

Login multiusuarios

Desarrollo backend -
Laravel 8

1 - Configuración de proyecto

1. Proyecto nuevo con laravel 8
2. Configuración de base de datos

1 - Configuración de proyecto

1. Creación de vistas y carpetas

- a. modules
 - i. cliente
 - 1. index.blade.php
 - ii. admin
 - 1. index.blade.php
 - iii. auth
 - 1. login.blade.php
- b. shared
- c. layouts
 - i. main_cliente.blade.php
 - ii. main_admin.blade.php

2 - Creación de controladores

- `php artisan make:controller Cliente`
- `php artisan make:controller Admin`
- `php artisan make:controller Auth`

3 - Creación de rutas en web

- `Route::get('/', [Auth::class, 'index'])->name('login');`
- `Route::post('/logear', [Auth::class, 'logear'])->name('logear');`
- `Route::get('/logout', [Auth::class, 'logout'])->name('logout');`
- `Route::get('/nuevoUsuario', [Auth::class, 'agregarNuevo']);`
- ❖ `Route::get('/cliente', [Cliente::class, 'index'])->name('cliente_index');`
- ❑ `Route::get('/admin', [Admin::class, 'index'])->name('admin_index');`

Creamos los métodos correspondientes a cada una de las rutas dentro de su respectivo controlador.

4 - Damos de alta la migración de user

- Agregamos el rol de usuario : `$table->string('rol');`
- En el modelo User agregamos rol en la propiedad `$fillable`
- Damos de alta la migración **php artisan migrate**
- Una vez creada la tabla editamos el campo de email para que no sea obligatorio

5 - Creamos un usuario válido

- Dentro del controlador auth creamos el siguiente metodo
- ```
➤ public function agregarNuevo () {
➤ $item = new User();
➤ $item->name = 'cliente';
➤ $item->password = Hash::make('12345');
➤ $item->rol = 'cliente';
➤ $item->save();
➤ return $item;
➤ }
```

Levantamos el servidor y en el navegador vamos a la ruta correspondiente.

## 6 - Configuraciones extra

- El middleware de autenticación debe tener una ruta a la cual regresar si no existe autenticación.
- app/http/Middleware/Authenticate.php
- 

```
protected function redirectTo($request)
{
 if (! $request->expectsJson()) {
 return route('login');
 }
}
```



## 7 - Creamos la vista que nos servirá como login

### Login de usuario

Usuario

roldan

Password

.....

Entrar

Por convención es bueno que usuario y password tengan como name el mismo nombre en la migración

## 8 - Crear método para salir en auth

```
public function logout() {
 Auth::logout();
 Session::flush();
 return redirect()->route('login');
}
```

## 9 - Crear método para logearse en Auth

- `public function logear(Request $request) {`
- `$credenciales = $request->only("name", "password");`
- `if (Auth::attempt($credenciales)) {`
- `if ( auth()->user()->rol == 'admin') {`
- `return redirect()->route('admin_index');`
- `} else if(auth()->user()->rol == 'cliente') {`
- `return redirect()->route('cliente_index');`
- `} else {`
- `return $this->logout();`
- `}`
- `} else {`
- `return back()->withInput($credenciales);`
- `}`
- `}`

## 10 - Configuramos la vista de login

```
<h2>Login de usuario</h2>
<form action="{ route('logear') }" method="post">
 @csrf
 <label for="name">Usuario</label>
 <input type="text" name="name" id="name" class="form-control">
 <label for="password">Password</label>
 <input type="password" name="password" id="password" class="form-control">
 <button class="btn btn-primary mt-3">
 Entrar
 </button>
</form>
```

# 11 - Creamos los middleware

## 2 - Creación de middleware