perplexity

▯ SharkFlow - ARCHIVOS COMPLETOS (Copia/Pega Listo)

Estructura completa + Netlify ready. Copia cada archivo en su ruta.

```
sharkflow-netlify/
├── package.json
├── next.config.js
├── tailwind.config.js
├── postcss.config.js
├── public/
│   └── manifest.json
└── src/
    ├── app/
    │   ├── globals.css
    │   ├── layout.tsx
    │   ├── page.tsx
    │   └── calculadora/
    │       └── page.tsx
    └── components/
        ├── SliderCompetitivo.tsx
        └── ui/
            ├── button.tsx
            ├── slider.tsx
            └── card.tsx
```

1. package.json

```
{
  "name": "sharkflow-netlify",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "next": "14.2.5",
    "react": "^18",
    "react-dom": "^18",
    "lucide-react": "^0.394.0",
    "next-pwa": "^5.6.0",
    "chart.js": "^4.4.2",
    "react-chartjs-2": "^5.2.0",
    "class-variance-authority": "^0.7.0",
    "clsx": "^2.1.0",
    "tailwind-merge": "^2.2.2"
  },
  "devDependencies": {
    "@types/node": "^20",
    "@types/react": "^18",
    "@types/react-dom": "^18",
    "autoprefixer": "^10.4.17",
    "eslint": "^8",
    "eslint-config-next": "14.2.5",
    "postcss": "^8.4.35",
    "tailwindcss": "^3.4.1",
    "typescript": "^5"
  }
}
```

2. next.config.js

```
const withPWA = require('next-pwa')({
  dest: 'public',
  register: true,
  skipWaiting: true,
  runtimeCaching: [{
    urlPattern: /^https?.*/,
    handler: 'NetworkFirst',
    options: { cacheName: 'offlineCache' }
  }]
})

module.exports = withPWA({
  experimental: { appDir: true },
  images: { remotePatterns: [{ protocol: 'https', hostname: '**' }] }
})
```

3. tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  darkMode: 'class',
  content: [
    './src/pages/**/*.{js,ts,jsx,tsx,mdx}',
    './src/components/**/*.{js,ts,jsx,tsx,mdx}',
    './src/app/**/*.{js,ts,jsx,tsx,mdx}',
  ],
  theme: {
    extend: {
      colors: {
        emerald: {
          50: '#ecfdf5', 500: '#10b981', 600: '#059669'
        }
      }
    },
  },
  plugins: [require('tailwindcss-animate')],
}
```

4. postcss.config.js

```
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
}
```

5. public/manifest.json

```
{
  "name": "SharkFlow - Calculadora Broker",
  "short_name": "SharkFlow",
  "description": "Análisis fundamental intuitivo",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#ecfdf5",
  "theme_color": "#10b981",
  "icons": [
    {
      "src": "data:image/svg+xml;base64,PHN2ZyB3aWR0aD0iMTkyIiBoZWlnaHQ9IjE5MiIgdmlld0Jve
      "sizes": "192x192",
      "type": "image/svg+xml"
    }
  ]
}
```

6. src/app/globals.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  :root {
    --background: 0 0% 100%;
  }
}
```

7. src/app/layout.tsx

```
import type { Metadata } from 'next'
import { Inter } from 'next/font/google'
import './globals.css'
import Disclaimer from '@/components/Disclaimer'

const inter = Inter({ subsets: ['latin'] })

export const metadata: Metadata = {
  title: 'SharkFlow - Calculadora Broker',
  description: 'Análisis fundamental intuitivo',
  manifest: '/manifest.json'
}

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html lang="es">
      <body className={inter.className}>{children}</body>
    </html>
  )
}
```

8. src/app/page.tsx (Landing)

```
export default function Home() {
  return (
    <main className="min-h-screen bg-gradient-to-br from-emerald-50 to-blue-50 flex flex-
      {/* Código landing del mensaje anterior */}
      <div className="max-w-4xl mx-auto space-y-8">
        <h1 className="text-5xl md:text-7xl font-black bg-gradient-to-r from-emerald-600
          ▯ SharkFlow
        </h1>
        <p className="text-xl md:text-2xl text-gray-700 max-w-2xl mx-auto leading-relaxed
          Calculadora de inversión intuitiva. 4 pasos simples para analizar cualquier emp
        </p>
        <a href="/calculadora" className="px-12 py-6 bg-gradient-to-r from-emerald-600 to
          ▯ Empezar Análisis
        </a>
      </div>
    </main>
  )
}
```

9. src/app/calculadora/page.tsx (App principal)

```
'use client'
import { useState } from 'react'
import SliderCompetitivo from '@/components/SliderCompetitivo'
import { Button } from '@/components/ui/button'
import { Card, CardContent, CardHeader, CardTitle } from '@/components/ui/card'
import { Progress } from '@/components/ui/progress'

export default function Calculadora() {
  const [datos, setDatos] = useState({
    ticker: 'AAPL',
    sector: 'Tecnología',
    ventajas: {costes: 3, cambio: 2, red: 4, intangibles: 5}
  })

  const score = Object.values(datos.ventajas).reduce((a,b)=>a+b,0) / 20 * 100

  return (
    <div className="min-h-screen bg-gradient-to-br from-slate-50 to-emerald-50 py-12 px-4
      <div className="max-w-2xl mx-auto space-y-8">
        <div className="text-center">
          <h1 className="text-4xl font-bold bg-gradient-to-r from-emerald-600 to-blue-600
            Análisis {datos.ticker}
          </h1>
          <Progress value={score} className="w-full h-4 mb-4 mx-auto max-w-md" />
          <div className="text-3xl font-black text-gray-800">
            {Math.round(score)} <span className="text-emerald-600">/ 100</span>
          </div>
        </div>

        <Card className="border-emerald-200 shadow-2xl">
          <CardHeader>
            <CardTitle className="text-2xl">□ Ventajas Competitivas</CardTitle>
          </CardHeader>
          <CardContent className="space-y-6">
            <SliderCompetitivo
              label="Ventajas costes"
              value={datos.ventajas.costes}
              onChange={(v) => setDatos({
                ...datos,
                ventajas: {...datos.ventajas, costes: v}
              })}
              tooltip="Economías de escala, tecnología eficiente"
            />
            <SliderCompetitivo
              label="Costes cambio"
              value={datos.ventajas.cambio}
              onChange={(v) => setDatos({
                ...datos,
                ventajas: {...datos.ventajas, cambio: v}
              })}
              tooltip="Clientes atados por contratos/ecosistema"
            />
            <SliderCompetitivo
              label="Efectos red"
              value={datos.ventajas.red}
              onChange={(v) => setDatos({
                ...datos,
                ventajas: {...datos.ventajas, red:
```