Soluciones - APIs para Desarrolladores (Ejercicios resueltos)

Este paquete incluye implementaciones de ejemplo para los ejercicios de la guía:
- Ejercicio 1: API de Tareas (Laravel + JWT) - controlador, modelo, migración, rutas y test de ejemplo
- Ejercicio 2: API de Blog con Comentarios - modelos, migraciones, controllers, resources y rutas

api_tareas_project/.env.example

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:GENERAR_CON_ARTISAN
APP_DEBUG=true
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=api_tareas
DB_USERNAME=root
DB_PASSWORD=

# JWT
JWT_SECRET=GENERAR_CON_php artisan jwt:secret
```

api_tareas_project/composer.json

```
{
  "name": "api-tareas",
  "require": {
    "php": "^8.0",
    "laravel/framework": "^9.0",
    "tymon/jwt-auth": "^1.0"
  },
  "scripts": {
    "post-root-package-install": [
      "@php -r \"file_exists('.env') || copy('.env.example', '.env');\""
    ]
  }
}
```

api_tareas_project/README.md

```
API de Tareas (Ejercicio 1) - Proyecto de ejemplo
Instrucciones rápidas:
1. Crear proyecto Laravel real: composer create-project laravel/laravel api-tareas
2. Copiar los archivos de 'app/Models', 'app/Http/Controllers', 'database/migrations', 'routes/api.php',
3. Instalar JWT: composer require tymon/jwt-auth
   php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"
   php artisan jwt:secret
4. Configurar .env (DB_*, JWT_SECRET)
5. Ejecutar migraciones: php artisan migrate
6. Ejecutar tests: php artisan test
```

api_tareas_project/app/Models/Tarea.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Tarea extends Model {
```

```
    protected $fillable = ['user_id','titulo','descripcion','completada','fecha_limite'];
    protected $casts = ['completada'=>'boolean','fecha_limite'=>'date'];
    public function usuario(){ return $this->belongsTo(User::class,'user_id'); }
}
```

api_tareas_project/database/migrations/2025_01_01_000000_create_tareas_table.php

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateTareasTable extends Migration {
    public function up(){
        Schema::create('tareas', function(Blueprint $table){
            $table->id();
            $table->foreignId('user_id')->constrained()->onDelete('cascade');
            $table->string('titulo');
            $table->text('descripcion')->nullable();
            $table->boolean('completada')->default(false);
            $table->date('fecha_limite')->nullable();
            $table->timestamps();
        });
    }
    public function down(){ Schema::dropIfExists('tareas'); }
}
```

api_tareas_project/app/Http/Controllers/TareaController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Tarea;
use Illuminate\Http\Request;
use App\Http\Resources\TareaResource;
class TareaController extends Controller {
    public function index(){
        $tareas = auth()->user()->tareas()->orderBy('created_at','desc')->paginate(15);
        return TareaResource::collection($tareas);
    }
    public function show($id){
        $tarea = Tarea::where('user_id', auth()->id())->findOrFail($id);
        return new TareaResource($tarea);
    }
    public function store(Request $request){
        $validated = $request->validate([
            'titulo'=>'required|string|max:255',
            'descripcion'=>'nullable|string',
            'fecha_limite'=>'nullable|date|after:today'
        ]);
        $validated['user_id'] = auth()->id();
        $tarea = Tarea::create($validated);
        return new TareaResource($tarea);
    }
    public function update(Request $request, $id){
        $tarea = Tarea::where('user_id', auth()->id())->findOrFail($id);
        $validated = $request->validate([
            'titulo'=>'sometimes|required|string|max:255',
            'descripcion'=>'nullable|string',
            'completada'=>'boolean',
            'fecha_limite'=>'nullable|date'
        ]);
        $tarea->update($validated);
        return new TareaResource($tarea);
    }
    public function destroy($id){
        $tarea = Tarea::where('user_id', auth()->id())->findOrFail($id);
        $tarea->delete();
```

```
        return response()->json(null,204);
    }
    public function completar($id){
        $tarea = Tarea::where('user_id', auth()->id())->findOrFail($id);
        $tarea->update(['completada'=>true]);
        return new TareaResource($tarea);
    }
}
```

api_tareas_project/app/Http/Resources/TareaResource.php

```
<?php
namespace App\Http\Resources;
use Illuminate\Http\Resources\Json\JsonResource;
class TareaResource extends JsonResource {
    public function toArray($request){
        return [
            'id'=>$this->id,
            'titulo'=>$this->titulo,
            'descripcion'=>$this->descripcion,
            'completada'=>$this->completada,
            'fecha_limite'=>$this->fecha_limite?->format('Y-m-d'),
            'created_at'=>$this->created_at->toISOString(),
            'usuario'=>[
                'id'=>$this->usuario->id,
                'nombre'=>$this->usuario->name
            ]
        ];
    }
}
```

api_tareas_project/routes/api.php

```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TareaController;
use App\Http\Controllers\AuthController;
Route::post('/registro',[AuthController::class,'registro']);
Route::post('/login',[AuthController::class,'login']);
Route::middleware('auth:api')->group(function(){
    Route::get('/perfil',[AuthController::class,'perfil']);
    Route::post('/logout',[AuthController::class,'logout']);
    Route::apiResource('tareas', TareaController::class);
    Route::post('tareas/{id}/completar',[TareaController::class,'completar']);
});
```

api_tareas_project/app/Http/Controllers/AuthController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use Tymon\JWTAuth\Facades\JWTAuth;
class AuthController extends Controller {
    public function registro(Request $request){
        $validated = $request->validate([
            'name'=>'required|string|max:255',
            'email'=>'required|email|unique:users,email',
            'password'=>'required|string|min:8|confirmed'
        ]);
        $user = User::create(['name'=>$validated['name'],'email'=>$validated['email'],'password'=>Hash::
        $token = JWTAuth::fromUser($user);
        return response()->json(['access_token'=>$token,'token_type'=>'bearer','expires_in'=>auth('api')
    }
    public function login(Request $request){
```

```
        $cred = $request->only('email','password');
        if(!$token = auth('api')->attempt($cred)){
            return response()->json(['error'=>'Credenciales inválidas'],401);
        }
        return $this->respuestaConToken($token);
    }
    protected function respuestaConToken($token){
        return response()->json(['access_token'=>$token,'token_type'=>'bearer','expires_in'=>auth('api')
    }
    public function perfil(){ return response()->json(auth('api')->user()); }
    public function logout(){ auth('api')->logout(); return response()->json(['mensaje'=>'Sesión cerrada
    public function refresh(){ $token = auth('api')->refresh(); return $this->respuestaConToken($token);
}
```

api_tareas_project/tests/Feature/Api/TareaTest.php

```
<?php
namespace Tests\Feature\Api;
use Tests\TestCase;
use App\Models\Tarea;
use App\Models\User;
use Illuminate\Foundation\Testing\RefreshDatabase;
class TareaTest extends TestCase {
    use RefreshDatabase;
    /** @test */
    public function usuario_puede_obtener_lista_de_tareas(){
        $user = User::factory()->create();
        Tarea::factory()->count(5)->create(['user_id'=>$user->id]);
        $response = $this->actingAs($user,'api')->getJson('/api/tareas');
        $response->assertStatus(200);
        $response->assertJsonStructure(['data'=>[['id','titulo','descripcion']]]);
    }
}
```

api_blog_project/README.md

```
API de Blog (Ejercicio 2) - Proyecto de ejemplo
Contiene modelos Articulo y Comentario, controladores, migraciones y resources.
Instrucciones: Similar a api_tareas_project; copiar a proyecto Laravel real y ejecutar migraciones.
```

api_blog_project/app/Models/Articulo.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Articulo extends Model {
    protected $fillable = ['user_id','titulo','slug','contenido','imagen','publicado'];
    protected $casts = ['publicado'=>'boolean'];
    public function usuario(){ return $this->belongsTo(User::class,'user_id'); }
    public function comentarios(){ return $this->hasMany(Comentario::class); }
}
```