



**PRIMER EJERCICIO PARTE PRÁCTICA DEL PROCESO
SELECTIVO DE
A1 GESTIÓN, DESARROLLO, DIGITALIZACIÓN Y SOPORTE
DE SISTEMAS DE INFORMACIÓN (ORDEN 4)
DE LA UNIVERSIDAD COMPLUTENSE DE MADRID**

**Concurso-Oposición Libre
Resolución de fecha 20-11-2024**

10 de febrero de 2026

SUPUESTO 1

Se requiere desarrollar una aplicación web en PHP con base de datos MySQL para la gestión de una biblioteca virtual. La aplicación debe permitir las siguientes funcionalidades principales:

- Registro de usuarios: Los usuarios pueden crear una cuenta proporcionando nombre, email y contraseña. La contraseña debe almacenarse de forma segura (usando hashing como password_hash en PHP).
- Inicio de sesión: Los usuarios inician sesión con email y contraseña, verificando credenciales contra la base de datos.
- Gestión de libros: Un administrador (o usuario autenticado) puede agregar nuevos libros con título, autor y estado de disponibilidad (por defecto disponible). También se puede listar todos los libros, mostrando solo los disponibles para préstamo.
- Préstamo de libros: Un usuario autenticado puede prestar un libro disponible, actualizando el estado del libro a "no disponible" y registrando el préstamo en una tabla adicional.
- Base de datos: Usa MySQL con las siguientes tablas mínimas:
 - o 'usuarios': id (INT AUTO_INCREMENT PRIMARY KEY), nombre (VARCHAR(100)), email (VARCHAR(100) UNIQUE), password (VARCHAR(255)).
 - o 'libros': id (INT AUTO_INCREMENT PRIMARY KEY), titulo (VARCHAR(200)), autor (VARCHAR(100)), disponible (BOOLEAN DEFAULT TRUE).
 - o 'prestamos': id (INT AUTO_INCREMENT PRIMARY KEY), usuario_id (INT FOREIGN KEY REFERENCES usuarios(id)), libro_id (INT FOREIGN KEY REFERENCES libros(id)), fecha_prestamo (DATE).

La aplicación debe manejar conexiones a la base de datos usando PDO o mysqli en PHP, validar entradas para prevenir inyecciones SQL, y mostrar mensajes de error o éxito en la interfaz web. Asume un entorno con sesiones PHP para mantener el estado de autenticación.

1. ¿Cuál es la forma recomendada en PHP para almacenar la contraseña de un usuario en la tabla 'usuarios' durante el registro?

- a) Usar md5(\$password)
- b) Usar password_hash(\$password, PASSWORD_DEFAULT)
- c) Almacenarla en texto plano
- d) Usar sha1(\$password)

2. Para conectar a la base de datos MySQL en PHP y prevenir inyecciones SQL, ¿qué extensión o método se recomienda usar en el supuesto?

- a) `mysql_connect` con queries directas
- b) `file_get_contents` para leer la BD
- c) PDO con prepared statements
- d) curl para acceder a MySQL

3. ¿Cuál sería una consulta SQL correcta para insertar un nuevo libro en la tabla 'libros'?

- a) `INSERT INTO libros (titulo, autor) VALUES ('Libro1', 'Autor1')`
- b) `UPDATE libros SET titulo='Libro1', autor='Autor1'`
- c) `SELECT * FROM libros WHERE titulo='Libro1'`
- d) `DELETE FROM libros WHERE autor='Autor1'`

4. En el inicio de sesión, ¿cómo se verifica la contraseña en PHP contra la almacenada en la base de datos?

- a) Comparando directamente con `==`
- b) Usando `password_verify($input_password, $hashed_password)`
- c) Usando `strcmp($input_password, $hashed_password)`
- d) Enviando la contraseña por email para verificación

5. ¿Qué constraint en MySQL asegura que el email en la tabla 'usuarios' sea único?

- a) PRIMARY KEY en email
- b) UNIQUE en email
- c) FOREIGN KEY en email
- d) INDEX en email

PREGUNTA DE RESERVA 1

6. En caso de error en una consulta SQL en PHP con PDO, ¿cómo se maneja típicamente?

- a) Usando try-catch con PDOException
- b) Ignorando el error y continuando
- c) Usando echo para mostrar el error
- d) Reiniciando el servidor MySQL

SUPUESTO 2

Dado el siguiente programa en PHP, conteste a las preguntas que se le plantean más abajo, todas ellas basadas en él:

```
<?php
$notas = [7.5, 8, 6, 9.2, 5, 4.8, 10, 6.3];
function calcularEstadisticas(array $notas): array {
    if (count($notas) === 0) {
        return [];
    }
    $total = 0;
    $maxima = $notas[0];
    $minima = $notas[0];
    $aprobados = 0;
    foreach ($notas as $nota) {
        $total += $nota;
        if ($nota > $maxima) {
            $maxima = $nota;
        }
        if ($nota < $minima) {
            $minima = $nota;
        }
        if ($nota >= 5) {
            $aprobados++;
        }
    }
    $media = $total / count($notas);
    return [
        "media" => round($media, 2),
        "maxima" => $maxima,
        "minima" => $minima,
        "aprobados" => $aprobados,
        "suspensos" => count($notas) - $aprobados
    ];
}
```

```
    ];  
}  
$resultados = calcularEstadisticas($notas);  
foreach ($resultados as $clave => $valor) {  
    echo ucfirst($clave) . ": " . $valor . "<br>";  
}  
?>
```

7. ¿Qué tipo de array es \$notas?

- a) Array asociativo.
- b) Array multidimensional.
- c) Array indexado.
- d) Array objeto.

8. ¿Qué ocurre si el array \$notas está vacío cuando se llama a calcularEstadisticas()?

- a) Se produce un error fatal.
- b) Se devuelve null.
- c) Se devuelve un array vacío.
- d) La media vale 0.

9. ¿Qué efecto tiene: array en la definición de la función?

- a) Convierte \$notas en array automáticamente.
- b) Obliga a que la función devuelva un array.
- c) Limita el tamaño del array devuelto.
- d) Es solo un comentario.

10. ¿Por qué se inicializan \$maxima y \$minima con \$notas[0]?

- a) Para evitar recorrer el array.
- b) Porque PHP lo exige.
- c) Para evitar valores incorrectos al comparar.
- d) Para mejorar el rendimiento del foreach.

11. ¿Qué función cumple `round($media, 2)`?

- a) Convierte la media en entero.
- b) Redondea la media a 2 decimales.
- c) Trunca la media.
- d) Formatea la salida en texto.

PREGUNTA DE RESERVA 2

12. ¿Qué pasaría si en la función *function calcularEstadisticas(\$notas)* se eliminara `array` como parámetro?

- a) El código dejaría de funcionar.
- b) PHP lanzaría una excepción automáticamente.
- c) La función aceptaría cualquier tipo de dato.
- d) El `foreach` no se ejecutaría.

SUPUESTO 3

Una organización pública quiere implantar un sistema web para la gestión de reclamaciones ciudadanas con el formato administrativo clásico "EXPONE / SOLICITA". El sistema debe permitir que cualquier ciudadano presente una reclamación a través de la web, y que un funcionario la estudie y emita una resolución que será enviada por correo electrónico al reclamante.

La plataforma se implementará con PHP 8.x en el lado servidor y MySQL como SGBD.

La organización dispone de un servidor de correo (SMTP autenticado) para el envío de notificaciones.

Finalmente esta es la solución propuesta:

- Cliente reclamante (navegador web)
 - o Accede a la URL pública <https://reclamaciones.ejemplo.org/nueva-reclamacion.php>.
 - o Visualiza un formulario HTML con campos: nombre, email, teléfono (opcional), asunto, EXPONE (texto largo), SOLICITA (texto largo).
 - o Rellena los campos y pulsa "Enviar reclamación".
 - o El navegador envía una petición POST al mismo script nueva-reclamacion.php con los datos del formulario (form-encoded).
- Capa de presentación (PHP – script pública nueva-reclamacion.php)
 - o Recibe la petición POST.
 - o Realiza validaciones en servidor: campos obligatorios, longitud mínima de EXPONE y SOLICITA, formato de email, normalización de cadenas.
 - o Escapa y valida los datos, y si hay errores vuelve a mostrar el formulario con mensajes de error específicos junto a cada campo.
 - o Si las validaciones son correctas, delega en la capa de acceso a datos la inserción de la reclamación en la base de datos.
- Capa de acceso a datos (PHP + PDO sobre MySQL)
 - o Establece una conexión PDO a la base de datos reclamaciones_db.
 - o Inserta una fila en la tabla reclamaciones usando consultas preparadas para evitar inyección SQL, con los campos:
 - id (INT AUTO_INCREMENT, PK)
 - nombre, email, telefono, asunto
 - expone, solicita
 - estado (valores: PENDIENTE, RESUELTA)
 - fecha_creacion, fecha_resolucion
 - resolucion (texto de la resolución)
 - funcionario (nombre o identificador del funcionario que resuelve).
- Devuelve al script de presentación un indicador de éxito o error.
- Respuesta al reclamante
 - o Si el alta se ha realizado correctamente, el script muestra una página de confirmación con un mensaje de éxito y, opcionalmente, el identificador interno de la reclamación.

- Si hay errores de base de datos u otros fallos, se muestra un mensaje de error genérico y se registran los detalles en el log del servidor.
- Cliente funcionario (navegador web interno)
 - Accede a una zona interna autenticada, por ejemplo <https://reclamaciones.ejemplo.org/admin/listado.php>.
 - Tras autenticarse (mecanismo fuera del alcance inmediato del caso, pero se asume sesión de PHP segura), visualiza un listado tabular de reclamaciones con columnas: ID, fecha, nombre, asunto, estado, enlace a detalle.
 - Al hacer clic en "Ver / Resolver" se carga <admin/detalle.php?id=X>, donde X es el ID de la reclamación.
- Capa de presentación interna (<admin/listado.php> y <admin/detalle.php>)
 - <admin/listado.php> consulta la tabla reclamaciones y muestra el listado, ordenado por fecha_creacion descendente.
 - <admin/detalle.php> recupera la reclamación específica y muestra: datos del reclamante, asunto, contenido de EXPONE y SOLICITA, estado actual y, si existe, la resolución previa.
 - El funcionario dispone de un formulario interno para introducir o modificar el texto de la resolución y el nombre del funcionario que resuelve.
- Resolución y envío de correo
 - Cuando el funcionario envía la resolución, <admin/detalle.php> recibe un POST con el texto de resolucion y funcionario.
 - El script valida que la resolución no esté vacía y abre una transacción en MySQL:
 - Bloquea la fila de la reclamación (SELECT ... FOR UPDATE) para evitar condiciones de carrera si dos funcionarios resuelven a la vez.
 - Actualiza la fila: estado = 'RESUELTA', resolucion = ..., funcionario = ..., fecha_resolucion = NOW().
 - Dentro de la misma unidad lógica de trabajo, el sistema construye el cuerpo del correo con saludo, referencia al asunto y texto de la resolución.
 - Para el envío del correo, en el entorno definitivo se emplea una librería específica (por ejemplo, PHPMailer) configurada para conectarse al servidor SMTP corporativo con TLS, autenticación y logs de entrega.
 - En caso de error crítico al enviar el correo, se decide (según política) si se hace rollback de la transacción o si se registra la incidencia para reintento posterior.
- Entrega del correo
 - El servidor SMTP recibe el mensaje y lo enruta al buzón del reclamante.
 - El reclamante recibe un correo con el texto de la resolución y, opcionalmente, el identificador de su reclamación para referencia futura.

Responda a estas cuestiones para determinar la mejor implementación:

13. ¿Cuál es la implementación MÁS ROBUSTA para validar el formato del email del reclamante?

a) ```php

```
if (!preg_match('/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/',  
$email)) { error; }
```

b) ```php

```
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) { error; }
```

c) ```php

```
if (strpos($email, '@') === false || strpos($email, '.') === false) { error; }
```

d) ```php

```
if (empty($email) || strlen($email) > 254) { error; }
```

14. ¿Cómo asegurar que la actualización de la reclamación sea atómica y confiable, y que el correo se envíe solo si esa actualización tuvo éxito?

a) ```php

```
$pdo->beginTransaction();  
  
$stmt = $pdo->prepare("UPDATE reclamaciones SET estado='RESUELTA' WHERE  
id=?");  
  
$stmt->execute([$id]);  
  
mail($email, $asunto, $mensaje); // Sin manejo de errores  
  
$pdo->commit();
```

b) ```php

```
$pdo->beginTransaction();  
  
$stmt = $pdo->prepare("SELECT * FROM reclamaciones WHERE id=? FOR  
UPDATE");  
  
$stmt->execute([$id]);  
  
$stmtUpd = $pdo->prepare("UPDATE reclamaciones SET estado='RESUELTA'  
WHERE id=?");  
  
$stmtUpd->execute([$id]);  
  
mail($email, $asunto, $mensaje);  
  
$pdo->commit();
```

c) ```php

```
try {  
    $pdo->beginTransaction();  
    $stmtUpd = $pdo->prepare("UPDATE reclamaciones SET estado='RESUELTA'  
    WHERE id=?");  
    $stmtUpd->execute([$id]);  
    $pdo->commit();  
    mail($email, $asunto, $mensaje); // Correo FUERA de transacción  
} catch(Exception $e) {  
    $pdo->rollBack();  
}
```

d) ```php

```
$stmt = $pdo->prepare("UPDATE reclamaciones SET estado='RESUELTA' WHERE  
id=?");  
$stmt->execute([$id]);  
mail($email, $asunto, $mensaje);
```

15. ¿Cuál es la implementación SEGURA para mostrar el asunto de una reclamación en admin/listado.php?

a) ```php

```
echo "<td>" . $rec['asunto'] . "</td>";
```

b) ```php

```
echo "<td>" . htmlspecialchars($rec['asunto'], ENT_QUOTES | ENT_HTML5, 'UTF-  
8') . "</td>";
```

c) ```php

```
echo "<td>" . strip_tags($rec['asunto']) . "</td>";
```

d) ```php

```
echo "<td>" . urlencode($rec['asunto']) . "</td>";
```

16. Índices en esquema de BD. ¿Cuál es el esquema de MySQL MÁS OPTIMIZADO para soportar 100.000 reclamaciones anuales?

a) Sin índices adicionales.

b) `` `sql

```
ALTER TABLE reclamaciones ADD INDEX idx_estado_fecha (estado, fecha_creacion);
```

```
ALTER TABLE reclamaciones ADD INDEX idx_email (email);
```

c) `` `sql

```
ALTER TABLE reclamaciones ADD UNIQUE KEY idx_email (email);
```

d) `` `sql

```
ALTER TABLE reclamaciones ADD FULLTEXT(expone, solicita);
```

17. ¿Cuál es el manejo de errores MÁS PROFESIONAL en `config/db.php`?

a) Sin configuración especial.

b) `` `php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

c) `` `php

```
error_reporting(0);
```

d) `` `php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_SILENT);
```

PREGUNTA RESERVA 3

18. El sistema de reclamaciones expone un endpoint público PHP que permite adjuntar documentos. Se detecta que algunos usuarios maliciosos intentan subir ficheros PHP renombrados a JPG para ejecutar código en el servidor. ¿Cuál es la medida de seguridad más efectiva en profundidad para mitigar este riesgo?

a) Almacenar los ficheros subidos fuera del árbol público de documentos del servidor web y servirlos a través de un script que controle el tipo MIME

b) Validar la extensión con PHP y rechazar todo lo que no sea .jpg o .png

c) Confiar en el lado cliente para validar el tipo de archivo mediante JavaScript

d) Cambiar el nombre del fichero a un UUID, pero mantenerlo en un directorio accesible públicamente

SUPUESTO 4

Supuesto práctico sobre la tecnología .NET codificado en C#, responda a las siguientes preguntas:

19. Para crear una array con el comando "List" que contenga una serie de valores "string" obtenidos de la consulta al método "OBTENCION_PROCESOS()", ¿cuál de los siguientes comandos es el correcto?

- a) `List<string> _Proc = new List<string>(OBTENCION_PROCESOS());`
- b) `List<string> _Proc = OBTENCION_PROCESOS();`
- c) `List<string> _Proc = new List() => (OBTENCION_PROCESOS());`
- d) `List<string> _Proc = new (OBTENCION_PROCESOS());`

20. Necesitamos que un programa ejecute una serie de operaciones cada 2 segundos, para ello utilizamos un temporizador con la sentencia "DispatcherTimer Timer1 = new DispatcherTimer();", ¿cuál de las siguientes opciones es la correcta?

- a) `DispatcherTimer.Interval = new TimeSpan(1000);`
- b) `DispatcherTimer.Interval = TimeSpan.FromSeconds(2);`
- c) `DispatcherTimer.Interval = TimeSpan(2s);`
- d) `DispatcherTimer.Interval = new TimeSpan(x => 1000);`

21. Desarrollando en el entorno WPF, para que la visualización de la pantalla inicial del programa esté centrada ¿qué debemos incluir?

- a) `WindowStartupLocation="Center"`
- b) `WindowLocation="Center"`
- c) `WindowStartupLocation="CenterScreen"`
- d) `WindowLocation="Center"`

22. Desarrollando en el entorno WPF, para que un botón deje de visualizarse, una vez haya sido pulsado por el usuario, ¿cuál de estas opciones hay que utilizar?

- a) `Ejecucion.Visibility = Visibility.Hidden;`
- b) `Ejecucion.Visibility = False;`
- c) `Ejecucion.VisibilityStartup = False;`
- d) `Ejecucion.VisibilityStartup = Hidden;`

23. Desarrollando en el entorno WPF ¿cuál de las siguientes opciones es la correcta para inhabilitar un botón "Ejecucion" de la pantalla?

- a) Ejecucion.IsEnabled = false;
- b) Ejecucion.IsEnabled = False;
- c) Ejecucion.Enabled = false;
- d) Ejecucion.Enabled = False;

PREGUNTA DE RESERVA 4

24. Desarrollando en el entorno WPF, si queremos añadir un texto a "ListBox1" (siendo "ListBox1" un componente de tipo ListBox creado en un programa), ¿qué opción es la correcta?

- a) ListBox1->Add("Texto");
- b) ListBox1.Items->Add("Texto");
- c) ListBox1.Items.Add("Texto");
- d) ListBox1->Add.Items("Texto");

SUPUESTO 5

El siguiente código gestiona una pequeña lista de libros y realiza una búsqueda filtrada:

```
import java.util.;
import java.util.stream.;
class Libro {
    String titulo;
    boolean disponible;
    Libro(String t, boolean d) { titulo = t; disponible = d; }
    public String getTitulo() { return titulo; }
    public boolean isDisponible() { return disponible; }
}
public class GestionBiblioteca {
    public static void main(String[] args) {
        List libros = Arrays.asList(
            new Libro("Java Avanzado", true),
            new Libro("Spring Boot", false),
            new Libro("Microservicios", true)
        );
        Optional resultado = libros.stream()
            .filter(Libro::isDisponible)
            .map(Libro::getTitulo)
            .filter(t -> t.startsWith("J"))
            .findFirst();
        resultado.ifPresent(System.out::println);
    }
}
```

25. ¿Cuál es el resultado de ejecutar este programa por consola?

- a) "Java Avanzado" y "Microservicios"
- b) "Java Avanzado"
- c) No imprime nada
- d) Lanza una excepción de puntero nulo (NullPointerException)

26. En la línea ".filter(Libro::isDisponible)", ¿qué tipo de interfaz funcional se está implementando mediante la referencia a método?

- a) Consumer < Libro >
- b) Supplier < Boolean >
- c) Predicate < Libro>
- d) Function < Libro, Boolean >

27. ¿Qué sucedería si cambiáramos "Arrays.asList(...)" por "List.of(...)" e intentáramos añadir un nuevo libro a la lista mediante "libros.add(...)"?

- a) Se añade correctamente.
- b) Error de compilación.
- c) Lanza una "UnsupportedOperationException" en tiempo de ejecución.
- d) El libro se añade pero el Stream no lo procesa.

28. ¿Cuál es el propósito del método ".findFirst()" en este flujo de datos?

- a) Es una operación intermedia que busca el primer elemento.
- b) Es una operación terminal que devuelve un contenedor de tipo Optional.
- c) Ordena la lista antes de devolver el primer resultado.
- d) Ejecuta el filtrado de forma paralela para optimizar recursos.

29. ¿Cuál es la principal ventaja de utilizar "Optional<String>" en lugar de devolver directamente un "String" o "null"?

- a) Permite que el código se ejecute más rápido.
- b) Obliga al desarrollador a gestionar explícitamente la ausencia de valor, reduciendo errores.
- c) Convierte automáticamente el resultado a mayúsculas.
- d) Permite almacenar múltiples valores en una sola variable.

PREGUNTA DE RESERVA 5

30. Si ningún libro cumpliera la condición `".filter(t->t.startsWith("J"))`, ¿qué valor contendría la variable "resultado"?

- a) null
- b) Una cadena vacía ("")
- c) Un objeto Optional vacío (Optional.empty())
- d) Se lanzaría una "NoSuchElementException"