# Informe de Funcionamiento y Funcionalidades - Módulo Fotos

## 1. Objetivo General

Definir el funcionamiento integral del módulo de Fotos (front-end y back-end) incluyendo la gestión desde la interfaz pública, controles de administrador y panel de gestión, así como la estructura de datos, endpoints y validaciones.

## 2. Flujo de Usuario y Páginas

- Público:  
 • /fotos: Galería pública con chip de ordenamiento (newest, oldest, taken-desc, taken-asc, relevant).  
 • /fotos/:id: Página de detalle con imagen, descripción y metadatos.  
- Controles admin en público (si logueado):  
 • Botón 'Agregar Nueva Foto' → /fotos/agregarfoto  
 • Botones en cada foto: Editar, Ocultar/Mostrar, Visitar  
- Panel /admin/fotos: Listado con búsqueda, filtros, acciones masivas y reordenamiento de destacadas.  
- Editor:  
 • Crear: /fotos/agregarfoto  
 • Editar: /fotos/:id/editar  
 • Diferencias: al crear exige imagen; al editar permite reemplazo y muestra historial.

## 3. Modelo de Datos (DTOs)

Photo:  
 id, title, subtitle, descriptionMd, alt, location, takenDate, uploadedDate, imageUrl,  
 featured, visits, order, tags, isVisible, createdBy, createdAt, updatedAt  
  
PhotoListResponse:  
 items, page, limit, totalItems, totalPages  
  
HistoryEntry:  
 id, photoId, at, by, action, changes

## 4. Endpoints Back-end

Público:  
 GET /api/public/photos  
 GET /api/public/photos/:id  
 POST /api/public/photos/:id/visit  
  
Admin:  
 POST /api/photos  
 PUT /api/photos/:id  
 PATCH /api/photos/:id/visibility  
 PATCH /api/photos/:id/featured  
 PATCH /api/photos/reorder  
 GET /api/photos/:id/history  
 DELETE /api/photos/:id (fase 2)

## 5. Validaciones y Errores

• title requerido (1-120 chars)  
• descriptionMd requerido (1-10k chars)  
• image obligatoria al crear (jpg/png/webp máx. 12MB)  
• tags ≤ 10, cada uno máx. 30 chars  
• takenDate ISO válida  
• order solo si featured=true

## 6. Manejo de Imágenes

• Almacenamiento en disco/S3/Cloudinary  
• Renombrado estándar: photo\_<id>.<ext>  
• Conversión opcional a WebP  
• Validación MIME real y tamaño  
• Eliminación de metadatos EXIF  
• Cache-Control y manejo de invalidación al reemplazar

## 7. Historial

• Registra: create, update, toggle visibility, toggle featured, reorder, delete  
• Guarda usuario, fecha y cambios por campo  
• Mostrado en el editor en lista paginada

## 8. Ordenamiento

• newest → uploadedDate DESC  
• oldest → uploadedDate ASC  
• taken-desc → takenDate DESC  
• taken-asc → takenDate ASC  
• relevant → visits DESC

## 9. Migración desde mock (photosData)

1) Crear servicio photosService que primero use mock  
2) Cambiar baseURL a /api/public cuando el back esté listo  
3) Adaptar paginación al servidor  
4) Inyectar controles admin según auth  
5) Renderizar descriptionMd en detalle

## 10. Internacionalización

• UI con i18n (ES/EN)  
• descriptionMd solo en ES en esta fase  
• Traducciones de contenido como fase 2

## 11. Métricas y Seguridad

• Incremento de visits solo en detalle  
• Rate limit en /visit  
• Optimistic UI con rollback en fallos  
• Confirmaciones en acciones críticas  
• Accesibilidad en UI

## 12. Checklist de GO

✓ DTO y contrato API  
✓ Sort keys  
✓ Rutas front  
✓ Endpoints definidos  
✓ Validaciones y errores  
✓ Manejo de imágenes  
✓ Historial desde inicio  
✓ Migración mock → API  
✓ Seguridad y UX

**Fase 1 – Backend base**

1. **Modelos y esquema** (Photo, History, User) usando lo que definimos y lo que ya tenés en el PDF del backendModelo Backend.
2. **Rutas públicas** (GET lista/detalle, POST visitas).
3. **Rutas admin** (crear, editar, visibilidad, featured, reorder, historial).
4. **Subida de imágenes** con validaciones y almacenamiento.
5. **Middleware de auth** (JWT + roles).

**Fase 2 – Front integración inicial**

1. Crear **photosService** que ahora lea photosData (mock) pero con misma interfaz que la API.
2. Inyectar en /fotos y /fotos/:id los **controles de admin**.
3. Crear rutas de editor /fotos/agregarfoto y /fotos/:id/editar con formulario + validaciones.

**Fase 3 – Conexión real**

1. Cambiar photosService a usar API real.
2. Pasar paginación y ordenamiento al servidor.
3. Integrar historial en editor.
4. Añadir /admin/fotos con listado masivo, filtros y drag&drop de destacadas.

**Fase 4 – Ajustes y extras**

* Optimistic UI, toasts, confirmaciones.
* WYSIWYG/Markdown en descripción.
* Manejo avanzado de imágenes (WebP, compresión).
* Seguridad extra (rate limit visitas, sanitización HTML si no usamos Markdown puro).