

Crear migració a la taula tasks . Ha de tenir els camps (id, name, description i completed)

Creemos la migración de la tabla task usando el comando php artisan make:migration create_tasks

```
→ MP09_Practica2 git:(main) × php artisan make:migration create_tasks

INFO Migration [database/migrations/2024_01_22_162151_create_tasks.php] created successfully.

→ MP09_Practica2 git:(main) ×
```

Ahora nos dirigimos al lugar indicado el cual es database/migration y pondremos lo siguiente dentro.

```
2 public function up(): void
3 {
4     Schema::create( table: 'tasks', function (Blueprint $table) {
5         $table->id();
6         $table->string( column: 'name');
7         $table->string( column: 'description');
8         $table->boolean( column: 'completed');
9         $table->timestamps();
10    });
11 }
12
```

Por último lo que haremos sera php artisan migrate:refresh --seed

```
→ MP09_Practica2 git:(main) × php artisan migrate:refresh --seed

INFO Rolling back migrations.

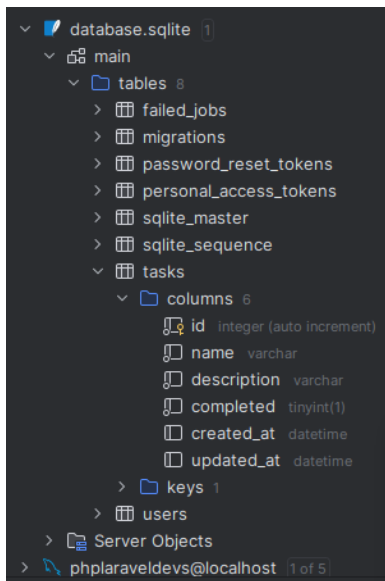
2024_01_22_162151_create_tasks ..... 18ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 17ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 17ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 7ms DONE
2014_10_12_000000_create_users_table ..... 7ms DONE

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 22ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 6ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 23ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 19ms DONE
2024_01_22_162151_create_tasks ..... 6ms DONE

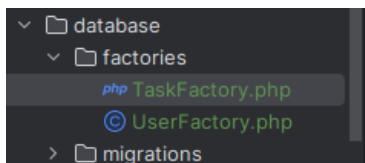
INFO Seeding database.
```

Y aquí vemos cómo se han creado las tablas con sus respectivas columnas.



Crear factory

Creamos el archivo TaskFactory.php



Y dentro tendrá lo siguiente

```
TaskFactory.php x UserFactory.php
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7 /**
8  * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
9  */
10 no usages new *
11 class TaskFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array<string, mixed>
17      */
18     new *
19     public function definition(): array
20     {
21         return [
22             'name' => $this->faker->name(),
23             'description' => $this->faker->unique()->safeEmail(),
24             'completed' => $this->faker->randomElement([0, 1])
25         ];
26     }
27 }
```

Crear seed

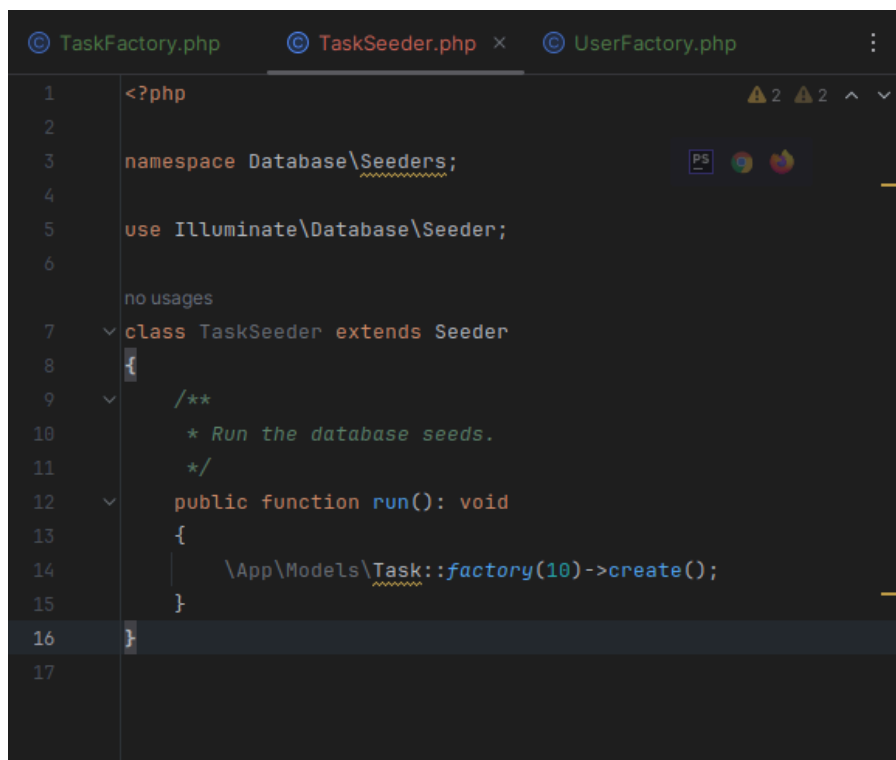
Para crear el seed haremos el siguiente comando.

```
→ MP09_Practica2 git:(main) × php artisan make:seeder TaskSeeder

INFO Seeder [database/seeder/TaskSeeder.php] created successfully.

→ MP09_Practica2 git:(main) ×
```

El contenido del archivo TaskSeeder sera el siguiente.



```
TaskFactory.php TaskSeeder.php × UserFactory.php
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6
7 no usages
8 class TaskSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         \App\Models\Task::factory(10)->create();
16     }
17 }
```

Crear inserts a partir del seed

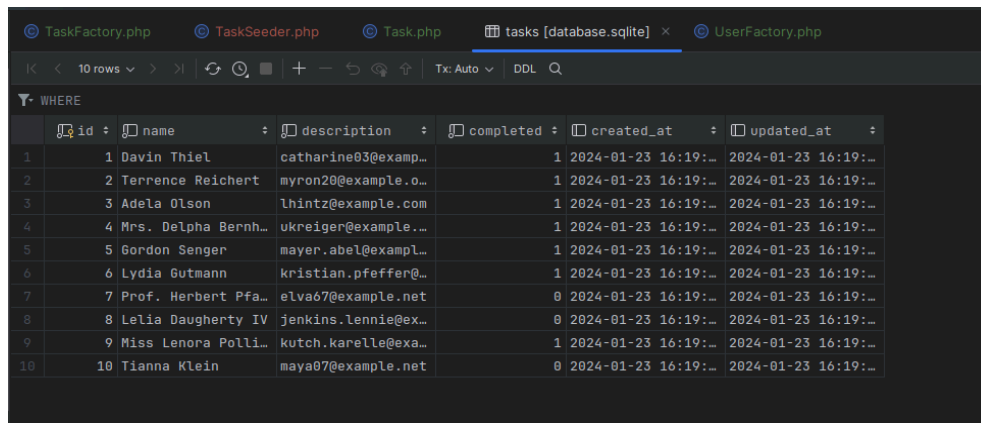
Usamos el siguiente comando para ejecutar el seed especifico para rellenar la base de datos.

```
→ MP09_Practica2 git:(main) × php artisan db:seed --class=TaskSeeder

INFO Seeding database.

→ MP09_Practica2 git:(main) ×
```

Y como podemos ver al ir a la tabla TASKS vemos como se han auto llenado todo



	id	name	description	completed	created_at	updated_at
1	1	Davin Thiel	catharine03@examp...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
2	2	Terrence Reichert	myron20@example.o...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
3	3	Adela Olson	lhintz@example.com	1	2024-01-23 16:19:...	2024-01-23 16:19:...
4	4	Mrs. Delpha Bernh...	ukreiger@example...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
5	5	Gordon Senger	mayer.abel@examp...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
6	6	Lydia Gutmann	kristian.pfeffer@...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
7	7	Prof. Herbert Pfa...	elva67@example.net	0	2024-01-23 16:19:...	2024-01-23 16:19:...
8	8	Lelia Daugherty IV	jenkins.lennie@ex...	0	2024-01-23 16:19:...	2024-01-23 16:19:...
9	9	Miss Lenora Polli...	kutch.karelle@exa...	1	2024-01-23 16:19:...	2024-01-23 16:19:...
10	10	Tianna Klein	maya07@example.net	0	2024-01-23 16:19:...	2024-01-23 16:19:...

Crear controlador de tasks

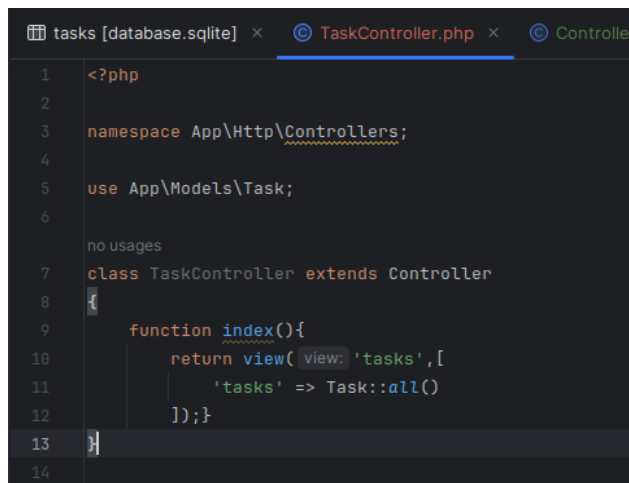
Para crear un controlador de las tablas usaremos el siguiente comando.

```
→ MP09_Practica2 git:(main) × php artisan make:controller TaskController

INFO Controller [app/Http/Controllers/TaskController.php] created successfully.

→ MP09_Practica2 git:(main) ×
```

El contenido de TaskController será el siguiente



```
tasks [database.sqlite] × TaskController.php × Controller.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Task;
6
7 no usages
8 class TaskController extends Controller
9 {
10     function index(){
11         return view( view: 'tasks',[
12             'tasks' => Task::all()
13         ]);}
14 }
```

Craer model de tasks

Para crear el modelo de task usaremos el comando en pantalla.

```
→ MP09_Practica2 git:(main) × php artisan make:model TaskController

INFO Model [app/Models/TaskController.php] created successfully.

→ MP09_Practica2 git:(main) ×
```

El contenido de Task.php será el siguiente.

```
se.sqlite] TaskController.php Task.php x Controller.php ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 new *
9 class Task extends Model
10 {
11     use HasFactory;
12
13     no usages
14     protected $fillable=[
15         'name',
16         'description',
17         'completed',
18     ];
19 }
```

Crear ruta a routes/web.php i afegir la nova ruta al navegador al fitxer resources/views/components/layout.blade.php

Añadimos la ruta dentro de web.php

```
33 Route::get( uri: '/about', [\App\Http\Controllers\PagesController::class, 'about']);
34 Route::get( uri: '/tasks', [\App\Http\Controllers\TaskController::class, 'index']);
35
```

Ahora dentro de layout.blade.php añadiremos las siguiente lineas

```
<a href="/about" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">About</a>  
<a href="/tasks" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">Tasks</a>  
</div>
```


crear la vista tasks.blade.php



Creamos el archivo `tasks.blade.php` que sera parecido al `user.blade.php`

```
tasks [database.sqlite] x TaskControl.php x Task.php x layout.blade.php x tasks.blade.php x mw.web.php x Controller.php
1 <!-- layout -->
2 <div class="bg-gray-800 pb-32">
3
4     <div class="py-10">
5         <div class="mx-auto max-w-7xl px-6 sm:px-6 lg:px-8">
6             <h1 class="text-3xl font-bold tracking-tight text-white">TASQUES</h1>
7         </div>
8     </div>
9 </div>
10
11 <main class="mt-32">
12     <div class="mx-auto max-w-7xl px-6 sm:px-6 lg:px-8">
13         <div class="rounded-lg bg-white px-5 py-6 shadow sm:px-6">
14
15             <div class="px-4 sm:px-6 lg:px-8">
16                 <div class="mt-8 flow-root">
17
18                     <div class="mx-4 my-2 overflow-x-auto sm:mx-6 lg:mx-8">
19                         <div class="inline-block min-w-full py-2 align-middle sm:px-6 lg:px-8">
20                             <table class="min-w-full divide-y divide-gray-300">
21                                 <thead>
22                                     <tr>
23                                         <th scope="col" class="px-3.5 pl-4 pr-3 text-left text-sm font-semibold text-gray-900 sm:pl-3">Id</th>
24                                         <th scope="col" class="px-3 py-3.5 text-left text-sm font-semibold text-gray-900">Nome</th>
25                                         <th scope="col" class="px-3 py-3.5 text-left text-sm font-semibold text-gray-900">Descripción</th>
26                                         <th scope="col" class="px-3 py-3.5 text-left text-sm font-semibold text-gray-900">Completada</th>
27                                     </tr>
28                                 </thead>
29                                 <tbody class="bg-white">
30                                     @foreach($tasks as $task)
31                                         <tr class="bg-gray-50">
32                                             <td class="whitespace-nowrap px-4 pl-4 pr-3 text-sm font-medium text-gray-900 sm:pl-3">{{ $task->id }}</td>
33                                             <td class="whitespace-nowrap px-3 py-4 text-sm font-medium text-gray-500">{{ $task->name }}</td>
34                                             <td class="whitespace-nowrap px-3 py-4 text-sm text-gray-500">{{ $task->description }}</td>
35                                             <td class="whitespace-nowrap px-3 py-4 text-sm text-gray-500">{{ $task->completed }}</td>
36                                         </tr>
37                                     @endforeach
38                                     <!-- More people... -->
39                                 </tbody>
40                             </table>
41                         </div>
42                     </div>
43                 </div>
44             </div>
45         </div>
46     </div>
47 </main>
48 </div>
```

Documentar en captures i comprovació de inserts de tasks i users

Haríamos un php artisan serve para comprobar como nos quedaría todo.

 [Usuaris](#) [Contact](#) [About](#) [Tasques](#)

TASQUES

Id	Nom	Descripció	Completada
1	Davin Thiel	catharine03@example.com	1
2	Terrence Reichert	myron20@example.org	1
3	Adela Olson	lhintz@example.com	1
4	Mrs. Delpha Bernhard V	ukreiger@example.com	1
5	Gordon Senger	mayer.abel@example.com	1
6	Lydia Gutmann	kristian.pfeffer@example.org	1
7	Prof. Herbert Pfannerstill	elva67@example.net	0
8	Lelia Daugherty IV	jenkins.lennie@example.org	0
9	Miss Lenora Pollich V	kutch.karelle@example.com	1
10	Tianna Klein	maya07@example.net	0