

**UNIVERSIDADE FEDERAL DE PELOTAS**  
**Centro de Desenvolvimento Tecnológico**  
**Curso de Bacharelado em Engenharia de Computação**



Trabalho de Conclusão de Curso

**VideoLearnAI: LLM Powered Web Application para aprendizagem ativa com  
vídeos do Youtube**

**Kevin Castro Weitgenant**

Pelotas, 2025

**Kevin Castro Weitgenant**

**VideoLearnAI: LLM Powered Web Application para aprendizagem ativa com vídeos do Youtube**

Trabalho de Conclusão de Curso apresentado ao Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Tiago Primo  
Coorientador: Prof. Dr. Marilton Sanchotene de Aguiar

Pelotas, 2025

**Insira AQUI a ficha catalográfica  
Quando finalizado o trabalho, deve ser  
solicitada através do Sistema Cobalto  
Biblioteca – Cadastro – Ficha catalográfica.**

**Kevin Castro Weitgenant**

**VideoLearnAI: LLM Powered Web Application para aprendizagem ativa com vídeos do Youtube**

Trabalho de Conclusão de Curso aprovado, como requisito parcial, para obtenção do grau de Bacharel em Engenharia de Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

**Data da Defesa:** 16 de março de 2025

**Banca Examinadora:**

Prof. Dr. Marilton Sanchotene de Aguiar (orientador)  
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Paulo Roberto Ferreira Jr.  
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Ricardo Matsumura Araujo  
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Luciano da Silva Pinto  
Doutor em Biotecnologia pela Universidade Federal de Pelotas.

Dedico...

## **AGRADECIMENTOS**

Agradeço...

*Só sei que nada sei.*

— SÓCRATES

## RESUMO

WEITGENANT, Kevin Castro. **VideoLearnAI: LLM Powered Web Application para aprendizagem ativa com vídeos do Youtube**. Orientador: Tiago Primo. 2025. 31 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

Este trabalho apresenta o desenvolvimento de uma plataforma educacional como serviço (SaaS) que utiliza Inteligência Artificial para aprimorar a experiência de aprendizagem com conteúdo em vídeo. O sistema implementa cinco funcionalidades principais: melhoria automática da legibilidade de legendas, geração de capítulos, transcrição sincronizada, geração de quizzes interativos e um sistema de bate-papo contextual com o conteúdo do vídeo. A solução emprega Large Language Models (LLMs) e arquitetura Transformer para processar e transformar o conteúdo audiovisual em material educacional interativo. A implementação foi realizada com foco em escalabilidade e performance, utilizando processamento em GPU e técnicas modernas de desenvolvimento de software. Os resultados demonstram o potencial da plataforma para transformar vídeos em experiências de aprendizagem mais engajadoras e efetivas.

Palavras-chave:      palavrachave-um;      palavrachave-dois;      palavrachave-tres;  
palavrachave-quatro.



## RESUMO

WEITGENANT, Kevin Castro. **AI-Powered Educational Platform: Transforming Video Content into Interactive Learning Experiences**. Orientador: Tiago Primo. 2025. 31 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

This work presents the development of an educational Software as a Service (SaaS) platform that leverages Artificial Intelligence to enhance video-based learning experiences. The system implements five main functionalities: automatic subtitle readability improvement, chapter generation, synchronized transcription, interactive quiz generation, and a contextual chat system for video content. The solution employs Large Language Models (LLMs) and Transformer architecture to process and transform audiovisual content into interactive educational material. The implementation focused on scalability and performance, utilizing GPU processing and modern software development techniques. The results demonstrate the platform's potential for transforming videos into more engaging and effective learning experiences.

Palavras-chave: keyword-one; keyword-two; keyword-three; keyword-four.

## **LISTA DE FIGURAS**

## **LISTA DE TABELAS**

## **LISTA DE ABREVIATURAS E SIGLAS**

ABNT	Associação Brasileira de Normas Técnicas
NUMA	Non-Uniform Memory Access
SIMD	Single Instruction Multiple Data
SMP	Symmetric Multi-Processor
SPMD	Single Program Multiple Data

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	15
1.1	Objetivo geral	16
1.2	Objetivos específicos	16
1.3	Estrutura do trabalho	16
<b>2</b>	<b>SOLUÇÕES RELACIONADAS</b>	18
2.1	—	18
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
3.1	Processo de Desenvolvimento	19
3.1.1	Definição de Requisitos	19
3.1.2	Prototipação	19
3.1.3	Desenvolvimento Iterativo	19
3.2	Arquitetura e Tecnologias	19
3.2.1	Tecnologias Usadas	19
3.2.2	Visão Geral da Arquitetura	19
<b>4</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO</b>	20
4.1	Melhoria da Legibilidade das Legendas	20
4.1.1	O Problema da Legibilidade	20
4.1.2	Primeira Abordagem com LLMs	20
4.1.3	Implementação com Transformers	21
4.1.4	Comparação e Resultados	21
	<b>REFERÊNCIAS</b>	22
4.2	Geração de Capítulos	22
4.2.1	Algoritmo e Implementação	22
4.2.2	Integração com LLMs	22
4.3	Transcrição	22
4.3.1	Processamento de Áudio	22
4.3.2	Sincronização e Refinamento	22
4.4	Geração de Quizzes	22
4.4.1	Extração de Conceitos-Chave	22
4.4.2	Geração via LLM	22
4.5	Bate-Papo com Vídeo	22
4.5.1	Processamento de Perguntas	22
4.5.2	Contextualização com Conteúdo	22
4.5.3	Geração de Respostas	22

<b>4.6</b>	<b>Desafios e Soluções</b>	22
4.6.1	Obtenção dos dados do youtube	22
4.6.2	Utilização de GPU's em produção	22
<b>5</b>	<b>RESULTADOS</b>	23
5.1	Análise de Desempenho	23
5.2	Feedback dos Usuários	23
<b>6</b>	<b>CONCLUSÃO</b>	24
6.1	Objetivos Alcançados	24
6.2	Trabalhos Futuros	24
<b>7</b>	<b>REFERÊNCIAS</b>	25
	REFERÊNCIAS	26
	APÊNDICE A UM APÊNDICE	28
	ANEXO A UM ANEXO	30
	ANEXO B OUTRO ANEXO	31

# 1 INTRODUÇÃO

Nos últimos anos, o consumo de conteúdo educacional em vídeo tem crescido exponencialmente, impulsionado por plataformas como YouTube, Coursera e Udemy. Hoje, é possível encontrar aulas completas de universidades de altíssimo nível, como MIT, Harvard e Stanford, gratuitamente disponíveis online. No entanto, apesar da abundância de material de qualidade, muitos usuários enfrentam dificuldades em absorver e reter conhecimento de forma eficiente. A maioria das pessoas consome esses conteúdos de maneira passiva, apenas assistindo aos vídeos sem um envolvimento ativo com o material. Isso limita a retenção e a compreensão das informações.

A aprendizagem ativa, por outro lado, é um modelo comprovadamente mais eficaz, pois envolve o estudante em processos como resumo, questionamento, reorganização do conteúdo e interação com o material. Pesquisas mostram que métodos ativos de estudo, como fazer perguntas sobre o conteúdo, testar-se frequentemente e organizar a informação de forma estruturada, levam a um aprendizado mais profundo e duradouro.

Diante desse cenário, este trabalho apresenta o desenvolvimento de um Software as a Service (SaaS) voltado para transformar o consumo passivo de vídeos educacionais em um processo de aprendizagem ativa. A solução utiliza modelos de linguagem natural (LLMs) para reestruturar legendas em textos mais legíveis, gerar capítulos automáticos, fornecer resumos e permitir interações como perguntas e respostas sobre o conteúdo. Além disso, o sistema oferece quizzes dinâmicos para reforçar o aprendizado e um mecanismo para salvar o progresso dos usuários, incentivando um envolvimento mais estruturado com os vídeos.

O desenvolvimento do SaaS seguiu uma abordagem iterativa. A arquitetura da aplicação integra tecnologias como FastAPI, Next.js e Transformers, além de estratégias de otimização para garantir eficiência e escalabilidade. A validação da ferramenta inclui métricas de desempenho e feedback dos usuários, avaliando sua eficácia na melhoria da compreensão e retenção do conhecimento.

Com esta pesquisa, buscamos não apenas oferecer uma ferramenta inovadora para aprendizado com vídeos, mas também contribuir para a democratização da edu-

cação de qualidade, permitindo que qualquer pessoa tenha acesso a um método mais eficaz para extrair o máximo de conhecimento dos conteúdos disponíveis online.

## 1.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver um Software as a Service (SaaS) que transforme o consumo passivo de vídeos educacionais em um processo de aprendizagem ativa. Para isso, a plataforma utilizará inteligência artificial para melhorar a legibilidade das legendas, gerar resumos, estruturar conteúdos em capítulos, criar quizzes interativos e permitir interações diretas com o conteúdo por meio de perguntas e respostas. O foco é tornar o aprendizado com vídeos mais eficiente, estruturado e acessível, permitindo que qualquer pessoa aproveite melhor o vasto acervo educacional disponível online.

## 1.2 Objetivos específicos

Para atingir o objetivo geral, este trabalho busca:

- Desenvolver um sistema que reestruture legendas de vídeos em textos mais legíveis e organizados, facilitando a compreensão.
- Implementar um mecanismo para geração automática de capítulos e resumos, permitindo uma navegação mais eficiente pelo conteúdo.
- Criar um módulo de perguntas e respostas, possibilitando interações com o vídeo de forma contextualizada.
- Desenvolver um sistema de quizzes automáticos baseados no conteúdo dos vídeos, reforçando o aprendizado ativo.
- Implementar um sistema de salvamento de progresso para permitir que usuários retomem facilmente seus estudos.

## 1.3 Estrutura do trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 2 – Revisão da Literatura:** apresenta os conceitos fundamentais de aprendizagem ativa e modelos de linguagem natural (LLMs), que embasam o desenvolvimento da aplicação.
- **Capítulo 3 – Metodologia:** descreve o processo de desenvolvimento do SaaS, incluindo a definição de requisitos, prototipação, escolha de tecnologias e critérios de avaliação.



- **Capítulo 4 – Desenvolvimento da Aplicação:** detalha as funcionalidades do sistema, explicando a implementação de cada módulo e os desafios enfrentados.
- **Capítulo 5 – Resultados:** analisa o desempenho da aplicação e apresenta o feedback dos usuários, avaliando o impacto da ferramenta na experiência de aprendizado.
- **Capítulo 6 – Conclusão:** discute os objetivos alcançados, as principais contribuições do trabalho e sugestões para aprimoramentos futuros.

## **2 SOLUÇÕES RELACIONADAS**

### **2.1 —**

## **3 FUNDAMENTAÇÃO TEÓRICA**

### **3.1 Processo de Desenvolvimento**

#### **3.1.1 Definição de Requisitos**

#### **3.1.2 Prototipação**

#### **3.1.3 Desenvolvimento Iterativo**

### **3.2 Arquitetura e Tecnologias**

#### **3.2.1 Tecnologias Usadas**

#### **3.2.2 Visão Geral da Arquitetura**

## 4 DESENVOLVIMENTO DA APLICAÇÃO

### 4.1 Melhoria da Legibilidade das Legendas

#### 4.1.1 O Problema da Legibilidade

As legendas automáticas de vídeos frequentemente apresentam problemas de formatação e segmentação, dificultando a compreensão do conteúdo. Esse problema ocorre porque as transcrições brutas costumam ser geradas como um fluxo contínuo de palavras, sem uma estrutura clara de frases e parágrafos. Além disso, quebras de linha mal posicionadas afetam a fluidez da leitura.

Além de melhorar a legibilidade das legendas, tornar o conteúdo escrito mais organizado pode ser útil para quem prefere ler um vídeo em vez de assisti-lo. A leitura permite revisar rapidamente uma parte específica sem precisar voltar no vídeo, além de ajudar a decidir se vale a pena assistir aquele trecho ou se a informação já foi absorvida apenas lendo. Para algumas pessoas, ler pode ser simplesmente uma forma mais eficiente e rápida de consumir o conteúdo.

Para resolver essa questão, foram testadas abordagens baseadas em modelos de linguagem natural, buscando aprimorar a estrutura das legendas sem alterar seu conteúdo original.

#### 4.1.2 Primeira Abordagem com LLMs

Inicialmente, foi utilizado um modelo de linguagem de grande porte (LLM) para segmentar e melhorar a legibilidade do texto das legendas. A implementação foi feita utilizando a biblioteca `instructor`, que permite a validação do texto gerado através do `pydantic`, garantindo conformidade com um formato estruturado.

Entretanto, essa abordagem apresentou desafios significativos:

- **Latência elevada:** O tempo de resposta do modelo era relativamente alto, tornando a solução pouco eficiente para processar grandes quantidades de legendas.
- **Alterações não desejadas no texto:** Apesar das instruções explícitas no

prompt para evitar adições ou remoções de palavras, o modelo ocasionalmente incluía frases como "Aqui está o seu texto otimizado" ou alterava partes do conteúdo original.

Esses fatores tornaram a abordagem com LLMs menos viável para o problema proposto.

#### 4.1.3 Implementação com Transformers

Diante das limitações dos LLMs, foi testada uma alternativa baseada em transformers especializados para segmentação de texto. A ferramenta `wtpsplit` mostrou-se uma solução eficiente para a melhoria da legibilidade das legendas.

Diferente dos LLMs, `wtpsplit` foca especificamente na segmentação de frases, apresentando vantagens como:

- **Baixa latência:** A segmentação ocorre de maneira rápida e eficiente.
- **Preservação do conteúdo original:** O modelo não adiciona ou remove palavras arbitrariamente, garantindo fidelidade ao texto original.
- **Facilidade de implementação:** A integração do `wtpsplit` ao pipeline de processamento foi direta, proporcionando bons resultados sem necessidade de ajustes complexos nos prompts.

#### 4.1.4 Comparação e Resultados

Para avaliar o desempenho das abordagens, foram realizadas comparações entre os textos segmentados pelos LLMs e pelo `wtpsplit`. Os critérios analisados incluíram:

- Tempo de processamento
- Fidelidade ao texto original
- Legibilidade subjetiva (avaliação qualitativa)

Os resultados mostraram que `wtpsplit` proporcionou uma segmentação mais precisa e eficiente, superando os LLMs em termos de velocidade e fidelidade ao conteúdo original. Assim, a implementação final optou pelo uso de `wtpsplit` como solução principal para a melhoria da legibilidade das legendas.

## **REFERÊNCIAS**

### **4.2 Geração de Capítulos**

#### **4.2.1 Algoritmo e Implementação**

#### **4.2.2 Integração com LLMs**

### **4.3 Transcrição**

#### **4.3.1 Processamento de Áudio**

#### **4.3.2 Sincronização e Refinamento**

### **4.4 Geração de Quizzes**

#### **4.4.1 Extração de Conceitos-Chave**

#### **4.4.2 Geração via LLM**

### **4.5 Bate-Papo com Vídeo**

#### **4.5.1 Processamento de Perguntas**

#### **4.5.2 Contextualização com Conteúdo**

#### **4.5.3 Geração de Respostas**

### **4.6 Desafios e Soluções**

#### **4.6.1 Obtenção dos dados do youtube**

#### **4.6.2 Utilização de GPU's em produção**

## **5 RESULTADOS**

### **5.1 Análise de Desempenho**

### **5.2 Feedback dos Usuários**

## **6 CONCLUSÃO**

### **6.1 Objetivos Alcançados**

### **6.2 Trabalhos Futuros**



## **7 REFERÊNCIAS**

## REFERÊNCIAS

## **Apêndices**

## APÊNDICE A – Um Apêndice

## **Anexos**

## ANEXO A – Um Anexo

Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla  
blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla.  
Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla  
blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla.  
Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla  
blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla. Bla blabla blablabbia bla.

[illegible]

