Puedes mejorar la presentación de la salida formateándola mejor. Puedes usar PHP_EOL para mantener saltos de línea adecuados y mejorar la legibilidad. Prueba con este código:

Código Mejorado:

```
$listaCategorias = $conexion->prepare("SELECT id, categoria FROM categorias");

$listaCategorias->execute();


$registrosListado = [];

while ($listado = $listaCategorias->fetch(PDO::FETCH_ASSOC)) {

   $registrosListado[] = $listado['id'] . " : " . $listado['categoria'];

}

$listaCategorias = null;


if ($rdo == 0) {

   salidaDatos(

      "Identificador de categoría inadecuado." . PHP_EOL .

      "Las categorías adecuadas son:" . PHP_EOL . PHP_EOL .

      implode(PHP_EOL, $registrosListado),

      array('HTTP/1.1 400 Bad Request')

   );

   exit;

}
```

Explicación de la mejora:

1. **Uso de PHP_EOL**: Se usa para asegurar saltos de línea adecuados en diferentes entornos.
2. **Mejor formateo**: Se separan los elementos de la lista con saltos de línea (PHP_EOL), lo que facilita la lectura.
3. **Evita la coma final**: En la versión anterior, la lista tenía comas innecesarias, ahora se presenta en una lista ordenada.

Ejemplo de Salida Mejorada:

Identificador de categoría inadecuado.

Las categorías adecuadas son:

1 : Turismo cultural

2 : Turismo deportivo y de eventos

3 : Turismo sostenible

4 : Turismo de aventura

5 : Turismo recreativo o viajes de ocio

6 : Turismo de negocios o corporativo

7 : Turismo rural

Esto hará que la respuesta en Postman sea mucho más clara y fácil de leer.

Para calcular la diferencia entre una fecha almacenada en una variable y el día actual en PHP, puedes usar la clase DateTime y la función diff().

Código PHP:

```
<?php

// Fecha almacenada en la variable

$fechaGuardada = new DateTime("2025-02-13 21:30:00");


// Fecha actual

$fechaActual = new DateTime(); // Obtiene la fecha y hora actual


// Calcula la diferencia

$diferencia = $fechaActual->diff($fechaGuardada);


// Formatear la salida

echo "Diferencia: " . $diferencia->format('%y años, %m meses, %d días, %h horas, %i minutos, %s segundos');

?>
```

Explicación:

1. **Se crea un objeto DateTime** con la fecha almacenada.
2. **Se obtiene la fecha y hora actual** con new DateTime().
3. **Se calcula la diferencia** con $fechaActual->diff($fechaGuardada), lo que devuelve un objeto DateInterval.
4. **Se formatea la salida** con %y años, %m meses, %d días, %h horas, %i minutos, %s segundos.

Ejemplo de salida (si hoy es 2025-02-08 18:15:00):

Diferencia: 0 años, 0 meses, 5 días, 3 horas, 15 minutos, 0 segundos

Si quieres obtener la diferencia en segundos, usa:

$segundosDiferencia = $fechaGuardada->getTimestamp() - $fechaActual->getTimestamp();

echo "Diferencia en segundos: " . $segundosDiferencia;

Esto te devolverá la diferencia en segundos entre ambas fechas.

Si la variable $fechaguardada es un string, puedes convertirla en un objeto DateTime antes de calcular la diferencia. Aquí tienes el código actualizado:

Código PHP:

```
<?php

// Variable con la fecha en formato string

$fechaguardada = "2025-02-13 21:30:00";


// Convertir el string a un objeto DateTime

$fechaGuardada = new DateTime($fechaguardada);


// Obtener la fecha y hora actual

$fechaActual = new DateTime();


// Calcular la diferencia
```

$diferencia = $fechaActual->diff($fechaGuardada);

// Formatear la salida

echo "Diferencia: " . $diferencia->format('%y años, %m meses, %d días, %h horas, %i minutos, %s segundos');

?>

Explicación:

1. **Convertimos el string** $fechaguardada en un objeto DateTime con new DateTime($fechaguardada).
2. **Obtenemos la fecha actual** con new DateTime().
3. **Calculamos la diferencia** con $fechaActual->diff($fechaGuardada), que devuelve un objeto DateInterval.
4. **Formateamos la salida** usando format().

Ejemplo de Salida (si hoy es 2025-02-08 18:15:00):

Diferencia: 0 años, 0 meses, 5 días, 3 horas, 15 minutos, 0 segundos

---

Si necesitas la diferencia en segundos:

$segundosDiferencia = strtotime($fechaguardada) - time();

echo "Diferencia en segundos: " . $segundosDiferencia;

Esto calcula la diferencia absoluta en segundos entre ambas fechas.

Para obtener la diferencia en días como un número y determinar si es menor o mayor a 7 días, puedes usar DateTime y diff(), o convertir la fecha a timestamp con strtotime().

- **Opción 1: Usando diff()**

<?php

$fechaguardada = "2025-02-13 21:30:00";

// Convertir el string a un objeto DateTime

$fechaGuardada = new DateTime($fechaguardada);

```
$fechaActual = new DateTime();

// Calcular la diferencia en días
$diferenciaDias = $fechaActual->diff($fechaGuardada)->days; // Devuelve solo el número de días

// Comprobar si es inferior o superior a 7 días
if ($diferenciaDias < 7) {
    echo "La diferencia es menor a 7 días.";
} else {
    echo "La diferencia es mayor o igual a 7 días.";
}
?>
```

🔹 **Opción 2: Usando strtotime() (más simple)**

```
<?php
$fechaguardada = "2025-02-13 21:30:00";

$diferenciaSegundos = strtotime($fechaguardada) - time();
$diferenciaDias = $diferenciaSegundos / (60 * 60 * 24); // Convertir segundos a días

if ($diferenciaDias < 7) {
    echo "La diferencia es menor a 7 días.";
} else {
    echo "La diferencia es mayor o igual a 7 días.";
}
?>
```

🔥 **¿Cuál opción elegir?**

✅ **Usa diff()** si necesitas manejar fechas con más precisión.
✅ **Usa strtotime()** si solo necesitas comparar días de forma rápida.

Ambas opciones funcionan y te darán un número que puedes comparar con 7 días. 🚀

La función strtotime() en PHP convierte una fecha en formato string en un **timestamp UNIX**, que es un número entero que representa la cantidad de segundos transcurridos desde el **1 de enero de 1970 a las 00:00:00 UTC**.

📌 **Ejemplo de uso:**

```
<?php

$fechaguardada = "2025-02-13 21:30:00";


// Convierte la fecha en un timestamp UNIX (segundos desde 1970)

$timestamp = strtotime($fechaguardada);


echo "Timestamp: " . $timestamp;

?>
```

🔹 **Salida esperada (valor puede cambiar según la zona horaria):**

```
Timestamp: 1744781400
```

🚀 **¿Para qué sirve strtotime()?**

✅ **Permite hacer cálculos de fechas fácilmente**, porque los timestamps son números enteros.
✅ **Permite restar y sumar fechas** sin necesidad de objetos DateTime.
✅ **Puede interpretar frases en inglés**, como "next Monday", "tomorrow", "+2 weeks", etc.

📌 **Ejemplo de comparación con la fecha actual:**

```
<?php

$fechaguardada = "2025-02-13 21:30:00";


// Timestamp de la fecha guardada y de la fecha actual
```

```
$timestampGuardado = strtotime($fechaguardada);

$timestampActual = time(); // Timestamp actual


// Calcular diferencia en días

$diferenciaDias = ($timestampGuardado - $timestampActual) / (60 * 60 * 24);


echo "Diferencia en días: " . floor($diferenciaDias);

?>
```

- 🔹 time() devuelve el timestamp actual.
- 🔹 Se divide entre 86400 (60 * 60 * 24) para obtener los días.
- 🔹 floor() redondea hacia abajo para evitar decimales.

🔥 **¿Cuándo usar strtotime() en vez de DateTime?**

✅ Cuando necesitas solo el número de segundos y quieres hacer cálculos rápidos.
✅ Cuando trabajas con fechas dinámicas en formato string (+1 week, next Sunday).
✅ Cuando prefieres una solución más simple en lugar de objetos DateTime.

Si necesitas más funcionalidades avanzadas, mejor usa DateTime. 🚀