**Laravel API**

**3. Creación de API REST en Laravel**

**3.1 Rutas**

* Crear el archivo de rutas:

php artisan install:api

* Definir rutas en routes/api.php:

Route::apiResource('articulos', ArticuloApiController::class);

**3.2 Controlador**

* Crear controlador API:

php artisan make:controller ArticuloApiController --api

**3.3 Resources**

* Crear Resource para formatear respuestas JSON:

php artisan make:resource ArticuloResource

* Usar en el controlador:

return new ArticuloResource($articulo);

**4. Insertar Datos**

**4.1 Rutas**

* POST /api/articulos → store()

**4.2 Controlador**

* Implementar store() con validación:

public function store(Request $request) {

$request->validate([

'titulo' => 'required|string|max:255',

'contenido' => 'required|string'

]);

return new ArticuloResource(Articulo::create($request->all()));

}

**5. Actualizar Datos**

**5.1 Rutas**

* PUT /api/articulos/{id} → update()

**5.2 Controlador**

* Crear Request personalizado:

php artisan make:request UpdateArticuloRequest

* Implementar validación condicional en rules():

if ($this->method() == 'PUT') {

return [

'titulo' => 'required|string|max:255',

'contenido' => 'required|string'

];

}

return [

'titulo' => 'sometimes|string|max:255',

'contenido' => 'sometimes|string'

];

**6. Autenticación con Sanctum**

**6.1 Generación de Tokens**

* Agregar trait HasApiTokens en User.php:

use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable {

use HasApiTokens;

}

* Generar token en el controlador:

public function crearToken(Request $request) {

$request->user()->tokens()->delete();

$token = $request->user()->createToken('token-api');

return ['token' => $token->plainTextToken];

}

**6.2 Protección de Rutas**

* Aplicar middleware en routes/api.php:

Route::middleware('auth:sanctum')->group(function () {

Route::apiResource('articulos', ArticuloApiController::class);

});

**6.3 Revocación de Tokens**

* Borrar todos los tokens:

$user->tokens()->delete();

* Borrar token actual:

$request->user()->currentAccessToken()->delete();

**6.4 Habilidades de Tokens**

* Crear token con permisos:

$token = Auth::user()->createToken('token-api', ['crear-articulo']);

* Verificar permisos:

if ($user->tokenCan('crear-articulo')) {

// Acción permitida

}

**7. Pruebas con Postman**

* Tipo de solicitud según la operación (GET, POST, PUT, DELETE).
* Enviar JSON en Body para crear o actualizar.
* Incluir Authorization: Bearer <TOKEN> en solicitudes protegidas.