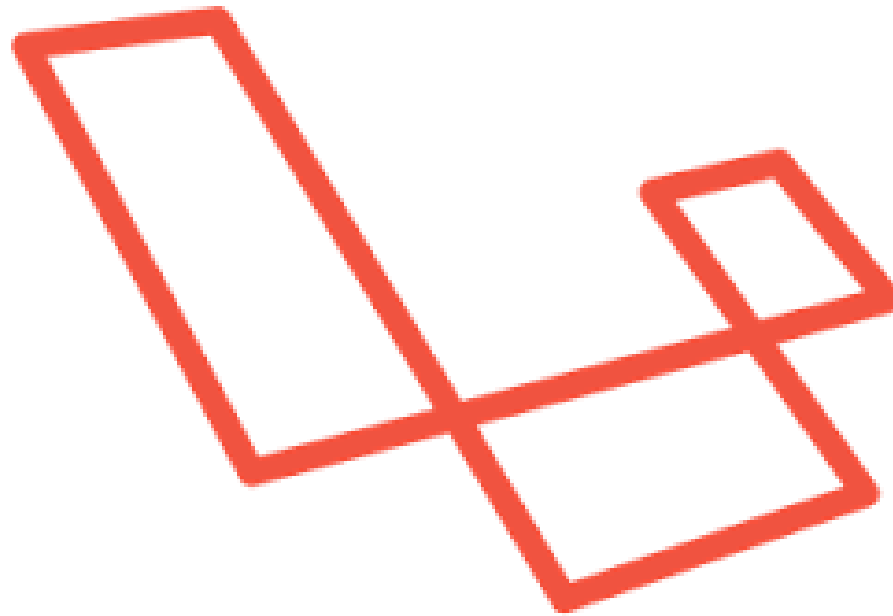


Curso de Laravel



laravel

Capítulo 8: Datos de entrada y control de usuarios

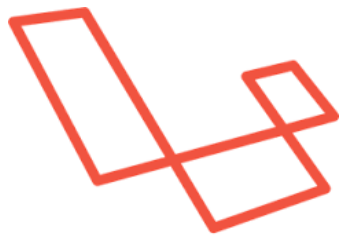


Laravel: Datos de entrada

Datos de Entrada HTTP

- Laravel provee para manejar los datos de entrada la clase Request, la que es utilizada mediante inyección de dependencias añadiéndola al constructor o método a usar

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Routing\Controller;
class UserController extends Controller
{
    public function store(Request $request)
    {
        $name = $request->input('nombre');
    }
}
```



laravel

Laravel: Datos de entrada

Datos de Entrada HTTP

- Utilizando Request, añadiendo más parámetros en la llamada al método

```
public function store(Request $request, $id)
{
    // ...
}
```



Laravel: Datos de entrada

Obtener Datos de Entrada

- Para obtener el valor de una variable

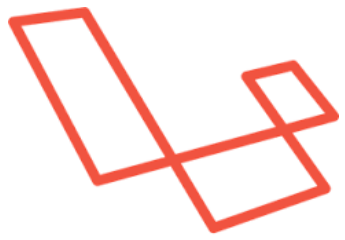
```
$name = $request->input('nombre');  
// o bien  
$name = $request->nombre;
```

- Definir un valor por defecto

```
$name = $request->input('nombre', 'juan');
```

- Obtener datos desde un arreglo (checkbox u otros)

```
$input = $request->input('productos.0.nombre');
```



laravel

Laravel: Datos de entrada

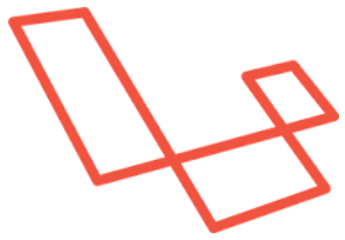
Obtener Datos de Entrada

- Comprobar si una variable existe

```
if($request->has('nombre')){  
    // ..  
}
```

- Obtener datos agrupados (el método retorna un array)

```
//Obtener todos:  
$input = $request->all();  
//Obtener solo los campos indicados:  
$input = $request->only('username', 'password');  
//Obtener todos excepto los indicados:  
$input = $request->except('credit_card');
```



laravel

Laravel: Datos de entrada

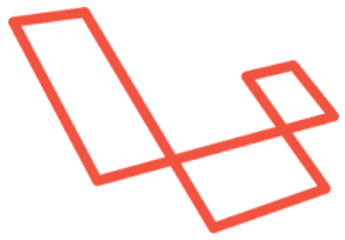
Archivos de entrada

- Archivos de entrada

```
$file = $request->file('archivo');  
//o bien  
$file = $request->archivo;
```

- Comprobar si un archivo es válido

```
if($request->file('archivo')->isValid()){  
    // ...  
}
```



laravel

Laravel: Datos de entrada

Archivos de entrada

- Mover archivos de entrada

```
//Mover el archivo a la ruta:  
$request->file('archivo')->move($ruta_de_destino);  
  
//Mover el archivo a la ruta con un nuevo nombre:  
$request->file('archivo')->move($ruta_de_destino, $nombre);
```

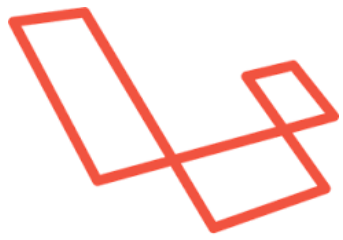


Laravel: Datos de entrada

Archivos de entrada

- Obtener información del archivo de entrada

```
//Obtener la ruta:  
$path = $request->file('file')->getRealPath();  
  
//Obtener el nombre original:  
$name = $request->file('file')->getClientOriginalName();  
  
//Obtener la extensión:  
$extension = $request->file('file')->getClientOriginalExtension();  
  
//Obtener el tamaño:  
$size = $request->file('file')->getSize();  
  
//Obtener el MIME Type:  
$mime = $request->file('file')->getMimeType();
```

laravel

Laravel: Control de usuarios

La clase Auth

- Laravel tiene clases y métodos para facilitar la gestión de usuarios
- Las configuraciones por defecto se encuentran en `config/auth.php`
- El modelo de datos de usuario `User.php` se encuentra en el directorio `app` con toda la implementación necesaria
- Los controladores encargados de gestionar todas las acciones del control de usuarios se encuentran en `app\Http\Controllers\Auth`



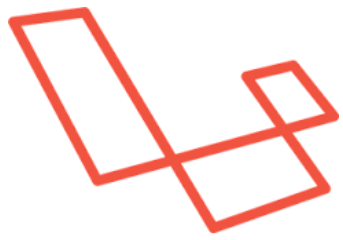
Laravel: Control de usuarios

Generar control de usuarios automatizado

- El siguiente comando genera las rutas y vistas necesarias para el control de usuarios

```
php artisan make:auth
```

- Las funcionalidades automáticas que genera para el control de usuarios son las siguientes:
 - Login
 - Registro
 - Recuperar contraseña



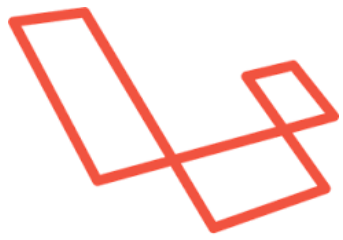
laravel

Laravel: Control de usuarios

Configuración inicial

- La configuración del sistema de autenticación está en `config/auth.php`
- La migración de la tabla usuarios está en `database/migrations/<timestamp>_create_users_table.php`
la implementación del método `up()`:

```
Schema::create('users', function (Blueprint $table) {  
    $table->increments('id');  
    $table->string('name');  
    $table->string('email')->unique();  
    $table->string('password');  
    $table->rememberToken();  
    $table->timestamps();  
});
```

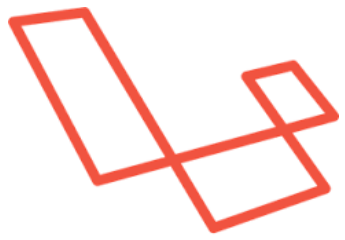


laravel

Laravel: Control de usuarios

Vistas

- Las vistas que genera el sistema de autenticación están en `resources/views/auth` con los siguientes nombres:
 - `login.blade.php`
 - `register.blade.php`
 - `passwords/email.blade.php`
 - `passwords/reset.blade.php`
- Todas las vistas usan plantillas blade que heredan del `layout layouts/app.blade.php`
- El sistema permite modificar el diseño y cualquier elemento de la vista siempre que se respeten las url y parámetros



laravel

Laravel: Control de usuarios

Manejo de usuarios

- Acceder a los datos de un usuario autenticado:

```
$user = Auth::user();  
echo $user->email;
```

- Cerrar sesión

```
Auth::logout();
```

- Comprobar si un usuario está autenticado

```
if(Auth::check()){  
    // ..  
}
```



Laravel: Control de usuarios

Proteger las rutas para usuarios no autenticados

- El sistema de autenticación provee un Middleware para proteger las rutas, el alias del Middleware es `auth`

```
//Para proteger una función:  
Route::get('admin/catalog', function(){  
    //Solo se permite el acceso a usuarios autenticados  
})->middleware('auth');
```

```
//Para proteger una acción de un controlador  
Route::get('profile', 'ProfileController@show')->middleware('auth');
```