*INYECCIÓN POR DEPENDENCIA EN LARAVEL.*

*¿Qué es la Inyección de Dependencias?*

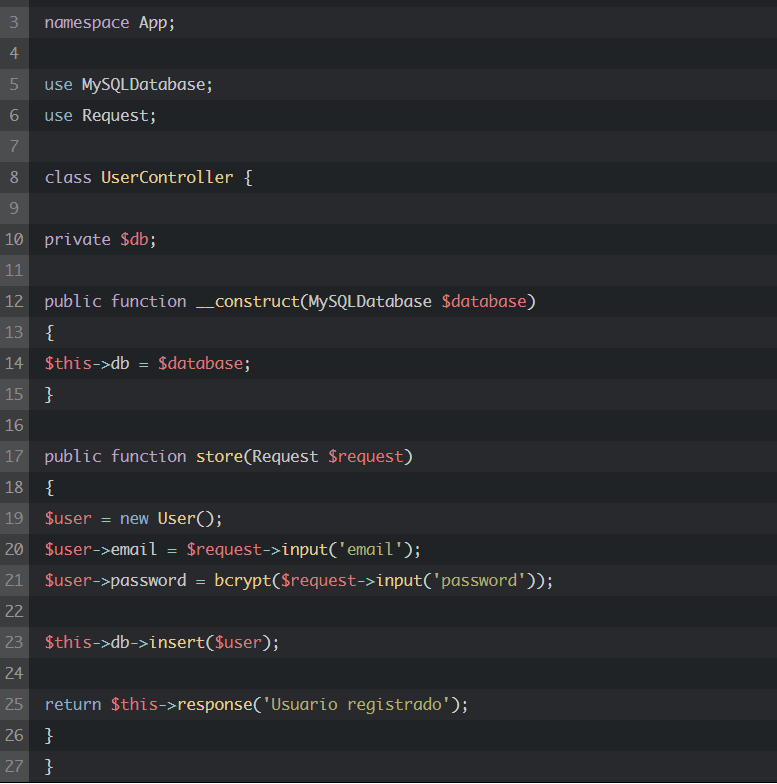
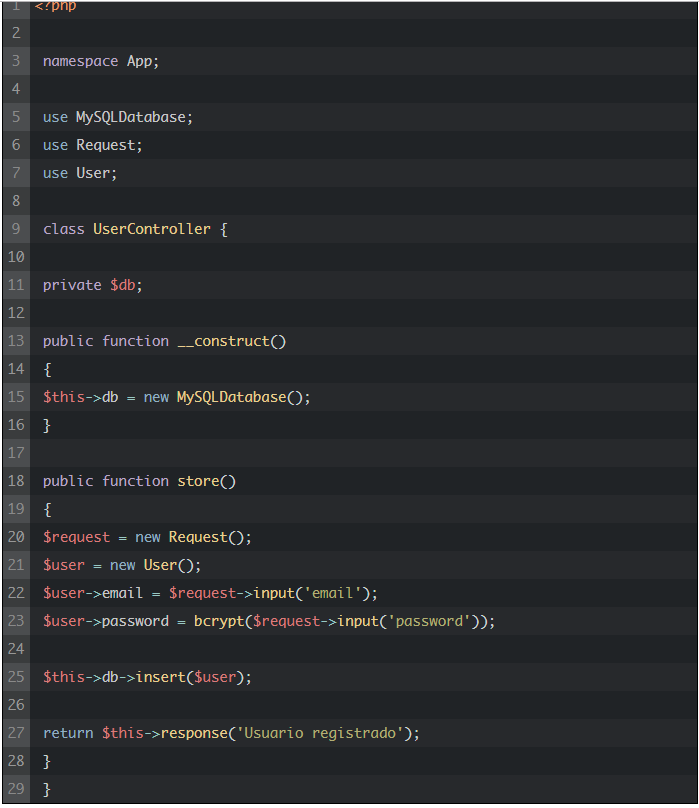
*La Inyección de Dependencias es un patrón de diseño orientado a objetos en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. El término fue acuñado por primera vez por Martin Fowler. En la práctica, consiste en pasar a los componentes de software sus dependencias mediante sus constructores, sus propiedades o sus métodos setter. Así de sencillo.*

*Ventajas de usar la Inyección de Dependencias*

*Como ventajas podemos nombrar:*

* *Nuestro código se vuelve más modular y menos acoplado, ya que los submódulos no necesitan saber detalles de la implementación de otros en tu código.*
* *Gracias a esta modularidad es que podemos hacer tests unitarios más fácilmente, cada parte del programa se puede aislar porque para realizar pruebas no tenemos que proveer a nuestros programas de las referencias reales, sino que podemos fingirlas mediante mocks.*
* *La inyección de dependencias puede hacer que escribas código más rápido y te quita preocupaciones de tener que instanciar objetos tu mismo.*

*Usando Inyección de Dependencia Sin usar Inyección de Dependencia*

*¿Cuál sería el proceso de crear un Request de nombre: “RecuperarContrasenaRequest”?*

*Ejecutar nuevamente el comando php artisan make:request pero con el nombre RecuperarContrasenaRequest.*

*Probando el código:*

1. *Intente recuperar la contraseña con un correo válido, dos contraseñas correctas (con más de 5 caracteres), pero con pregunta y/o respuesta equivocada. Anote los resultados*

*En el caso que ponga los datos diferentes*

*El programa muestra el mensaje de error y que los datos no coinciden*

*Si la contraseña es pequeña muestra el mensaje que exige que la contraseña sea mayor que 6 dígitos.*

*En el caso que brinde las credenciales correctas*

*El programa me manda un mensaje verde que el cambio se realizó correctamente.*

1. *Intente recuperar la contraseña con todos los datos correctos. ¿Qué sucede?*

*Muestra un mensaje de éxito*

*Conclusiones*

* *Se llegó a la conclusión que las inyecciones de Dependencias permiten imlementar menos código porque los submódulos no necesitan saber detalles de la implementación de otros en tu código.*
* *El control de excepciones en Laravel nos permite tener al usuario más informado en que parte se está equivocando.*
* *En todo momento utilizamos mensajes para interactuar con el usuario si su requerimiento fue exitoso o no para ello utilizamos tipos de clases alert.*
* *Laravel maneja el uso de contraseñas mediante encriptación de estas para que en el camino de este hacia la base de datos sea segura.*
* *Creamos 2 Request para que nos facilitara todos los datos de la solicitud actual a través HTTP Request, un objeto sobre el que podremos consultar información sobre el cliente que realiza la solicitud y datos que pueda estar enviando. Toda aplicación web recibe solicitudes para completar todo tipo de acciones.*

*Observaciones*

* *Utilizamos las clases seeder para agregar datos en cantidad a nuestras tablas.*
* *Ejecutamos nuestros modelos para que nuestras tablas fueran creadas.*
* *Realizamos mantenimiento a la base de datos.*
* *Laravel ya por defecto administra toda la parte de la validación e inicio de sesión.*
* *Cuando queremos tener acceso a un dato de la base de datos se nombra {{Auth::\_\_\_\_\_\_}}*
* *Se realizó uso de === para comprobar las estructura de dos variables en la validación.*