



Curso 2020/21

Pedro Jesús Sánchez Sánchez

Programación Web

Tutorial Laravel

1 Primeros pasos

Una vez creados nuestro proyecto y nuestra base de datos siguiendo el tutorial sin problemas, procederé a crear mi primer controller, layout y modelo y mi primera vista.

1.1 Primer controller

Para crear el primer controller simplemente he seguido las instrucciones tal y como están en el tutorial, se trata del HolaController, donde tendremos 3 métodos, uno llamado **welcome**, otro llamado **saludo** y otro llamado **saludo2**. A continuación muestro capturas de los archivos **HolaController.php** y **web.php** donde se muestra el código necesario para el funcionamiento del controlador.

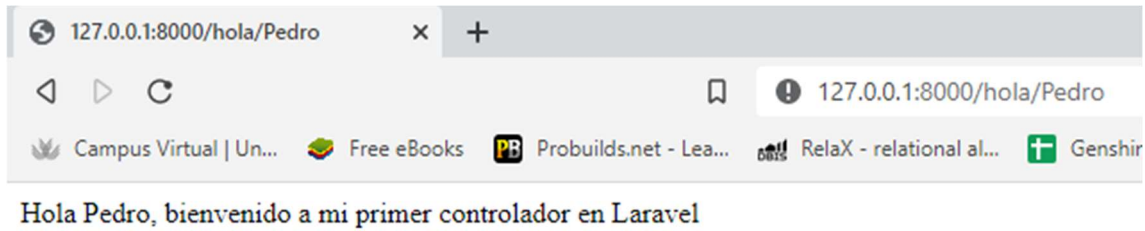
HolaController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class HolaController extends Controller
8  {
9      public function welcome()
10     {
11         return view('welcome');
12     }
13
14     public function saludo($nombre)
15     {
16         return "Hola {$nombre}, bienvenido a mi primer controlador en Laravel";
17     }
18
19     public function saludo2($nombre, $edad)
20     {
21         return "Hola {$nombre}, tienes {$edad} años";
22     }
23 }
```

Web.php

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6   |-----
7   | Web Routes
8   |-----
9   |
10  | Here is where you can register web routes for your application. These
11  | routes are loaded by the RouteServiceProvider within a group which
12  | contains the "web" middleware group. Now create something great!
13  |
14  */
15
16  Route::get('/', 'HolaController@welcome')->name('welcome');
17
18  Route::get('/hola/{nombre}', 'HolaController@saludo')->name('saludo');
19
20  Route::get('/hola/{nombre}/{edad}', 'HolaController@saludo2')->name('saludo2');
21
22  Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
23      return view('dashboard');
24  })->name('dashboard');
25
```

Y ahora, mostraré como al acceder a localhost:8000/hola/Pedro se muestra un saludo con mi nombre para así demostrar el correcto funcionamiento del controlador.



1.2 Primera vista

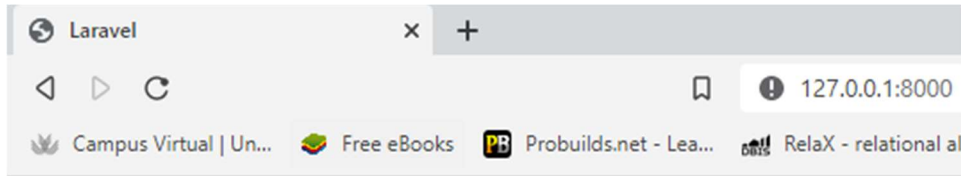
Para realizar nuestra primera vista, continuaremos con el tutorial y sustituiremos el contenido de **welcome.blade.php** por este código html tal y como se nos indica.

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6
7     <title>Laravel</title>
8   </head>
9
10  <body class="antialiased">
11    <h1>Tutorial Laravel 2021</h1>
12    <h2>Programación web</h2>
13    <h1>Bienvenido {{$nombre}}</h1>
14  </body>
15 </html>
```

A continuación, tendremos que especificar dicha variable nombre en nuestro controller, pasaremos una variable desde un controller a una vista con compact como se especifica en el tutorial, de tal manera que nuestro método **welcome** deberá ser modificado de la siguiente manera:

```
9     public function welcome()
10     {
11         $nombre = "Pedro";
12         return view('welcome', compact('nombre'));
13     }
```

Y como vemos a continuación, ya tendríamos acabada nuestra vista.



Tutorial Laravel 2021

Programación web

Bienvenido Pedro

1.3 Primer layout

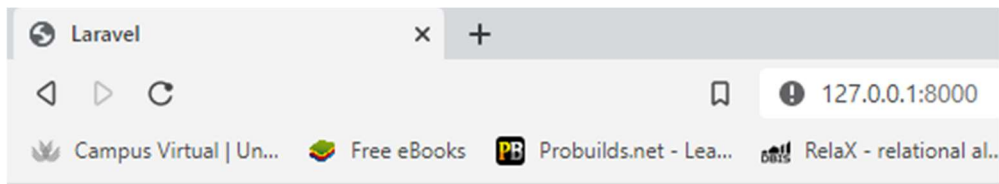
Para este primer layout también me ha servido a la perfección lo que especificaba el tutorial. Mostraré el archivo **header.blade.php**.

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3     <head>
4         <meta charset="utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6
7         <title>Laravel</title>
8     </head>
9
10    <body class="antialiased">
11        <h1>Título con layouts Laravel 2021</h1>
12        @yield('contenido')
13    </body>
14 </html>
```

Y ahora mostrare el **welcome.blade.php** utilizando el layout anterior.

```
1 @extends('layouts.header')
2 @section('contenido')
3     <h2>Programacion web</h2>
4     @if (is_null ($nombre))
5         <h1>Hola persona anónima</h1>
6     @else
7         <h1>Bienvenido {{$nombre}}</h1>
8     @endif
9 @endsection
```

En nuestro navegador, se verá de la siguiente manera:



Título con layouts Laravel 2021

Programacion web

Bienvenido Pedro

1.4 Primer modelo

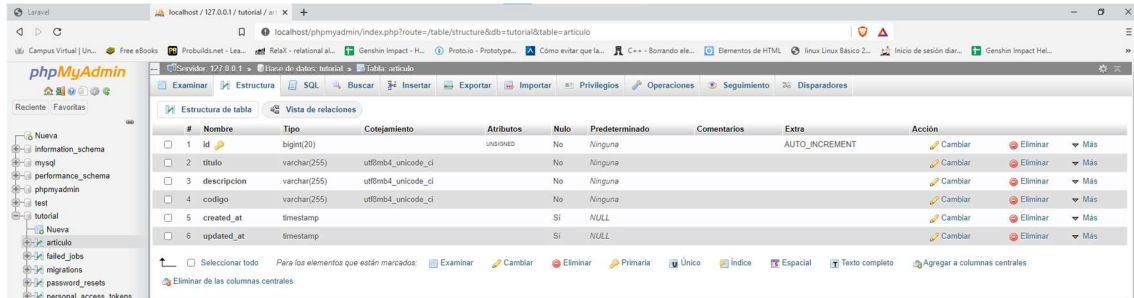
En nuestro primer modelo, también podremos seguir el tutorial y realizarlo satisfactoriamente sin ningún problema, muestro a continuación el archivo **Articulo.php**.

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Articulo extends Model
9  {
10     use HasFactory;
11     protected $table = 'articulo';
12 }
```

Ahora mostrare las modificaciones necesarias en el archivo necesario para hacer la migración:

```
public function up()
{
    Schema::create('Articulo', function (Blueprint $table) {
        $table->id();
        $table->string('titulo');
        $table->string('descripcion');
        $table->string('codigo');
        $table->timestamps();
    });
}
```

Y por último una captura en la que se ve como se ha migrado correctamente a la base de datos:



The screenshot shows the phpMyAdmin interface with the 'articulos' table selected. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	bigint(20)		unsigned	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	titulo	varchar(255)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
3	descripcion	varchar(255)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
4	codigo	varchar(255)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
5	created_at	timestamp			Si	NULL			Cambiar Eliminar Más
6	updated_at	timestamp			Si	NULL			Cambiar Eliminar Más

2 Primer prototipo web artículos

En este punto no he tenido ninguna clase de problema siguiendo el tutorial, primero creamos el controlador **ArticuloController.php** y lo rellenamos de la siguiente manera:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ArticuloController extends Controller
8  {
9      public function index()
10     {
11         return view('articulos.articulos');
12     }
13 }
```

Ahora el tutorial nos sugiere crear un layout pero yo ya lo tengo creado del punto primer layout así que simplemente utilizo ese layout ya que es idéntico al que tenemos que crear.

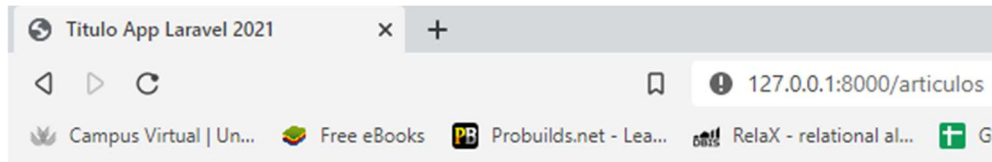
A continuación, crearemos la vista **articulos.blade.php** y la rellenaremos con el siguiente código:

```
1  @extends('layouts.header')
2  @section('contenido')
3  <h2>Bienvenido a la web sobre artículos</h2>
4  <a href="{{route('insertar_articulo')}}" title="Añadir artículo">Añadir artículo</a>
5  <p>Estos son los artículos publicados.</p>
6  @endsection
```

Registramos la ruta en **routes/web.php** (también he registrado la ruta de insertar_articulo aunque aún no este implementado para ahorrar trabajo posteriormente).

```
20 Route::get('/articulos', 'ArticuloController@index')->name('articulos');
21 Route::get('/articulos/nuevo', 'ArticuloController@nuevo')->name('insertar_articulo');
```

Y con eso, ya lo tenemos todo. El resultado en mi navegador es el siguiente:



App Laravel 2021

Bienvenido a la web sobre artículos

[Añadir artículo](#)

Estos son los articulos publicados.

3 Crear formulario para introducir artículos

En este apartado he tenido un problema a la hora de seguirlo en el tutorial y es que el código de la vista **nuevo.blade.php** tiene un pequeño fallo ya que en el tutorial pone en el código **@extends('layouts.app')** y en realidad es **@extends('layouts.header')**. Además a la hora de actualizar esta misma vista para que el formulario reporte los fallos tampoco viene en el tutorial como hacerlo. Para realizarlo, primero añadimos el siguiente método en **ArticuloController.php**.

```
public function nuevo()
{
    return view('articulos.nuevo');
}
```


Ahora creamos la vista **nuevo.blade.php** pero teniendo en cuenta la diferencia con el tutorial que he citado antes.

```
1 @extends('layouts.header')
2 @section('contenido')
3 <h2>Insertar artículo</h2>
4 <a href="{{route('articulos')}}" title="Artículos">Artículos</a>
5 <form action="{{route('guardar_articulo')}}" method="POST">
6 <label for="titulo">Título:</label><br>
7 <input type="text" name="titulo" class="input" value=""><br><br>
8 <label for="descripcion">Descripción:</label><br>
9 <textarea name="descripcion" rows="5" class="form_control"></textarea>
10 <br><br>
11 <label for="cuerpo">Cuerpo:</label><br>
12 <textarea name="cuerpo" rows="8" class="form_control"></textarea>
13 <br><br>
14 <button type="submit" class="btn btn-primary">Crear artículo</button>
15 </form>
16 @endsection
```

Y añadimos en **web.php** la siguiente ruta:

```
Route::post('/articulos', 'ArticuloController@guardar')->name('guardar_articulo');
```

Y ya tendríamos nuestro formulario para insertar artículos. Este sería el resultado funcionando en mi navegador.



App Laravel 2021

Insertar artículo

[Artículos](#)

Título:

Descripción:

Cuerpo:

Crear artículo

Ahora primero escribiremos en la consola **php artisan make:request NuevoArticuloRequest** para crear un `form_request` que rellenaremos con el siguiente código:

```
1  <?php
2
3  namespace App\Http\Requests;
4
5  use Illuminate\Foundation\Http\FormRequest;
6
7  class NuevoArticuloRequest extends FormRequest
8  {
9      /**
10       * Determine if the user is authorized to make this request.
11       *
12       * @return bool
13       */
14      public function authorize()
15      {
16          return true;
17      }
18
19      /**
20       * Get the validation rules that apply to the request.
21       *
22       * @return array
23       */
24      public function rules()
25      {
26          return [
27              'titulo' => 'required|max:100',
28              'descripcion' => 'required|max:200',
29              'cuerpo' => 'required'
30          ];
31      }
32  }
```

Modificamos la vista **nuevo.blade.php** de la siguiente manera (este código no viene en el tutorial).

```
1  @extends('layouts.header')
2  @section('contenido')
3      <h2>Insertar artículo</h2>
4      <a href="{{route('articulos')}}" title="Artículos">Artículos</a>
5
6      <form action="{{route('guardar_articulo')}}" method="POST">
7          {{csrf_field()}}
8
9          @if ($errors->any())
10              <div class="alert alert-danger">
11                  <ul>
12                      @foreach ($errors->all() as $error)
13                          <li>{{ $error }}</li>
14                      @endforeach
15                  </ul>
16              </div>
17          @endif
18
19          <label for="titulo">Título:</label><br>
20          <input type="text" name="titulo" class="input" value=""><br><br>
21          <label for="descripcion">Descripción:</label><br>
22          <textarea name="descripcion" rows="5" class="form_control"></textarea>
23          <br><br>
24          <label for="cuerpo">Cuerpo:</label><br>
25          <textarea name="cuerpo" rows="8" class="form_control"></textarea>
26          <br><br>
27          <button type="submit" class="btn btn-primary">Crear artículo</button>
28      </form>
29  @endsection
```

Y, por último, añadiremos el siguiente código a nuestro archivo **ArticuloController.php**.

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Http\Requests\NuevoArticuloRequest;
6  use Illuminate\Http\Request;
7
8  class ArticuloController extends Controller
9  {
10     public function index()
11     {
12         return view('articulos.articulos');
13     }
14
15     public function nuevo()
16     {
17         return view('articulos.nuevo');
18     }
19
20     public function guardar(NuevoArticuloRequest $request){}
21 }
```

Y así es como quedará en nuestro navegador cuando dejemos sin rellenar los campos:

Titulo App Laravel 2021

127.0.0.1:8000/articulos/nuevo

App Laravel 2021

Insertar artículo

[Artículos](#)

- The titulo field is required.
- The descripcion field is required.
- The cuerpo field is required.

Titulo:

Descripción:

Cuerpo:

Crear artículo

4 Acceso a la base de datos

Para crear la tabla además del código sql que venía en el tutorial para el create table me ha hecho añadir en la línea anterior a todo ese código **SET sql_mode = ''**. Además, a la hora de modificar el modelo **articulo.php**, en el tutorial especifica la tabla articulo, sin embargo, la tabla se llama artículos. Aquí se puede ver el código de dicho modelo.

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Articulo extends Model
9  {
10     use HasFactory;
11     protected $table = 'articulos';
12
13     protected $fillable = ['titulo', 'descripcion', 'cuerpo'];
14 }
```

Ahora, modificaremos el método index del controlador **ArticuloController.php**. En esto, me fallaba que no encontraba el modelo dado que en el tutorial no se especifica añadir la línea use App\Models\Articulo.

```
7  use App\Models\Articulo;
8
9  class ArticuloController extends Controller
10 {
11     public function index()
12     {
13         $articulos = Articulo::all();
14         return view('articulos.articulos')->with(['articulos' => $articulos]);
15     }
16 }
```

Modificamos la vista **articulos.blade.php**.

```
1  @extends('layouts.header')
2  @section('contenido')
3      <h2>Bienvenido a la web sobre artículos</h2>
4      <a href="{{route('insertar_articulo')}}" title="Añadir artículo">Añadir artículo</a>
5      <p>Estos son los artículos publicados.</p>
6
7      @foreach($articulos as $fila)
8          <a href="{{route('un_articulo', ['art'=>$fila->id])}}" title="Ver artículo">
9              <h2>{{ $fila->titulo }}</h2></a>
10             <p>{{ $fila->descripcion }}</p>
11             <br>
12     @endforeach
13 @endsection
```

Creamos una ruta para un artículo en específico:

```
22 Route::get('/articulos/{art}', 'ArticuloController@mostrar_articulo')->name('un_articulo');
```

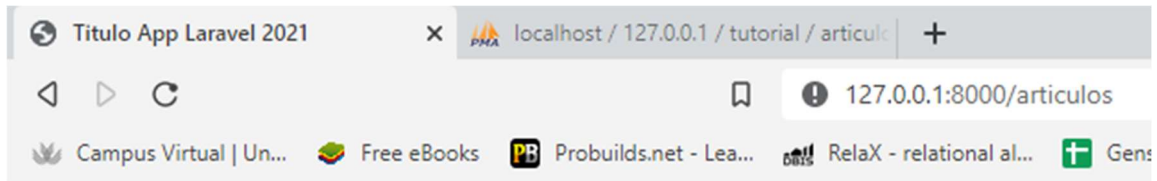
Modificamos el controlador ArticuloController.php para añadir el método mostrar_articulo:

```
public function mostrar_articulo(Articulo $art)
{
    return view('articulos.mostrar_articulo')->with(['articulo' => $art]);
}
```

Creamos la vista **mostrar_articulo.blade.php** con el siguiente código:

```
1 @extends('layouts.header')
2 @section('contenido')
3     <a href="{{route('articulos')}}" title="Ver todos los articulos">Ver todos los articulos</a>
4     <p>Este es el articulo.</p>
5
6     <h2>{{$articulo->titulo}}</h2>
7     <h3>{{$articulo->descripcion}}</h3>
8     <p>{{$articulo->cuerpo}}</p>
9 @endsection
```

Y aquí se puede ver funcionando, tanto una lista con todos los artículos, como la página de visualización de un artículo:



App Laravel 2021

Bienvenido a la web sobre artículos

[Añadir artículo](#)

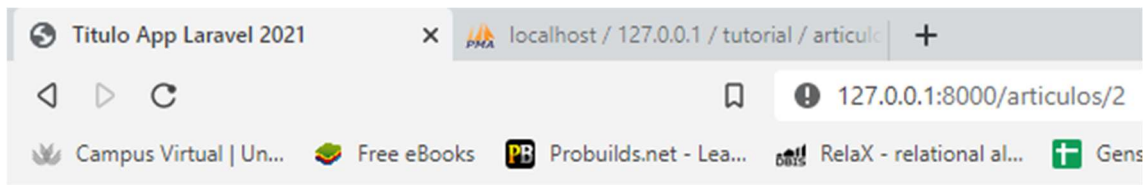
Estos son los articulos publicados.

[hola](#)

prueba

[hola 2](#)

prueba 2



App Laravel 2021

[Ver todos los articulos](#)

Este es el articulo.

hola 2

prueba 2

Esto es una prueba 2

5 Insertar artículos en la base de datos desde el formulario

No he tenido ningún problema con este apartado, aunque hay un par de erratas en el código del tutorial dado que el modelo se llama Artículo y no Articulos. Tendremos que modificar el método guardar de nuestro controlador.

```
22     public function guardar(NuevoArticuloRequest $request)
23     {
24         $art = Artículo::create($request->only('titulo', 'descripcion', 'cuerpo'));
25         return redirect()->route('un_articulo',['art'=>$art->id]);
26     }
```

6 Paginación de los artículos

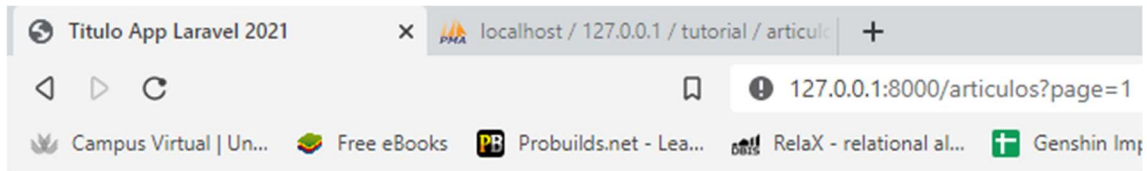
En este apartado tampoco he tenido ningún problema a la hora de seguir el tutorial y hacer que funcione. Modificamos el método index de nuestro controlador.

```
public function index()
{
    $articulos = Artículo::orderBy('id', 'desc')->paginate(2);
    return view('articulos.articulos')->with(['articulos' => $articulos]);
}
```

Añadimos a nuestra vista **articulos.blade.php** el método `render` y nos quedaría así:

```
1 @extends('layouts.header')
2 @section('contenido')
3     <h2>Bienvenido a la web sobre artículos</h2>
4     <a href="{{route('insertar_articulo')}}" title="Añadir artículo">Añadir artículo</a>
5     <p>Estos son los artículos publicados.</p>
6
7     @foreach($articulos as $fila)
8         <a href="{{route('un_articulo', ['art'=>$fila->id])}}" title="Ver artículo">
9             <h2>{{ $fila->titulo }}</h2></a>
10        <p>{{ $fila->descripcion }}</p>
11        <br>
12    @endforeach
13
14    {{ $articulos->render() }}
15 @endsection
```

Y si tenemos más de dos artículos así es como se ve en nuestro navegador:



App Laravel 2021

Bienvenido a la web sobre artículos

[Añadir artículo](#)

Estos son los artículos publicados.

[hola 3](#)

prueba 3

[hola 2](#)

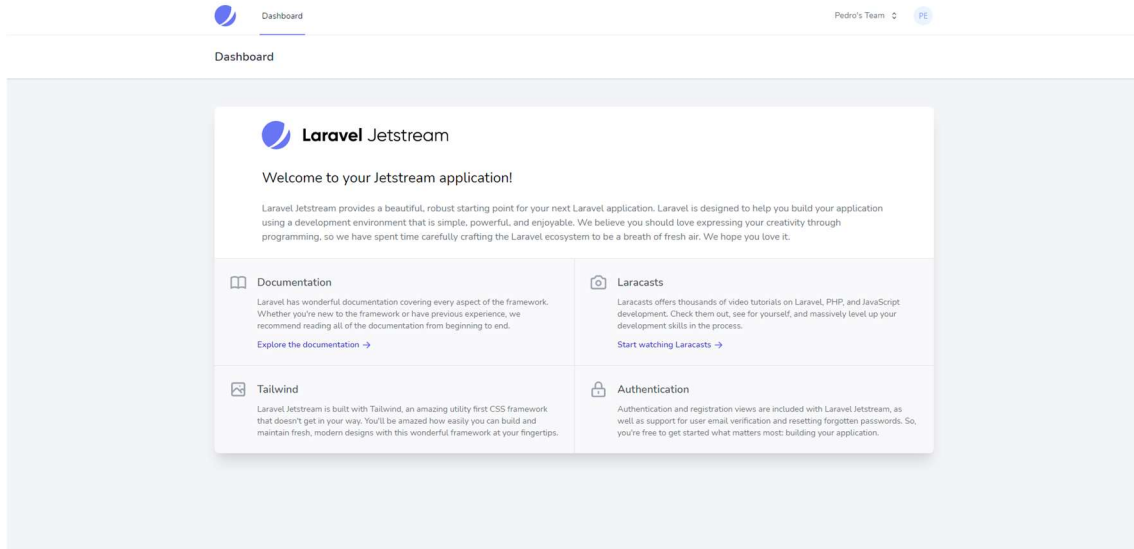
prueba 2

« [Previous](#) [Next](#) »

Showing 1 to 2 of 3 results

7 Login y registro usuario

No he tenido ningún problema para registrarme y acceder. Aquí lo muestro en una captura:



8 Integración de los formularios en la web

Tampoco he tenido ningún problema en este apartado. Modifico los archivos `register.blade.php` y `login.blade.php` tal y como se indica en el tutorial:

login.blade.php

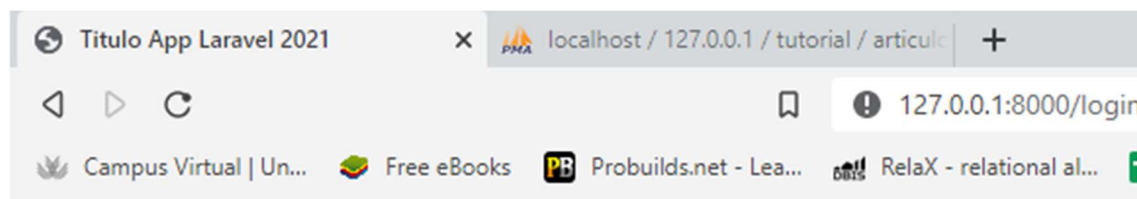
```
1 @extends('layouts.header')
2 @section('contenido')
3     <x-jet-authentication-card>
4         <x-slot name="logo">
5             </x-slot>
6
7         <x-jet-validation-errors class="mb-4" />
8
9         @if (session('status'))
10             <div class="mb-4 font-medium text-sm text-green-600">
11                 {{ session('status') }}
12             </div>
13         @endif
14
15         <form method="POST" action="{{ route('login') }}">
16             @csrf
17
18             <div>
19                 <x-jet-label for="email" value="{{ __('Email') }}" />
20                 <x-jet-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')" required autofocus />
21             </div>
22
23             <div class="mt-4">
24                 <x-jet-label for="password" value="{{ __('Password') }}" />
25                 <x-jet-input id="password" class="block mt-1 w-full" type="password" name="password" required autocomplete="current-password" />
26             </div>
27
28             <div class="block mt-4">
29                 <label for="remember_me" class="flex items-center">
30                     <x-jet-checkbox id="remember_me" name="remember" />
31                     <span class="ml-2 text-sm text-gray-600">{{ __('Remember me') }}</span>
32                 </label>
33             </div>
34
35             <div class="flex items-center justify-end mt-4">
36                 @if (Route::has('password.request'))
37                     <a class="underline text-sm text-gray-600 hover:text-gray-900 href="{{ route('password.request') }}">
38                         {{ __('Forgot your password?') }}
39                     </a>
40                 @endif
41
42                 <x-jet-button class="ml-4">
43                     {{ __('Log in') }}
44                 </x-jet-button>
45
46                 <a href="{{ route('register') }}" class="btn btn-primary">Registrarse</a>
47             </div>
48         </form>
49     </x-jet-authentication-card>
50 @endsection
```


register.blade.php

```
1 @extends('layouts.header')
2 @section('contenido')
3     <x-jet-authentication-card>
4         <x-slot name="logo">
5             </x-slot>
6
7         <x-jet-validation-errors class="mb-4" />
8
9         <form method="POST" action="{{ route('register') }}">
10             @csrf
11
12             <div>
13                 <x-jet-label for="name" value="{{ __('Name') }}" />
14                 <x-jet-input id="name" class="block mt-1 w-full" type="text" name="name" :value="old('name')" required autofocus autocomplete="name" />
15             </div>
16
17             <div class="mt-4">
18                 <x-jet-label for="email" value="{{ __('Email') }}" />
19                 <x-jet-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')" required />
20             </div>
21
22             <div class="mt-4">
23                 <x-jet-label for="password" value="{{ __('Password') }}" />
24                 <x-jet-input id="password" class="block mt-1 w-full" type="password" name="password" required autocomplete="new-password" />
25             </div>
26
27             <div class="mt-4">
28                 <x-jet-label for="password_confirmation" value="{{ __('Confirm Password') }}" />
29                 <x-jet-input id="password_confirmation" class="block mt-1 w-full" type="password" name="password_confirmation" required autocomplete="new-password" />
30             </div>
31
32             @if (Laravel\Jetstream\Jetstream::hasTermsAndPrivacyPolicyFeature())
33                 <div class="mt-4">
34                     <x-jet-label for="terms">
35                         <div class="flex items-center">
36                             <x-jet-checkbox name="terms" id="terms">
37
38                                 <div class="ml-2">
39                                     {!! __('I agree to the :terms_of_service and :privacy_policy', [
40                                         'terms_of_service' => '<a target="_blank" href="'.route('terms.show').'" class="underline text-sm text-gray-600 hover:text-gray-900">'.__('Terms
41                                         'privacy_policy' => '<a target="_blank" href="'.route('policy.show').'" class="underline text-sm text-gray-600 hover:text-gray-900">'.__('Privacy
42                                     ]) !!}
43                                 </div>
44                             </div>
45                         </div>
46                     </div>
47                 @endif
48
49                 <div class="flex items-center justify-end mt-4">
50                     <a class="underline text-sm text-gray-600 hover:text-gray-900 href="{{ route('login') }}">
51                         {{ __('Already registered?') }}
52                     </a>
53
54                     <x-jet-button class="ml-4">
55                         {{ __('Register') }}
56                     </x-jet-button>
57                 </div>
58             </form>
59         </x-jet-authentication-card>
60 @endsection
```

Y en mi navegador quedarían de la siguiente manera:

login.blade.php



App Laravel 2021

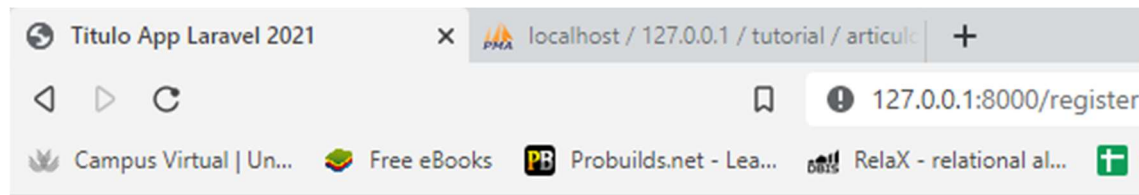
Email

Password

☐ Remember me

[Forgot your password?](#) [Registrarse](#)

register.blade.php



App Laravel 2021

Name

Email

Password

Confirm Password

☐

I agree to the [Terms of Service](#) and [Privacy Policy](#)

[Already registered?](#)

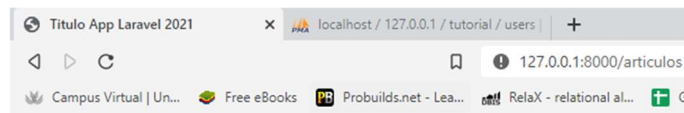
Ahora modificamos el archivo RouteServiceProvider, para entrar en la página /artículos una vez nos logeados o registrados.

```
class RouteServiceProvider extends ServiceProvider
{
    /**
     * The path to the "home" route for your application.
     *
     * This is used by Laravel authentication to redirect users after login.
     *
     * @var string
     */
    public const HOME = '/articulos';
}
```

A continuación, modificamos la vista **articulos.blade.php** con el siguiente código para añadir un botón con nuestro usuario y otro para cerrar sesión:

```
1 @extends('layouts.header')
2 @section('contenido')
3     <h2>Bienvenido a la web sobre artículos</h2>
4     <a href="{{route('insertar_articulo')}}" title="Añadir artículo">Añadir artículo</a>
5     <p>Estos son los articulos publicados.</p>
6
7     @if(Auth::check())
8         <button type="button" class="btn btn-primary">{{Auth::user()->name}}</button>
9         <form action="{{route('logout')}}" method="POST">
10             @csrf
11
12             <x-jet-dropdown-link href="{{route('logout')}}"
13                 onclick="event.preventDefault();
14                     this.closest('form').submit();">
15                 
16
17             </x-jet-dropdown-link>
18         </form>
19     @endif
20
21     @foreach($articulos as $fila)
22         <a href="{{route('un_articulo', ['art'=>$fila->id])}}" title="Ver artículo">
23             <h2>{{ $fila->titulo }}</h2></a>
24             <p>{{ $fila->descripcion }}</p>
25             <br>
26         @endforeach
27
28     {{ $articulos->render() }}
29 @endsection
```

Y una vez modificado, al iniciar sesión aparece así la página /artículos en mi navegador:



App Laravel 2021

Bienvenido a la web sobre artículos

[Añadir artículo](#)

Estos son los articulos publicados.

Pedro


[hola 3](#)

prueba 3

[hola 2](#)

prueba 2

« Previous [Next](#) »

Showing 1 to 2 of 3 results

9 Bootstrap

Y así es como se vería la página anterior aplicando Bootstrap a nuestro proyecto junto a la línea que he tenido que añadir a la cabecera para incluirlo.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
```

