

EJERCICIOS DE PROGRAMACIÓN ORIENTADA A OBJETOS  
(POO)

Aprendiz:  
Sara Elizabeth Pulido Beltrán

Ficha:  
2828866

Tecnólogo:  
Análisis y Desarrollo de Software  
(ADSO)

Complejo de Paloquemao  
Centro de Diseño y Metrología

## Ejercicio Métodos y Clases

```
cuadernos.php > Cuadernos > imprimir()
1  <?php
   6 references | 0 implementations
2  class Cuadernos {
3      // Definición de Atributos
4      private $marca, $port_color, $tamaño, $tipo_hoja, $cantidad_hojas, $gram_hojas;
5
6      // Constructor para inicializar los atributos
   6 references | 0 overrides
7      public function __construct($marca, $port_color, $tamaño, $tipo_hoja, $cantidad_hojas, $gram_hojas) {
8          $this->marca = $marca;
9          $this->port_color = $port_color; // Aquí corregimos el nombre
10         $this->tamaño = $tamaño;
11         $this->tipo_hoja = $tipo_hoja;
12         $this->cantidad_hojas = $cantidad_hojas;
13         $this->gram_hojas = $gram_hojas; // Aquí corregimos el nombre
14     }
15
16
17     // Método para imprimir
   3 references | 0 overrides
18     public function imprimir(): void {
19         echo "Marca: " . $this->marca . "<br>";
20         echo "Color de Portada: " . $this->port_color . "<br>"; // Ahora sí mostrará el color
21         echo "Tamaño: " . $this->tamaño . "<br>";
22         echo "Tipo de Hoja: " . $this->tipo_hoja . "<br>";
23         echo "Cantidad de Hojas: " . $this->cantidad_hojas . "<br>";
24         echo "Gramaje de Hojas: " . $this->gram_hojas . "<br><br>"; // Ahora sí mostrará el gramaje
25     }
26 }
27
28 // Crear objetos
29 $cuaderno1 = new Cuadernos(marca: "Norma", port_color: "Azul", tamaño: "A4", tipo_hoja: "Rayadas", canti_100, "56 g/m²");
30 $cuaderno2 = new Cuadernos(marca: "Scribe", port_color: "Rojo", tamaño: "Carta", tipo_hoja: "Cuadrículadas", canti_80, "60 g/m²");
31 $cuaderno3 = new Cuadernos(marca: "Fabriano", port_color: "Negro", tamaño: "A3", tipo_hoja: "Opalina", canti_50, "120 g/m²");
32 //mostrar objetos
33 echo $cuaderno1->imprimir();
34 echo $cuaderno2->imprimir();
35 echo $cuaderno3->imprimir();
36 ?>
```

← → ↻ ⓘ localhost/php%20poo/cuadernos.php

Marca: Norma  
Color de Portada: Azul  
Tamaño: A4  
Tipo de Hoja: Rayadas  
Cantidad de Hojas: 100  
Gramaje de Hojas: 56 g/m²

Marca: Scribe  
Color de Portada: Rojo  
Tamaño: Carta  
Tipo de Hoja: Cuadrículadas  
Cantidad de Hojas: 80  
Gramaje de Hojas: 60 g/m²

Marca: Fabriano  
Color de Portada: Negro  
Tamaño: A3  
Tipo de Hoja: Opalina  
Cantidad de Hojas: 50  
Gramaje de Hojas: 120 g/m²

## Ejercicio Creación de Tabla

```
tabla.php > Tabla > inicioTabla()
1  <html>
2  <head>
3  |   <title>Tabla de Libros</title>
4  </head>
5  <body>
6  <?php
7  //clase de tabla c
  1 reference | 0 implementations
8  class Tabla {
  2 references
9  |   private $mat = array();
  2 references
10 |   private $cantFilas;
  2 references
11 |   private $cantColumnas;
12 //creacion funciones con constructor filas y columnas
  1 reference | 0 overrides
13 |   public function __construct($filas, $columnas) {
14 |       $this->cantFilas = $filas;
15 |       $this->cantColumnas = $columnas;
16 |   }
17
  9 references | 0 overrides
18 |   public function cargar($fila, $columna, $valor): void {
19 |       $this->mat[$fila][$columna] = $valor;
20 |   }
21
  1 reference | 0 overrides
22 |   public function inicioTabla(): void {
23 |       echo '<table border="1" cellspacing="0" cellpadding="5">';
24 |   }
25
  1 reference | 0 overrides
26 |   public function inicioFila(): void {
27 |       echo '<tr>';
28 |   }
29
  1 reference | 0 overrides
30 |   public function mostrar($fila, $columna): void {
31 |       echo '<td>'. $this->mat[$fila][$columna]. '</td>';
32 |   }
```

```

33 |
34 | 1 reference | 0 overrides
35 | public function finFila(): void {
36 |     echo '</tr>';
37 | }
38 |
39 | 1 reference | 0 overrides
40 | public function finTabla(): void {
41 |     echo '</table>';
42 | }
43 |
44 | 1 reference | 0 overrides
45 | public function graficar(): void {
46 |     $this->inicioTabla();
47 |
48 |     // Encabezados de la tabla
49 |     echo '<tr><th>Título</th><th>Autor</th><th>Año</th></tr>';
50 |
51 |     for ($f = 1; $f <= $this->cantFilas; $f++) {
52 |         $this->inicioFila();
53 |         for ($c = 1; $c <= $this->cantColumnas; $c++) {
54 |             $this->mostrar(fila: $f, columna: $c);
55 |         }
56 |         $this->finFila();
57 |     }
58 |     $this->finTabla();
59 | }

```

```

59 | // Crear tabla con 3 filas y 3 columnas (Título, Autor, Año)
60 | $tablaLibros = new Tabla(filas: 3, columnas: 3);
61 | $tablaLibros->cargar(fila: 1, columna: 1, valor: "1984");
62 | $tablaLibros->cargar(fila: 1, columna: 2, valor: "George Orwell");
63 | $tablaLibros->cargar(fila: 1, columna: 3, valor: "1949");
64 |
65 | $tablaLibros->cargar(fila: 2, columna: 1, valor: "Cien años de soledad");
66 | $tablaLibros->cargar(fila: 2, columna: 2, valor: "Gabriel García Márquez");
67 | $tablaLibros->cargar(fila: 2, columna: 3, valor: "1967");
68 |
69 | $tablaLibros->cargar(fila: 3, columna: 1, valor: "El Gran Gatsby");
70 | $tablaLibros->cargar(fila: 3, columna: 2, valor: "F. Scott Fitzgerald");
71 | $tablaLibros->cargar(fila: 3, columna: 3, valor: "1925");
72 |
73 | // Mostrar la tabla
74 | $tablaLibros->graficar();
75 | ?>
76 | </body>
77 | </html>

```

← → ↻ ⓘ localhost/php%20poo/tabla.php

Título	Autor	Año
1984	George Orwell	1949
Cien años de soledad	Gabriel García Márquez	1967
El Gran Gatsby	F. Scott Fitzgerald	1925

## Ejercicio Condicional

```
condicional.php > Empleado > pagaImpuestos()
1  <html>
2  <head>
3  <title>Pruebas</title>
4  </head>
5  <body>
6  <?php
7  //Clase
   4 references | 0 implementations
8  class Empleado {
   2 references
9  |   private $nombre;
   4 references
10 |   private $sueldo;
11 |
   4 references | 0 overrides
12 |   public function inicializar($nom, $sue): void {
13 |       $this->nombre = $nom;
14 |       $this->sueldo = $sue;
15 |   }
16 | //Funciones constructor
   4 references | 0 overrides
17 |   public function pagaImpuestos(): void {
18 |       echo $this->nombre . " - Sueldo: $" . $this->sueldo . " - ";
19 | //condicional
20 |       if ($this->sueldo > 3000) {
21 |           echo "Debe pagar impuestos. Categoría: ALTO";
22 |       } elseif ($this->sueldo <= 1500) {
23 |           echo "No paga impuestos. Categoría: BAJO";
24 |       } else {
25 |           echo "No paga impuestos. Categoría: MEDIO";
26 |       }
27 |
28 |       echo "<br>";
29 |   }
30 }

31 //Objetos
32 $emplead01 = new Empleado();
33 $emplead01->inicializar(nom: 'Luis', sue: 1200);
34 $emplead01->pagaImpuestos();
35
36 $emplead02 = new Empleado();
37 $emplead02->inicializar(nom: 'Carla', sue: 2500);
38 $emplead02->pagaImpuestos();
39
40 $emplead03 = new Empleado();
41 $emplead03->inicializar(nom: 'Pedro', sue: 4300);
42 $emplead03->pagaImpuestos();
43
44 $emplead04 = new Empleado();
45 $emplead04->inicializar(nom: 'Carmen', sue: 5300);
46 $emplead04->pagaImpuestos();
47 ?>
48 </body>
49 </html>
50
```

← → ↻ ⓘ localhost/php%20poo/condicional.php

Luis - Sueldo: \$1200 - No paga impuestos. Categoría: BAJO  
Carla - Sueldo: \$2500 - No paga impuestos. Categoría: MEDIO  
Pedro - Sueldo: \$4300 - Debe pagar impuestos. Categoría: ALTO  
Carmen - Sueldo: \$5300 - Debe pagar impuestos. Categoría: ALTO

## Ejercicio Completo Tabla y Condicional con ejemplo antiguo

```
cuadernostabla.php > Cuadernos > tabla()
1  <html>
2  <head>
3      <title>Ejemplo Cuaderno</title>
4      <style>
5          table {
6              width: 60%;
7              border-collapse: collapse;
8              margin: 20px 0;
9              font-size: 18px;
10             text-align: center;
11         }
12         th, td {
13             padding: 12px;
14             border: 3px solid black;
15         }
16         th {
17             background-color: #000080;
18             color: white;
19         }
20         .premium {
21             color: #4b0082;
22             font-weight: bold;
23         }
24     </style>
25 </head>
26 <body>
27 <?php

28 class Cuadernos {
29     // Definición de Atributos
30     private $marca, $port_color, $tamaño, $tipo_hoja, $cantidad_hojas, $gram_hojas, $precio;
31
32     // Constructor para inicializar los atributos
33     public function __construct($marca, $port_color, $tamaño, $tipo_hoja, $cantidad_hojas, $gram_hojas, $precio) {
34         $this->marca = $marca;
35         $this->port_color = $port_color; // Aquí corregimos el nombre
36         $this->tamaño = $tamaño;
37         $this->tipo_hoja = $tipo_hoja;
38         $this->cantidad_hojas = $cantidad_hojas;
39         $this->gram_hojas = $gram_hojas; // Aquí corregimos el nombre
40         $this->precio = floatval(value: str_replace(search: '$', replace: '', subject: $precio)); //convertimos en numero
41     }
42
43     //Metodo de Iva
44     public function CalculoImpuesto(): float {
45         return round(num: $this->precio * 1.19, precision: 2);
46     }
47     public function Premium(): bool {
48         return $this->precio > "4000";
49     }
}
```

```

50 public function tabla(): void {
51     echo "<tr>";
52     echo "<td>" . $this->marca . "</td>";
53     echo "<td>" . $this->port_color . "</td>";
54     echo "<td>" . $this->tamaño . "</td>";
55     echo "<td>" . $this->tipo_hoja . "</td>";
56     echo "<td>" . $this->cantidad_hojas . "</td>";
57     echo "<td>" . $this->gram_hojas . "</td>";
58     echo "<td>$" . number_format(num: $this->precio, decimals: 2) . "</td>";
59     echo "<td>$" . number_format(num: $this->CalculoImpuesto(), decimals: 2) . "</td>";
60
61     if ($this->Premium()) {
62         echo "<td class='premium'>Producto Premium </td>";
63     } else {
64         echo "<td>Producto Normal</td>";
65     }
66
67     echo "</tr>";
68 }
69
70
71 // Crear objetos
72 $cuadernos=[
73 $cuaderno1 = new Cuadernos(marca: "Norma", port_color: "Azul", tamaño: "A4", tipo_hoja: "Rayadas", canti_100, "56 g/m²", "$2500"),
74 $cuaderno2 = new Cuadernos(marca: "Scribe", port_color: "Rojo", tamaño: "Carta", tipo_hoja: "Cuadriculadas", canti_80, "60 g/m²", "$3000"),
75 $cuaderno3 = new Cuadernos(marca: "Fabriano", port_color: "Negro", tamaño: "A3", tipo_hoja: "Opalina", canti_50, "120 g/m²", "$5000"),
76 ];

```

```

77 // Imprimir información de cada cuaderno
78 echo "<table>";
79 echo "<tr>
80     <th>Marca</th>
81     <th>Color de Portada</th>
82     <th>Tamaño</th>
83     <th>Tipo de Hoja</th>
84     <th>Cantidad de Hojas</th>
85     <th>Gramaje de Hojas</th>
86     <th>Precio Base</th>
87     <th>Precio con IVA</th>
88     <th>Clasificación</th>
89 </tr>";
90
91 foreach ($cuadernos as $cuaderno) {
92     $cuaderno->tabla();
93 }
94
95 echo "</table>";
96 ?>
97 </body>
98 </html>

```

localhost/php%20poo/cuadernostabla.php

Marca	Color de Portada	Tamaño	Tipo de Hoja	Cantidad de Hojas	Gramaje de Hojas	Precio Base	Precio con IVA	Clasificación
Norma	Azul	A4	Rayadas	100	56 g/m²	\$2,500.00	\$2,975.00	Producto Normal
Scribe	Rojo	Carta	Cuadriculadas	80	60 g/m²	\$3,000.00	\$3,570.00	Producto Normal
Fabriano	Negro	A3	Opalina	50	120 g/m²	\$5,000.00	\$5,950.00	Producto Premium

## Ejercicio Herencia y Encapsulamiento

```
AprendicesEjemplo.php > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Lista de Aprendices</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             margin: 20px;
11         }
12         h2 {
13             color: #333;
14         }
15         ol {
16             background: white;
17             padding: 5px;
18             border-radius: 8px;
19         }
20         li {
21             padding: 5px 0;
22             font-size: 18px;
23         }
24     </style>
25 </head>
26 <body>
27
28     <h2>Lista de Aprendices</h2>
29
30
31     <?php
32     require_once('Aprendices.php');
```



```

31 <?php
32 require_once('Aprendices.php');
33
34 1 reference | 0 implementations
35 class Curso
36 {
37     2 references
38     private $aprendices = [];
39
40     1 reference | 0 overrides
41     public function agregarAprendiz(Aprendices $aprendiz): void
42     {
43         $this->aprendices[] = $aprendiz;
44     }
45
46     1 reference | 0 overrides
47     public function obtenerNombresAprendices(): array
48     {
49         return array_map(callback: fn($aprendiz): mixed => $aprendiz->getNombre(), array: $this->aprendices);
50     }
51 }
52
53 // Crear el curso y agregar aprendices
54 $objCurso = new Curso();
55 $listaAprendices = ["Juan", "María", "Carlos", "Ana", "Luis"];
56
57 foreach ($listaAprendices as $nombre) {
58     $objCurso->agregarAprendiz(aprendiz: new Aprendices(nombre: $nombre));
59 }
60
61 // Mostrar los nombres de los aprendices en una lista ordenada
62 $nombresAprendices = $objCurso->obtenerNombresAprendices();
63

```

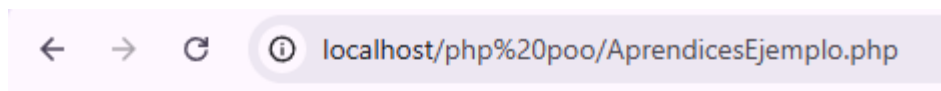
```

46     }
47 }
48
49 // Crear el curso y agregar aprendices
50 $objCurso = new Curso();
51 $listaAprendices = ["Juan", "María", "Carlos", "Ana", "Luis"];
52
53 foreach ($listaAprendices as $nombre) {
54     $objCurso->agregarAprendiz(aprendiz: new Aprendices(nombre: $nombre));
55 }
56
57 // Mostrar los nombres de los aprendices en una lista ordenada
58 $nombresAprendices = $objCurso->obtenerNombresAprendices();
59
60 echo "<ol>";
61 foreach ($nombresAprendices as $nombre) {
62     echo "<li>$nombre</li>";
63 }
64 echo "</ol>";
65 ?>
66
67 </body>
68 </html>
69

```

Código del incluye las clases en otra carpeta

```
Aprendices.php > ...
1  <?php
   2 references | 0 implementations
2  class Aprendices {
3      // Atributo privado
   2 references
4      private $nombre;
5
6      // Constructor
   1 reference | 0 overrides
7      public function __construct($nombre) {
8          $this->nombre = $nombre;
9      }
10
11     // Método para obtener el nombre
   0 references | 0 overrides
12     public function getNombre(): mixed {
13         return $this->nombre;
14     }
15 }
16 ?>
17
```



## Lista de Aprendices

1. Juan
2. María
3. Carlos
4. Ana
5. Luis

Ejercicio de Herencia

```

ejemplousuarioscrum.php > ...
1  <?php
2  require_once('Usuario.php'); // Archivo donde estarán definidas las clases base
3
4  // Clase Estudiante que hereda de Usuario
3 references | 0 implementations
5  class Estudiante extends Usuario {
6      2 references
7      private $curso;
8
9      3 references | 0 overrides | prototype
10     public function __construct($id, $nombre, $email, $curso) {
11         parent::__construct($id, nombre: $nombre, email: $email);
12         $this->curso = $curso;
13     }
14
15     0 references | 0 overrides
16     public function getCurso(): mixed {
17         return $this->curso;
18     }
19 }
20
21 // Crear instancias
22 $estudiante1 = new Estudiante(id: 1, nombre: "Juan Pérez", email: "juan@hotmail.com", curso: "ciencias");
23 $pago1 = new Pago(monto: 500, fecha: "2024-02-27", estudiante: $estudiante1);
24 $estudiante2 = new Estudiante(id: 1, nombre: "Lina calderon", email: "lina@gmail.com", curso: "Matemáticas");
25 $pago2 = new Pago(monto: 1000, fecha: "2024-06-26", estudiante: $estudiante2);
26
27 // Mostrar detalles del pago
28 echo $pago1->getDetallesPago(). "<br>";
29 echo $pago2->getDetallesPago();
30 ?>

```

Código que tiene las clases y esta en otea carpeta

```
Usuario.php > Usuario
1  <?php
2  // Archivo Usuarios.php
3
4  // Clase base Usuario
   2 references | 1 implementation
5  class Usuario {
   1 reference
6  |   protected $id;
   2 references
7  |   protected $nombre;
   2 references
8  |   protected $email;
9
10 |   1 reference | 1 override
   public function __construct($id, $nombre, $email) {
11 |       $this->id = $id;
12 |       $this->nombre = $nombre;
13 |       $this->email = $email;
14 |   }
15
16 |   1 reference | 0 overrides
   public function getNombre(): mixed {
17 |       return $this->nombre;
18 |   }
19
20 |   0 references | 0 overrides
   public function getEmail(): mixed {
21 |       return $this->email;
22 |   }
23 }
```

```
// Clase Pago
2 references | 0 implementations
class Pago {
   2 references
   private $monto;
   2 references
   private $fecha;
   2 references
   private $estudiante;

   2 references | 0 overrides
   public function __construct($monto, $fecha, Estudiante $estudiante) {
       $this->monto = $monto;
       $this->fecha = $fecha;
       $this->estudiante = $estudiante;
   }

   2 references | 0 overrides
   public function getDetallesPago(): string {
       return "Pago de $this->monto realizado el $this->fecha por el estudiante " . $this->estudiante->getNombre();
   }
}
?>
```

← → ↻ ⓘ localhost/php%20poo/ejemplousuarioscrum.php

Pago de 500 realizado el 2024-02-27 por el estudiante Juan Pérez  
Pago de 1000 realizado el 2024-06-26 por el estudiante Lina calderon