

Ejercicios

Aclaración: Queda en responsabilidad del alumno modificar la ejecución del código para probar correctamente los métodos generados.

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. A la Clase usuario pongamosle 3 atributos (nombre, mail y contraseña)
4. Instanciar un segundo usuario e imprimirlo.
5. Crear un constructor para la clase que reciba los 3 parámetros y los asigne a sus atributos correspondientes.
6. Volver a crear ambas instancias en **prueba.php** pero utilizando los nuevos constructores.
7. Imprimir el mail de ambos usuarios en pantalla, modificarlos y luego volverlos a imprimir.
8. Redefinir los atributos como privados. ¿Qué sucede ahora cuando queremos acceder o modificar los atributos?
9. Crear un método llamado "*saludar*" que imprima "Hola NOMBRE" en donde el nombre debe ser el nombre del usuario.
10. Probar llamar al método *saludar*.
11. Agregar los siguientes métodos (que permitan consultar y modificar los atributos):
 - a. `getNombre`
 - b. `getMail`
 - c. `getPass`
 - d. `setNombre`
 - e. `setMail`
 - f. `setPass`
12. Agregaremos el método privado *encriptarPass* que dado un string devuelve su versión **hasheada**.
13. Haremos que el método `setPass` debe utilizar *encriptarPass*.
14. Modificar el constructor para que llame a `setPass` y así asegurar que nuestra contraseña quede encriptada.
15. Crear en un archivo nuevo una clase llamada Celular. La misma debe tener cuatro atributos: marca, modelo, proveedor de línea y número telefónico. Definir los atributos como privados. Definir los getters, setters y constructor de la clase.
16. Modificar la clase Usuario para que tenga un Celular.
17. Modificar el archivo **prueba.php** para que existan dos instancias de la clase usuario, ambas asociadas a 2 Celulares diferentes.
18. Agregar en **usuario.php** el método **mostrarTelefono**. El método debe retornar un string con una descripción del teléfono incluyendo marca y modelo. Si la marca del teléfono es Apple debe agregarse al final del string el texto "*y soy fan de los iphones*".

19. Agregar en **usuario.php** el método **llamar**. El mismo debe recibir un Usuario y el tiempo de duración de la llamada. La función debe retornar el costo de la llamada. Si ambos usuarios cuentan con el mismo proveedor de línea, la llamada es gratis. Si esto no es así la llamada cuesta \$10 por minuto.
20. Crear una clase llamada Habilidad. La misma debe tener dos atributos: nombre y expertise (expertise es un número entre 0 y 5). Definir los atributos como privados. Definir los getters, setters y constructor de la clase.
21. Agregar en la clase Usuario la posibilidad de que un Usuario pueda tener muchas Habilidades
22. Modificar las instancias en **prueba.php** para que cada usuario tenga 3 habilidades.
23. Agregar en la clase Usuario el método **sabeHacer** la misma recibe un string y un puntaje. La función devuelve verdadero únicamente si el string recibido es el nombre de una de las habilidades del usuario y si el puntaje es menor al expertise del usuario en esa habilidad. En caso contrario, retorna falso.
24. Agregarle al usuario el atributo **id** junto con su getter y setter.
25. Agregarle al usuario el método **guardar**. Para esto:
 - a. Queda a criterio del alumno en qué soporte almacenar la información del Usuario
 - b. Si el usuario no tiene id, esto significa que el usuario es nuevo y debería hacerse una inserción
 - c. Si el usuario ya tiene id asignado debe realizarse un update
26. Generar una función que retorne todos los usuarios almacenados. Esta función no debe retornar arrays sino objetos Usuario. Tener cuidado con que estos objetos usuarios tengan su id correspondiente y no volver a encriptar la contraseña
27. Agregarle al usuario el método **eliminar** que lo elimine de la base de datos en caso de ya tener id asignado.