



StatuApp — README (Case Study)

Proyecto cultural de Rosario (AR): catálogo digital de estatuas y monumentos con **QR + mapa + fichas** (autor, año, material, barrio, fotos, fuentes). Diseñado para educación y turismo local.



Resumen

- **Problema:** la información sobre esculturas y monumentos de Rosario está dispersa, desactualizada o difícil de encontrar en sitio.
- **Solución:** una web/app ligera que permite **escanear un QR** en la vía pública y abrir una **ficha completa**, con **galería, mapa, bibliografía** y contenido curado. También navegable por **barrio, autor y material**.
- **Beneficiarios:** escuelas, turistas, guías y ciudadanía.

Demo: pendiente → <https://statuapp.vercel.app>

Repositorio: pendiente → <https://github.com/bereail/statuapp>

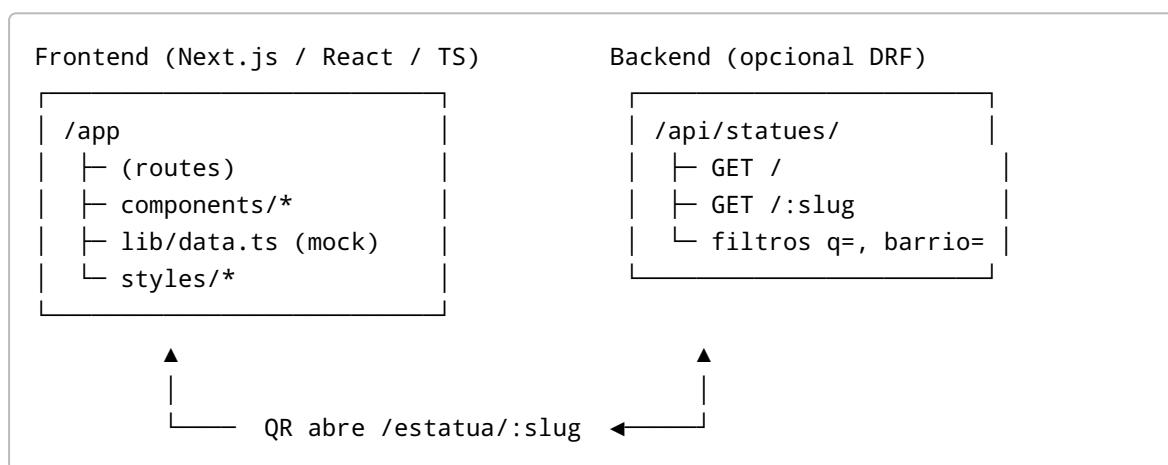


Features clave

- Lectura de **QR** → abre `/estatua/:slug`
- **Mapa** interactivo con marcadores por barrio
- Filtros por **barrio / autor / material**
- Fichas con **medios (fotos/videos), descripción, ubicación, fuentes**
- **Diseño mobile-first**; accesible (etiquetas alt, contraste, foco)
- Arquitectura preparada para **datos locales o API REST**



Arquitectura (vista general)



Stack sugerido

- **Frontend:** Next.js (App Router) + TypeScript + Tailwind + shadcn/ui (opcional) + Leaflet/Mapbox -

Backend (opcional): Django REST Framework + SQLite/MySQL - **Infra:** Vercel (frontend) + Render/Fly.io (backend)



Estructura de datos (TS)

```
export interface Media { id: string; type: 'image'|'video'; url: string;
caption?: string }
export interface Author { id: string; name: string; years?: string }
export interface Location { lat: number; lng: number; barrio: string;
address?: string }
export interface StatueDetail {
  slug: string; title: string; year?: string; material?: string;
  author?: Author; location: Location; medias: Media[]; description?:
  string; sources?: string[];
}
```



Screenshots (añadir)

- Incluí al menos 4 imágenes en `/assets` y referéncialas en el README:
1. Home con listado + filtros
 2. Ficha de estatua (galería, mapa)
 3. Vista de mapa con marcadores
 4. Flujo QR → pantalla de detalle

```
![Home](./assets/home.png)
![Detalle](./assets/detail.png)
![Mapa](./assets/map.png)
![QR](./assets/qr.png)
```



Configuración / Local

Opción A — Solo Frontend (datos en `lib/data.ts`)

```
# 1) Crear proyecto
npx create-next-app@latest statuapp --ts --eslint --tailwind --app
cd statuapp

# 2) Instalar dependencias opcionales
npm i leaflet react-leaflet

# 3) Ejecutar
npm run dev
```

Opción B — Fullstack con DRF

- Backend: `Django==4.x`, `djangorestframework`, CORS
- Endpoints:
 - `GET /api/statues` (listado, filtros `?barrio=&autor=&material=`)
 - `GET /api/statues/:slug` (detalle)
- Frontend consume `NEXT_PUBLIC_API_URL`

`.env.local`

```
NEXT_PUBLIC_API_URL=http://localhost:8000
```

Pruebas (sugeridas)

- **Frontend:** `vitest` + `@testing-library/react`
- **Backend:** tests DRF para endpoints y filtros

Deploy

Frontend (Vercel) 1. Subí el repo a GitHub: `statuapp` 2. En Vercel → `New Project` → importa el repo 3. Variables (si usa API): `NEXT_PUBLIC_API_URL` 4. Deploy → URL pública `https://statuapp.vercel.app`

Backend (Render/Fly.io) 1. Crear servicio Web (Gunicorn)
2. Setear `ALLOWED_HOSTS`, `CORS`, `DATABASE_URL`

Roadmap

- [] Búsqueda por texto y filtros combinados
- [] Lazy-load de imágenes + compresión
- [] Roles (curador / editor)
- [] Export PDF de fichas para escuelas
- [] Modo offline (PWA) para escaneo sin datos

Autora

Berenice Solohaga (bereail) — Full-Stack Developer
LinkedIn | GitHub | Email

Landing pública — contenido y código base

Objetivo: una página liviana y elegante para entrevistas, con CTA y links al repo + demo.

1) Estructura propuesta (secciones)

- **Hero:** título, subtítulo, CTA “Ver demo” y “Ver código”
- **Qué resuelve:** problema/solución en 3 bullets
- **Cómo funciona:** QR → ficha → mapa
- **Tecnologías:** logos o badges
- **Capturas:** 3–4 imágenes
- **CTA final:** “Probar ahora” + contacto

2) Comandos iniciales

```
npx create-next-app@latest statuapp-landing --ts --tailwind --eslint --app  
cd statuapp-landing  
npm i lucide-react  
npm run dev
```

3) `app/layout.tsx` (SEO básico)

```
export const metadata = {  
  title: 'StatuApp – Rosario',  
  description:  
  'Catálogo digital de estatuas y monumentos de Rosario con QR + mapa.',  
  openGraph: { title: 'StatuApp', description: 'Catálogo con QR + mapa',  
    url: 'https://statuapp.vercel.app', type: 'website' }  
};  
export default function RootLayout({ children }: { children:  
  React.ReactNode }) {  
  return (  
    <html lang="es">  
      <body className="min-h-screen bg-neutral-950 text-neutral-100  
antialiased">{children}</body>  
    </html>  
  );  
}
```

4) `app/page.tsx` (Landing simple)

```
import { ArrowRight } from 'lucide-react'  
  
export default function Page() {  
  return (  
    <main className="mx-auto max-w-5xl px-6 py-16">
```

```

    {/* Hero */}
    <section className="text-center space-y-6">
      <h1 className="text-4xl md:text-6xl font-bold">StatuApp</h1>
      <p className="text-lg md:text-xl text-neutral-300">
        Catálogo digital de estatuas y monumentos de Rosario con
      <strong>QR</strong> + <strong>mapa</strong> + fichas.
      </p>
      <div className="flex items-center justify-center gap-3">
        <a href="#capturas" className="inline-flex items-center gap-2
rounded-xl px-5 py-3 bg-white text-black font-medium">
          Ver capturas <ArrowRight size={18} />
        </a>
        <a href="https://github.com/bereail/statuapp" className="inline-
flex items-center gap-2 rounded-xl px-5 py-3 border border-neutral-700">
          Ver código
        </a>
      </div>
    </section>

    {/* Qué resuelve */}
    <section className="mt-16 grid md:grid-cols-3 gap-6" id="que-resuelve">
      {[{
        t:'Información curada', d:'Fichas con autor, año, material, barrio
y fuentes verificadas.'
      },{
        t:'Acceso universal', d:'Escaneá un QR y abrí la ficha en segundos
(mobile-first).'
      },{
        t:'Aprendizaje', d:'Pensado para escuelas, guías y turismo local.'
      }].map((c,i)=>
        <div key={i} className="rounded-2xl border border-neutral-800 p-6">
          <h3 className="text-xl font-semibold mb-2">{c.t}</h3>
          <p className="text-neutral-400">{c.d}</p>
        </div>
      )))
    </section>

    {/* Cómo funciona */}
    <section className="mt-16">
      <h2 className="text-2xl font-semibold mb-4">Cómo funciona</h2>
      <ol className="list-decimal pl-6 space-y-2 text-neutral-300">
        <li>Escaneás un QR ubicado junto a la obra.</li>
        <li>Se abre la ficha con galería, historia y ubicación exacta.</li>
        <li>Podés explorar en el mapa por barrios, autores y materiales.</li>
      </ol>
    </section>

    {/* Capturas */}
    <section id="capturas" className="mt-16">
      <h2 className="text-2xl font-semibold mb-4">Capturas</h2>

```

```

<div className="grid md:grid-cols-3 gap-4">
  {[1,2,3].map(i => (
    <div key={i} className="aspect-[4/3] rounded-2xl bg-neutral-900 border border-neutral-800 flex items-center justify-center">
      <span className="text-neutral-500">Imagen {i}</span>
    </div>
  )))
</div>
</section>

/* CTA */
<section className="mt-16 text-center">
  <a href="#">
    Probar demo <ArrowRight size={18} />
  </a>
</section>

<footer className="mt-20 text-center text-neutral-500 text-sm">
  Hecho por Berenice Solohaga □ {new Date().getFullYear()}
</footer>
</main>
)
}

```

5) Tailwind (`tailwind.config.ts` mínimo)

```

import type { Config } from 'tailwindcss'
export default {
  content: ['./app/**/*.{ts,tsx}', './components/**/*.{ts,tsx}'],
  theme: { extend: {} },
  plugins: [],
} satisfies Config

```

6) Deploy a Vercel (landing)

1. Subí el repo `statuapp-landing` a GitHub
2. Vercel → `New Project` → importa el repo
3. Build por defecto (Next.js)
4. Listo → `https://statuapp.vercel.app` (o similar)



Checklist de entrega para LinkedIn/entrevistas

- [] README “case study” con problema/solución, capturas y demo
- [] Landing pública con CTA y correo visible
- [] Repo limpio (sin `node_modules`, sin secretos, `LICENSE`, `CODE_OF_CONDUCT` opcional)

- [] Issues etiquetadas: `good first issue`, `enhancement`, `bug`
 - [] Proyecto en Vercel + badge en README
-

Snippets útiles (README)

```
[![Vercel](https://img.shields.io/badge/Deploy-Vercel-black)](https://statuapp.vercel.app)
[![Tech](https://img.shields.io/badge/Next.js-14-000)](#)
[![License](https://img.shields.io/badge/License-MIT-green)](./LICENSE)
```

Fuentes / Bibliografía

- Museo Castagnino+Macro – Catálogo en línea
- Archivo Municipal – Nomencladores de calles
- Crónicas periodísticas (La Capital, etc.)

Tip: generá los **QR** con `slug` por obra (ej. `/estatua/monumento-leandro-alem`) y pegá un sticker con URL corta (`sttu.app/alem`).