

```
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/registrarse', {
9     pagina : 'Crear Cuenta',
10     csrfToken: req.csrfToken()
11   });
12 }
13
14 export default formularioRegistro;
```

# Templates Engines

## ¿Qué son y ventajas?

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

Juan Pablo De la torre Valdez

[www.codigoconjuan.com](http://www.codigoconjuan.com)



# ¿Qué es un Template Engine?



Template Engine o Motores de Plantilla son tecnologías que nos permiten crear el código HTML y mostrar información contenida en variables de una forma más compacta y clara

Pug, Handlebars, EJS son las opciones más populares para Node.js.

También es posible utilizar React, Angular, Svelte o Vue como tu Template Engine; pero necesitarás crear una API con respuestas JSON.



# Ventajas de Un Template Engine



La forma en que se crea el Código HTML tiende a ser más simple.

Puedes utilizar e imprimir información del servidor o base de Datos de una forma más sencilla.

La seguridad es más sencilla de implementar ya que la información es renderizada por el mismo servidor.



# Desventajas de

## Un Template Engine



Cuando mostramos información con un template Engine, esta información y sus interacciones no son muy dinámicas a comparación de React o Vue.

Consumen más recursos ya que el código HTML es creado por el servidor a diferencia de Vue o React donde es creado por el cliente (navegador)



# Vamos a instalar e implementar un Template Engine

