

Ejercicios puntuales

1. Crear un archivo **usuario.php** en donde definiremos la clase "Usuario"
2. Crear un archivo **prueba.php** que incluya el archivo anterior e instancie un usuario. Imprimir en pantalla (a través de la función **var_dump**) dicha variable.
3. Instanciar un segundo usuario e imprimirlo.
4. Agregar al usuario 3 atributos públicos:
 - a. Nombre
 - b. Fecha de Nacimiento
 - c. Mail
5. Imprimir nuevamente ambas instancias en pantalla
6. Crear un constructor para la clase que reciba los 3 parámetros (nombre, fecha de nacimiento y mail) y los asigne a sus atributos correspondientes.
7. Volver a crear ambas instancias en **prueba.php** pero utilizando los nuevos constructores.
8. Imprimir el mail de ambos usuarios en pantalla, modificarlos y luego volverlos a imprimir.
9. Redefinir los atributos como privados. ¿Qué sucede ahora cuando queremos acceder o modificar los atributos?
10. Crear un método llamado "*saludar*" que imprima "Hola NOMBRE" en donde el nombre debe ser el nombre del usuario.
11. Probar llamar al método *saludar*.
12. Agregar los siguientes métodos (que permitan consultar y modificar los atributos):
 - a. `getNombre`
 - b. `getMail`
 - c. `getEdad`
 - d. `setNombre`
 - e. `setMail`
 - f. `setEdad`
13. Agregar una validación en `setEdad` para que el valor sea un número.
14. Agregar una validación en `setMail` para que el valor sea un mail.
15. Modificar el constructor para que utilice los métodos `setMail` y `setEdad`, utilizando así las validaciones ya armadas.
16. Le agregaremos al usuario el atributo privado **password** el cual también debe poder definirse en el constructor.
17. Agregaremos el método privado *encriptarPassword* que dado un string devuelve su versión **hasheada**.
18. Haremos los métodos `getPassword` y `setPassword` (este último debe utilizar *encriptarPassword*)
19. Probar llamar a estos últimos 3 métodos creados desde alguna de las instancias creadas.

Ejercicios complementarios

1. Partiendo de los documentos subidos de la clase de **repaso** (o un código propio que también haga registración y login) modificar el código para que exista una Clase **usuario** que modele al usuario. Esta Clase debe ser utilizada a lo largo del código tanto al recuperar un usuario, guardarlo, validarlo y en cada una de sus referencias.

