

Todos los archivos listados a continuación se encuentran en el repositorio

<https://github.com/Hikendo/modMoodle>

El archivo config.php va en la raíz del Moodle se modifica la línea 30, agregamos el siguiente código:

```
//configura nuestras rutas de ingreso alternas $CFG->alternateloginurl, Las  
separamos por una coma y la primera se tomara como base, usa rutas fuera de  
moodle  
//$CFG->alternateloginurl = 'http://127.0.0.1:8081/LoginSample.html,  
http://127.0.0.1:8081/LoginSample1.html,  
http://127.0.0.1:8081/LoginSample2.html';  
  
//configura nuestras rutas de ingreso alternas $CFG->alternateloginurl, Las  
separamos por una coma y la primera se tomara como base  
//Si las definimos dentro de moodle las podemos usar como rutas relativas  
  
$CFG->alternateloginurl = '/general/, /cursos/, /examen/';  
  
//Nuestros roles que se usaran para redirigir a su correspondiente  
ruta ejemplo '/cursos/' se traduce como http://localhost:8080/cursos/  
$CFG->alternatelogouturl = array(  
    'student' => '/cursos/',  
    'teacher' => '/examen/'  
);  
  
//Nuestra ruta default para cualquier rol no definido en nuestros roles de  
redireccion  
$CFG->alternateLogoutUrlDefaultRol = '/general/';
```

El archivo index va en el directorio login se modifica a partir de la linea 329, también podemos modificar el archivo /login/index.php después de la linea 328, para que quede de la siguiente manera:

```
/// Redirect to alternative login URL if needed
if (!empty($CFG->alternateloginurl) && $loginredirect) {
    //La instruccion original
    // $loginurl = new moodle_url($CFG->alternateLoginurl);

    //Inicia Bloque para que la redireccion del sitio tome la primera ruta
    //tomamos nuestras rutas de $CFG->alternateloginurl y generamos un array
    al usar ',' como divisor
    $urls = explode(',', $CFG->alternateloginurl);
    //enviamos el primer argumento como ruta default
    $url=trim($urls[0]);
    // $url='http://127.0.0.1:8081/loginSample.html';
    $loginurl = new moodle_url($url);
    //Aquí termina el bloque
    $loginurlstr = $loginurl->out(false);

    if ($SESSION->wantsurl != '' && strpos($SESSION->wantsurl, $loginurlstr)
=== 0) {
        // We do not want to return to alternate url.
        $SESSION->wantsurl = null;
    }
}
```

El archivo logout va en el directorio login se modifica todo el archivo, para ignorar otros plugins de redireccionamiento. Su contenido queda de la siguiente manera:

```
require_once(__DIR__ . '/../config.php');
require_once($CFG->libdir . '/authlib.php');
require_once($CFG->libdir . '/datalib.php');
require_once($CFG->dirroot . '/user/lib.php');

// Obtener el id del usuario
$userid = $USER-> id;
// Obtener todos los cursos en los que el usuario está inscrito.
$user_courses = enrol_get_users_courses($userid);

// Define las URLs de redirección según el rol obtenidos del archivo
config.php en la raíz
$redirect_urls = $CFG->alternatelogouturl;
// Recorrer cada curso y obtener los roles del usuario en ese curso.
foreach ($user_courses as $course) {
    //verifica el rol del usuario por el id del curso
    $context = context_course::instance($course->id);
    $roles = get_user_roles($context, $userid);
    // Si se encontraron roles, se recorren para buscar una coincidencia con
    nuestra lista de redirección
    if (!empty($roles)) {
        // Redirige según el rol del usuario
        foreach ($roles as $role) {
            if (isset($redirect_urls[$role->shortname])) {
                $redirect_url = $redirect_urls[$role->shortname];
                break;
            }
        }
    }
}

// Si no se encuentra un rol específico, redirige a una página por defecto
definido en el archivo config de la raíz
if (!isset($redirect_url)) {
    $redirect_url=$CFG->alternateLogoutUrlDefaultRol;
}

//destruir la sesion
require_logout();
// Redirige al usuario
redirect($redirect_url);
```

Cron Moodle.

La configuración la encontramos en la pagina oficial:

<https://docs.moodle.org/all/es/Cron>

Debemos encontrar la ruta de nuestro archivo cron Moodle, algo similar a:

`/usr/bin/php /ruta/a/moodle/admin/cli/cron.php`

En la terminal de comandos Linux ejecutar el comando:

```
crontab -u www-data -e
```

podemos cambiar al usuario www-data dependiendo de el tipo de servidor que usemos, generalmente deberá ser el que ejecuta las tareas web, ejemplo “apache”, etc.

El cron debe ejecutarse regularmente, generalmente lo hace ca minuto, por lo que agregaríamos la instrucción a nuestras tareas del cron tab con la linea:

```
* * * * * /usr/bin/php /path/to/moodle/admin/cli/cron.php >/dev/null
```