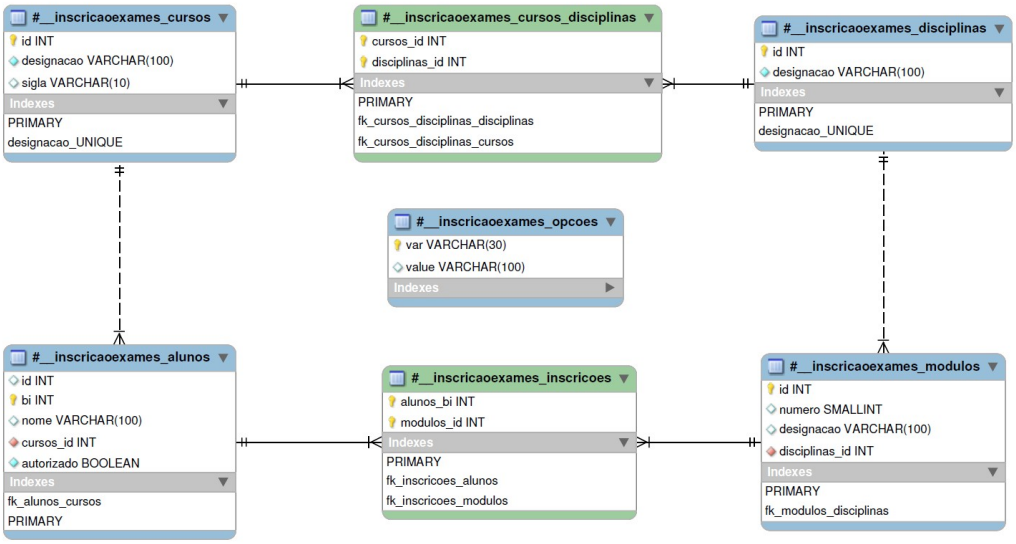
com_inscricaoexames

Short technical description

It's a Joomla extension used in Portuguese K12 schools to allow for students to enroll in internal exams. Most of the class/method/var names are in Portuguese because this was made in a Portuguese context but my development capabilities are not limited to programming this way.

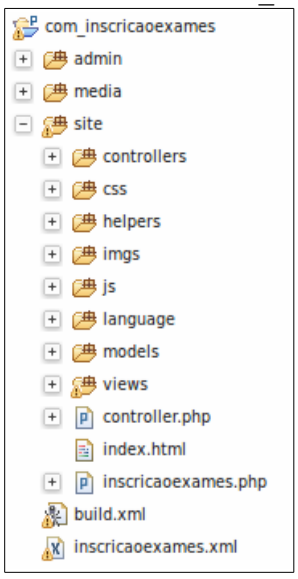
Database

Database logic is based on courses (cursos), each one with different subjects (disciplinas), and each one with different themes (modulos). A student (alunos) belongs to a course and can enroll (inscricoes) in exams in several subjects. The database E/R model is as follows:



The extension consists of a component which will do almost all the work, and a plugin which is only used to delete a user from the application database when that Joomla user is deleted (not included in code sample). So when you decompress the code sample zip file, you end up with the structure below.

The folder `com_inscricaoexames` is the main component folder, which has 2 other folders:



- **`site`**: contains all the core front-end code, being the most important:
  - **`/models`**: contains all the model classes, related to database data manipulation;
  - **`/views`**: contains the UI code, like forms and data reports (`view.html.php`). It also contains the code related to the AJAX responses (`view.raw.php`);
  - **`/controllers`**: contains specific controlers, in this case only one, which was not mandatory, but I wanted to try out some stuff;
  - **`controller.php`**: Main application controller;
  - **`inscricaoexames.php`**: entry point for the application. It only instantiates framework code.
- **`admin`**: contains the code for the Joomla back-end related to this extension. This code consists of the object to DB table mapping and some options related to the functioning of the extension which only the site admin should set up.

TODO

- **Security related issues**
  - Filter data
    - In the controllers, filter the data sent from the view forms, so that it is what's expected to be and does not ruin the database;
    - This will prevent the following attacks:
      - Spoofed form
      - SQL Injection
  - Escape data
    - In each view report, escape the data shown, so that no undesired javascript is output to the browser;
    - This will prevent the following attacks:
      - Cross Site Scripting (XSS)
  - Assign random ID to each form
    - In each view form, generate a token ID that will be checked when data is received in the controller, so to attest that it was a legitimate form that sent the data;
    - This will prevent the following attacks:
      - Cross Site Request Forgery

Example of a running use

updating course data

1. You start in a form that shows all the course data. That form is generated by the view file:
   - `site/views/cursos/view.html.php` which will use the template file:
   - `site/views/cursos/tmpl/editarcursos.php`

```php
<?php
// No direct access to this file
defined('_JEXEC') or die('Restricted access');

// import Joomla view library
jimport('joomla.application.component.view');

/**
 * Classe used to show a view
 */
class inscricaoexamesViewcursos extends JView{

    function display($tpl = null){
        //choose the layout and assign data to the view
        switch(JRequest::getVar('layout')){
            case 'editarcursos':
                $document = JFactory::getDocument();
                $document->addScript( JURI::base().'/components/com_inscricaoexames/js/cursos.js' );
                // Get data from the model
                $items = $this->get('Cursos');
                $this->assignRef('items',$items);
            break;

            case 'editardisccursos':
                //contar os cursos
                $modelCursos = $this->getModel('cursos');
                $this->numCursos = $modelCursos->countCursos();
                if($this->numCursos>0){
                    //contar as disciplinas
                    $this->setModel(JModel::getInstance('disciplinas', 'inscricaoexamesModel'));
                    $modelDisciplinas = $this->getModel('disciplinas');
                    $this->numDisc = $modelDisciplinas->countDisciplinas();
                    if($this->numDisc>0){
                        $document = JFactory::getDocument();
                        $document->addScript( JURI::base().'/components/com_inscricaoexames/js/cursos.js' );
                        // Get data from the model
                        $cursos = $this->get('Cursos');
                        $this->cursos=$cursos;
                        $disciplinas=$modelDisciplinas->getDisciplinas();
                        $this->disciplinas=$disciplinas;
                    }
                }
            break;
        }
        // Check for errors.
        if (count($errors = $this->get('Errors'))){
            JError::raiseError(500, implode('<br />', $errors));
            return false;
        }
        // Display the view
        parent::display($tpl);
    }
}
```


```php
<?php
// no direct access
defined('_JEXEC') or die('Restricted access');
?>
<H1><?php echo JText::_( 'VER_CURSOS' ); ?></H1>
<form id="adminForm" name="adminForm" action="index.php" method="post" onsubmit="return checkData_editarcursos('document.getElementById('task').value', '<?= JText::_('ERRO_NENHUM_CURSO_SELECCIONADO') ?>');">
    <div id="editcell">
        <table class="adminlist" width="100%">
        <thead>
                <tr align="center">
                        <th width="30"><?php echo JText::_( 'ID' ); ?></th>
                        <th width="30"><input type="checkbox" name="toggle" value="" onclick="checkAll(<?php echo count( $this->items ); ?>);" /></th>
                        <th width="100"><?php echo JText::_( 'SIGLA' ); ?></th>
                        <th><?php echo JText::_( 'NOME_DO_CURSO' ); ?></th>
                </tr>
        </thead>
        <?php
        $k = 0;
        for ($i=0, $n=count( $this->items ); $i < $n; $i++)
        {
                $row = $this->items[$i];
                $checked        = JHTML::_('grid.id',   $i, $row->id );
                ?>
                <tr class="<?php echo 'row'.$k; ?>">
                        <td align="center"><input type="hidden" name="cursos[<?php echo $i; ?>][id]" value="<?php echo $row->id; ?>" /><?php echo $row->id; ?></td>
                        <td align="center"><?php echo $checked; ?></td>
                        <td align="center"><input type="text" name="cursos[<?php echo $i; ?>][sigla]" size="10" value="<?php echo $row->sigla; ?>"/></td>
                        <td align="center"><input type="text" name="cursos[<?php echo $i; ?>][designacao]" size="60" value="<?php echo $row->designacao; ?>"/></td>
                </tr>
                <?php
                $k = 1 - $k;
        }
        ?>
        </table><BR>
        <input type="submit" name="Submit" class="button" value="<?php echo JText::_( 'ELIMINAR_CURSOS' ); ?>" onclick="document.getElementById('task').value='eliminar_cursos'">
        <input type="submit" name="Submit" class="button" value="<?php echo JText::_( 'GUARDAR_ALTERACOES' ); ?>" onclick="document.getElementById('task').value='actualizar_cursos'">
    </div>

    <input type="hidden" name="option" value="com_inscricaoexames" />
    <input type="hidden" id="task" name="task" value="" />
    <input type="hidden" name="boxchecked" value="0" />

</form>
```

2. In that form you make some changes and press the submit button (GUARDAR_ALTERACOES). This will send all data to Joomla, including the following:
    - `<input type="hidden" name="option" value="com_inscricaoexames" />`
    - `<input type="hidden" id="task" name="task" value="actualizar_cursos" />`

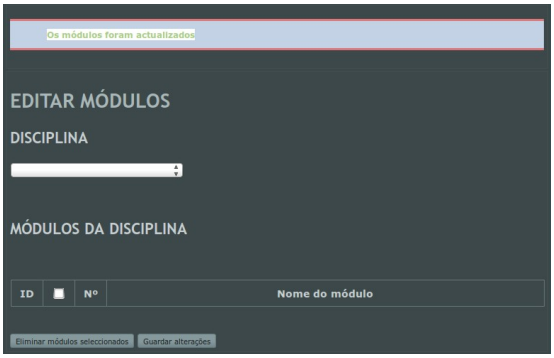
3. This will tell Joomla to handle the request with the extension `com_inscricaoexames` and will tell its controller to run the method `actualizar_cursos`.

```php
<?php

jimport('joomla.application.component.controller');

/**
 * Component Controller
 * @author Herberto Graça
 *
 * @package     inscricaoexames
 */
class inscricaoexamesController extends JController
{

    /**
     * updates data in several courses
     * @author Herberto Graça
     *
     */
    public function actualizar_cursos(){

        $lista_de_cursos=JRequest::getVar('cursos');

        $model = $this->getModel('cursos');

        if (!$model->insCursos($lista_de_cursos)){
            $msg=$model->getError();
            $msgType="Error";
        }
        else{
            $msg=JText::_( 'CURSOS_INSERIDOS' );
            $msgType="Message";
        }

        $link = 'index.php?option=com_inscricaoexames&view=cursos&layout=editarcursos';
        $this->setRedirect($link, $msg, $msgType);
    }
```

4. The controller (`site/controller.php`) runs that method which uses the relevant model (model cursos, in `site/models/cursos.php`) to insert the received data into the database.

```php
<?php
// No direct access to this file
defined('_JEXEC') or die('Restricted access');

// import Joomla modelitem library
jimport('joomla.application.component.modelitem');

class inscricaoexamesModelcursos extends JModelItem{

    /**
     * Returns a reference to the a Table object, always creating it.
     */
    public function getTable($type = 'cursos', $prefix = 'inscricaoexamesTable', $config = array()) {

    /**
     * Inserts data of a list of courses into the DB
     */
    public function insCursos($lista_de_cursos){
        $erro="";
        $out=true;


        foreach($lista_de_cursos as $curso) {


            if(!$this->insCurso($curso['id'], trim($curso['designacao'],'\r'), $curso['sigla'])){
                $erro.="\n<BR>".$this->getError();
                $out=false;
            }
        }
        if (!$out) $this->setError(JText::_( 'ERRO_INS_CURSOS' ).$erro);
        return $out;
    }

    /**
     * Inserts a course data into the DB
     */
    private function insCurso($id, $designacao, $sigla){

        $row= $this->getTable();
        $row->set('id',$id);
        $row->set('designacao',$designacao);
        $row->set('sigla',$sigla);


        // Make sure the record is valid
        if (!$row->check()) {
            $this->setError($row->getError());
            return false;
        }

        // Store the data in the database
        if (!$row->store()) {
            $this->setError($row->getError());
            return false;
        }

        return true;
    }
```

5. Finally the controller outputs the initial view with the initial template, and the return message stating success or error.



Herberto Graça