

```
1 import { check, validationResult } from 'express-validator'
2 import bcrypt from 'bcrypt';
3 import Usuario from '../models/Usuario.js';
4 import { generarId, generarJWT } from '../helpers/tokens.js'
5 import { emailRegistro, emailOlvidePassword } from '../helpers/emails.js'
6
7 const formularioRegistro = (req, res) => {
8   res.render('auth/registrar', {
9     pagina : 'Crear Cuenta',
10     csrfToken: req.csrfToken()
11   });
12 }
```

MVC

Model View Controller

```
25   csrfToken: req.csrfToken(),
26   errores : resultado.array(),
27   pagina : 'Crear Cuenta',
28   usuario : {
```

Juan Pablo De la torre Valdez

www.codigoconjuan.com



¿Qué es MVC?



MVC son las iniciales de Model View Controller

Es un patrón de arquitectura de software que permite la separación de obligaciones de cada pieza de tu código.

Enfatiza la separación de la lógica de programación con la presentación



Ventajas de MVC



MVC no mejora el performance del código, tampoco da seguridad; pero tu código tendrá un mejor orden y será fácil de mantener.

En un grupo de trabajo, el tener el código ordenado permite que más de una persona pueda entender que es lo que hace cada parte de él.

Aprender MVC, hará que otras tecnologías como Laravel, Nest, Rails, Django, Net Core, Spring Boot te sean más sencillas de aprender.



Partes de

MVC



M = Model o Modelo

V = View o Vista

C = Controller o Controlador



Modelo en

MVC



Encargado de todas las interacciones en la base de datos, obtener datos, actualizarlos y eliminar.

El Modelo se encarga de consultar una base de datos, obtiene la información pero no la muestra, eso es trabajo de la vista.

El Modelo tampoco se encarga de actualizar la información directamente; es el Controlador quien decide cuándo llamarlo.



Vista en MVC



Se encarga de todo lo que se ve en pantalla (HTML)

Node soporta múltiples Template Engine como son EJS, Pug, Handlebars

Si utilizas React, Vue, Angular, Svelte, estos serían tu vista.

El Modelo consulta la base de datos, pero es por medio del Controlador que se decide que Vista hay que llamar y que datos presentar.



Controlador en



MVC

Es el que comunica modelo y vista; antes de que el Modelo consulte la base de datos el Controlador es el encargado de llamar un Modelo en específico.

Una vez consultado el Modelo, el controlador recibe esa información, manda llamar a la vista y le pasa la información.

El controlador es el que mandar llamar la vista y modelos, que se requieren en cada parte de tu aplicación.



Router en MVC



Es el encargado de registrar todas las URL's o Endpoints que va a soportar nuestra aplicación.

Ejemplo: Si el Usuario accede a /clientes el router ya tiene registrada esa ruta y un controlador con una función que sabe que Modelo debe llamar y que vista mostrar cuando el usuario visita esa URL.



/clientes

CÓDIGO
CON
JUAN

controlador

modelo

vista

