**Documentação de Requisitos – Projeto de Banco de Dados Híbrido**

**Tema: Plataforma de Streaming Personalizada**

**1. Introdução**

Este documento descreve os requisitos funcionais e não funcionais da **Plataforma de Streaming Personalizada**, que utiliza **MySQL** (banco relacional) e **MongoDB Atlas** (banco não relacional).

A solução permitirá gerenciar **usuários, planos de assinatura, catálogos de filmes/séries, pagamentos e histórico de consumo** no MySQL, enquanto o MongoDB será responsável por armazenar **avaliações, preferências dinâmicas, registros de interação e recomendações personalizadas**.

**2. Definições Conceituais**

**2.1 Requisitos Funcionais**

Funcionalidades obrigatórias que o sistema deve executar.

Exemplo para este projeto:

* O sistema deve permitir cadastro e autenticação de usuários no MySQL.
* O sistema deve salvar preferências e histórico de reprodução no MongoDB.

**2.2 Requisitos Não Funcionais**

Qualidades e restrições da plataforma.

Exemplo para este projeto:

* O sistema deve responder a consultas de catálogos em até 2 segundos.
* O MongoDB deve suportar alta taxa de inserções de eventos de navegação.
* O MySQL deve estar em conformidade com a LGPD para dados pessoais.

**3. Escopo do Projeto**

**No MySQL (relacional):**

* Cadastro de usuários e perfis
* Planos de assinatura e pagamentos
* Catálogo estruturado de filmes, séries, temporadas e episódios
* Histórico de reprodução e transações

**No MongoDB Atlas (não relacional):**

* Avaliações de usuários (comentários, notas, reações)
* Recomendações personalizadas (documentos dinâmicos por perfil)
* Registros de navegação e comportamento (buscas, cliques, tempo assistido)
* Metadados não estruturados (ex.: legendas alternativas, imagens de capa)

**4. Requisitos Funcionais**

**4.1 Cadastro e Autenticação**

* **RF01**: O sistema deve permitir cadastro, autenticação e gerenciamento de usuários no MySQL.
* **RF02**: O sistema deve validar login e plano ativo em tempo real.

**4.2 Gestão de Conteúdo**

* **RF03**: O sistema deve gerenciar catálogo de filmes, séries e episódios no MySQL.
* **RF04**: O sistema deve registrar histórico de reprodução (play, pause, stop) no MySQL.

**4.3 Conteúdo Não Estruturado e Personalização**

* **RF05**: O sistema deve armazenar avaliações e comentários de usuários no MongoDB.
* **RF06**: O sistema deve salvar logs de interação (tempo de exibição, buscas, cliques) no MongoDB.
* **RF07**: O sistema deve integrar dados de ambos os bancos para gerar recomendações personalizadas.

**4.4 Relatórios e Consultas**

* **RF08**: O sistema deve gerar relatórios financeiros e de assinaturas no MySQL.
* **RF09**: O sistema deve permitir consultas híbridas (ex.: buscar um filme no MySQL e exibir junto as avaliações no MongoDB).

**5. Requisitos Não Funcionais**

**5.1 Desempenho**

* **RNF01**: Consultas de catálogo no MySQL devem ter tempo de resposta máximo de 2s.
* **RNF02**: O MongoDB deve suportar inserção mínima de 2000 eventos/segundo (logs de navegação).
* **RNF03**: Relatórios consolidados devem ser gerados em no máximo 8s.

**5.2 Segurança**

* **RNF04**: Dados sensíveis no MySQL devem ser criptografados com AES-256.
* **RNF05**: O sistema deve utilizar autenticação JWT para acesso às APIs.
* **RNF06**: Deve atender à LGPD para dados pessoais e preferências de consumo.

**5.3 Disponibilidade e Escalabilidade**

* **RNF07**: A plataforma deve ter disponibilidade mínima de 99,7%.
* **RNF08**: O MongoDB deve ser escalável horizontalmente para suportar picos de uso (lançamento de filmes/séries).
* **RNF09**: Deve haver replicação no MySQL para alta disponibilidade.

**5.4 Usabilidade e Manutenibilidade**

* **RNF10**: O sistema deve possuir interface administrativa intuitiva para gestão de catálogo e relatórios.
* **RNF11**: O código deve seguir boas práticas de clean code e possuir documentação técnica.

**5.5 Portabilidade**

* **RNF12**: O sistema deve ser compatível com containers Docker.
* **RNF13**: Deve permitir deploy em ambientes cloud (AWS, GCP ou Azure).