

# Issue 8 del 28 de enero

Temas

Crud=acciones básicas para crear, leer, actualizar y eliminar datos en una base de datos.

-searchbar(crud)

-register

-what is \$fillable and \$guarded

## Searchbar (crud)

Ahora veremos como añadir una barra de búsqueda en nuestra web.

Primero añadiendo el código de vista de la barra de búsqueda pidiendo que recoja el valor en su casilla al dar clic

```
<div class="card">
  <div class="card-header pb-0 border-0">
    <h5 class="">Search</h5>
  </div>
  <div class="card-body">
    <form action="{{ route('dashboard') }}" method="GET">
      <input name="search" placeholder="..." class="form-control w-100" type="text">
      <button class="btn btn-dark mt-2"> Search</button>
    </form>
  </div>
</div>
```

Y añadimos al index que según lo que se busque aparezca filtrado

```
if(request()->has('search')){
    $ideas = $ideas->where('content', 'like', '%'. request()->get('search', ''));
}

return view('dashboard', [
```

# Issue 8 del 28 de enero

## Register

Añadiremos un registro a la web

Comenzamos asignando las rutas una para el registro y otra para posterior al registro

```
Route::get('/register', [AuthController::class, 'register']->name('register'));  
Route::post('/register', [AuthController::class, 'store'] );
```

Su respectivo archivo Blade

```
@extends('layout.layout')  
  
@section('content')  
    <div class="row justify-content-center">  
    </div>  
@endsection  
|
```

Además no nos podemos olvidar de la funcionalidad aquí le decimos a laravel que variables y con que condiciones queremos guardarlas , creamos el usuario lo validamos , guardamos y redireccionamos a la pagina principal

```
public function store()  
{  
    $validated = request()->validate(  
        [  
            'name' => 'required|min:3|max:40',  
            'email' => 'required|email|unique:users,email',  
            'password' => 'required|confirmed|min:8'  
        ]  
    );  
    User::create(  
        [  
            'name' => $validated['name'],  
            'email' => $validated['email'],  
            'password' => Hash::make($validated['password']),  
        ]  
    );  
    return redirect()->route('dashboard')->with('success', 'Account created Successfully!');
```

# Issue 8 del 28 de enero

En Laravel, "\$fillable" y "\$guarded" son propiedades utilizadas dentro de los modelos de Eloquent para controlar qué atributos se

pueden asignar en masa al crear o actualizar datos, donde "\$fillable" define los atributos permitidos (lista blanca) y "\$guarded"

especifica los atributos que no pueden asignarse masivamente (lista negra).

```
protected $guarded = [
    'id',
    'created_at',
    'updated_at',
];

protected $fillable = [
    'content',
    'like'
];

public function comments(){
    return $this->hasMany(Comment::class);
}
```

Un ejemplo de como se aplican son listas que guardan las variables que consideremos guardar en masa como en fillable o ser un poco mas reservado como en guarded.