

# Acortador de URLs con Código QR

Un servicio moderno de acortamiento de URLs con generación de códigos QR, construido con Node.js, Express, MongoDB y React.

## Características

- ✨ **Acortamiento de URLs** - Convierte URLs largas en enlaces cortos y memorables
- 📱 **Generación de códigos QR** - Crea códigos QR automáticamente para cada URL
- 📊 **Estadísticas de clics** - Rastrea el número de visitas y fecha de último acceso
- 🎨 **Interfaz moderna** - Diseño responsive con animaciones suaves
- 📄 **Descarga de códigos QR** - Guarda los códigos QR como imágenes
- 🔒 **Validación de URLs** - Verifica que las URLs sean válidas antes de procesarlas
- 🎯 **Redirección automática** - Redirige instantáneamente a la URL original

## Tecnologías Utilizadas

### Backend

- **Node.js** - Entorno de ejecución de JavaScript
- **Express.js** - Framework web minimalista
- **MongoDB** - Base de datos NoSQL
- **Mongoose** - ODM para MongoDB
- **nanoid** - Generador de IDs únicos
- **valid-url** - Validador de URLs

### Frontend

- **React** - Biblioteca de JavaScript para interfaces de usuario
- **TypeScript** - JavaScript con tipado estático
- **Framer Motion** - Biblioteca de animaciones
- **qrcode.react** - Generador de códigos QR para React
- **CSS moderno** - Estilos con efectos glassmorphism

## Requisitos Previos

Antes de comenzar, asegúrate de tener instalado:

- Node.js (versión 14 o superior)
- MongoDB (local o MongoDB Atlas)

- npm o yarn

## Instalación

### 1. Clonar el repositorio

```
git clone https://github.com/tu-usuario/short-url.git
cd short-url
```

### 2. Instalar dependencias del backend

```
npm install
```

### 3. Instalar dependencias del frontend

```
cd frontend
npm install
cd ..
```

### 4. Configurar variables de entorno

Crea un archivo `.env` en la raíz del proyecto:

```
MONGODB_URI=mongodb://localhost:27017/shorturl
PORT=3000
NODE_ENV=development
```

## Ejecución

### Desarrollo

Para ejecutar la aplicación en modo desarrollo:

**Backend** (desde la raíz del proyecto):

```
npm run dev
```

**Frontend** (en una nueva terminal):

```
cd frontend
```

npm start

La aplicación estará disponible en:

- Frontend: <http://localhost:3001>
- Backend API: <http://localhost:3000>

## Producción

Para ejecutar en modo producción:

npm start



## Uso de la API

### Acortar una URL

POST /api/shorten

Content-Type: application/json

```
{
  "originalUrl": "https://www.ejemplo.com/url-muy-larga"
}
```

### Respuesta exitosa:

```
{
  "success": true,
  "data": {
    "shortUrl": "abc123",
    "originalUrl": "https://www.ejemplo.com/url-muy-larga",
    "qrCode": "data:image/png;base64,...",
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

### Redirección

GET /:shortUrl

Redirige automáticamente a la URL original y registra la visita.

## Obtener estadísticas

GET /api/stats/:shortUrl

### Respuesta:

```
{
  "shortUrl": "abc123",
  "originalUrl": "https://ejemplo.com",
  "clicks": 42,
  "createdAt": "2024-01-15T10:30:00Z",
  "lastAccessed": "2024-01-15T15:45:00Z"
}
```



## Estructura del Proyecto

```
short-url/
├── src/
│   ├── index.js          # Servidor principal
│   ├── models/
│   │   └── Url.js        # Modelo de base de datos
│   ├── routes/
│   │   └── urls.js       # Rutas de la API
│   ├── middleware/
│   │   └── validation.js # Middleware de validación
│   └── frontend/
│       ├── src/
│       │   ├── components/
│       │   │   ├── UrlShortener.tsx
│       │   │   ├── QRDisplay.tsx
│       │   │   └── Statistics.tsx
│       │   ├── App.tsx
│       │   └── App.css
│       ├── public/
│       └── package.json
├── .env.example
├── package.json
└── README.md
```



## Características de la Interfaz

## Diseño Visual

- **Efecto glassmorphism** - Elementos con apariencia de cristal
- **Diseño responsive** - Adaptable a cualquier tamaño de pantalla
- **Paleta de colores moderna** - Colores vibrantes y profesionales
- **Tipografía clara** - Fuentes legibles y atractivas

## Animaciones

- **Fade in** al cargar la página
- **Transiciones suaves** en formularios y botones
- **Efectos hover** en elementos interactivos
- **Fondo animado** con gradientes dinámicos



## Seguridad

El proyecto implementa varias medidas de seguridad:

- **Validación de URLs** - Verifica que las URLs sean válidas antes de procesarlas
- **Sanitización de entradas** - Limpia los datos de entrada para prevenir inyecciones
- **Protección contra XSS** - Escapado de caracteres especiales
- **Manejo seguro de redirecciones** - Previene redirecciones maliciosas



## Opciones de Despliegue

### Backend

- **Heroku** - Plataforma como servicio (PaaS)
- **Railway** - Plataforma moderna para desarrolladores
- **DigitalOcean** - Servidor privado virtual (VPS)
- **AWS** - Amazon Web Services

### Frontend

- **Vercel** - Plataforma optimizada para React
- **Netlify** - Hosting para sitios estáticos
- **GitHub Pages** - Hosting gratuito de GitHub

### Base de Datos

- **MongoDB Atlas** - Base de datos en la nube
- **MongoDB local** - Instalación local

## Configuración de Dominio

1. Registra un dominio personalizado
2. Configura los registros DNS
3. Actualiza las variables de entorno
4. Configura certificados SSL/HTTPS

## Contribuir

Las contribuciones son bienvenidas. Para contribuir:

1. Haz fork del proyecto
2. Crea una rama para tu feature (`git checkout -b feature/nueva-funcionalidad`)
3. Realiza tus cambios y commitea (`git commit -m 'Agregar nueva funcionalidad'`)
4. Push a la rama (`git push origin feature/nueva-funcionalidad`)
5. Abre un Pull Request

## Pautas para Contribuir

- Sigue las convenciones de código existentes
- Escribe tests para nuevas funcionalidades
- Actualiza la documentación según sea necesario
- Describe claramente los cambios en el Pull Request

## Licencia

Este proyecto está bajo la Licencia MIT. Ver el archivo [LICENSE.md](#) para más detalles.

## Roadmap

### Funcionalidades Planificadas

#### Corto Plazo:

- ☐ Autenticación de usuarios
- ☐ Personalización de enlaces cortos
- ☐ Panel de administración básico
- ☐ Estadísticas más detalladas

#### Mediano Plazo:

- [ ] API keys para desarrolladores
- [ ] Soporte para dominios personalizados
- [ ] Integración con servicios de análisis
- [ ] Bulk operations (operaciones masivas)

#### Largo Plazo:

- [ ] Aplicación móvil
- [ ] Integración con redes sociales
- [ ] Sistema de planes premium
- [ ] API pública completa

## Soporte

Si encuentras algún problema o tienes alguna sugerencia:

- **Issues:** Abre un issue en el repositorio de GitHub
- **Email:** [contacto@ejemplo.com](mailto:contacto@ejemplo.com)
- **Documentación:** Consulta la wiki del proyecto

## Solución de Problemas

### Problemas Comunes

#### Error de conexión a MongoDB:

- Verifica que MongoDB esté ejecutándose
- Comprueba la URL de conexión en el archivo `.env`

#### Puerto ya en uso:

- Cambia el puerto en las variables de entorno
- Verifica que no haya otras aplicaciones usando el puerto

#### Dependencias no encontradas:

- Ejecuta `npm install` en la raíz y en la carpeta frontend
- Verifica que tengas la versión correcta de Node.js