

laravel



Joaquin

Joaquin Ciudad Marin
ASIX 2n MP09
Implantació d'Aplicacions Web

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

Primer creen la migració de tasks que serveix per a el context de les bases de dades fan referència a scripts o fitxers que s'utilitzen per fer canvis en l'estructura d'una base de dades. Aquests canvis poden incloure la creació de noves taules, la modificació de taules existents (com afegir, modificar o eliminar columnes), o qualsevol altra alteració a l'estructura de la base de dades.

Comanda per a crear-la:

```
→ laravel-app git:(main) php artisan make:migration create_tasks_table

INFO Migration [database/migrations/2023_12_24_162854_create_tasks_table.php] created successfully.

→ laravel-app git:(main) ×
```

Captura en la migration creada:

```
└─ migrations
  ├── 2014_10_12_000000_create_users_table.php
  ├── 2014_10_12_100000_create_password_reset_tokens_table.php
  ├── 2014_10_12_200000_add_two_factor_columns_to_users_table.php
  ├── 2019_08_19_000000_create_failed_jobs_table.php
  ├── 2019_12_14_000001_create_personal_access_tokens_table.php
  └── 2023_12_11_162545_create_sessions_table.php
  └─ 2023_12_24_162854_create_tasks_table.php

└─ seeders
  └─ DatabaseSeeder.php
```

```
php 2023_12_24_162854_create_tasks_table.php x
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       new *
13       public function up(): void
14       {
15           Schema::create( table: 'tasks', function (Blueprint $table) {
16               $table->id();
17               $table->string( column: 'name');
18               $table->text( column: 'description');
19               $table->boolean( column: 'completed')->default( value: false);
20               $table->timestamps();
21           });
22       }
23
24       /**
25       * Reverse the migrations.
26       */
27       new *
28       public function down(): void
29       {
30           Schema::dropIfExists( table: 'tasks');
31       }
32   };
```

2. Crear factory

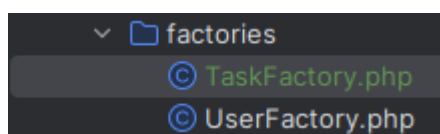
Perquè Laravel reconegui el factory, és possible que necessitem executar algunes ordres d'Artisan. Obrim el terminal a l'arrel del nostre projecte Laravel i executem:

```
→ laravel-app git:(main) x php artisan make:factory TaskFactory

INFO Factory [database/factories/TaskFactory.php] created successfully.

→ laravel-app git:(main) x
```

En aquesta captura veiem el fitxer creat ja.



I aquí tenim el interior del fitxer com seria “factory” que serveix per generar dades de prova de manera ràpida i consistent. És una eina que t'ajuda a crear registres ficticis a la teva base de dades per tal de fer proves, omplir la base de dades durant el desenvolupament, o simular situacions específiques sense haver d'ingressar manualment cada conjunt de dades.

```
© TaskFactory.php x
1  <?php
2
3  namespace Database\Factories;
4
5  use App\Models\Task;
6  use Illuminate\Database\Eloquent\Factories\Factory;
7
8  class TaskFactory extends Factory
9  {
10     /**
11      * Define the model's default state.
12      *
13      * @return array<string, mixed>
14      */
15     public function definition(): array
16     {
17         return [
18             'name' => $this->faker->sentence,
19             'description' => $this->faker->paragraph,
20             'completed' => $this->faker->boolean,
21         ];
22     }
23 }
```

3. Crear seed

Aquí tenim la captura en la comanda per a crear el “seed” que serveix és un terme que s'utilitza per referir-se a l'acció de poblar la base de dades amb dades de prova. Els seeds són scripts o classes que permeten inserir registres ficticis o predefinitos a la base de dades d'una aplicació. Això és especialment útil durant el desenvolupament i les proves de l'aplicació.

Els seeds sovint treballen en conjunt amb les "fàbriques" (factories), que són utilitzades per generar dades de prova específiques per a cada model de la base de dades.

```
→ laravel-app git:(main) ✕ php artisan migrate:fresh --seed

Dropping all tables ..... 5ms DONE

INFO: Preparing database.

Creating migration table ..... 12ms DONE

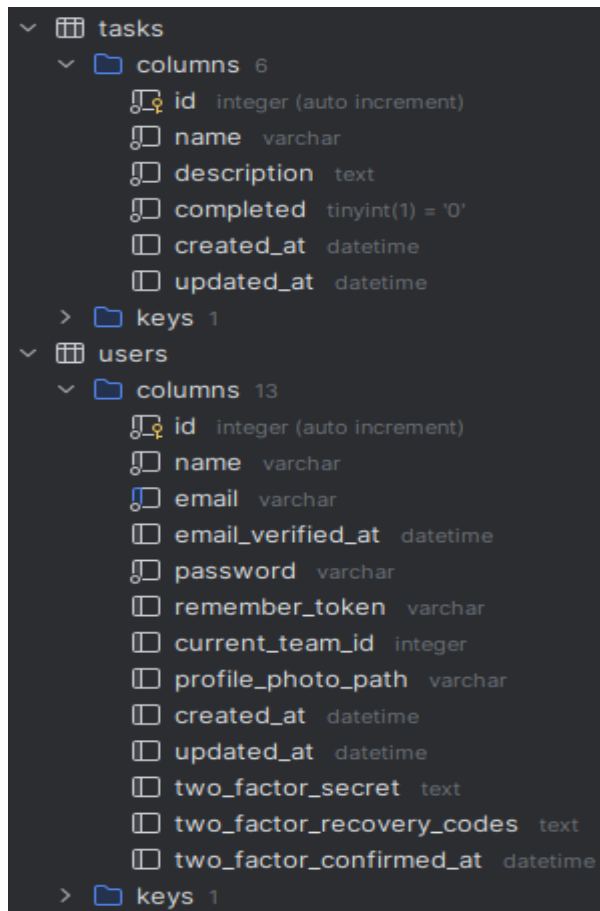
INFO: Running migrations.

2014_10_12_000000_create_users_table ..... 12ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 7ms DONE
2014_10_12_200000_add_two_factor_columns_to_users_table ..... 26ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 10ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 28ms DONE
2023_12_11_162545_create_sessions_table ..... 32ms DONE
2023_12_24_162854_create_tasks_table ..... 7ms DONE

INFO: Seeding database.

→ laravel-app git:(main) ✕
```

Aquí la captura on se veu el seed que hem creat i la DataBase de tasks i users.



4. Crear inserts a partir del seed

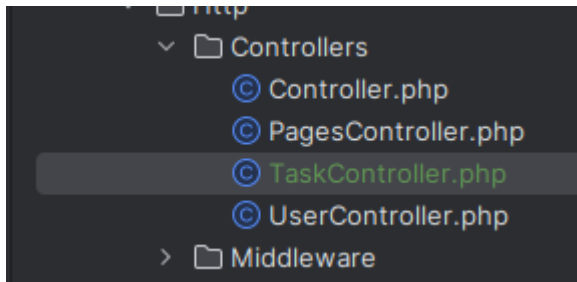
Després de crear el seed podem insertar manualment els inserts com se veu a la captura.

The screenshot shows a database interface with a table containing 3 rows of data. The table has columns: id, name, description, completed, created_at, and updated_at. The data is ordered by description.

	id	name	description	completed	created_at	updated_at
1	1	Planificar vacances	Investigar destins, reservar vols i allotjament.	1	<null>	<null>
2	3	Organitzar un sopar...	Preparar un sopar especial i gaudiu d'una nit amb amics.	1	<null>	<null>
3	2	Desenvolupar una ap...	Utilitzar Laravel per crear una aplicació web interactiva.	1	<null>	<null>

5. Crear controlador de tasks

Ara toca instal·lar els controladors que fa referència a un component específic d'un projecte PHP que maneja les sol·licituds HTTP i coordina la lògica de l'aplicació. Els controladors són part fonamental de l'arquitectura MVC (Model-Vista-Controlador) utilitzada en molts frameworks web PHP, com ara Laravel, Symfony, CodeIgniter, etc.

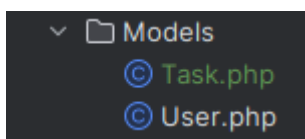


T'ensenyo l'interior del fitxer com poden apreciar porta l'ensencial per a fer una sol·licitud HTTP.

```
© TaskController.php x
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Task;
6
7  new *
8  class TaskController extends Controller
9  {
10     new *
11     public function index()
12     {
13         return view( view: 'tasks', [
14             'tasks' => Task::all(),
15         ]);
16     }
17 }
```

6. Crear model de tasks

Després creen el fitxer “Task” que es refereix a la representació de dades de l'aplicació i les regles de negocis associades a la capa d'accés a dades. En el context de Laravel i molts altres frameworks, els models són classes que interactuen amb la base de dades. Cada model sol representar una taula a la base de dades i proporciona una interfície orientada a objectes per treballar amb aquestes dades.

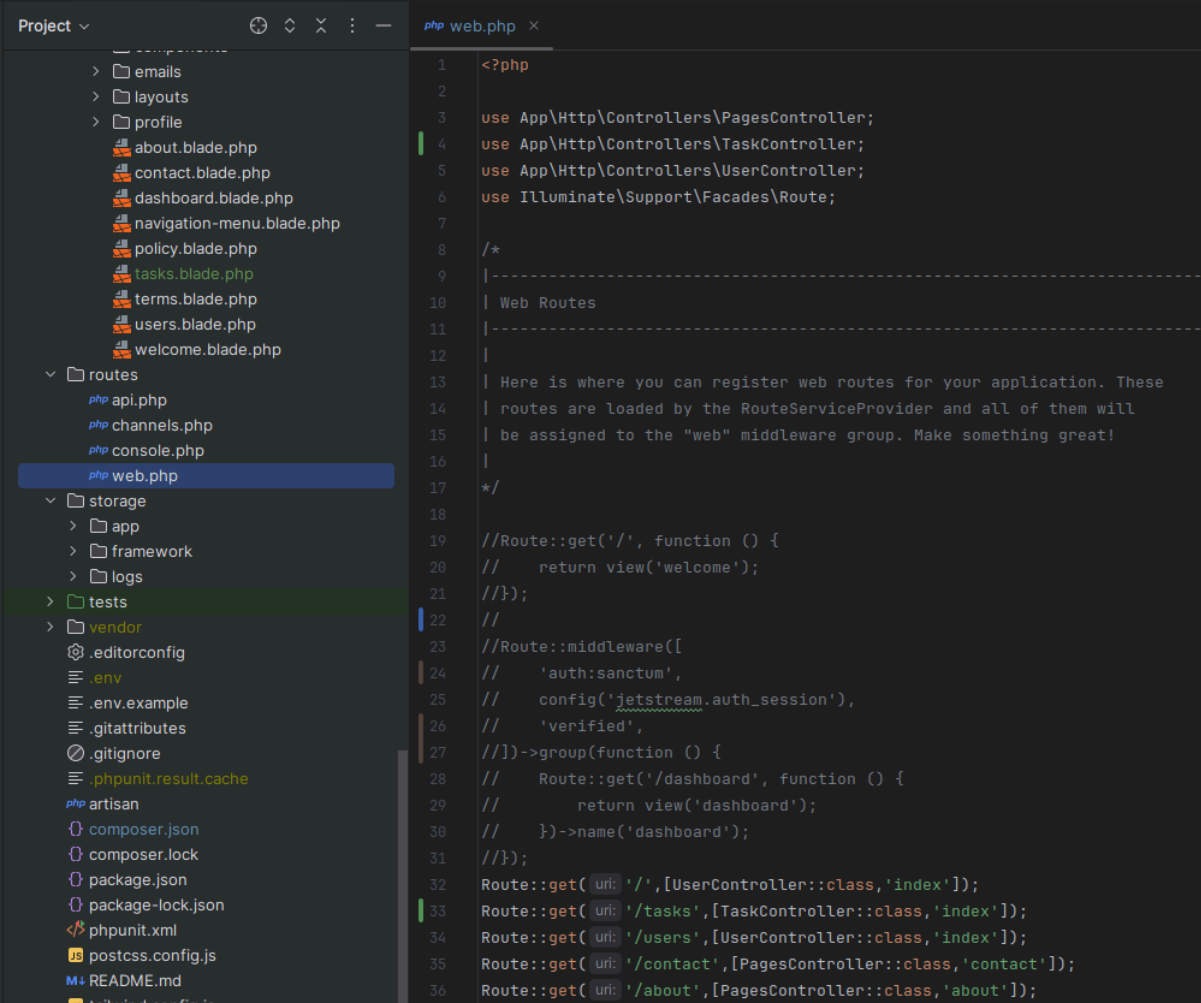


Aquí pots veure l'interior del fitxer.

```
Task.php x
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     /**
14      * The attributes that are mass assignable.
15      *
16      * @var array
17      */
18     no usages
19     protected $fillable = [
20         'name',
21         'description',
22         'completed',
23     ];
24
25     /**
26      * The attributes that should be cast.
27      *
28      * @var array
29      */
30     no usages
31     protected $casts = [
32         'completed' => 'boolean', // Assuming 'completed' is a boolean field
33     ];
34 }
```

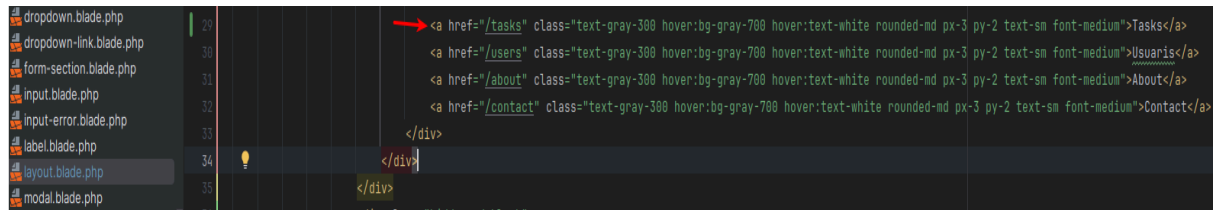

7. *Crear ruta a routes/web.php i afegir la nova ruta al navegador al fitxer resources/views/components/layout.blade.php*

Ara creen les rutes que és el fitxer on defineixen les rutes per manejar les sol·licituds web a la teva aplicació Laravel. És un component fonamental per dirigir les sol·licituds HTTP a les funcions i controladors apropiats a la teva aplicació.



```
1 <?php
2
3 use App\Http\Controllers\PagesController;
4 use App\Http\Controllers\TaskController;
5 use App\Http\Controllers\UserController;
6 use Illuminate\Support\Facades\Route;
7
8 /*
9 |-----
10 | Web Routes
11 |-----
12 |
13 | Here is where you can register web routes for your application. These
14 | routes are loaded by the RouteServiceProvider and all of them will
15 | be assigned to the "web" middleware group. Make something great!
16 |
17 */
18
19 //Route::get('/', function () {
20 //    return view('welcome');
21 //});
22 //
23 //Route::middleware([
24 //    'auth:sanctum',
25 //    config('jetstream.auth_session'),
26 //    'verified',
27 //])->group(function () {
28 //    Route::get('/dashboard', function () {
29 //        return view('dashboard');
30 //    }->name('dashboard'));
31 //});
32 Route::get(uri: '/', [UserController::class, 'index']);
33 Route::get(uri: '/tasks', [TaskController::class, 'index']);
34 Route::get(uri: '/users', [UserController::class, 'index']);
35 Route::get(uri: '/contact', [PagesController::class, 'contact']);
36 Route::get(uri: '/about', [PagesController::class, 'about']);
```

I dins de layout.blade.php creen la linea de codi per a que es veigue a la web al navegador “Tasks”.



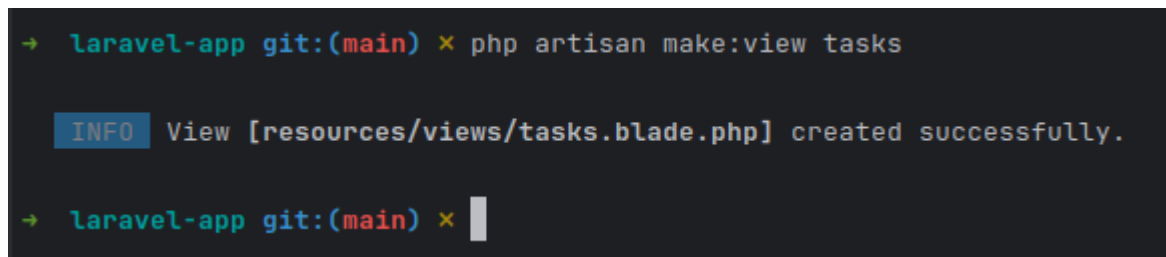
```
dropdown.blade.php
dropdown-link.blade.php
form-section.blade.php
input.blade.php
input-error.blade.php
label.blade.php
layout.blade.php
modal.blade.php

28
29
30
31
32
33
34
35
36

<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>
<a href="/users" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">Users</a>
<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="/contact" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">Contact</a>
</div>
</div>
```

8. crear la vista tasks.blade.php

Ara en aquesta comanda creen tasks.blade.php.

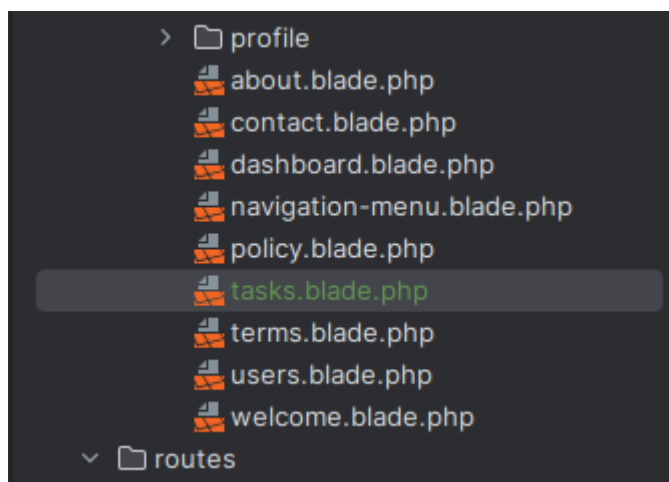


```
→ laravel-app git:(main) × php artisan make:view tasks

INFO View [resources/views/tasks.blade.php] created successfully.

→ laravel-app git:(main) ×
```

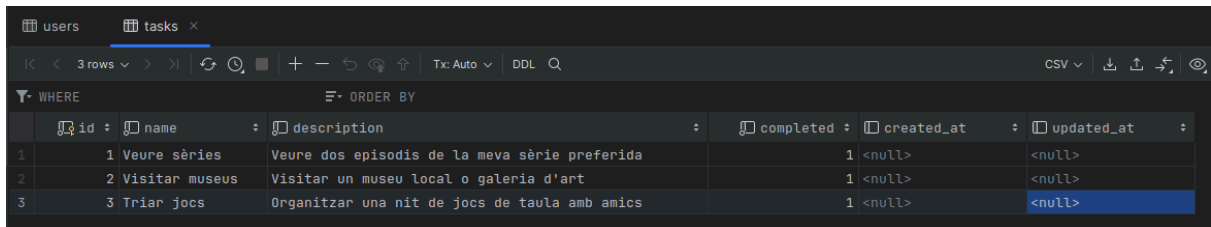
Comprovació de que s'ha creat el fitxer.



9. Documentar en captures i comprovació de inserts de tasks i users

TASKS

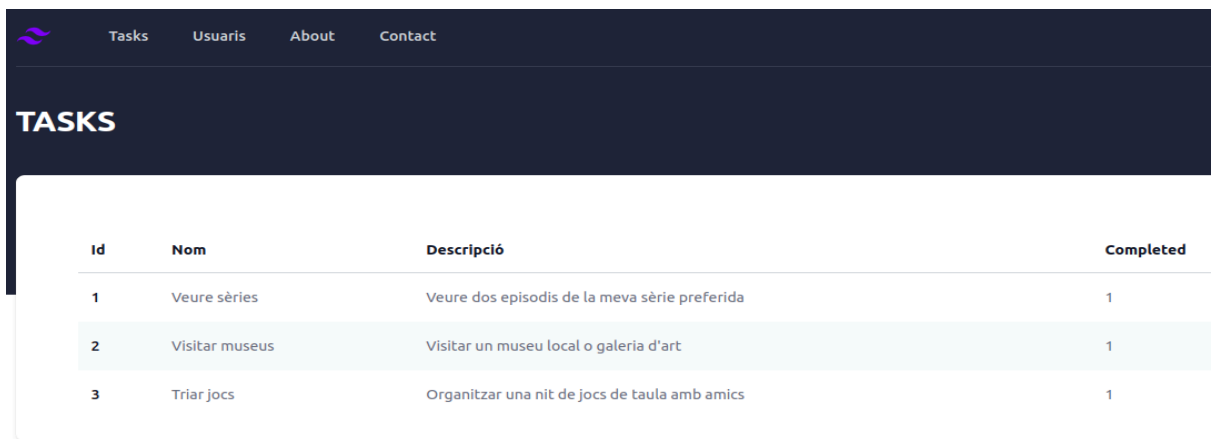
Captura de que puc fer inserts a tasks.



The screenshot shows a database client interface with the 'tasks' table selected. The table has 3 rows and 6 columns: id, name, description, completed, created_at, and updated_at. The 'updated_at' column for the third row is highlighted in blue.

	id	name	description	completed	created_at	updated_at
1	1	Veure sèries	Veure dos episodis de la meva sèrie preferida	1	<null>	<null>
2	2	Visitar museus	Visitar un museu local o galeria d'art	1	<null>	<null>
3	3	Triar jocs	Organitzar una nit de jocs de taula amb amics	1	<null>	<null>

I es veu a la web.

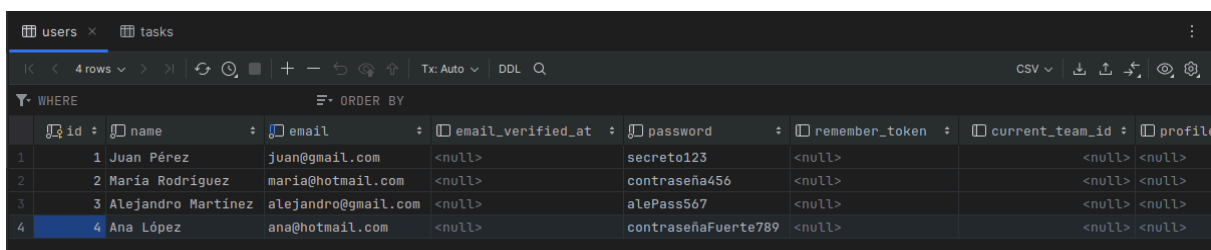


The screenshot shows a web application interface with a navigation bar containing 'Tasks', 'Usuaris', 'About', and 'Contact'. The 'TASKS' section is active, displaying a table with 4 columns: Id, Nom, Descripció, and Completed. The table contains 3 rows of data.

Id	Nom	Descripció	Completed
1	Veure sèries	Veure dos episodis de la meva sèrie preferida	1
2	Visitar museus	Visitar un museu local o galeria d'art	1
3	Triar jocs	Organitzar una nit de jocs de taula amb amics	1

USERS


Captura de que puc fer inserts a users.



The screenshot shows a database client interface with the 'users' table selected. The table has 4 rows and 9 columns: id, name, email, email_verified_at, password, remember_token, current_team_id, and profile_picture. The fourth row is highlighted in blue.

	id	name	email	email_verified_at	password	remember_token	current_team_id	profile_picture
1	1	Juan Pérez	juan@gmail.com	<null>	secreto123	<null>	<null>	<null>
2	2	Maria Rodríguez	maria@hotmail.com	<null>	contraseña456	<null>	<null>	<null>
3	3	Alejandro Martínez	alejandro@gmail.com	<null>	alePass567	<null>	<null>	<null>
4	4	Ana López	ana@hotmail.com	<null>	contraseñaFuerte789	<null>	<null>	<null>

I es veu a la web.

 Tasks Usuaris About Contact		
USUARIS		
Id	Nom	Correu
1	Juan Pérez	juan@gmail.com
2	María Rodríguez	maria@hotmail.com
3	Alejandro Martínez	alejandro@gmail.com
4	Ana López	ana@hotmail.com