**Documento de Requisitos Funcionais – MVP**

**Projeto: Algoritmo para Diversificação de Contéudo nas Redes Sociais**

**Versão: 1.0**

**Escopo: Foco exclusive na entrega técnica mínima: coleta, processamento e geração de sugestões com base em dados do YouTube de um usuário.**

**Visão Geral:**

Este documento especifica os requisitos funcionais da versão MVP do sistema, cujo objetivo é coletar dados de consumo de conteúdo do YouTube via API, identificar padrões de consumo por categoria e gerar sugestões de vídeos diversificados com base em categorias distintas.

**Tecnologias envolvidas (MVP):**

1. Linguagem: Python 3.x
2. Biblioteca de API: google-api-python-client
3. Processamento: pandas, collections, datetime, typing
4. Saída: Arquivo JSON com sugestões

**Requisitos Funcionais:**

**RF01 –** Autenticação do usuário na API do YouTube:

**Descrição:** O sistema deve permitir que o usuario autentique sua conta Google para que a aplicação tenha acesso ao seu historico publico (caso esteja autorizado).

**Dependencia:** OAuth 2.0 com escopo youtube.readonly;

**Alternativa no MPV:** Arquivo JSON contendo simulação do histórico de vídeos (evitar complexidade do OAuth).

**RF02 –** Coleta de Vídeos Consumidos:

**Descrição:** O sistema deve extrair informações sobre os vídeos consumidos recentemente pelo usuário.

**Campos mínimos coletados por vídeo:**

1. video\_id
2. title
3. channel
4. publishedAt
5. categoryId

**Fonte dos dados:** YouTube Data API v3, endpoint: playlistItems.list com playlistId=watchHistory (ou JSON simulado);

**RF03 –** Agrupamento por Categorias:

**Descrição:** O sistema deve agrupar os vídeos consumidos com base em seus categoryId, identificando as categorias mais recorrentes.

**Critério de bolha:** Categorias com mais de X% de presença (ex: 60%) são consideradas dominantes.

**RF04 –** Geração de Sugestões Diversificadas:

**Descrição:** O sistema deve retornar uma lista de vídeos de categorias pouco ou nunca acessadas pelo usuário.

**Critério de seleção:**

1. Categorias diferentes das dominantes
2. Vídeos populares ou recentes nessas categorias (simulados no MVP)

**RF05 –** Exportação de Sugestões em JSON:

**Descrição:** O sistema deve gerar um arquivo recomendacoes.json contendo:

[

{

“video\_id”: “1”,

“title”: “Exemplo de vídeo”,

“category”: “Documentário”

}

]

**RF06 –** Interface de Linha de Comando:

**Descrição:** O MPV deve rodar via terminal, com o seguinte comando:

python recomendador.py --input=historico.json --output=recomendacoes.json

**Requisitos Não-Funcionais:**

NFR01: O sistema deve funcionar offile apos a obtenção dos dados (sem chamadas adicionais à API).

NFR02: A execução total deve durar no máximo 15 segundos com até 100 vídeos no histórico.

NFR03: O sistema deve ser modularizado, com separação entre:

1. Modulo de leitura de dados
2. Modulo de analise e geracoes de sugestoes
3. Modulo de exportacao JSON

**Futuras Extensões (fora do MVP):**

1. Implementação completa do OAuth 2.0;
2. Acesso em tempo real à API com credenciais do usuário;
3. Interface gráfica ou web com frontend em JavaScript;
4. Acompanhamento de mudança de comportamento ao longo do tempo;
5. Curso educativo integrado ao sistema;