

Nata Foto - Galería Fotográfica Personal

Una aplicación web moderna y elegante para gestionar y mostrar una galería de fotos personal con panel de administrador integrado.

Características

- **Galería Pública:** Visualización atractiva de fotos con modal de vista detallada
- **Panel de Administrador:** Sistema completo de gestión con autenticación
- **Subida de Imágenes:** Drag & drop interface para subir fotos fácilmente
- **Responsive Design:** Funciona perfectamente en móviles y escritorio
- **Backend Seguro:** Integración completa con Supabase (Database + Storage + Edge Functions)
- **Interfaz en Español:** Toda la UI y mensajes en español
- **Diseño Profesional:** Estética fotográfica con colores cálidos y animaciones suaves

Tecnologías Utilizadas

- **Frontend:** React 18 + TypeScript + Vite
- **Styling:** Tailwind CSS + Custom Components
- **Backend:** Supabase (PostgreSQL + Storage + Edge Functions)
- **UI Components:** Custom components con React
- **File Upload:** React Dropzone
- **Notifications:** React Hot Toast
- **Icons:** Lucide React

Estructura del Proyecto

```
nata-foto/
├── src/
│   ├── components/
│   │   ├── admin/          # Panel de administrador
│   │   ├── auth/           # Autenticación
│   │   ├── gallery/        # Galería de fotos
│   │   ├── layout/         # Layout components
│   │   ├── ui/             # Components UI base
│   │   └── upload/         # Subida de archivos
│   ├── hooks/              # React hooks personalizados
│   ├── lib/                # Utilidades y configuración
│   ├── App.tsx             # Componente principal
│   └── main.tsx            # Punto de entrada
├── supabase/
│   ├── functions/          # Edge Functions
│   │   ├── upload-photo/   # Subida segura de imágenes
│   │   ├── admin-auth/     # Autenticación de administrador
│   │   └── manage-photos/   # Gestión de fotos
└── public/                 # Archivos estáticos
```

Configuración y Deployment

Prerequisitos

1. **Cuenta de Supabase** configurada con:
 - Base de datos PostgreSQL
 - Storage bucket: `nata-foto-gallery`
 - Edge Functions desplegadas
 - Tablas: `admin_users`, `photos`, `gallery_settings`
2. **Cuenta de Vercel** para deployment

Configuración del Backend (Ya Configurado)

El backend de Supabase ya está completamente configurado con:

- **Base de Datos:**
 - `admin_users` : Usuarios administradores
 - `photos` : Metadatos de las fotos
 - `gallery_settings` : Configuraciones de la galería
- **Storage:**
 - Bucket `nata-foto-gallery` con acceso público
 - Límite de 10MB por archivo
 - Solo archivos de imagen permitidos
- **Edge Functions:**
 - `upload-photo` : Subida segura de imágenes
 - `admin-auth` : Autenticación de administradores
 - `manage-photos` : Gestión completa de fotos

Credenciales de Administrador

Usuario: `natafoto`

Contraseña: `NataFoto123`

Deployment en Vercel

1. Subir a Git:

```
bash # En la carpeta nata-foto/ git init git add . git commit -m  
"Initial commit - Nata Foto application" git branch -M main git  
remote add origin YOUR_REPOSITORY_URL git push -u origin main
```

2. Conectar con Vercel:

- Ve a vercel.com
- Conecta tu repositorio Git
- Vercel detectará automáticamente que es un proyecto Vite/React

3. Configuración Automática:

- Build Command: `pnpm build`
- Output Directory: `dist`
- Install Command: `pnpm install`

4. **Deploy:** Vercel desplegará automáticamente tu aplicación

Variables de Entorno

Las credenciales de Supabase están hardcodeadas en el código para simplificar el deployment. En un entorno de producción real, deberías usar variables de entorno:

```
VITE_SUPABASE_URL=your_supabase_url  
VITE_SUPABASE_ANON_KEY=your_supabase_anon_key
```

Funcionalidades Principales

Para Visitantes

- Ver galería de fotos públicas
- Modal de vista detallada de cada foto
- Subir fotos (sin autenticación requerida)
- Diseño responsive

Para Administradores

- Panel de administración completo
- Estadísticas de la galería
- Gestión de todas las fotos (ver, eliminar, actualizar)
- Subida de fotos con metadatos completos
- Autenticación segura

Desarrollo Local

```
# Instalar dependencias
pnpm install





# Ejecutar en desarrollo
pnpm dev

# Build para producción
pnpm build

# Preview del build
pnpm preview
```

Responsive Design

La aplicación está completamente optimizada para:

-  Móviles (320px+)
-  Tablets (768px+)
-  Desktop (1024px+)
-  Large screens (1440px+)

Diseño y UX

- **Paleta de colores:** Tonos cálidos (amber, orange) que evocan la fotografía
- **Tipografía:** Inter font para legibilidad óptima
- **Animaciones:** Transiciones suaves y micro-interacciones
- **Accesibilidad:** Soporte para lectores de pantalla y navegación por teclado

Seguridad

- Autenticación segura para administradores
- Validación de tipos de archivo
- Límites de tamaño de archivo
- Edge Functions para operaciones sensibles
- Sanitización de inputs

Métricas y Analytics

El panel de administrador incluye:

- Total de fotos subidas
- Fotos subidas este mes
- Uso total de almacenamiento
- Tamaño promedio de archivos

Actualizaciones Futuras

Posibles mejoras:

- ☐ Sistema de etiquetas avanzado
- ☐ Búsqueda y filtros
- ☐ Categorías personalizables
- ☐ Comentarios en fotos
- ☐ Sistema de likes
- ☐ Integración con redes sociales
- ☐ Watermarks automáticos
- ☐ Backup automático

Solución de Problemas

Error de Upload

- Verificar que el archivo sea una imagen válida
- Comprobar que el tamaño no exceda 10MB
- Revisar la conexión a internet

Error de Autenticación

- Verificar credenciales: `natafoto` / `NataFoto123`
- Comprobar que las Edge Functions estén funcionando

Problemas de Performance

- Las imágenes se optimizan automáticamente
- Lazy loading implementado para mejor performance

Soporte

Esta aplicación está lista para uso en producción con todas las funcionalidades implementadas y probadas.

Desarrollado con  para crear experiencias fotográficas memorables