



DESARROLLO WEB EN ENTORNO CLIENTE

PROYECTO ECOHUERTO

GRUPO 2

Contenido

PROYECTO AGUA, VIDA Y BIENESTAR	2
1- Guía para configura y ejecutar la aplicación en Laravel en tu Servidor Local.	2
2- Aplicación Web.....	3
Vistas	3
Controladores.....	5
3- Resultado en el navegador.....	6
4- Participantes del Grupo 2.....	7

PROYECTO AGUA, VIDA Y BIENESTAR

1- Guía para configura y ejecutar la aplicación en Laravel en tu Servidor Local.

- Descargar el proyecto:
 - El proyecto está en un archivo comprimido. Descarga y descomprime el archivo en la carpeta htdocs de tu servidor web.
- Requisitos:
 - PHP: la versión 7.
 - Xampp, que ya incluye PHP y MySQL. (Se adjunta el script)
 - Laravel 10.
 - Composer
 - Visual Studio Code
- Base de datos:
 - Se tienen que crear una base de datos llamada practicacentros para que luego funcione la conexión desde Laravel.
 - Se adjunta el script con todas las tablas, se debe ejecutar en phpMyAdmin.
- Instalar las dependencias en Laravel:
 - Asegúrate de tener Composer instalado: **composer --version**
 - Navegar a la carpeta del proyecto. (**htdocs**)
 - Ejecutar el comando **composer install** para instalar las dependencias del proyecto.
- Configurar el archivo .env:

Editar el archivo. env y reemplazar los valores:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=practicacentros
DB_USERNAME=root
DB_PASSWORD=
```
- Iniciar la aplicación:
 - Ejecuta el comando **php artisan serve** en la terminal para iniciar el servidor web de desarrollo de Laravel. La aplicación estará disponible en la URL <http://127.0.0.1:8000/>

2- Aplicación Web

Dentro del este proyecto, vamos a comentar las partes que se ha utilizado JQUERY y AJAX en combinación con PHP. También indicar que partes del código no se han visto en clase. Se han tenido que usar para poder combinar el proyecto ya empezado para la otra asignatura.

Vistas

- ListadoExperiencias.

A esta vista, se le pasa la información mediante los métodos del controlador a la página, en el foreach, se le establece que en cada iteración el "id" del <div> lleve el nombre del centro, para luego recogerlo en el script.

```
<div id="{{ $experiencia->centro }}" class="card col-9 align-self-center margen" style="width: 18rem;">
```

Posteriormente, cuando la página este cargada completamente, se ejecutará el cambio de color. Para ello, se utiliza JavaScript con jQuery, esto cambia el color de fondo de los elementos div según su ID.

```
<script>
$(document).ready(function() {
    // Cambiar el color de fondo de los elementos según su ID
    $("div[id*='IBQ']").css("background-color", "#D4EFD0");

    $("div[id*='IPCC']").css("background-color", "#F5B7B1");

    $("div[id*='IBCF']").css("background-color", "#D6EAF8");

    $("div[id*='IMSI']").css("background-color", "#FDEBD0");

    $("div[id*='IPJB']").css("background-color", "#EBDEF0");

});
</script>
```

- ExperienciasUsuarios.

En esta vista JQuery se encarga de:

- **Capturar el evento "submit" del formulario:** Cuando el usuario da clic al botón "Enviar", jQuery activa una función para manejar la información del formulario.
- **Evitar el envío tradicional del formulario:** jQuery impide que el formulario se envíe de la manera de siempre, lo que significa que la página no se refrescará y no se perderá la información del usuario.
- **Crear un objeto con los datos del formulario:** Usando jQuery, se crea un objeto que guarda todos los datos del formulario, incluso los archivos.

En esta vista AJAX se encarga de:

- **Enviar la información del formulario al servidor:** Se usa una solicitud AJAX para mandar la información del formulario a una ruta específica en el servidor. Esto se hace de forma asíncrona, sin recargar la página.
- **Especificar el método POST:** AJAX especifica que se está enviando información al servidor usando el método POST, ideal para enviar datos.
- **Procesar y codificar los datos del formulario:** AJAX asegura que los datos del formulario se envíen al servidor en un formato que este pueda entender. Así se garantiza que la información llegue correctamente y de forma segura.
- **Manejar la respuesta del servidor:** Después de enviar la información, AJAX maneja lo que el servidor responde. Si todo sale bien, se puede mostrar un mensaje al usuario y hacer una acción.
- **Manejar errores:** Si hay algún problema con la solicitud, como una conexión perdida o un error en el servidor, AJAX puede manejarlo y mostrar un mensaje al usuario para informarle sobre el problema.

```
<!-- Script JavaScript para manejar el envío del formulario a través de AJAX -->
<script>
    $(document).ready(function() {
        //Incluye CSRF en las solicitudes
        $.ajaxSetup({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            }
        });
        $("#formulario").submit(function(event) {
            event.preventDefault();

            // Creación de un objeto FormData con los datos del formulario
            var formData = new FormData($(this)[0]);
            let idUsuario = $("#usuario").val();

            // Envío de la solicitud AJAX para guardar la experiencia
            $.ajax({
                url: '/altaExperiencia/' + idUsuario, //Ruta que guarda la experiencia
                type: 'POST', //Método
                processData: false,
                contentType: false,
                enctype: 'multipart/form-data', // Tipo de codificación de datos
                data: formData, //Datos del formulario
                success: function(response) { //Respuesta con éxito
                    console.log(response);
                    window.location.href = "/experienciasUsuario/" + idUsuario;
                },
                error: function(err) { //Respuesta que muestra si hay error
                    console.log(err);
                }
            });
        });
    });
</script>
```

Controladores

- ExperienciasController.

En este método llamado guardar vamos a ver qué hace y dónde se utiliza AJAX:

- **Creación de Experiencia:** Este método crea una nueva experiencia y le asigna valores como el borrador, el texto y el título, utilizando los datos recibidos en la solicitud HTTP. Luego, guarda esta experiencia en la base de datos.
- **Guardado de Archivo Adjunto:** Si se adjunta un archivo en la solicitud, el método lo guarda en la base de datos como parte de la experiencia. Primero, verifica si se envió un archivo adjunto. Si es así, lo guarda en la base de datos como un blob binario.
- **Respuesta AJAX:** Después de guardar la experiencia y el archivo adjunto, el método busca y recupera la información del usuario relacionado con esta experiencia. Luego, devuelve esta información en formato JSON como respuesta a la solicitud. En esta parte se utiliza una solicitud AJAX, ya que la respuesta se devuelve en formato JSON (respuesta recogida en la vista de experienciaUsuario).

```
1 reference | 0 overrides
public function guardar(Request $request, $id)
{
    //Crea la experiencia
    $exp = new Experiencia();
    $exp->borrador = $request->borrador;
    $exp->texto = $request->texto;
    $exp->título = $request->título;
    $exp->usuario_id = $id;
    $exp->save();

    $file = $request->file('adjunto');
    // Guarda el archivo adjunto
    if($file != null){
        $arch = new Archivo();
        $file = $request->file('adjunto');
        $fileData = file_get_contents($file->getRealPath());
        if ($fileData !== false) {
            $arch->archivo = $fileData;
            $arch->experiencias_id = $exp->id;
            $arch->save();
        }
    }

    // Recoge el usuario y se pasa el usuario en formato JSON
    $usuario = Usuario::find($id);
    return response()->json($usuario);
}
```

3- Resultado en el navegador

- Experiencias Publicadas: se vería así, con un título, una descripción, el centro al que pertenecen y el usuario que lo ha escrito. Como se indicaba en la actividad, el IBQ sale de color verde.



- Añadir experiencia: el usuario tiene que estar logueado para poder hacerlo.



En la barra se muestra el usuario.



Si no esta registrado no aparece la opción de la nueva experiencia.



4- Participantes del Grupo 2

- Raul Antuña de la Rosa
- Noemí Juan Pérez
- Alejandro Menéndez Clouxç
- Benigno Ruisanchez Santoveña