

Pista para el examen de Eloy

```
Route::post('libros', [OtroController::class, 'otrafuncion']);  
Route::apiResource('libros', LibroController::class);
```

En cas de dos rutes en el mateix nom, sols capturarà la primera.
Trobar la forma de solucionar.

```
Route::get('/usuarios', [UsuariosController::class, 'search']); //Una forma de solucionar es una  
funcio específica.  
Route::resource('/usuarios', UsuariosController::class);
```

ENUNCIAT DEL PROJECTE

Es desitja ampliar el **projecte inventari** realitzat a classe. Fins ara teníem **dispositius**, **ubicacions** i **usuaris**.

Afegirem un nou recurs: **incidències**. Aquest tindrà els camps: `id`, `descripcio`, `dispositiu_id`, `usuari_registra`, `descripcio_solucio` (pot ser nul), `estat` ('pendent', 'assignat', 'finalitzat') per defecte 'pendent'.

A més, afegirem un camp a la taula **usuaris**: **tipus**. Aquest podrà ser `admin`, `client` o `tècnic`.

Els usuaris **clients** només podran veure les seues incidències i crear-ne noves.



Els **tècnics** consultar les incidències i actualitzar-les. Els camps que podran modificar seran `descripcio_solucio` i `estat`.

Per últim, quan es consulte el **detall d'un dispositiu**, també retornarà totes les incidències que ha tingut.

1. Crear la migración para la tabla `incidencies`

Ejecuta el comando para crear una nueva migración:



bash

 Copiar  Editar

```
php artisan make:migration create_incidencies_table
```

En el archivo generado, define la estructura de la tabla:

php

 Copiar  Editar

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('incidencies', function (Blueprint $table) {
            $table->id();
            $table->string('descripcio');
            $table->unsignedBigInteger('dispositiu_id');
            $table->unsignedBigInteger('usuari_registra');
            $table->text('descripcio_solucio')->nullable();
            $table->enum('estat', ['pendent', 'assignat', 'finalitzat'])->default('pendent');
            $table->timestamps();

            $table->foreign('dispositiu_id')->references('id')->on('dispositius')->onDelete('cascade');
            $table->foreign('usuari_registra')->references('id')->on('usuarios')->onDelete('cascade');
        });
    }



    public function down(): void
    {
        Schema::dropIfExists('incidencies');
    }
};
```



2. Modificar la tabla `usuarios` para añadir el campo `tipus`

Crea una nueva migración para agregar el campo:



bash

 Copiar  Editar

```
php artisan make:migration add_tipus_to_usuarios_table
```

Modifica el archivo de migración:

php

 Copiar  Editar

```
<?php



use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::table('usuarios', function (Blueprint $table) {
            $table->enum('tipus', ['admin', 'client', 'tecnic'])->default('client');
        });
    }

    public function down(): void
    {
        Schema::table('usuarios', function (Blueprint $table) {
            $table->dropColumn('tipus');
        });
    }
};
```

Ejecuta las migraciones:

bash

 Copiar  Editar



```
php artisan migrate
```



3. Crear el modelo Incidencia

Ejecuta:


bash

 Copiar  Editar

```
php artisan make:model Incidencia
```

Define las relaciones en el modelo:

php

 Copiar  Editar

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Incidencia extends Model
{
    protected $table = 'incidencias';

    protected $fillable = [
        'descripcio',
        'dispositiu_id',
        'usuari_registra',
        'descripcio_solucio',
        'estat'
    ];

    public function dispositiu()
    {
        return $this->belongsTo(Dispositiu::class);
    }

    public function usuari()
    {
        return $this->belongsTo(Usuario::class, 'usuari_registra');
    }
}
```



4. Modificar el modelo `Dispositiu` para incluir incidencias

Asegúrate de que el modelo `Dispositiu` tenga una relación con `Incidencia`:

```
php Copiar Editar  
  
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Dispositiu extends Model  
{  
    public function incidencies()  
    {  
        return $this->hasMany(Incidencia::class);  
    }  
}
```

5. Crear el controlador `IncidenciaController`

Ejecuta:

```
bash Copiar Editar  
  
php artisan make:controller Api/IncidenciaController
```

Define las rutas y controladores:

```
php Copiar Editar  
  
<?php  
  
namespace App\Http\Controllers\Api;  
  
use App\Models\Incidencia;  
use App\Models\Dispositiu;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Auth;  
  
class IncidenciaController extends Controller  
{  
    public function __construct()  
    {  
        $this->middleware('auth:sanctum');  
    }  
}
```



```
public function index()
{
    $user = Auth::user();

    if ($user->tipus == 'client') {
        $incidencies = Incidencia::where('usuari_registra', $user->id)->get();
    } elseif ($user->tipus == 'tecnic') {
        $incidencies = Incidencia::all();
    } else {
        return response()->json(['error' => 'No autorizado'], 403);
    }

    return response()->json($incidencies, 200);
}

public function store(Request $request)
{
    $request->validate([
        'descripcio' => 'required',
        'dispositiu_id' => 'required|exists:dispositius,id',
    ]);

    $incidencia = Incidencia::create([
        'descripcio' => $request->descripcio,
        'dispositiu_id' => $request->dispositiu_id,
        'usuari_registra' => Auth::id(),
    ]);

    return response()->json($incidencia, 201);
}

public function update(Request $request, Incidencia $incidencia)
{
    $user = Auth::user();

    if ($user->tipus != 'tecnic') {
        return response()->json(['error' => 'No autorizado'], 403);
    }
}
```

```

$request->validate([
    'descripcio_solucio' => 'nullable|string',
    'estat' => 'required|in:pendent,assignat,finalitzat',
]);

$incidencia->update([
    'descripcio_solucio' => $request->descripcio_solucio,
    'estat' => $request->estat,
]);

return response()->json($incidencia, 200);
}
}

```

6. Configurar las rutas

Añade las rutas en `routes/api.php`:

```

php

use App\Http\Controllers\Api\IncidenciaController;

Route::apiResource('incidencies', IncidenciaController::class);

```

7. Modificar el detalle de un dispositivo

En el controlador de dispositivos, ajusta el método `show` para incluir las incidencias relacionadas:

```

php

public function show(Dispositiu $dispositiu)
{
    $dispositiu->load('incidencies');
    return response()->json($dispositiu, 200);
}

```

