

ÍNDICE

ACTIVIDAD 1	2
SOLUCIÓN 1.1 claseUsuario.php	3
SOLUCIÓN 1.2 actividad1.php	3
ACTIVIDAD 2	4
SOLUCIÓN 2.1 claseProducto.php	5
ACTIVIDAD 3	6
SOLUCIÓN 3.1 claseCoche.php	7
SOLUCIÓN 3.2	7

ACTIVIDAD 1

1. Crea una clase **Usuario** en PHP que represente a un usuario de un sistema de registro.

Requisitos:

- **Atributos:**
 - **nombre:** Cadena de texto que almacena el nombre del usuario.
 - **email:** Cadena de texto que almacena la dirección de correo electrónico del usuario.
 - **password:** Cadena de texto que almacena la contraseña del usuario encriptada.
 - **Constructor:**
 - Recibe como parámetros el nombre, el email y la contraseña del usuario.
 - Asigna los valores a los atributos correspondientes.
 - Encripta la contraseña utilizando la función `password_hash()` con el algoritmo `PASSWORD_DEFAULT`.
 - **Método `getNombre()`:**
 - Devuelve el nombre del usuario.
2. Crea un programa en PHP que utilice la clase **Usuario** para registrar a un nuevo usuario y mostrar su nombre. Además, implementa un método para cambiar la contraseña del usuario.

SOLUCIÓN 1.1 `claseUsuario.php`

```
class Usuario {
    private $nombre;
    private $email;
    private $password;

    public function __construct($nombre, $email, $password) {
        $this->nombre = $nombre;

        $this->email = $email;
        $this->password
        = password_hash($password, PASSWORD_DEFAULT); // Encriptamos
la contraseña
    }

    public function getNombre() {
        return $this->nombre;
    }

    // Otros métodos: cambiar contraseña, iniciar sesión, etc.
}
```

SOLUCIÓN 1.2 `actividad1.php`

```
<?php
require_once ('claseUsuario.php')
// Define la clase Usuario (se asume que ya está definida como en el
ejemplo anterior)

// Crear un nuevo usuario
$usuario1 = new Usuario("Juan Pérez", "juan@ejemplo.com",
"micontraseñasecreta");

// Obtener el nombre del usuario e imprimirlo
echo "Nombre del usuario: " . $usuario1->getNombre() . "\n";

?>
```

ACTIVIDAD 2

Crea una clase **Producto** en PHP que represente un producto en una tienda en línea.

Requisitos:

- **Atributos:**
 - **id**: Un identificador único para el producto (entero).
 - **nombre**: El nombre del producto (cadena).
 - **precio**: El precio del producto (flotante).
 - **stock**: La cantidad de producto disponible en el inventario (entero).
- **Constructor:**
 - Recibe como parámetros el **id**, **nombre**, **precio** y opcionalmente el **stock** (con valor por defecto 0).
 - Asigna los valores a los atributos correspondientes.
- **Método disminuirStock:**
 - Recibe como parámetro la cantidad de producto a restar del stock.
 - Si hay suficiente stock, resta la cantidad y devuelve **true**.
 - Si no hay suficiente stock, devuelve **false**.

Una vez creada la clase, deberás poder ejecutar el siguiente código y obtener el resultado esperado:

```
require_once ('claseProducto.php')
// Crear un nuevo producto
$producto1 = new Producto(1, "Camiseta", 19.99, 10);

// Intentar vender 5 unidades
if ($producto1->disminuirStock(5)) {
    echo "Venta realizada con éxito.";
} else {
    echo "No hay suficiente stock.";
}
```

SOLUCIÓN 2.1 claseProducto.php

```
class Producto {
    private $id;
    private $nombre;
    private $precio;
    private $stock;

    public function __construct($id, $nombre, $precio, $stock = 0) {
        $this->id = $id;
        $this->nombre = $nombre;
        $this->precio = $precio;
        $this->stock = $stock;
    }

    public function disminuirStock($cantidad) {
        if ($this->stock >= $cantidad) {
            $this->stock -= $cantidad;
            return true;
        } else {
            return false;
        }
    }
}
```

ACTIVIDAD 3

3.1 Crea una clase **Coche** en PHP para simular un vehículo.

Requisitos:

- **Atributos:**
 - **marca:** La marca del coche (cadena).
 - **modelo:** El modelo del coche (cadena).
 - **velocidad:** La velocidad actual del coche (número).
- **Constructor:**
 - Recibe como parámetros la **marca** y el **modelo**.
 - Inicializa los atributos correspondientes y establece la velocidad inicial a 0.
- **Métodos:**
 - **acelerar(\$cantidad):** Incrementa la velocidad del coche en la cantidad especificada.
 - **frenar(\$cantidad):** Disminuye la velocidad del coche en la cantidad especificada. Si la velocidad se vuelve negativa, se establece a 0.

3.2 Crea un programa en PHP que simule un coche.

Requisitos:

- **Utilizar la clase **Coche** proporcionada:** Esta clase ya define los atributos (marca, modelo, velocidad) y los métodos (constructor, acelerar, frenar).
- **Crear un objeto **Coche**:** Instancia un nuevo objeto de la clase **Coche** con una marca y modelo de tu elección.
- **Simular un viaje:** Utiliza los métodos **acelerar** y **frenar** para modificar la velocidad del coche.
- **Mostrar la velocidad final:** Imprime en pantalla la velocidad final del coche después de las aceleraciones y frenadas.

SOLUCIÓN 3.1 claseCoche.php

```
class Coche {
    private $marca;
    private $modelo;
    private $velocidad;

    public function __construct($marca, $modelo) {
        $this->marca = $marca;
        $this->modelo = $modelo;
        $this->velocidad = 0;
    }

    public function acelerar($cantidad) {
        $this->velocidad += $cantidad;
    }

    public function frenar($cantidad) {
        $this->velocidad -= $cantidad;
        if ($this->velocidad < 0) {
            $this->velocidad = 0;
        }
    }
}
```

SOLUCIÓN 3.2

```
require_once ('claseCoche.php')
// Crear un nuevo coche
$miCoche = new Coche("Toyota", "Corolla");

// Acelerar el coche
$miCoche->acelerar(50);

// Frenar el coche
$miCoche->frenar(20);

// Mostrar la velocidad actual
echo "La velocidad actual es: " . $miCoche->velocidad . " km/h";
```