

CREAR UN CRUD

Para ver la web, lo primero que debemos hacer es ejecutar los siguientes comandos:

- `php artisan serve &`
- `npm run dev`
- `docker ps`, si vemos que hay contenedores levantados, lo primero que debemos hacer es pararlos y, para ello, ejecutaremos el comando `docker stop` seguido del nombre del contenedor y, una vez parados los contenedores, volveremos a levantarlos. Para levantar los contenedores, deberemos ejecutar el comando `docker compose up -d`

Ahora, una vez realizamos los pasos anteriores, empezamos a crear nuestro proyecto y, vamos a seguir los siguientes pasos:

1. Creamos el ecosistema (todas las clases que tenemos)

```
``` bash
php artisan make:model Profesor --all
```
```

Esto crea los siguientes elementos:

- *Migración*: es una clase que permite crear las tablas
- *Factoría*: crea valores para una fila de la tabla
- *Seeder*: invoca a la factoría de un modelo e inserta los valores en la tabla
- *Controlador*: son los métodos (respuesta) que vamos a ejecutar ante solicitudes del cliente
- *Modelo*: clase para interactuar con una tabla concreta de la base de datos y hacer acciones típicas como insertar, borrar, consultar y actualizar.
- *Request*: autoriza y valida los datos del formulario.
- *Policy*: suena a políticas que definirán accesos.
- *Rutas*: tenemos que crearlas y dirán que recursos ofrece nuestra aplicación

2. Ajustamos los valores por defecto

Como el modelo se llama Profesor y la tabla quiero que se llame profesores y no professors, tengo que indicarlo. Para ello, voy al `modelo.php` y, allí, debajo de `use`, añado:

- `protected $table="profesores";`

3. Creamos las rutas

Debemos ir al fichero de rutas (`web.php`) y, allí, agregamos:

- `use App\Http\Controllers\ProfesorController;`
- `Route::resource("profesores", ProfesorController::class);`

El método `resource` crea todas las solicitudes posibles para acceder a un recurso.

Después ejecutamos el comando `php artisan route:list --path="profesores"`; al ejecutarlo, vemos las rutas.

4. Creamos la tabla

Para crear la tabla, debemos ejecutar el comando `php artisan migrate`

5. Poblamos la tabla

Ahora para poblar la tabla, desde el fichero seeder se llama al factory. Si ejecutamos el comando `php artisan migrate: fresh --seed`, lo que nos hace este comando es borrar y crear de nuevo la tabla.

6. Creamos el botón “Profesores”

En el fichero `nav.blade.php`, debemos crear un botón llamado Profesores para que al clicar sobre éste, me lleve a una lista de profesores y, ésta, me permita crear, editar y borrar profesores.

7. Creamos el campo dni en profesores

En el fichero `profesores.listado.blade.php`, añadimos el campo dni

8. Creamos la paginación dentro de profesores

Para crear la paginación dentro de profesores, debemos de ir al fichero `ProfesorController.php`, que se encuentra dentro de la carpeta `auth`, localizada dentro de la carpeta `controllers` y, dicha carpeta, se localiza dentro de la carpeta `http` y, ésta, la podemos localizar dentro de la carpeta `app`.

Accedemos al fichero `ProfesorController.php` y, en él una vez que lo redactemos, debería de quedar de la siguiente manera:

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreProfesorRequest;
use App\Http\Requests\UpdateProfesorRequest;
use App\Models\Profesor;

2 usages  MAlejandroR *
class ProfesorController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    MAlejandroR *
    public function index()
    {
        $profesores = Profesor::all();

        return view("profesores.listado", ["profesores" => $profesores]);
        //
    }
}
```

```

/**
 * Show the form for creating a new resource.
 */
@MAlejandraR
public function create()
{
    return view ("profesores.create");
    //
}

/**
 * Store a newly created resource in storage.
 */
@MAlejandraR
public function store(StoreProfesorRequest $request)
{
    $valores = $request->input();
    $profesor = new Profesor($valores);
    $profesor->save();
    session()->flash("status", "Se ha creado el profesor $profesor->nombre");

    $profesores = Profesor::all();
    return view ("profesores.listado",["profesores"=>$profesores]);
}

```

```

/**
 * Display the specified resource.
 */
@MAlejandraR
public function show(Profesor $profesor)
{
    //
}

/**
 * Show the form for editing the specified resource.
 */
@MAlejandraR *
public function edit(Profesor $profesor)
{
    // $profesor= Profesor::find($id);
    return view ("profesores.edit", ['profesor'=>$profesor]);
    //
}

```

```

/**
 * Update the specified resource in storage.
 */
MAlejandroR
public function update(UpdateProfesorRequest $request, Profesor $profesor)
{
    session()->flash("status", "Se ha actualizado El profesor $profesor->nombre");

    //
}

/**
 * Remove the specified resource from storage.
 */
no usages MAlejandroR
public function destroy(int $id)
{
    $profesor = Profesor::find($id);
    $profesor->delete();
    $profesores = Profesor::all();
    session()->flash("status", "Se ha borrado El profesor $profesor->nombre");
    return view ("profesores.listado", ["profesores"=>$profesores]);
    //
}
}

```

Luego, accedemos al fichero listado.blade llamamos al componente tailwind.blade.php. Nuestro fichero listado.blade quedaría de la siguiente manera:

```

<x-layouts.layout>
<div class="overflow-x-auto max-h-full">
    @if(session('status'))
        <script>
            Swal.fire('{{session("status")}}');
        </script>

        {{-- <div id="alertSession" role="alert" class="alert alert-success">--}}
        {{-- <svg xmlns="http://www.w3.org/2000/svg" class="stroke-current shrink-0 h-6 w-6" fill="none" viewBox="0 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 12l2 4 4m6 2a9 9 0 11-18 0 9 9 0 0118 0z" /></svg>--}}
        {{-- <span>{{session('status')}}</span>--}}
        {{-- </div>--}}
    @endif

    <a href="/profesores/create" class="btn btn-primary w-full text-3xl"> Añadir Profesor</a>
    <table class="table table-xs table-pin-rows ">

        <tr>
            <th>nombre</th>
            <th>apellidos</th>
            <th>email</th>
            <th>departamento</th>
        </tr>

        @foreach($profesores as $profesor)
            <tr>
                <td>{{ $profesor->nombre }}</td>
                <td>{{ $profesor->apellidos }}</td>
                <td>{{ $profesor->email }}</td>
                <td>{{ $profesor->departamento }}</td>
                <td>
                    <form action="/profesores/{{ $profesor->id }}" method="POST">
                        @csrf
                        @method("DELETE")
                        <button onClick="confirmDelete(event, this)" class="btn" type="submit">
                            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
                                stroke-width="1.5" stroke="currentColor" class="w-6 h-6 text-red-500">
                                <path stroke-linecap="round" stroke-linejoin="round"
                                    d="m14.74 9-.346 9m-4.788 0 9.26 9m9.968-3.21c.342.342.682.682 1.022 1.022-.165 1.18 1.9 6.73a2.25 2.25 0 0 1-2.244 2.077H8.084a2.25 2.25 0 0 1-2.244 2.077L4.772
                                    5.79m14.456 0a4.188 4.188 0 0 0-3.478-.397m-12 .562c.34-.059.68-.114 1.022-.165m0 0a4.188 4.188 0 0 1 3.478-.397m7.5 0v-.916c0-1.18-.91-2.164-2.09-2.281a5.196 5.196 0
                                    0-3.32 0c-1.18-.037-2.09-1.022-2.09-2.281v.916m7.5 0a4.188 4.188 0 0 0-3.478-.397" />
                                </svg>
                            </button>
                        </form>
                    </td>
                </tr>
            @endforeach
        </table>

```

```

<td>
    {{-- Para editar --}}
    <a href="{{route('profesores.edit',$profesor->id, $page->page)}}" class="btn">
        <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5"
            stroke="currentColor" class="w-6 h-6 text-blue-700">
            <path stroke-linecap="round" stroke-linejoin="round"
                d="m16.862 4.487 1.687-1.688a1.875 1.875 0 1 1 2.652 2.652l10.582 16.074a4.5 4.5 0 0 1 1.897 1.13L6 181.8-2.685a4.5 4.5 0 0 1 1.13-1.897l8.932-8.932a0 0L19.5 7.125M18
                14v4.75A2.25 2.25 0 0 1 15.75 21H5.25A2.25 2.25 0 0 1 3 18.75V8.25A2.25 2.25 0 0 1 5.25 6H10"/>
            </svg>
        </a>
    </td>
</tr>
</tbody>
</table>
</div>

```

```

<script>
    function confirmDelete(event, button) {
        event.preventDefault();
        Swal.fire({
            title: '¿Estás seguro?',
            text: "No podrás reventir esto!",
            icon: 'warning',
            showCancelButton: true,
            confirmButtonColor: '#3085d6',
            cancelButtonColor: '#d33',
            confirmButtonText: 'Sí, borrarlo!'
        }).then((result) => {
            console.log("Resultado "+result);
            if (result.isConfirmed)
                button.closest('form').submit();
        })
    }
}

```

```

}

// window.onload=funcion()
// {
//     setTimeout(function() {
//         document.getElementById("alertSession").style.display = "none";5000);
//     }
// }
// window.onload=()=>
//     setTimeout(()=>
//         document.getElementById("alertSession").style.display = "none",5000);
</script>

</x-layouts.layout>

```

Publicamos las vistas de vendor en su directorio principal para así, poderlas visualizarlas en view; para ello, debemos ejecutar el comando `php artisan vendor:publish --tag=laravel-pagination`.

Al ejecutar el comando anterior, se nos crea la carpeta vendor localizada dentro de la carpeta views.

Si accedemos a la carpeta vendor, podemos comprobar que se nos ha creado una carpeta llamada pagination, que es como la hemos denominado. Si accedemos a dicha carpeta, podemos comprobar

que se han creado distintos ficheros. Entre los ficheros creados, se encuentran los ficheros llamados `paginacion.blade.php` y `tailwind.blade.php`.

A continuación, muestro como queda redactado el fichero `paginacion.blade.php`:

```
@if ($paginator->hasPages())
<div class="flex flex-row justify-center items-center">
<nav role="navigation" aria-label="Pagination Navigation" class="flex justify-between">
    {{-- Previous Page Link --}}
    @if ($paginator->onFirstPage())
        <span class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default leading-5 rounded-md dark:text-gray-600 dark:bg-gray-800 dark:border-gray-600">
            {!! __('pagination.previous') !!}
        </span>
    @else
        <a href="{{ $paginator->previousPageUrl() }}" rel="prev" class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-700 bg-white border border-gray-300 leading-5 rounded-md hover:text-gray-500 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-700 transition ease-in-out duration-150 dark:bg-gray-800 dark:border-gray-600 dark:text-gray-300 dark:focus:border-blue-700 dark:active:bg-gray-700 dark:active:text-gray-300">
            .focus\outline-none:focus {
                outline: 2px solid transparent;
                outline-offset: 2px;
            }
        </a>
    @endif
    {{-- Next Page Link --}}
    @if ($paginator->hasMorePages())
        <a href="{{ $paginator->nextPageUrl() }}" rel="next" class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-700 bg-white border border-gray-300 leading-5 rounded-md hover:text-gray-500 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-700 transition ease-in-out duration-150 dark:bg-gray-800 dark:border-gray-600 dark:text-gray-300 dark:focus:border-blue-700 dark:active:bg-gray-700 dark:active:text-gray-300">
            {!! __('pagination.next') !!}
        </a>
    @else
        <span class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default leading-5 rounded-md dark:text-gray-600 dark:bg-gray-800 dark:border-gray-600">
            {!! __('pagination.next') !!}
        </span>
    @endif
</nav>
</div>
@endif
```

El fichero `tailwind.blade.php`, quedaría de la siguiente manera:

```
@if ($paginator->hasPages())
<div class="flex flex-row justify-center items-center">
    <nav role="navigation" aria-label="{{ __('Pagination Navigation') }}" class="flex items-center justify-between">
        <div class="flex justify-between flex-1 sm:hidden">
            @if ($paginator->onFirstPage())
                <span class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default leading-5 rounded-md dark:text-gray-600 dark:bg-gray-800 dark:border-gray-600">
                    {!! __('pagination.previous') !!}
                </span>
            @else
                <a href="{{ $paginator->previousPageUrl() }}" class="relative inline-flex items-center px-4 py-2 text-sm font-medium text-gray-700 bg-white border border-gray-300 leading-5 rounded-md hover:text-gray-500 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-700 transition ease-in-out duration-150 dark:bg-gray-800 dark:border-gray-600 dark:text-gray-300 dark:focus:border-blue-700 dark:active:bg-gray-700 dark:active:text-gray-300">
                    {!! __('pagination.previous') !!}
                </a>
            @endif
        </div>
        @if ($paginator->hasMorePages())
            <a href="{{ $paginator->nextPageUrl() }}" class="relative inline-flex items-center px-4 py-2 ml-3 text-sm font-medium text-gray-700 bg-white border border-gray-300 leading-5 rounded-md hover:text-gray-500 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-700 transition ease-in-out duration-150 dark:bg-gray-800 dark:border-gray-600 dark:text-gray-300 dark:focus:border-blue-700 dark:active:bg-gray-700 dark:active:text-gray-300">
                {!! __('pagination.next') !!}
            </a>
        @else
            <span class="relative inline-flex items-center px-4 py-2 ml-3 text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default leading-5 rounded-md dark:text-gray-600 dark:bg-gray-800 dark:border-gray-600">
                {!! __('pagination.next') !!}
            </span>
        @endif
    </div>
</div>
```

```
<div class="hidden sm:flex sm:items-center sm:justify-between">  
  <div>  
    <p class="text-sm text-gray-700 leading-5 dark:text-gray-600">  
      {!! __('Showing') !!}  
      @if ($paginator->firstItem())  
        <span class="font-medium">{{ $paginator->firstItem() }}</span>  
        {!! __('to') !!}  
        <span class="font-medium">{{ $paginator->lastItem() }}</span>  
      @else  
        {{ $paginator->count() }}  
      @endif  
      {!! __('of') !!}  
      <span class="font-medium">{{ $paginator->total() }}</span>  
      {!! __('results') !!}  
    </p>  
  </div>  
  
  <div>  
    <span class="relative z-0 inline-flex rtl:flex-row-reverse shadow-sm rounded-md">  
      {{{ Previous Page Link }}}  
      @if ($paginator->onFirstPage())  
        <span aria-disabled="true" aria-label="{{ __('pagination.previous') }}">  
          <span  
            class="relative inline-flex items-center px-2 py-2 text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default rounded-l-md leading-5  
              dark:bg-gray-800 dark:border-gray-600  
              aria-hidden="true">  
                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20">  
                  <path fill-rule="evenodd"  
                    d="M12.787 5.293a1 0 0 0 1.414 0l3.293 3.293a1 0 0 1 -1.414 1.414l-4-4a1 0 0 0 -1.414 1.414l-4 4a1 0 0 1 .414 1.414z"  
                    clip-rule="evenodd"/>  
                </svg>  
              </span>  
            </span>  
          </span>  
        </div>
```

```

@else
<a href="{{ $paginator->previousPageUrl() }}" rel="prev"
class="relative inline-flex items-center px-2 py-2 text-sm font-medium text-gray-500 bg-white border border-gray-300 rounded-l-md leading-5 hover:text-gray-400
focus:z-10 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-500 transition ease-in-out duration-150
dark:bg-gray-800 dark:border-gray-600 dark:active:bg-gray-700 dark:focus:border-blue-800"
aria-label="{{ __('pagination.previous') }}">
<svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20">
<path fill-rule="evenodd"
d="M12.707 5.293a1 0 0 0 1 1.414L9.414 10l3.293 3.293a1 0 0 1 1.414 1.414l-4 4a1 0 0 0 1.414 1.414l4 4a1 0 0 1 1.414 1.414 0z"
clip-rule="evenodd"/>
</svg>
</a>
@endif

{{-- Pagination Elements --}}
@foreach ($elements as $element)
    {{{-- "Three Dots" Separator --}}
    @if (is_string($element))
        <span aria-disabled="true">
            <span
                class="relative inline-flex items-center px-4 py-2 -ml-px text-sm font-medium text-gray-700 bg-white border border-gray-300 cursor-default leading-5
                dark:bg-gray-800 dark:border-gray-600">{{ $element }}</span>
            </span>
        @endif

    {{{-- Array Of Links --}}
    @if (is_array($element))
        @foreach ($element as $page => $url)
            @if ($page == $paginator->currentPage())
                <span aria-current="page">
                    <span
                        class="relative inline-flex items-center px-4 py-2 -ml-px text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default
                        leading-5 dark:bg-gray-800 dark:border-gray-600">{{ $page }}</span>
                    </span>
            @endif
        @endforeach
    @endif
    }

```

```

@else
<a href="{{ $url }}"
class="relative inline-flex items-center px-4 py-2 -ml-px text-sm font-medium text-gray-700 bg-white border border-gray-300 leading-5
hover:text-gray-500 focus:z-10 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-700 transition
ease-in-out duration-150 dark:bg-gray-800 dark:border-gray-600 dark:text-gray-400 dark:hover:text-gray-300 dark:active:bg-gray-700
dark:focus:border-blue-800"
aria-label="{{ __('Go to page :page', ['page' => $page]) }}">
{{ $page }}
</a>
@endif
@endforeach
@endif
@endforeach
{{-- Next Page Link --}}
@if ($paginator->hasMorePages())
<a href="{{ $paginator->nextPageUrl() }}" rel="next"
class="relative inline-flex items-center px-2 py-2 -ml-px text-sm font-medium text-gray-500 bg-white border border-gray-300 rounded-r-md leading-5
hover:text-gray-400 focus:z-10 focus:outline-none focus:ring ring-gray-300 focus:border-blue-300 active:bg-gray-100 active:text-gray-500 transition ease-in-out
duration-150 dark:bg-gray-800 dark:border-gray-600 dark:active:bg-gray-700 dark:focus:border-blue-800"
aria-label="{{ __('pagination.next') }}">
<svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20">
<path fill-rule="evenodd"
d="M7.293 14.707a1 1 0 010-1.414L10.586 10 7.293 6.707a1 1 0 011.414-1.414L4 4a1 1 0 010-1.414L10 2a1 1 0 011.414 0z"
clip-rule="evenodd"/>
</svg>
</a>
@endif
<span aria-disabled="true" aria-label="{{ __('pagination.next') }}">
<span
class="relative inline-flex items-center px-2 py-2 -ml-px text-sm font-medium text-gray-500 bg-white border border-gray-300 cursor-default rounded-r-md
leading-5 dark:bg-gray-800 dark:border-gray-600"
aria-hidden="true">
<svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20">
<path fill-rule="evenodd"
d="M7.293 14.707a1 1 0 010-1.414L10.586 10 7.293 6.707a1 1 0 011.414-1.414L4 4a1 1 0 010-1.414L10 2a1 1 0 011.414 0z"
clip-rule="evenodd"/>
</svg>
</span>
</span>
@endif
</div>
</div>
</nav>
</div>
</div>
@endif

```

9. Instalamos React en Laravel; para ello, necesitamos 3 elementos:

- **Librería de react** → npm install [react-dom@latest](#)
- **Componente de react para renderizar los componentes** → npm install [react@latest](#)
- **Plugin de vite para que orqueste con react** → npm install --save-dev @vitejs/plugin-react

Después modificamos el fichero vite.config.js

