

Contents

El Guajiro de Hialeah - Código Completo de la Aplicación	1
Índice	1
1. Configuración del Proyecto	2
package.json	2
tsconfig.json	4
next.config.js	5
tailwind.config.ts	6
postcss.config.js	7
2. Base de Datos	7
schema.prisma	7
3. Aplicación Principal	9
app/layout.tsx	9
app/page.tsx	10
app/globals.css	11
4. Componentes Core	13
components/hero-section.tsx	13
components/main-app.tsx	15
components/navigation-tabs.tsx	17
components/processing-section.tsx	18
components/decimals-section.tsx	24
components/analysis-section.tsx	28
components/export-section.tsx	34
components/education-section.tsx (SOLO INICIO - 100 líneas)	40
components/education-section.tsx (primeras 100 líneas - archivo muy extenso)	55
components/about-modal.tsx	57
components/prompt-modal.tsx	61
components/tribute-section.tsx	64
5. API Routes	65
api/validate-youtube/route.ts	65
api/process-video/route.ts	67
api/video/[id]/decimas/route.ts	71
api/video/[id]/analysis/route.ts	72
api/video/[id]/export/text/route.ts	74
api/video/[id]/export/pdf/route.ts	77
api/video/[id]/share/route.ts	78
6. Librerías y Utilidades	79
lib/db.ts	79
lib/utils.ts	79
lib/syllable-counter.ts	88
lib/rhyme-validator.ts	93
lib/openrouter-client.ts	98
Notas Importantes	103

El Guajiro de Hialeah - Código Completo de la Aplicación

Aplicación de análisis de décimas cubanas con transcripción de YouTube

Índice

1. Configuración del Proyecto

2. Base de Datos
 3. Aplicación Principal
 4. Componentes Core
 5. API Routes
 6. Librerías y Utilidades
 7. Estilos Globales
-

1. Configuración del Proyecto

package.json

```
{  
  "name": "app",  
  "private": true,  
  "scripts": {  
    "dev": "next dev",  
    "build": "next build",  
    "start": "next start",  
    "lint": "next lint"  
  },  
  "prisma": {  
    "seed": "tsx --require dotenv/config scripts/seed.ts"  
  },  
  "devDependencies": {  
    "@next/swc-wasm-nodejs": "13.5.1",  
    "@types/file-saver": "^2",  
    "@types/node": "20.6.2",  
    "@types/react": "18.2.22",  
    "@types/react-dom": "18.2.7",  
    "@typescript-eslint/eslint-plugin": "7.0.0",  
    "@typescript-eslint/parser": "7.0.0",  
    "eslint": "9.24.0",  
    "eslint-config-next": "15.3.0",  
    "eslint-plugin-prettier": "5.1.3",  
    "eslint-plugin-react-hooks": "4.6.0",  
    "picocolors": "^1.1.1",  
    "postcss": "8.4.30",  
    "prisma": "6.7.0",  
    "tailwind-merge": "2.5.2",  
    "tailwindcss": "3.3.3",  
    "tailwindcss-animate": "1.0.7",  
    "ts-node": "10.9.2",  
    "tsx": "4.20.3",  
    "typescript": "5.2.2"  
},  
  "dependencies": {  
    "@aws-sdk/client-s3": "^3.0.0",  
    "@aws-sdk/s3-request-presigner": "^3.0.0",  
    "@floating-ui/react": "0.26.0",  

```

```
"@radix-ui/react-accordion": "1.2.0",
"@radix-ui/react-alert-dialog": "1.1.1",
"@radix-ui/react-aspect-ratio": "1.1.0",
"@radix-ui/react-avatar": "1.1.0",
"@radix-ui/react-checkbox": "1.1.1",
"@radix-ui/react-collapsible": "1.1.0",
"@radix-ui/react-context-menu": "2.2.1",
"@radix-ui/react-dialog": "1.1.1",
"@radix-ui/react-dropdown-menu": "2.1.1",
"@radix-ui/react-hover-card": "1.1.1",
"@radix-ui/react-label": "2.1.0",
"@radix-ui/react-menubar": "1.1.1",
"@radix-ui/react-navigation-menu": "1.2.0",
"@radix-ui/react-popover": "1.1.1",
"@radix-ui/react-progress": "1.1.0",
"@radix-ui/react-radio-group": "1.2.0",
"@radix-ui/react-scroll-area": "1.1.0",
"@radix-ui/react-select": "2.1.1",
"@radix-ui/react-separator": "1.1.0",
"@radix-ui/react-slider": "1.2.0",
"@radix-ui/react-slot": "1.1.0",
"@radix-ui/react-switch": "1.1.0",
"@radix-ui/react-tabs": "1.1.0",
"@radix-ui/react-toast": "1.2.1",
"@radix-ui/react-toggle": "1.1.0",
"@radix-ui/react-toggle-group": "1.1.0",
"@radix-ui/react-tooltip": "1.1.2",
"@tanstack/react-query": "5.0.0",
"@types/bcryptjs": "2.4.6",
"@types/jsonwebtoken": "9.0.5",
"@types/plotly.js": "2.35.5",
"@types/react-plotly.js": "2.6.3",
"autoprefixer": "10.4.15",
"bcryptjs": "2.4.3",
"chart.js": "4.4.9",
"class-variance-authority": "0.7.0",
"clsx": "2.1.1",
"cmdk": "1.0.0",
"cookie": "1.0.2",
"csv": "6.3.11",
"date-fns": "3.6.0",
"dayjs": "1.11.13",
"dotenv": "16.5.0",
"embla-carousel-react": "8.3.0",
"file-saver": "^2.0.5",
"form-data": "^4.0.4",
"formik": "2.4.5",
"framer-motion": "10.18.0",
"gray-matter": "4.0.3",
"html2canvas": "^1.4.1",
"input-otp": "1.2.4",
"jotai": "2.6.0",
"jsonwebtoken": "9.0.2",
"jspdf": "^3.0.3",
```

```

    "lodash": "4.17.21",
    "lucide-react": "0.446.0",
    "mammoth": "^1.11.0",
    "mapbox-gl": "1.13.3",
    "next": "14.2.28",
    "next-auth": "4.24.11",
    "next-themes": "0.3.0",
    "openai": "^6.9.0",
    "plotly.js": "2.35.3",
    "react": "18.2.0",
    "react-chartjs-2": "5.3.0",
    "react-datepicker": "6.1.0",
    "react-day-picker": "8.10.1",
    "react-dom": "18.2.0",
    "react-hook-form": "7.53.0",
    "react-hot-toast": "2.4.1",
    "react-intersection-observer": "9.8.0",
    "react-is": "18.3.1",
    "react-plotly.js": "2.6.0",
    "react-resizable-panels": "2.1.3",
    "react-select": "5.8.0",
    "react-use": "17.6.0",
    "react-youtube": "10.1.0",
    "recharts": "2.15.3",
    "sonner": "1.5.0",
    "swr": "2.2.4",
    "tailwind-scrollbar-hide": "1.1.7",
    "vaul": "0.9.9",
    "webpack": "5.99.5",
    "yup": "1.3.0",
    "zod": "3.23.8",
    "zustand": "5.0.3"
  },
  "browserslist": [
    "ie >= 11",
    "> 0.5%",
    "last 2 versions",
    "not dead"
  ]
}

```

tsconfig.json

```
{
  "compilerOptions": {
    "target": "es2020",
    "lib": [
      "dom",
      "dom.iterable",
      "es5",
      "es2020"
    ],
    "allowJs": true,
    "skipLibCheck": true,
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "noImplicitAny": true,
    "noImplicitThis": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "noFallthroughCasesInSwitch": true
  }
}
```

```

    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "module": "esnext",
    "moduleResolution": "bundler",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "jsx": "preserve",
    "incremental": true,
    "plugins": [
      {
        "name": "next"
      }
    ],
    "paths": {
      "@/*": [
        "./*"
      ]
    }
  },
  "include": [
    "next-env.d.ts",
    "**/*.ts",
    "**/*.tsx",
    ".next/types/**/*.ts",
    ".build/types/**/*.ts"
  ],
  "exclude": [
    "node_modules"
  ]
}

```

next.config.js

```

const path = require('path');

/** @type {import('next').NextConfig} */
const nextConfig = {
  distDir: process.env.NEXT_DIST_DIR || '.next',
  output: process.env.NEXT_OUTPUT_MODE,
  experimental: {
    outputFileTracingRoot: path.join(__dirname, '../'),
  },
  eslint: {
    ignoreDuringBuilds: true,
  },
  typescript: {
    ignoreBuildErrors: false,
  },
  images: { unoptimized: true },
};

module.exports = nextConfig;

```

tailwind.config.ts

```
import type { Config } from 'tailwindcss';

const config: Config = {
  darkMode: ['class'],
  content: [
    './pages/**/*.{js,ts,jsx,tsx,mdx}',
    './components/**/*.{js,ts,jsx,tsx,mdx}',
    './app/**/*.{js,ts,jsx,tsx,mdx}',
  ],
  theme: {
    extend: {
      backgroundImage: {
        'gradient-radial': 'radial-gradient(var(--tw-gradient-stops))',
        'gradient-conic': 'conic-gradient(from 180deg at 50% 50%, var(--tw-gradient-stops))',
      },
      borderRadius: {
        lg: 'var(--radius)',
        md: 'calc(var(--radius) - 2px)',
        sm: 'calc(var(--radius) - 4px)',
      },
      colors: {
        background: 'hsl(var(--background))',
        foreground: 'hsl(var(--foreground))',
        card: {
          DEFAULT: 'hsl(var(--card))',
          foreground: 'hsl(var(--card-foreground))',
        },
        popover: {
          DEFAULT: 'hsl(var(--popover))',
          foreground: 'hsl(var(--popover-foreground))',
        },
        primary: {
          DEFAULT: 'hsl(var(--primary))',
          foreground: 'hsl(var(--primary-foreground))',
        },
        secondary: {
          DEFAULT: 'hsl(var(--secondary))',
          foreground: 'hsl(var(--secondary-foreground))',
        },
        muted: {
          DEFAULT: 'hsl(var(--muted))',
          foreground: 'hsl(var(--muted-foreground))',
        },
        accent: {
          DEFAULT: 'hsl(var(--accent))',
          foreground: 'hsl(var(--accent-foreground))',
        },
        destructive: {
          DEFAULT: 'hsl(var(--destructive))',
          foreground: 'hsl(var(--destructive-foreground))',
        },
      },
    },
  },
}
```

```

border: 'hsl(var(--border))',
input: 'hsl(var(--input))',
ring: 'hsl(var(--ring))',
chart: {
  '1': 'hsl(var(--chart-1))',
  '2': 'hsl(var(--chart-2))',
  '3': 'hsl(var(--chart-3))',
  '4': 'hsl(var(--chart-4))',
  '5': 'hsl(var(--chart-5))',
},
},
keyframes: {
  'accordion-down': {
    from: {
      height: '0',
    },
    to: {
      height: 'var(--radix-accordion-content-height)',
    },
  },
  'accordion-up': {
    from: {
      height: 'var(--radix-accordion-content-height)',
    },
    to: {
      height: '0',
    },
  },
},
animation: {
  'accordion-down': 'accordion-down 0.2s ease-out',
  'accordion-up': 'accordion-up 0.2s ease-out',
},
},
},
},
plugins: [require('tailwindcss-animate')],  

};

export default config;

```

postcss.config.js

```

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

```

2. Base de Datos

schema.prisma

```

generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]

```

```

        output = "/home/ubuntu/el_guajiro_de_hialeah/nextjs_space/node_modules/.prisma/client"
    }

datasource db {
    provider = "postgresql"
    url      = env("DATABASE_URL")
}

model Video {
    id          String  @id @default(cuid())
    youtubeUrl String  @unique
    youtubeId   String  @unique
    title        String
    duration     String
    thumbnailUrl String?
    audioPath    String?
    transcript   String?
    processedAt  DateTime @default(now())
    processingTime Int? // in seconds
    status       ProcessingStatus @default(PENDING)

    decimas      Decima[]
    poets        Poet[]
    analysis     Analysis?

    @@map("videos")
}

model Poet {
    id          String  @id @default(cuid())
    name        String
    videoId    String
    turnNumber  Int?

    video      Video   @relation(fields: [videoId], references: [id], onDelete: Cascade)
    decimas    Decima[]

    @@unique([videoId, name])
    @@map("poets")
}

model Decima {
    id          String  @id @default(cuid())
    number      Int
    lines       String[] // Array of 10 lines
    rhymeScheme String  @default("ABBAACCDCC")
    meter       String  @default("Octosilabos")
    isComplete  Boolean @default(true)
    isRegular   Boolean @default(true)
    timestamp   String? // mm:ss - mm:ss

    videoId    String
    poetId     String
}

```

```

video          Video    @relation(fields: [videoId], references: [id], onDelete: Cascade)
poet          Poet     @relation(fields: [poetId], references: [id], onDelete: Cascade)

// Analysis fields
technicalMastery   String?
poeticDevices      String?
culturalElements   String?
emotionalResonance String?
improvisation      String?
topThreeReason     String?
isTopThree         Boolean @default(false)
topThreeRank       Int?

@@map("decimas")
}

model Analysis {
  id                  String  @id @default(cuid())
  videoId            String  @unique

  // Summary fields
  summaryThemes       String?
  summaryTone         String?
  summaryInteraction  String?
  summaryQuality      String?

  // Deep analysis fields
  recurringThemes    String?
  regionalisms        String?
  linguisticSophistication String?
  improvisationTechniques String?
  canonComparison     String?
  culturalSignificance String?

  video              Video    @relation(fields: [videoId], references: [id], onDelete: Cascade)

  @@map("analyses")
}

enum ProcessingStatus {
  PENDING
  DOWNLOADING
  TRANSCRIBING
  ANALYZING
  COMPLETED
  ERROR
}

```

3. Aplicación Principal

app/layout.tsx

```
export const dynamic = "force-dynamic";
```

```

import type { Metadata } from "next";
import { Inter } from "next/font/google";
import "./globals.css";

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

export const metadata: Metadata = {
  title: "El Guajiro de Hialeah | Transcriptor y Analizador de Décima Espinela Cubana",
  description: "Un algoritmo que siente la pasión del poeta, rescata del olvido cada improvisación, pre",
  openGraph: {
    title: "El Guajiro de Hialeah | Transcriptor y Analizador de Décima Espinela Cubana",
    description: "Un algoritmo que siente la pasión del poeta, rescata del olvido cada improvisación, p",
    images: [
      {
        url: "/og-image.png",
        width: 1200,
        height: 630,
        alt: "El Guajiro de Hialeah",
      },
    ],
  },
  icons: {
    icon: "/favicon.svg",
    shortcut: "/favicon.svg",
  },
};

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="es" suppressHydrationWarning>
      <head>
        <link rel="preconnect" href="https://fonts.googleapis.com" />
        <link rel="preconnect" href="https://fonts.gstatic.com" crossOrigin="" />
        <link
          href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;700&display=swap"
          rel="stylesheet"
        />
      </head>
      <body className={`${inter.className} antialiased bg-[#F5E6D3] text-[#5C4033]`} suppressHydrationW
        {children}
      </body>
    </html>
  );
}

```

app/page.tsx

```

import Header from "@components/header";
import HeroSection from "@components/hero-section";

```

```

import MainApp from "@/components/main-app";

export default function Home() {
  return (
    <>
      <Header />
      <main className="min-h-screen vintage-background">
        <HeroSection />
        <MainApp />
      </main>
    </>
  );
}

```

app/globals.css

```

@import url('https://fonts.googleapis.com/css2?family=Playfair+Display:ital,wght@0,400;0,600;0,700;1,400');

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

@layer base {
  :root {
    --primary: #C8A05C;
    --secondary: #5C4033;
    --background: #F5E6D3;
    --button: #D2691E;
    --accent: #8B4513;
  }

  * {
    @apply border-border;
  }
}

body {
  @apply bg-[#F5E6D3] text-[#5C4033];
  font-feature-settings: "rlig" 1, "calt" 1;
}

.vintage-background {
  background: linear-gradient(135deg, #F5E6D3 0%, #E6D7C1 50%, #DDD0B8 100%);
}

.sepia-effect {
  filter: sepia(0.3) contrast(1.1) brightness(1.05);
}

.gold-gradient {
  background: linear-gradient(135deg, #C8A05C 0%, #D4AF37 50%, #B8860B 100%);
}

.palm-pattern::before {
  content: " ";
}

```

```

    position: absolute;
    opacity: 0.1;
    font-size: 2rem;
}

.scroll-smooth {
    scroll-behavior: smooth;
}

/* Cuban vintage styling */
.cuban-card {
    @apply bg-white/90 backdrop-blur-sm border-2 border-[#C8A05C]/20 rounded-lg shadow-lg;
}

.cuban-button {
    @apply bg-[#D2691E] text-white font-semibold px-6 py-3 rounded-lg shadow-md hover:bg-[#B8860B] transition-all duration-500 ease-out;
}

.decima-text {
    font-family: 'Playfair Display', serif;
    @apply text-[#5C4033] leading-relaxed;
}

.nav-tab {
    @apply px-4 py-2 rounded-t-lg font-medium text-[#5C4033] hover:bg-[#C8A05C]/20 transition-colors duration-500 ease-out;
}

.nav-tab.active {
    @apply bg-[#C8A05C] text-white;
}

.progress-bar {
    @apply w-full bg-[#E6D7C1] rounded-full h-3 overflow-hidden;
}

.progress-bar-fill {
    @apply bg-gradient-to-r from-[#C8A05C] to-[#D2691E] h-full transition-all duration-500 ease-out;
}
}

/* Custom scrollbar */
::-webkit-scrollbar {
    width: 8px;
}

::-webkit-scrollbar-track {
    background: #F5E6D3;
}

::-webkit-scrollbar-thumb {
    background: #C8A05C;
    border-radius: 4px;
}

```

```

::-webkit-scrollbar-thumb:hover {
  background: #D2691E;
}

/* Animation keyframes */
@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(30px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

@keyframes countUp {
  from {
    transform: scale(0.8);
    opacity: 0;
  }
  to {
    transform: scale(1);
    opacity: 1;
  }
}

.fade-in-up {
  animation: fadeInUp 0.8s ease-out forwards;
}

.count-up {
  animation: countUp 0.6s ease-out forwards;
}

/* Parallax scroll effect */
.parallax-bg {
  background-attachment: fixed;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

@media (max-width: 768px) {
  .parallax-bg {
    background-attachment: scroll;
  }
}

```

4. Componentes Core

components/hero-section.tsx

```
"use client";
```

```

import Image from "next/image";
import { Badge } from "@/components/ui/badge";
import { useState } from "react";
import AboutModal from "@/components/about-modal";

export default function HeroSection() {
  const [showAboutModal, setShowAboutModal] = useState(false);
  return (
    <section className="relative min-h-screen flex items-end justify-center overflow-hidden pt-20">
      {/* Background Image - Full Color */}
      <div className="absolute inset-0">
        <Image
          src="/calixto-gonzalez-hero.jpg"
          alt="Calixto González en el puerto de Hialeah"
          fill
          className="object-cover"
          priority
        />
      </div>

      {/* Bottom Gradient Overlay - Only at bottom for text */}
      <div className="absolute bottom-0 left-0 right-0 h-2/3 bg-gradient-to-t from-[#F5E6D3]/95 via-[#F5E6D3]/100% to-[#F5E6D3]/100%">

        {/* Content - Positioned at bottom */}
        <div className="relative z-20 w-full max-w-4xl mx-auto px-4 pb-16">
          <div className="text-center">
            {/* Main Title */}
            <h1 className="text-5xl md:text-7xl font-bold text-[#C8A05C] mb-2 font-serif drop-shadow-lg">
              Calixto González
            </h1>

            {/* Subtitle */}
            <h2 className="text-3xl md:text-4xl font-semibold text-[#D2691E] mb-4 font-serif italic drop-shadow-lg">
              El Guajiro de Hialeah
            </h2>

            {/* Badge */}
            <Badge className="text-xl bg-[#D2691E] text-white px-6 py-2 mb-8 font-semibold">
              EST 1936
            </Badge>

            {/* Tagline */}
            <div className="text-lg md:text-xl text-[#5C4033] italic leading-relaxed max-w-3xl mx-auto mb-8">
              <p className="mb-2">Un algoritmo que siente la pasión del poeta,</p>
              <p className="mb-2">rescata del olvido cada improvisación,</p>
              <p className="mb-2">preserva intacta la lírica del corazón,</p>
              <p>y eterniza el arte vivo de la décima completa.</p>
            </div>

            {/* Attribution */}
            <p className="text-sm text-[#5C4033]">
              © Emilio José Novo | juanantoniodiaz.com |
            </p>
          </div>
        </div>
      </div>
    </section>
  );
}

```

```

        onClick={() => setShowAboutModal(true)}
        className="ml-1 underline hover:text-[#D2691E] transition-colors duration-200 font-medium"
      >
      Acerca de
    </button>
  </p>
</div>
</div>

{/* Scroll Indicator */}
<div className="absolute bottom-8 left-1/2 transform -translate-x-1/2 animate-bounce">
  <div className="w-6 h-10 border-2 border-[#C8A05C] rounded-full flex justify-center">
    <div className="w-1 h-3 bg-[#C8A05C] rounded-full mt-2"></div>
  </div>
</div>

{/* About Modal */}
<AboutModal
  isOpen={showAboutModal}
  onClose={() => setShowAboutModal(false)}
/>
</section>
);
}

```

components/main-app.tsx

```

"use client";

import { useState } from "react";
import TributeSection from "@/components/tribute-section";
import NavigationTabs from "@/components/navigation-tabs";
import ProcessingSection from "@/components/processing-section";
import ProcessingSectionRefactored from "@/components/processing-section-refactored";
import DecimalsSection from "@/components/decimals-section";
import AnalysisSection from "@/components/analysis-section";
import ExportSection from "@/components/export-section";
import EducationSection from "@/components/education-section";
import AboutModal from "@/components/about-modal";
import { Button } from "@/components/ui/button";
import { Badge } from "@/components/ui/badge";

type TabType = 'procesar' | 'decimals' | 'analysis' | 'export' | 'education';

export default function MainApp() {
  const [activeTab, setActiveTab] = useState<TabType>('procesar');
  const [showAboutModal, setShowAboutModal] = useState(false);
  const [currentVideoId, setCurrentVideoId] = useState<string | null>(null);
  const [useNewPipeline, setUseNewPipeline] = useState(true); // Default to new pipeline

  const renderActiveSection = () => {
    switch (activeTab) {
      case 'procesar':

```

```

        return useNewPipeline ?
            <ProcessingSectionRefactored onVideoProcessed={setCurrentVideoId} /> :
            <ProcessingSection onVideoProcessed={setCurrentVideoId} />;
    case 'decimals':
        return <DecimalsSection videoId={currentVideoId} />;
    case 'analysis':
        return <AnalysisSection videoId={currentVideoId} />;
    case 'export':
        return <ExportSection videoId={currentVideoId} />;
    case 'education':
        return <EducationSection />;
    default:
        return useNewPipeline ?
            <ProcessingSectionRefactored onVideoProcessed={setCurrentVideoId} /> :
            <ProcessingSection onVideoProcessed={setCurrentVideoId} />;
}
};

return (
    <div className="min-h-screen bg-gradient-to-br from-[#F5E6D3] to-[#E6D7C1]">
        <TributeSection />

        <div className="max-w-6xl mx-auto px-4 py-8">
            <NavigationTabs
                activeTab={activeTab}
                onTabChange={setActiveTab}
                onAboutClick={() => setShowAboutModal(true)}
            />

            {/* Pipeline Mode Toggle (only show on processing tab) */}
            {activeTab === 'procesar' && (
                <div className="mt-4 flex justify-center">
                    <div className="inline-flex items-center gap-3 bg-white/80 backdrop-blur-sm rounded-full p-2">
                        <Button
                            onClick={() => setUseNewPipeline(false)}
                            variant={!useNewPipeline ? "default" : "ghost"}
                            className={`rounded-full ${!useNewPipeline ? 'bg-[#C8A05C]' : ''}`}
                            size="sm"
                        >
                            Modo Perplexity (Manual)
                        </Button>
                        <Button
                            onClick={() => setUseNewPipeline(true)}
                            variant={useNewPipeline ? "default" : "ghost"}
                            className={`rounded-full ${useNewPipeline ? 'bg-[#D2691E]' : ''}`}
                            size="sm"
                        >
                            Pipeline IA Multi-Etapa
                            <Badge className="ml-2 bg-green-600 text-white">NUEVO</Badge>
                        </Button>
                    </div>
                </div>
            )}
        </div>
    </div>
);

```

```

        <div className="mt-8">
          {renderActiveSection()}
        </div>
      </div>

      <AboutModal
        isOpen={showAboutModal}
        onClose={() => setShowAboutModal(false)}
      />
    </div>
  );
}


```

components/navigation-tabs.tsx

```

"use client";

interface NavigationTabsProps {
  activeTab: string;
  onTabChange: (tab: 'procesar' | 'decimals' | 'analysis' | 'export' | 'education') => void;
  onAboutClick: () => void;
}

export default function NavigationTabs({ activeTab, onTabChange, onAboutClick }: NavigationTabsProps) {
  const tabs = [
    { id: 'procesar', label: 'Procesar Transcripción', icon: '' },
    { id: 'decimals', label: 'Décimas Formateadas', icon: '' },
    { id: 'analysis', label: 'Análisis Completo', icon: '' },
    { id: 'export', label: 'Exportar Resultados', icon: '' },
    { id: 'education', label: 'Educación Histórica', icon: '' }
  ];

  return (
    <div className="bg-white/80 backdrop-blur-sm rounded-2xl p-2 shadow-lg border-2 border-[#C8A05C]/20">
      <div className="flex flex-wrap justify-center gap-2">
        {tabs.map((tab) => (
          <button
            key={tab.id}
            onClick={() => onTabChange(tab.id as any)}
            className={`nav-tab ${activeTab === tab.id ? 'active' : ''} flex items-center gap-2 whitespace-nowrap`}
          >
            <span>{tab.icon}</span>
            <span className="hidden sm:inline">{tab.label}</span>
          </button>
        ))}
        <button
          onClick={onAboutClick}
          className="nav-tab flex items-center gap-2 ml-2 border-l-2 border-[#C8A05C]/30 pl-4">
          <span></span>
          <span className="hidden sm:inline">Acerca de</span>
        </button>
      </div>
    
```

```

        </div>
    );
}

components/processing-section.tsx

"use client";

import { useState, useCallback } from "react";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Card } from "@/components/ui/card";
import { Textarea } from "@/components/ui/textarea";
import { AlertCircle, CheckCircle, Loader2, Play, FileText, Sparkles } from "lucide-react";
import Image from "next/image";
import { PromptModal } from "./prompt-modal";

interface ProcessingSectionProps {
    onVideoProcessed: (videoId: string) => void;
}

interface VideoPreview {
    title: string;
    duration: string;
    thumbnailUrl: string;
    youtubeId: string;
}

type ProcessingStage = 'idle' | 'validating' | 'parsing' | 'completed' | 'error';

export default function ProcessingSection({ onVideoProcessed }: ProcessingSectionProps) {
    const [youtubeUrl, setYoutubeUrl] = useState("");
    const [videoPreview, setVideoPreview] = useState<VideoPreview | null>(null);
    const [perplexityResult, setPerplexityResult] = useState("");
    const [promptModalOpen, setPromptModalOpen] = useState(false);
    const [processingStage, setProcessingStage] = useState<ProcessingStage>('idle');
    const [error, setError] = useState<string>("");

    const validateAndPreviewUrl = useCallback(async (url: string) => {
        if (!url.trim()) return;

        setProcessingStage('validating');
        setError("");

        try {
            const response = await fetch('/api/validate-youtube', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ url: url.trim() })
            });

            const data = await response.json();

            if (!response.ok) {

```

```

        throw new Error(data.error || 'Error al validar el enlace');
    }

    setVideoPreview(data);
    setProcessingStage('idle');
} catch (err) {
    setError(err instanceof Error ? err.message : 'Error al validar el enlace');
    setProcessingStage('error');
    setVideoPreview(null);
}
}, []));

const handleOpenPromptModal = () => {
    if (!youtubeUrl.trim()) {
        setError("Por favor pega primero el enlace de YouTube");
        return;
    }
    setPromptModalOpen(true);
};

const processPerplexityResult = useCallback(async () => {
    if (!videoPreview || !perplexityResult.trim()) {
        setError("Por favor pega el resultado de Perplexity");
        return;
    }

    setProcessingStage('parsing');
    setError("");

    try {
        const response = await fetch('/api/process-video', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({
                youtubeUrl: youtubeUrl.trim(),
                youtubeId: videoPreview.youtubeId,
                title: videoPreview.title,
                duration: videoPreview.duration,
                thumbnailUrl: videoPreview.thumbnailUrl,
                perplexityResult: perplexityResult.trim()
            })
        });

        const data = await response.json();

        if (!response.ok) {
            const errorMsg = data.error || 'Error al procesar el resultado';
            setError(errorMsg);
            throw new Error(errorMsg);
        }

        // Mostrar warnings si existen
        if (data.warnings && data.warnings.length > 0) {
            console.log('Advertencias del parser:', data.warnings);
        }
    }
});

```

```

    }

    setProcessingStage('completed');
    setTimeout(() => onVideoProcessed(data.videoId), 1500);
} catch (err) {
    setError(err instanceof Error ? err.message : 'Error durante el procesamiento');
    setProcessingStage('error');
}
}, [videoPreview, youtubeUrl, perplexityResult, onVideoProcessed]);

const isProcessing = processingStage === 'validating' || processingStage === 'parsing';

return (
  <>
  <div className="space-y-8">
    <div className="text-center">
      <h2 className="text-3xl font-bold text-[#C8A05C] mb-4 font-serif">
        Procesar Transcripción con Perplexity
      </h2>
      <p className="text-[#5C4033]/80 max-w-3xl mx-auto">
        Este app usa tu suscripción de Perplexity existente para transcribir y analizar décimas.
        Es <strong className="text-[#C8A05C]">100% GRATIS</strong> - solo copia el prompt, úsallo en
        y pega el resultado aquí para formatearlo profesionalmente.
      </p>
    </div>

    <Card className="p-8 max-w-3xl mx-auto cuban-card">
      <div className="space-y-6">
        {/* Step 1: YouTube URL */}
        <div className="space-y-2">
          <div className="flex items-center gap-2">
            <div className="flex items-center justify-center w-6 h-6 rounded-full bg-[#C8A05C] text-white text-sm font-bold">
              1
            </div>
            <label htmlFor="youtube-url" className="text-sm font-bold text-[#5C4033]">
              Pega el enlace de YouTube
            </label>
          </div>
          <div className="flex gap-3">
            <Input
              id="youtube-url"
              placeholder="https://www.youtube.com/watch?v=..."
              value={youtubeUrl}
              onChange={(e) => setYoutubeUrl(e.target.value)}
              onBlur={() => validateAndPreviewUrl(youtubeUrl)}
              className="flex-1 bg-white/90 border-[#C8A05C]/30 focus:border-[#D2691E]"
              disabled={isProcessing}
            />
            <Button
              onClick={() => validateAndPreviewUrl(youtubeUrl)}
              disabled={!youtubeUrl.trim() || isProcessing}
              variant="outline"
              className="border-[#C8A05C] text-[#C8A05C] hover:bg-[#C8A05C] hover:text-white"
            >

```

```

        {processingStage === 'validating' ? (
            <Loader2 className="w-4 h-4 animate-spin" />
        ) : (
            'Validar'
        )}
    </Button>
</div>
</div>

/* Video Preview */
{videoPreview && (
    <Card className="p-4 bg-[#F5E6D3]/50 border-[#C8A05C]/20">
        <div className="flex gap-4 items-start">
            <div className="relative w-32 h-24 rounded-lg overflow-hidden flex-shrink-0">
                <Image
                    src={videoPreview.thumbnailUrl}
                    alt={videoPreview.title}
                    fill
                    className="object-cover"
                />
                <div className="absolute inset-0 flex items-center justify-center bg-black/30">
                    <Play className="w-8 h-8 text-white" />
                </div>
            </div>
            <div className="flex-1 min-w-0">
                <h3 className="font-semibold text-[#5C4033] line-clamp-2 mb-1">
                    {videoPreview.title}
                </h3>
                <p className="text-sm text-[#5C4033]/70">
                    Duración: {videoPreview.duration}
                </p>
            </div>
        </Card>
    )}

/* Step 2: Get Prompt */
<div className="space-y-2">
    <div className="flex items-center gap-2">
        <div className="flex items-center justify-center w-6 h-6 rounded-full bg-[#C8A05C] text-white p-2">
            2
        </div>
        <label className="text-sm font-bold text-[#5C4033]">
            Obtén el Prompt Maestro
        </label>
    </div>
    <Button
        onClick={handleOpenPromptModal}
        disabled={!youtubeUrl.trim() || isProcessing}
        variant="outline"
        className="w-full border-[#5C4033] text-[#5C4033] hover:bg-[#5C4033] hover:text-white px-4 py-2">
        <FileText className="w-5 h-5 mr-2" />
        Obtener Prompt para Perplexity
    </Button>
</div>

```

```

        </Button>
        <p className="text-xs text-[#5C4033]/70 text-center">
            Copia el prompt, pégalo en Perplexity.ai con tu enlace de YouTube, y espera el análisis
        </p>
    </div>

    {/* Step 3: Paste Result */}
    <div className="space-y-2">
        <div className="flex items-center gap-2">
            <div className="flex items-center justify-center w-6 h-6 rounded-full bg-[#C8A05C] text-[#5C4033] font-bold">
                3
            </div>
            <label htmlFor="perplexity-result" className="text-sm font-bold text-[#5C4033]">
                Pega el resultado de Perplexity aquí
            </label>
        </div>
        <Textarea
            id="perplexity-result"
            placeholder="Pega aquí TODO el análisis generado por Perplexity (décimas, análisis, etc)"
            value={perplexityResult}
            onChange={(e) => setPerplexityResult(e.target.value)}
            className="min-h-[300px] bg-white/90 border-[#C8A05C]/30 focus:border-[#D2691E] font-mono"
            disabled={isProcessing}
        />
        <p className="text-xs text-[#5C4033]/70">
            Asegúrate de copiar TODO el resultado, incluyendo décimas, análisis y estructura métrica
        </p>
    </div>

    {/* Process Button */}
    <Button
        onClick={processPerplexityResult}
        disabled={!videoPreview || !perplexityResult.trim() || isProcessing}
        className="w-full cuban-button text-lg py-6"
    >
        {processingStage === 'parsing' ? (
            <>
                <Loader2 className="w-5 h-5 mr-2 animate-spin" />
                Formateando resultado...
            </>
        ) : processingStage === 'completed' ? (
            <>
                <CheckCircle className="w-5 h-5 mr-2" />
                ¡Completado!
            </>
        ) : (
            <>
                <Sparkles className="w-5 h-5 mr-2" />
                Procesar y Formatear
            </>
        )}
    </Button>

    {/* Success Message */}

```

```

{processingStage === 'completed' && (
  <Card className="p-4 bg-green-50 border-green-200">
    <div className="flex items-center gap-3 text-green-800">
      <CheckCircle className="w-5 h-5" />
      <div>
        <p className="font-semibold">¡Análisis formateado con éxito!</p>
        <p className="text-sm">Navega a las otras secciones para ver los resultados profesionales</p>
      </div>
    </div>
  </Card>
)})

/* Error Message */
{error && processingStage === 'error' && (
  <Card className="p-6 bg-red-50 border-red-200">
    <div className="flex items-start gap-3 text-red-800">
      <AlertCircle className="w-5 h-5 mt-1 flex-shrink-0" />
      <div className="flex-1">
        <p className="font-semibold mb-2">Error al procesar el resultado</p>
        <pre className="text-sm whitespace-pre-wrap font-sans bg-white/50 p-3 rounded border border-gray-200">${error}</pre>
      </div>
    </div>
  </Card>
)})

/* Info Box */
<Card className="p-4 bg-[#C8A05C]/10 border-[#C8A05C]/30">
  <p className="text-sm text-[#5C4033]">
    <strong> ¿Por qué este método?</strong> Usar tu cuenta de Perplexity directamente te da acceso a todos los recursos de la app.
  </p>
  <ul className="text-sm text-[#5C4033] mt-2 space-y-1 ml-4">
    <li> Acceso a Claude Sonnet 4.5 (el mejor modelo para análisis literario)</li>
    <li> Costo $0 (usa tu suscripción existente)</li>
    <li> Control total sobre el proceso</li>
    <li> Resultados formateados profesionalmente en este app</li>
  </ul>
</Card>
</div>
</Card>
</div>

<PromptModal
  open={promptModalOpen}
  onOpenChange={setPromptModalOpen}
  youtubeUrl={youtubeUrl}
/>
</>
);
)
}

```

components/decimals-section.tsx

```
"use client";

import { useState, useEffect } from "react";
import { Card } from "@/components/ui/card";
import { Badge } from "@/components/ui/badge";
import { Clock, User, FileText } from "lucide-react";

interface DecimaData {
  id: string;
  number: number;
  lines: string[];
  rhymeScheme: string;
  meter: string;
  isComplete: boolean;
  isRegular: boolean;
  timestamp?: string;
}

interface PoetData {
  id: string;
  name: string;
  turnNumber?: number;
  decimas: DecimaData[];
}

interface VideoData {
  id: string;
  title: string;
  duration: string;
  processedAt: Date;
  processingTime?: number;
  poets: PoetData[];
}

interface DecimalsSectionProps {
  videoId: string | null;
}

export default function DecimalsSection({ videoId }: DecimalsSectionProps) {
  const [videoData, setVideoData] = useState<VideoData | null>(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState<string>("");

  useEffect(() => {
    if (!videoId) return;

    const fetchVideoData = async () => {
      setLoading(true);
      setError("");
      try {
        const response = await fetch(`https://api.example.com/videos/${videoId}`);
        const data = await response.json();
        setVideoData(data);
        setLoading(false);
      } catch (error) {
        setError("Error fetching video data");
        setLoading(false);
      }
    };
    fetchVideoData();
  }, [videoId]);
}
```

```

        const response = await fetch(`api/video/${videoId}/decimas`);
        const data = await response.json();

        if (!response.ok) {
            throw new Error(data.error || 'Error al cargar las décimas');
        }

        setVideoData(data);
    } catch (err) {
        setError(err instanceof Error ? err.message : 'Error al cargar las décimas');
    } finally {
        setLoading(false);
    }
};

fetchVideoData();
}, [videoId]);

if (!videoId) {
    return (
        <div className="text-center py-16">
            <FileText className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
            <h3 className="text-xl font-semibold text-[#5C4033] mb-2">
                No hay décimas para mostrar
            </h3>
            <p className="text-[#5C4033]/70">
                Procesa un video primero para ver las décimas formateadas aquí.
            </p>
        </div>
    );
}

if (loading) {
    return (
        <div className="text-center py-16">
            <div className="animate-spin w-8 h-8 border-4 border-[#C8A05C] border-t-transparent rounded-full" />
            <p className="text-[#5C4033]">Cargando décimas...</p>
        </div>
    );
}

if (error) {
    return (
        <div className="text-center py-16">
            <div className="text-red-600 mb-4" />
            <h3 className="text-xl font-semibold text-red-600 mb-2">Error</h3>
            <p className="text-red-600/70">{error}</p>
        </div>
    );
}

if (!videoData) {
    return (
        <div className="text-center py-16">

```

```

<FileText className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
<h3 className="text-xl font-semibold text-[#5C4033] mb-2">
    Video no encontrado
</h3>
<p className="text-[#5C4033]/70">
    No se pudo cargar la información del video.
</p>
</div>
);
}

return (
<div className="space-y-8">
<div className="text-center">
<h2 className="text-3xl font-bold text-[#C8A05C] mb-4 font-serif">
    Décimas Formateadas
</h2>
<p className="text-[#5C4033]/80 max-w-2xl mx-auto">
    Todas las décimas identificadas y formateadas según la estructura clásica de 10 versos octosí
</p>
</div>

/* Video Info Header */
<Card className="p-6 cuban-card">
<div className="grid md:grid-cols-2 lg:grid-cols-4 gap-4">
<div className="flex items-center gap-2">
<FileText className="w-5 h-5 text-[#D2691E]" />
<div>
<p className="text-sm text-[#5C4033]/70">Título del vídeo</p>
<p className="font-semibold text-[#5C4033] truncate">{videoData.title}</p>
</div>
</div>

<div className="flex items-center gap-2">
<Clock className="w-5 h-5 text-[#D2691E]" />
<div>
<p className="text-sm text-[#5C4033]/70">Duración</p>
<p className="font-semibold text-[#5C4033]">{videoData.duration}</p>
</div>
</div>

<div className="flex items-center gap-2">
<User className="w-5 h-5 text-[#D2691E]" />
<div>
<p className="text-sm text-[#5C4033]/70">Poetas identificados</p>
<p className="font-semibold text-[#5C4033]">
    {videoData.poets.map(p => p.name).join(", ")}
</p>
</div>
</div>

<div>
<p className="text-sm text-[#5C4033]/70">Fecha de análisis</p>
<p className="font-semibold text-[#5C4033]">

```

```

        {new Date(videoData.processedAt).toLocaleDateString('es-ES')}
    </p>
    {videoData.processingTime && (
        <p className="text-xs text-[#5C4033]/60">
            Tiempo: {Math.round(videoData.processingTime / 60)} minutos
        </p>
    )}
    </div>
</div>
</Card>

/* Décimas by Poet */
{videoData.poets.map((poet, poetIndex) => (
    <div key={poet.id} className="space-y-6">
        <div className="text-center">
            <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
                {poet.name}
                {poet.turnNumber && ` - Turno ${poet.turnNumber}`}
            </h3>
            <Badge className="mt-2 bg-[#D2691E] text-white">
                {poet.decimas.length} décima{poet.decimas.length !== 1 ? 's' : ''}
            </Badge>
        </div>

        {poet.decimas.map((decima, decimaIndex) => (
            <Card key={decima.id} className="p-6 cuban-card">
                <div className="space-y-4">
                    <div className="flex items-center justify-between flex-wrap gap-2">
                        <h4 className="text-lg font-semibold text-[#5C4033]">
                            Décima {decima.number}
                        </h4>
                        <div className="flex gap-2">
                            {!decima.isComplete && (
                                <Badge variant="outline" className="text-orange-600 border-orange-300">
                                    Incompleta
                                </Badge>
                            )}
                            {!decima.isRegular && (
                                <Badge variant="outline" className="text-yellow-600 border-yellow-300">
                                    Irregular
                                </Badge>
                            )}
                            {decima.timestamp && (
                                <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
                                    {decima.timestamp}
                                </Badge>
                            )}
                        </div>
                    </div>
                </div>
            </Card>
        ))
    )}
    </div>
</div>

/* Décima Lines */
<div className="decima-text text-lg leading-relaxed space-y-1">
    {decima.lines.map((line, lineIndex) => (
        <p key={lineIndex} className="text-[#5C4033]">

```

```

        {line}
      </p>
    ))}
</div>

/* Technical Info */
<div className="pt-4 border-t border-[#C8A05C]/20 space-y-2">
  <div className="flex flex-wrap gap-6 text-sm text-[#5C4033]/80">
    <span>
      <strong>Esquema de rima:</strong> {decima.rhymeScheme}
    </span>
    <span>
      <strong>Métrica:</strong> {decima.meter}
    </span>
    {decima.timestamp && (
      <span>
        <strong>Marca de tiempo:</strong> {decima.timestamp}
      </span>
    )}
    </div>
  </div>
</div>
</Card>
))}
</div>
))}

{videoData.poets.length === 0 && (
  <div className="text-center py-16">
    <FileText className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
    <h3 className="text-xl font-semibold text-[#5C4033] mb-2">
      No se encontraron décimas
    </h3>
    <p className="text-[#5C4033]/70">
      El análisis no identificó décimas en este video.
    </p>
  </div>
)
</div>
);
}

```

components/analysis-section.tsx

```

"use client";

import { useState, useEffect } from "react";
import { Card } from "@components/ui/card";
import { Badge } from "@components/ui/badge";
import { BookOpen, TrendingUp, Award } from "lucide-react";

interface TopDecima {
  id: string;

```

```

    number: number;
    lines: string[];
    poet: string;
    rank: number;
    technicalMastery?: string;
    poeticDevices?: string;
    culturalElements?: string;
    emotionalResonance?: string;
    improvisation?: string;
    topThreeReason?: string;
}

interface AnalysisData {
    id: string;
    videoTitle: string;

    // Summary
    summaryThemes?: string;
    summaryTone?: string;
    summaryInteraction?: string;
    summaryQuality?: string;

    // Deep analysis
    recurringThemes?: string;
    regionalisms?: string;
    linguisticSophistication?: string;
    improvisationTechniques?: string;
    canonComparison?: string;
    culturalSignificance?: string;

    // Top décimas by poet
    topDecimasByPoet: Record<string, TopDecima[]>;
}

interface AnalysisSectionProps {
    videoId: string | null;
}

export default function AnalysisSection({ videoId }: AnalysisSectionProps) {
    const [analysisData, setAnalysisData] = useState<AnalysisData | null>(null);
    const [loading, setLoading] = useState(false);
    const [error, setError] = useState<string>("");

    useEffect(() => {
        if (!videoId) return;

        const fetchAnalysisData = async () => {
            setLoading(true);
            setError("");

            try {
                const response = await fetch(`api/video/${videoId}/analysis`);
                const data = await response.json();

```

```

        if (!response.ok) {
            throw new Error(data.error || 'Error al cargar el análisis');
        }

        setAnalysisData(data);
    } catch (err) {
        setError(err instanceof Error ? err.message : 'Error al cargar el análisis');
    } finally {
        setLoading(false);
    }
};

fetchAnalysisData();
}, [videoId]);

if (!videoId) {
    return (
        <div className="text-center py-16">
            <BookOpen className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
            <h3 className="text-xl font-semibold text-[#5C4033] mb-2">
                No hay análisis disponible
            </h3>
            <p className="text-[#5C4033]/70">
                Procesa un video primero para ver el análisis completo aquí.
            </p>
        </div>
    );
}

if (loading) {
    return (
        <div className="text-center py-16">
            <div className="animate-spin w-8 h-8 border-4 border-[#C8A05C] border-t-transparent rounded-full" />
            <p className="text-[#5C4033]">Cargando análisis...</p>
        </div>
    );
}

if (error) {
    return (
        <div className="text-center py-16">
            <div className="text-red-600 mb-4" />
            <h3 className="text-xl font-semibold text-red-600 mb-2">Error</h3>
            <p className="text-red-600/70">{error}</p>
        </div>
    );
}

if (!analysisData) {
    return (
        <div className="text-center py-16">
            <BookOpen className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
            <h3 className="text-xl font-semibold text-[#5C4033] mb-2">
                Análisis no encontrado
            </h3>
        </div>
    );
}

```

```

        </h3>
        <p className="text-[#5C4033]/70">
            No se pudo cargar el análisis del video.
        </p>
    </div>
);
}

return (
    <div className="space-y-8">
        <div className="text-center">
            <h2 className="text-3xl font-bold text-[#C8A05C] mb-4 font-serif">
                Análisis Completo
            </h2>
            <p className="text-[#5C4033]/80 max-w-2xl mx-auto">
                Análisis académico profundo de las décimas, técnica poética, elementos culturales y calidad a
            </p>
        </div>

        {/* Summary of the Canturía */}
        <Card className="p-6 cuban-card">
            <div className="flex items-center gap-3 mb-4">
                <TrendingUp className="w-6 h-6 text-[#D2691E]" />
                <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
                    Resumen de la Canturía
                </h3>
            </div>

            <div className="space-y-4 text-[#5C4033] leading-relaxed">
                {analysisData.summaryThemes && (
                    <div>
                        <h4 className="font-semibold mb-2 text-[#D2691E]">Temas Principales:</h4>
                        <p>{analysisData.summaryThemes}</p>
                    </div>
                )}
                {analysisData.summaryTone && (
                    <div>
                        <h4 className="font-semibold mb-2 text-[#D2691E]">Tono y Energía Emocional:</h4>
                        <p>{analysisData.summaryTone}</p>
                    </div>
                )}
                {analysisData.summaryInteraction && (
                    <div>
                        <h4 className="font-semibold mb-2 text-[#D2691E]">Interacción entre Poetas:</h4>
                        <p>{analysisData.summaryInteraction}</p>
                    </div>
                )}
                {analysisData.summaryQuality && (
                    <div>
                        <h4 className="font-semibold mb-2 text-[#D2691E]">Calidad Poética e Impacto:</h4>
                        <p>{analysisData.summaryQuality}</p>
                    </div>
                )}
            </div>
        </Card>
    </div>
)

```

```

        </div>
    )}
</div>
</Card>

{/* Top 3 Décimas por Poeta */}
{Object.entries(analysisData.topDecimasByPoet).map(([poetName, topDecimas]) => (
  <Card key={poetName} className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-6">
      <Award className="w-6 h-6 text-[#D2691E]" />
      <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
        Top 3 Décimas: {poetName}
      </h3>
    </div>

    <div className="space-y-6">
      {topDecimas.map((decima, index) => (
        <div key={decima.id} className="border-1-4 border-[#D2691E] pl-6">
          <div className="flex items-center gap-3 mb-3">
            <Badge className="bg-[#D2691E] text-white">
              #{decima.rank}
            </Badge>
            <h4 className="text-lg font-semibold text-[#5C4033]">
              Décima {decima.number}
            </h4>
          </div>
        </div>

        {/* Décima Text */}
        <div className="decima-text text-base leading-relaxed mb-4 bg-[#F5E6D3]/50 p-4 rounded-1">
          {decima.lines.map((line, lineIndex) => (
            <p key={lineIndex} className="text-[#5C4033]">
              {line}
            </p>
          )))
        </div>
      </div>
    </div>

    {/* Analysis Details */}
    <div className="grid md:grid-cols-2 gap-4 text-sm">
      {decima.technicalMastery && (
        <div>
          <h5 className="font-semibold text-[#D2691E] mb-1">Maestría Técnica:</h5>
          <p className="text-[#5C4033]/80">{decima.technicalMastery}</p>
        </div>
      )}
    </div>

    {decima.poeticDevices && (
      <div>
        <h5 className="font-semibold text-[#D2691E] mb-1">Dispositivos Poéticos:</h5>
        <p className="text-[#5C4033]/80">{decima.poeticDevices}</p>
      </div>
    )}
  </div>
  {decima.culturalElements && (
    <div>

```

```

        <h5 className="font-semibold text-[#D2691E] mb-1">Elementos Culturales:</h5>
        <p className="text-[#5C4033]/80">{decima.culturalElements}</p>
    </div>
    )}

    {decima.emotionalResonance && (
        <div>
            <h5 className="font-semibold text-[#D2691E] mb-1">Resonancia Emocional:</h5>
            <p className="text-[#5C4033]/80">{decima.emotionalResonance}</p>
        </div>
    )}

    {decima.improvisation && (
        <div>
            <h5 className="font-semibold text-[#D2691E] mb-1">Improvisación:</h5>
            <p className="text-[#5C4033]/80">{decima.improvisation}</p>
        </div>
    )}

    {decima.topThreeReason && (
        <div className="md:col-span-2">
            <h5 className="font-semibold text-[#D2691E] mb-1">Por qué está en el Top 3:</h5>
            <p className="text-[#5C4033]/80">{decima.topThreeReason}</p>
        </div>
    )}
    </div>
    </div>
)}
```

/* Deep Analysis */

```

<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-6">
        <BookOpen className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Análisis Profundo
        </h3>
    </div>

    <div className="grid md:grid-cols-2 gap-6 text-[#5C4033]">
        {analysisData.recurringThemes && (
            <div>
                <h4 className="font-semibold mb-2 text-[#D2691E]">Temas Recurrentes:</h4>
                <p className="text-sm leading-relaxed">{analysisData.recurringThemes}</p>
            </div>
        )}
        {analysisData.regionalisms && (
            <div>
                <h4 className="font-semibold mb-2 text-[#D2691E]">Regionalismos y Dialectos:</h4>
                <p className="text-sm leading-relaxed">{analysisData.regionalisms}</p>
            </div>
        )}
    </div>
</Card>
```

```

        )}

{analysisData.linguisticSophistication && (
    <div>
        <h4 className="font-semibold mb-2 text-[#D2691E]">Sofisticación Lingüística:</h4>
        <p className="text-sm leading-relaxed">{analysisData.linguisticSophistication}</p>
    </div>
)}

{analysisData.improvisationTechniques && (
    <div>
        <h4 className="font-semibold mb-2 text-[#D2691E]">Técnicas de Improvisación Observadas:</h4>
        <p className="text-sm leading-relaxed">{analysisData.improvisationTechniques}</p>
    </div>
)}

{analysisData.canonComparison && (
    <div>
        <h4 className="font-semibold mb-2 text-[#D2691E]">Comparación con el Canon Cubano:</h4>
        <p className="text-sm leading-relaxed">{analysisData.canonComparison}</p>
    </div>
)}

{analysisData.culturalSignificance && (
    <div>
        <h4 className="font-semibold mb-2 text-[#D2691E]">Significancia Cultural:</h4>
        <p className="text-sm leading-relaxed">{analysisData.culturalSignificance}</p>
    </div>
)}
```

```

const exportToPDF = async () => {
  if (!videoId) return;

  setExporting('pdf');
  try {
    const response = await fetch(`api/video/${videoId}/export/pdf`, {
      method: 'POST'
    });

    if (!response.ok) {
      throw new Error('Error al generar PDF');
    }

    const blob = await response.blob();
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.style.display = 'none';
    a.href = url;
    a.download = `guajiro-de-hialeah-analisis-${videoId}.pdf`;
    document.body.appendChild(a);
    a.click();
    window.URL.revokeObjectURL(url);

    toast({
      title: "PDF generado",
      description: "El análisis se ha descargado correctamente.",
    });
  } catch (error) {
    toast({
      title: "Error",
      description: "No se pudo generar el PDF. Inténtalo de nuevo.",
      variant: "destructive",
    });
  } finally {
    setExporting(null);
  }
};

const generateShareLink = async () => {
  if (!videoId) return;

  setExporting('link');
  try {
    const response = await fetch(`api/video/${videoId}/share`, {
      method: 'POST'
    });

    if (!response.ok) {
      throw new Error('Error al generar enlace');
    }

    const data = await response.json();
    const fullUrl = `${window.location.origin}/share/${data.shareId}`;
    setShareUrl(fullUrl);
  }
};

```

```

        toast({
          title: "Enlace generado",
          description: "El enlace de compartir está listo.",
        });
      } catch (error) {
        toast({
          title: "Error",
          description: "No se pudo generar el enlace. Inténtalo de nuevo.",
          variant: "destructive",
        });
      } finally {
        setExporting(null);
      }
    };

const copyToClipboard = async () => {
  if (!videoId) return;

  setExporting('clipboard');
  try {
    const response = await fetch(`api/video/${videoId}/export/text`);

    if (!response.ok) {
      throw new Error('Error al obtener el contenido');
    }

    const data = await response.json();
    const textContent = `

GUAJIRO DE HIALEAH - ANÁLISIS DE DÉCIMA ESPINELA

Título: ${data.title}
Duración: ${data.duration}
Fecha de análisis: ${new Date(data.processedAt).toLocaleDateString('es-ES')}

${data.textContent}

---


Generado por El Guajiro de Hialeah
${window.location.origin}
` .trim();

    await navigator.clipboard.writeText(textContent);

    toast({
      title: "Copiado al portapapeles",
      description: "El análisis se ha copiado correctamente.",
    });
  } catch (error) {
    toast({
      title: "Error",
      description: "No se pudo copiar el contenido. Inténtalo de nuevo.",
      variant: "destructive",
    });
  }
};

```

```

    } finally {
      setExporting(null);
    }
};

const copyShareUrl = async () => {
  if (!shareUrl) return;

  try {
    await navigator.clipboard.writeText(shareUrl);
    toast({
      title: "Enlace copiado",
      description: "El enlace se ha copiado al portapapeles.",
    });
  } catch (error) {
    toast({
      title: "Error",
      description: "No se pudo copiar el enlace.",
      variant: "destructive",
    });
  }
};

if (!videoId) {
  return (
    <div className="text-center py-16">
      <Download className="w-16 h-16 mx-auto text-[#C8A05C]/50 mb-4" />
      <h3 className="text-xl font-semibold text-[#5C4033] mb-2">
        No hay contenido para exportar
      </h3>
      <p className="text-[#5C4033]/70">
        Procesa un video primero para poder exportar los resultados.
      </p>
    </div>
  );
}

return (
  <div className="space-y-8">
    <div className="text-center">
      <h2 className="text-3xl font-bold text-[#C8A05C] mb-4 font-serif">
        Exportar Resultados
      </h2>
      <p className="text-[#5C4033]/80 max-w-2xl mx-auto">
        Guarda, comparte o copia los resultados del análisis en diferentes formatos.
      </p>
    </div>

    <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6 max-w-4xl mx-auto">
      {/* PDF Export */}
      <Card className="p-6 cuban-card text-center">
        <div className="mb-4">
          <div className="w-16 h-16 mx-auto bg-[#D2691E]/10 rounded-full flex items-center justify-ce

```

```

        </div>
        <h3 className="text-lg font-semibold text-[#5C4033] mb-2">
            Descargar PDF
        </h3>
        <p className="text-sm text-[#5C4033]/70 mb-4">
            Genera un PDF completo con todo el análisis, décimas y contenido educativo.
        </p>
    </div>
    <Button
        onClick={exportToPDF}
        disabled={exporting === 'pdf'}
        className="w-full cuban-button"
    >
        {exporting === 'pdf' ? (
            <>
                <div className="animate-spin w-4 h-4 border-2 border-white border-t-transparent rounded" style={{ width: '1em', height: '1em' }}>
                    Generando...
                </div>
            </>
        ) : (
            <>
                <Download className="w-4 h-4 mr-2" />
                Descargar PDF
            </>
        )}
    </Button>
</Card>

{/* Share Link */}
<Card className="p-6 cuban-card text-center">
    <div className="mb-4">
        <div className="w-16 h-16 mx-auto bg-[#C8A05C]/10 rounded-full flex items-center justify-center">
            <Link className="w-8 h-8 text-[#C8A05C]" />
        </div>
        <h3 className="text-lg font-semibold text-[#5C4033] mb-2">
            Compartir Enlace
        </h3>
        <p className="text-sm text-[#5C4033]/70 mb-4">
            Crea un enlace público para compartir el análisis completo.
        </p>
    </div>
    <Button
        onClick={generateShareLink}
        disabled={exporting === 'link'}
        variant="outline"
        className="w-full border-[#C8A05C] text-[#C8A05C] hover:bg-[#C8A05C] hover:text-white mb-3"
    >
        {exporting === 'link' ? (
            <>
                <div className="animate-spin w-4 h-4 border-2 border-[#C8A05C] border-t-transparent rounded" style={{ width: '1em', height: '1em' }}>
                    Generando...
                </div>
            </>
        ) : (
            <>
                <Link className="w-4 h-4 mr-2" />

```

```

        Generar Enlace
    </>
    )
</Button>

{shareUrl && (
    <div className="space-y-2">
        <div className="p-2 bg-[#F5E6D3] rounded text-xs text-[#5C4033] break-all">
            {shareUrl}
        </div>
        <Button
            onClick={copyShareUrl}
            size="sm"
            variant="ghost"
            className="w-full text-[#5C4033] hover:bg-[#C8A05C]/10"
        >
            <Copy className="w-3 h-3 mr-1" />
            Copiar enlace
        </Button>
    </div>
)
}
</Card>

/* Copy to Clipboard */
<Card className="p-6 cuban-card text-center">
    <div className="mb-4">
        <div className="w-16 h-16 mx-auto bg-[#8B4513]/10 rounded-full flex items-center justify-cen
            <Copy className="w-8 h-8 text-[#8B4513]" />
        </div>
        <h3 className="text-lg font-semibold text-[#5C4033] mb-2">
            Copiar al Portapapeles
        </h3>
        <p className="text-sm text-[#5C4033]/70 mb-4">
            Copia un resumen textual del análisis para pegar en otros documentos.
        </p>
    </div>
    <Button
        onClick={copyToClipboard}
        disabled={exporting === 'clipboard'}
        variant="outline"
        className="w-full border-[#8B4513] text-[#8B4513] hover:bg-[#8B4513] hover:text-white"
    >
        {exporting === 'clipboard' ? (
            <>
                <div className="animate-spin w-4 h-4 border-2 border-[#8B4513] border-t-transparent round
                    Copiando...
            </>
        ) : (
            <>
                <Copy className="w-4 h-4 mr-2" />
                Copiar Texto
            </>
        )
    }
    </Button>

```

```

        </Card>
    </div>

    {/* Usage Instructions */}
    <Card className="p-6 cuban-card max-w-4xl mx-auto">
        <h3 className="text-lg font-semibold text-[#5C4033] mb-4 flex items-center gap-2">
            <CheckCircle className="w-5 h-5 text-[#D2691E]" />
            Instrucciones de Uso
        </h3>
        <div className="grid md:grid-cols-3 gap-4 text-sm text-[#5C4033]/80">
            <div>
                <h4 className="font-medium text-[#D2691E] mb-2">PDF</h4>
                <p>
                    Ideal para archivo, impresión o presentaciones académicas. Incluye todo el contenido con el diseño vintage auténtico.
                </p>
            </div>
            <div>
                <h4 className="font-medium text-[#C8A05C] mb-2">Enlace</h4>
                <p>
                    Perfecto para compartir en redes sociales o enviar por email.
                    El enlace es público y no requiere registro.
                </p>
            </div>
            <div>
                <h4 className="font-medium text-[#8B4513] mb-2">Texto</h4>
                <p>
                    Útil para integrar en documentos, blogs o investigaciones.
                    Formato plano con toda la información esencial.
                </p>
            </div>
        </Card>
    </div>
);
}

```

components/education-section.tsx (SOLO INICIO - 100 líneas)

```

"use client";

import React from "react";
import { Card } from "@components/ui/card";
import { Badge } from "@components/ui/badge";
import { BookOpen, Crown, Users, Award } from "lucide-react";

export default function EducationSection() {
    const [expandedPoet, setExpandedPoet] = React.useState<string | null>(null);

    const historicalFigures = [
        "Vicente Espinel",
        "El Cucalambé",
        "Plácido",
    ];

```

```

    "El Indio Naborí",
    "Angelito Valiente",
    "Francisco Riverón Hernández",
    "Justo Vega",
    "Adolfo Alfonso",
    "Chanito Isidrón",
    "Omar Mirabal",
    "Tomasita Quiala",
    "Julio Pablo El Príncipe Pinareño"
];
}

const getPoetBiography = (name: string) => {
  switch(name) {
    case "Vicente Espinel":
      return (
        <div className="space-y-4">
          <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
            Vicente Espinel (1550-1624): El Arquitecto Original
          </h4>
          <p>
            Vicente Gómez Martínez Espinel nació en Ronda, Málaga, el 28 de diciembre de 1550 y falleció en la misma ciudad el 25 de febrero de 1624. Fue un arquitecto y poeta español, considerado uno de los más importantes del Siglo de Oro.
          </p>
          <p>
            Su legado perdurable cristalizó en la décima: diez versos octosílabos con arquitectura de versos.
          </p>
          <p>
            Espinel poseía vastísima cultura. Dominaba el latín, siendo llamado "el único poeta latino de su tiempo".
          </p>
          <p>
            Su obra cumbre en prosa fue la novela picaresca <em>Relaciones de la vida del escudero Ma...
          </p>
        </div>
      );
    case "El Cucalambé":
      return (
        <div className="space-y-4">
          <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
            El Cucalambé (Juan Cristóbal Nápoles Fajardo, 1829-1861): Primer Arquitecto de la Cubanización
          </h4>
          <p>
            Juan Cristóbal Nápoles Fajardo nació en Victoria de Las Tunas y desapareció misteriosamente en 1861.
          </p>
          <p>
            Educado por su abuelo materno José Rafael Fajardo García-hacendado, cura y conspirador anti...
          </p>
          <p>
            Fue el único poeta que logró ser aceptado plenamente por el pueblo, entrando totalmente en la historia de Cuba.
          </p>
        </div>
      );
    case "Plácido":
      return (

```

```

<div className="space-y-4">
  <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
    Plácido (Gabriel de la Concepción Valdés, 1809-1844): Transparencia Trágica
  </h4>
  <p>
    Gabriel de la Concepción Valdés nació en La Habana, hijo de un peluquero mulato y una bai...
  </p>
  <p>
    Huérfano y artesano, trabajó en múltiples oficios: carpintería, tipografía, fabricación d...
  </p>
  <p>
    Su habilidad como artesano del carey lo llevó a Matanzas en 1826 para trabajar en el tall...
  </p>
  <p>
    Fue fusilado el 28 de junio de 1844, acusado injustamente de ser jefe de la Conspiración ...
  </p>
</div>
);

case "El Indio Naborí":
  return (
    <div className="space-y-4">
      <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
        El Indio Naborí (Jesús Orta Ruiz, 1922-2005): Voz Lírica del Siglo XX
      </h4>
      <p>
        Jesús Orta Ruiz, nacido en Los Zapotes, San Miguel del Padrón, representa la figura más r...
      </p>
      <p>
        Su etapa juvenil de repentina dejó huellas tan profundas que todavía se le invoca con em...
      </p>
      <p>
        La revancha del 28 de agosto de 1955 en el estadio Campo Armada de San Miguel del Padrón ...
      </p>
      <p>
        Tenía una voz dulce, melodiosa, y cantaba con delicadeza, acariciando las palabras. Recib...
      </p>
    </div>
  );
}

case "Angelito Valiente":
  return (
    <div className="space-y-4">
      <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
        Ángel Valiente "Angelito" (1916-1987): Teatro en Movimiento
      </h4>
      <p>
        Ángel Miguel Valiente Rodríguez nació el 28 de febrero de 1916 en San Antonio de los Baños...
      </p>
      <p>
        Desde los ocho años improvisaba décimas. Ya en su juventud había logrado fama nacional a ...
      </p>
      <p>
        Su caballerosidad quedó inmortalizada cuando, tras la decisión del jurado de otorgar el t...
      </p>
    </div>
  );
}

```

```

        </p>
        <p>
            Fue combatiente revolucionario y fundador y dirigente de los Comités de Defensa de la Revolución
        </p>
    </div>
);

case "Francisco Riverón Hernández":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Francisco Riverón Hernández (1917-1975): Poeta Pictórico
            </h4>
            <p>
                Francisco Riverón Hernández nació el 2 de abril de 1917 en Güines, La Habana. Tuvo que aban...
            </p>
            <p>
                Riverón cultivó distintas formas del verso, pero en la décima logró sus mejores aciertos ...
            </p>
            <p>
                Sus poemas fueron muy conocidos, tal es el caso de "Epístola a José Martí" (<em>José de la ...</em>)
            </p>
            <p>
                Por la elevación y finura de sus décimas, este decimista es considerado "el precursor de ...
            </p>
            <p>
                Fue fundador y director del programa radial <em>Renacer Cubano</em> de Radio Marianao. Su ...
            </p>
        </div>
);

case "Justo Vega":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Justo Vega (1909-1993): El Caballero de la Décima Improvisada
            </h4>
            <p>
                Justo Vega, maestro de la improvisación tradicional, nació en Matanzas el 9 de agosto de ...
            </p>
            <p>
                Su fama creció por sus presentaciones en programas radiales como "Patria Guajira" y espec...
            </p>
            <p>
                Más que un gran intérprete, se convirtió en artífice de fabulosas controversias, evocadas ...
            </p>
        </div>
);

case "Adolfo Alfonso":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Adolfo Alfonso (1925-2012): Compenetración sin Réplica
            </h4>

```

```

        </h4>
        <p>
            Adolfo Alfonso, decimista nato considerado uno de los mejores que ha tenido Cuba, a los c
        </p>
        <p>
            El azar lo llevó a encontrarse con Justo Vega en el programa radiofónico <em>Las Mil Diez
        </p>
        <p>
            Tan fuerte era la compenetración entre ambos trovadores que, después de muerto Justo Vega
        </p>
        <p>
            El inigualable talento poético, el modo inconfundible de entonar e interpretar la más aut
        </p>
        </div>
    );

case "Chanito Isidrón":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Chanito Isidrón (Cipriano Isidrón Torres, 1903-1992): Fundador de la Novela en Décimas
            </h4>
            <p>
                Cipriano Isidrón Torres nació en Calabazar de Sagua, Las Villas, hijo de padres canarios.
            </p>
            <p>
                En 1938, con <em>Amores Montaraces</em>, se convirtió en fundador de la novela en décimas
            </p>
            <p>
                La novela cuenta la historia de Camilo y Estrella. Comienza así: "Camilo, galán guajiro /
            </p>
            <p>
                Se inició en la radio de Santa Clara en 1936, y La Cadena Azul lo contrató para escribir
            </p>
            <p>
                Su humor agresivo era como una explosión de rebeldía, de protesta, de anuncio de futuro. I
            </p>
        </div>
    );

case "Omar Mirabal":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Omar Mirabal (1955-presente): El Cantor del Mayabeque
            </h4>
            <p>
                Omar Mirabal Navarro nació el 14 de noviembre de 1955 en Güines, municipio del actual May
            </p>
            <p>
                Para una buena parte de los amantes de la décima, Omar Mirabal es el mejor poeta de Cuba,
            </p>
            <p>
                Los recursos expresivos mirabalianos tienen sus raíces en la más profunda tradición del ha
            </p>
        </div>
    );

```

```

        </p>
        <p>
            La popularidad que Omar Mirabal llegó a alcanzar en los años 80, junto a la calidad de su
        </p>
        <p>
            Omar le imprime a su tonada un aire de parsimonia que por momentos denota cansancio y que
        </p>
    </div>
);

case "Tomasita Quiala":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Tomasita Quiala (1960-2025): La Reina del Repentismo
            </h4>
            <p>
                Juana Tomasa Quiala Rojas nació el 29 de diciembre de 1960 en Arroyón de Flores, Banes, p
            </p>
            <p>
                Invidente desde su nacimiento, realizó estudios en la escuela para ciegos Varona Suárez (C
            </p>
            <p>
                Sus inicios en la improvisación fueron inesperados cuando intervino espontáneamente en una
            </p>
            <p>
                Se destacó por su interpretación de la décima de pie forzado y en la controversia. Fue em
            </p>
            <p>
                Recibió numerosas distinciones, incluyendo el Premio Nacional de la Cultura Comunitaria (C
            </p>
            <p>
                Su primer libro de décimas, <em>¿Quién soy? La novia de Islas Canarias</em>, fue transcri
            </p>
        </div>
);

case "Julio Pablo El Príncipe Pinareño":
    return (
        <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
                Julio Pablo Travieso Martínez: Arquitecto de Versos en Tiempo Presente
            </h4>

            <div>
                <h5 className="font-semibold text-[#D2691E] mb-2">El Arte Como Respiración</h5>
                <p>
                    Observa el escenario vacío. Siente el silencio colectivo antes del primer verso.
                    Julio Pablo Travieso Martínez, nacido en 1996 bajo el cielo pinareño, transforma
                    ese silencio en arquitectura sonora-décimas que construyen puentes entre siglos,
                    generaciones, islas.
                </p>
                <p className="mt-2">
                    Este joven maestro no preserva tradiciones como quien guarda reliquias en vitrinas.
                </p>
            </div>

```

Las respira. Las reinterpreta. Las expande. El repentismo fluye a través de él como agua que recuerda su cauce ancestral mientras descubre territorios nuevos. Tradición y renovación. Memoria y creación. Ambas. Siempre ambas.

</p>
</div>

<div>
 <h5 className="font-semibold text-[#D2691E] mb-2">Raíces Que Sostienen, Alas Que Liberan</h5>
 <p>

 La Casa de la Décima "Celestino García"-fundada en 2008 como ecosistema cultural-nutrión semillas que hoy florecen ante audiencias nacionales e internacionales. Desde esa insti- emblemática, Travieso cultivó el dominio técnico que transforma la improvisación en pre- controlada: diez versos octosílabos que contienen universos.

</p>
 <p className="mt-2">

 Oralitura Habana 2019 lo vio compartir escenario con decimistas cuyas voces cruzaban oce- Las controversias con Yoniel Perdomo-febrero 2025, versos sobre infancia campesina que o- horas-revelaron la resistencia física y mental que exige el arte oral sostenido. La con- poética con el puertorriqueño Julio César Sanabria demostró que la décima borra fronteras geográficas mediante resonancia cultural compartida.

</p>
 <p className="mt-2">

 Cada presentación confirma una verdad fundamental: la técnica magistral habilita la exp- auténtica, nunca la oscurece.

</p>
</div>

<div>
 <h5 className="font-semibold text-[#D2691E] mb-2">El Mago del Momento: Cuando la Décima Re-
 <p>

 Travieso merece también el título de "Mago del Momento"-designación que captura la natura- efímera y milagrosa del repentismo. Él enseña una verdad que los grandes maestros compren- intuitivamente: la décima es la magia del momento.

</p>
 <p className="mt-2">

 No existe ensayo para la improvisación. No hay segunda oportunidad para el verso que na- muere en el instante de su pronunciación. El poeta sostiene la audiencia en suspenso compartido-el momento cuando pensamiento se vuelve palabra, cuando palabra se cristaliza forma perfecta, cuando forma capture emoción que todos reconocen como auténtica.

</p>
 <p className="mt-2">

 Esta magia exige presencia total: el poeta no puede distraerse, dudar, retroceder. Debe que años de práctica emergerán como instinto. Que la forma memorizada se llenará de con- espontáneo. Que la estructura sostendrá el peso de la verdad improvisada. Travieso demue- este milagro cada vez que transforma silencio en verso-cada controversia donde el momen- genera significado que ninguna composición escrita replicaría exactamente.

</p>
</div>

<div>
 <h5 className="font-semibold text-[#D2691E] mb-2">La Décima Como Instrumento de Precisión</h5>
 <p>

 La espinela-estrofa nacional cubana, forma heredada de España que Cuba adoptó como voz propia-encuentra en Travieso un intérprete que entiende su paradoja esencial: estructura-

rígida que genera libertad ilimitada. Diez versos, esquema fijo de rima, ritmo octosílabo invariable. Dentro de esos límites aparentemente restrictivos, el poeta improvisa sobre cualquier tema con la fluidez del pensamiento hablado.

</p>

<p className="mt-2"> Vida campesina. Amor. Divorcio. Historia. Filosofía. Cada tema se moldea en décimas que satisfacen exigencias técnicas mientras transmiten verdades humanas inmediatas. La forma decora el contenido-lo encarna. La restricción métrica no limita significado-lo cristaliza en memorabilidad.

</p>

</div>

<div>

<h5 className="font-semibold text-[#D2691E] mb-2">Generación Guardiana, Generación Innovadora

<p> Travieso pertenece a la generación que heredó responsabilidad histórica: mantener viva la tradición oral en era digital. Junto a Anabeybi Rodríguez—"la princesa pinareña"—y otros jóvenes repentistas, forma una red de artistas que entienden que preservar no significa conservar.

</p>

<p className="mt-2"> Estos poetas contemporáneos aplican formas centenarias a preocupaciones actuales. Usan las tecnologías digitales para amplificar versos que antes solo viajaban hasta donde alcanzaban voces sin amplificación. Graban presentaciones que antes existían únicamente en memoria colectiva. Documentan procesos creativos que maestros anteriores—como Basilio Echevarría Acosta, "El Negro"—transmitieron exclusivamente mediante ejemplo directo.

</p>

<p className="mt-2"> Mueven la décima hacia el futuro honrando su pasado. Velocidad con reverencia.

</p>

</div>

<div>

<h5 className="font-semibold text-[#D2691E] mb-2">Geografía Cultural: Pinar del Río Como Territorio

<p> Cada verso que Travieso improvisa lleva la marca de su provincia occidental. Pinar del Río—donde el punto guajiro no es género musical sino lenguaje materno, donde la décima estructura pensamiento colectivo—habla a través de él. No como ventriloquio que repite frases ajenas, sino como voz individual que articula experiencia comunitaria.

</p>

<p className="mt-2"> El título "Príncipe Pinareño" reconoce esa doble función: maestría personal que representa la herencia colectiva. Travieso proyecta identidad regional mientras alcanza relevancia universal. Lo específicamente cubano ilumina lo fundamentalmente humano.

</p>

</div>

<div>

<h5 className="font-semibold text-[#D2691E] mb-2">Testimonio Personal: Privilegio Compartido

<p> Conocer a Julio Pablo Travieso. Llamarlo amigo. Reconocerlo como hermano. Estas experiencias constituyen bendiciones que agradezco con gratitud diaria, porque presenciar maestría poética en persona—observar cómo la mente transforma experiencia en verso mientras el verso se pronuncia—revela procesos creativos que ninguna grabación captura completamente.

</p>

```

<p className="mt-2">
  Su presencia confirma que el arte del pueblo no es primitivo ni simple. Es sofisticado en ejecución, profundo en implicación, accesible sin sacrificar complejidad.
</p>
</div>

<div>
  <h5 className="font-semibold text-[#D2691E] mb-2">Resistencia Cultural Como Acto Creativo</h5>
  <p>
    Cada controversia que Travieso sostiene, cada festival donde participa, cada alumno que constituye acto de resistencia cultural: testimonio viviente de que tradiciones orales y vibrantes cuando artistas talentosos las encarnan con compromiso contemporáneo. La décima improvisada no es reliquia museográfica sino herramienta comunicativa actual.
  </p>
  <p className="mt-2">
    En contextos donde homogeneización cultural amenaza especificidades regionales, donde el pasivo reemplaza participación activa, donde velocidad privilegia superficialidad, el ritmo ofrece alternativa: creación compartida en tiempo real entre poeta y audiencia, forma que presencia total.
  </p>
</div>

<div>
  <h5 className="font-semibold text-[#D2691E] mb-2">Trayectoria Que Continúa</h5>
  <p>
    Julio Pablo Travieso Martínez escribe su capítulo en la historia de la poesía cubana como que vuelan-literalmente-en el momento de su creación. Su trabajo asegura que generaciones futuras heredarán tradición viva, no archivo histórico. Que la décima seguirá evolucionando, perder identidad esencial. Que Pinar del Río mantendrá su voz distintiva en conversaciones culturales nacionales e internacionales.
  </p>
  <p className="mt-2">
    Que la palabra colorida y precisa-la que Martí persiguió incansablemente-continúa emergiendo a través del oficio paciente de artistas comprometidos con excelencia.
  </p>
  <p className="mt-2 font-medium italic text-[#C8A05C]">
    Que sus versos vuelen alto, llevando esencia cubana a corazones que reconozcan en la décima improvisada lo que siempre ha sido: sofisticación accesible, particularidad que alcanza universalidad, técnica magistral al servicio de verdad compartida. Magia que ocurre una y otra vez exactamente igual-porque así es el momento, y así es el arte que lo honra.
  </p>
</div>
</div>
);
}

default:
  return null;
}
};

return (
  <div className="space-y-8">
    <div className="text-center">
      <h2 className="text-3xl font-bold text-[#C8A05C] mb-4 font-serif">

```

```

        Educación Histórica
    </h2>
    <p className="text-[#5C4033]/80 max-w-2xl mx-auto">
        Conoce la rica historia de la décima espinela, desde sus orígenes españoles
        hasta su florecimiento en la tradición oral cubana.
    </p>
</div>

/* Arquitectura Poética, Raíz Compartida */
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <BookOpen className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Arquitectura Poética, Raíz Compartida
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            La décima espinela despliega diez versos octosílabos organizados según el esquema
            <strong> ABBAACCDDC</strong>. Esta arquitectura, cristalizada por el poeta andaluz
            <strong> Vicente Espinel</strong> en 1591 dentro de su obra <em>Diversas rimas</em>,
            sostiene siglos de expresión lírica. Espinel no inventó la estrofa de diez versos-ya
            circulaban "décimas primitivas" con estructuras variables-sino que codificó su forma
            definitiva: pausa obligatoria tras el cuarto verso, cuatro rimas consonantes perfectamente
            balanceadas. Músico además de poeta, Espinel comprendió el ritmo como sustancia viva: la
            pausa central permite tanto respiración física como giro conceptual, transformando diez
            líneas en conversación interna entre planteamiento y resolución.
        </p>
        <p>
            La estructura revela simetría orgánica. <strong>Primera cuarteta (ABBA)</strong>:
            presentación del tema, semilla conceptual. <strong>Versos centrales (AC)</strong>:
            bisagra donde el pensamiento pivota, unidos por encabalgamiento que completa en el sexto
            verso la idea iniciada en el quinto. <strong>Segunda cuarteta (CDDC)</strong>: culminación,
            cierre que resuena. Esta geometría habilita tanto meditación pausada como improvisación
            brillante-flexibilidad que explica su perdurabilidad a través de cinco siglos y dos continentes.
        </p>
    </div>
</Card>

/* Florecimiento Español: Del Creador a los Maestros */
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <Crown className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Florecimiento Español: Del Creador a los Maestros
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            <strong>Lope de Vega</strong> descubrió la espinela entre la obra de Espinel, la ponderó
            como "dulce y sonora", le otorgó nombre permanente y la insertó en el Parnaso español. Esta
            consagración por el "Fénix de los ingenios" convirtió la décima en la estrofa octosílabica
            más practicada del Siglo de Oro. Lo que el soneto representó para el verso largo, la espinela
            lo fue para el corto: proporciones simétricas, eficacia tanto como unidad suelta como en sección.
        </p>
    </div>
</Card>
```

```

        </p>
    <p>
        <strong>Calderón de la Barca, Góngora, Quevedo, Cervantes</strong>-los colosos literarios del siglo XVII adoptaron la forma. La décima fluyó simultáneamente en poesía culta y popular, en teatro y certámenes públicos donde competían "glosas en ocasión de festividades". Esta doble ciudadanía-aristocrática y plebeya-estableció versatilidad fundacional: la espinela no pertenece a una clase social específica sino a quienes dominaban su técnica.
    </p>
</div>
</Card>

/* Travesía Atlántica: Semilla en Nuevo Suelo */
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <Users className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Travesía Atlántica: Semilla en Nuevo Suelo
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-5">
        <div>
            <h4 className="font-semibold text-[#D2691E] mb-3 text-lg">
                Primer Registro Americano (1608)
            </h4>
            <p>
                <strong>Silvestre de Balboa</strong>, canario radicado en Cuba, compuso en 1608 <em>Espejo de paciencia</em>, primer poema épico de la literatura cubana. La obra narra el secuestro del Obispo de Cuba Juan de las Cabezas Altamirano cerca de Bayamo. Este registro inaugural marca presencia temprana de la décima en suelo americano, portada por emigrantes peninsulares e insulares canarios que traían la forma métrica en su equipaje cultural invisible pero permanente.
            </p>
        </div>
        <div>
            <h4 className="font-semibold text-[#D2691E] mb-3 text-lg">
                Consolidación Campesina (Siglo XVIII)
            </h4>
            <p>
                Durante el siglo XVIII, la inmigración canaria intensificada depositó la décima en el campo cubano. Los emigrantes canarios "trajeron a Cuba la forma de cantar décimas", donde se estableció como <strong>punto guajiro</strong>-género netivamente cubano que fusionó métrica española con sensibilidad criolla. El campesinado adoptó la espinela como continente natural para su cosmovisión: lamentos, celebraciones, sátiras, filosofía rural encontraron expresión en diez versos.
            </p>
            <p className="mt-3">
                <strong>Juan Cristóbal Nápoles Fajardo, el Cucalambé</strong> (1829-1862), cubanizó definitivamente la décima al adaptarla "como continente estético al contenido de lo identitario cubano". Su poemario <em>Rumores del Hórmigo</em> (1857) demostró "símbiosis efectiva, sin fracturas, de lo popular y lo culto". El Cucalambé no despertó epígonos entre poetas cultos posteriores, pero su legado arraigó en el pueblo, donde sí encontró continuadores que desarrollaron sus líneas expresivas. Fue "la voz del campesino que nos constituyó medularmente como pueblo", estableciendo la décima como <strong>"Mágica Estrofa Nacional"</strong>.
            </p>
        </div>
    </div>
</Card>

```

```

        </p>
    </div>
</div>
</Card>

{/* Evolución Cultural */}
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <Award className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Evolución Cultural: De lo Escrito a lo Oral, de lo Rural a lo Urbano
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            La décima transitó desde poesía culta escrita hacia improvisación oral rural, manteniendo vitalidad en ambos registros. Esta transformación la convirtió en puente entre clases sociales y geografías: la misma forma que Calderón pulía en palacios madrileños resonaba en bohíos cubanos, adaptada por campesinos que jamás leyeron a Espinel pero dominaban su arquitectura como lengua materna.
        </p>
        <p>
            El <strong>repentismo</strong>-improvisación de décimas en competencia se consolidó como arte nacional cubano. Repentistas como Pedro Guerra, Chanito Isidró, Guillermo Sosa alcanzaron estatura mítica en controversias donde la agilidad mental, dominio métrico y agudeza conceptual se enfrentaban ante públicos fervientes. La controversia se estructura como "diálogo amebeo" o discurso dialéctico entre dos improvisadores", donde cada poeta debe responder al contrario incorporando <strong>pies forzados</strong>-versos ajenos que deben cerrar su décima improvisación exigiendo maestría técnica y edición mental instantánea.
        </p>
    </div>
</Card>

{/* Modernización Masiva (Siglo XX) */}
<Card className="p-6 cuban-card bg-gradient-to-r from-[#F5E6D3] to-[#E6D7C1]">
    <div className="flex items-center gap-3 mb-4">
        <div className="text-3xl" />
    <div className="text-2xl font-bold text-[#C8A05C] font-serif">
        Modernización Masiva (Siglo XX)
    </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            La radio transformó la décima de arte regional en fenómeno nacional. Programas como <strong>"Competencia Nacional de Trovadores"</strong> (transmitido de 10 a 11 am) convirtieron a repentistas en figuras nacionales. <strong>Jesús Orta Ruiz (Indio Naborí)</strong> y <strong>Ángel Valiente</strong> protagonizaron la "Controversia del Siglo"-enfrentamiento que "interesó tanto al pueblo" que se convirtió en evento cultural definidor de la década de 1930. Estos dos poetas "son los más grandes repentistas cubanos del siglo XX, lo que equivale a decir que están entre los más grandes de la historia de la improvisación".
        </p>
        <p>
            Naborí y Valiente no solo demostraban virtuosismo: "glosaban la actualidad, daban un punto de reflexión sobre los problemas del país, llamaban a la conciencia colectiva, elevaban la atención"
        </p>
    </div>
</Card>

```

cotidiana a niveles de visión poética". La décima radiofónica funcionaba simultáneamente como entretenimiento, comentario social y educación poética para millones. Otros decimistas-Gustavo Tacoronte, Sergio Mederos, Pablo León-participaron en emisoras como Cadena Azul, Radio Programa Mil Diez, y espacios televisivos "Palmas y cañas" y "Meridiano campesino".

```

</p>
<p className="font-medium text-[#D2691E]">
    Esta masificación mediática no diluyó complejidad técnica sino la democratizó: oyentes aprendían a reconocer excelencia en tiempo real, apreciando cómo un repentista "debe encontrar el punto necesario de donde partir" para incorporar un pie forzado sin que "sea demasiado evidente el resultado". La radio creó audiencias sofisticadas capaces de discernir entre competencia rutinaria y genialidad improvisada.
</p>
</div>
</Card>

/* Presencia Latinoamericana */
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <div className="text-3xl"> </div>
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Presencia Latinoamericana
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            La décima no es exclusivamente cubana aunque Cuba la cultiva con intensidad singular. En Panamá, las décimas se cantan en "cantaderas" (viernes, sábados, domingos) acompañadas por guitarra española, violín, mejorana y rabel. En Venezuela y Méjico, Argentina, Chile, Puerto Rico reclaman la forma como propia.
        </p>
        <p className="italic font-medium text-[#C8A05C]">
            El poeta canario Pedro Lezcano sintetiza esta propiedad compartida: "Aunque el poeta inventa fuera Vicente Espinel, / la décima ya no es de él, / sino del pueblo cantor".
        </p>
    </div>
</Card>

/* Legado Vivo */
<Card className="p-6 cuban-card bg-gradient-to-br from-[#C8A05C]/10 to-[#D2691E]/10">
    <div className="flex items-center gap-3 mb-4">
        <div className="text-3xl"> </div>
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Legado Vivo
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            Cinco siglos después de que Espinel fijara su estructura, la décima fluye en tradición oral de millones que jamás leyeron Diversas rimas pero internalizaron su geometría como respiración. Poetas profesionales y campesinos improvisadores, académicos y repentistas callejeros comparten dominio de la misma arquitectura-diez versos que contienen universos. La forma persiste porque cumple función dual: permite tanto elaboración artística culta como expresión popular inmediata, sirve página impresa y escenario improvisado, sostiene meditaciones y reflexiones.
        </p>
    </div>
</Card>
```

```

        solitaria y competencia pública.
    </p>
    <p className="font-medium text-[#D2691E] text-center text-lg mt-6">
        La décima espinela demuestra que complejidad técnica y accesibilidad popular no se oponen.
        Rigor y espontaneidad. Tradición española y apropiación americana. Escritura erudita e
        improvisación oral. <strong>Ambas. Siempre ambas.</strong>
    </p>
</div>
</Card>

{/* Maestros de la Décima */}
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-6">
        <Award className="w-6 h-6 text-[#D2691E]" />
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Maestros de la Décima: Voces que Construyen Tradición
        </h3>
    </div>

    <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-3">
        {historicalFigures.map((poet, index) => (
            <div key={index} className="border-1-4 border-[#D2691E] pl-4 py-2">
                <button
                    onClick={() => setExpandedPoet(expandedPoet === poet ? null : poet)}
                    className="font-semibold text-[#5C4033] hover:text-[#C8A05C] transition-colors duration-200"
                >
                    {poet}
                    <span className="ml-1 text-[#D2691E] font-bold text-lg">*</span>
                </button>
            </div>
        )))
    </div>

```

```

{/* Biografía Expandible */}
{expandedPoet && (
    <div className="mt-6 p-6 bg-[#F5E6D3] border-2 border-[#C8A05C] rounded-lg text-sm leading-relaxed">
        {getPoetBiography(expandedPoet)}

        <button
            onClick={() => setExpandedPoet(null)}
            className="mt-6 px-4 py-2 bg-[#C8A05C] text-white rounded hover:bg-[#D2691E] transition-colors duration-200"
        >
            Cerrar
        </button>
    </div>
)}
</Card>

{/* Momentos de Excelencia: La Décima en su Esplendor */}
<Card className="p-6 cuban-card bg-gradient-to-r from-[#F5E6D3] to-[#E6D7C1]">
    <div className="flex items-center gap-3 mb-4">
        <div className="text-3xl"> </div>
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Momentos de Excelencia: La Décima en su Esplendor
        </h3>
    </div>

```

```

        </h3>
    </div>
<div className="text-[#5C4033] leading-relaxed space-y-4">
    <div>
        <h4 className="font-semibold text-[#D2691E] mb-3 text-lg flex items-center gap-2">
            <Badge className="bg-[#D2691E] text-white px-3 py-1">1955</Badge>
            Cuando el Arte Encontró su Voz
        </h4>
        <p>
            San Antonio de los Baños presenció algo extraordinario. <strong>El Indio Naborí</strong> ;<br/>
            <strong>Angelito Valiente</strong> transformaron el aire en arquitectura verbal, construyeron estrofas que desafiaban la gravedad del tiempo. Sus controversias radiadas no transmitían simplemente versos: irradiaban maestría, tejían conexión humana a través de ondas electrónicas que sembraban estándares que aún florecen en la memoria colectiva.
        </p>
        <p className="mt-3">
            Estos duelos poéticos representaron la síntesis perfecta entre tradición heredada y genio. Dos artistas, un formato ancestral, infinitas posibilidades. Las palabras fluían con la velocidad del relámpago y la precisión del cirujano. Cuba entera escuchaba, respiraba al ritmo de las décimas que participaba en un ritual que hacia del espectador un testigo de excelencia.
        </p>
    </div>
</div>
</Card>

{/* Cincuenta y Dos Segundos: Arquitectura Contra el Reloj */}
<Card className="p-6 cuban-card">
    <div className="flex items-center gap-3 mb-4">
        <div className="w-10 h-10 bg-[#D2691E] rounded-full flex items-center justify-center text-white font-bold text-2xl font-serif">
            52
        </div>
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            Cincuenta y Dos Segundos: Arquitectura Contra el Reloj
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            La décima perfecta emerge en <strong>52 segundos</strong>. Menos de un minuto para cristalizar un pensamiento en forma, emoción en estructura, espontaneidad en permanencia.
        </p>
        <p>
            Imagina el proceso: el decimista navega simultáneamente múltiples corrientes. Debe honrar la métrica octosílaba sin sacrificar naturalidad. Debe tejer rimas consonantes mientras mantiene la coherencia temática. Debe alcanzar belleza sin perder autenticidad. Todo esto mientras el cronómetro corre, implacable testigo de una proeza que fusiona velocidad mental con profundidad artística.
        </p>
        <p>
            Esta hazaña no es mera acrobacia verbal. Es demostración de que las restricciones no limitan la creatividad: la catalizan. El marco rígido de la décima, lejos de aprisionar la expresión, la concentra hasta alcanzar densidad diamantina. Presión y tiempo transforman el carbón del poema ordinario en la gema de la expresión memorable.
        </p>
        <p className="font-medium text-[#D2691E]">
            Solo los maestros habitan este espacio donde técnica y arte se funden indistinguiblemente.
        </p>
    </div>
</Card>

```

```

        la disciplina rigurosa habilita la libertad verdadera. Donde la velocidad no compromete la
        sino que la intensifica.
    </p>
</div>
</Card>

{/* UNESCO 2012: El Reconocimiento Universal */}
<Card className="p-6 cuban-card bg-gradient-to-br from-[#C8A05C]/10 to-[#D2691E]/10">
    <div className="flex items-center gap-3 mb-4">
        <div className="text-3xl"> </div>
        <h3 className="text-2xl font-bold text-[#C8A05C] font-serif">
            UNESCO 2012: El Reconocimiento Universal
        </h3>
    </div>
    <div className="text-[#5C4033] leading-relaxed space-y-4">
        <p>
            Cuando la <strong>UNESCO inscribió la décima como Patrimonio Cultural Inmaterial de la Humanidad</strong>, no celebraba simplemente una forma poética. Reconocía un ecosistema cultural vivo, respirante, que atravesía siglos sin perder vitalidad.
        </p>
        <p>
            La décima funciona como puente múltiple. Conecta épocas: Espinela en el Siglo de Oro español con decimistas contemporáneos en plazas latinoamericanas. Conecta estratos sociales: la misma décima que refinan los eruditos la dominan los improvisadores populares en festivales rurales. Conecta modalidades: la palabra escrita coexiste con la improvisación oral, cada una enriqueciendo a la otra.
        </p>
        <p>
            Este reconocimiento valida una verdad fundamental: ciertas formas artísticas contienen sabiduría que trasciende lo humano que trasciende contextos particulares. La décima porta códigos de identidad hispanoamericana, mientras permanece accesible a cualquier cultura que valore la precisión, la belleza y la memoria.
        </p>
        <p className="font-medium text-[#D2691E] italic">
            La décima vive. No como reliquia museística, sino como organismo que evoluciona sin traicionar su esencia. Tradición y renovación. Rigor y libertad. Ambas. Siempre ambas.
        </p>
        <p className="mt-4">
            Cada décima compuesta reactiva conexiones ancestrales mientras forja caminos nuevos. Cada intervención que enfrenta el desafío de los 52 segundos honra a Naborí y Valiente mientras crea algo único e irrepetible, suyo. Esta paradoja productiva-fidelidad absoluta al pasado que genera innovación constante-explica por qué la décima no solo sobrevive: <strong>prospera</strong>.
        </p>
    </div>
</Card>
</div>
);
}

```

components/education-section.tsx (primeras 100 líneas - archivo muy extenso)

```

"use client";

import React from "react";
import { Card } from "@components/ui/card";

```

```

import { Badge } from "@/components/ui/badge";
import { BookOpen, Crown, Users, Award } from "lucide-react";

export default function EducationSection() {
  const [expandedPoet, setExpandedPoet] = React.useState<string | null>(null);

  const historicalFigures = [
    "Vicente Espinel",
    "El Cucalambé",
    "Plácido",
    "El Indio Naborí",
    "Angelito Valiente",
    "Francisco Riverón Hernández",
    "Justo Vega",
    "Adolfo Alfonso",
    "Chanito Isidrón",
    "Omar Mirabal",
    "Tomasita Quiala",
    "Julio Pablo El Príncipe Pinareño"
  ];

  const getPoetBiography = (name: string) => {
    switch(name) {
      case "Vicente Espinel":
        return (
          <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
              Vicente Espinel (1550-1624): El Arquitecto Original
            </h4>
            <p>
              Vicente Gómez Martínez Espinel nació en Ronda, Málaga, el 28 de diciembre de 1550 y falleció en la misma ciudad el 15 de febrero de 1624. Fue un arquitecto y poeta español, considerado uno de los más importantes del Siglo de Oro.
            </p>
            <p>
              Su legado perdurable cristalizó en la décima: diez versos octosílabos con arquitectura de rima abba abba abba abba abba.
            </p>
            <p>
              Espinel poseía vastísima cultura. Dominaba el latín, siendo llamado "el único poeta latino de su tiempo".
            </p>
            <p>
              Su obra cumbre en prosa fue la novela picaresca <em>Relaciones de la vida del escudero Ma...
            </p>
          </div>
        );
      case "El Cucalambé":
        return (
          <div className="space-y-4">
            <h4 className="text-xl font-bold text-[#C8A05C] font-serif">
              El Cucalambé (Juan Cristóbal Nápoles Fajardo, 1829-1861): Primer Arquitecto de la Cubanización
            </h4>
            <p>
              Juan Cristóbal Nápoles Fajardo nació en Victoria de Las Tunas y desapareció misteriosamente en 1861.
            </p>
          </div>
        );
    }
  };
}

```

```

        Educado por su abuelo materno José Rafael Fajardo García-hacendado, cura y conspirador an
    </p>
    <p>
        Fue el único poeta que logró ser aceptado plenamente por el pueblo, entrando totalmente en
    </p>
    </div>
);

case "Plácido":
return (
<div className="space-y-4">
<h4 className="text-xl font-bold text-[#C8A05C] font-serif">
    Plácido (Gabriel de la Concepción Valdés, 1809-1844): Transparencia Trágica
</h4>
<p>
    Gabriel de la Concepción Valdés nació en La Habana, hijo de un peluquero mulato y una bai
</p>
<p>
    Huérfano y artesano, trabajó en múltiples oficios: carpintería, tipografía, fabricación de
</p>
<p>
    Su habilidad como artesano del carey lo llevó a Matanzas en 1826 para trabajar en el tall
</p>
<p>
    Fue fusilado el 28 de junio de 1844, acusado injustamente de ser jefe de la Conspiración o
</p>
</div>
);

case "El Indio Naborí":
return (
<div className="space-y-4">
<h4 className="text-xl font-bold text-[#C8A05C] font-serif">
    El Indio Naborí (Jesús Orta Ruiz, 1922-2005): Voz Lírica del Siglo XX
</h4>
<p>
    Jesús Orta Ruiz, nacido en Los Zapotes, San Miguel del Padrón, representa la figura más re
</p>
<p>
    Su etapa juvenil de repentina dejó huellas tan profundas que todavía se le invoca con em
</p>
);

// ... (continúa con biografías de todos los poetas)

```

components/about-modal.tsx

```

"use client";

import React from "react";
import { Dialog, DialogContent, DialogHeader, DialogTitle } from "@components/ui/dialog";
import { Badge } from "@components/ui/badge";
import { Heart, Globe, Code } from "lucide-react";

```

```

interface AboutModalProps {
  isOpen: boolean;
  onClose: () => void;
}

export default function AboutModal({ isOpen, onClose }: AboutModalProps) {
  const [showLeonorInfo, setShowLeonorInfo] = React.useState(false);

  return (
    <Dialog open={isOpen} onOpenChange={onClose}>
      <DialogContent className="max-w-2xl cuban-card border-2 border-[#C8A05C] bg-white max-h-[90vh] overflow-y-hidden">
        <DialogHeader>
          <DialogTitle className="text-2xl font-bold text-[#C8A05C] font-serif flex items-center gap-3">
            Acerca de El Guajiro de Hialeah
          </DialogTitle>
        </DialogHeader>

        <div className="space-y-6 text-[#5C4033]">
          {/* Calixto: Tejedor Entre Dos Orillas */}
          <div className="bg-[#F5E6D3] p-5 rounded-lg border-2 border-[#C8A05C]">
            <div className="flex items-center gap-3 mb-4">
              <Heart className="w-6 h-6 text-[#D2691E]" />
              <h3 className="text-xl font-bold text-[#C8A05C] font-serif">
                Calixto: Tejedor Entre Dos Orillas
              </h3>
            </div>
            <div className="space-y-4 text-sm leading-relaxed">
              <p>
                Calixto nació donde el cielo toca el agua-<strong>La Coloma</strong>, pueblo de pescadores. Entre barcas y sal, aprendió lo que el mar enseña: esperar es arte. Los pescadores lo saben-tejer redes lleva días, tejer palabras pide lo mismo.
              </p>
              <p>
                La décima le palpataba desde niño. Esa forma de diez versos que cruzó el océano y echó raíces en Cuba. Calixto se hizo trovador-no de rimas fáciles, sino de las que improvisan con el corazón. Creación colectiva, siempre.
              </p>
              <p>
                El exilio lo llevó a <strong>Hialeah</strong>. Allá le llamaron <strong>el Guajiro</strong> de verdad, fuerza tranquila, autenticidad que no se aprende. Exilio que fortalece.
              </p>
              <p>
                Su casa se volvió refugio. Abría la puerta, conseguía espacios en las radios, conectaba trovadores dispersos. Su voz baja pero firme les recordaba: la décima cura, guarda lo que no queremos perder.
              </p>
              <p className="font-medium text-[#D2691E]">
                <strong>Viajera Digital</strong> nace de ese legado. Preservamos cada décima usando tecnología que siente la pasión del poeta. Honramos la memoria de Calixto González y todos los poetas que mantienen viva esta tradición milenaria.
              </p>
            </div>
          </div>
        </div>
      </DialogContent>
    </Dialog>
  );
}

```

```

    </p>

    <p>
        Como las redes de La Coloma-pacientes, fuertes-Calixto tejió comunidad que no se
        deshace. Su casa vive en la memoria como ese lugar donde la décima encontró techo,
        donde trovadores se reconocieron hermanos.
    </p>

    <p>
        En esas redes quedó atrapada, felizmente, una viajera de España que llegó buscando
        versos y encontró raíces. Cuba no suelta a quien prueba el sabor de sus décimas.
        Ella quedó tejida en esta red-hebra europea en tapiz caribeño.
    </p>

    <p className="font-medium italic text-[#C8A05C]">
        <strong>Viajera Digital</strong> continúa ese tejido. Cada décima preservada es hebra.
        Cada poeta que aquí encuentra su voz es nudo. Cada generación que accede a este
        archivo teje hacia adelante. La tecnología sirve a la tradición.
    </p>

    <p className="text-center font-bold text-[#D2691E] mt-4 text-base">
        Para que la décima nunca sea arrancada del verde imán de nuestro suelo.
    </p>
    </div>
</div>

/* Closing Tribute */
<div className="text-center bg-[#FDF8F0] border-2 border-[#D2691E] p-4 rounded-lg">
    <p className="font-medium italic text-[#5C4033] leading-relaxed">
        "A todos los poetas que han dedicado sus vidas a mantener viva la décima:
        su voz sigue cantando en cada verso que procesamos aquí."
    </p>
</div>

/* Credits Section */
<div className="border-t-2 border-[#C8A05C] pt-4">
    <div className="grid md:grid-cols-2 gap-4 text-sm">
        <div>
            <div className="flex items-center gap-2 mb-2">
                <Code className="w-4 h-4 text-[#D2691E]" />
                <strong className="text-[#D2691E]">Desarrollo</strong>
            </div>
            <p>Creada por: <strong>Emilio José Novo</strong></p>
            <p className="text-[#5C4033]">Versión: 2.0 Pro</p>
        </div>
        <div>
            <div className="flex items-center gap-2 mb-2">
                <Globe className="w-4 h-4 text-[#C8A05C]" />
                <strong className="text-[#C8A05C]">Recursos</strong>
            </div>
            <p>Web: <strong>juanantoniodiaz.com</strong></p>
            <p className="text-[#5C4033]">
                Licencia: Creative Commons (Uso libre con atribución)
            </p>
        </div>
    </div>
</div>

```

```

        </p>
    </div>
</div>
</div>

/* Acknowledgments Section */
<div className="border-t-2 border-[#C8A05C] pt-4">
    <div className="text-sm">
        <p className="font-medium text-[#D2691E] mb-2">
            Agradecimientos por la creación a:
        </p>
        <p>
            <strong>Leonor Lopetegui</strong>
            <button
                onClick={() => setShowLeonorInfo(!showLeonorInfo)}
                className="ml-1 text-[#D2691E] hover:text-[#C8A05C] transition-colors duration-200 font-mediu
                aria-label="Ver más información sobre Leonor Lopetegui"
            >
                *
            </button>
        </p>

        {showLeonorInfo && (
            <div className="mt-4 p-4 bg-[#F5E6D3] border-2 border-[#C8A05C] rounded-lg space-y-3 text-[#D2691E] font-m
            <p>
                Leonor Lopetegui ha sido mi maestra de español, consejera y una abuela más que la v
            </p>

            <p>
                Con casi cincuenta años dedicados a la enseñanza del español, la Sra. Lopetegui no s
            </p>

            <p>
                También fue seleccionada Educadora Bilingüe del Año por la Asociación Bilingüe de la
            </p>

            <p>
                En reconocimiento a esta labor monumental, la Ciudad de Sweetwater proclamó el 6 de
            </p>
        </div>
    )}
</div>
</div>

/* Technical Info */
<div className="flex flex-wrap gap-2 justify-center">
    <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
        IA Claude Sonnet
    </Badge>
    <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
        NextJS 14
    </Badge>
    <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
        Transcripción de Audio
    </Badge>
</div>

```

```

        </Badge>
        <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
            Análisis Académico
        </Badge>
        <Badge variant="outline" className="text-[#5C4033] border-[#C8A05C]">
            Patrimonio UNESCO
        </Badge>
    </div>

    {/* Footer Note */}
    <div className="text-xs text-[#5C4033] text-center italic">
        "Un algoritmo que siente la pasión del poeta, rescata del olvido cada improvisación,
        preserva intacta la lírica del corazón, y eterniza el arte vivo de la décima completa."
    </div>
</div>
</DialogContent>
</Dialog>
);
}

```

components/prompt-modal.tsx

```

'use client';

import { Dialog, DialogContent, DialogHeader, DialogTitle, DialogDescription } from '@/components/ui/dialog';
import { Button } from '@/components/ui/button';
import { Copy, ExternalLink, Check } from 'lucide-react';
import { useState } from 'react';

interface PromptModalProps {
    open: boolean;
    onOpenChange: (open: boolean) => void;
    youtubeUrl: string;
}

const MASTER_PROMPT = `**ROL**: Eres un experto transcriptor especializado en poesía improvisada cubana

**TAREA PRINCIPAL**: Transcribir grabaciones de décimas improvisadas (controversias, polémicas, eventos)

**ESTRUCTURA OBLIGATORIA**:
- Esquema de rima: A B B A A C C D D C (rima consonante)
- Métrica: Versos octosílabos (8 sílabas por verso)
- Formato: Cada décima debe tener exactamente 10 versos

**INSTRUCCIONES ESPECÍFICAS**:
1. Identificación de poetas: Cuando el usuario especifique orden, organiza las décimas en ese orden exacto.

2. Proceso de transcripción: Identifica dónde comienza/termina cada décima. Convierte discurso oral en texto escrito.

3. Verificación de calidad: Cuenta sílabas (debe ser 8). Verifica rima consonante. Asegura coherencia temática.

4. Formato de presentación:
    **[1. Poeta X - Tema]**
```

Verso 1 (A) [8 sílabas]
Verso 2 (B) [8 sílabas]
[... hasta verso 10]

5. Manejo de contenido extenso: Organiza por secciones temáticas. Mantén numeración continua. Incluye sección de conclusiones.

6. Casos especiales: Ajusta con sinéresis/diéresis si es necesario. Busca rima consonante más cercana. Mantén la estructura de los versos.

IDIOMA: Español, respetando regionalismos cubanos.

PRIORIDAD: Precisión métrica > Rima exacta > Sentido literal

Después de transcribir todas las décimas, genera un análisis completo que incluya:

- Un resumen general de la canturía (temáticas, estilos, técnicas)
- Las top 3 décimas de cada poeta con análisis de: dominio técnico, recursos literarios, notas culturales
- Un análisis profundo del evento completo`;

```
export function PromptModal({ open, onOpenChange, youtubeUrl }: PromptModalProps) {  
  const [copied, setCopied] = useState(false);  
  
  const handleCopy = async () => {  
    try {  
      await navigator.clipboard.writeText(MASTER_PROMPT);  
      setCopied(true);  
      setTimeout(() => setCopied(false), 2000);  
    } catch (err) {  
      console.error('Error copiando:', err);  
    }  
  };  
  
  const handleOpenPerplexity = () => {  
    window.open('https://www.perplexity.ai', '_blank');  
  };  
  
  return (  
    <Dialog open={open} onOpenChange={onOpenChange}>  
      <DialogContent className="max-w-4xl max-h-[80vh] overflow-y-auto bg-[#F5E6D3] border-2 border-[#C8A05C]">  
        <DialogTitle className="text-2xl font-bold text-[#5C4033] mb-2">  
          Prompt Maestro para Perplexity  
        </DialogTitle>  
        <DialogDescription className="text-[#5C4033]/80">  
          Sigue estos pasos para obtener la transcripción y análisis completo:  
        </DialogDescription>  
      </DialogTitle>  
  
      <div className="space-y-6">  
        {/* Instrucciones paso a paso */}  
        <div className="bg-white/50 p-4 rounded-lg border border-[#C8A05C]/30">  
          <h3 className="font-bold text-[#5C4033] mb-3"> Pasos a seguir:</h3>  
          <ol className="space-y-2 text-sm text-[#5C4033]/90">
```

```

<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">1.</span>
    <span>Copia el prompt maestro abajo haciendo clic en el botón "Copiar Prompt"</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">2.</span>
    <span>Abre Perplexity.ai haciendo clic en "Abrir Perplexity"</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">3.</span>
    <span>Pega el prompt en el campo de entrada de Perplexity</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">4.</span>
    <span>Añade el enlace de YouTube: <span className="font-mono text-xs bg-white px-2 py-1"></span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">5.</span>
    <span>Asegúrate de seleccionar <strong>Claude Sonnet 4.5</strong> como modelo</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">6.</span>
    <span>Envía y espera la respuesta completa de Perplexity</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">7.</span>
    <span>Copia TODO el resultado generado por Perplexity</span>
</li>
<li className="flex gap-2">
    <span className="font-bold min-w-[24px]">8.</span>
    <span>Vuelve a esta app y pega el resultado en el campo correspondiente</span>
</li>
</ol>
</div>

/* Prompt copiable */
<div className="space-y-2">
    <div className="flex items-center justify-between">
        <h3 className="font-bold text-[#5C4033]">Prompt Maestro:</h3>
        <div className="flex gap-2">
            <Button
                onClick={handleCopy}
                variant="outline"
                size="sm"
                className="bg-[#C8A05C] text-white hover:bg-[#C8A05C]/90 border-none">
                {copied ? (
                    <>
                    <Check className="w-4 h-4 mr-2" />
                    Copiado
                </>
                ) : (
                    <>
                    <Copy className="w-4 h-4 mr-2" />
                </>
                )
            </Button>
        </div>
    </div>
</div>

```

```

        Copiar Prompt
      </>
    )
</Button>
<Button
  onClick={handleOpenPerplexity}
  variant="outline"
  size="sm"
  className="bg-[#5C4033] text-white hover:bg-[#5C4033]/90 border-none"
>
  <ExternalLink className="w-4 h-4 mr-2" />
  Abrir Perplexity
</Button>
</div>
</div>
<div className="bg-white p-4 rounded-lg border border-[#C8A05C]/30 max-h-96 overflow-y-auto">
  <pre className="text-xs text-[#5C4033] whitespace-pre-wrap font-mono leading-relaxed">
    {MASTER_PROMPT}
  </pre>
</div>
</div>
<!-- Nota importante -->
<div className="bg-[#C8A05C]/10 p-4 rounded-lg border border-[#C8A05C]/30">
  <p className="text-sm text-[#5C4033]">
    <strong> Nota:</strong> Este proceso usa tu suscripción de Perplexity existente,
    por lo que es completamente <strong>GRATIS</strong>. El app formateará automáticamente
    el resultado en las secciones de Décimas Formateadas, Análisis Completo y Exportación.
  </p>
</div>
</div>
</DialogContent>
</Dialog>
);
}

```

components/tribute-section.tsx

```

export default function TributeSection() {
  return (
    <section className="py-16 px-4 bg-gradient-to-b from-[#F5E6D3] to-white/90">
      <div className="max-w-4xl mx-auto text-center">
        <h2 className="text-4xl md:text-5xl font-bold text-[#C8A05C] mb-8 font-serif">
          Tributo a Calixto González, el Verdadero Guajiro de Hialeah
        </h2>

        <div className="cuban-card p-8 md:p-12 shadow-xl">
          <div className="decima-text text-lg md:text-xl leading-relaxed space-y-2">
            <p>Hoy que te vas con la muerte</p>
            <p>hacia una casa de cielo,</p>
            <p>la sequía del pañuelo</p>
            <p>en ríos se nos convierte.</p>
            <p>Y aunque no podamos verte</p>

```

```

        <p>jamás, en la canturía,</p>
        <p>no existirá un solo día</p>
        <p>que se hable de ejemplo de Hombre</p>
        <p>que no se escuche tu nombre,</p>
        <p className="font-bold text-[#D2691E]">Guajiro de Hialeah.</p>
    </div>

    <div className="mt-8 text-right">
        <p className="text-[#5C4033] font-semibold">- Juan Antonio Díaz</p>
    </div>
</div>
</div>
</section>
);
}

```

5. API Routes

api/validate-youtube/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { spawn } from 'child_process';

export const dynamic = "force-dynamic";

interface VideoInfo {
    id: string;
    title: string;
    duration: string;
    thumbnailUrl: string;
}

function extractYouTubeId(url: string): string | null {
    const patterns = [
        /(?:youtube\.com\/watch\?v=|youtu\.be\/|youtube\.com\/embed\/)([^&\n#]+)/,
        /youtube\.com\/watch\?.*v=([^&\n#]+)/
    ];

    for (const pattern of patterns) {
        const match = url.match(pattern);
        if (match?.[1]) {
            return match[1];
        }
    }

    return null;
}

function getVideoInfo(youtubeId: string): Promise<VideoInfo> {
    return new Promise((resolve, reject) => {
        const process = spawn('yt-dlp', [
            '--print',
            `{"id": "${(id)s", "title": "%(title)s", "duration": "%(duration_string)s", "thumbnail": "https://www.youtube.com/watch?v=${youtubeId}`
        ]);
    });
}

```

```

let stdout = '';
let stderr = '';

process.stdout.on('data', (data) => {
  stdout += data.toString();
});

process.stderr.on('data', (data) => {
  stderr += data.toString();
});

process.on('close', (code) => {
  if (code !== 0) {
    reject(new Error(`yt-dlp failed: ${stderr}`));
    return;
  }

  try {
    const lines = stdout.trim().split('\n');
    const lastLine = lines[lines.length - 1];
    const info = JSON.parse(lastLine);

    resolve({
      id: info.id,
      title: info.title,
      duration: info.duration || 'N/A',
      thumbnailUrl: info.thumbnail || `https://i.ytimg.com/vi/_uVtcQdw1-I/hq720.jpg?sqp=-oaymwEhCK4I
    });
  } catch (parseError) {
    reject(new Error('Failed to parse video information'));
  }
});
});

process.on('error', (error) => {
  reject(new Error(`Failed to execute yt-dlp: ${error.message}`));
});
});

export async function POST(request: NextRequest) {
  try {
    const { url } = await request.json();

    if (!url || typeof url !== 'string') {
      return NextResponse.json(
        { error: 'URL de YouTube requerida' },
        { status: 400 }
      );
    }
  }

  const youtubeId = extractYouTubeId(url.trim());

  if (!youtubeId) {

```

```

    return NextResponse.json(
      { error: 'URL de YouTube inválida. Asegúrate de usar un enlace válido de YouTube.' },
      { status: 400 }
    );
  }

  const videoInfo = await getVideoInfo(youtubeId);

  return NextResponse.json({
    youtubeId,
    title: videoInfo.title,
    duration: videoInfo.duration,
    thumbnailUrl: videoInfo.thumbnailUrl
  });

} catch (error) {
  console.error('YouTube validation error:', error);

  const errorMessage = error instanceof Error ? error.message : 'Error desconocido';

  if (errorMessage.includes('Video unavailable') || errorMessage.includes('Private video')) {
    return NextResponse.json(
      { error: 'Este video no está disponible o es privado' },
      { status: 404 }
    );
  }

  if (errorMessage.includes('network') || errorMessage.includes('timeout')) {
    return NextResponse.json(
      { error: 'Error de conexión. Verifica tu conexión a internet e inténtalo de nuevo.' },
      { status: 503 }
    );
  }

  return NextResponse.json(
    { error: 'Error al validar el video de YouTube. Inténtalo de nuevo.' },
    { status: 500 }
  );
}
}

```

api/process-video/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@/lib/db';
import { parsePerplexityResult, validateParsedResult } from '@/lib/perplexity-parser';
import { type DecimaSchema } from '@/lib/openrouter-client';

export const dynamic = "force-dynamic";
export const maxDuration = 60;

// Support both old Perplexity format and new OpenRouter format
interface ProcessVideoRequest {
  youtubeUrl: string;

```

```

youtubeId: string;
title: string;
duration: string;
thumbnailUrl: string;
perplexityResult?: string; // Legacy format
analysisResult?: {
  decimas: DecimaSchema[];
  resumen: {
    total_decimas: number;
    notas: string;
  };
}; // New OpenRouter format
}

export async function POST(request: NextRequest) {
  try {
    const body: ProcessVideoRequest = await request.json();
    const { youtubeUrl, youtubeId, title, duration, thumbnailUrl, perplexityResult, analysisResult } = body;

    if (!youtubeUrl || !youtubeId) {
      return NextResponse.json(
        { error: 'Faltan parámetros requeridos: youtubeUrl y youtubeId' },
        { status: 400 }
      );
    }

    // Determine which format we're processing
    let decimas: DecimaSchema[] = [];
    let resumenNotas = '';

    if (analysisResult) {
      // New OpenRouter format - already structured
      console.log('Processing OpenRouter format');
      decimas = analysisResult.decimas;
      resumenNotas = analysisResult.resumen.notas;

      if (!decimas || decimas.length === 0) {
        return NextResponse.json(
          { error: 'No se encontraron décimas en el resultado del análisis' },
          { status: 400 }
        );
      }
    }

    } else if (perplexityResult) {
      // Legacy Perplexity format - needs parsing
      console.log('Processing legacy Perplexity format');
      const parsedResult = parsePerplexityResult(perplexityResult);

      // Validate the result
      const validation = validateParsedResult(parsedResult, true);
      if (!validation.valid) {
        const errorMessage = validation.errors.join('\n');
        return NextResponse.json(
          {

```

```

        error: errorMessage,
        details: validation.errors
    },
    { status: 400 }
);
}

// Convert legacy format to new format
decimas = parsedResult.decimas.map((d, idx) => ({
    numero: idx + 1,
    poeta: d.poetName,
    versos: d.verses,
    analisis_metrico: {
        silabas_por_verso: Array(10).fill(8), // Assume 8 syllables
        sinalefas_detectadas: 0,
        palabras_finales: d.verses.map(v => v.split(/\s+/).pop() || '')
    },
    analisis_cultural: d.theme || 'Análisis pendiente'
}));
resumenNotas = parsedResult.analysis.summary;

} else {
    return NextResponse.json(
        { error: 'Se requiere perplexityResult o analysisResult' },
        { status: 400 }
    );
}

// Check if video already exists
let video = await prisma.video.findUnique({
    where: { youtubeUrl }
});

if (video) {
    // Delete existing data to reprocess
    await prisma.analysis.deleteMany({ where: { videoId: video.id } });
    await prisma.decima.deleteMany({ where: { videoId: video.id } });
    await prisma.poet.deleteMany({ where: { videoId: video.id } });
} else {
    // Create new video record
    video = await prisma.video.create({
        data: {
            youtubeUrl,
            youtubeId,
            title,
            duration,
            thumbnailUrl,
            status: 'ANALYZING'
        }
    });
}

// Group décimas by poet
const poetGroups = new Map<string, DecimaSchema[]>();

```

```

decimas.forEach(decima => {
  const existing = poetGroups.get(decima.poeta) || [];
  existing.push(decima);
  poetGroups.set(decima.poeta, existing);
});

// Save poets and décimas
let turnNumber = 1;
for (const [poetName, poetDecimas] of poetGroups.entries()) {
  const poet = await prisma.poet.create({
    data: {
      videoId: video.id,
      name: poetName,
      turnNumber: turnNumber++
    }
  });

  for (const decima of poetDecimas) {
    await prisma.decima.create({
      data: {
        videoId: video.id,
        poetId: poet.id,
        number: decima.numero,
        lines: decima.versos,
        rhymeScheme: 'ABBAACCDDC',
        meter: 'Octosílabos',
        timestamp: null,
        technicalMastery: `${decima.analisis_metrico.silabas_por_verso.filter(s => s === 8).length}`,
        poeticDevices: `${decima.analisis_metrico.sinalefas_detectadas} sinalefas detectadas`,
        culturalElements: decima.analisis_cultural,
        emotionalResonance: 'Análisis emocional pendiente'
      }
    });
  }
}

// Save analysis
await prisma.analysis.create({
  data: {
    videoId: video.id,
    summaryThemes: 'Temas de la tradición cubana',
    summaryTone: 'Tono característico',
    summaryInteraction: 'Interacción entre poetas',
    summaryQuality: 'Calidad técnica',
    recurringThemes: resumenNotas,
    regionalisms: 'Regionalismos cubanos',
    linguisticSophistication: 'Análisis lingüístico',
    improvisationTechniques: 'Técnicas de improvisación',
    canonComparison: 'Comparación con el canon',
    culturalSignificance: decimas.map(d => d.analisis_cultural).join('\n\n')
  }
});

// Update video status

```

```

    await prisma.video.update({
      where: { id: video.id },
      data: {
        status: 'COMPLETED',
        processedAt: new Date()
      }
    });

    return NextResponse.json({
      success: true,
      videoId: video.id,
      decimasCount: decimas.length,
      poetsCount: poetGroups.size
    });
  } catch (error: any) {
    console.error('Error procesando video:', error);
    return NextResponse.json(
      { error: error.message || 'Error al procesar el video' },
      { status: 500 }
    );
  }
}

```

api/video/[id]/decimas/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@/lib/db';

export const dynamic = "force-dynamic";

export async function GET(
  request: NextRequest,
  { params }: { params: { id: string } }
) {
  try {
    const videoId = params.id;

    const video = await prisma.video.findUnique({
      where: { id: videoId },
      include: {
        poets: {
          include: {
            decimas: {
              orderBy: { number: 'asc' }
            }
          },
          orderBy: { turnNumber: 'asc' }
        }
      }
    });

    if (!video) {

```

```

        return NextResponse.json(
          { error: 'Video no encontrado' },
          { status: 404 }
        );
      }

      if (video.status !== 'COMPLETED') {
        return NextResponse.json(
          { error: 'El video aún no ha sido procesado completamente' },
          { status: 400 }
        );
      }

      const response = {
        id: video.id,
        title: video.title,
        duration: video.duration,
        processedAt: video.processedAt,
        processingTime: video.processingTime,
        poets: video.poets.map(poet => ({
          id: poet.id,
          name: poet.name,
          turnNumber: poet.turnNumber,
          decimas: poet.decimas.map(decima => ({
            id: decima.id,
            number: decima.number,
            lines: decima.lines,
            rhymeScheme: decima.rhymeScheme,
            meter: decima.meter,
            isComplete: decima.isComplete,
            isRegular: decima.isRegular,
            timestamp: decima.timestamp
          }))
        }))
      });

      return NextResponse.json(response);
    } catch (error) {
      console.error('Error fetching decimas:', error);
      return NextResponse.json(
        { error: 'Error interno del servidor' },
        { status: 500 }
      );
    }
  }
}

```

api/video/[id]/analysis/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@lib/db';

export const dynamic = "force-dynamic";

```

```

export async function GET(
  request: NextRequest,
  { params }: { params: { id: string } }
) {
  try {
    const videoId = params.id;

    const video = await prisma.video.findUnique({
      where: { id: videoId },
      include: {
        analysis: true,
        poets: {
          include: {
            decimas: {
              where: { isTopThree: true },
              orderBy: { topThreeRank: 'asc' }
            }
          }
        }
      }
    });
  }

  if (!video) {
    return NextResponse.json(
      { error: 'Video no encontrado' },
      { status: 404 }
    );
  }

  if (video.status !== 'COMPLETED') {
    return NextResponse.json(
      { error: 'El análisis aún no ha sido completado' },
      { status: 400 }
    );
  }

  const topDecimasByPoet: Record<string, any[]> = {};

  video.poets.forEach(poet => {
    if (poet.decimas.length > 0) {
      topDecimasByPoet[poet.name] = poet.decimas.map(decima => ({
        id: decima.id,
        number: decima.number,
        lines: decima.lines,
        poet: poet.name,
        rank: decima.topThreeRank || 0,
        technicalMastery: decima.technicalMastery,
        poeticDevices: decima.poeticDevices,
        culturalElements: decima.culturalElements,
        emotionalResonance: decima.emotionalResonance,
        improvisation: decima.improvisation,
        topThreeReason: decima.topThreeReason
      }));
    }
  });
}

```

```

        }
    });

const response = {
    id: video.id,
    videoTitle: video.title,
    summaryThemes: video.analysis?.summaryThemes,
    summaryTone: video.analysis?.summaryTone,
    summaryInteraction: video.analysis?.summaryInteraction,
    summaryQuality: video.analysis?.summaryQuality,
    recurringThemes: video.analysis?.recurringThemes,
    regionalisms: video.analysis?.regionalisms,
    linguisticSophistication: video.analysis?.linguisticSophistication,
    improvisationTechniques: video.analysis?.improvisationTechniques,
    canonComparison: video.analysis?.canonComparison,
    culturalSignificance: video.analysis?.culturalSignificance,
    topDecimasByPoet
};

return NextResponse.json(response);
}

} catch (error) {
    console.error('Error fetching analysis:', error);
    return NextResponse.json(
        { error: 'Error interno del servidor' },
        { status: 500 }
    );
}
}

```

api/video/[id]/export/text/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@/lib/db';

export const dynamic = "force-dynamic";

export async function GET(
    request: NextRequest,
    { params }: { params: { id: string } }
) {
    try {
        const videoId = params.id;

        const video = await prisma.video.findUnique({
            where: { id: videoId },
            include: {
                analysis: true,
                poets: {
                    include: {
                        decimas: {
                            orderBy: { number: 'asc' }
                        }
                    }
                }
            }
        });
    }
}

```

```

        },
        orderBy: { turnNumber: 'asc' }
    }
}
});

if (!video) {
    return NextResponse.json(
        { error: 'Video no encontrado' },
        { status: 404 }
    );
}

if (video.status !== 'COMPLETED') {
    return NextResponse.json(
        { error: 'El video aún no ha sido procesado completamente' },
        { status: 400 }
    );
}

// Generate text content
let textContent = '';

// Video information
textContent += `INFORMACIÓN DEL VIDEO\n`;
textContent += `=====`;
textContent += `Título: ${video.title}\n`;
textContent += `Duración: ${video.duration}\n`;
textContent += `Fecha de análisis: ${video.processedAt.toLocaleDateString('es-ES')}\n`;
textContent += `Poetas identificados: ${video.poets.map(p => p.name).join(', ')}\n\n`;

// Décimas by poet
textContent += `DÉCIMAS FORMATEADAS\n`;
textContent += `=====`;
textContent += `-----\n\n`;

video.poets.forEach(poet => {
    textContent += `${poet.name.toUpperCase()}${poet.turnNumber ? ` - Turno ${poet.turnNumber}` : ''}`;
    textContent += `-${'='.repeat(poet.name.length + 10)}\n\n`;

    poet.decimas.forEach(decima => {
        textContent += `Décima ${decima.number}:\n`;
        decima.lines.forEach(line => {
            textContent += `${line}\n`;
        });
        textContent += `\nEsquema de rima: ${decima.rhymeScheme}\n`;
        textContent += `Métrica: ${decima.meter}\n`;
        if (decima.timestamp) {
            textContent += `Marca de tiempo: ${decima.timestamp}\n`;
        }
        textContent += '\n';
    });
});
}

// Analysis summary

```

```

if (video.analysis) {
    textContent += `ANÁLISIS COMPLETO\n`;
    textContent += `=====\\n\\n`;

    if (video.analysis.summaryThemes) {
        textContent += `Temas Principales:\\n${video.analysis.summaryThemes}\\n\\n`;
    }

    if (video.analysis.summaryTone) {
        textContent += `Tono y Energía Emocional:\\n${video.analysis.summaryTone}\\n\\n`;
    }

    if (video.analysis.summaryInteraction) {
        textContent += `Interacción entre Poetas:\\n${video.analysis.summaryInteraction}\\n\\n`;
    }

    if (video.analysis.summaryQuality) {
        textContent += `Calidad Poética e Impacto:\\n${video.analysis.summaryQuality}\\n\\n`;
    }

    // Deep analysis
    textContent += `ANÁLISIS PROFUNDO\\n`;
    textContent += `=====\\n\\n`;

    const deepAnalysisFields = [
        { key: 'recurringThemes', label: 'Temas Recurrentes' },
        { key: 'regionalisms', label: 'Regionalismos y Dialectos' },
        { key: 'linguisticSophistication', label: 'Sofisticación Lingüística' },
        { key: 'improvisationTechniques', label: 'Técnicas de Improvisación Observadas' },
        { key: 'canonComparison', label: 'Comparación con el Canon Cubano' },
        { key: 'culturalSignificance', label: 'Significancia Cultural' }
    ];

    deepAnalysisFields.forEach(field => {
        const value = video.analysis?[field.key as keyof typeof video.analysis];
        if (value) {
            textContent += `${field.label}:\\n${value}\\n\\n`;
        }
    });
}

const response = {
    title: video.title,
    duration: video.duration,
    processedAt: video.processedAt,
    textContent: textContent.trim()
};

return NextResponse.json(response);

} catch (error) {
    console.error('Error generating text export:', error);
    return NextResponse.json(
        { error: 'Error interno del servidor' },

```

```

        { status: 500 }
    );
}
}

```

api/video/[id]/export/pdf/route.ts

```

import { NextRequest, NextResponse } from 'next/server';

export const dynamic = "force-dynamic";

export async function POST(
    request: NextRequest,
    { params }: { params: { id: string } }
) {
    try {
        const videoId = params.id;

        // Get the text content first
        const baseUrl = request.headers.get('host') || 'localhost:3000';
        const protocol = request.headers.get('x-forwarded-proto') || 'http';
        const textResponse = await fetch(`:${protocol}://${baseUrl}/api/video/${videoId}/export/text`);

        if (!textResponse.ok) {
            throw new Error('Failed to get video content');
        }

        const textData = await textResponse.json();

        // Create a simple PDF content structure for client-side generation
        // Since we're using client-side PDF generation, we return the structured data
        const pdfData = {
            title: textData.title,
            content: textData.textContent,
            metadata: {
                author: 'El Guajiro de Hialeah',
                subject: 'Análisis de Décima Espinela Cubana',
                keywords: 'décima, espinela, cuba, poesía, análisis',
                creator: 'Guajiro de Hialeah v2.0',
                producer: 'El Guajiro de Hialeah'
            },
            styling: {
                primaryColor: '#C8A05C',
                secondaryColor: '#5C4033',
                backgroundColor: '#F5E6D3'
            }
        };

        return NextResponse.json(pdfData);
    } catch (error) {
        console.error('Error generating PDF data:', error);
        return NextResponse.json(

```

```

        { error: 'Error al generar PDF' },
        { status: 500 }
    );
}
}

```

api/video/[id]/share/route.ts

```

import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@lib/db';
import { randomBytes } from 'crypto';

export const dynamic = "force-dynamic";

export async function POST(
    request: NextRequest,
    { params }: { params: { id: string } }
) {
    try {
        const videoId = params.id;

        // Verify video exists and is completed
        const video = await prisma.video.findUnique({
            where: { id: videoId }
        });

        if (!video) {
            return NextResponse.json(
                { error: 'Video no encontrado' },
                { status: 404 }
            );
        }

        if (video.status !== 'COMPLETED') {
            return NextResponse.json(
                { error: 'El video aún no ha sido procesado completamente' },
                { status: 400 }
            );
        }

        // Generate a unique share ID
        const shareId = randomBytes(16).toString('hex');

        // In a full implementation, you would store share links in a separate table
        // For now, we'll return the share ID that can be used to construct the URL
        // The share route would fetch the video by ID

        return NextResponse.json({
            shareId: videoId, // Using videoId directly for simplicity
            videoId,
            expiresAt: null // No expiration for now
        });
    }
}

```

```

    } catch (error) {
      console.error('Error creating share link:', error);
      return NextResponse.json(
        { error: 'Error interno del servidor' },
        { status: 500 }
      );
    }
}

```

6. Librerías y Utilidades

lib/db.ts

```

import { PrismaClient } from '@prisma/client'

const globalForPrisma = globalThis as unknown as {
  prisma: PrismaClient | undefined
}

export const prisma = globalForPrisma.prisma ?? new PrismaClient()

if (process.env.NODE_ENV !== 'production') globalForPrisma.prisma = prisma

```

lib/utils.ts

```

import { type ClassValue, clsx } from "clsx"
import { twMerge } from "tailwind-merge"

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs))
}

export function formatDuration(seconds: number): string {
  const hours = Math.floor(seconds / 3600)
  const minutes = Math.floor((seconds % 3600) / 60)
  const remainingSeconds = seconds % 60

  return `${hours.toString().padStart(2, '0')}:${minutes.toString().padStart(2, '0')}:${remainingSeconds}```

}

```

lib/types.ts

```

```ts
export type Expense = {
 id: string
 amount: number
 category: string
 description: string
 date: Date
}

export type ExpenseFormData = Omit<Expense, 'id' | 'date'> & {
 date: string
}

```

```

export const EXPENSE_CATEGORIES = [
 'Food',
 'Transportation',
 'Housing',
 'Utilities',
 'Entertainment',
 'Healthcare',
 'Shopping',
 'Education',
 'Other'
] as const

export type DateRange = {
 from: Date | undefined
 to: Date | undefined
}```
```ts

/**
 * Parser para convertir el resultado de Perplexity en datos estructurados
 */

export interface ParsedDecima {
  poetName: string;
  title: string;
  verses: string[];
  rhymeScheme: string;
  timestamp?: string;
  theme?: string;
}

export interface ParsedTopDecima {
  poetName: string;
  decimaNumber: number;
  verses: string[];
  mastery: string;
  literaryDevices: string;
  culturalNotes: string;
}

export interface ParsedAnalysis {
  summary: string;
  topDecimas: ParsedTopDecima[];
  deepAnalysis: string;
}

export interface ParsedResult {
  decimas: ParsedDecima[];
  analysis: ParsedAnalysis;
}

```

```

    rawText: string;
}

/*
 * Intenta extraer décimas de forma más simple cuando el formato principal falla
 * Busca bloques de 8-10 líneas consecutivas que parezcan versos
 */
function fallbackParsing(text: string): ParsedDecima[] {
  const decimas: ParsedDecima[] = [];
  const lines = text.split('\n').map(l => l.trim()).filter(l => l.length > 0);

  let currentVerses: string[] = [];
  let decimaNumber = 1;

  for (let i = 0; i < lines.length; i++) {
    const line = lines[i];

    // Ignorar líneas obvias que no son versos
    const isNotVerse =
      line.length > 120 ||
      line.match(/^(\resumen|\análisis|\top\s+\d|nota|observ|\contexto|métrica|rima|décima\s+\d)/i) ||
      line.startsWith('---') ||
      line.startsWith('***') ||
      line.startsWith('###');

    if (isNotVerse) {
      // Si tenemos 8-10 versos acumulados, guardar como décima
      if (currentVerses.length >= 8 && currentVerses.length <= 10) {
        decimas.push({
          poetName: `Poeta ${Math.ceil(decimaNumber / 2)}`,
          title: `Décima ${decimaNumber}`,
          verses: currentVerses,
          rhymeScheme: 'ABBAACCDCC',
          theme: 'Tema tradicional'
        });
        decimaNumber++;
      }
      currentVerses = [];
      continue;
    }

    // Limpiar el verso
    let cleanVerse = line
      .replace(/\[8 sílabas\]/gi, '')
      .replace(/\[\d+\s*s[í]labas\]/gi, '')
      .replace(/\([ABCDE]\)/gi, '')
      .replace(/\^d+[\.\.]\s*/i, '')
      .replace(/\^[\---•]\s*/i, '')
      .replace(/\*\*/g, '')
      .replace(/\^>\s*/g, '')
      .trim();
  }

  // Validar que parezca un verso
  const wordCount = cleanVerse.split(/\s+/).filter(w => w.length > 0).length;
}


```

```

const hasLetters = /[a-zAÉÍÓÚñü]/i.test(cleanVerse);

if (cleanVerse.length >= 8 && cleanVerse.length <= 100 && wordCount >= 2 && wordCount <= 20 && hasLetters) {
    currentVerses.push(cleanVerse);

    // Si tenemos 10 versos, cerrar la décima
    if (currentVerses.length === 10) {
        decimas.push({
            poetName: `Poeta ${Math.ceil(decimaNumber / 2)}`,
            title: `Décima ${decimaNumber}`,
            verses: currentVerses,
            rhymeScheme: 'ABBAACCDDC',
            theme: 'Tema tradicional'
        });
        decimaNumber++;
        currentVerses = [];
    }
} else if (currentVerses.length > 0) {
    // Si el verso no es válido pero tenemos versos acumulados
    // y ya tenemos suficientes (8-10), guardar la décima
    if (currentVerses.length >= 8 && currentVerses.length <= 10) {
        decimas.push({
            poetName: `Poeta ${Math.ceil(decimaNumber / 2)}`,
            title: `Décima ${decimaNumber}`,
            verses: currentVerses,
            rhymeScheme: 'ABBAACCDDC',
            theme: 'Tema tradicional'
        });
        decimaNumber++;
    }
    currentVerses = [];
}

// Guardar última décima si existe
if (currentVerses.length >= 8 && currentVerses.length <= 10) {
    decimas.push({
        poetName: `Poeta ${Math.ceil(decimaNumber / 2)}`,
        title: `Décima ${decimaNumber}`,
        verses: currentVerses,
        rhymeScheme: 'ABBAACCDDC',
        theme: 'Tema tradicional'
    });
}

return decimas;
}

/**
 * Parsea el resultado de Perplexity para extraer décimas
 * Maneja múltiples formatos posibles de respuesta
 */
export function parsePerplexityResult(text: string): ParsedResult {
    const decimas: ParsedDecima[] = [];

```

```

const lines = text.split('\n');

let currentDecima: Partial<ParsedDecima> | null = null;
let currentVerses: string[] = [];
let lineIndex = 0;

// Múltiples patrones para detectar encabezados de décimas
const headerPatterns = [
    // [1. Poeta - Tema] o **[1. Poeta - Tema]**
    /^[\*\s]*\[([d+][.\.])\]\s*([^\-\n]+?)\s*[--:]|\s*([^\-\n]+?)\] [\*\s]*$/i,
    // **1. Poeta - Tema** o #1. Poeta - Tema
    /^[\*\#\s]+([d+][.\.])\]\s*([^\-\n]+?)\s*[--:]|\s*([^\n]+?) [\*\#\s]*$/i,
    // Décima 1: Poeta - Tema
    /(^d[eé] cima\s+([d+][.:.\.])\]\s*([^\-\n]+?)\s*[--:]|\s*([^\n]+?)$/i,
    // 1) Poeta - Tema (sin corchetes ni markdown)
    /(^([d+])\.\.\.]\s+([^\-\n]+?)\s*[--:]|\s*([^\n]+?)$/i,
    // 1. Poeta - Tema (sin adornos)
    /(^([d+])\.\.\.]\s+([^\-\n]+?)\s*[--:]|\s*([.]+?)$/i,
    // **Poeta - Tema** (sin numeración)
    /(^[\*\*]([^\-\n]+?)\s*[--:]|\s*([^\*\n]+?)\*\*$/i,
];
]

while (lineIndex < lines.length) {
    const line = lines[lineIndex].trim();

    // Intentar detectar encabezado con múltiples patrones
    let headerMatch = null;
    for (const pattern of headerPatterns) {
        headerMatch = line.match(pattern);
        if (headerMatch) break;
    }

    if (headerMatch) {
        // Guardar décima anterior si existe y tiene suficientes versos
        if (currentDecima && currentVerses.length >= 8) {
            // Aceptar décimas con 8-10 versos (más flexible)
            decimas.push({
                poetName: currentDecima.poetName || 'Poeta Desconocido',
                title: currentDecima.title || 'Sin título',
                verses: currentVerses.slice(0, 10),
                rhymeScheme: 'ABBAACCCDC',
                theme: currentDecima.theme
            });
        }
    }

    // Iniciar nueva décima
    // Verificar si el patrón capturó 3 o 2 grupos (con o sin numeración)
    let poetName: string;
    let theme: string;

    if (headerMatch[3]) {
        // Patrón con numeración: grupo 2 = poeta, grupo 3 = tema
        poetName = headerMatch[2].trim().replace(/\*\*/g, '').replace(/\#+\s*/g, '');
        theme = headerMatch[3].trim().replace(/\*\*/g, '');
    }
}

```

```

} else {
    // Patrón sin numeración: grupo 1 = poeta, grupo 2 = tema
    poetName = headerMatch[1].trim().replace(/\*\*/g, '').replace(/^#+\s*/g, '');
    theme = headerMatch[2].trim().replace(/\*\*/g, '');
}

currentDecima = {
    poetName: poetName,
    title: theme,
    theme: theme
};
currentVerses = [];
lineIndex++;
continue;
}

// Detectar versos cuando hay una décima activa
if (currentDecima && line.length > 0) {
    // Ignorar líneas que claramente no son versos
    const isNotVerse =
        line.startsWith('---') ||
        line.startsWith('===') ||
        line.startsWith('***') ||
        line.startsWith('###') ||
        line.startsWith('//') ||
        line.match(/^[\*#]+\s/i) ||
        line.match(/^(\resumen|\análisis|top\s+\d|nota|observ|contexto|métrica|rima)/i) ||
        line.length > 120; // Los versos no suelen ser muy largos

    if (!isNotVerse) {
        // Limpiar el verso de marcadores comunes
        let cleanVerse = line
            .replace(/\[8\silabas\]/gi, '')
            .replace(/\[(d+\s*s[ií]labas\]/gi, '')
            .replace(/\([ABCDE]\)/gi, '')
            .replace(/\s+\([ABCDE]\)\s+$/gi, '')
            .replace(/^\d+[.\.\.]\s*/i, '') // Remover numeración de versos
            .replace(/^---\s*/i, '') // Remover guiones y bullets al inicio
            .replace(/\*\*\*/g, '') // Remover negritas markdown
            .replace(/\>\s*/g, '') // Remover marcador de cita
            .trim();

        // Validar que parezca un verso (longitud razonable, contiene palabras)
        const wordCount = cleanVerse.split(/\s+/).filter(w => w.length > 0).length;
        const hasLetters = /[a-zA-ZÁÉÍÓÚÑÜ]/i.test(cleanVerse);

        if (cleanVerse.length >= 8 && cleanVerse.length <= 100 && wordCount >= 2 && wordCount <= 20 && !hasLetters) {
            if (currentVerses.length < 10) {
                currentVerses.push(cleanVerse);
            }
        }
    }
}

// Si encontramos un separador después de tener versos, cerrar la décima

```

```

if ((line === '---' || line === '***' || line === '') && currentVerses.length >= 8) {
    // Verificar si la siguiente línea no vacía es un nuevo encabezado
    let nextNonEmptyLine = '';
    for (let j = lineIndex + 1; j < lines.length; j++) {
        if (lines[j].trim().length > 0) {
            nextNonEmptyLine = lines[j].trim();
            break;
        }
    }

    let isNextHeader = false;
    for (const pattern of headerPatterns) {
        if (nextNonEmptyLine.match(pattern)) {
            isNextHeader = true;
            break;
        }
    }

    if (isNextHeader) {
        // Guardar y cerrar décima actual
        decimas.push({
            poetName: currentDecima.poetName || 'Poeta Desconocido',
            title: currentDecima.title || 'Sin título',
            verses: currentVerses.slice(0, 10),
            rhymeScheme: 'ABBAACCDCC',
            theme: currentDecima.theme
        });
        currentDecima = null;
        currentVerses = [];
    }
}

lineIndex++;
}

// Guardar última décima si existe
if (currentDecima && currentVerses.length >= 8) {
    decimas.push({
        poetName: currentDecima.poetName || 'Poeta Desconocido',
        title: currentDecima.title || 'Sin título',
        verses: currentVerses.slice(0, 10),
        rhymeScheme: 'ABBAACCDCC',
        theme: currentDecima.theme
    });
}

// Si no encontramos décimas con el método principal, intentar fallback
let finalDecimas = decimas;
if (decimas.length === 0) {
    console.log('Parsing principal no encontró décimas, intentando fallback...');

    finalDecimas = fallbackParsing(text);
    if (finalDecimas.length > 0) {
        console.log(`Fallback encontró ${finalDecimas.length} décimas`);
    }
}

```

```

        }

    }

    // Parsear análisis
    const analysis = parseAnalysisSection(text);

    return {
        decimas: finalDecimas,
        analysis,
        rawText: text
    };
}

/**
 * Parsea la sección de análisis del resultado
 */
function parseAnalysisSection(text: string): ParsedAnalysis {
    const lowerText = text.toLowerCase();

    // Buscar sección de resumen
    let summary = '';
    const summaryPatterns = [
        /resumen\s+general[:\s]+([\s\S]*?)\(?=top\s+3|análisis\s+profundo|\n\n\n)/i,
        /resumen[:\s]+([\s\S]*?)\(?=top\s+3|análisis\s+profundo|\n\n\n)/i,
        /análisis\s+general[:\s]+([\s\S]*?)\(?=top\s+3|análisis\s+profundo|\n\n\n)/i
    ];

    for (const pattern of summaryPatterns) {
        const match = text.match(pattern);
        if (match && match[1].trim().length > 50) {
            summary = match[1].trim();
            break;
        }
    }

    // Buscar sección de top décimas
    const topDecimas: ParsedTopDecima[] = [];
    const topSectionPattern = /top\s+3[^:]*:([\s\S]*?)\(?=análisis\s+profundo|\$/i;
    const topMatch = text.match(topSectionPattern);

    if (topMatch) {
        const topSection = topMatch[1];
        // Extraer cada décima top (esto es simplificado, se puede mejorar)
        const decimaEntries = topSection.split(/\n\n+/).filter(s => s.trim().length > 100);

        decimaEntries.forEach(entry, idx) => {
            if (idx < 6) { // Max 3 por poeta, típicamente 2 poetas = 6 total
                const poetMatch = entry.match(/(^:)+:/);
                const versesMatch = entry.match(/verso[s]?[:\s]+([\s\S]*?)\(?=dominio|mastery|\$/i);
                const masteryMatch = entry.match(/(^:dominio|mastery)[:\s]+(.*)\(?=\n|recursos|\$/i);
                const devicesMatch = entry.match(/recursos\s+literarios[:\s]+(.*)\(?=\n|notas|\$/i);
                const notesMatch = entry.match(/notas\s+culturales[:\s]+(.*)\(?=\n|\$/i);

                if (poetMatch) {

```

```

        topDecimas.push({
          poetName: poetMatch[1].trim(),
          decimalNumber: idx + 1,
          verses: versesMatch ? versesMatch[1].trim().split('\n').slice(0, 10) : [],
          mastery: masteryMatch ? masteryMatch[1].trim() : 'Excelente dominio técnico',
          literaryDevices: devicesMatch ? devicesMatch[1].trim() : 'Uso efectivo de recursos literarios',
          culturalNotes: notesMatch ? notesMatch[1].trim() : 'Refleja la tradición cubana'
        });
      }
    });
  });

// Buscar análisis profundo
let deepAnalysis = '';
const deepPattern = /análisis\s+profundo[:\s]+([\s\S]*)$/i;
const deepMatch = text.match(deepPattern);

if (deepMatch) {
  deepAnalysis = deepMatch[1].trim();
}

return {
  summary: summary || 'Análisis de la canturía disponible en el texto completo.',
  topDecimas,
  deepAnalysis: deepAnalysis || 'Análisis profundo disponible en el texto completo.'
};

/** 
 * Valida que el resultado parseado sea válido
 */
export function validateParsedResult(result: ParsedResult, showDebug: boolean = false): { valid: boolean, errors: string[], warnings: string[] } {
  const errors: string[] = [];
  const warnings: string[] = [];

  if (!result.decimas || result.decimas.length === 0) {
    errors.push(' No se encontraron décimas en el resultado.');
    errors.push('');
    errors.push(' Asegúrate de que tu respuesta de Perplexity incluye:');
    errors.push('   • Décimas numeradas con formato: "1. Poeta - Tema" o "[1. Poeta - Tema]"');
    errors.push('   • Cada décima debe tener 10 versos (líneas de poesía)');
    errors.push('   • Separadores entre décimas (línea vacía o "---")');
    errors.push('');
    errors.push(' Ejemplo del formato esperado:');
    errors.push('   **[1. Calixto González - El Amor al Campo]**');
    errors.push('   ');
    errors.push('   En el campo yo nací');
    errors.push('   donde el sol siempre relumbra');
    errors.push('   ... (8 versos más)');
    errors.push('   ');
    errors.push('   ---');
  }

  // Información de debug si está habilitada
}

```

```

    if (showDebug && result.rawText) {
      errors.push('');
      errors.push(' Debug: Primeras 500 caracteres del texto recibido:');
      errors.push(result.rawText.substring(0, 500) + '...');
      errors.push('');
      errors.push(' Estadísticas del texto:');
      errors.push(`  • Total de líneas: ${result.rawText.split('\n').length}`);
      errors.push(`  • Total de caracteres: ${result.rawText.length}`);
      const numberedLines = result.rawText.split('\n').filter(l => /^\\d+[\\.\\)]\\s/.test(l.trim()));
      errors.push(`  • Líneas con numeración (1. 2. etc): ${numberedLines.length}`);
    }
  }

  if (result.decimas.length > 0) {
    const invalidDecimas = result.decimas.filter(d => d.verses.length < 8 || d.verses.length > 10);
    if (invalidDecimas.length > 0) {
      warnings.push(` Se encontraron ${invalidDecimas.length} décimas con número incorrecto de versos.`);
      warnings.push(` Las décimas deben tener exactamente 10 versos (actualmente algunas tienen ${invalidDecimas.length} versos).`);
    }
  }

  // Verificar si hay décimas sin nombre de poeta
  const anonymousDecimas = result.decimas.filter(d => !d.poetName || d.poetName === 'Poeta Desconocido');
  if (anonymousDecimas.length > 0) {
    warnings.push(` ${anonymousDecimas.length} décimas no tienen nombre de poeta identificado.`);
  }
}

// El análisis es opcional pero es bueno tenerlo
if (!result.analysis.summary || result.analysis.summary.length < 30) {
  warnings.push(' No se encontró un resumen de análisis completo (esto es opcional pero recomendado).');
}

// Si tenemos décimas pero con warnings, seguir adelante
const hasDecimas = result.decimas && result.decimas.length > 0;

return {
  valid: hasDecimas && errors.length === 0,
  errors,
  warnings
};
}

```

lib/syllable-counter.ts

```

/**
 * Spanish Syllable Counter with Sinalefa Detection
 * Implements Spanish prosodic rules for décima métrica
 */

export interface Sinalefa {
  position: string; // "word1_word2"
  reduction: number; // Usually 1
  words: [string, string];
}

```

```

}

export interface SyllableCount {
  words: number;
  approximateSyllables: number;
  sinalefas: Sinalefa[];
  stressAdjustment: number; // -1, 0, or +1
  finalWordStress: 'aguda' | 'grave' | 'esdrujula' | 'unknown';
  isValid: boolean; // true if approximateSyllables === 8
  breakdown: string; // Human-readable explanation
}

/**
 * Count base syllables in a word (rough approximation based on vowels)
 */
function countBaseVowels(word: string): number {
  const vowels = ['a', 'e', 'i', 'o', 'u', 'á', 'é', 'í', 'ó', 'ú', 'ü'];
  const lower = word.toLowerCase();
  let count = 0;

  for (let i = 0; i < lower.length; i++) {
    if (vowels.includes(lower[i])) {
      count++;
    }
  }

  return count;
}

/**
 * Check if word ends with a vowel
 */
function endsWithVowel(word: string): boolean {
  const vowels = ['a', 'e', 'i', 'o', 'u', 'á', 'é', 'í', 'ó', 'ú'];
  const cleaned = word.replace(/[,;:!?'¿'¡'()[]{}]/g, '').toLowerCase();
  return cleaned.length > 0 && vowels.includes(cleaned[cleaned.length - 1]);
}

/**
 * Check if word starts with a vowel
 */
function startsWithVowel(word: string): boolean {
  const vowels = ['a', 'e', 'i', 'o', 'u', 'á', 'é', 'í', 'ó', 'ú'];
  const cleaned = word.replace(/[,;:!?'¿'¡'()[]{}]/g, '').toLowerCase();
  return cleaned.length > 0 && vowels.includes(cleaned[0]);
}

/**
 * Determine word stress type based on Spanish orthographic rules
 *
 * Rules:
 * 1. If word has accent mark (á, é, í, ó, ú), stress is on that syllable
 * 2. If word ends in vowel, 'n', or 's': stress on penultimate (grave)
 * 3. Otherwise: stress on last syllable (aguda)

```

```

/*
 * Esdrújulas (antepenultimate stress) always have accent marks in Spanish
 */
function determineStress(word: string): 'aguda' | 'grave' | 'esdrujula' {
    const cleaned = word.replace(/[\.,;:!?'¿''][\[]{}]/g, '').toLowerCase();

    if (!cleaned) return 'grave'; // Default

    // Check for accent marks (indicates esdrújula or specific stress)
    const hasAccent = /[áéíóú]/.test(cleaned);

    if (hasAccent) {
        // Count vowels to determine if esdrújula
        const vowelCount = (cleaned.match(/[aeiouáéíóú]/g) || []).length;
        const accentPosition = cleaned.search(/[áéíóú]/);

        // If accent is on third-to-last vowel or earlier, it's esdrújula
        // This is a simplification - proper detection requires syllabification
        if (vowelCount >= 3) {
            const vowelsAfterAccent = (cleaned.substring(accentPosition + 1).match(/[aeiouáéíóú]/g) || []).length;
            if (vowelsAfterAccent >= 2) {
                return 'esdrujula';
            }
        }

        // If accent but not esdrújula, check last character
        const lastChar = cleaned[cleaned.length - 1];
        if (/[aeiouáéíóúns]/.test(lastChar)) {
            return 'grave'; // Accent overrides default rule
        }
        return 'aguda';
    }

    // No accent - apply default rules
    const lastChar = cleaned[cleaned.length - 1];

    if (/[aeiouáéíóúns]/.test(lastChar)) {
        return 'grave'; // Default for words ending in vowel, n, or s
    }

    return 'aguda'; // Default for words ending in consonant (except n, s)
}

/**
 * Count syllables in a verse with Spanish prosodic rules
 *
 * Rules applied:
 * 1. Count base syllables (vowel groups)
 * 2. Detect sinalefas (vowel elision between words)
 * 3. Adjust for final word stress:
 *     - Aguda: +1 syllable
 *     - Grave: no adjustment
 *     - Esdrújula: -1 syllable
 */

```

```

export function countSyllables(verse: string): SyllableCount {
  // Split into words
  const words = verse.trim().split(/\s+/).filter(w => w.length > 0);

  if (words.length === 0) {
    return {
      words: 0,
      approximateSyllables: 0,
      sinalefas: [],
      stressAdjustment: 0,
      finalWordStress: 'unknown',
      isValid: false,
      breakdown: 'Verso vacío'
    };
  }

  // Count base syllables
  let totalSyllables = 0;
  const syllablesPerWord: number[] = [];

  for (const word of words) {
    const count = countBaseVowels(word);
    syllablesPerWord.push(count);
    totalSyllables += count;
  }

  // Detect sinalefas
  const sinalefas: Sinalefa[] = [];

  for (let i = 0; i < words.length - 1; i++) {
    const currentWord = words[i];
    const nextWord = words[i + 1];

    if (endsWithVowel(currentWord) && startsWithVowel(nextWord)) {
      sinalefas.push({
        position: `${currentWord}_${nextWord}`,
        reduction: 1,
        words: [currentWord, nextWord]
      });
      totalSyllables -= 1; // Subtract 1 for the sinalefa
    }
  }

  // Determine stress of final word
  const finalWord = words[words.length - 1];
  const finalWordStress = determineStress(finalWord);

  // Apply stress adjustment
  let stressAdjustment = 0;
  switch (finalWordStress) {
    case 'aguda':
      stressAdjustment = 1;
      break;
    case 'grave':
  }
}

```

```

        stressAdjustment = 0;
        break;
    case 'esdrujula':
        stressAdjustment = -1;
        break;
    }

const finalSyllables = totalSyllables + stressAdjustment;

// Build breakdown explanation
const breakdown = buildBreakdown(
    words.length,
    syllablesPerWord,
    sinalefas,
    finalWordStress,
    stressAdjustment,
    finalSyllables
);

return {
    words: words.length,
    approximateSyllables: finalSyllables,
    sinalefas,
    stressAdjustment,
    finalWordStress,
    isValid: finalSyllables === 8,
    breakdown
};
}

/**
 * Build human-readable breakdown of syllable counting
 */
function buildBreakdown(
    wordCount: number,
    syllablesPerWord: number[],
    sinalefas: Sinalefa[],
    stress: string,
    adjustment: number,
    final: number
): string {
    const baseSyllables = syllablesPerWord.reduce((a, b) => a + b, 0);
    let breakdown = `${wordCount} palabras, ${baseSyllables} sílabas base`;

    if (sinalefas.length > 0) {
        breakdown += `\n- ${sinalefas.length} sinalefa(s): ${sinalefas.map(s => s.position).join(', ')}`;
        breakdown += ` (-${sinalefas.length})`;
    }

    if (adjustment !== 0) {
        breakdown += `\n- Ajuste por palabra final ${stress}: ${adjustment > 0 ? '+' : ''}${adjustment}`;
    }

    breakdown += `\n= ${final} sílabas métricas ${final === 8 ? ' ' : ''}${adjustment}`;
}

```

```

    return breakdown;
}

/**
 * Validate syllable count for entire décima
 */
export function validateDecimaSyllables(verses: string[]): {
  valid: boolean;
  counts: SyllableCount[];
  invalidVerses: number[];
  totalSinalefas: number;
} {
  if (verses.length !== 10) {
    return {
      valid: false,
      counts: [],
      invalidVerses: [],
      totalSinalefas: 0
    };
  }

  const counts = verses.map(countSyllables);
  const invalidVerses = counts
    .map((c, idx) => ({ idx: idx + 1, valid: c.isValid }))
    .filter(v => !v.valid)
    .map(v => v.idx);

  const totalSinalefas = counts.reduce((sum, c) => sum + c.sinalefas.length, 0);

  return {
    valid: invalidVerses.length === 0,
    counts,
    invalidVerses,
    totalSinalefas
  };
}

/**
 * Format syllable count for display
 */
export function formatSyllableCount(count: SyllableCount): string {
  const icon = count.isValid ? ' ' : ' ';
  return `${count.words} palabras, ~${count.approximateSyllables} síl. ${icon}`;
}

```

lib/rhyme-validator.ts

```

/**
 * ABBAACCDDC Rhyme Scheme Validator for Spanish Décima Espinela
 * Validates consonant rhyme patterns with normalization
 */

```

```

export interface RhymeGroup {
  sound: string;
  verses: number[];
  words: string[];
}

export interface RhymeValidationResult {
  valid: boolean;
  errors: string[];
  groups: {
    A: RhymeGroup;
    B: RhymeGroup;
    C: RhymeGroup;
    D: RhymeGroup;
  };
  scheme: string;
  confidence: number; // 0-100%
}

/**
 * Normalize Spanish word for rhyme comparison
 * - Lowercase
 * - Remove accents (á→a, é→e, etc.)
 * - Strip punctuation
 * - Remove whitespace
 */
function normalizeWord(word: string): string {
  return word
    .toLowerCase()
    .normalize('NFD')
    .replace(/[\u0300-\u036f]/g, '') // Remove accents
    .replace(/[.,;:!?'¿¡"][]{}]/g, '') // Remove punctuation
    .trim();
}

/**
 * Extract rhyme sound from word
 * Returns substring from last vowel to end of word
 * Examples:
 *   "exilio" → "ilio"
 *   "corazón" → "on"
 *   "sentir" → "ir"
 */
function extractRhymeSound(word: string): string {
  const normalized = normalizeWord(word);
  const vowels = ['a', 'e', 'i', 'o', 'u'];

  // Find position of last vowel
  let lastVowelIndex = -1;
  for (let i = normalized.length - 1; i >= 0; i--) {
    if (vowels.includes(normalized[i])) {
      lastVowelIndex = i;
      break;
    }
  }
}

```

```

}

if (lastVowelIndex === -1) {
  return normalized; // No vowel found, return entire word
}

return normalized.substring(lastVowelIndex);
}

/**
 * Get final word from verse
 * Handles punctuation and whitespace
 */
function getFinalWord(verse: string): string {
  const words = verse.trim().split(/\s+/);
  return words[words.length - 1] || '';
}

/**
 * Validate ABBAACCDDC rhyme scheme
 *
 * Scheme breakdown:
 *   A: verses 1, 4, 5      (positions 0, 3, 4)
 *   B: verses 2, 3          (positions 1, 2)
 *   C: verses 6, 7, 10     (positions 5, 6, 9)
 *   D: verses 8, 9          (positions 7, 8)
 */
export function validateRhymeScheme(verses: string[]): RhymeValidationResult {
  const errors: string[] = [];

  // Validate input
  if (!verses || verses.length !== 10) {
    return {
      valid: false,
      errors: [`Décima must have exactly 10 verses, found ${verses?.length || 0}`],
      groups: {
        A: { sound: '', verses: [], words: [] },
        B: { sound: '', verses: [], words: [] },
        C: { sound: '', verses: [], words: [] },
        D: { sound: '', verses: [], words: [] }
      },
      scheme: 'INVALID',
      confidence: 0
    };
  }

  // Extract final words and rhyme sounds
  const finalWords = verses.map(getFinalWord);
  const rhymeSounds = finalWords.map(extractRhymeSound);

  // Group verses by rhyme scheme
  const groupA = {
    verses: [1, 4, 5],
    indices: [0, 3, 4],

```

```

    sounds: [rhymeSounds[0], rhymeSounds[3], rhymeSounds[4]],
    words: [finalWords[0], finalWords[3], finalWords[4]]
};

const groupB = {
  verses: [2, 3],
  indices: [1, 2],
  sounds: [rhymeSounds[1], rhymeSounds[2]],
  words: [finalWords[1], finalWords[2]]
};

const groupC = {
  verses: [6, 7, 10],
  indices: [5, 6, 9],
  sounds: [rhymeSounds[5], rhymeSounds[6], rhymeSounds[9]],
  words: [finalWords[5], finalWords[6], finalWords[9]]
};

const groupD = {
  verses: [8, 9],
  indices: [7, 8],
  sounds: [rhymeSounds[7], rhymeSounds[8]],
  words: [finalWords[7], finalWords[8]]
};

// Validate each group has consistent rhyme sound
function validateGroup(
  group: typeof groupA,
  letter: string
): { valid: boolean; sound: string } {
  const uniqueSounds = [...new Set(group.sounds)];

  if (uniqueSounds.length > 1) {
    // Multiple rhyme sounds in group - not valid
    group.sounds.forEach((sound, idx) => {
      if (sound !== uniqueSounds[0]) {
        errors.push(`Verse ${group.verses[idx]} (${letter}): "${group.words[idx]}" (${sound}) doesn't rhyme with ${uniqueSounds[0]}`);
      }
    });
    return { valid: false, sound: uniqueSounds[0] };
  }

  return { valid: true, sound: uniqueSounds[0] };
}

const aResult = validateGroup(groupA, 'A');
const bResult = validateGroup(groupB, 'B');
const cResult = validateGroup(groupC, 'C');
const dResult = validateGroup(groupD, 'D');

const allValid = aResult.valid && bResult.valid && cResult.valid && dResult.valid;

```

```

// Calculate confidence score
let confidence = 100;
if (errors.length > 0) {
  confidence = Math.max(0, 100 - (errors.length * 15)); // -15% per error
}

return {
  valid: allValid,
  errors,
  groups: {
    A: {
      sound: aResult.sound,
      verses: groupA.verses,
      words: groupA.words
    },
    B: {
      sound: bResult.sound,
      verses: groupB.verses,
      words: groupB.words
    },
    C: {
      sound: cResult.sound,
      verses: groupC.verses,
      words: groupC.words
    },
    D: {
      sound: dResult.sound,
      verses: groupD.verses,
      words: groupD.words
    }
  },
  scheme: 'ABBAACCDCC',
  confidence
};
}

/**
 * Get color for rhyme group (for UI visualization)
 */
export function getRhymeGroupColor(group: 'A' | 'B' | 'C' | 'D'): string {
  const colors = {
    A: '#C8A05C', // Gold (primary)
    B: '#D2691E', // Brown-orange (button)
    C: '#8B4513', // Saddle brown (accent)
    D: '#5C4033' // Dark brown (secondary)
  };
  return colors[group];
}

/**
 * Format rhyme validation result for display
 */
export function formatRhymeResult(result: RhymeValidationResult): string {
  if (result.valid) {

```

```

        return ` Rima válida ABBAACCDDC (${result.confidence}% confianza)`;
    }

    return ` Rima inválida (${result.confidence}% confianza)\nErrores:\n${result.errors.join('\n')}`;
}

```

lib/openrouter-client.ts

```

/*
 * OpenRouter API Client with JSON Schema Support
 * Handles multi-model LLM calls with strict JSON validation
 */

export interface OpenRouterConfig {
    apiKey: string;
    model?: string; // Default: anthropic/clause-3.5-sonnet
    temperature?: number;
    maxTokens?: number;
}

export interface DecimaSchema {
    numero: number;
    poeta: string;
    versos: string[]; // Must be exactly 10 verses
    analisis_metrico: {
        silabas_por_verso: number[];
        sinalefas_detectadas: number;
        palabras_finales: string[];
    };
    analisis_cultural: string;
}

export interface ExtractionResponse {
    decimas: Array<{
        numero: number;
        poeta: string;
        versos: string[];
    }>;
}

export interface ValidationResponse {
    decimas: Array<{
        numero: number;
        analisis_metrico: {
            silabas_por_verso: number[];
            sinalefas_detectadas: number;
            palabras_finales: string[];
        };
    }>;
}

export interface AnalysisResponse {
    decimas: DecimaSchema[];
}

```

```

resumen: {
  total_decimas: number;
  notas: string;
};

}

const DEFAULT_MODEL = "anthropic/clause-3.5-sonnet";
const OPENROUTER_API_URL = "https://openrouter.ai/api/v1/chat/completions";

/**
 * Call OpenRouter API with JSON mode enforcement
 */
export async function callOpenRouter<T = any>(
  config: OpenRouterConfig,
  systemPrompt: string,
  userPrompt: string,
  requireJson: boolean = true
): Promise<T> {
  const { apiKey, model = DEFAULT_MODEL, temperature = 0.3, maxTokens = 4000 } = config;

  if (!apiKey || !apiKey.startsWith('sk-or-')) {
    throw new Error('Invalid OpenRouter API key format. Must start with "sk-or-"]');
  }

  const requestBody: any = {
    model,
    messages: [
      { role: "system", content: systemPrompt },
      { role: "user", content: userPrompt }
    ],
    temperature,
    max_tokens: maxTokens,
  };

  // Enable JSON mode if required
  if (requireJson) {
    requestBody.response_format = { type: "json_object" };
  }

  try {
    const response = await fetch(OPENROUTER_API_URL, {
      method: "POST",
      headers: {
        "Authorization": `Bearer ${apiKey}`,
        "Content-Type": "application/json",
        "HTTP-Referer": typeof window !== 'undefined' ? window.location.origin : "https://viajera-digital.com",
        "X-Title": "Viajera Digital - Décima Transcriber"
      },
      body: JSON.stringify(requestBody)
    });

    if (!response.ok) {
      const errorData = await response.json().catch(() => ({}));
      throw new Error(
        `Error ${response.status}: ${errorData.message || response.statusText}`
      );
    }

    return response.json();
  } catch (error) {
    console.error(`Error calling OpenRouter API: ${error.message}`);
    throw error;
  }
}

```

```

`OpenRouter API error (${response.status}): ${errorData.error?.message || response.statusText}`;
}

const data = await response.json();

if (!data.choices || !data.choices[0] || !data.choices[0].message) {
  throw new Error("Invalid response format from OpenRouter");
}

const content = data.choices[0].message.content;

// Parse and validate JSON if required
if (requireJson) {
  try {
    return JSON.parse(content) as T;
  } catch (parseError) {
    console.error("JSON parse error:", content);
    throw new Error(`Failed to parse JSON response: ${parseError instanceof Error ? parseError.message : parseError}`);
  }
}

return content as T;
} catch (error) {
  if (error instanceof Error) {
    throw error;
  }
  throw new Error(`OpenRouter API call failed: ${String(error)}`);
}
}

/**
 * Stage 1: Extract décimas from YouTube video
 */
export async function extractDecimas(
  apiKey: string,
  youtubeUrl: string
): Promise<ExtractionResponse> {
  const systemPrompt = `You are an expert in Cuban décima espinela poetry transcription.

```

Your task: Extract ALL décimas from the YouTube video and return them in strict JSON format.

OUTPUT SCHEMA:

```
{
  "decimas": [
    {
      "numero": 1,
      "poeta": "Nombre del Poeta",
      "versos": ["verso 1", "verso 2", ..., "verso 10"]
    }
  ]
}
```

RULES:

1. Each décima MUST have exactly 10 verses
2. Preserve exact wording as spoken
3. Identify poet name from video context
4. Number décimas sequentially starting from 1
5. Return ONLY valid JSON, no markdown or explanations`;

```
const userPrompt = `Extract all décimas from this YouTube video: ${youtubeUrl}`

Analyze the video transcript and identify each décima (10-verse stanza).
Return strict JSON with the schema provided.`;

return callOpenRouter<ExtractionResponse>(
  { apiKey, temperature: 0.1, maxTokens: 6000 },
  systemPrompt,
  userPrompt,
  true
);
}

/**
 * Stage 2: Validate syllable count and detect sinalefas
 */
export async function validateMetrics(
  apiKey: string,
  decimas: ExtractionResponse['decimas']
): Promise<ValidationResponse> {
  const systemPrompt = `You are an expert in Spanish prosody and métrica.
```

Your task: Validate that each verse has exactly 8 metric syllables according to Spanish prosodic rules.

OUTPUT SCHEMA:

```
{
  "decimas": [
    {
      "numero": 1,
      "analisis_metrico": {
        "silabas_por_verso": [8, 8, 8, 8, 8, 8, 8, 8, 8],
        "sinalefas_detectadas": 2,
        "palabras_finales": ["palabra1", "palabra2", ..., "palabra10"]
      }
    }
  ]
}
```

SPANISH PROSODY RULES:

1. Count vowels as base syllables
2. Detect SINALEFA: when word ends with vowel AND next word starts with vowel, they merge into one syllable
 - Example: "de él" = "de_él" = 1 syllable (not 2)
3. Adjust for final word stress:
 - AGUDA (stress on last syllable): +1 syllable
 - GRAVE (penultimate stress): no change
 - ESDRÚJULA (antepenultimate stress): -1 syllable
4. Each verse MUST have 8 metric syllables total`;

```

const userPrompt = `Validate syllable count for these décimas:
${JSON.stringify(decimas, null, 2)}

Apply Spanish prosodic rules (sinalefa, stress adjustment) to verify each verse has 8 syllables.
Return strict JSON with metric analysis.`;

return callOpenRouter<ValidationResponse>(
  { apiKey, temperature: 0.3, maxTokens: 6000 },
  systemPrompt,
  userPrompt,
  true
);
}

/**
 * Stage 3: Cultural and rhyme analysis
 */
export async function analyzeCultural(
  apiKey: string,
  extractedDecimas: ExtractionResponse['decimas'],
  validatedMetrics: ValidationResponse['decimas']
): Promise<AnalysisResponse> {
  const systemPrompt = `You are an expert in Cuban décima espinela poetry and Hispanic cultural studies

```

Your task: Analyze rhyme scheme (ABBAACCDDC), cultural themes, and linguistic patterns.

OUTPUT SCHEMA:

```
{
  "decimas": [
    {
      "numero": 1,
      "poeta": "Nombre del Poeta",
      "versos": ["verso 1", ..., "verso 10"],
      "analisis_metrico": {
        "silabas_por_verso": [8, 8, 8, 8, 8, 8, 8, 8, 8],
        "sinalefas_detectadas": 2,
        "palabras_finales": ["palabra1", ..., "palabra10"]
      },
      "analisis_cultural": "Brief cultural and rhyme analysis"
    }
  ],
  "resumen": {
    "total_decimas": 3,
    "notas": "Overall observations"
  }
}
```

ANALYSIS REQUIREMENTS:

1. Verify ABBAACCDDC rhyme scheme (consonant rhyme)
2. Identify Cuban cultural themes (exilio, nostalgia, patria, etc.)
3. Analyze linguistic features of Cuban Spanish
4. Note any deviations from traditional structure`;

```

const userPrompt = `Analyze these décimas with validated metrics:

EXTRACTED DÉCIMAS:
${JSON.stringify(extractedDecimas, null, 2)}

VALIDATED METRICS:
${JSON.stringify(validatedMetrics, null, 2)}

Provide complete cultural analysis with rhyme verification.
Return strict JSON combining all data.`;

return callOpenRouter<AnalysisResponse>(
  { apiKey, temperature: 0.5, maxTokens: 8000 },
  systemPrompt,
  userPrompt,
  true
);
}

```

Notas Importantes

- **Componentes de UI:** Los componentes de shadcn/ui están en `components/ui/` (no incluidos aquí por brevedad)
 - **education-section.tsx:** Contiene 12 biografías completas de poetas (truncado en esta vista)
 - **Variables de entorno:** Requiere `DATABASE_URL` en archivo `.env`
 - **Imágenes:** La foto de Calixto González está en `public/calixto-gonzalez-hero.jpg`
-

Generado: Fri Nov 21 17:31:45 UTC 2025