



## PRUEBA PHP

Nombre y Apellido: Norman Fernando Avendaño Cubilos\_\_CC: 1073230650\_\_\_\_

### 1. Para definir tipos de variables en PHP utilizamos la siguiente sintaxis:

a. `int $a;`

b. `a= new Integer();`

c. `$a = 0;`

d. `var$a = 0;`

### 2. Para declarar un arreglo en PHP utilizamos las siguientes sintaxis:

a. `<?php$v=array ('1'=>'juan','2'=>'pedro','3' =>'maria'); ?>`

b. `<?php$v = new array ('1'=>'juan','2' => 'pedro','3' =>'maria'); ?>`

c. `<? $v = new array {'1'=>'juan','2' => 'pedro','3' =>'maria'}); ?>`

d. Ninguna de las anteriores

### 3. Para capturar una Exception en PHP utilizamos la siguiente definición:

a. `<?phptry {...} catch Exception e){... } ?>`

b. `<?phptry {...} catch ($Exception $e){... } ?>`

c. `<?phptry {...} catch (new Exception($e){...} ?>`

d. `<?phptry {...} catch (Exception $e){... }?>`



4. Para incluir una librería en PHP las formas correctas son (Seleccione las formas correctas):

- a. `<?phpinclude 'http://www.example.com/file.txt?foo=1&bar=2';?>`
- b. `<?phpinclude 'http://www.example.com/file.php?foo=1&bar=2'; ?>`
- c. `<?phpinclude 'config.php';?>`
- d. `<?php #include <config.php?>`

5. El resultado del siguiente código es:

```
1 <?php
2 function mostrarError($e){
3     echo $e;
4 }
5
6 function ejecutarProceso($a){
7     if ($a === 'juan'){
8         echo $a." es igual juan";
9     }else{
10
11         throw new Exception($a." No es igual juan" );
12     }
13 }
14 }
15
16 try {
17     ejecutarProceso("juan");
18 } catch(Exception $e) {
19     mostrarError($e);
20 }
21
22 ?>
```

a. El código no compila da un error en la línea 7 === no es permitido en PHP.

b. juan es igual a juan

c. Exception: juan No es igual juan

d. Ninguna de las anteriores



6. Que retorna el siguiente código:

```
1 <?php
2
3 abstract class B{
4
5     function mostrarMensaje($nombre){
6         echo "Hola ".$nombre;
7     }
8
9 }
10
11 class A extends B
12 {
13 }
14
15 $b = new B;
16 $a = new A;
17 $a->mostrarMensaje("Maria");|
18 $b->mostrarMensaje("Juan");
19
20
21
22
23
24 ?>
25
```

- a. Hola Juan  
    Hola Maria
- b. Hola Maria  
    Hola Juan
- c. Hola Maria  
    Error: No se puede instanciar clase abstracta B in source\_file.php:15
- d. Error: No se puede instanciar clase abstracta B in source\_file.php:15



7. Escriba en PHP el código para realizar una conexión a base de datos con ip: 192.168.10.20 puerto = 5432 base de datos = dbprueba verificar si la conexión fue exitosa o no imprimiendo en pantalla.

```
$host = "192.168.10.20: 5432";
$usuario = "usuario";
$clave = "10101010";
$bd = " dbprueba ";
try{
    $db = new PDO('mysql:host='.$host.';dbname='.$bd .' ;charset=utf8mb4', ".$usuario.",
    ".$clave.");
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
    if($db){
        echo "Conexion Establecida";
    }else{
        echo "Error en la Conexion";
    }
}catch(Exception $e){
    echo $e;
}
```



8. Que es una variable POST y una variable GET cuál es la diferencia y realice un ejemplo en código HTML y PHP que reciba un formulario de login con correo y contraseña que redireccione a un página PHP de validación saludando al usuario que ingreso y validando el usuario y la clave en un array de usuarios y claves ya definido sin utilizar bases de datos.

**POST: VARIABLES QUE VAN POR DEBAJO DEL SERVIDOR SIN MOSTRAR NI INTERACTUAR CON EL USUARIO**

**GET: VARIABLES QUE VAN EN LA URL**

9. Escriba en PHP el código que consuma un Webservice (debe crear el webservice de prueba para el punto fácil de editar) y le envíe dos parámetros usuario y clave , luego consulte el método validar Usuario el cual retorna false o true y muestre en pantalla "Usuario valido" o "Usuario no valido" dependiendo del resultado.

10. Que patrón de diseño se está utilizando en el siguiente código?

```
1 <?php
2 interface IUserario
3 {
4     function getNombre();
5 }
6 class Usuario implements IUserario
7 {
8     public function __construct( $id ) { }
9
10    public function getNombre()
11    {
12        return "Juan";
13    }
14 }
15 class ServicioUsuario
16 {
17     public static function Create( $id )
18     {
19         return new Usuario( $id );
20     }
21 }
22
23 $uo = ServicioUsuario::Create( 1 );
24 echo( $uo->getNombre()."\n" );
25 ?>
```

**a. PatrónComposite**

**b. PatrónFacade**

**c. Patrón Factory**





I. La Segunda parte de la prueba se trata del despliegue de la aplicación, para este punto deberá subir su aplicación a un repositorio git de su preferencia e incluir un archivo **README** en el que deberá describir en forma de manual técnico cómo se reproduce la web, siendo muy específico y entendible ya que se va a seguir dicho manual como parte de la prueba.

II. Contestar el formulario de google, donde debe diligenciar información básica sobre usted, el desarrollo de la aplicación, el repositorio para descargar el código fuente de la aplicación y el apk para instalar en el dispositivo móvil.

**FORMULARIO GOOGLE:**

<https://forms.gle/YtVkmQHvReVhdqJa9>