UNIVERSIDADE FEDERAL DO AMAZONAS

INSTITUTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Sistema de Extração de Padrões de Conteúdo Viral em Vídeos Curtos via PLN e Geração Assistida por LLM

Josival Salvador Monteiro Júnior

Manaus - AM

Julho de 2026

Josival Salvador Monteiro Júnior

Sistema de Extração de Padrões de Conteúdo Viral em Vídeos Curtos via PLN e Geração Assistida por LLM

Monografia de Graduação apresentada ao Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de bacharel em Ciência da Computação.

Orientador

Prof. Dr. Moisés Gomes de Carvalho

Universidade Federal do Amazonas

Instituto de Computação

Manaus - AM

Julho de 2026

Monografia de Graduação sob o título *Sistema de Extração de Padrões de Conteúdo Viral em Vídeos Curtos via PLN e Geração Assistida por LLM* apresentada por Josival Salvador Monteiro Júnior e aprovada pelo Instituto de Computação da Universidade Federal do Amazonas, tendo sido aprovada por todos os membros da banca examinadora abaixo especificada:

Prof. Dr. Moisés Gomes de Carvalho

Orientador

Instituto de Computação

Universidade Federal do Amazonas

Prof. Dr. David Braga Fernandes de Oliveira

Membro da Banca Examinadora

Instituto de Computação

Universidade Federal do Amazonas

Manaus - AM, 14 de julho de 2026.

Sistema de Extração de Padrões de Conteúdo Viral em Vídeos Curtos via PLN e Geração Assistida por LLM

Autor: Josival Salvador Monteiro Júnior

Orientador: Prof. Dr. Moisés Gomes de Carvalho

Resumo

Este trabalho apresenta um sistema de extração automatizada de padrões de conteúdo viral em vídeos curtos, segmentados por nicho. A partir desses padrões, o sistema gera, de forma assistida, guias de conteúdo estratégico. Ele coleta metadados e transcrições do YouTube, refinando-os por processamento de linguagem natural e agrupamento não supervisionado do formato narrativo. Os padrões de cada nicho são extraídos por agrupamento baseado em densidade. Um mecanismo de similaridade semântica resolve nichos ainda não presentes na base. O guia final, estruturado em abertura, desenvolvimento e fechamento, é gerado por um modelo de linguagem e entregue por uma aplicação web full-stack. O funcionamento de ponta a ponta foi validado qualitativamente para três nichos distintos. Os guias refletem o padrão real de cada nicho, não uma recomendação genérica. O trabalho identifica limitações de dependência de serviços gratuitos de terceiros, erros ocasionais de tradução e instabilidade do modelo de linguagem. A ausência de validação junto a criadores reais é apontada como prioridade para trabalhos futuros.

Palavras-chave: Processamento de Linguagem Natural. Detecção de Viralidade. Modelos de Linguagem. Geração de Conteúdo. Aprendizado de Máquina.

Sistema de Extração de Padrões de Conteúdo Viral em Vídeos Curtos via PLN e Geração Assistida por LLM

Autor: Josival Salvador Monteiro Júnior

Orientador: Prof. Dr. Moisés Gomes de Carvalho

Abstract

This work presents a system for automated extraction of viral content patterns in short videos, segmented by niche. From these patterns, the system generates strategic content guides in an assisted manner. It collects metadata and transcripts from YouTube, refining them through natural language processing and unsupervised clustering of narrative format. Patterns for each niche are extracted through density-based clustering. A semantic similarity mechanism resolves niches not yet present in the base. The final guide, structured into opening, development, and closing, is generated by a language model and delivered through a full-stack web application. End-to-end operation was qualitatively validated for three distinct niches. The guides reflect each niche's actual pattern, not a generic recommendation. The work identifies limitations related to dependency on free third-party services, occasional translation errors, and language model instability. The absence of validation with real creators is pointed out as a priority for future work.

Keywords: Natural Language Processing. Virality Detection. Language Models. Content Generation. Machine Learning.

Lista de figuras

Figura 1 – Diagrama de Casos de Uso: Acesso e Conta 25
Figura 2 – Diagrama de Casos de Uso: Geração e Gestão 26
Figura 3 – Diagrama Entidade-Relacionamento do schema da aplicação 27
Figura 4 – Diagrama de arquitetura geral do sistema 28
Figura 5 – Diagrama de sequência: Login 29
Figura 6 – Diagrama de sequência: Renovação do Token de Acesso . . . 35
Figura 7 – Diagrama de sequência: Geração de Guia de Conteúdo, Requisição e Inferência de Padrões 36
Figura 8 – Diagrama de sequência: Geração de Guia de Conteúdo, Síntese via LLM e Persistência 37
Figura 9 – Diagrama de fluxo: Integração e Entrega Contínuas 38
Figura 10 – Página inicial 40
Figura 11 – Painel principal 40
Figura 12 – Tela de geração 41
Figura 13 – Detalhamento de uma geração no histórico 41
Figura 14 – Fluxo de dados: da coleta ao guia final 42

Lista de Quadros

Quadro 1 – Requisitos funcionais do sistema 17
Quadro 2 – Requisitos não funcionais do sistema 20
Quadro 3 – Fluxo do caso de uso Solicitar Geração de Conteúdo 21
Quadro 4 – Fluxo do caso de uso Gerenciar Nichos 22
Quadro 5 – Fluxo do caso de uso Fazer Cadastro 23

Lista de abreviaturas e siglas

API	Application Programming Interface (Interface de Programação de Aplicações)
BERT	Bidirectional Encoder Representations from Transformers
CI/CD	Continuous Integration / Continuous Delivery (Integração Contínua / Entrega Contínua)
CSR	Client-Side Rendering (Renderização no Lado do Cliente)
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
HTTP	HyperText Transfer Protocol
IA	Inteligência Artificial
JSON	JavaScript Object Notation
JWT	JSON Web Token
LLM	Large Language Model (Modelo de Linguagem de Grande Escala)
ORM	Object-Relational Mapping (Mapeamento Objeto-Relacional)
PLN	Processamento de Linguagem Natural
SEO	Search Engine Optimization (Otimização para Mecanismos de Busca)
SSR	Server-Side Rendering (Renderização no Lado do Servidor)
TF-IDF	Term Frequency – Inverse Document Frequency

Sumário

1 INTRODUÇÃO 10
1.1 Problema 10
1.2 Justificativa 11
1.3 Objetivos 12
1.3.1 Objetivo Geral 12
1.3.2 Objetivos Específicos 12
1.4 Organização do Trabalho 13

2 REFERENCIAL TEÓRICO 14
2.1 Trabalhos Relacionados 14
2.2 Engenharia de Software: Conceitos e Práticas Adotadas . . 15
2.3 Conceitos de PLN e Extração de Padrão 15

3 ESPECIFICAÇÃO E ARQUITETURA DO SISTEMA 17
3.1 Requisitos Funcionais 17
3.2 Requisitos Não Funcionais 19
3.3 Casos de Uso 21
3.4 Modelagem de Dados 24
3.4.1 Esquema Relacional 24
3.4.2 Camadas de Dados do Pipeline de IA 24
3.5 Arquitetura Geral 26
3.6 Fluxos Principais 28
3.7 Pipeline de PLN/LLM 30
3.7.1 Normalização de Idioma 30
3.7.2 Geração de Embeddings Semânticos 31
3.7.3 Agrupamento por Padrão Estilístico 31
3.7.4 Extração de Termos e Exemplos Representativos 31
3.8 Tecnologias e Ferramentas 32

3.9 Estratégia de Testes 33
3.10 Infraestrutura e CI/CD 34

4 IMPLEMENTAÇÃO 39
4.1 Telas e Interface do Sistema 39
4.2 Fluxo Ponta a Ponta / Estudo de Caso 41
4.3 Decisões e Trade-offs 43

5 RESULTADOS 44
5.1 Validação dos Guias Gerados 44
5.2 Discussão / Limitações Observadas 45

6 CONCLUSÃO 47
6.1 Síntese 47
6.2 Limitações 48
6.3 Trabalhos Futuros 48

Referências 50

APÊNDICE A – EXEMPLO DE GUIA GERADO 52

APÊNDICE B – ESTRUTURA DO JSON DE PADRÕES POR NICHO 58

1 Introdução

O vídeo curto se tornou um dos formatos dominantes de consumo audiovisual. TikTok, Reels (Instagram) e Shorts (YouTube) somam, respectivamente, 1,6 bilhão, 2 bilhões e 1,5 bilhão de usuários mensais ativos (ROBERTS; DAVID, 2025). O tempo médio diário de uso chega a 54, 33 e 49 minutos em cada uma dessas plataformas (ROBERTS; DAVID, 2025). Esse alcance supera em muito o vídeo tradicional, de produção mais longa e elaborada. Um ciclo de produção mais curto ampliou o número de criadores ativos, mesmo sem estrutura profissional.

Esse crescimento não distribuiu o interesse do público de forma uniforme. O conteúdo se organiza em nichos, temas e formatos compartilhados por criadores e espectadores. Histórias narradas, humor, curiosidades e avaliação de produtos são exemplos comuns. Dentro de cada nicho, a competição por atenção é constante. A viralização, alcance muito acima da média do criador ou do nicho, é o principal indicador de sucesso de uma publicação.

A viralização pode parecer resultado do acaso. Ainda assim, vídeos virais de um mesmo nicho tendem a repetir padrões. Um gancho de abertura instantâneo e uma edição em ritmo acelerado ajudam a reter o espectador nos primeiros segundos (HEORHII, 2025). A compressão narrativa, abertura, desenvolvimento e fechamento em poucos segundos, também caracteriza esses vídeos (HEORHII, 2025). É desse padrão, e não da sorte, que parte o presente trabalho.

1.1 Problema

O criador de vídeo curto enfrenta uma dificuldade específica: identificar, de forma sistemática, os elementos que tornam um vídeo viral dentro do próprio nicho. Hoje, essa identificação depende de dois caminhos, ambos limitados.

O primeiro caminho é a tentativa e erro. O criador publica com base na própria intuição e observa o resultado depois. Esse ciclo é lento, já que cada ajuste só é validado após novo vídeo publicado. Também é caro: uma escolha equivocada custa alcance e tempo de produção. E aprende apenas com a própria amostra, pequena diante do volume de vídeos publicados no nicho.

O segundo caminho é a análise manual de concorrentes. O criador assiste a vídeos de referência e tenta identificar, por conta própria, o que funciona. Esse processo não é sistemático: cada observação depende do critério de quem assiste e não escala. O número de vídeos que uma pessoa consegue analisar é pequeno diante do necessário para confirmar um padrão real. Padrões estruturais mais sutis, presentes apenas quando muitos vídeos são comparados, tendem a passar despercebidos.

Nenhum dos dois caminhos oferece ao criador uma leitura confiável do próprio nicho. É essa lacuna, entre a necessidade de um padrão confiável e a ausência de um método sistemático para encontrá-lo, que o presente trabalho busca resolver.

1.2 Justificativa

Sem um método sistemático de leitura do próprio nicho, o criador permanece na dependência de intuição descrita na seção anterior. A literatura sobre estratégia de conteúdo trata essa dependência como o ponto de partida a ser superado: estratégia orientada por dado é definida pelo uso de métricas de engajamento para ajustar tópico, formato, horário e plataforma de publicação, em oposição direta à publicação guiada apenas por intuição (KORDZADEH; YOUNG, 2020). Persistir na tentativa e erro significa abrir mão de um processo cuja alternativa já está documentada.

A relevância de fechar essa lacuna se sustenta em resultado concreto. Um estudo de otimização de conteúdo em múltiplos canais reportou aumento na taxa de interação de 8,9% para 12,4%, além de crescimento de 50% na taxa

de cliques, ao aplicar análise preditiva sobre o conteúdo publicado, com TikTok e Instagram registrando as maiores taxas de interação entre as plataformas analisadas (SWAMI et al., 2025). O ganho concentrado justamente nessas plataformas de vídeo curto reforça que a ausência de um método sistemático representa perda de desempenho mensurável, não apenas uma lacuna teórica.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um sistema de extração automatizada de padrões de conteúdo viral em vídeos curtos, segmentados por nicho, e de geração assistida de guias de conteúdo estratégico a partir desses padrões.

1.3.2 Objetivos Específicos

1. Coletar metadados e transcrições de vídeos curtos segmentados por nicho, via interfaces de programação do YouTube, formando a camada prata do conjunto de dados.

2. Processar a camada prata por meio de processamento de linguagem natural sobre título, descrição, transcrição e elementos visuais, classificando o formato narrativo de cada vídeo por agrupamento não supervisionado e formando a camada ouro.

3. Consolidar, a partir da camada ouro, os padrões de conteúdo viral de cada nicho, extraindo termos discriminativos e exemplos representativos por meio de agrupamento baseado em densidade.

4. Implementar um mecanismo de similaridade semântica capaz de inferir padrões para nichos ainda não presentes na base consolidada.

5. Gerar, a partir dos padrões consolidados, guias de conteúdo estruturados

em abertura, desenvolvimento e fechamento, por meio de modelo de linguagem.

6. Desenvolver uma aplicação web *full-stack* que exponha o serviço ao usuário final, permitindo vincular nichos, solicitar gerações e consultar o histórico de conteúdo gerado.

1.4 Organização do Trabalho

O Capítulo 2 apresenta o referencial teórico do trabalho, comparando-o com ferramentas e soluções já existentes e revisando os conceitos de engenharia de software e de processamento de linguagem natural que fundamentam as escolhas técnicas do projeto. O Capítulo 3 descreve a especificação e a arquitetura do sistema, dos requisitos levantados até a infraestrutura de implantação. O Capítulo 4 apresenta a implementação construída, cobrindo as principais telas do sistema, um fluxo real de uso e as decisões de arquitetura assumidas ao longo do desenvolvimento. O Capítulo 5 discute os resultados obtidos, avaliando os guias gerados para diferentes nichos e as limitações observadas na operação do sistema. Por fim, o Capítulo 6 apresenta as considerações finais do trabalho, retomando os objetivos alcançados e apontando direções para trabalhos futuros.

2 Referencial Teórico

Este capítulo situa o trabalho frente ao que já existe: ferramentas voltadas ao criador de conteúdo, os conceitos de engenharia de software que sustentam a construção do sistema e os conceitos de processamento de linguagem natural por trás da extração dos padrões.

2.1 Trabalhos Relacionados

Ferramentas como TubeBuddy e VidIQ já auxiliam criadores de vídeo, oferecendo sugestões de palavra-chave, análise de tendência e otimização de título e descrição a partir de dados de desempenho do próprio canal. Nenhuma delas, porém, entrega um guia de produção estruturado, abertura, desenvolvimento e fechamento, adaptado ao padrão viral específico de um nicho.

Essa lacuna se torna mais evidente quando a geração de conteúdo passa a depender de modelos de linguagem. A adaptação cultural de saídas de LLM exige mais do que tradução: um arcabouço que combina detecção de contexto cultural, integração de conhecimento e uma etapa explícita de verificação elevou a adequação cultural das respostas de 62–83% (baseline) para 91%, com a detecção de contexto isolada respondendo pela maior parte desse ganho (LUO, 2023). É essa combinação, entre extração de padrão real de nicho e adaptação cultural fundamentada na geração, que diferencia o presente trabalho das ferramentas de otimização hoje disponíveis.

2.2 Engenharia de Software: Conceitos e Práticas Adotadas

Os três serviços do sistema (`web`, `server`, `ia`) são versionados em um único repositório por meio de *workspaces*, estrutura de monorepo que favorece a visibilidade entre projetos e centraliza a coordenação de mudanças que afetam mais de um serviço (JASPAN et al., 2017). Cada serviço, ainda assim, é empacotado em um contêiner Docker próprio, garantindo que a aplicação execute de forma consistente entre o ambiente de desenvolvimento e o de produção (WAN et al., 2018).

Essa paridade entre ambientes é o que sustenta a integração contínua: a cada alteração, um pipeline automatizado valida o código sob condições equivalentes às de produção, antecipando a detecção de falhas em vez de descobri-las apenas após o deploy (JANI, 2023; NAZáRIO et al., 2025). Essa validação segue o modelo de pirâmide de testes, priorizando um volume maior de testes unitários rápidos e isolados, com uma camada menor de testes de integração concentrada nos pontos de maior risco (RADZIWILL; FREEMAN, 2020).

2.3 Conceitos de PLN e Extração de Padrão

A similaridade semântica entre textos pode ser obtida por meio de *embeddings* de sentença, representações vetoriais de tamanho fixo que capturam o significado de um texto e permitem comparação direta por similaridade de cosseno. O Sentence-BERT é um exemplo desse tipo de técnica: ao ajustar uma arquitetura BERT em par siamês, ele reduz drasticamente o custo computacional de comparar grandes volumes de texto frente a abordagens que processam cada par de sentenças individualmente (REIMERS; GUREVYCH, 2019).

Um segundo conceito é o agrupamento não supervisionado baseado em densidade, no qual clusters são definidos como regiões de alta densidade

separadas por regiões de baixa densidade, sem exigir que o número de grupos seja definido previamente. O HDBSCAN é um exemplo dessa abordagem: ele constrói uma hierarquia de densidade e extrai dela um agrupamento plano, o que o torna mais robusto a clusters de densidade variável do que métodos que dependem de um único limiar fixo (MCINNES; HEALY; ASTELS, 2017).

Por fim, a extração de termos discriminativos de um texto costuma combinar duas etapas conceituais distintas: uma métrica de relevância de termo, como o TF-IDF, que pondera uma palavra pela frequência no documento em relação à frequência no corpus; e um mecanismo de reordenação por diversidade, que penaliza termos redundantes entre si para ampliar a cobertura temática dos termos selecionados (CRISTIN et al., 2025).

3 Especificação e Arquitetura do Sistema

Definida a fundamentação teórica que sustenta as escolhas técnicas do projeto, parte-se agora para a sua concretização. Este capítulo descreve como o sistema foi especificado e estruturado, partindo dos requisitos levantados e avançando até os fluxos de execução e a infraestrutura que sustenta seu funcionamento, base sobre a qual a implementação apresentada no capítulo seguinte se apoia.

3.1 Requisitos Funcionais

Os requisitos funcionais (RF) descrevem as ações que o sistema deve executar, levantados a partir das rotas e regras de negócio implementadas nos serviços de *server* e *web*. O Quadro 1 apresenta a lista consolidada.

Quadro 1 – Requisitos funcionais do sistema

ID	**Descrição**
RF01	O sistema deve exibir uma página inicial pública com acesso às opções de login e de cadastro.
RF02	O sistema deve apresentar, na página inicial, as funcionalidades oferecidas e o funcionamento do processo de geração de guias.
RF03	O sistema deve permitir que o usuário realize login informando email e senha.
RF04	O sistema deve permitir a alternância de visibilidade da senha digitada nos campos de login e de cadastro.
RF05	O sistema deve permitir que o usuário encerre a sessão, invalidando o refresh token associado.
RF06	O sistema deve permitir o cadastro de um novo usuário mediante nome, email, senha e confirmação de senha.

Continuação do Quadro 1

ID	**Descrição**
RF07	O sistema deve exigir, no cadastro, a seleção de ao menos um nicho, existente ou recém-criado.
RF08	O sistema deve permitir que o usuário visualize os dados do próprio perfil.
RF09	O sistema deve permitir que o usuário atualize nome e email do próprio perfil.
RF10	O sistema deve permitir que o usuário altere a própria senha.
RF11	O sistema deve permitir que o usuário exclua a própria conta, com exclusão lógica (soft delete) e preservação do histórico de gerações.
RF12	O sistema deve permitir que o usuário visualize o conjunto global de nichos disponíveis.
RF13	O sistema deve permitir que o usuário visualize os nichos vinculados à própria conta.
RF14	O sistema deve permitir que o usuário vincule à própria conta um nicho existente ou crie e vincule um nicho novo.
RF15	O sistema deve permitir que o usuário desvincule um nicho da própria conta, impedindo a remoção quando este for o único nicho vinculado.
RF16	O sistema deve permitir que o usuário selecione um nicho vinculado à própria conta e, opcionalmente, um público-alvo, para solicitar uma geração de guia de conteúdo.
RF17	O sistema deve exibir uma tela de carregamento durante o processamento da geração e apresentar o resultado, composto por estrutura e roteiro, ao término.
RF18	O sistema deve permitir que o usuário visualize o histórico de gerações realizadas, com filtro por favoritas.

Continuação do Quadro 1

ID	**Descrição**
RF19	O sistema deve permitir que o usuário visualize os detalhes de uma geração específica.
RF20	O sistema deve permitir que o usuário marque ou desmarque uma geração como favorita.
RF21	O sistema deve permitir que o usuário envie feedback, de forma independente, para a estrutura e para o roteiro de uma geração.
RF22	O sistema deve restringir o acesso à área de gestão a usuários com role ADMIN ou SUPPORTER, mantendo os usuários com role USER restritos à área de aplicação do cliente.
RF23	O sistema deve restringir, exclusivamente a usuários com role ADMIN, a listagem, a alteração de role e a exclusão de usuários.
RF24	O sistema deve restringir, exclusivamente a usuários com role ADMIN, a consulta de estatísticas de usuários, nichos, gerações e feedbacks.

Fonte: elaborado pelo autor (2026).

3.2 Requisitos Não Funcionais

Os requisitos não funcionais (RNF) descrevem as restrições e atributos de qualidade que o sistema deve satisfazer, organizados por categoria. Foram levantados a partir da configuração de segurança do servidor (*app.ts*), dos serviços de autenticação e do comportamento da interface. O Quadro 2 apresenta a lista consolidada.

Quadro 2 – Requisitos não funcionais do sistema

ID	**Categoria**	**Descrição**
RNF01	Segurança	Senhas devem ser armazenadas com hash (bcrypt, fator de custo 10), nunca em texto plano.
RNF02	Segurança	O token de acesso (JWT) deve expirar em 10 minutos.
RNF03	Segurança	O token de atualização (refresh token) deve expirar em 7 dias e ser armazenado no banco de dados.
RNF04	Segurança	O sistema deve invalidar o refresh token no momento do logout.
RNF05	Segurança	O sistema deve exigir a confirmação da senha atual antes de permitir sua alteração.
RNF06	Segurança	O sistema deve aplicar cabeçalhos de segurança HTTP (Content-Security-Policy), restringindo a origem de scripts, estilos e fontes carregados pela aplicação.
RNF07	Segurança	O sistema deve limitar a taxa de requisições por cliente a 100 por minuto, retornando código 429 quando o limite for excedido.
RNF08	Confiabilidade	O sistema deve retornar erro apropriado (502) e evitar a criação de uma geração inconsistente caso o serviço de IA esteja indisponível ou falhe ao gerar o conteúdo solicitado.
RNF09	Usabilidade	O sistema deve exibir indicadores de carregamento durante requisições assíncronas nas páginas de listagem e estatística.
RNF10	Usabilidade	A interface deve ser responsiva, adaptando a navegação para uso em dispositivos móveis.

Fonte: elaborado pelo autor (2026).

3.3 Casos de Uso

Os casos de uso descrevem as interações entre os atores do sistema e as funcionalidades disponíveis, levantados a partir das rotas do *server* e das telas do *web*. Três atores foram identificados: o Visitante, que ainda não possui conta e acessa apenas o cadastro e o login; o Usuário, que concentra a maior parte das funcionalidades do sistema, incluindo o vínculo de nichos e a solicitação de gerações; e o Administrador, restrito à área de gestão, com acesso a estatísticas e ao gerenciamento de usuários. Pela quantidade de casos de uso envolvidos, o diagrama foi dividido em duas figuras. A Figura 1 apresenta os casos de uso de acesso e conta, cobrindo o Visitante e as ações do Usuário sobre o próprio perfil e os nichos vinculados.

A Figura 2 apresenta os casos de uso de geração e gestão, cobrindo a solicitação de conteúdo e o histórico de gerações do Usuário, além da área de gestão restrita ao Administrador.

Entre os casos de uso identificados, três concentram regras de negócio relevantes o suficiente para justificar um detalhamento de fluxo: a solicitação de geração de conteúdo, por ser o caso de uso central do sistema; o gerenciamento de nichos, por sustentar a segmentação sobre a qual os padrões virais são extraídos; e o cadastro de usuário, por embutir a exigência de vínculo de nicho já no primeiro acesso. Os Quadros 3, 4 e 5 descrevem, respectivamente, a pré-condição, o fluxo principal e os fluxos alternativos de cada um.

Quadro 3 – Fluxo do caso de uso Solicitar Geração de Conteúdo

Ator	Usuário
Pré-condição	O usuário está autenticado e possui ao menos um nicho vinculado à própria conta.

Continuação do Quadro 3	
Fluxo principal	1. O usuário acessa a página de geração. 2. O usuário seleciona um nicho vinculado à própria conta. 3. O usuário informa, opcionalmente, um público-alvo. 4. O usuário aciona o botão de gerar. 5. O sistema exibe uma tela de carregamento e envia a requisição ao serviço de IA. 6. O serviço de IA consulta os padrões extraídos para o nicho selecionado e sintetiza o guia de conteúdo. 7. O sistema persiste a geração e exibe o resultado, composto por estrutura e roteiro.
Fluxo alternativo	A1. Nicho sem padrões previamente extraídos: o serviço de IA aciona o mecanismo de similaridade semântica para redirecionar a geração ao nicho mais próximo, preservando o nicho originalmente solicitado para fins de rastreabilidade. A2. Falha do serviço de IA: o sistema retorna erro (502) ao usuário e não persiste uma geração inconsistente.

Fonte: elaborado pelo autor (2026).

Quadro 4 – Fluxo do caso de uso Gerenciar Nichos

Ator	Usuário
Pré-condição	O usuário está autenticado.

Continuação do Quadro 4	
Fluxo principal	1. O usuário acessa a página de nichos. 2. O sistema exibe o conjunto global de nichos e os nichos já vinculados à conta do usuário. 3. O usuário seleciona um nicho existente ou informa o nome de um nicho novo. 4. O sistema vincula o nicho à conta do usuário, criando o nicho no conjunto global caso ainda não exista. 5. O usuário pode, alternativamente, optar por desvincular um nicho já associado à própria conta.
Fluxo alternativo	A1. Vínculo já existente: o sistema recusa a operação (409) caso o nicho selecionado já esteja vinculado à conta. A2. Único nicho vinculado: o sistema recusa a desvinculação (422) caso o nicho selecionado seja o único vinculado à conta do usuário.

Fonte: elaborado pelo autor (2026).

Quadro 5 – Fluxo do caso de uso Fazer Cadastro

Ator	Visitante
Pré-condição	O visitante não possui conta no sistema.
Fluxo principal	1. O visitante acessa a página de cadastro. 2. O visitante informa nome, email, senha e confirmação de senha. 3. O visitante seleciona um nicho existente ou cria um nicho novo, etapa obrigatória do cadastro. 4. O visitante confirma o cadastro. 5. O sistema cria a conta, vincula o nicho selecionado e autentica o usuário.

Continuação do Quadro 5	
Fluxo alternativo	A1. Email já cadastrado: o sistema recusa o cadastro (409) e informa o conflito ao visitante.

Fonte: elaborado pelo autor (2026).

3.4 Modelagem de Dados

A modelagem de dados do sistema cobre duas partes: o esquema relacional que sustenta a aplicação e as camadas de dados que alimentam o pipeline de IA. A Figura 3 mostra o diagrama entidade-relacionamento do esquema relacional.

3.4.1 Esquema Relacional

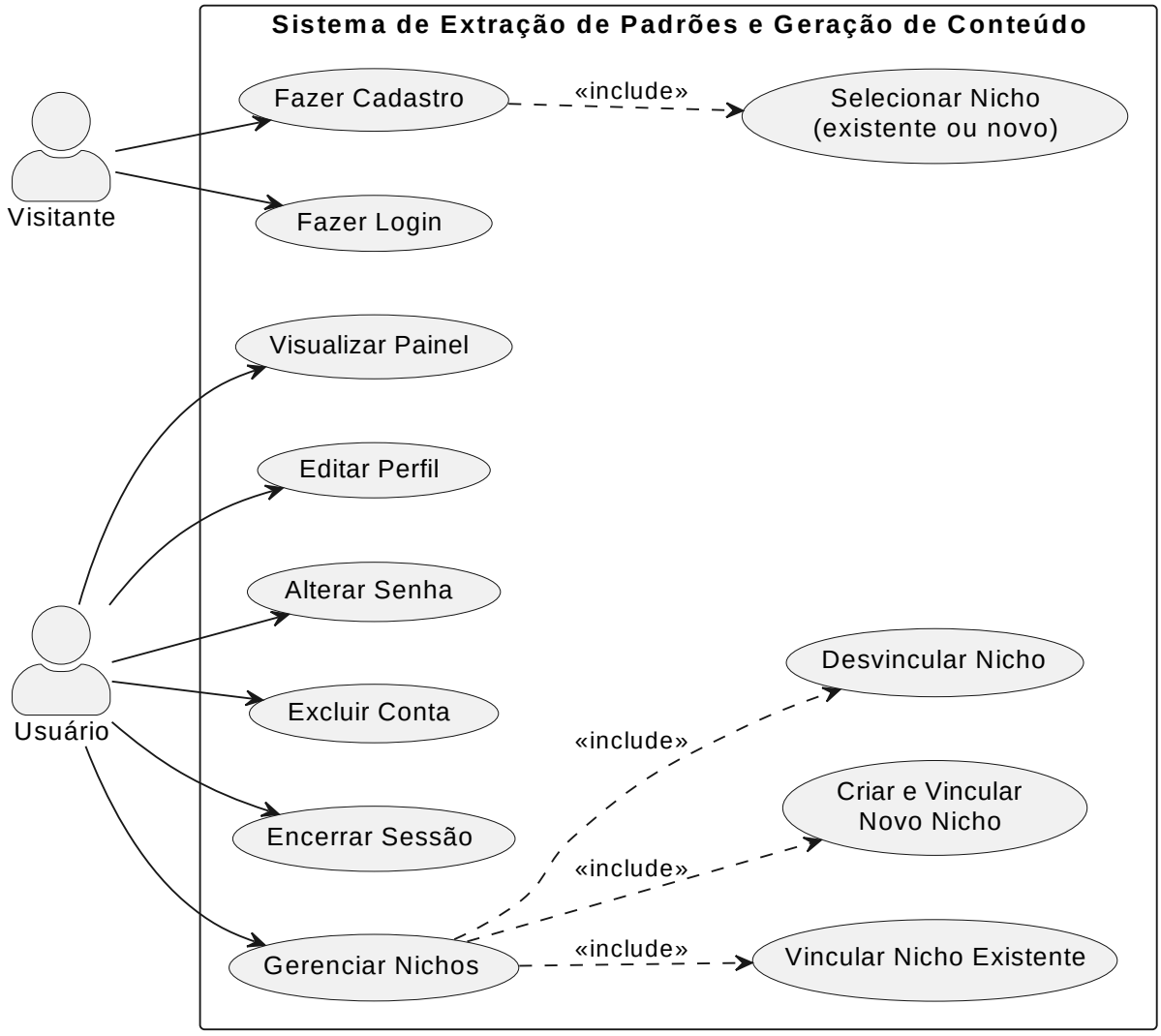
O esquema prioriza preservar o histórico da aplicação em vez de remover dados fisicamente. A exclusão de usuário é lógica, não física, e as regras de exclusão em cascata seguem a mesma lógica: registros sem valor independente, como tokens e feedbacks, somem junto com o registro pai, enquanto nichos e gerações ficam protegidos contra remoção enquanto houver vínculo ativo.

O nicho é modelado como entidade própria, e não como texto livre na geração, porque precisa ser comparável entre usuários e agregável em estatísticas. Cada geração também guarda o nicho originalmente pedido, separado do nicho efetivamente usado, já que o serviço de ML pode redirecionar a geração por similaridade quando não encontra correspondência exata. Essa separação preserva a rastreabilidade entre a intenção do usuário e o resultado do sistema.

3.4.2 Camadas de Dados do Pipeline de IA

O pipeline de IA organiza os dados em duas camadas de refinamento progressivo, prática comum em arquiteturas de dados estruturadas por camadas

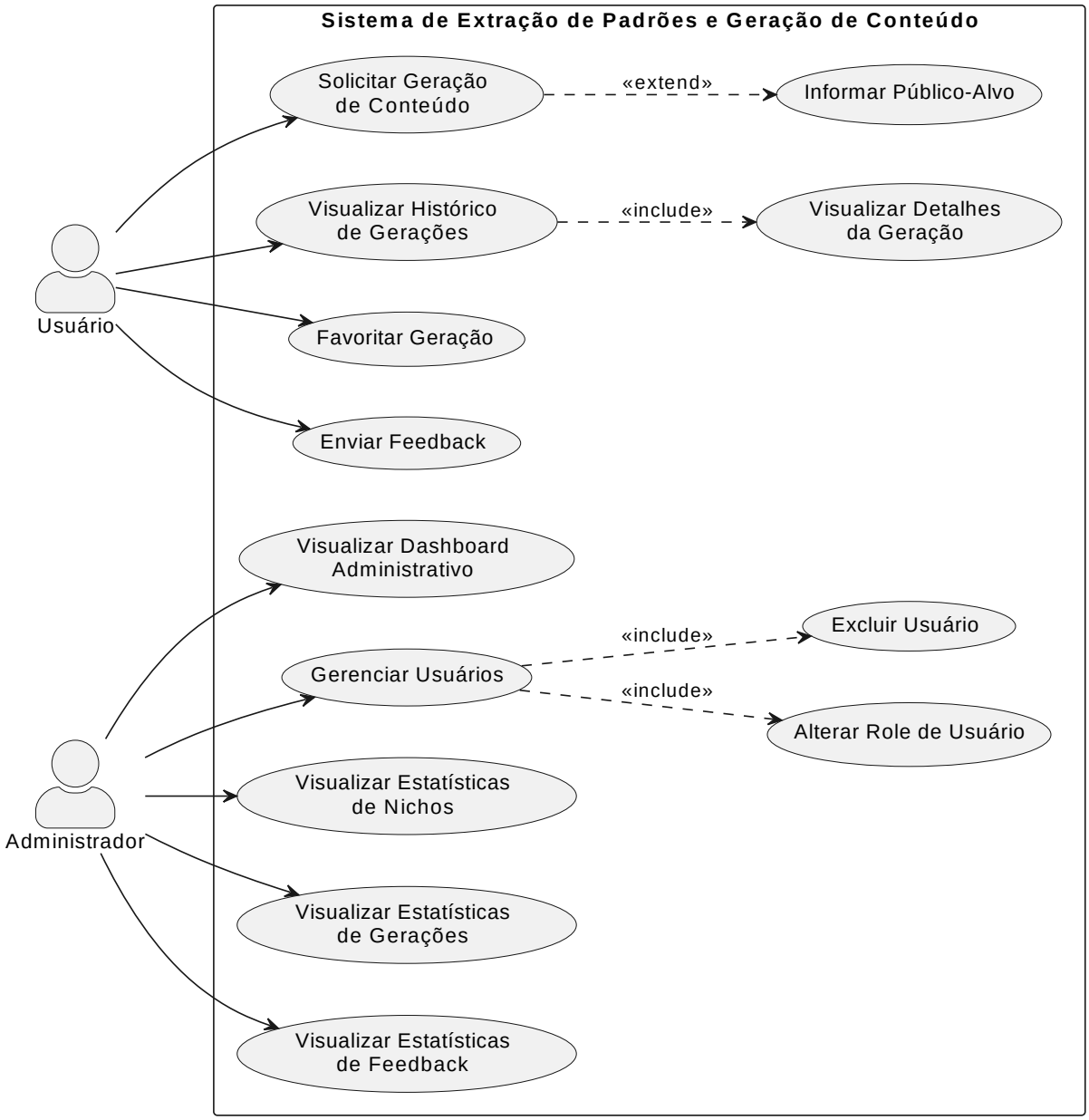
Figura 1 – Diagrama de Casos de Uso: Acesso e Conta



Fonte: O autor (2026)

(SAWADOGO; DARMONT, 2020). A camada prata reúne o material coletado e já limpo, servindo de base tanto para a análise exploratória quanto para o pré-processamento seguinte. A camada ouro reúne as *features* derivadas dessa base, selecionadas e criadas especificamente para alimentar a extração de padrões. Separar as camadas isola o efeito de cada etapa: a análise exploratória e a engenharia de *features* evoluem de forma independente, sem exigir repetir a coleta.

Figura 2 – Diagrama de Casos de Uso: Geração e Gestão



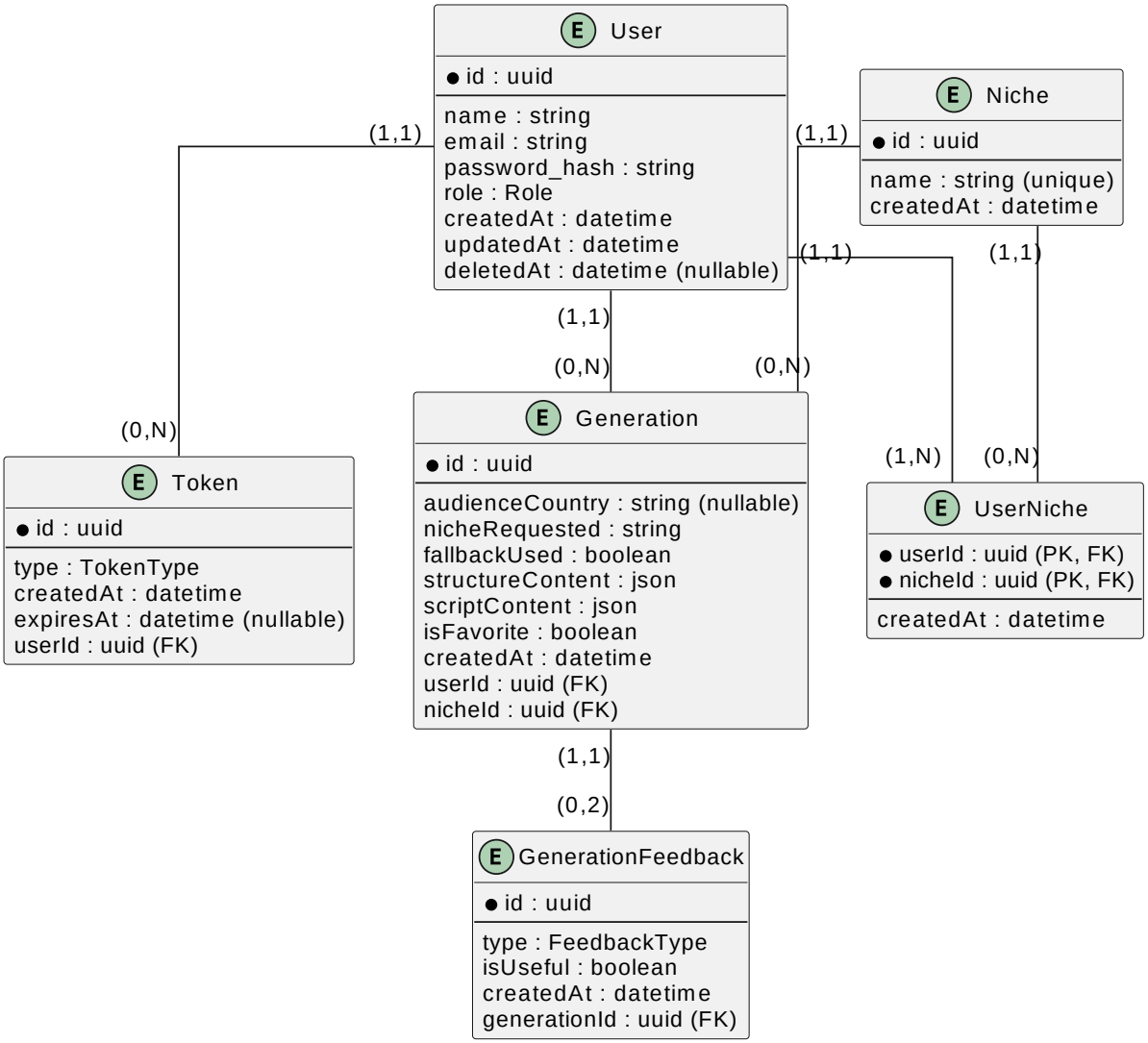
Fonte: O autor (2026)

3.5 Arquitetura Geral

A aplicação foi dividida em três serviços independentes, em vez de uma aplicação única, conforme mostra a Figura 4. O *web* concentra a interface do usuário, o *server* cuida de autenticação, orquestração das gerações e persistência dos dados, e o *ia* é responsável pela geração de conteúdo por aprendizado de máquina.

A geração de conteúdo depende de um ecossistema de bibliotecas pró-

Figura 3 – Diagrama Entidade-Relacionamento do schema da aplicação

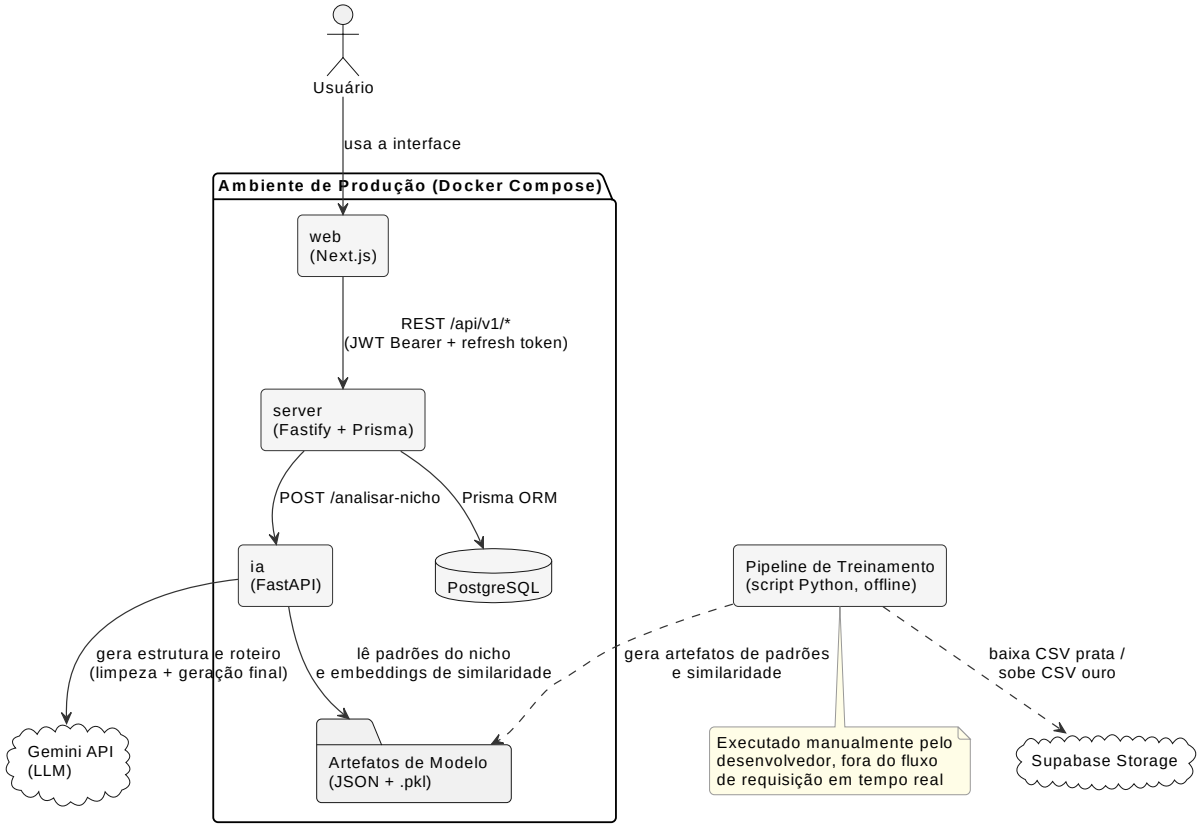


Fonte: O autor (2026)

prio, voltado a PLN, aprendizado de máquina e orquestração de modelos de linguagem. Esse ecossistema difere do necessário para autenticação e entrega web. Manter esse serviço isolado permite que sua stack evolua de forma independente, sem impor restrições de versão ou dependência ao restante do sistema.

O processamento mais custoso da aplicação é a extração dos padrões que orientam cada geração. Ela ocorre em um pipeline de treinamento executado separadamente, antes de qualquer requisição do usuário. O pipeline produz artefatos prontos para uso, que o serviço de IA apenas consulta no momento da geração. Assim, o tempo de resposta de uma geração depende apenas dos

Figura 4 – Diagrama de arquitetura geral do sistema



Fonte: O autor (2026)

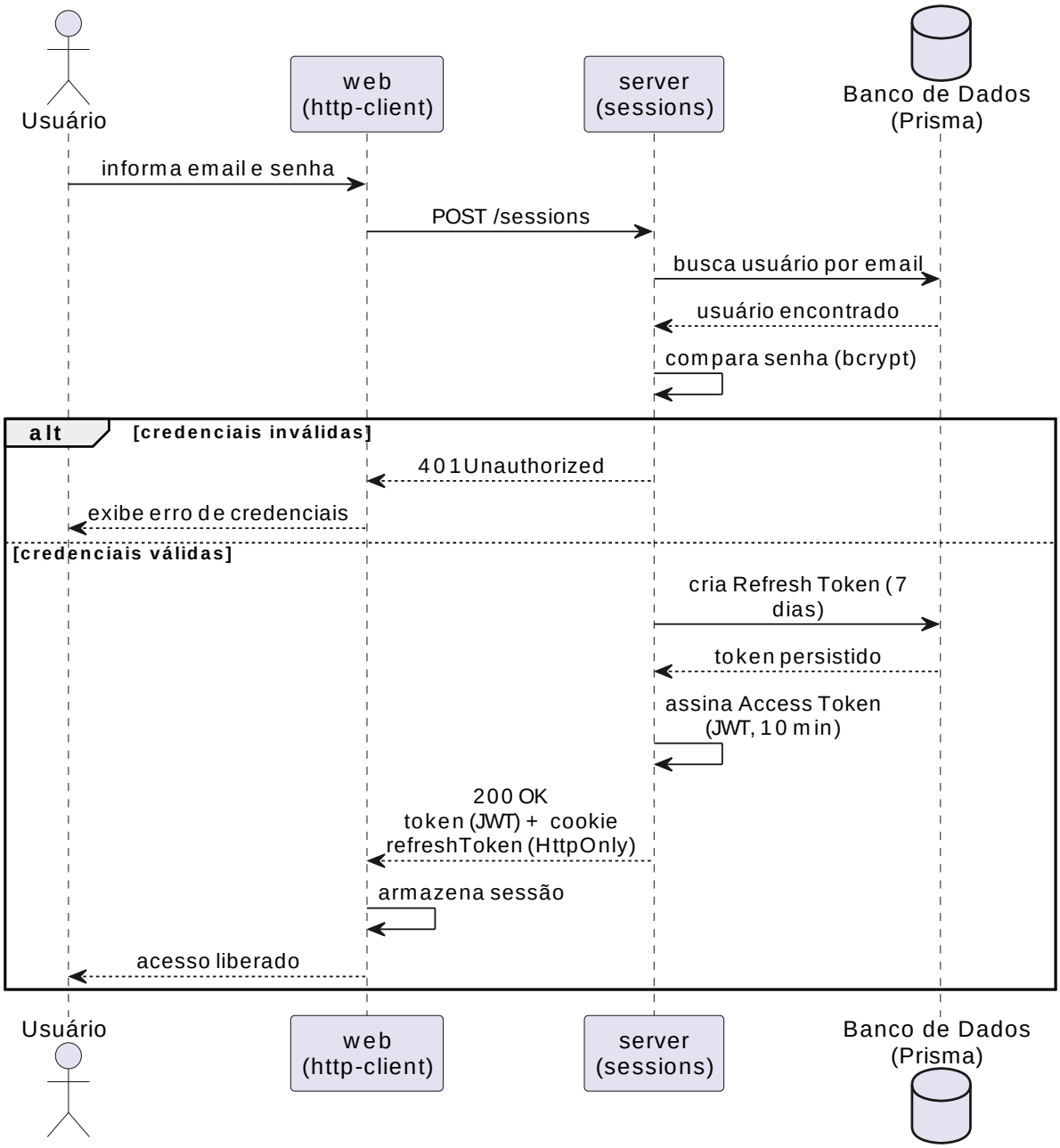
artefatos disponíveis e da chamada ao modelo de linguagem, não do tamanho do histórico de treinamento.

3.6 Fluxos Principais

Os fluxos principais detalham, em nível de sequência, as interações entre os serviços do sistema para os cenários mais críticos de uso: a autenticação do usuário, a renovação do token de acesso e a geração de um guia de conteúdo. A Figura 5 apresenta o fluxo de login, cobrindo a validação de credenciais e a emissão do par de tokens.

Como o token de acesso expira em dez minutos (RNF02), o sistema precisa renová-lo de forma transparente ao usuário sempre que uma requisição autenticada falha por expiração. A Figura 6 apresenta esse fluxo, incluindo a fila de renovação no cliente e a rotação do refresh token no servidor.

Figura 5 – Diagrama de sequência: Login



Fonte: O autor (2026)

Por fim, o fluxo ponta a ponta da geração de um guia de conteúdo é apresentado em duas figuras, dada a quantidade de serviços e decisões envolvidas. A Figura 7 cobre a solicitação inicial e a inferência de padrões, incluindo os dois pontos de possível redirecionamento por fallback: a ausência de vínculo do nicho à conta do usuário e a ausência de padrões extraídos diretamente para o nicho solicitado, resolvida por similaridade semântica.

A Figura 8 dá continuidade ao fluxo, cobrindo a síntese do conteúdo

pelas duas chamadas ao modelo de linguagem, uma de limpeza do dado bruto e outra de geração final, e a persistência do resultado.

3.7 Pipeline de PLN/LLM

O pipeline de extração de padrões transforma o conjunto de dados coletado, na camada prata, no conjunto de dados que alimenta a geração dos guias, na camada ouro. As etapas seguem uma ordem fixa: normalização de idioma, geração de embeddings semânticos, agrupamento por padrão estilístico e extração de termos e exemplos representativos.

Diferente da geração de um guia, que responde a uma requisição do usuário, esse pipeline roda de forma independente e antecipada. O resultado é um arquivo de padrões por nicho, consultado posteriormente sem necessidade de reprocessar o conjunto de dados a cada geração.

3.7.1 Normalização de Idioma

O conjunto de dados é multilíngue, já que os vídeos são coletados diretamente do YouTube. Antes de qualquer etapa de PLN, o texto falado é traduzido para o inglês, unificando os dados em um único idioma. Essa normalização reduz a variação textual e permite reaproveitar ferramentas de PLN já consolidadas para o inglês, em vez de adaptar cada técnica a múltiplos idiomas (LIND et al., 2021).

A tradução ocorre antes das demais etapas do pipeline porque as técnicas seguintes, sobretudo a geração de embeddings semânticos, dependem de um vocabulário consistente para produzir comparações confiáveis entre vídeos de origens distintas.

3.7.2 Geração de Embeddings Semânticos

Os textos normalizados são convertidos em vetores por meio do Sentence-BERT. Diferente de embeddings tradicionais no nível de palavra, o Sentence-BERT produz um vetor por sentença, o que permite comparar o significado de textos inteiros por similaridade de cosseno (REIMERS; GUREVYCH, 2019).

Essa representação vetorial é a base para as duas etapas seguintes: é sobre ela que o agrupamento identifica vídeos estilisticamente semelhantes, e é a partir dela que os exemplos mais representativos de cada padrão são posteriormente selecionados.

3.7.3 Agrupamento por Padrão Estilístico

Os embeddings gerados alimentam o agrupamento dos vídeos do nicho por padrão estilístico, via HDBSCAN. Diferente de algoritmos como K-means, o HDBSCAN não exige a definição prévia do número de grupos, o que é adequado a um conjunto de dados em que a quantidade de estilos dentro de um nicho não é conhecida de antemão (MCINNES; HEALY, 2017).

Cada grupo formado reúne vídeos que compartilham um mesmo padrão de construção, seja de título, de roteiro ou de ritmo. Essa organização é o que permite, na etapa seguinte, extrair exemplos que representem de fato a diversidade de estilos observada dentro do nicho, e não apenas o vídeo mais comum.

3.7.4 Extração de Termos e Exemplos Representativos

A partir dos grupos formados, dois processos ocorrem em paralelo. Termos e expressões discriminativos do nicho são identificados por TF-IDF. Um filtro semântico via Sentence-BERT elimina, em seguida, os candidatos frequentes mas sem relação real com o tema (WANG; NING, 2020). Em paralelo, exemplos reais e representativos de cada grupo são selecionados por Maximum Marginal

Relevance, técnica que privilegia ao mesmo tempo a centralidade e a diversidade dos exemplos escolhidos (CARBONELL; GOLDSTEIN-STEWART, 1998).

Esses dois resultados, termos discriminativos e exemplos reais, compõem o núcleo do arquivo de padrões de cada nicho. É esse material que orienta posteriormente a geração assistida por LLM, servindo como evidência concreta em vez de instrução abstrata sobre o que caracteriza o nicho.

3.8 Tecnologias e Ferramentas

Na persistência de dados, adotou-se o Prisma como ORM (mapeamento objeto-relacional), com tipagem estática. As consultas ao banco são verificadas antes da execução, e não apenas em tempo de execução como em consultas de texto puro (COOK; RAI, 2005). Como banco de dados, utilizou-se o PostgreSQL, compatível com a natureza relacional do modelo de dados do sistema.

A API do sistema utiliza o Fastify, framework voltado a alto desempenho sob carga. Em comparações entre frameworks Node.js, o Fastify apresentou os melhores resultados de vazão em ambientes conteinerizados (AGUS et al., 2023). Esse cenário é próximo da infraestrutura adotada no projeto. A validação de dados usa a biblioteca Zod, com o mesmo esquema de tipos compartilhado entre servidor e cliente.

O frontend foi construído com Next.js, framework React com suporte a SSR (renderização no lado do servidor). O SSR favorece o carregamento inicial e a indexação em buscadores. Já o CSR (renderização no cliente) favorece a interatividade após o carregamento (CLAPTON et al., 2025). Esse equilíbrio é relevante para a página pública do sistema.

O serviço de processamento de linguagem natural foi implementado em Python com o FastAPI, framework assíncrono. Frameworks assíncronos tendem a apresentar maior vazão e estabilidade sob concorrência do que frameworks síncronos (ALA'ANZY; YESHPATOV, 2026). Esse ganho aparece principalmente em tarefas de entrada e saída, perfil que combina com as chamadas do serviço

a modelos de linguagem externos.

3.9 Estratégia de Testes

Cada serviço adota uma estratégia de testes própria, alinhada às suas responsabilidades dentro do sistema. Essa divisão evita que um único tipo de teste seja forçado a cobrir responsabilidades muito distintas, como lógica de negócio, interface e modelos de aprendizado de máquina.

O *server* utiliza o Vitest, cobrindo tanto testes unitários de serviço quanto testes de integração de controller. Os testes de serviço isolam a lógica de negócio de cada recurso, como autenticação, geração e feedback, sem depender da camada HTTP. Já os testes de controller são executados contra a aplicação real e o banco de dados de testes, validando o comportamento efetivo de cada endpoint da API, do recebimento da requisição até a resposta (WANG et al., 2022).

O *web* também utiliza o Vitest, mas restrito a testes unitários de funções e módulos isolados, como utilitários de formatação e a camada de gerenciamento de sessão no cliente. Para o comportamento de interface como um todo, o ambiente conta com o Playwright configurado para testes end-to-end, cobrindo fluxos completos de navegação em vez de componentes isolados.

O *ia* conta com o pytest configurado como executor de testes, com um teste em execução no pipeline de integração contínua. Sua superfície testável concentra-se na API FastAPI, responsável por expor a inferência de padrões e a geração de conteúdo.

O sistema documenta sua própria API por meio do Swagger, gerado automaticamente a partir das rotas do *server*. Além dessa documentação nativa, o Insomnia é utilizado como cliente HTTP para testes manuais e exploratórios durante o desenvolvimento. Já o Husky garante uma primeira camada de verificação antes de qualquer código chegar ao repositório: a cada tentativa de commit, ele executa a checagem de lint em cada um dos três serviços, seguida

da checagem de tipos do monorepo.

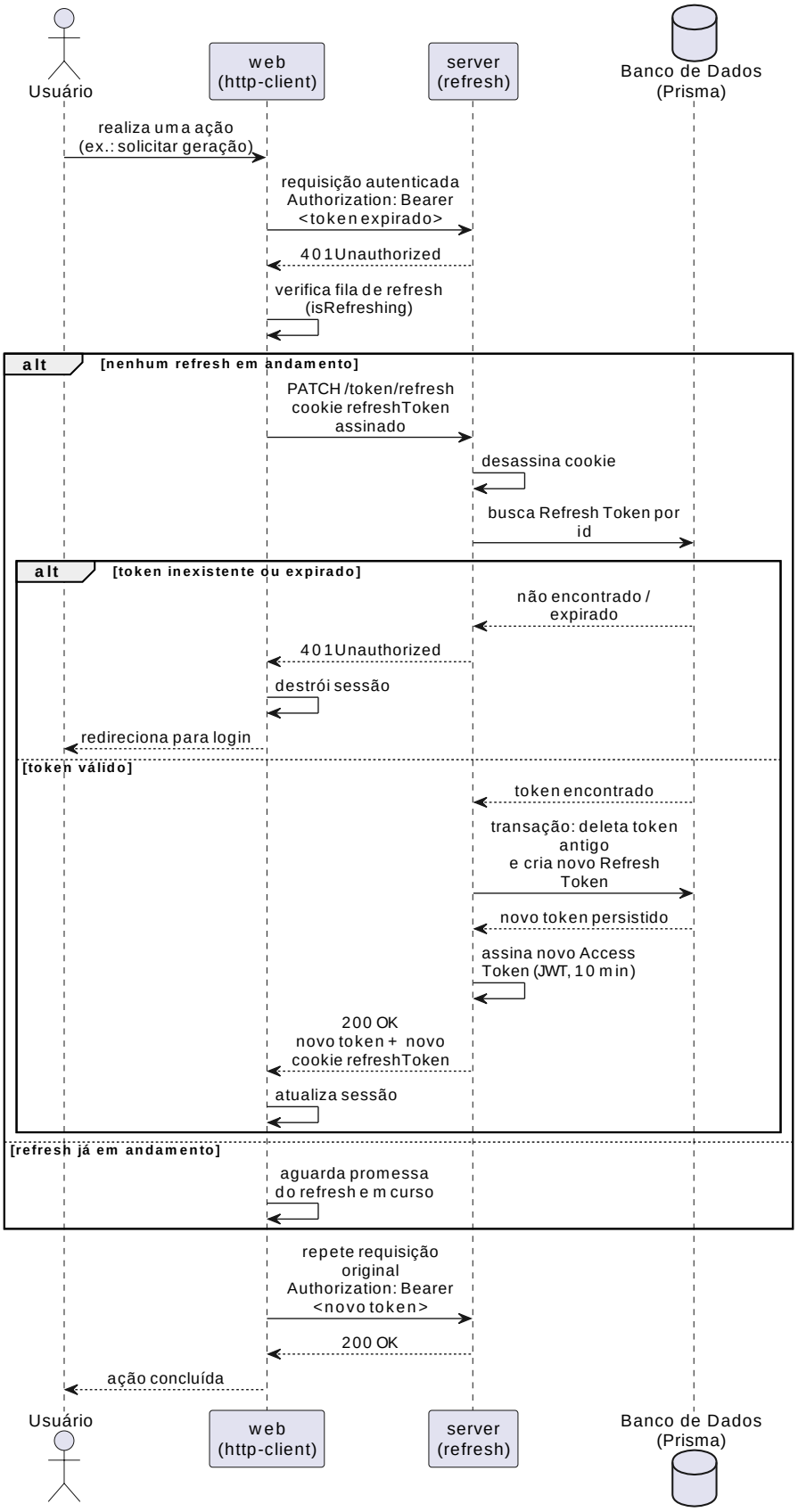
3.10 Infraestrutura e CI/CD

A infraestrutura do sistema é definida por meio de Docker Compose, garantindo que os três serviços rodem em ambientes consistentes independentemente da máquina onde são executados (CITO; GALL, 2016). Um arquivo cobre o ambiente de desenvolvimento, incluindo um banco de dados isolado para testes, mantido em memória e reiniciado a cada execução. Outro arquivo cobre o ambiente de produção, no qual os serviços consomem imagens já publicadas, em vez de serem construídos localmente.

A integração contínua valida cada mudança antes que ela avance no repositório, combinando checagem de qualidade de código, testes automatizados e testes end-to-end contra o ambiente conteinerizado. Somente após essa validação completa uma *pull request* para a branch principal é aberta automaticamente, conforme ilustra a Figura 9.

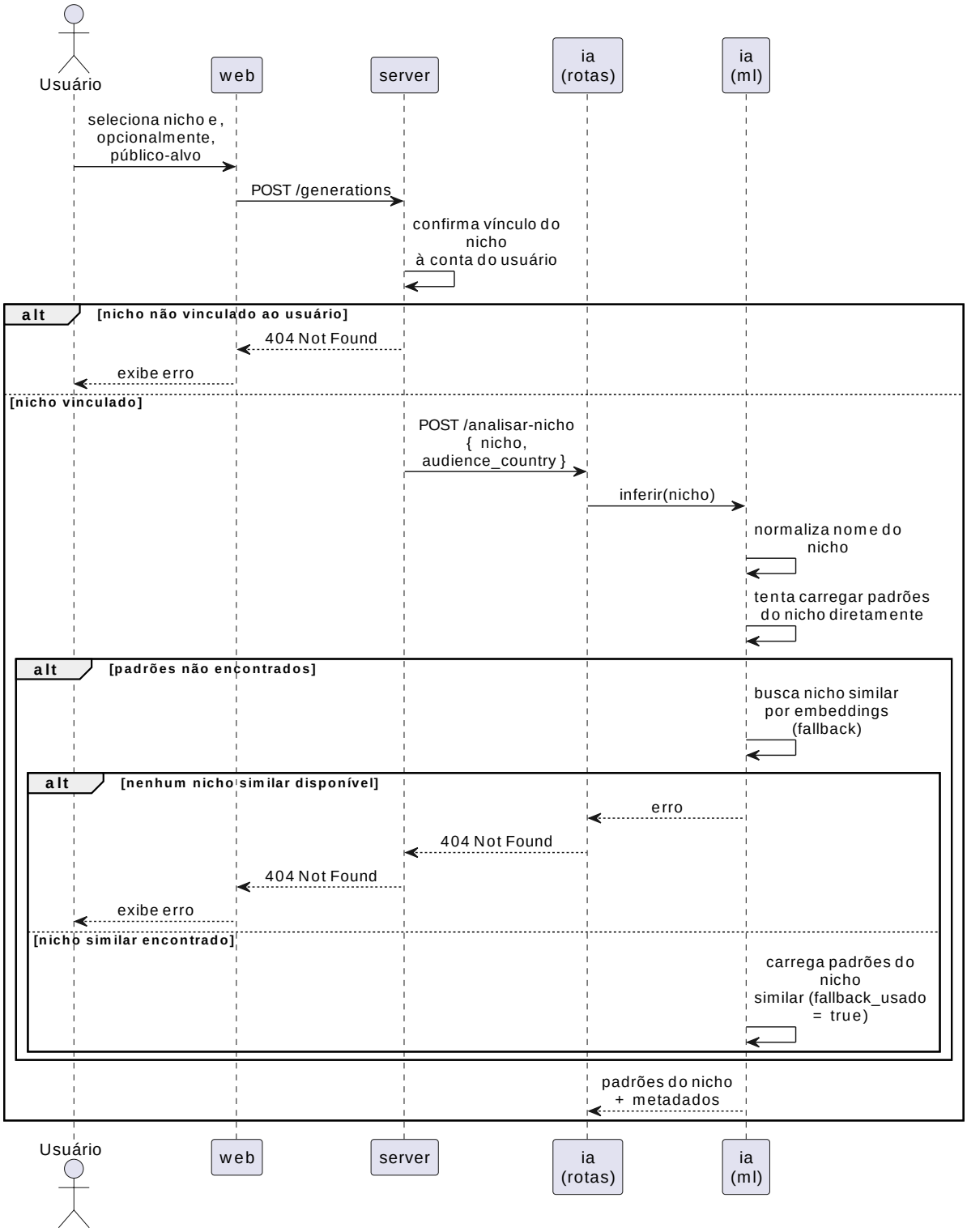
A entrega contínua ocorre quando essa *pull request* é aprovada e mesclada à branch principal, disparando a publicação das três imagens do sistema em um repositório de imagens Docker, já preparadas para múltiplas arquiteturas de processador. É essa mesma publicação que o ambiente de produção consome ao subir os serviços.

Figura 6 – Diagrama de sequência: Renovação do Token de Acesso



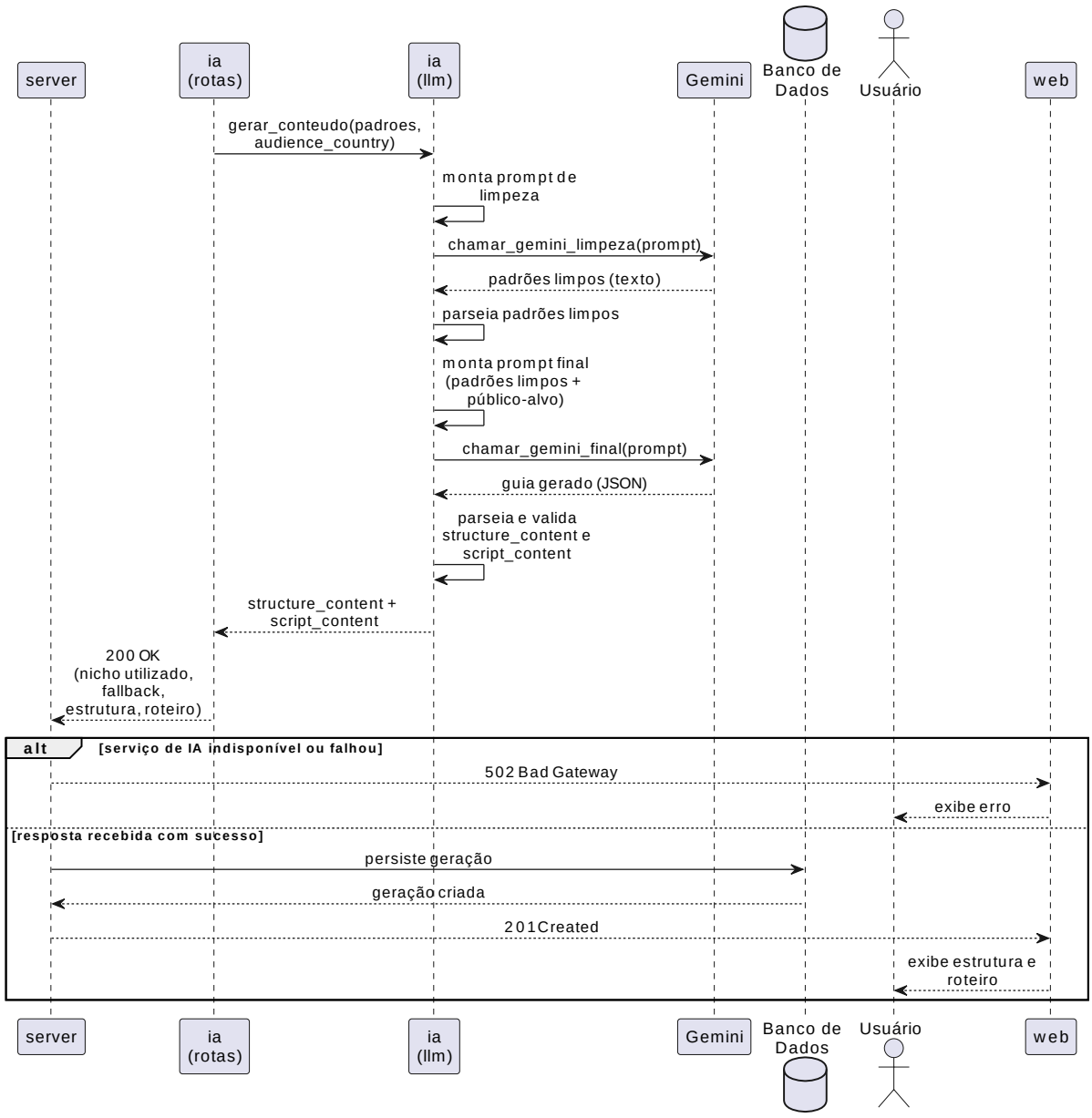
Fonte: O autor (2026)

Figura 7 – Diagrama de sequência: Geração de Guia de Conteúdo, Requisição e Inferência de Padrões



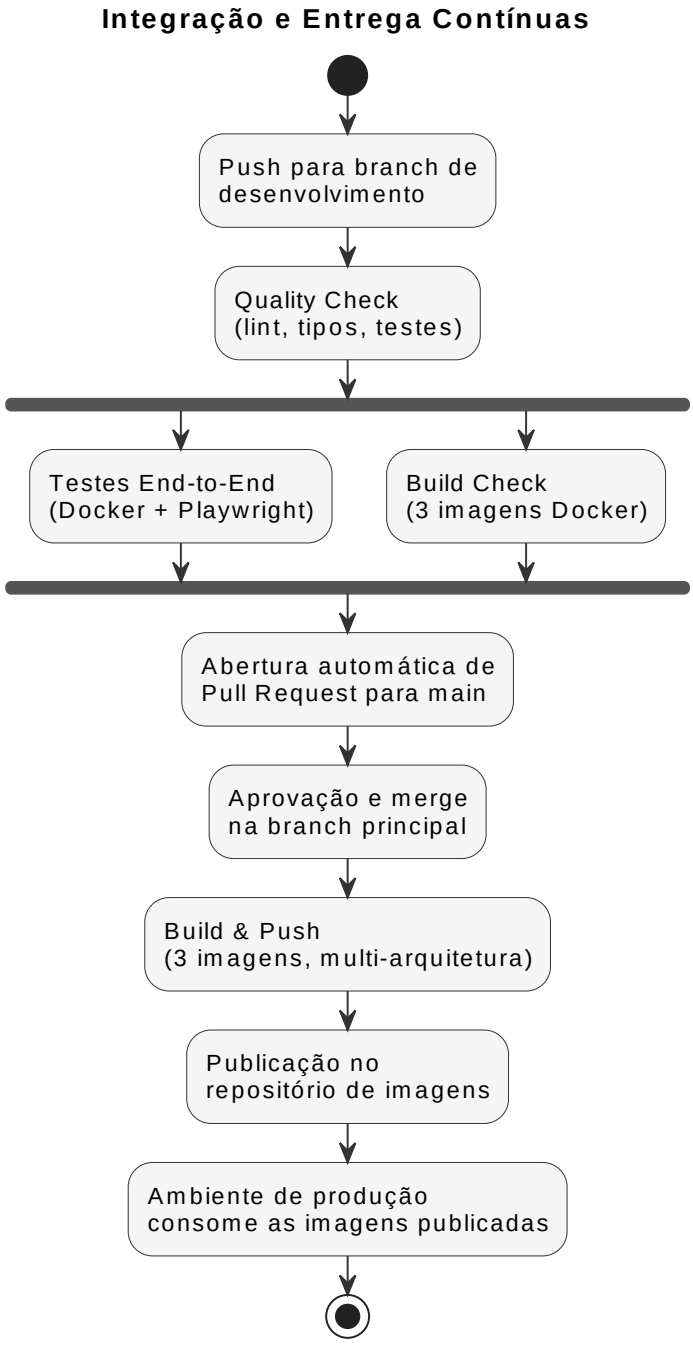
Fonte: O autor (2026)

Figura 8 – Diagrama de sequência: Geração de Guia de Conteúdo, Síntese via LLM e Persistência



Fonte: O autor (2026)

Figura 9 – Diagrama de fluxo: Integração e Entrega Contínuas



Fonte: O autor (2026)

4 Implementação

Definida a arquitetura e a especificação do sistema, este capítulo mostra o resultado concreto dessas decisões. Apresentam-se as principais telas construídas, um fluxo real de uso, do dado bruto ao guia final, e as decisões e trade-offs de arquitetura assumidos ao longo do desenvolvimento, base sobre a qual os resultados discutidos no capítulo seguinte se apoiam.

4.1 Telas e Interface do Sistema

Esta seção apresenta as principais telas do sistema, cobrindo a página pública, o painel do usuário logado, a geração de conteúdo e a visualização do resultado gerado. As telas foram escolhidas por concentrarem o núcleo da proposta do trabalho, deixando de fora fluxos secundários já descritos nos requisitos funcionais, como autenticação e gerenciamento de nichos.

A Figura 10 apresenta a página inicial pública, primeiro contato do visitante com o sistema. Nela, a proposta central é comunicada por meio de uma prévia do guia gerado, dividida nas mesmas fases em que o resultado final é organizado.

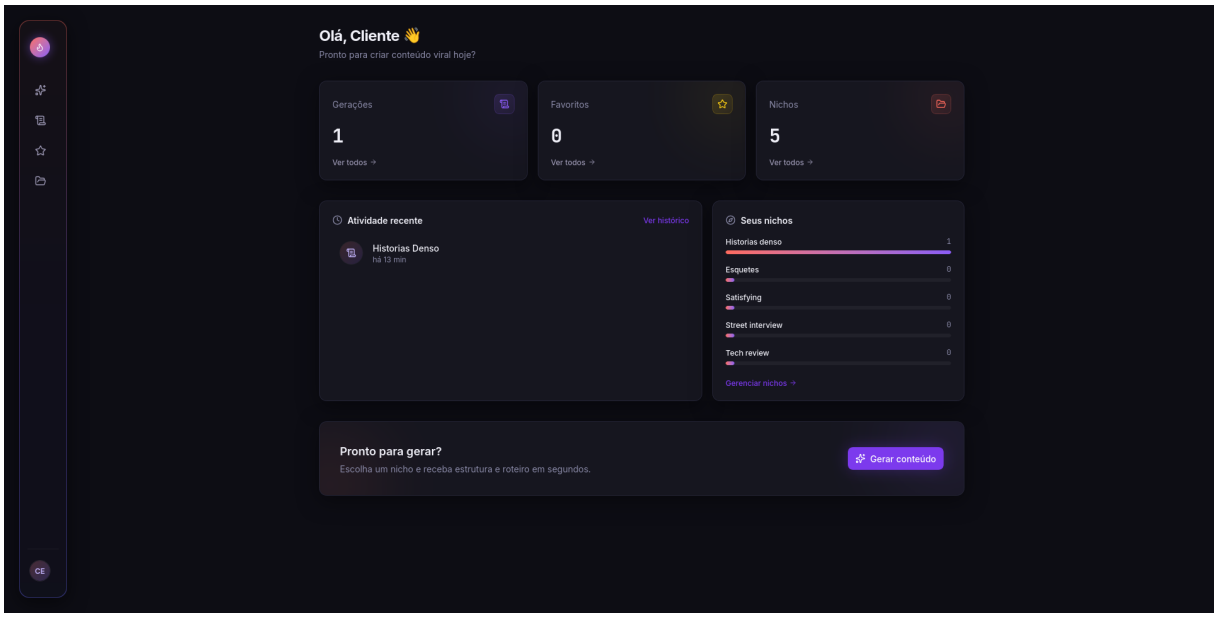
Já logado, o usuário é recebido pelo painel principal, exibido na Figura 11. A tela reúne, num mesmo ponto, um resumo numérico das gerações, favoritos e nichos vinculados à conta, a lista das gerações mais recentes e a distribuição de gerações entre os nichos vinculados, funcionando como visão geral da atividade do usuário no sistema.

A Figura 12 mostra a tela de geração, onde o usuário seleciona um nicho vinculado e, opcionalmente, um público-alvo, para solicitar um novo guia de conteúdo. É a partir dessa seleção que o serviço de processamento de linguagem natural é acionado, retornando a estrutura e o roteiro que compõem o guia.

Figura 10 – Página inicial

Fonte: O autor (2026)

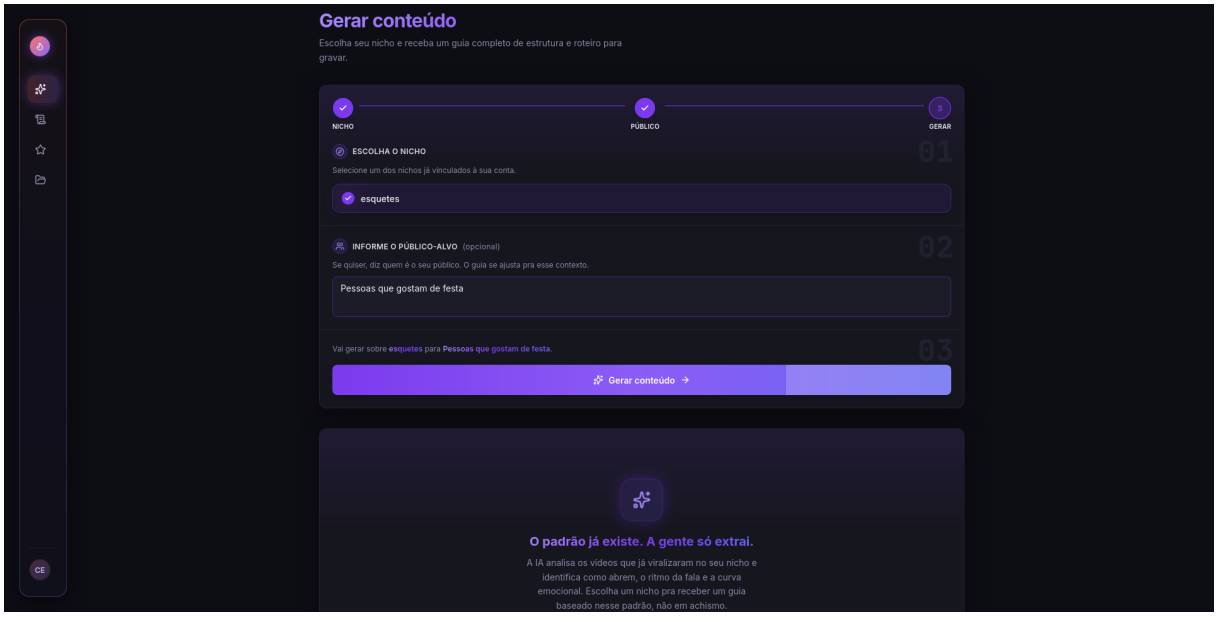
Figura 11 – Painel principal



Fonte: O autor (2026)

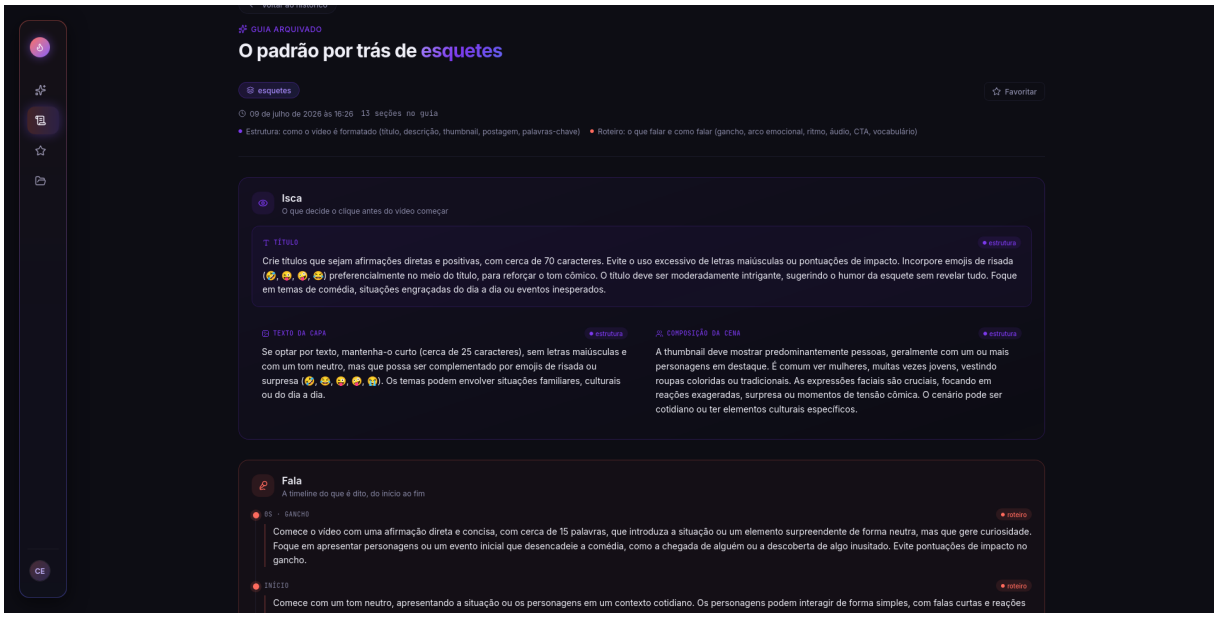
Por fim, a Figura 13 apresenta o detalhamento de uma geração no histórico. O resultado é organizado em fases, cada uma reunindo os elementos do guia relacionados a um aspecto específico da produção do vídeo, da concepção do gancho inicial até as recomendações de alcance e postagem.

Figura 12 – Tela de geração



Fonte: O autor (2026)

Figura 13 – Detalhamento de uma geração no histórico



Fonte: O autor (2026)

4.2 Fluxo Ponta a Ponta / Estudo de Caso

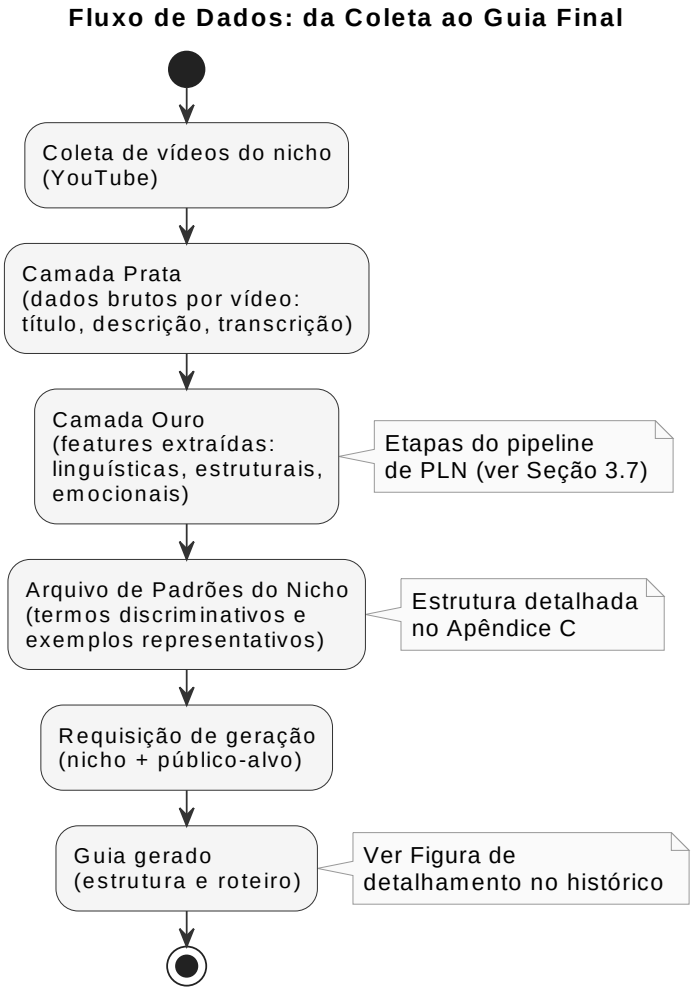
Esta seção acompanha o percurso de um nicho real através do sistema, da coleta do dado bruto até o guia entregue ao usuário, ilustrado na Figura 14.

Os vídeos coletados do nicho compõem a camada prata, com o dado bruto de cada vídeo, como título, descrição e transcrição. O pipeline de processamento de linguagem natural, descrito na Seção 3.7, transforma esse dado na camada

ouro, com as features linguísticas, estruturais e emocionais extraídas de cada vídeo. A partir dessa camada, o pipeline consolida um arquivo de padrões próprio do nicho, reunindo os termos discriminativos e os exemplos mais representativos identificados, cuja estrutura é detalhada no Apêndice B.

É esse arquivo de padrões que orienta cada nova geração solicitada por um usuário vinculado ao nicho. Ao selecionar o nicho e, opcionalmente, um público-alvo, o sistema utiliza esse padrão como evidência para compor o guia de estrutura e roteiro entregue ao final, apresentado anteriormente na Figura 13.

Figura 14 – Fluxo de dados: da coleta ao guia final



Fonte: O autor (2026)

4.3 Decisões e Trade-offs

A divisão em três serviços independentes, apresentada na Seção 3.5, não é uma decisão sem custo. Cada serviço passa a exigir sua própria configuração de build, execução e implantação, além da orquestração entre eles, elevando a complexidade operacional do sistema em relação a uma aplicação única.

Em contrapartida, essa divisão permite que cada serviço evolua e seja implantado de forma independente. Uma mudança restrita ao serviço de geração de conteúdo, por exemplo, não exige reconstruir nem reiniciar o serviço de interface ou o de autenticação, o que tende a reduzir o tempo de indisponibilidade em atualizações pontuais (SAI et al., 2025).

Essa divisão também abre a possibilidade de escalar cada serviço de forma independente, direcionando mais recursos especificamente ao ponto de maior demanda, em vez de replicar a aplicação inteira (SAI et al., 2025). No estágio atual do projeto, essa vantagem permanece potencial: sem tráfego real em produção, o ganho de escalabilidade seletiva ainda não se traduziu em necessidade concreta, mas a arquitetura já está preparada para absorvê-la caso surja.

5 Resultados

Construído e implantado o sistema, resta avaliar o que ele de fato produz. Este capítulo examina os guias gerados para diferentes nichos, discutindo em que medida atendem à proposta do trabalho, e as limitações observadas ao longo desse processo, servindo de base para as considerações finais apresentadas no capítulo seguinte.

5.1 Validação dos Guias Gerados

Esta seção avalia qualitativamente os guias gerados para três nichos distintos, examinando se as instruções retornadas fazem sentido para o contexto de cada um. O Apêndice A traz a saída completa de uma dessas gerações.

O resultado mais relevante não está em nenhum guia isolado, mas na diferença entre eles. Os três nichos analisados produzem roteiros com propósitos comunicativos distintos, um cômico e autocontido, outro narrativo e prolongado, o terceiro quase silencioso e centrado na ação visual, e o sistema responde a essa diferença sem que qualquer regra específica de nicho tenha sido escrita manualmente. A orientação nasce inteiramente do padrão extraído dos vídeos que de fato viralizaram naquele contexto.

Essa correspondência se sustenta mesmo nos aspectos que, a princípio, pareceriam genéricos, como o momento recomendado de publicação e o equilíbrio entre palavras-chave amplas e específicas. Em vez de uma recomendação única aplicada a todo o sistema, cada guia reflete o comportamento observado dentro do próprio nicho, reforçando que a diferenciação não se limita ao conteúdo do roteiro, mas se estende à estratégia de distribuição sugerida.

A limitação inerente a essa validação é sua natureza qualitativa: sem uma métrica de desempenho real do conteúdo publicado, o julgamento de que um guia *faz sentido* para o nicho permanece uma avaliação de coerência

interna, não uma comprovação de eficácia. Essa lacuna é retomada na seção seguinte.

5.2 Discussão / Limitações Observadas

Nem todo componente do sistema opera com a mesma margem de folga observada em desenvolvimento local. Esta seção discute os pontos de atrito identificados ao longo do trabalho, concentrados sobretudo na dependência de serviços externos gratuitos.

Na etapa de coleta, a obtenção de transcrições depende de um scraper sujeito a limites de uso rígidos, o que obriga o espaçamento das requisições ao longo do tempo. Uma restrição semelhante ocorre na leitura de texto embutido nas capas dos vídeos, também dependente de um provedor externo gratuito, cujo uso em maior volume exige alternar entre múltiplas credenciais para não ser bloqueado. Em ambos os casos, o gargalo não está na lógica de coleta em si, mas na cota disponível do serviço de terceiro utilizado.

A tradução do texto coletado para o inglês, etapa anterior à extração de padrões, também nem sempre preserva o sentido original: termos que não deveriam ser traduzidos, como nomes próprios, ocasionalmente são tratados como palavras comuns do idioma de destino. Esse tipo de erro não é eliminado na origem; parte do tratamento posterior do dado, na camada ouro, existe justamente para mitigar esse tipo de ruído antes que ele chegue à extração de padrões.

A limitação mais significativa está na geração do guia em si. Cada solicitação do usuário depende de duas chamadas sequenciais ao modelo de linguagem, uma de limpeza e outra de geração final, o que soma um tempo de resposta que pode ultrapassar 50 segundos. Por operar no nível gratuito do provedor, o serviço está sujeito a instabilidade sob alta demanda, retornando erro de indisponibilidade nesses momentos, sem mecanismo de nova tentativa implementado. A alternativa testada durante o desenvolvimento, um provedor

de inferência mais rápido, reduziu esse tempo de resposta às custas de uma qualidade de saída perceptivelmente inferior, o que motivou a manutenção do provedor original apesar de sua instabilidade.

6 Conclusão

Avaliados os resultados obtidos no capítulo anterior, resta fechar o trabalho retomando o que foi de fato entregue. Este capítulo sintetiza os objetivos alcançados, situa as limitações que restringiram o escopo do sistema construído e aponta as direções de continuidade mais imediatas.

6.1 Síntese

O objetivo geral deste trabalho foi cumprido. O sistema desenvolvido extrai padrões de conteúdo viral em vídeos curtos, segmentados por nicho, e gera guias de conteúdo estratégico a partir desses padrões. O funcionamento de ponta a ponta, da coleta de dados brutos à entrega do guia, foi confirmado na avaliação apresentada no Capítulo 5.

Os objetivos específicos também foram atingidos. A coleta de metadados e transcrições por nicho formou a camada prata do conjunto de dados. O processamento de linguagem natural e o agrupamento não supervisionado do formato narrativo refinaram essa base na camada ouro, conforme descrito na Seção 3.7. A partir dela, o agrupamento baseado em densidade consolidou os padrões de cada nicho, com termos discriminativos e exemplos representativos extraídos.

O mecanismo de similaridade semântica resolve nichos ainda sem padrões extraídos, sustentando a arquitetura apresentada na Seção 3.5. Os guias gerados, estruturados em abertura, desenvolvimento e fechamento, foram validados qualitativamente para três nichos na Seção 5.1. O resultado reflete o padrão de cada nicho, não uma orientação genérica. Por fim, a aplicação web *full-stack* que expõe esse serviço foi apresentada no Capítulo 4.

6.2 Limitações

Além dos pontos operacionais discutidos na Seção 5.2, algumas decisões de escopo, motivadas por tempo e recurso, delimitam o que o sistema cobre.

A coleta de dados é restrita ao YouTube. O sistema não foi testado em outras plataformas de vídeo curto, como TikTok e Reels. A generalização do pipeline de extração de padrões para elas permanece uma hipótese não verificada.

A validação apresentada na Seção 5.1 é interna e qualitativa, feita pelo próprio autor. O trabalho não envolveu criadores reais avaliando os guias em suas próprias publicações. Por isso, a conclusão de que um guia *faz sentido* para o nicho fica restrita à coerência interna, sem comprovação de ganho real de desempenho.

O voto de utilidade enviado pelo usuário para a estrutura e o roteiro de uma geração (RF21) não é devolvido pela API junto com a própria geração. Isso difere do que ocorre com o estado de favorito, que é retornado normalmente. Como consequência, o voto já registrado não aparece na interface após uma atualização de página, mesmo estando persistido no banco. O usuário pode acabar votando de novo no mesmo item, sem ver o que havia escolhido antes.

6.3 Trabalhos Futuros

A direção mais prioritária de continuidade é a validação do conteúdo gerado junto a criadores reais. A avaliação atual, apresentada na Seção 5.1, é interna e qualitativa. Um próximo passo natural é acompanhar o desempenho de vídeos publicados a partir dos guias gerados, medindo alcance e engajamento reais.

Uma segunda direção é expandir a coleta para outras plataformas de vídeo curto, como TikTok e Reels. Isso exigiria adaptar a etapa de coleta a cada API, mas reaproveitaria o restante do pipeline sem alteração, já que ele opera sobre texto normalizado em inglês. A questão de fundo é testar se os padrões extraídos generalizam entre plataformas.

No curto prazo, a persistência do voto de feedback (Seção 6.2) é um ajuste concreto e já mapeado: basta incluir o voto do usuário na resposta da geração, junto ao que já ocorre com o estado de favorito.

Referências

AGUS, P. et al. Node.js performance benchmarking and analysis at virtualbox, docker, and podman environment using node-bench method. *JOIV : International Journal on Informatics Visualization*, 2023. 32

ALA'ANZY, M. A.; YESHPATOV, Z. A performance and scalability evaluation of monolithic (django) vs. microservice (fastapi) architectures for asynchronous api workloads in python. In: *2026 18th International Conference on Electronics, Computer, and Computation (ICECCO)*. [S.l.: s.n.], 2026. p. 1–6. 32

CARBONELL, J.; GOLDSTEIN-STEWART, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. p. 335–336, 1998. 32

CITO, J.; GALL, H. Using docker containers to improve reproducibility in software engineering research. In: *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. [S.l.: s.n.], 2016. p. 906–907. 34

CLAPTON, J. et al. Client-side vs server-side rendering: Impacts on performance and user experience in web applications. p. 165–172, 2025. 32

COOK, W.; RAI, S. Safe query objects: statically typed objects as remotely executable queries. In: *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.* [S.l.: s.n.], 2005. p. 97–106. 32

CRISTIN, P. et al. Performance comparison of embeddings and keyword selection methods in enterprise document. *Journal of Applied Informatics and Computing*, 2025. 16

HEORHII, H. Visual storytelling without dialogue: Techniques of emotional framing in short video. *Universal Library of Innovative Research and Studies*, 2025. 10

JANI, Y. Implementing continuous integration and continuous deployment (ci/cd) in modern software development. *International Journal of Science and Research (IJSR)*, 2023. 15

JASPAN, C. et al. Advantages and disadvantages of a monolithic repository: A case study at google. In: *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. [S.l.: s.n.], 2017. p. 225–234. 15

KORDZADEH, N.; YOUNG, D. K. How social media analytics can inform content strategies. *Journal of Computer Information Systems*, v. 62, p. 128 – 140, 2020. 11

LIND, F. et al. Greasing the wheels for comparative communication research: Supervised text classification for multilingual corpora. *Computational Communication Research*, 2021. 30

LUO, X. Cross-cultural adaptation framework for enhancing large language model outputs in multilingual contexts. *Journal of Advanced Computing Systems*, 2023. 14

MCINNES, L.; HEALY, J. Accelerated hierarchical density based clustering. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. [S.l.: s.n.], 2017. p. 33–42. 31

MCINNES, L.; HEALY, J.; ASTELS, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, v. 2, p. 205, 2017. 16

NAZáRIO, M. et al. Strategies to mitigate configuration differences in software development: A rapid review of grey literature. *J. Softw. Eng. Res. Dev.*, v. 13, 2025. 15

RADZIWILL, N.; FREEMAN, G. Reframing the test pyramid for digitally transformed organizations. *ArXiv*, abs/2011.00655, 2020. 15

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: *ArXiv*. [S.l.: s.n.], 2019. abs/1908.10084. 15, 31

ROBERTS, J. A.; DAVID, M. E. Technology affordances, social media engagement, and social media addiction: An investigation of tiktok, instagram reels, and youtube shorts. *Cyberpsychology, Behavior, and Social Networking*, v. 28, p. 318 – 325, 2025. 10

SAI, S. et al. Microservices architecture in e-commerce: A comparative analysis of performance, scalability, and maintainability. *International Journal for Research Publication and Seminar*, 2025. 43

SAWADOGO, P. N.; DARMONT, J. On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, v. 56, p. 97 – 120, 2020. 25

SWAMI, S. et al. Optimizing social media management and content marketing: A strategic approach to meeting organic digital demands. In: *2025 International Conference on Visual Analytics and Data Visualization (ICVADV)*. [S.l.: s.n.], 2025. p. 50–57. 12

WAN, X. et al. Application deployment using microservice and docker containers: Framework and optimization. *J. Netw. Comput. Appl.*, v. 119, p. 97–109, 2018. 15

WANG, X.; NING, H. *TF-IDF Keyword Extraction Method Combining Context and Semantic Classification*. [S.l.]: Proceedings of the 3rd International Conference on Data Science and Information Technology, 2020. 31

WANG, Y. et al. Test automation maturity improves product quality - quantitative study of open source projects using continuous integration. *ArXiv*, abs/2202.04068, 2022. 33

APÊNDICE A – Exemplo de Guia Gerado

Este apêndice apresenta a saída completa e real de uma geração para o nicho de esquetes, referenciada na Seção 5.1, no schema entregue pelo serviço de IA ao usuário final.


```
{
  "nicho_solicitado": "esquetes",
  "nicho_utilizado": "esquetes",
  "fallback_usado": false,
  "structure_content": {
    "titulo": {
      "instrucao": "Crie titulos que sejam afirmacoes diretas e
         engracadas, com aproximadamente 70 caracteres. Evite o uso
         de CAPS LOCK e pontuacao de impacto excessiva. O tom deve
         ser positivo e divertido, com um uso moderado de elementos
         que gerem curiosidade. Inclua emojis de risada (como
         amarelinhos rindo) no meio ou no inicio do titulo para
         reforcar o humor, focando em situacoes comicas, reacoes
         inesperadas ou travessuras do cotidiano."
    },
    "descricao": {
      "instrucao": "Elabore uma descricao concisa, com cerca de 195
         caracteres. A primeira linha, com ate 89 caracteres, deve
         ser neutra e contextualizar brevemente o video. Incorpore
         termos relevantes para SEO relacionados a humor, comedia,
         videos curtos e viralidade. Evite chamadas explicitas para
         acao (CTAs) e a inclusao de links. Utilize emojis de humor,
         agradecimento ou afeto para complementar o texto.",
      "hashtags": ["#EsquetesDeHumor", "#ComediaCurta", "#
         VideosEngracados", "#ShortsDeComedia", "#HumorViral", "#
         RisadasGarantidas", "#MemesBrasil", "#ConteudoDivertido", "#
         Pegadinhas", "#FamiliaEngracada", "#ReacoesEngracadas", "#
         CenasComicas", "#DiaADiaEngracado", "#EntretenimentoCurto",
```

```
        "#Viralizou"],
    "quantidade_hashtags": 15
  },
  "thumbnail": {
    "texto_capa_instrucao": "Se optar por incluir texto na capa,
       mantenha-o curto, com ate 26 caracteres. O tom deve ser
       neutro, sem o uso de CAPS LOCK, e pode incluir emojis de
       humor ou surpresa. O conteudo do texto deve complementar a
       cena visual, talvez indicando um tema de relacionamento ou
       cotidiano.",
    "cena_instrucao": "A thumbnail deve apresentar uma cena visual
       com forte presenca humana, geralmente com uma ou mais
       pessoas. As expressoes faciais devem ser neutras ou de
       surpresa, capturando um momento chave da esquete. A
       ambientacao pode incluir elementos do cotidiano ou
       vestimentas coloridas/tradicionais, focando em um visual que
        sugira uma situacao inusitada ou engracada."
  },
  "palavras_chave": {
    "instrucao": "Priorize o uso de palavras-chave de cauda curta (
       aproximadamente 80% do total) para alcancar um publico amplo
       . Complemente com palavras-chave de cauda media (cerca de
       12%) para segmentar um pouco mais o publico interessado em
       comedia e videos curtos. Palavras-chave de cauda longa (
       cerca de 7%) sao menos comuns, mas podem ser usadas para
       atrair buscas muito especificas.",
    "shorttail": ["Humor", "Comedia", "Esquetes", "Engracado", "
       Viral", "Shorts", "Risada", "Divertido", "Familia", "
       Tendencia"],
    "midtail": ["Video de comedia", "Esquete engracada", "Humor
       familiar", "Shorts virais", "Conteudo divertido"],
    "longtail": ["Melhores esquetes de humor", "Videos engracados
       para rir", "Comedia curta viral", "Novas piadas engracadas
       "],
```

```
      "proporcao_ideal": {"shorttail": 0.81, "midtail": 0.12, "
         longtail": 0.07}
    },
    "postagem": {
      "instrucao": "O periodo ideal para postar seus videos de
         esquetes e maximizar as chances de viralidade e no final da
         semana. Concentre suas postagens nas sextas-feiras, tanto no
          periodo da manha quanto a tarde, e tambem nas tardes de
         sabado. Esses horarios demonstraram a maior ocorrencia de
         videos virais neste nicho.",
      "slots": [
        {"dia": "Sexta-feira", "janela": "Manha", "justificativa": "
           Este e o horario com a maior concentracao de videos virais
           , indicando um alto engajamento."},
        {"dia": "Sexta-feira", "janela": "Tarde", "justificativa": "
           Um periodo de bom desempenho, capturando o publico que ja
           esta no clima de fim de semana."},
        {"dia": "Sabado", "janela": "Tarde", "justificativa": "
           Excelente para alcancar o publico que busca entretenimento
            durante o fim de semana."}
      ]
    }
  },
  "script_content": {
    "estrutura": {
      "instrucao": "Seu video deve ser uma esquete ultra curta,
         focada em uma unica situacao comica. A comunicacao principal
          nao se baseia em dialogos extensos entre varios personagens
         , mas pode envolver um narrador, um monologo ou reacoes
         visuais e sonoras. A estrutura deve ser concisa, sem que a
         repeticao da ideia central seja o elemento estrutural
         principal. Geralmente, ha um ou mais personagens humanos em
         cena, reagindo a eventos.",
      "limite_palavras": 0
```

```
},
"gancho": {
  "instrucao": "Abra o video com uma afirmacao direta ou uma
     situacao que introduza um elemento inesperado ou um problema
     . O gancho deve ser conciso, com cerca de 15 palavras, e ter
      um tom neutro, mas intrigante. Foque em cenarios do
     cotidiano ou interacoes familiares que rapidamente levem a
     uma reviravolta ou reacao comica."
},
"ritmo": {
  "instrucao": "O ritmo da fala deve ser moderado a lento, com
     frases de comprimento consider vel. No entanto, o mais
     importante e a forte variacao de ritmo ao longo do video,
     alternando momentos mais lentos e pausados com outros mais
     rapidos e intensos para criar impacto comico. Evite o uso
     excessivo de pontuacao de impacto no roteiro."
},
"arco_emocional": {
  "inicio": {
    "instrucao": "Comece o video com uma situacao cotidiana ou
       uma interacao familiar que pareca normal, mas que contenha
        um elemento de introducao a algo inusitado. Os
       personagens devem se apresentar de forma neutra, talvez
       com uma leve curiosidade ou expectativa, preparando o
       terreno para a comedia que se seguira."
  },
  "meio": {
    "instrucao": "No meio do video, a situacao se desenvolve com
       a introducao de um problema, uma surpresa ou uma revelacao
       . Os personagens devem reagir de forma exagerada ou
       inesperada a esse elemento, criando o ponto alto da
       comedia. As emocoes podem escalar para o desespero comico
       ou a confusao, mantendo um tom geral neutro, mas com picos
        de intensidade emocional."
```

```
    },
    "fim": {
      "instrucao": "Conclua o video com a resolucao comica da
         situacao, ou com uma reacao final que ressalte o absurdo
         do que aconteceu. Os personagens podem expressar uma
         aceitacao resignada, uma frustracao engracada ou uma
         surpresa continua. O final deve ser abrupto e deixar o
         espectador com uma sensacao de diversao e leveza, sem a
         necessidade de uma conclusao emocional complexa."
    }
  },
  "audio": {
    "instrucao": "Utilize musica de fundo de forma moderada,
       presente em cerca de 28% dos videos. O clima sonoro
       predominante deve ser de trilha sonora que complemente o
       humor, podendo incluir efeitos de risada ou sons de surpresa
       /grito em momentos de climax comico. A musica deve ser mais
       um elemento de apoio do que o foco principal.",
    "tipo_dominante": "Musica de fundo",
    "clima_sonoro": "Trilha sonora leve e efeitos de risada/
       surpresa",
    "frequencia_uso_no_nicho": 0.282
  },
  "repeticao": {
    "instrucao": "A repeticao de elementos ou ideias e uma
       estrategia que pode ser ativada em cerca de um terco dos
       videos do nicho. Quando usada, repita a ideia central por
       volta de 8 vezes ao longo do video para reforcar o humor.
       Varie a forma como a repeticao e apresentada para manter o
       engajamento, seja com pequenas alteracoes na fala, nas
       reacoes dos personagens ou na situacao.",
    "ativar": true,
    "taxa_no_nicho": 0.331,
    "frequencia_media": 8
```

```
    },
    "cta": {
      "instrucao": "Este nicho raramente utiliza chamadas explicitas
         para acao (CTAs). O video deve encerrar naturalmente dentro
         da narrativa da esquete, sem pedidos diretos para curtir,
         comentar ou seguir. O final deve ser a conclusao comica da
         historia, deixando o publico com uma boa impressao e o
         desejo de assistir mais, sem interrupcoes para uma acao."
    },
    "vocabulario_geral": {
      "instrucao": "Utilize um vocabulario que remeta a interacoes
         familiares e situacoes cotidianas. A linguagem deve ser
         informal e direta, com termos que sugiram convites,
         surpresas, negacoes e reacoes comuns em ambientes domesticos
          ou sociais. Foque em palavras que criem um senso de
         familiaridade e humor situacional.",
      "termos_recomendados": ["irma", "vem", "sim", "ei", "comer", "
         irmao", "sogra", "trouxe", "nao", "filho", "filha", "olha",
         "dizer", "deixa", "festa", "mae", "gostar", "agua", "
         diversao", "rapido", "ok", "senhor", "pouco", "espera"],
      "coocorrencias": [["vem", "irma"], ["filha", "sogra"], ["vem",
         "deixa"], ["pegar", "sim"], ["sogra", "mae"], ["irmao", "
         sogra"], ["gostar", "dizer"]]
    }
  }
}
```

APÊNDICE B – Estrutura do JSON de Padrões por Nicho

Este apêndice apresenta um recorte representativo da estrutura de dados consolidada pelo pipeline de extração de padrões (Seção 3.7), consultada pelo serviço de IA no momento da geração de um guia.


```
{
  "nicho": "tech_review",
  "structure_content": {
    "titulo": {
      "formula_construcao": "afirmacao",
      "comprimento_alvo_chars": 55,
      "termos_obrigatorios": ["ultra", "phone", "samsung", "galaxy",
         "iphone"],
      "caps_intensidade": 0.0,
      "emojis": [
        {"emoji": "U+1F4F1", "posicao": "meio", "frequencia": 3},
        {"emoji": "U+26A1", "posicao": "meio", "frequencia": 3}
      ],
      "sentimento_alvo": "neutro"
    }
  },
  "script_content": {
    "gancho": {
      "exemplos_reais": [
        "this phone changed everything for me"
      ]
    }
  }
}
```