Carpeta de las vistas en plural

LIVEWIREE

1 el escrip debe estar después de livewire en app

        @livewireScripts

      @stack('scripts')

2.- cuando guardo sin Storage

$image = $this->image->store('brands', 'public');

brands/cocacola.jpg

Le debo llamar con Storage

{{ Storage::url($brandd->image) }}

3.- cuando guardo con Storage

$image = $this->image->store('brands', 'public');

      $urlimage = Storage::url($image);

/storage/brands/cocacola.jpg

Le debo llamar sin Storage

{{ url($brandd->image) }}

4.puede servir para buscar tallas y colers en el array

[contains()](https://www.oulub.com/docs/laravel/es-es/collections#method-contains) en el manual de laravel

#### 5. sirve para quitar las claves que no quieres mostrar [except()](https://www.oulub.com/docs/laravel/es-es/collections#method-except)

6.- flatten aplana

7.flip invierte la clave valor

8.- keys devuelve todas las claves de la colección

9.pluck, filterexcept,take,partition,unique

10. al pasar parámetro de un componente a otro tener en cuenta en el modelo

Public function getRouteKeyName()

    {

        return 'slug';

    }

11 en route la variable no se pasa con Compac

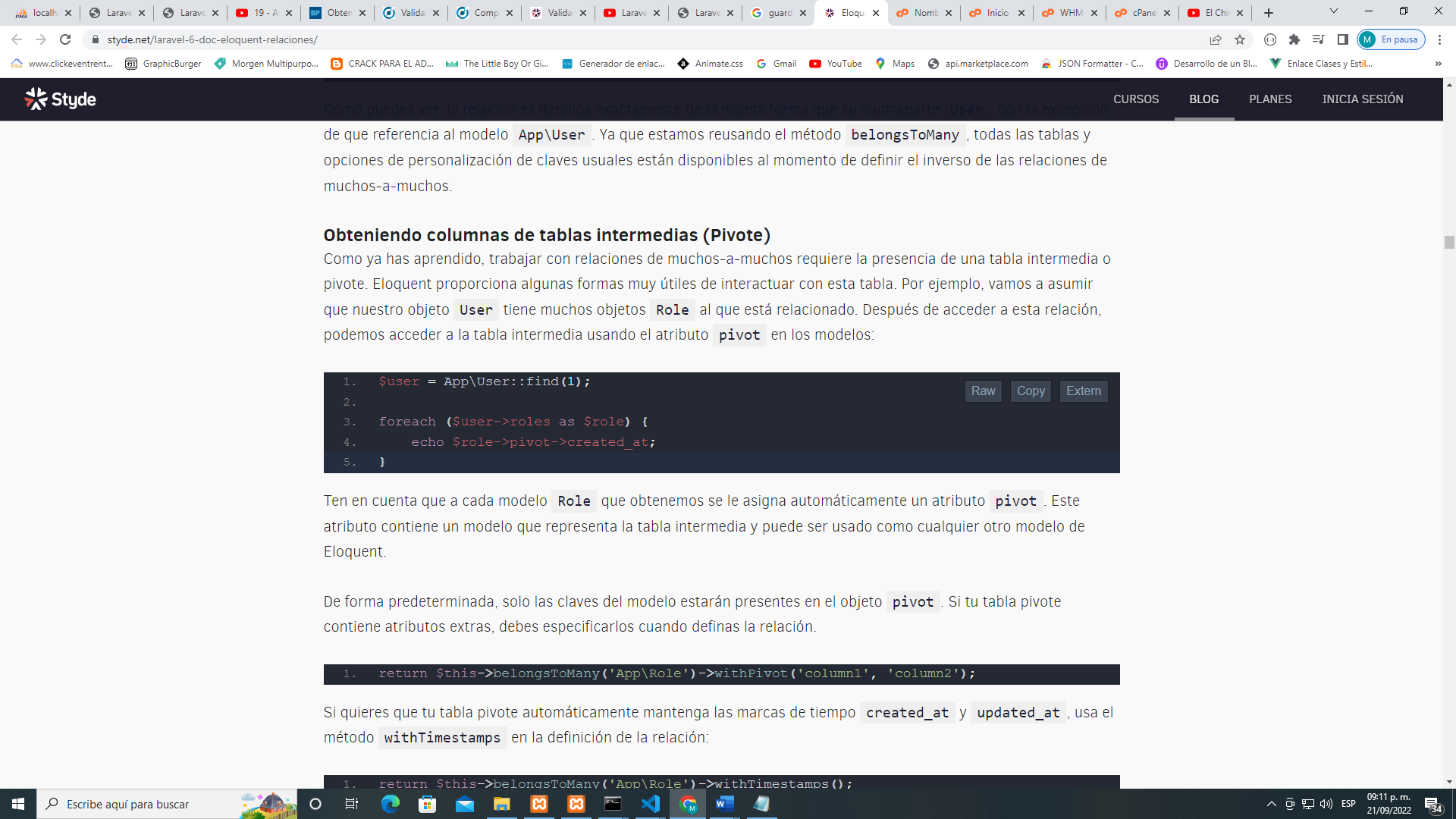
12. poner get en el where para que se convierta en colección

13.- usar $category = Category::where('id', $this->category\_id)->first() de lo contrario no es un objeto

14. el wire:ignore no cambia dentro del ignore

15. el load lo active en un div <div wire:init="loadBrands"></div>

16.- Ten en cuenta que a cada modelo Role que obtenemos se le asigna automáticamente un atributo pivot. Este atributo contiene un modelo que representa la tabla intermedia y puede ser usado como cualquier otro modelo de Eloquent.



De forma predeterminada, solo las claves del modelo estarán presentes en el objeto pivot. Si tu tabla pivote contiene atributos extras, debes especificarlos cuando definas la relación.

return $this-**>**belongsToMany**(**'App\Role'**)**-**>**withPivot**(**'column1', 'column2'**)**;

Si quieres que tu tabla pivote automáticamente mantenga las marcas de tiempo created\_at y updated\_at, usa el método withTimestamps en la definición de la relación:

return $this->belongsToMany('App\Role')->withTimestamps();

Personalizando el nombre del atributo pivot

Como se señaló anteriormente, los atributos de la tabla intermedia pueden ser accedidos en modelos usando el atributo pivot. Sin embargo, eres libre de personalizar el nombre de este atributo para que refleje mejor su propósito dentro de tu aplicación.

Por ejemplo, si tu aplicación contiene usuarios que pueden suscribirse a podcasts, probablemente tengas una relación de muchos-a-muchos entre usuarios y podcasts. Si éste es el caso, puedes desear renombrar tu tabla pivote intermedia como subscription en lugar de pivot. Esto se puede hacer usando el método as al momento de definir la relación:

return $this-**>**belongsToMany**(**'App\Podcast'**)**

-**>**as**(**'subscription'**)**

-**>**withTimestamps**()**;

Una vez hecho esto, puedes acceder a los datos de la tabla intermedia usando el nombre personalizado:

$users = User::with**(**'podcasts'**)**-**>**get**()**;

foreach **(**$users->flatMap->podcasts as $podcast**)** **{**

echo $podcast->subscription->created\_at;

**}**

users

roles

user\_role (en orden alfabetico y en singular)

17.

-otra forma es generando manifest.json con npm run build

Se genera la carpeta build y dentro de ella **manifest.json**

-para que funcione vite puse esto en vite.config.js

    server: {

        host: '0.0.0.0'

    },

Pero darle **npm run dev**

18.queryString, para ver parámetros en la url y poder compartir el url

19. al publicar peruawki los enlaces no funcionaban, entonces volvi a publicar lo de public\_htm y nodemodules

20. migraciones en devstagram

php artisan make:migration add\_username\_to\_users\_table

21. **npm run build** para publicar build

22. videos de livewire <https://laravel-livewire.com/screencasts/form-submits>

22. para generar seeder a partir de datos existentes orangehill/iseed

composer require orangehill/iseed

Asi se genera los seeders

php artisan iseed infrastructures

23. cuando dejas espacio guarda con espacio

<option value="{{$category->id}}">{{$category->name}}</option>

Category\_id=1

<option value="{{$category->id}} ">{{$category->name}}</option>

Category\_id=1\_ 1 con espacio

24 para la variable modelo\_id el \_se remmplaza por mayuscula

 public function updatingModeloId(){

25. no ejecutaba por que el script fallaba ScanCode($('#code').val()) específicamente ($('#code').val()

 <input type="text" id="code" class="block w-full bg-gray-100" wire:keydown.enter.prevent="ScanCode($('#code').val())"/>

Entonces reemplace

<input type="text" wire:model="name" class="block w-full bg-gray-100" wire:keydown.enter.prevent="ScanCode({{ $name }})" />

26.- en allinted usamos vite y el java script no funcionaba hasta que instale el cdn de jquery en layout/app

      <script

        src="https://code.jquery.com/jquery-3.6.1.min.js"

        integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="

        crossorigin="anonymous"></script>

27. la ruta de los componentes jetstream están en: **views/vendor/jetstream/componentes**

28.- agregue en el modal las propiedades del tamaño del modal

$maxWidth = [

'sm' => 'sm:max-w-sm', 'md' => 'sm:max-w-md', 'lg' => 'sm:max-w-lg', 'xl' => 'sm:max-w-xl', '2xl' => 'sm:max-w-2xl', **'3xl' => 'sm:max-w-3xl', '4xl' => 'sm:max-w-4xl'**, '5xl' => 'sm:max-w-5xl', '6xl' => 'sm:max-w-6xl', '7xl' => 'sm:max-w-7xl', 'full' => 'sm:max-w-full',

][$maxWidth ?? '2xl'];

29. los parámetros que paso es distinto en estos 2 casos

"$emit('deleteItem', '{{ $item->id }}')" pasa como string

"$emit('deleteItem', {{ $item->id }} )" pasa entero

wire:click="ScanCode( {{$productt->codigo}} )" pasa entero

wire:click="ScanCode('{{$productt->codigo}}')" pasa como string

30 en el .env cuando se desarrolla debe estar la siguiente configuración porque da error si esta en produccion

APP\_ENV=local

APP\_DEBUG=true

31.- al guardar producto salia error en validación diciendo que falta un parámetro y era por

 'name'=> 'required|unique:products',

Faltaba ponerlo la pabla products en unique ose debe ir unique:products'

32.- hay un error cuando cargamos un archivo distinto a imagen, entonces le pusimos en el input accept="image/jpeg,image/png"

<input type="file" wire:model="image1" accept="image/jpeg,image/png" >

Lo recomendable es no mostrar la imagen porque puede salir error si escoge por ejemplo un pdf, o se puese usar dropzone que ya tiene controlado el tipo de imagen que se va cargar

33.- para gravar automáticamente el slug del producto, en el modelo Producto del proyecto hprd se debe poner también el name, de lo contrario te muestra error en name

    public function setNameAttribute($name)

    {

        $this->attributes['name'] = $name;

        $this->attributes['slug'] = str::slug($name);

    }

Y al crer producto se debe poner también el name

$product = new Product();

        $product->name = $this->name;

34. poniendo los productos de hpr no salia en menú superior era porque los estilos intermos estaban direccionados de esta forma(viendo en el inspector el estilo debería estar en una carpeta productos)

style: './components/flotchart/flotchart.css',

lo pusimos asi y funciono (ojo en el index funcionaba bien el problema era en los enlaces como www.hprd.test/producto/ropas)

style: '/components/flotchart/flotchart.css',

35. al crear rutas generaba error

Route::get('productos/{subcategory}', 'App\Http\Controllers\SubcategoryhostingsController@showproduct')->name('sucategoryproducts.show');

Route::get(‘productos/{producto}’, ‘App\Http\Controllers\SubcategoryhostingsController@showproductp’)->name(‘sucategoryproducts.showp’);

En amarillo genera error, entonces puse su carpeta show y funciono

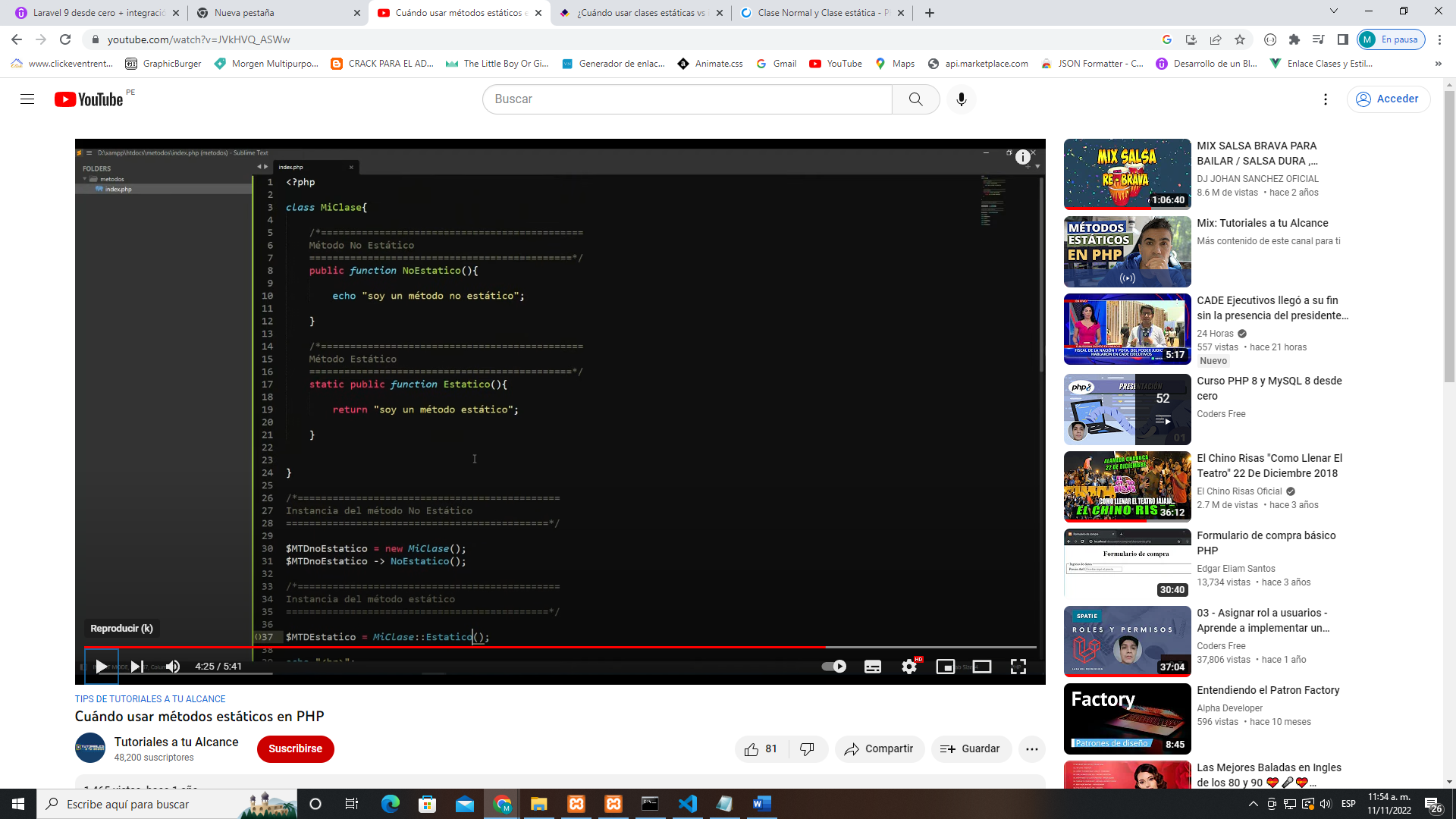
Route::get('productos/show/{product}', 'App\Http\Controllers\SubcategoryhostingsController@showproductp')->name('sucategoryproducts.showp');

36.- La clase estática se usa para una sola instancia de esa clase, mientras que la clase instanciada se usa cuando se requiere más de una instancia.

La clase estática contiene variables estáticas y métodos estáticos, mientras que la clase instanciada contiene variables no estáticas y métodos no estáticos.

Explicación de método estatico y no estatico

<https://www.youtube.com/watch?v=JVkHVQ_ASWw>



el método no estatico te muestra de inmediato

el método estatico se almacena en una variable

37. en la ruta de las imágenes salia ticomperu al inicio y no mostraba al revisar era porque faltaba poner los parámetros correctos en el .env

38.

    public function update(Request $request, Testimony $testimony)

    {

        $this->validate(request(), [

            'name' => 'required'

        ]);

        $testimony->update($request->all());

      if($request->hasFile('image')){

       $url = Storage::disk('s3')->put('testimonies', $request->file('image'), 'public');

            if($testimony->image){

                Storage::disk('s3')->delete($testimony->image);

                $testimony->update([

                    'image' => $url

                ]);

            }else{

                //estaba puesto create pero era cuando hay otra tabla para imagenes

                $testimony->update([

                    'image' => $url

                ]);

            }

        }

        return redirect()->route('testimonies.index')->with('flash', 'Testimonio Actualizado con exito');

    }

39.- php artisan db:seed es para ejecutar los seeders (ojo ponerlo en DataBaseSeeder.php)

php artisan db:seed --class=UserSeeder

40.- para que funcione creación de imágenes en faker

<https://hobbyappgame.com/noticia/faker-no-descarga-las-imagenes-en-laravel/40>

vendor\fakerphp\faker\src\Faker\Provider\Image.php

Una vez abierto este archivo buscaremos en las lineas aprox 114 y agregaremos el siguiente codigo, tomemos como referencia el codigo existente

curl\_setopt($ch, CURLOPT\_FILE, $fp); //línea existente

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);//nueva línea

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);//nueva línea

$success = curl\_exec($ch) && curl\_getinfo($ch, CURLINFO\_HTTP\_CODE) === 200;//línea existente

41.- composer dumpautoload

42.- los accesores son como las propiedades de los modelos a las cuales se puede acceder

 public function get**Stock**Attribute(){

{{ $product->stock}}

43.- **get** devuelve una colección y **first** un objeto

44.- Para ver las traducciones tener en cuenta que

Ticom3/Lang

Ticom3/resources/Lang

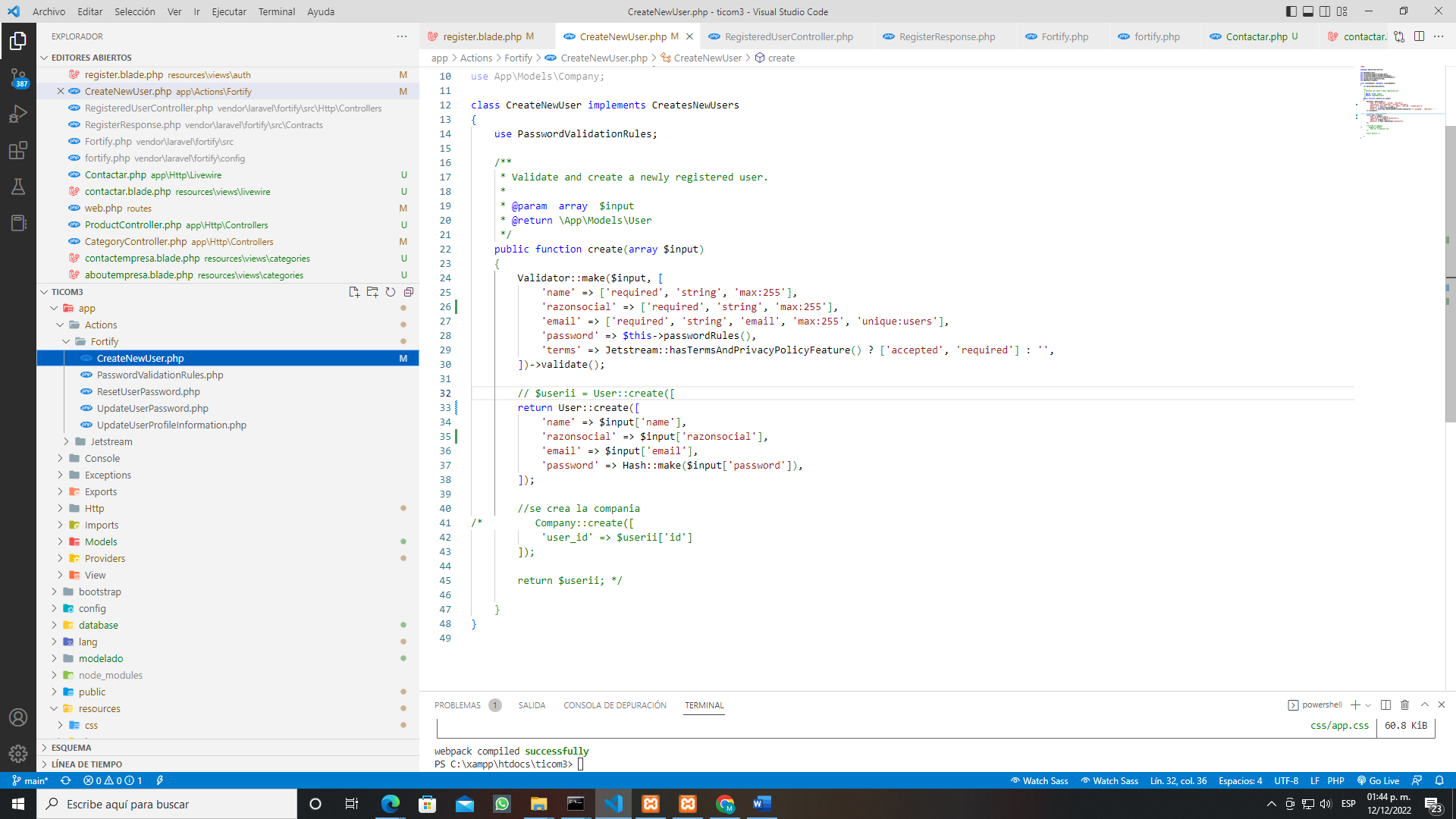
Ambas carpetas deben estar con el mismo contenido

**45. para liberar memoria cache**

php artisan config:cache

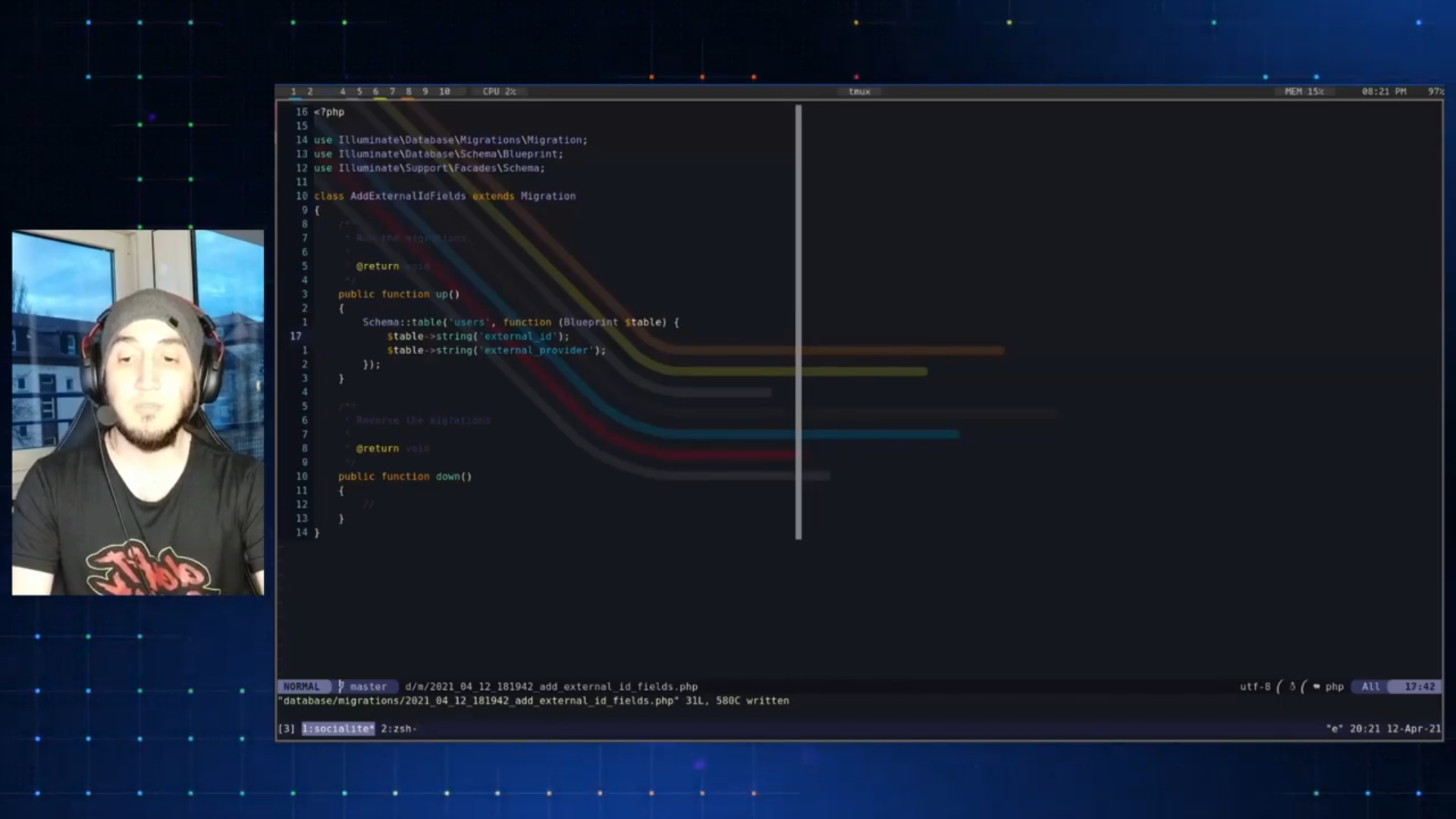
46. para crear un campo más en el registro de usuarios

Ir a **app/actions/fortify**

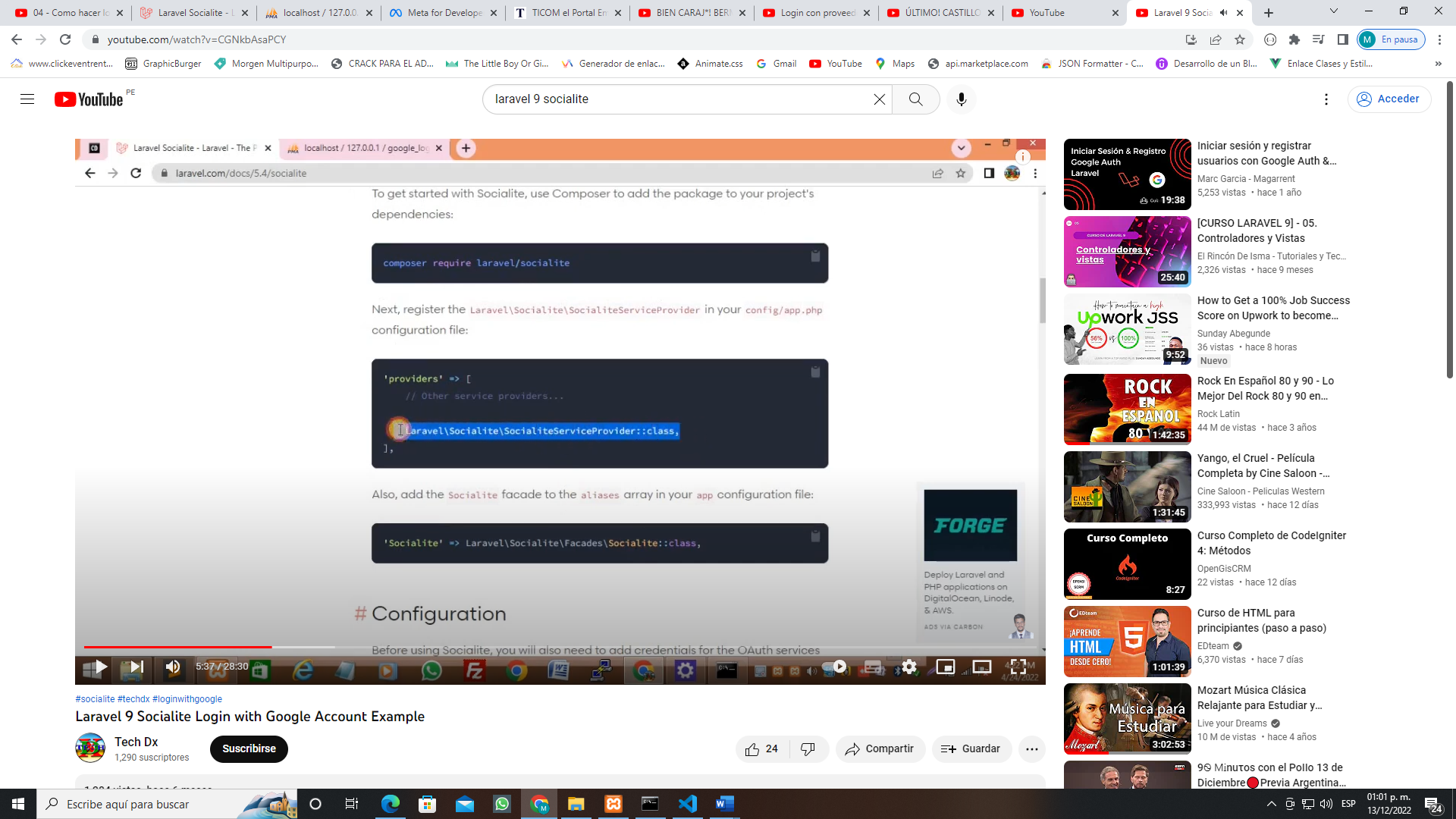


47.- php artisan make:migration add\_external\_id\_fileds

Para agregar un campo en la tabla user



En otro vide agrega en app.php en la opción providers



48. EL slider se trabajo con

https://tailwind-elements.com/docs/standard/components/carousel/#crossfade

https://flowbite.com/docs/components/carousel/#javascript-behaviour

49. para guardar con livewire

   protected $rules = [

        'razonsocial'=> 'required|unique:users,razonsocial',

        'slug'=> 'required',

        'ruc'=> 'required'

        'description'=>'required|min:1',

    ];

Eliminé el slug de la vista

Al guardar generaba error porque no encontraba el input type del slug, porque se le puso en el rules pero no existía en el type

Para guardar no es necesario poner en rules

50 en las validaciones

Tener cuidado cuando se sobre escriben las validaciones

51.- al subir imágenes con laravel intervention generaba error y esto era porque no existía la carpeta storage(storage link)

Pero se creo la ruta directamente sin usar el storage link y funciono perfectamente bien.

52 de los ciclos de vida de livewiere

**updated**

  protected $rules = [

        'title'=> 'required|max:10',

        'content'=>'required|min:10',

        'image'=>'required|image|max:2048',

    ];

    public function updated($propertyNameeer){

        $this->validateOnly($propertyNameeer);

    }

**Updating**

updating() cuando se cambia una de las propiedades

**Updatingfoo**

updatingSearch() cuando se cambia la propiedad search

public function updatingSearch(){

        $this->resetPage();

        //RESETEA la paginación,

}

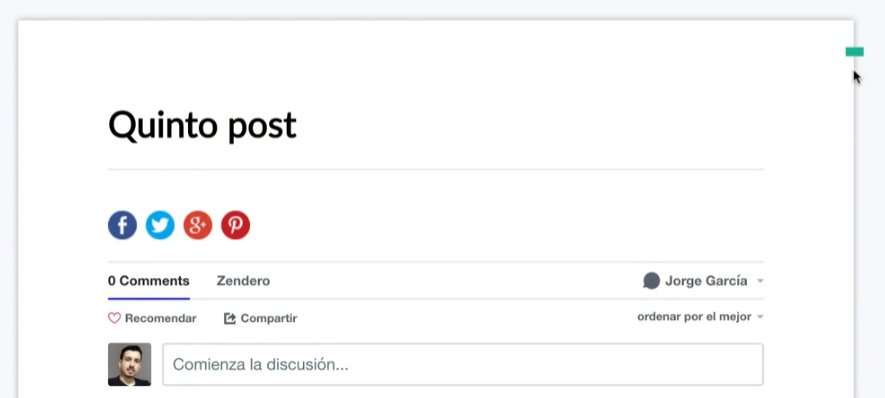
53.- para controlar que no existe las imágenes, fechas o datos se puede usar el helper optional



Con optional quedaría asi



Pero usa if para controlar todo y que no aparezca la franja



Para usar abort



Retornando verdadero o falso de una variable



54.- {{ auth->check()}} para ver si el usuario esta autenticado

55.-para que lea la carpeta public\_html

        'local' => [

            'driver' => 'local',

            //'root' => storage\_path('app'),

            //'root' => public\_path('img'), //es para el local

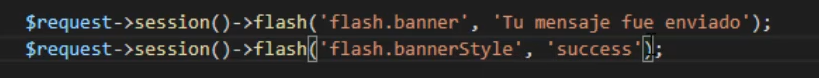
            'root' => base\_path('public\_html/img'),//funciona en el servidor

        ],

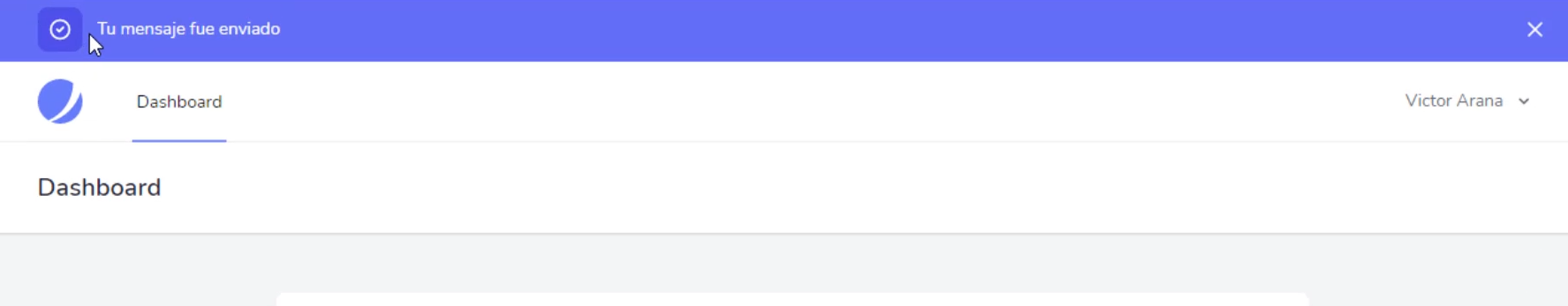
56.- para no permitir registros duplicados ver en CategoriesproductsController ver en el proyecto de megaeditores



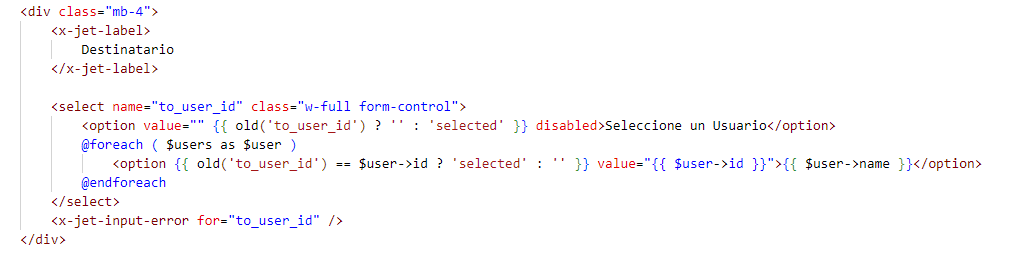
57. mostrando mensajes, x-jet-banner, esta en layout, le pasamos dos mensajes de sesión



Esto sería el mensaje



58. validación en el selec (Notificacion capitulo 13)



59. para llenar una tabla de muchos a muchos en el seeder, tener en cuenta el orden alfabetico

Brand::create([

            'name' => 'adidas',

            'slug'=>'adidas',

            'state' => 1,

        ])->categories()->sync([2,3,4,5]);

**Estuve creando en:**

Category::create([

            'name' => 'vidrios templados',

            'slug' => Str::slug('vidrios templados'),

            'state'=>1,

        ])->brands()->sync([2,7,9]);

Pero el orden de ingreso en la tabla es brand\_id, category\_id al poner varios datos generaba error de integridad

Para ver las rutas de un resource especifico

php artisan r:l - -name=admin.roles

60.- las rutas mostradas una no funciona, la que tiene show no funciona

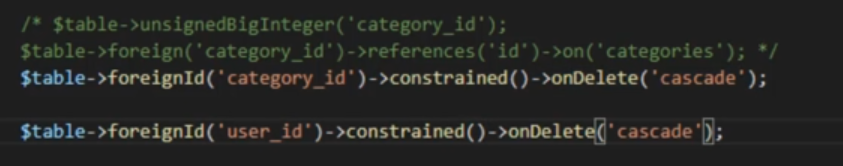
Route::post('producto/{product}', [OrderController::class,'store'])->name('orders');

Route::post('producto/show/{product}', [OrderController::class,'store'])->name('orders');

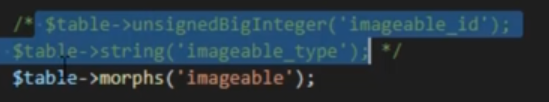
En esta ruta(product.show) estoy mostrando el producto y allí hay un formulario para realizar cotización de este producto, para enviar la cotización debo llamar a la ruta de arriba sombreado de amarillo

Route::get('producto/{product}', [ProductController::class, 'show'])->name('product.show');

61. Nueva forma de poner las relaciones en laravel constrained hace lo mismo



Y para las polimórficas



Instalar una versión especifica de laravel ver en esta URL

https://www.vicolinker.net/laravel-instalar-version-especifica/

Limpiar cache en LARAVEL

php artisan cache:clear

php artisan view:clear

php artisan route:cache

php artisan config:cache

php artisan optimize

php artisan cache:clear

php artisan view:clear

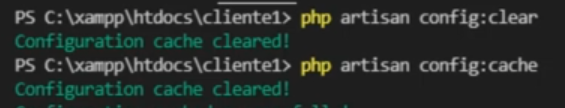
php artisan route:cache  
php artisan config:clear  
php artisan config:cache

php artisan optimize

composer\_memory\_limit=-1 para que use toda la memoria disponible



Pala limpiar cahe de la configuración .env (limpia y luego genera)



62. generador de id en laravel

Ver este video corto

<https://www.youtube.com/watch?v=dY9VYCmu74A>

<https://www.youtube.com/watch?v=1XIMI5kOPuc>

https://www.youtube.com/watch?v=uh-BD0ll1uI

63.- solución de primarykey compuesto de laravel con livewire

Es cierto que Laravel y Livewire no admiten directamente claves primarias compuestas en modelos. Por defecto, Eloquent espera una clave primaria simple llamada "id" en cada modelo. Sin embargo, puedes trabajar con claves primarias compuestas en Laravel siguiendo algunos pasos adicionales.

Aquí hay una forma de trabajar con claves primarias compuestas en Laravel y Livewire:

1. Define una columna adicional en tu tabla llamada "id" y establece esta como la clave primaria. Esta columna será una clave primaria simple y se utilizará para cumplir con los requisitos de Eloquent y Livewire.
2. Mantén tus claves primarias compuestas "local\_id" y "productatribute\_id" en tu tabla. Estas columnas no serán claves primarias desde el punto de vista de Eloquent, pero aún puedes usarlas para establecer relaciones y consultas.
3. Modifica tu modelo **Localproductatribute** de la siguiente manera:



Para eliminar las vistas y rutas de cache use: php artisan clear

php artisan view:clear

php artisan route:clear

**Limpiar cache más completo**

php artisan config:clear

php artisan route:clear

php artisan view:clear

php artisan cache:clear

para crear el carrito de compras en fe, crearemos el componente

php artisan make:livewire ShopingCart

8 de octubre importante

Esto no funciona

Route::get('storage-link',function(){

    if(file\_exists(public\_path('storage'))){

         return 'the "public\_html/storage" directoory already exists';

     }

    app('files')->link(

         storage\_path('app/public\_html'), public\_path('storage')

     );

     return 'the "public\_html/storage" directoory has bee linked';

 });

Pero si funciona este ver pagina de camisas bonaroti en web.php

**Route::get('storage-link',function(){**

**if(file\_exists(public\_path('storage'))){**

**return 'the "public\_html/storage" directoory already exists';**

**}**

**app('files')->link(**

**storage\_path('../storage'), public\_path('storage')**

**);**

**return 'the "public\_html/storage" directoory has bee linked';**

**});**

<?php

$output = shell\_exec('php artisan storage:link');

echo "<pre>$output</pre>";

para que muestre la imagen agregar este código lo puse en views/livwire/admin/edit-poduct.blade .php

'app/public\_html/'.

Quedaría asi:

**<img class="object-cover w-32 h-20" src="{{ Storage::url('app/public\_html/'.$image->url) }}" alt="">**

Interesante de comillas dobles y simples

$id necesita comillas dobles para expandirse

get("http://api.codersfree.test/v1/categories/$id"); funciona

get(http://api.codersfree.test/v1/categories/$id); no funciona

Cuando descargas un proyecto Laravel desde GitHub, generalmente **no incluye** las carpetas vendor/ y node\_modules/ porque están en el archivo .gitignore, lo que significa que no se suben al repositorio.

**¿Qué está pasando?**

1. **Faltan dependencias**: Laravel usa **Composer** para gestionar las dependencias de PHP (que van en vendor/) y **npm** para las de JavaScript (que van en node\_modules/).
2. **Instalación de dependencias**: Cuando ejecutas:
   * composer install → Descarga las dependencias de PHP y crea la carpeta vendor/.
   * npm install → Descarga las dependencias de JavaScript y crea node\_modules/.
3. **El proyecto está listo**: Después de esto, Laravel ya tiene todo lo necesario para ejecutarse correctamente.

Básicamente, composer install y npm install reconstruyen las dependencias necesarias que no vienen en el repositorio.

INERTIA

1. No funcionaba el enlace productos en jetinertia.test, era porque no se puso bien los enlaces en el script

    import {Inertia} from "@inertiajs/inertia"

    import { Head, Link } from '@inertiajs/inertia-vue3'

    import { defineComponent } from 'vue'

    import AppLayout from '@/Layouts/AppLayout.vue'

1. El @ hace que te ubiques en js eso es ta puesto en webpack.config.js

'@': path.resolve('resources/js'),

1. Dd

    public function update(UpdateUserPut $request, User $customer)

    {

        //$request solo controla lo puesto en updateUserPut

        $customer->update($request->validated());

        if(request()->hasFile('avatar')){//request() es global

            $customer->updateProfilePhoto(request()->file("avatar"));

        }

        //request()->session()->flash('message', 'usuario Editado');

        return Redirect::route('user.index', $customer)->with('message', 'usuario Editado');

    }

1. Para manejo de imgenes se pone

@input="form.avatar = $event.target.files[0]"

En reemplazo del v-model

1. En las validaciones