



**CICLO: D.A.W.**  
**MÓDULO DE DESARROLLO WEB**  
**ENTORNO SERVIDOR**

# **Tarea N° 08**

**Alumno:**  
**Armando Herrero Silva**  
**76442135W**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

*© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.*

## Contenido

1.	RA08_e. ....	2
2.	RA08_f. ....	3
3.	RA08_g. ....	6

## 1. RA08\_e.

- *Crear un formulario que contiene un campo de texto con id "texto", configurar y/o verificar el campo para que solamente se puedan introducir caracteres.*
- *Para conseguir la máxima nota, crear un formulario Laravel.*

```
<body>
  <h1>Tarea 8 DWES</h1>
  <form id="formulario" action="" method="post">
    <label for="texto">Escriba un texto sin números:</label><br><br>
    <textarea id="texto" name="texto" required></textarea><br><br>
    <input type="submit" value="Enviar">
  </form>
  <div>
    <p id="resultado"></p>
  </div>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    document.getElementById('formulario').addEventListener('submit', function(event) {
      var texto = document.getElementById('texto').value;
      var verificacion = /^[a-zA-Z\s]+$/;

      event.preventDefault();

      if (!verificacion.test(texto)) {
        $("#resultado").html('<strong>El texto no es correcto porque contiene números.</strong>');
      } else {
        $("#resultado").text('El texto es correcto');
      }
    });
  </script>
```

Hemos usado JavaScript para realizar el formulario junto a jQuery. Para comprobar si el texto introducido era correcto hemos usado una expresión regular (regex). Luego para evitar que el formulario se envíe automáticamente, incluso al recargar la página, usamos el “event.preventDefault();”. Luego con test comparamos la regex con el texto introducido con “.test” y dependiendo del resultado ponemos un mensaje u otro.

The image displays four screenshots of a web form titled "Tarea 8 DWES". Each screenshot shows a text input field and an "Enviar" button.

- Top Left:** The form is empty. The text input field is blank, and the "Enviar" button is visible.
- Top Right:** The form shows a successful submission. The text input field contains "Hola", and the message "El texto es correcto" is displayed below the button.
- Bottom Left:** The form shows an incorrect submission. The text input field contains "Hola2", and the message "El texto no es correcto porque contiene números." is displayed below the button.
- Bottom Right:** The form shows another incorrect submission. The text input field contains "123", and the message "El texto no es correcto porque contiene números." is displayed below the button.

## 2. RA08\_f.

*Partiendo de la base de datos de las tareas 6 y 7:*

- *Según las letras introducidas en el campo con id "texto", se muestran todos los libros que contengan esos caracteres.*
- *Se utiliza AJAX, para evitar recargar la página para ver el resultado de la búsqueda.*
- *Para conseguir la máxima nota, configurar la respuesta en Laravel. También se puede reutilizar la API de la tarea 7, con las modificaciones necesarias.*

En este apartado hemos usado una API externa de la serie de Rick y Morty para buscar algunos de sus personajes.

Primero tenemos que poner el script para poder usar jQuery:

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Buscar Personajes de Rick y Morty</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<style>
  body{
    background-color: black;
    color: white;
```

Ahora usamos la sentencia de Ajax para enlazar la Api, establecer el método y decirle lo que queremos que haga en caso de que haya éxito en nuestra consulta. También, además del nombre de los personajes vamos a poner su imagen correspondiente:

```
<script>
$(document).ready(function() {
    $('#texto').on('input', function() {
        let query = $(this).val().toLowerCase();

        if (query.length > 0) {
            $.ajax({
                url: `https://rickandmortyapi.com/api/character/?name=${query}`,
                method: 'GET',
                success: function(response) {
                    if (response.results && response.results.length > 0) {
                        $('#personajes').empty();
                        $('#div').show();

                        response.results.forEach(function(character) {
                            $('#personajes').append(`
                                <div id="info">
                                    <h3>${character.name}</h3>
                                    
                                </div>
                            `);
                        });
                    } else {
                        $('#personajes').html('<p>No se encontraron personajes.</p>');
                    }
                },
                error: function() {
                    $('#personajes').html('<p>Hubo un error al obtener los datos.</p>');
                }
            });
        } else {
            $('#personajes').empty();
            $('#div').hide();
        }
    });
});
```

## Busca Personajes de Rick y Morty

Buscar Personaje:

**Resultados:**

## Busca Personajes de Rick y Morty

Buscar Personaje:

**Resultados:**

Morty Smith 	Summer Smith 	Beth Smith 	Jerry Smith 
Beth Smith 	Beth Smith 	Jerry Smith 	Jerry Smith 
Joyce Smith 	Leonard Smith 	Morty Smith 	Morty Smith 
			

Activar Windows  
Ve a Configuración para activar Windows.

### 3. RA08\_g.

- **Desarrollar una página sencilla que permita probar la aplicación. Se valorará el aspecto gráfico de la solución. Para conseguir la máxima nota, configurar una aplicación Laravel**
- **Utilizando la sintaxis básica de PHPDoc, comentar las funciones creadas. Crear un comentario `/** */` en cada función que incluya un comentario sobre la función**
  - **Utilizar `@param` para describir los parámetros que recibe la función y `@return` para comentar el valor resuelto**
  - **Se puede verificar el correcto funcionamiento de los comentarios creados usando PHPDocumentor.**
  - **Descargar el fichero `phpdocumentor.phar` de la página de PHPDocumentor**
  - **Crear un directorio `doc` fuera del directorio donde se esté creando la página web**
  - **Ejecutar `c:\xampp\php\bin\php.exe phpDocumentor.phar -d c:\directorio de trabajo -t c:\directorio de documentacion creado`. Estos directorios son de ejemplo.**
  - **Tras esto se debería haber creado una página web con el contenido de la documentación.**

Como no tenemos parte en PHP, solo vamos a poner los correspondientes comentarios:

```
<script>
    $(document).ready(function() {

        /**
         * La función va a manejar el evento que se produce al introducir un texto.
         * Buscará las coincidencias dentro de la Api de los personajes de Rick y Morty.
         *
         * @param {string} query - El texto introducido por el usuario para realizar la búsqueda.
         * @return {void} - No hay ningun resultado, pero se actualiza el DOM.
         */

        $('#texto').on('input', function() {
            let query = $(this).val().toLowerCase();

            if (query.length > 0) {
```