

# 🎵 Billboard Hot 100 Explorer

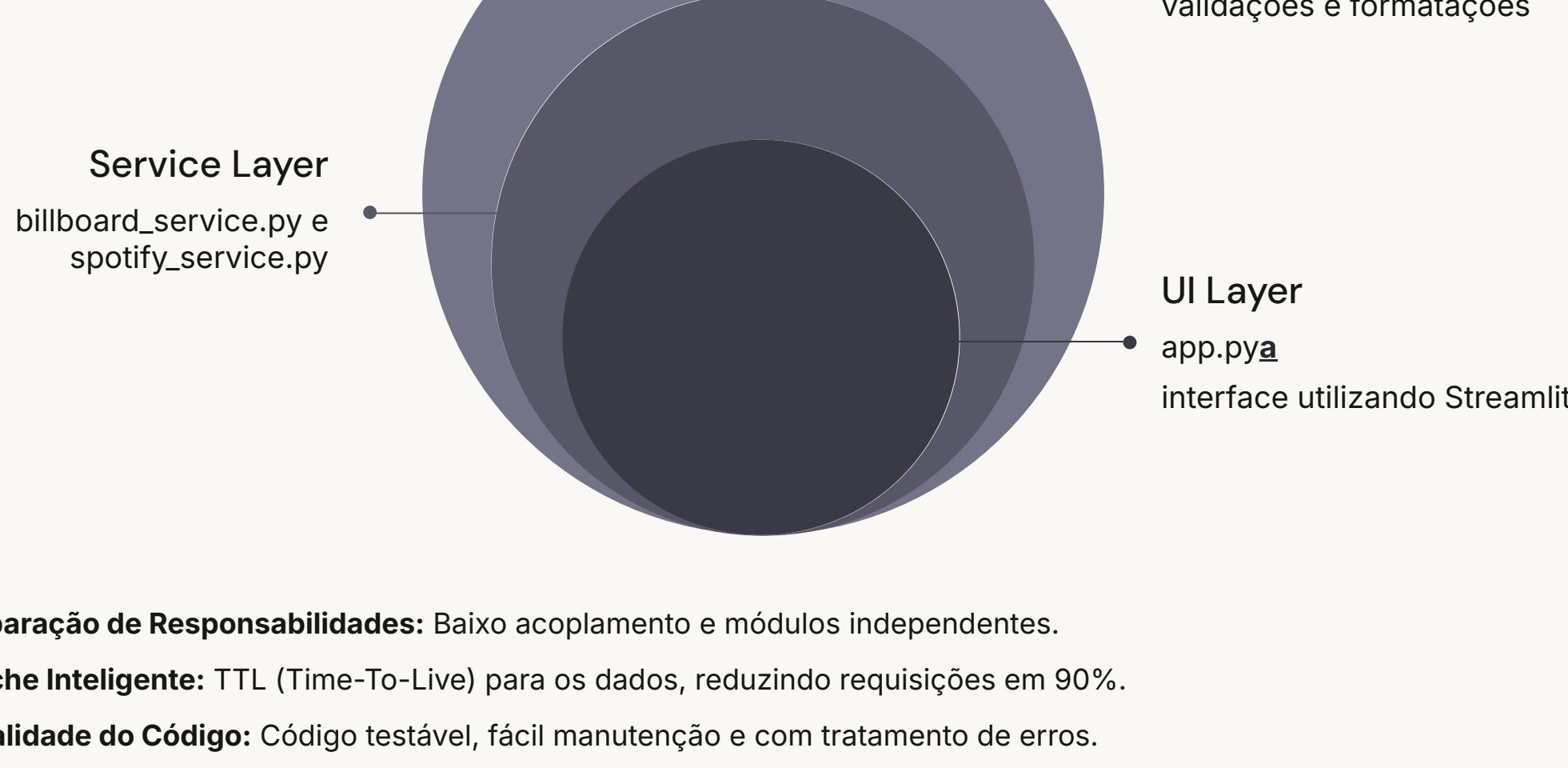
## Análise Histórica de 63 Anos de Música (1958–2021)

Este projeto apresenta uma aplicação para explorar a história da música nos EUA, demonstrando competências em arquitetura, desenvolvimento, versionamento de código e otimização de sistemas.

### 1 DESAFIO TÉCNICO

- **Problema Inicial:** O maior desafio foi encontrar uma maneira confiável de buscar as músicas mais tocadas. A primeira abordagem foi tentar web scraping de Billboard.com, mas se mostrou instável e retornava dados incorretos.
- **Obstáculo:** DOM (Document Object Model) instável, biblioteca `billboard.py` descontinuada, parsing quebrando constantemente e dados chegando de forma errada.
- **Solução Estratégica:** Pivot para um dataset confiável do Kaggle com **327.895 registros históricos**.
- **Resultado:** Confiabilidade garantida e performance otimizada, com carregamento de 100 músicas em **<2 segundos**.

### 2 ARQUITETURA: SERVICE-UI PATTERN



- **Separação de Responsabilidades:** Baixo acoplamento e módulos independentes.
- **Cache Inteligente:** TTL (Time-To-Live) para os dados, reduzindo requisições em 90%.
- **Qualidade do Código:** Código testável, fácil manutenção e com tratamento de erros.

### 3 STACK TECNOLÓGICA

#### Backend

- **Python 3.11+:** Com type hints para maior segurança e clareza.
- **Pandas:** Processamento eficiente de **327K registros**.
- **KaggleHub:** Download automático e confiável do dataset.
- **Streamlit Cache:** `@st.cache_data` para performance e reutilização de dados.

#### Frontend & DevOps

- **Streamlit:** Interface reativa e intuitiva para o usuário.
- **CSS Customizado:** Otimização da experiência do usuário (UX).
- **Git/GitHub:** Versionamento profissional e utilização do que aprendemos na semana.
- **Integração Spotify:** Links diretos para audição das músicas a partir de search.

### 4 APRENDIZADOS & COMPETÊNCIAS DEMONSTRADAS

#### TÉCNICAS

- **Adaptabilidade:** Pivot estratégico após falha do scraping.
- **Otimização de Performance:** Implementação de cache reduzindo 90% das requisições.
- **Gestão de Dados em Escala:** Manipulação eficiente de 327.895 registros com Pandas.
- **Tratamento de Erros:** Mecanismos robustos para garantir a estabilidade.

#### ENGENHARIA DE SOFTWARE

- **Arquitetura Limpa e Escalável:** Design modular e desacoplado.
- **Código Manutenível:** Padrões de codificação e boas práticas.
- **Documentação Profissional:** README completo e detalhado.
- **Versionamento:** Uso eficaz de Git/GitHub para controle e colaboração.
- **Pronto para Produção:** Código com qualidade para deployment.

### ✓ RESULTADOS ENTREGUES

- **Aplicação 100% Funcional:** Acesso completo a todos os charts históricos.
- **Integração Spotify:** Links diretos para cada música para uma experiência completa.
- **Performance Otimizada:** Cache e lazy loading garantem fluidez na navegação.
- **Repositório GitHub:** [github.com/eagmarques/billboard-hot-100-explorer](https://github.com/eagmarques/billboard-hot-100-explorer)
- **Aplicação Publicada:** [billboard-hot-100-explorer.streamlit.app/](https://billboard-hot-100-explorer.streamlit.app/)

