



Manual para desarrolladores.

Evaluación de programación para medir el estilo de codificación.

Documento elaborado por: Isaías Caamal García.

Lugar y fecha: Cancún, Quintana Roo a 5 de octubre de 2016.

Contenido

Objetivo.....	3
Características	4
Base de datos	4
Diagrama	4
Interfaz gráfica	4
Inicio de sesión.....	4
Registro de nuevo usuario	5
Estructura	5
Rutas.....	5
Conexión a base de datos	6
Controladores.....	6
Modelos.....	7
Vistas	7

Objetivo

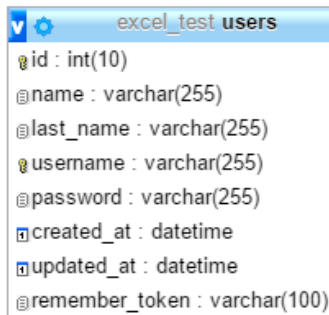
El presente documento tiene como objetivo explicar al usuario desarrollador un inicio de sesión, en el lenguaje PHP con conexión a base de datos a través del gestor MySQL; utilizando el patrón arquitectónico MVC.

Características

- Acceder al sistema a través de un nombre de usuario y una contraseña.
- Registrar un nuevo usuario, mediante un formulario.

Base de datos

Diagrama

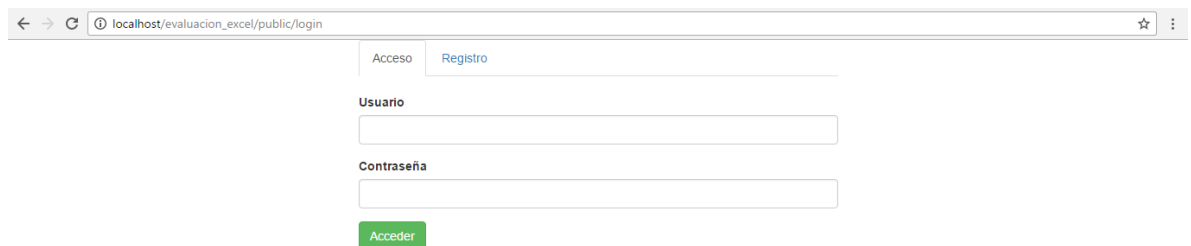


excel_test users	
id	int(10)
name	varchar(255)
last_name	varchar(255)
username	varchar(255)
password	varchar(255)
created_at	datetime
updated_at	datetime
remember_token	varchar(100)

Ilustración 1 Diagrama de la base de datos

Interfaz gráfica

Inicio de sesión



The screenshot shows a web browser window with the address bar displaying `localhost/evaluacion_excel/public/login`. The page has two tabs: 'Acceso' (selected) and 'Registro'. Below the tabs, there are two input fields labeled 'Usuario' and 'Contraseña'. A green 'Acceder' button is positioned below the password field.

Ilustración 2 Vista de formulario de inicio de sesión

Esta ventana cuenta con dos campos de texto para introducir el nombre de usuario y la contraseña, un botón para acceder, y una pestaña para dirigirse al formulario de registro.

Registro de nuevo usuario

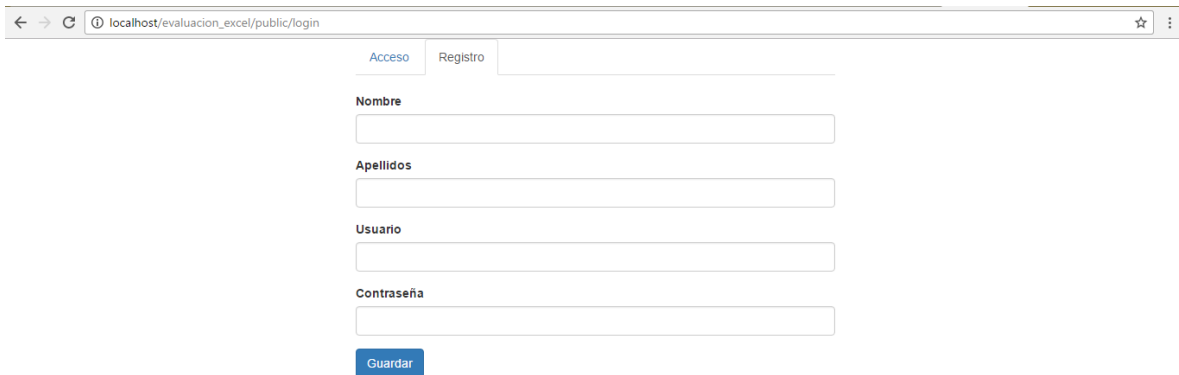


Ilustración 3 Vista de formulario de registro

Esta ventana consta de cuatro campos de texto (Nombre, Apellidos, Usuario y Contraseña), un botón para guardar los datos y la pestaña de Acceso al sistema.

Estructura

Rutas

Esta sección consta basado en el patrón arquitectónico MVC, de las rutas que utiliza el sistema para realizar las diferentes comunicaciones entre el cliente y el servidor.

Se encuentran ubicadas en el archivo `app/routes.php`.

`<?php`

```
Route::get('login', 'Login@show');
Route::post('login', 'Login@login');
Route::get('logout', 'Login@logout');

Route::post('register', 'Login@register');

Route::group(['before' => 'auth'], function()
{
    Route::get('/', 'HomeController@showWelcome');
});
```

Conexión a base de datos

Esta se realiza mediante la capa de abstracción PDO, hacia el gestor MySQL. Para ello realizamos la configuración dentro del fichero app/config/database.php. Donde establecemos parámetros de host, nombre de la base de datos, usuario y contraseña.

```
'connections' => array(
    'mysql' => array(
        'driver' => 'mysql',
        'host' => 'localhost',
        'database' => 'excel_test',
        'username' => 'root',
        'password' => '',
        'charset' => 'utf8',
        'collation' => 'utf8_unicode_ci',
        'prefix' => '',
    ),
),
```

Controladores

El controlador Login se encarga del inicio de sesión/registro, que como indica el patrón MVC se encarga de realizar las peticiones en el servidor hacia el modelo User, y corresponde a las rutas establecidas a través de sus diferentes métodos. Esta clase se ubica en app/controllers/Login.php. Su constructor contiene el objeto de la clase User para hacer las peticiones correspondientes a la entidad que corresponde en la base de datos.

```
class Login extends BaseController
{
    protected $user;

    public function __construct(User $user)
    {
        $this->user = $user;
    }

    public function show()
    {
    }

    public function login()
    {
    }

    public function logOut()
    {
    }

    public function register()
    {
    }
}
```

El controlador HomeController, se encarga de manipular la vista principal del sistema.

```
class HomeController extends BaseController {  
  
    public function showWelcome()  
    {  
        return View::make('hello');  
    }  
}
```

Modelos

Dado que la base de datos, cuenta únicamente con una tabla, solo se hace uso de un Modelo nombrado User, el cual se ubica en app/models/User.php, este hace la relación con dicha tabla, estableciendo campos que pueden ser rellenados fácilmente al crear un objeto de esta clase, para tomarlos de un arreglo enviado a través de una petición http. Así mismo establece reglas para cada campo de la entidad, utilizados en la validación previa a una inserción o posible modificación de un registro.

```
class User extends BaseModel implements UserInterface,  
RemindableInterface {  
  
    use UserTrait, RemindableTrait;  
    protected $table = 'users';  
  
    public $fillable = ['name', 'last_name', 'username', 'password'];  
  
    public static $rules = [  
        'name' => 'required|string|min:3',  
        'last_name' => 'required|string|min:3',  
        'username' => 'required|unique|string|min:3',  
        'password' => 'required|string|min:8'  
    ];  
  
    /**  
     * The attributes excluded from the model's JSON form.  
     *  
     * @var array  
     */  
    protected $hidden = array('password', 'remember_token');  
}
```

Vistas

Esta sección contiene los archivos HTML ubicados en la ruta app/views/ en los cuales se encuentran los formularios de inicio de sesión, registro de usuario y vista principal. De igual forma se contemplan dentro de esta rama los ficheros de estilo(CSS) con ruta public/css/ y scripts con la ruta public/js. Estos últimos funcionan con gran importancia ya que hacen la validación previa de campos del formulario a través del plugin JQuery y su método validate(), y tras un resultado válido la petición Ajax correspondiente; como se muestra en el siguiente código:

```
$(document).ready(function() {  
  
    $('form').each(function() {  
        $(this).validate({  
            rules: {  
                name: { required: true, minlength: 3},  
                last_name: { required: true, minlength: 2},  
                username: { required: true, minlength: 3},  
                pass: { required: true, minlength: 8}  
            },  
            messages: {  
                name: "Debe introducir su nombre. Mínimo 3 caracteres",  
                last_name: "Debe introducir su apellido(s). Mínimo 3 caracteres",  
                username: "Debe introducir su nombre de usuario. Mínimo 3  
caracteres",  
                pass: "Debe introducir su contraseña. Mínimo 8 caracteres.",  
            },  
            errorClass: "error",  
            validClass: "success",  
            focusInvalid: true,  
            submitHandler: function(form){  
                var dataForm = form.serialize();  
                $.ajax({  
                    type: "POST",  
                    url: form.attr('action'),  
                    data: dataForm,  
                    success: function(data){  
                        console.log('enviado');  
                    }  
                });  
            }  
        });  
    })  
});  
});
```