

Evaluación Unidad 2



INSTITUTO PROFESIONAL
SAN SEBASTIAN

“Integración API JWT + CRUD Web ”

Asignatura:

Desarrollo de Aplicaciones Web 1

Alumnos:

Efren Tovar Silva Rut 25698445-8

Eduardo Ahumada catalán Rut 17304258-2

Profesor:

Victor Cofre

ACREDITACIÓN 2025
*Construimos juntos una
formación de calidad*



1. Introducción	2
2. Entorno de desarrollo	3
3. Desarrollo	4
4.1. Autenticación JWT (API)	4
- Rutas API	
- Controlador de Autenticación	
- Middleware JWT	
4.2. Web de Proyectos (CRUD)	5
- Rutas web	
- Controlador CRUD	
4.3. Modelos y Migraciones	6
- Modelo Proyecto	
- Modelo User	
- Migraciones de base de datos	
4.4. Vistas clave y componente UF	7
- Listar proyectos	
- Crear proyecto	
- Editar proyecto	
- Detalles de proyecto	
- Confirmar eliminación	
- Componente UF (valor del día)	
4. Evidencias	8
Registro de Usuario (API)	
Inicio de Sesión (API)	
Middleware JWT	
Modelo Proyecto + Migración	
Vista CRUD – Listado de Proyectos	
5. Resumen de implementación y entorno de ejecución	10
6. Conclusión	11
7. Referencias	12

INTRODUCCION

Este proyecto aborda la gestión de proyectos en una pyme de tecnología, permitiendo registrar, consultar y administrar iniciativas con atributos como nombre, fecha de inicio, estado, responsable y monto. La solución integra una API REST y una interfaz web unificadas bajo una misma base de código, priorizando seguridad, trazabilidad y una experiencia de uso simple para el equipo.

Objetivo de la entrega

Implementar una API con autenticación JWT para registro e inicio de sesión, junto con un CRUD web protegido para proyectos. La API expone endpoints seguros para autenticación y recursos (crear, listar, actualizar, eliminar), mientras que la capa web consume la API y restringe las acciones sensibles a usuarios autenticados.

ENTORNO DE DESARROLLO

SO: Windows 10/11

Stack local: Vscode, Laragon (NGINX/Apache + MySQL + PHP)

Framework: Laravel 11, PHP 8.2, Composer

Base de datos: MySQL 8 (dump incluido)

Front: Blade, Tailwind vía CDN, Font Awesome

Autenticación JWT (API)

Ruta: routes/api.php (fragmento de código)

```
1. use App\Http\Controllers\AuthController;
Route::post('/register', [AuthController::class, 'register']);
Route::post('/login', [AuthController::class, 'login']);
Route::middleware('jwt.verify')->get('/ping', fn() => response()->json(['ok'=>true]));
```

Controlador: app\Http\Controllers\AuthController.php

Middleware: alias jwt.verify registrado en app\Http\Kernel.php

Esperado:

- POST /api/register -> crea usuario (password hasheada) y devuelve token
- POST /api/login -> devuelve token (bearer)
- GET /api/ping con Authorization: Bearer <token> -> {"ok":true}

Web de Proyectos (CRUD)

Ruta: C:\laragon\www\Tech-Solutions\routes\web.php (fragmento de código)

```

1. use App\Http\Controllers\RutaTechSolutionsController;
2.
3. Route::get('/', fn()=>redirect()->route('proyectos.lista'));
4.
5. Route::resource('proyectos-tech-solutions', RutaTechSolutionsController::class)->names([
6. 'index' => 'proyectos.lista',
7. 'create' => 'proyectos.crear',
8. 'store' => 'proyectos.guardar',
9. 'show' => 'proyectos.mostrarDetalles',
10. 'edit' => 'proyectos.editar',
11. 'update' => 'proyectos.actualizar',
12. 'destroy' => 'proyectos.eliminar',
13. ]);
14.
15. Route::get('proyectos-tech-solutions/{proyectos_tech_solution}/confirmar-eliminar',
16. [RutaTechSolutionsController::class, 'confirmarEliminar']
17. )->name('proyectos.confirmarEliminar');

```

Modelo y Migraciones

Ruta: C:\laragon\www\Tech-Solutions\app\Models\Proyecto.php (fragmento de código)

Modelo

```

1. namespace App\Models;
2.
3. use Illuminate\Database\Eloquent\Factories\HasFactory;
4. use Illuminate\Database\Eloquent\Model;
5. use Illuminate\Database\Eloquent\Relations\BelongsTo;
6.
7. class Proyecto extends Model
8. {
9.     use HasFactory;
10.
11.     protected $table = 'proyectos';
12.
13.     protected $fillable = [
14.         'nombre',
15.         'fecha_inicio',
16.         'estado',
17.         'responsable',
18.         'monto',
19.         'created_by',
20.     ];
21.
22.     protected $casts = [
23.         'fecha_inicio' => 'date',
24.         'monto' => 'decimal:2',
25.     ];
26.
27.     // Mapas estado <-> etiqueta
28.     public const ESTADO_CANON = [
29.         'Pendiente' => 'pendiente',
30.         'En Progreso' => 'en_progreso',
31.         'Completado' => 'completado',
32.         'Pausado' => 'pausado',
33.         'Cancelado' => 'cancelado', // ver Nota al final
34.     ];
35.
36.     public const ESTADO_LABEL = [
37.         'pendiente' => 'Pendiente',
38.         'en_progreso' => 'En Progreso',
39.         'completado' => 'Completado',
40.         'pausado' => 'Pausado',
41.         'cancelado' => 'Cancelado',
42.     ];

```

```

44. // ---- Normaliza ANTES de guardar en BD
45. public function setEstadoAttribute($value): void
46. {
47.     $value = is_string($value) ? trim($value) : $value;
48.
49.     if (isset(self::ESTADO_CANON[$value])) {
50.         $value = self::ESTADO_CANON[$value];
51.     } elseif (is_string($value)) {
52.         $value = strtolower(str_replace(' ', '_', $value));
53.     }
54.
55.     $this->attributes['estado'] = $value;
56. }
57.
58. // ---- Devuelve etiqueta BONITA al leer (para que tus vistas sigan igual)
59. public function getEstadoAttribute($value)
60. {
61.     return self::ESTADO_LABEL[$value] ?? $value;
62. }
63.
64. public function creador()
65. {
66.     return $this->belongsTo(\App\Models\User::class, 'created_by');
67. }
68.
69. // (Opcional) obtener el valor crudo guardado en BD cuando lo necesites
70. public function getEstadoRawAttribute()
71. {
72.     return $this->getRawOriginal('estado');
73. }
74. }

```

```

1. <?php
2.
3. // app\Models\User.php
4.
5. namespace App\Models;
6.
7. use Illuminate\Database\Eloquent\Factories\HasFactory;
8. use Illuminate\Foundation\Auth\User as Authenticatable;
9. use Illuminate\Notifications\Notifiable;
10. use PHPOpenSourceSaver\JWTAuth\Contracts\JWTSubject;
11.
12. class User extends Authenticatable implements JWTSubject
13. {
14.     use HasFactory, Notifiable;
15.
16.     /**
17.      * Campos asignables masivamente.
18.      */
19.     protected $fillable = [
20.         'name',
21.         'email',
22.         'password',
23.     ];
24.
25.     /**
26.      * Campos ocultos en la serialización.
27.      */
28.     protected $hidden = [
29.         'password',
30.         'remember_token',
31.     ];
32.
33.     /**
34.      * Casts de atributos.
35.      * (No incluimos 'password' => 'hashed' para no doble-hashear,
36.      * ya que en el controlador usas Hash::make()).
37.      */

```

```

38. protected function casts(): array
39. {
40.     return [
41.         'email_verified_at' => 'datetime',
42.     ];
43. }
44. /**
45.  * Requerido por JWTSubject.
46.  */
47. public function getJWTIdentifier()
48. {
49.     return $this->getKey();
50. }
51. }
52.
53. public function proyectosCreados()
54. {
55.     return $this->hasMany(Proyecto::class, 'created_by');
56. }
57. /**
58.  * Requerido por JWTSubject.
59.  */
60. public function getJWTCustomClaims(): array
61. {
62.     return [];
63. }
64. }
65. }

```

Migraciones

Ruta: C:\laragon\www\Tech-Solutions\database\migrations(fragmento de código)

- migrations\0001_01_01_000000_create_users_table.php

```

1. <?php
2.
3. // database\migrations\0001_01_01_000000_create_users_table.php
4.
5. use Illuminate\Database\Migrations\Migration;
6. use Illuminate\Database\Schema\Blueprint;
7. use Illuminate\Support\Facades\Schema;
8.
9. return new class extends Migration
10. {
11.     /**
12.      * Ejecutar las migraciones.
13.      */
14.     public function up(): void
15.     {
16.         // Crear la tabla de usuarios
17.         Schema::create('users', function (Blueprint $table) {
18.             $table->id();
19.             $table->string('name');
20.             $table->string('email')->unique();
21.             $table->timestamp('email_verified_at')->nullable();
22.             $table->string('password');
23.             $table->rememberToken();
24.             $table->timestamps();
25.         });
26.         // Crear la tabla de tokens de restablecimiento de contraseña
27.         Schema::create('password_reset_tokens', function (Blueprint $table) {
28.             $table->string('email')->primary();
29.             $table->string('token');
30.             $table->timestamp('created_at')->nullable();
31.         });
32.         // Crear la tabla de sesiones
33.         Schema::create('sessions', function (Blueprint $table) {
34.             $table->string('id')->primary();
35.             $table->foreignId('user_id')->nullable()->index();
36.             $table->string('ip_address', 45)->nullable();
37.             $table->text('user_agent')->nullable();

```

```

40.     $table->longText('payload');
41.     $table->integer('last_activity')->index();
42. });
43. }
44. /**
45.  * Revertir las migraciones
46.  */
47. public function down(): void
48. {
49.     // Eliminar la tabla de sesiones
50.     Schema::dropIfExists('sessions');
51.     Schema::dropIfExists('users');
52.     Schema::dropIfExists('password_reset_tokens');
53.     Schema::dropIfExists('sessions');
54. }
55. }
56. };

```

- migrations\2025_08_14_025652_create_proyectos_table.php

```

1. <?php
2.
3. // database\migrations\2025_08_14_025652_create_proyectos_table.php
4.
5. use Illuminate\Database\Migrations\Migration;
6. use Illuminate\Database\Schema\Blueprint;
7. use Illuminate\Support\Facades\Schema;
8.
9. return new class extends Migration
10. {
11.     /**
12.      * Ejecutar las migraciones
13.      */
14.
15.     public function up(): void
16.     {
17.         // Crear la tabla de proyectos
18.         Schema::create('proyectos', function (Blueprint $table) {
19.             $table->id();
20.             $table->string('nombre');
21.             $table->date('fecha_inicio');
22.             $table->enum('estado', ['pendiente', 'en_progreso', 'finalizado']);
23.             $table->string('responsable');
24.             $table->decimal('monto', 12, 2);
25.             $table->foreignId('created_by')->constrained('users')->cascadeOnDelete();
26.             $table->timestamps();
27.         });
28.     }
29.
30.     /**
31.      * Revertir las migraciones
32.      */
33.     public function down(): void
34.     {
35.         // Eliminar la tabla de proyectos
36.         Schema::dropIfExists('proyectos');
37.     }
38. };

```

- migrations\2025_08_14_172803_alter_estado_enum_in_proyectos.php

```

1. <?php
2.
3. // database\migrations\2025_08_14_172803_alter_estado_enum_in_proyectos.php
4.
5. use Illuminate\Database\Migrations\Migration;
6. use Illuminate\Support\Facades\DB;
7. use Illuminate\Support\Facades\Schema;
8.
9. return new class extends Migration
10. {
11.     /**
12.      * Ejecuta las migraciones

```

```

13. */
14. public function up(): void
15. {
16.     DB::statement("
17.         ALTER TABLE `proyectos`
18.         MODIFY `estado` ENUM('pendiente','en_progreso','completado','pausado','cancelado') NOT NULL
19.     ");
20. }
21.
22. /**
23.  * Revertir las migraciones
24.  */
25. public function down(): void
26. {
27.     DB::statement("
28.         ALTER TABLE `proyectos`
29.         MODIFY `estado` ENUM('pendiente','en_progreso','completado','pausado') NOT NULL
30.     ");
31. }
32. };

```

- migrations\0001_01_01_000002_create_jobs_table.php
- migrations\0001_01_01_000001_create_cache_table.php

Vistas clave (Blade) y componente UF

Vistas web:

resources\views\proyectos\listar.blade.php
resources\views\proyectos\crear.blade.php
resources\views\proyectos\editar.blade.php
resources\views\proyectos\mostrarDetalles.blade.php
resources\views\proyectos\confirmarEliminar.blade.php

Componente UF (caché 23h):

Ruta: resources\views\components\uf.blade.php (fragmento)

```

1. @php
2. $ufValue = null;
3. $cacheKey = 'uf_value_today';
4. $cacheDuration = now()->addHours(23);
5.
6. try {
7.     // Intenta obtener el valor de la caché. Si no existe, lo busca en la API.
8.     $ufValue = \Illuminate\Support\Facades\Cache::remember($cacheKey, $cacheDuration, function () {
9.         $response = \Illuminate\Support\Facades\Http::timeout(5)->get('https://mindicador.cl/api/uf');
10.
11.         // Verifica si la respuesta de la API es exitosa
12.         if ($response->successful()) {
13.             $data = $response->json();
14.             if (isset($data['serie'][0]['valor'])) {
15.                 return (float) $data['serie'][0]['valor'];
16.             }
17.         }
18.
19.         // Si la API falla o el formato es incorrecto, registra el error y no cachea nada.
20.         \Illuminate\Support\Facades\Log::warning('Falló la obtención de la UF desde mindicador.cl.', [
21.             'response_body' => $response->body() ?? 'No se recibió respuesta.'
22.         ]);
23.
24.         return null; // Retorna null para indicar un fallo
25.     });
26.
27. } catch (\Illuminate\Http\Client\RequestException $e) {
28.     // Manejo de errores específicos de HTTP (e.g., 404, 500)

```



```

29.     \Illuminate\Support\Facades\Log::error('Error en la petición a la API de Mindicador: ' . $e->getMessage());
30.     $ufValue = 'Error en la petición a la API de UF';
31. } catch (\Throwable $e) {
32.     // Manejo de cualquier otro tipo de error (conexión, etc.)
33.     \Illuminate\Support\Facades\Log::error('Error general al obtener UF: ' . $e->getMessage());
34.     $ufValue = 'Error de conexión con la API de UF';
35. }
36. @endphp
37.

```

EVIDENCIAS

Con estas Capturas de pantalla Cubro: API funcionando (registro, login, ping), y **CRUD web** visible con estética unificada.

1. Registro de Usuario (API)

Ruta: C:\laragon\www\Tech-Solutions\app\Http\Controllers\AuthController.php

Muestra la función register() con validación, hash de contraseña y emisión del JWT.
Evidencia el cumplimiento de “Controlador de Registro de Usuario con cifrado de clave”.

```

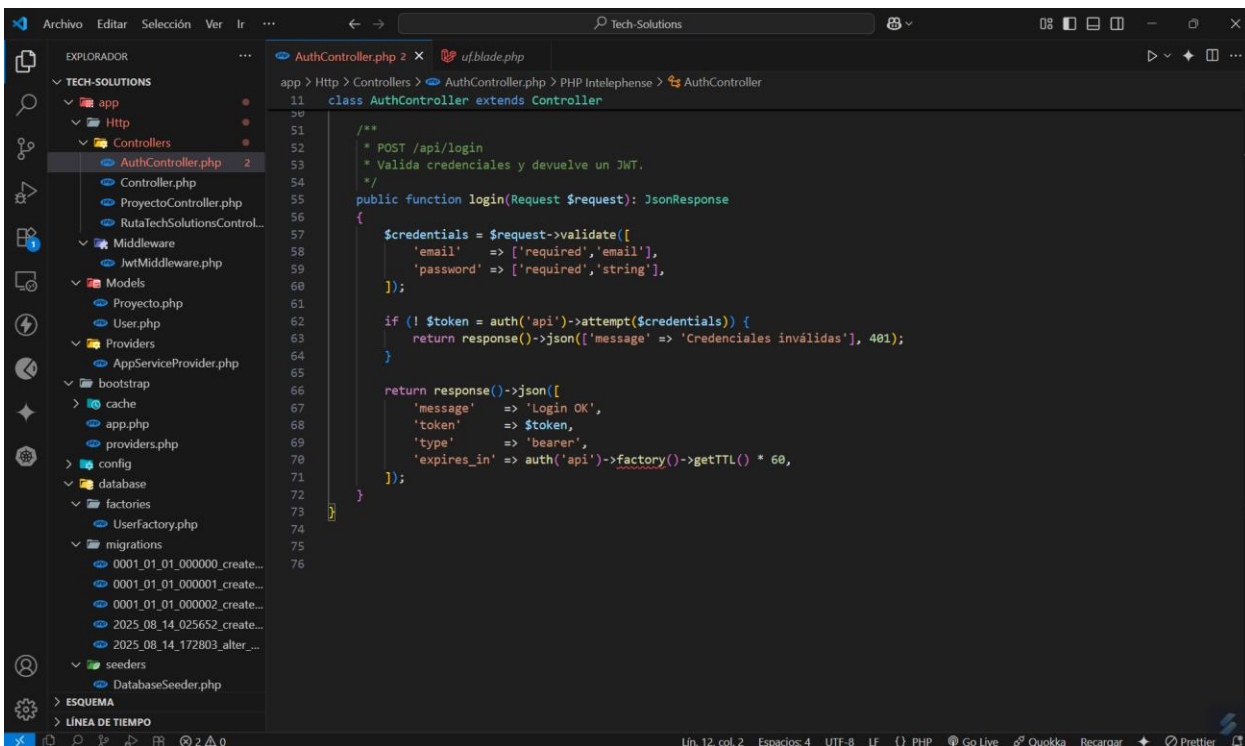
11 class AuthController extends Controller
12 {
13     /**
14      * Register a new user.
15      *
16      * @param Request $request
17      * @return JsonResponse
18      */
19     public function register(Request $request): JsonResponse
20     {
21         $data = $request->validate([
22             'name' => ['required', 'string', 'max:100'],
23             'email' => ['required', 'email', 'max:150', 'unique:users,email'],
24             'password' => ['required', 'string', 'min:8'],
25         ]);
26
27         $user = User::create([
28             'name' => $data['name'],
29             'email' => $data['email'],
30             'password' => Hash::make($data['password']),
31         ]);
32
33         try {
34             $token = auth('api')->login($user);
35         } catch (Throwable $e) {
36             // Si hay un problema con el guard o JWT
37             return response()->json([
38                 'message' => 'Usuario creado, pero no se pudo emitir el token.',
39                 'user' => $user->only(['id', 'name', 'email']),
40                 'error' => 'JWT error',
41             ], 201);
42         }
43
44         return response()->json([
45             'message' => 'Usuario registrado',
46             'user' => $user->only(['id', 'name', 'email']),
47             'token' => $token,
48             'type' => 'bearer',
49             'expires_in' => auth('api')->factory()->getTTL() * 60,
50         ], 201);
51     }
52 }

```

2. Inicio de Sesión (API)

Ruta: C:\laragon\www\Tech-Solutions\app\Http\Controllers\AuthController.php

Muestra la función login() validando credenciales y retornando el JWT.
Evidencia “Controlador de Inicio de Sesión” y la correcta autenticación con token.

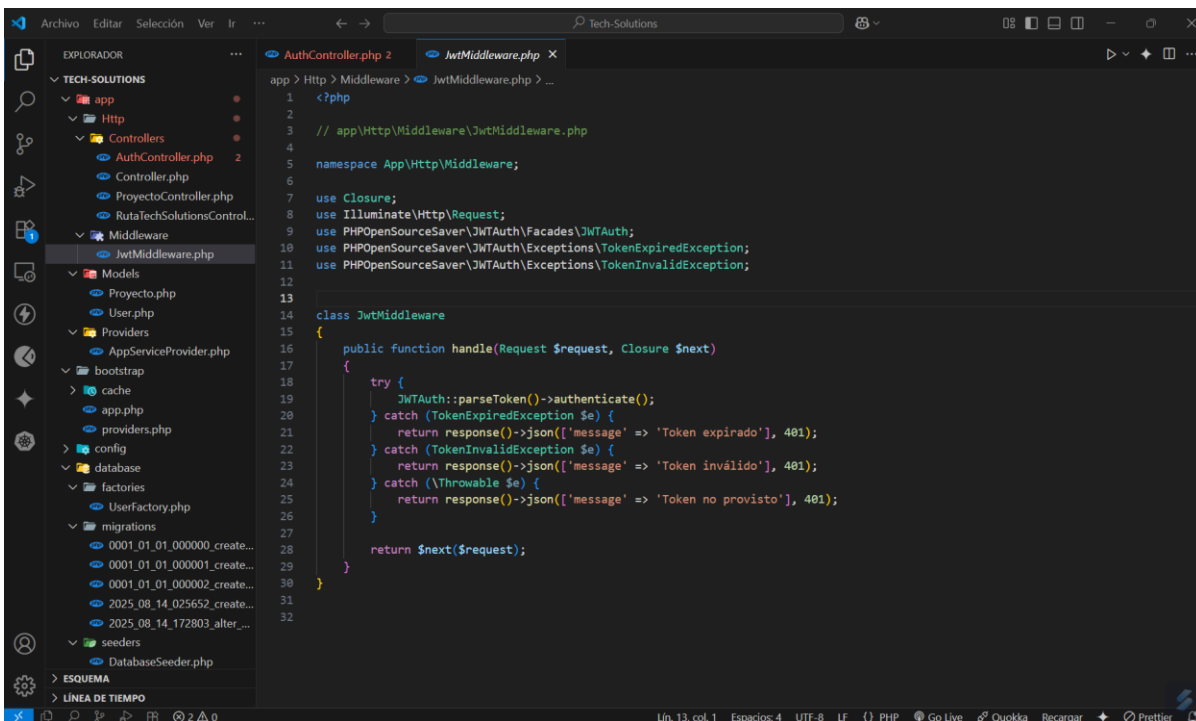


```
11 class AuthController extends Controller
12 {
13     /**
14      * POST /api/login
15      * Valida credenciales y devuelve un JWT.
16      */
17     public function login(Request $request): JsonResponse
18     {
19         $credentials = $request->validate([
20             'email' => ['required', 'email'],
21             'password' => ['required', 'string'],
22         ]);
23
24         if (! $token = auth('api')->attempt($credentials)) {
25             return response()->json(['message' => 'Credenciales inválidas'], 401);
26         }
27
28         return response()->json([
29             'message' => 'Login OK',
30             'token' => $token,
31             'type' => 'bearer',
32             'expires_in' => auth('api')->factory()->getTTL() * 60,
33         ]);
34     }
35 }
```

3. Middleware JWT

Ruta:C:\laragon\www\Tech-Solutions\app\Http\Middleware\JwtMiddleware.php

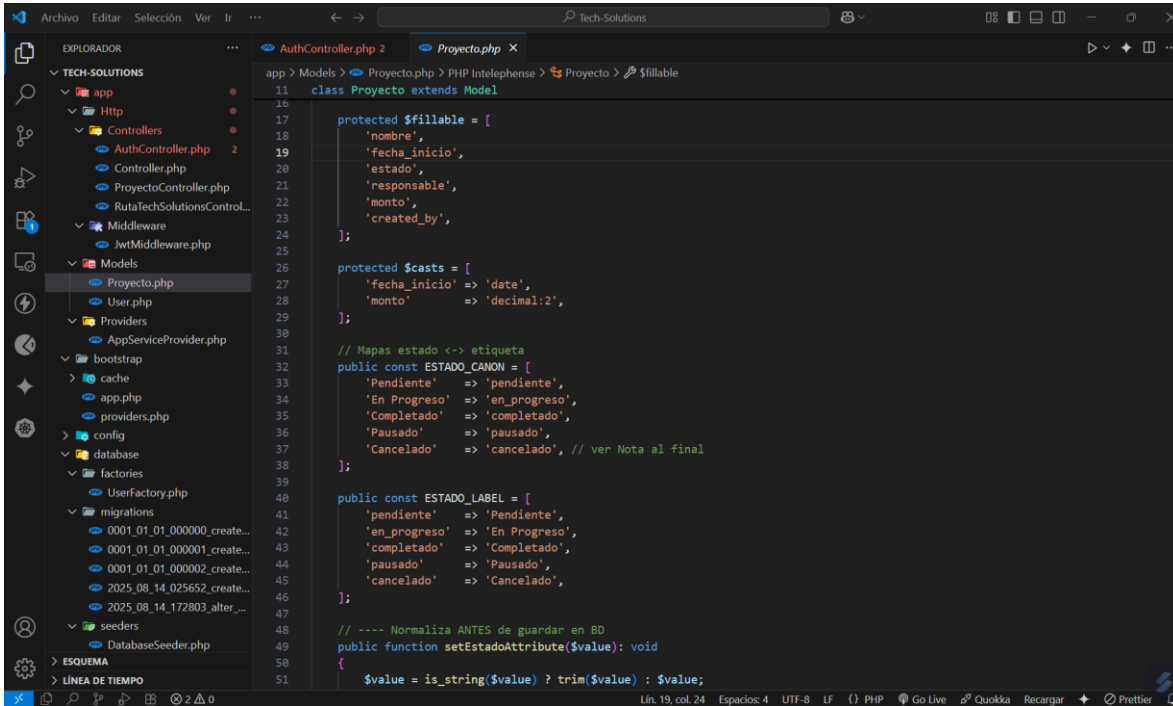
Muestra la validación del token, cumpliendo con “*Middleware para validar autenticación con JWT*”.



```
1 <?php
2
3 // app\Http\Middleware\JwtMiddleware.php
4
5 namespace App\Http\Middleware;
6
7 use Closure;
8 use Illuminate\Http\Request;
9 use PHPOpenSourceSaver\JWTAuth\Facades\JWTAuth;
10 use PHPOpenSourceSaver\JWTAuth\Exceptions\TokenExpiredException;
11 use PHPOpenSourceSaver\JWTAuth\Exceptions\TokenInvalidException;
12
13 class JwtMiddleware
14 {
15     public function handle(Request $request, Closure $next)
16     {
17         try {
18             JWTAuth::parseToken()->authenticate();
19         } catch (TokenExpiredException $e) {
20             return response()->json(['message' => 'Token expirado'], 401);
21         } catch (TokenInvalidException $e) {
22             return response()->json(['message' => 'Token inválido'], 401);
23         } catch (\Throwable $e) {
24             return response()->json(['message' => 'Token no provisto'], 401);
25         }
26
27         return $next($request);
28     }
29 }
30
31
32
```

4. Modelo Proyecto + Migración

Ruta : C:\laragon\www\Tech-Solutions\app\Models\Proyecto.php



```
class Proyecto extends Model
{
    protected $fillable = [
        'nombre',
        'fecha_inicio',
        'estado',
        'responsable',
        'monto',
        'created_by',
    ];

    protected $casts = [
        'fecha_inicio' => 'date',
        'monto' => 'decimal:2',
    ];

    // Mapas estado <-> etiqueta
    public const ESTADO_CANON = [
        'Pendiente' => 'pendiente',
        'En Progreso' => 'en_progreso',
        'Completado' => 'completado',
        'Pausado' => 'pausado',
        'Cancelado' => 'cancelado', // ver Nota al final
    ];

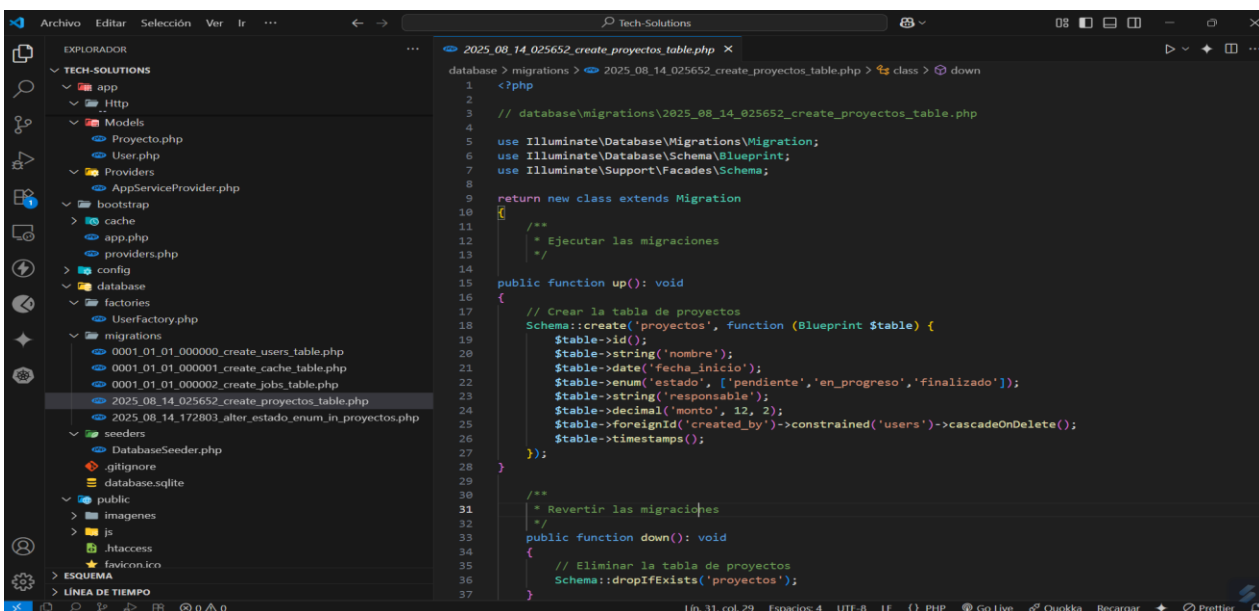
    public const ESTADO_LABEL = [
        'pendiente' => 'Pendiente',
        'en_progreso' => 'En Progreso',
        'completado' => 'Completado',
        'pausado' => 'Pausado',
        'cancelado' => 'Cancelado',
    ];

    // ---- Normaliza ANTES de guardar en BD
    public function setEstadoAttribute($value): void
    {
        $value = is_string($value) ? trim($value) : $value;
    }
}
```

Ruta :

C:\laragon\www\TechSolutions\database\migrations\2025_08_14_025652_create_proyectos_table.php

Evidencia “Modelos completos con todos los campos solicitados y base de datos configurada”.



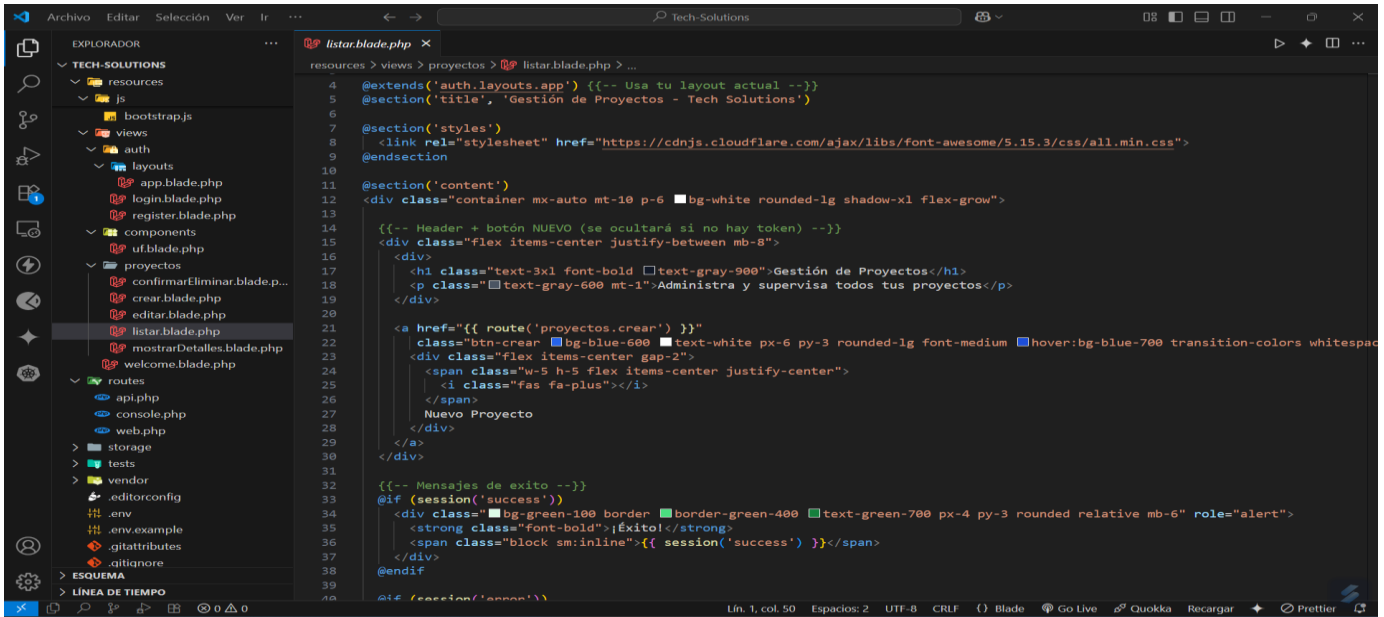
```
class <name> extends Migration
{
    /**
     * Ejecutar las migraciones
     */
    public function up(): void
    {
        // Crear la tabla de proyectos
        Schema::create('proyectos', function (Blueprint $table) {
            $table->id();
            $table->string('nombre');
            $table->date('fecha_inicio');
            $table->enum('estado', ['pendiente', 'en_progreso', 'finalizado']);
            $table->string('responsable');
            $table->decimal('monto', 12, 2);
            $table->foreignId('created_by')->constrained('users')->cascadeOnDelete();
            $table->timestamps();
        });
    }

    /**
     * Revertir las migraciones
     */
    public function down(): void
    {
        // Eliminar la tabla de proyectos
        Schema::dropIfExists('proyectos');
    }
}
```

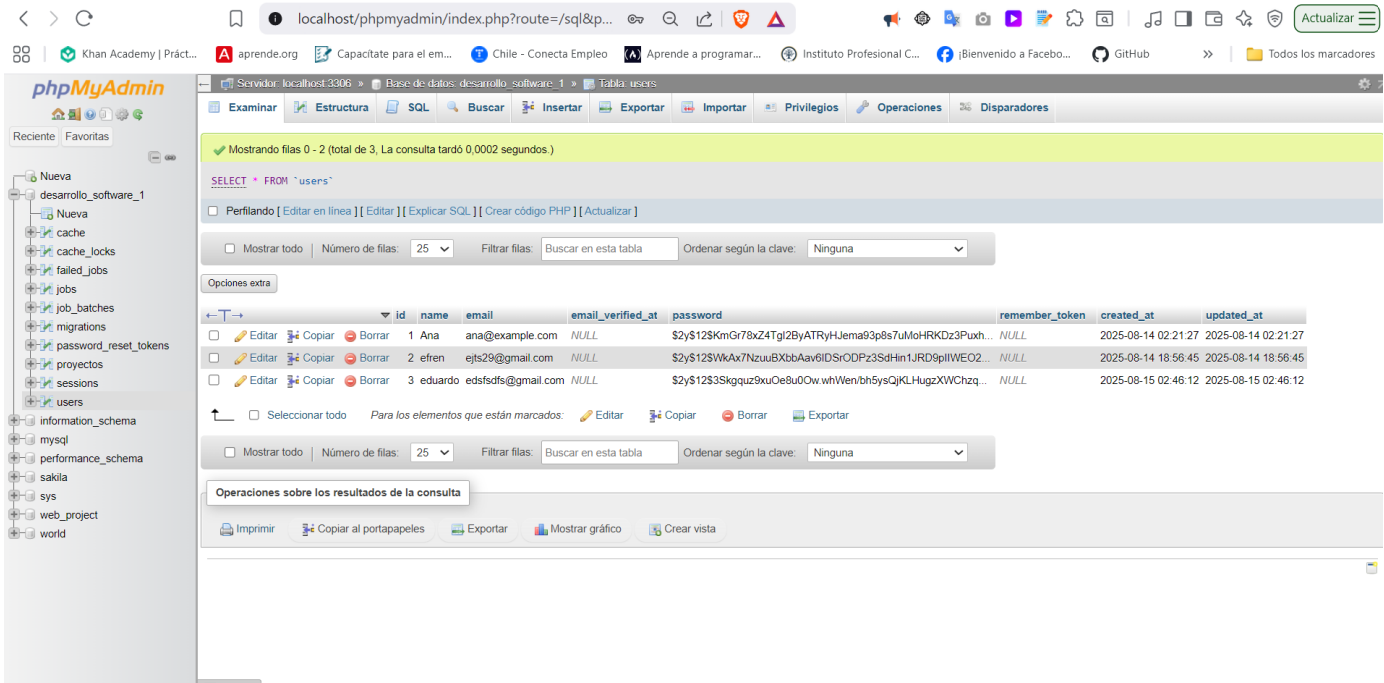
5. Vista CRUD – Listado de Proyectos

Ruta:C:\laragon\www\Tech-Solutions\resources\views\proyectos\listar.blade.php

Demuestra la capa web protegida y operativa, cumpliendo la parte visual solicitada en la evaluación.



```
4 @extends('auth.layouts.app') {{-- Usa tu layout actual --}}
5 @section('title', 'Gestión de Proyectos - Tech Solutions')
6
7 @section('styles')
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
9 @endsection
10
11 @section('content')
12 <div class="container mx-auto mt-10 p-6 bg-white rounded-lg shadow-xl flex-grow">
13
14 <!-- Header + botón NUEVO (se ocultará si no hay token) -->
15 <div class="flex items-center justify-between mb-8">
16 <div>
17 <h1 class="text-3xl font-bold text-gray-900">Gestión de Proyectos</h1>
18 <p class="text-gray-600 mt-1">Administra y supervisa todos tus proyectos</p>
19 </div>
20
21 <a href="{{ route('proyectos.crear') }}"
22 class="btn btn-crear bg-blue-600 text-white px-6 py-3 rounded-lg font-medium hover:bg-blue-700 transition-colors whitespace-no-wrap">
23 <div class="flex items-center gap-2">
24 <span class="w-5 h-5 flex items-center justify-center">
25 <i class="fas fa-plus"></i>
26 </span>
27 Nuevo Proyecto
28 </div>
29 </a>
30 </div>
31
32 <!-- Mensajes de éxito -->
33 <!-- if (session('success')) -->
34 <div class="bg-green-100 border border-green-400 text-green-700 px-4 py-3 rounded relative mb-6 role="alert">
35 <strong class="font-bold">¡Éxito!</strong>
36 <span class="block sm:inline">{{ session('success') }}</span>
37 </div>
38 @endif
39
40 @if (session('error'))
```



localhost/phpmyadmin/index.php?route=/sql&p...

phpMyAdmin

Reciente Favoritas

Nueva desarrollo_software_1 Nueva cache cache_locks failed_jobs jobs job_batches migrations password_reset_tokens proyectos sessions users information_schema mysql performance_schema sakila sys web_project world

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Disparadores

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0.0002 segundos)

SELECT * FROM `users`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
<input type="checkbox"/>	1	Ana	ana@example.com	NULL	\$2y\$12\$KmGr78xZ4Tgl2ByATryHJema93p8s7uMoHrKdZ3Puxh...	NULL	2025-08-14 02:21:27	2025-08-14 02:21:27
<input type="checkbox"/>	2	efren	ejs29@gmail.com	NULL	\$2y\$12\$WkAx7NzuuBXbbAav6IDSrODPz3SdHn1JRD9pIIWE02...	NULL	2025-08-14 18:56:45	2025-08-14 18:56:45
<input type="checkbox"/>	3	eduardo	edsfsdfs@gmail.com	NULL	\$2y\$12\$3Skquz9kuOe8uOwwhWen/bh5ysQjKLHugzXWChzq...	NULL	2025-08-15 02:46:12	2025-08-15 02:46:12

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

127.0.0.1:8000/proyectos-tech-solutions

Khan Academy | Práct... Aprende.org Capacitate para el em... Chile - Conecta Empleo Aprende a programar... Instituto Profesional C... ¡Bienvenido a Facebo... GitHub Todos los marcadores

Tech Solutions Tech Solutions

Proyectos Cerrar sesión UF del Día: **CLP 39.201,37**

Gestión de Proyectos

Administra y supervisa todos tus proyectos

+ Nuevo Proyecto

Total Proyectos 6

Completados 1

En Progreso 2

Pausados 1

Pendientes 1

Lista de Proyectos

ID	Nombre del Proyecto	Fecha de Inicio	Estado	Responsable	Monto	Acciones
7	proyecto5	2025-08-30	Pausado	mirian	\$959.595,00	Ver Editar Eliminar
6	proyecto4	2025-08-29	Cancelado	alejandra	\$858.585,00	Ver Editar Eliminar
5	proyecto3	2025-08-23	Completado	fabiana	\$658.000,00	Ver Editar Eliminar
4	proyecto2	2025-08-23	En Progreso	eduardo ahumada	\$117.000,00	Ver Editar Eliminar
3	proyecto1	2025-08-30	Pendiente	efren tovar	\$50.000,00	Ver Editar Eliminar
1	Implementación PMOOO	2025-08-30	En Progreso	Ana	\$18.000.000,00	Ver Editar Eliminar

RESUMEN DE IMPLEMENTACIÓN Y ENTORNO DE EJECUCIÓN

- **API JWT operativa:** registro/login emiten tokens válidos y ping responde protegido.
- **CRUD web protegido visualmente:** si no hay token, se ocultan acciones y se redirige a login; con token, la UI permite crear, editar y eliminar.
- **Modelo consistente y estados normalizados:** Proyecto con casts, fillable y enum de estado.
- **UI unificada** (Blade + Tailwind) y **componente UF**
- Todo se ejecuta en **Windows + Laragon + Laravel 11**, reproducible con .env, migraciones y phpMyAdmin SQL.

REFERENCIAS

Referencias (APA)

- Laravel. (2024). *Laravel Documentation (v11)*. <https://laravel.com/docs/11.x>
- *jwt-auth (v2.x)*. <https://medium.com/@a3rxander/how-to-implement-jwt-authentication-in-laravel-11-26e6d7be5a41>
- Mindicador. (s. f.). *API de indicadores económicos de Chile*. <https://mindicador.cl/>
- Eva del ipss https://eva.ipss.cl/pluginfile.php/93608/mod_resource/content/8/Apunte%20U2.pdf

- Eva del ipss (2025) <https://eva.ipss.cl/course/view.php?id=979>

CONCLUSION

El proyecto **Tech-Solutions** cumplió los objetivos planteados: integrar una **API REST con autenticación JWT** y un **CRUD web** de proyectos con estética unificada y control visual de acceso. Sobre **Windows + Laragon + Laravel 11**, se implementó el registro y login que devuelven token, la verificación de acceso a rutas protegidas (/api/ping), y la interfaz web (listar/crear/editar/detalles/eliminar) alineada al layout común. Se normalizó el estado del proyecto con **ENUM**, se aplicaron **validaciones server-side**, se manejó el **formateo/casts** (fechas y montos) y se resolvieron incidencias de rutas y nombres de vistas hasta lograr un flujo estable: registro → login → lista. El control de acciones en la UI mediante **token en localStorage** permitió cumplir la rúbrica sin introducir complejidad extra en middleware web.

Las **5 evidencias** seleccionadas demuestran lo esencial: registro, login, ping con token, listado de proyectos y formulario (crear/editar). Con ello se acredita que la API y la capa web funcionan de forma integrada y coherente, y que el sistema persiste datos correctamente en MySQL.

También se utilizó la IA Géminis para organizar ideas, realizar la búsqueda de información y elaborar este informe, con el fin de mejorar su estructura y facilitar su comprensión.

También Se incluye el archivo desarrollo_software_1.sql en la ruta database/backup/ como respaldo completo de la base de datos utilizada, con los registros generados

Ruta : "C:\laragon\www\Tech-Solutions\database\backup\desarrollo_software_1.sql"

También este es el link para github <https://github.com/ejts29/tech-solutions-eva2>