

Historias de usuario

Escaneo de Objetos

Descripción conceptual

Módulo Escaneo de Objetos	<i>Ejemplo: Módulo de Gestión de Usuarios, Módulo de Reportes, etc.</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<p><i>Este módulo se encarga de permitir a los usuarios capturar imágenes de objetos y enviarlas al servidor. Los usuarios pueden</i></p> <ul style="list-style-type: none">• <i>Tomar imágenes de los objetos.</i>• <i>Recibir información procesada de la imagen por un modelo entrenado de ia.</i>• <i>Determinar la categoría del objeto</i>• <i>Recibir recomendaciones o consejos para los objetos</i>• <i>Realizar un análisis básico del objeto</i>• <i>Almacenar los datos del objeto en un historial.</i>

Descripción técnica

En este apartado existen dos partes, la del backend y la del frontend

Backend

Aquí se debe explicar de manera técnica lo esperado,

URL http://localhost:8080/scan	Método POST	Código html 200 OK 400 Bad Request
Descripción <p>Este endpoint recibe una imagen enviada y opcionalmente otros datos (como por ejemplo, un identificador de usuario en caso de usuario registrado). El servidor procesa la imagen utilizando módulos de visión por computadora y aprendizaje automático para clasificar el objeto y generar recomendaciones específicas.</p>		
Datos de entrada POST /scan HTTP/1.1 Host: localhost:8080 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu -----WebKitFormBoundary7MA4YWxkTrZu	Datos de salida 200 OK { "status": "success",	

0gW Content-Disposition: form-data; name="image"; filename="objeto.jpg" Content-Type: image/jpeg [contenido binario de la imagen] -----WebKitFormBoundary7MA4YWxkTrZu 0gW Content-Disposition: form-data; name="userId" 123 -----WebKitFormBoundary7MA4YWxkTrZu 0gW-	"data": { "scanId": 45, "processedAt": "2025-01-31, 14:30:06" } }
POST /scan HTTP/1.1 Host: localhost:8080 Content-Type: multipart/form-data; boundary=-----WebKitFormBoundary7MA4Y WxkTrZu0gW -----WebKitFormBoundary7MA4YWxkTrZu 0gW Content-Disposition: form-data; name="image"; filename="xxxxxxxxx" Content-Type: image/jpeg [contenido binario de la imagen] -----WebKitFormBoundary7MA4YWxkTrZu 0gW Content-Disposition: form-data; name="userId" 123 -----WebKitFormBoundary7MA4YWxkTrZu 0gW-	400 Bad Request { "status": "error", "message": "Imagen no válida o parámetros incompletos" }

URL http://localhost:8080/scan	Método GET	Código html 200 OK 400 Bad Request
Descripción Este endpoint permite consultar los detalles de un escaneo previamente realizado, devolviendo la información de clasificación y recomendaciones asociadas.		
Datos de entrada	Datos de salida 200 OK {	

	<pre> "status": "success", "data": { "scanId": 45, "classification": "Plástico", "recommendations": "Lavar el objeto y separarlo en la sección de plásticos", "processedAt": "2025-01-31, 14:30:06", "imageUrl": "http://localhost:8080/images/scan_45.jpg" } } </pre>
	<p style="text-align: center;">400 Bad Request</p> <pre> { "status": "error", "message": "No se encontró un escaneo con el ID proporcionado" } </pre>

Frontend

Interacción esperada:

El usuario al entrar al escaneo de objeto se le abrirá un menú con los siguientes botones.

Botón “Escanear nuevo objeto” → Abre un formulario modal o activa la cámara para capturar una imagen del objeto.

Botón “Ver detalles” → Muestra un resumen detallado del escaneo, incluyendo la imagen, recomendaciones y fecha de procesamiento.

Botón “Reintentar análisis” (si el estado es “Error”) → Permite volver a enviar la imagen para procesamiento.

Botón “Subir imagen” → Permite al usuario subir una imagen de su galería.

Mockups/Prototipos:

1. *Menú principal del scan*
2. *Cámara del scan*
3. *Detalles del scan. Cuando el usuario escanea correctamente y presiona el botón ver detalles*
4. *Mensaje de error de escaneo.*

Flujo visual y eventos:

