

# Documento de Design Orientado ao Domínio (DDD) – Hackaton FIAP Vídeos

## 1. Introdução

### 1.1. Visão Geral

Este documento apresenta a modelagem de domínio e o design orientado ao domínio (DDD) para o sistema de **upload, processamento e disponibilização de vídeos** do projeto *FIAP X*.

O objetivo é definir as entidades centrais do sistema, identificar os limites dos contextos, descrever a infraestrutura, práticas de segurança e incluir diagramas para visualização das relações e fluxos.

### 1.2. Escopo do Sistema

O sistema abrange:

- upload e armazenamento de vídeos,
- processamento (transcodificação, geração de thumbnails, compressão),
- consultas de status,
- notificações de conclusão,
- autenticação e autorização,
- cache e filas para tarefas assíncronas.

### 1.3. Tecnologias e Infraestrutura

- **Linguagem / Framework:** Java 17, Spring Boot
- **Banco de Dados:** MySQL (produção), H2 (testes)
- **Mensageria:** RabbitMQ
- **Cache:** Redis
- **Serviços:** Feign Client para integrações externas
- **Infra:** Docker / Docker-Compose (desenvolvimento e deploy)
- **Hospedagem:** Pode ser provisionada em nuvem (AWS, GCP, Azure)

## 2. Domínio Principal: Gestão de Vídeos

### 2.1. Descrição do Domínio

O domínio principal envolve o **ciclo de vida de um vídeo**: upload → processamento → disponibilização → notificação.

### 2.2. Objetivos do Sistema

- Permitir upload seguro e escalável de vídeos.
- Processar vídeos em segundo plano sem bloquear usuários.

- Disponibilizar vídeos em formatos adequados.
- Garantir confiabilidade e segurança no armazenamento e acesso.

## 3. Bounded Contexts

### 3.1. Contexto de Identidade e Autenticação

**Responsabilidade:** gerenciar usuários do sistema (administradores, clientes) e autenticação via tokens JWT.

- **Entidades:** Usuário
- **Serviços:** Autenticação, Criptografia de Senhas
- **Repositórios:** Repositório de Usuários
- **Segurança:** Tokens de acesso, expiração automática, renovação.

### 3.2. Contexto de Upload e Processamento

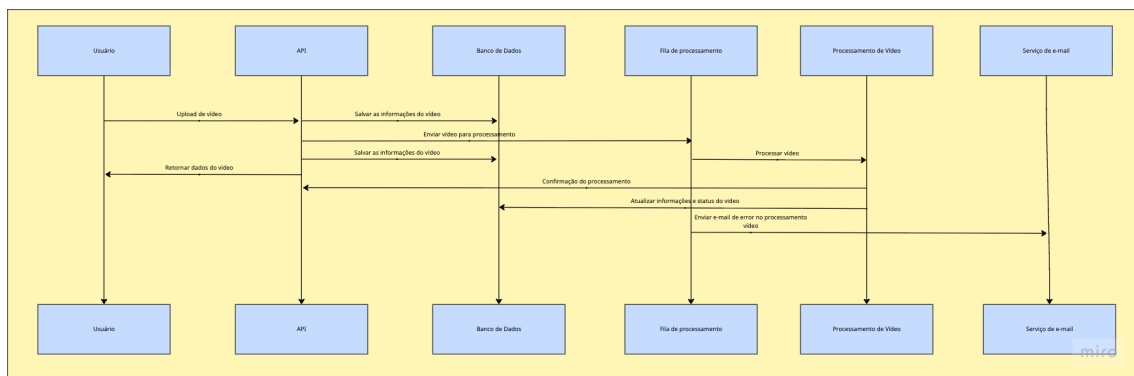
**Responsabilidade:** upload de vídeos e processamento assíncrono.

- **Entidades:** Vídeo, StatusProcessamento
- **Agregados:** Vídeo (contém metadados e estado do processamento)
- **Serviços:** Upload, Transcodificação, Compressão, Geração de Thumbnails
- **Repositórios:** Repositório de Vídeos
- **Filas:** RabbitMQ para processamento assíncrono

### 3.3. Contexto de Notificações

**Responsabilidade:** envio de notificações sobre status de processamento.

- **Entidades:** Notificação
- **Serviços:** Envio de e-mail / mensagem
- **Eventos de Domínio:** VídeoProcessadoSucesso, VídeoProcessadoFalha



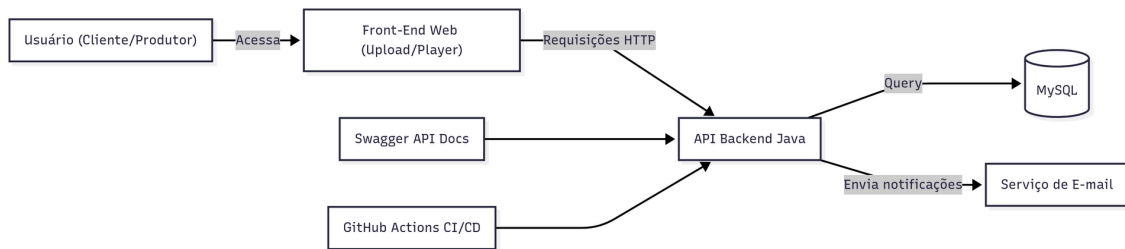
## 4. Infraestrutura e Arquitetura

### 4.1. Arquitetura da Aplicação

- API REST em Spring Boot
- Worker de processamento integrado com RabbitMQ
- Redis para cache de status
- MySQL para persistência de metadados
- Docker para empacotamento e deploy

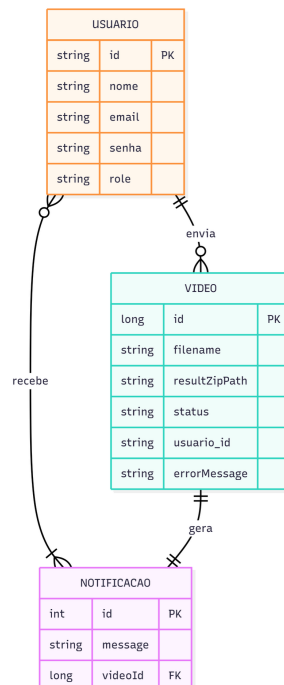
## 4.2. Escalabilidade e Resiliência

- Processamento de vídeos assíncrono via filas
- Workers escaláveis horizontalmente
- Redis como cache de consultas frequentes



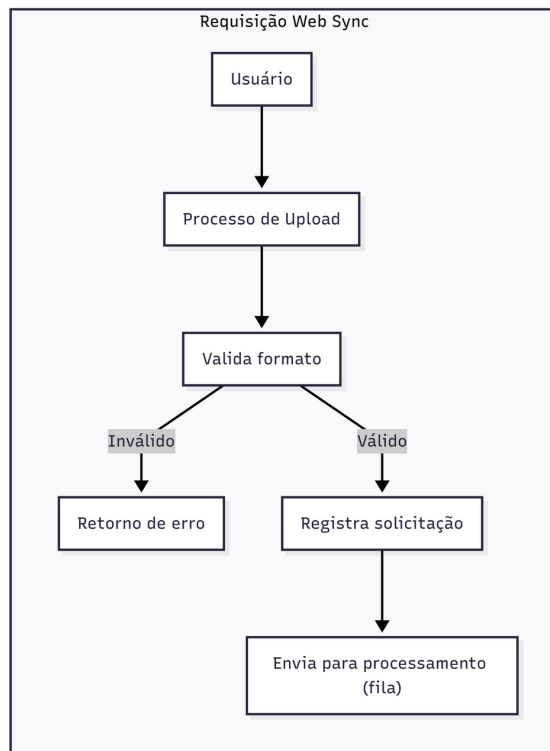
## 5. Entidades

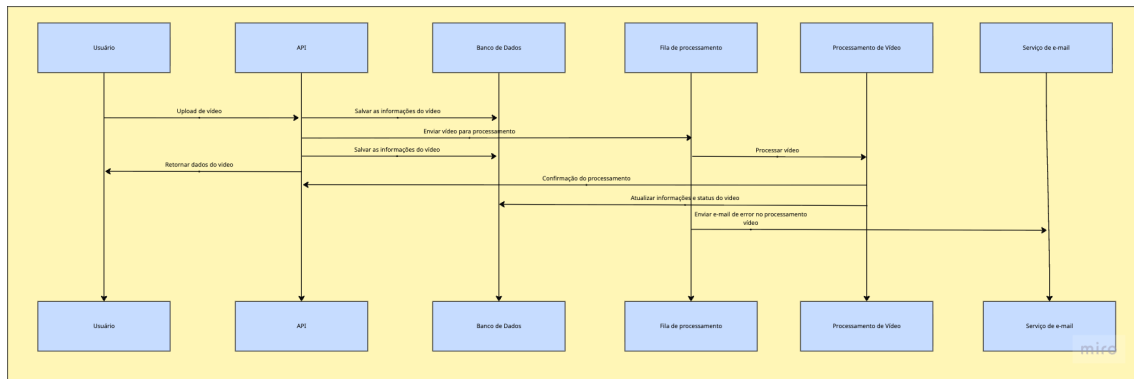
- **Usuário:** credenciais e permissões
- **Vídeo:** metadados (nome, tamanho, formato), estado de processamento
- **Notificação:** mensagem de status enviada ao usuário



## 6. Serviços de Domínio

### 6.1. Serviço de Autenticação





## 6.4. Serviço de Anonimização

Este serviço lida com as solicitações de anonimização de dados pessoais, conforme a Lei Geral de Proteção de Dados (LGPD). O processo inclui a solicitação de anonimização por parte do usuário, o arquivo zip do vídeo processado, o agendamento de um job para processar a anonimização.

## 7. Segurança

- Criptografia de senhas (BCrypt)
- Tokens JWT com expiração
- Proteção contra uploads maliciosos (validação de formato, limite de tamanho)
- Conformidade com LGPD (possibilidade de anonimização/exclusão de dados pessoais dos usuários)

## 8. Repositórios

- **Repositório de Usuários**
- **Repositório de Vídeos**
- **Repositório de Notificações**

## 9. Agregados

- **Vídeo:** controla consistência entre upload, processamento e status
- **Usuário:** controla permissões de acesso

## 10. Eventos de Domínio

- VideoUploaded
- VideoProcessingStarted
- VideoProcessingCompleted
- VideoProcessingFailed

## 10. Eventos de Domínio

## 10.1.Diagrama de Sequência

Este diagrama mostra os possíveis estados de um processamento e as transições entre eles:

